UNIVERSITY OF CINCINNATI

Date: August 3, 2006

I, Masoud Ghaffari,

hereby submit this work as part of the requirements for the degree of:

Doctorate of Philosophy

in:

Industrial Engineering

It is entitled:

Perception-based Control for Intelligent Systems

This work and its defense approved by:

Chair: Dr. Ernest Hall

Dr. Richard Shell

Dr. Ash Genaidy

Dr. Ronald Huston

Dr. Ricardo Moena

Perception-based Control for Intelligent Systems

A dissertation submitted to the

Division of Research and Advanced Studies of the University of Cincinnati

in partial fulfillment of the requirements for the degree of

DOCTORATE OF PHILOSOPHY

in the Department of Mechanical, Industrial, and Nuclear Engineering
of the College of Engineering

2006

by

Masoud Ghaffari

B.S. in Electrical Eng., Isfahan Univ. of Technology, 1995

M.S. in Socio-Economic Systems Eng., Amirkabir Univ. of Technology, 1998

Committee Chair: Dr. Ernest L. Hall

ABSTRACT

Intelligent systems theory tries to study the most amazing feature of living creatures: intelligence. One active research area with many promising applications is autonomous navigation of unmanned vehicles which relies heavily on intelligent systems theory. The purpose of this dissertation is to apply an ambiguous concept in intelligent systems, called *perception,* in robot navigation.

Several approaches have been used to model perception for robot navigation. A learning framework, equipped with a perception-based task control center, has been proposed. A statistical approach for *uncertainty* modeling has been investigated as well. In addition, a spatial knowledge model was used to model robot navigation. Finally, an optimization approach toward perception was used to model robot design and navigation.

Several case studies of robot design will be presented. An unmanned ground vehicle, called the Bearcat Cub, was designed and developed for the Intelligent Ground Vehicle Competition (IGVC). This robot was used to demonstrate spatial knowledge modeling. In another design, a soil sampling survey robot was developed to measure the soil strength in remote areas. And finally, the design and development of a snow accumulation prevention robot will be presented. This autonomous robot can prevent accumulation of snow in areas such as driveways and small parking lots.

The implementation of unique hardware and software systems in several robotic systems, as well as promoting a multifaceted view of perception modeling, are significant contributions made by this dissertation. The proposed framework uses optimization approach; it has learning capability, and is able to handle *uncertain* situations that are common in robot navigation.

To

My family

Acknowledgment

I wish to express my appreciation toward my adviser Dr. Ernest Hall for all his guidance and support. Committee members Drs. Shell, Genaidy, Huston, and Moena, as well as Dr. Lee, have been my mentors and have assisted me in a variety of ways throughout my studies at the University of Cincinnati. I would like to express my gratitude to all of them.

Working with my fellow students and friends Sherry Liao, Souma Alhaj Ali, Peter Cao, Justin Gaylor, Mark McCrate, and all the members of the robotics team and IMS Center, has made my studies all the more enjoyable. I wish them the best. I especially would like to thank Patrick Brown and Leila Maybodi for their help in editing this work.

Finally, I am deeply indebted to Jalil, Karen, Alexander, and David for their love and support all these years. I wish to dedicate this dissertation to my family, especially my beloved mother and father to whom I owe the most.

Table of Contents

List of Figures

List of Tables

“If the doors of perception were cleansed every thing would appear to man as it is, infinite.”

William Blake (1757, 1827)

Chapter 1 : Introduction

"The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents."

H. P. Lovecraft (1890 - 1937)

The potential application areas for autonomous navigation of mobile robots include automatic driving, guidance for the blind and disabled, exploration of dangerous environments, transporting objects in factory or office environments, collecting geographical information in unknown terrains, such as unmanned exploration of a new planetary surface, and many others [1].

Intelligent robotics is an active area of research with many promising applications. The recent DARPA Grand Challenge, which involved autonomously driving for 132 miles, and the DARPA Urban Challenge are examples of such activities.

In fact, the US army spent more that 1 billion dollars on unmanned combat vehicles in 2004 alone [2].

Autonomous navigation in unknown environments is very difficult, and calls for sophisticated sensing strategies and control architectures [3]. Most current methods have basically been realized by combining algorithms acquired by humans. The real challenge is realizing the capability of acquiring algorithms automatically, much like the human brain does. This is a very difficult challenge, but it is the key to true autonomy [4].

Artificial intelligence, psychology, biology, neuroscience, linguistics, and many other fields try to gain insight into what is called human intelligence. This intelligence includes the ability to model the world, plan and predict events, learn, make decisions, as well as many other abilities.

The difficulty of autonomous navigation comes from the unstructured, unpredictable, dynamic environment of robots and the current technological restraints in the field of artificial intelligence. Most research in this area is experimental in nature and the development of real products is rare. Real world conditions are often quite hostile to robotic systems. Things can move and change without warning; at the best only partial knowledge of the world is available, and any prior information may be incorrect, inaccurate, or obsolete [5].

Making loops, backtracking, or visiting some areas more than once is common behavior for unmanned ground vehicles (UGVs) that lack the capacity for learning. When the robot visits a certain area several times, the information obtained can be used to improve the quality of navigation. However, in many real world applications it is not possible to have prior knowledge of the course and predict the dynamic environment.

One example is driving a car in a city. That is a place that human perception shows its extra ordinary capability.

1.1 Problem definition

Human and robot navigation have similarities and differences, but the basic navigation issues are the same. Therefore, general theoretical and analytical approaches dealing with navigation in both areas can be integrated, enabling both fields to benefit from each other. The difference is in how research in each area deals with the problem. In human navigation the main question is how a human processes information and what the mechanism is that enables humans and animals to navigate. However, robotics researchers are looking for techniques to implement navigational abilities in real applications, and whether or not it is biologically inspired is not an issue.

These two approaches are complementary. Biological systems are proof that an efficient and practical navigation system is achievable. On the other hand, robotics research can provide a valuable tool to test biological hypotheses. It can isolate a specific problem and examine psychological and biological questions surrounding it.

1.2 Study purpose

The purpose of this research is to study different facets of human perception and how they could possibly be applied to robot navigation. Taking an industrial engineering approach, this research will try to model human perception from several disciplinary points of view.

This research is a part of the studies aimed at human perception processing [6, 7]. The purpose of these studies is to develop a model for computing these perceptions and to implement a model for robot control that is influenced by human behavior. The goal is to enhance mobile robot technology by adding this perception-processing model to what is currently considered the state of the art, which mainly is based on converting the propositions to measurements and not computing perceptions directly. The proposed research tries to offer a complementary approach to these traditional methods. It also tries to explore similarities and differences in methods of navigation between humans and robots, and present a model for the perception-based navigation.

1.3 Significance

Intelligent mobile robotics is an active area of research with many promising applications. Any new development in this area could enhance the theory and application of mobile robots, intelligent systems and computer studies in artificial intelligence. In addition, the computational theory of perceptions is still in its infancy with many unanswered questions. This project could provide a better understanding of these two fields, and could explore the possibility and applicability of perception-based control in robotics.

1.4 Objectives

The research will be conducted in several steps. The objective of each step is as follows:

- Defining the problem
- Reviewing the state of the art in biologically inspired robotics
- Evaluating perception-based reasoning models
- Developing a test-bed robot and three robot case studies

- Developing a natural language based perception model
- Developing an estimation based perception model
- Developing a spatial knowledge based perception model

1.5 Contribution to the current state of the art

This research builds upon previous research activities conducted at the University of Cincinnati, Center for Robotics Research. It enhances the task control center of the creative learning framework with a perception-based module. It also offers a unique multifaceted approach for perception modeling, and its application in robot navigation. It addition, it presents some novel designs for several robots, one of which has a pending patent.

1.6 Outline of work

This dissertation is organized into seven chapters. Chapter 1 defines the problem, the purpose of study, and its objectives. In Chapter 2, after a background introduction, biologically inspired robotics will be reviewed. Chapter 3 discusses several perception modeling methodologies and the application of natural language in human perception modeling.

Design and development of several practical test-bed robots will be presented in Chapter 4. The Bearcat Cub robot has been developed at the University of Cincinnati robotics lab, where the author has served as the team leader. A soil sampling survey robot, which was developed by Masoud Ghaffari, Peter Cao, and Ernie Hall, will also be introduced in this chapter. The snow accumulation prevention robot, patent pending, will be discussed in this section as well.

Chapter 5 will offer several models for human perception and its application in robot navigation. A model based on natural language, an estimation-based model, and a spatial knowledge model, have been developed and will be discussed in this chapter. Chapter 6 will present a novel optimization approach for perception modeling and robot design. Chapter 7 summarizes the results and gives recommendations for future work.

Chapter 2 : Biologically Inspired Robot Navigation

"A static hero is a public liability. Progress grows out of motion."

Richard Byrd (1888 – 1957)

2.1 Background

The classical AI methodology has two important assumptions: the ability to represent hierarchical structure by abstraction, and the use of "strong" knowledge that utilizes explicit representational assertions about the world [8]. The assumption is that knowledge and knowledge representation are central to intelligence, and that robotics is not exempt from this. Perhaps these were the results of studying higher human-level intelligence and not lower life forms of creatures. Behavior-based robotics reacted against these traditions [5].

Behavior-based control shows potential for use in a robot-navigation environment since it does not need the building of an exact world model or a complex reasoning process [9]. However, much effort should be made to solve problems like the formulation of behaviors and the efficient coordination of conflicts and competition among multiple behaviors. In order to overcome these deficiencies, some fuzzy-logic-based behavior control schemes have been proposed [10].

Behavior-based control is an effective method for designing low-level primitives that can cope with real-world uncertainties, and the field of AI has developed effective tools for symbol manipulation and reasoning [11]. Integration of these two could result a better understanding and modeling of human perception.

The application of combined techniques- neural network, fuzzy logic, genetic algorithm, reinforcement learning, dynamic programming, and others- is becoming increasingly popular among various researchers.

2.2 Review

Pratihar et al. used a genetic-fuzzy approach for solving the motion planning problem of a mobile robot [12]. They used genetic algorithms to tune the scaling factor of the state variables and rule sets of a fuzzy logic controller, which a robot uses to navigate among moving obstacles.

Al-Khatib and Saade report a data-driven fuzzy approach for solving the motion planning problem of a mobile robot in the presence of moving obstacles. The approach consists of devising a general method for the derivation of input–output data to construct a fuzzy logic controller (FLC) off-line [13].

In another research Tunstel et al. presents an approach to hierarchical control design and synthesis for the case where the collection of subsystems is comprised of fuzzy logic controllers and fuzzy knowledge-based decision systems. The approach is used to implement hierarchical behavior-based controllers for autonomous navigation of one or more mobile robots [14].

Seraji presented a concept called *Fuzzy Traversability Index* for field mobile robots operating on natural terrain. This index is expressed by *linguistic* variables represented by fuzzy sets that quantify the suitability of the terrain for traversal based on its geometrical and physical properties, such as slope, roughness, and hardness [15]. Three simulation studies were presented to demonstrate the capability of the mobile robot.

Fuzzy logic based approaches uses different types of behavior using fuzzy reasoning rather than simply inhibiting some types of behavior according to an assigned priority. As a result, unstable oscillations between different types of behavior can be avoided [16].

Michaud et al. proposed an architectural methodology that is based on the idea of intentional configuration of behaviors. They use three levels of *behavior*, responsible for driving actions from sensory information, *recommendation*, which recommends different behaviors, and *motivation*, which is used to monitor the agent's goals and to coordinate the proper working with other modules [17].

Skubic et al. developed two modes of human-robot communication that utilized spatial relationships. First, using sonar sensors on a mobile robot, a model of the environment was built, and a spatial description of that environment was generated, providing linguistic communication from the robot to the user [18, 19]. Second, a hand-drawn map was sketched on a PDA, as a means of communicating a navigation task to a robot [20]. The sketch, which represented an approximate map, was analyzed using spatial reasoning, and the navigation task was extracted as a sequence of spatial navigation states.

2.3 Humanoid robots

For thousands of years, humans have looked to nature to find solutions for their problems. This trend has affected the robotics field as well as the fields of artificial intelligence, manufacturing, biomechanics, vision and many others. In the robotics field, there are many unsolved problems which have been solved in nature. These problems vary from basic motion control to high level intelligence problems. Insect motions, a human's ability to walk, drive, explore an unstructured environment, and recognize objects are examples of these problems. Although robotics researchers have looked to nature to find solutions to these problems, what is missing is human-like computational ability. The presumption is that if we want to create a human-like robot, we should implement systems which perceive and operate similar to a human's [21].

Scientists, predictors, and entertainers have similar dreams about the future of robots; however, they may choose different paths to realize them. What is fascinating is the social aspect of this technology. This transcending fascination with robotics is not necessarily about comfort, physical needs, or economical advantages. Humans are looking for companionship as well, which is something they have always traditionally looked at nature to find. This chapter will describe a survey of biologically inspired and humanoid robotics. It will cover the social aspect of robotics as well as some state of the art products. In addition, methodological challenges of humanoid robots will be discussed.

The term *robot* was coined in a 1923 play by the Karel Capek, entitled *RUR* (Rossum's Universal Robots), as a derivative of the Czech *robota* which means "forced labor" [22]. The word *robotics* was coined by the renowned science fiction writer, Isacc Asimov, in the 1942 story, "Runabout"[23]. Science fiction is one of the areas which have stimulated creative activity in robotics.

The laws of robotics are an attempt to counteract some fears by building safeguards into such machines. Isaac Asimov is generally credited with creating these laws, and writing a series of short stories (collected in *I, Robot*) about the application of the laws. Nevertheless, Asimov published two robot stories--"Robbie" and "Reason"--which introduced robots with brains, and alluded to restrictions on robot behavior to counter the Frankenstein motif started by Mary Shelly.

The three original laws were first introduced *in toto* in "Runabout" (1942). These laws are so ingrained in the conventions of science fiction that most authors routinely refer to the laws or explain why they are not in effect. The Three [Original] Laws of Robotics:

First law: A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

Second law: A robot must obey the orders given it by human beings except where such orders conflict with the first law.

Third law: A robot must protect its existence as long as such protection does not conflict with the first or second law.

Asimov added a fourth, or Zeroth, Law in *Robots and Empire* (1985):

Zeroth law: A robot may not injure humanity or, through inaction, allow humanity to come to harm.

Today, humanoid robotics labs across the globe are working on creating a new set of robots that take us one step closer to the androids of science fiction. Building a humanlike robot is a difficult engineering task that requires a combination of mechanical engineering, electrical engineering, computer architecture, real-time control, and software engineering [22]. Human-like service robots, which can work interactively with humans in the same environment by using their natural communication means, is one of the biggest challenges for future intelligent machines [24].

The most common goal of robotics researchers is to understand how to design and build machines able to perform specific tasks in the production of final products or services. In personal robotics, the researcher is directly developing the final product; thus, new factors typical of the product engineering design processes (such as task analysis, marketing, industrial design, reliability, and safety) must be included in the design phase. Whereas the performance of industrial robots can be measured by means of objective parameters, the success of a personal robot should be evaluated by applying subjective, user-based criteria [25].

Biologically inspired designs are based on theories drawn from the natural and social sciences, including anthropology, cognitive science, developmental psychology, ethology, sociology, structure of interaction, and the theory of mind. Generally speaking, these theories are used to guide the design of robot cognitive, behavioral, motivational (drives and emotions), motor and perceptual systems. Two primary arguments are made for drawing inspiration from biological systems. First, numerous researchers contend that nature is the best model for "life-like" activity. The hypothesis is that in order for a robot to be understandable by humans, it must have a naturalistic embodiment, it must interact with its environment in the same way living creatures do, and it must perceive the same things that humans find to be salient and relevant [26]. The second rationale for biological inspiration is that it allows us to directly examine, test and refine those scientific theories upon which the design is based [22]. This is particularly true with humanoid robots [27].

Adams et al. hope not only to produce robots that are inspired by biological capabilities, but also to help shape and refine our understanding of those capabilities. By bringing this theory to bear on a real system, the proposed hypotheses are tested in the real world and can be more easily judged on their content and coverage [22].

2.4 Social complexity and evolution theory

Humans and animals have faced similar physical challenges during evolution. If our needs were only physical in nature and concerned only with survival, we would not need the brain and intelligence that we already have. It seems social complexity was the driving force behind the development of the comparatively large human brain in relation to that of other mammals. Brain size alone is not the key difference though, since elephants have much larger brains than humans. However, the use of perception and natural language must be a major factor in human intelligence. This shows the magnitude of the problem that designers of humanoid robots are facing. While our current technology is not comparable, even with a low level creature's abilities, humanoid robotics is facing the social complexity of a human, in addition to the challenges inherent in replicating its physical abilities. Some companies already claim that their products are true companions and that they should be treated similar to a pet.

2.4.1 Social robots

Many species of mammals (including humans, birds, and other animals) often form individualized societies. Although individuals may live in groups, they form relationships and social networks, they create alliances, and they often adhere to societal norms and conventions [28].

Dautenhahn and Billard proposed the following definition: “Social robots are embodied agents that are part of a heterogeneous group: a society of robots or humans. They are able to recognize each other and engage in social interactions, they possess histories (perceive and interpret the world in terms of their own experience), and they explicitly communicate with and learn from each other.”[29].

In particular, social learning and imitation, gesture and natural language communication, emotion, and recognition of interaction partners are all important factors. Moreover, most research in this area has focused on the application of “benign” social behavior. Thus, social robots are usually designed as assistants, companions, or pets, in addition to the more traditional role of servants [27].

“Socially interactive robots” is a term that has been used to describe robots for which social interaction plays a key role. It is important to distinguish these robots from other robots that involve “conventional” human–robot interaction, such as those used in teleoperation scenarios. The focus is on peer-to-peer human–robot interaction. Specifically, robots that exhibit the following “human social” characteristics:

- Express and/or perceive emotions;
- Communicate with high-level dialogue;
- Learn/recognize models of other agents;
- Establish/maintain social relationships;
- Use natural cues (gaze, gestures, etc.);
- Exhibit distinctive personality and character;
- May learn/develop social competencies [27].

Socially interactive robots can be used for a variety of purposes: as research platforms, as toys, as educational tools, or as therapeutic aids. The common, underlying assumption is that *humans prefer to interact with machines in the same way that they interact with other people* [27].

2.4.2 Some biologically inspired robots

Cog, Figure (2-1), began as an upper torso, capable of 14 degrees-of-freedom, with one arm and a rudimentary visual system. In this first incarnation, multimodal behavior systems, such as reaching for a visual target, were implemented. Currently, Cog features two six degree-of-freedom arms, a seven degree-of-freedom head, three torso joints, and a much richer array of sensors. Each eye has one camera with a narrow field-of-view for high resolution vision and one with a wide field-of-view for peripheral vision, giving the robot a binocular, variable-resolution view of its environment.

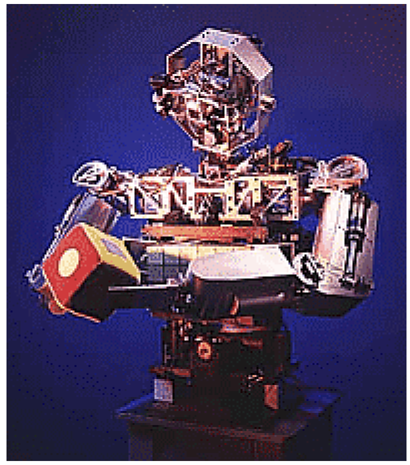


Figure 2-1: MIT humanoid robot, Cog

Following the success of Sony Corporation's 'AIBO,' robot cats and dogs are multiplying rapidly. AIBO means 'companion' in Japanese, and is also an acronym for Artificial Intelligence roBOt.

"Robot pets" employing sophisticated artificial intelligence and animatronic technologies are now being marketed as toys and companions by a number of large consumer electronics corporations [30].

A legged robot application developed by MIT Leg Laboratory is shown in Figure 2-2. Troody is an 11-pound walking birdlike robotic dinosaur which is being marketed to natural history museums for educational and entertainment purposes.

Figure 2-2: Troody the dinosaur robot of MIT

Figure 2-3: AIBO from Sony

The "Sprawl" family of hand-sized hexapedal robots are prototypes designed to test ideas about locomotion dynamics, leg design and leg arrangement and to identify areas that can be improved by Shape Deposition Manufacturing. Sprawlita is a dynamically-stable running hexapod based on functional principles from biomechanical studies of the cockroach. The prototype was fabricated using Shape Deposition Manufacturing and is capable of speeds of approximately 3 body-lengths per second [31].

Honda engineers created ASIMO with 26 Degrees of Freedom that help it walk and perform tasks much like a human. One degree of freedom is the ability to move right and left or up and down. At birth, the human body has about 350 bones, but by the time a human reaches adulthood, some of our bones have fused together to give us a total of 206 bones in our body.

If each of these bones is considered as a link that moves with a single degree of freedom, then over 200 degrees of freedom are needed to emulate human motion. These degrees of freedom act much like human joints for optimum movement and flexibility. According to Honda, ASIMO stands for "Advanced Step in Innovative Mobility." It probably also derived from the late Isaac Asimov’s name, who authored "I, Robot" in 1950 and announced the three laws of robotics.

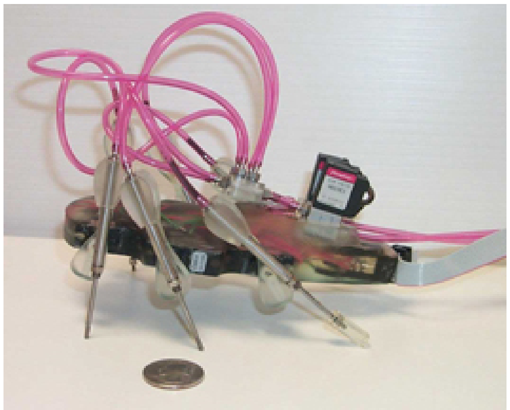


Figure 2-4: Sprawlita from Stanford's biomimetic robotics lab

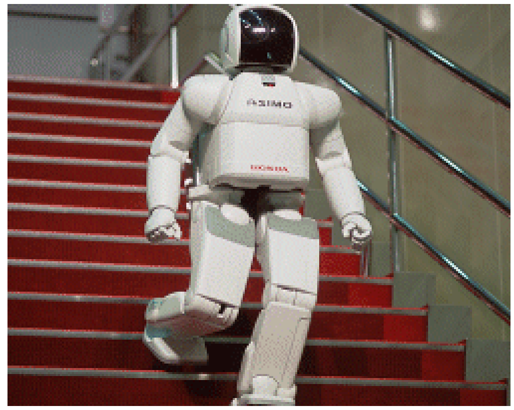


Figure 2-5: ASIMO from Honda

The Hasbro / Wow-wee B.I.O. Bugs are a series of battery-powered autonomous / remote-controlled robot bugs. They are substantial beasts, measuring 25cm x 29cm (9.8" x 11.4") excluding sensors and weighing 0.492kg (1.08lbs) in the case of the Predator bug. The name B.I.O.-Bugs stands for biomechanical integrated organisms. The basic capabilities of these 2 motor walkers is considerable: they can traverse surfaces as deep as shag carpeting (on full batteries) with little problem. They can perform a fairly sharp turn, within 1.5x their body length, which is a fairly impressive feat using only 2 motors. They broadcast and receive IR data via the front / rear ports, and can recognize friend, foe, and IR controller input. An additional transmitter is located on the rear of the robot, leaving a blind spot only when the remote is aimed at the mid-riff from an angle of about 110 degrees from the angle of travel.

According to the NASA website, Robonaut is a humanoid robot designed by the Robot Systems Technology Branch at NASA's Johnson Space Center in a collaborative effort with DARPA. The Robonaut project seeks to develop and demonstrate a robotic system that can function as an EVA (extra-vehicular activity) astronaut equivalent. Robonaut jumps generations ahead by eliminating the robotic scars (e.g., special robotic grapples and targets) and specialized robotic tools of traditional on-orbit robotics. However, it still keeps the human operator in the control loop through its tele-presence control system. Robonaut is designed to be used for "EVA" tasks, i.e., those which were not specifically designed for robots.


Figure 2-6: BIO bug from Wow-wee toy maker

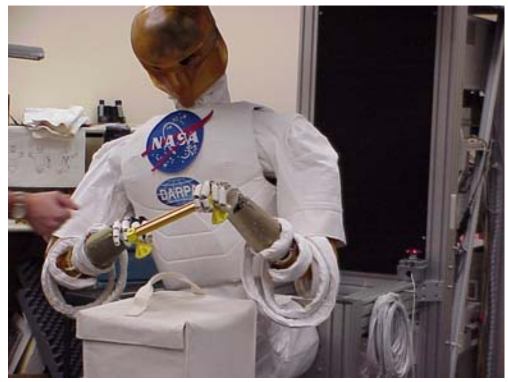


Figure 2-7: Robonaut, a humanoid robot from NASA

Chapter 3 : Perception-based Reasoning

"It is the nature of all greatness not to be exact."

Edmund Burke (1729 - 1797)

"In this world nothing can be said to be certain, except death and taxes."

Benjamin Franklin (1706 - 1790)

3.1 Soft computing

Traditional (hard) computing methods do not provide sufficient capabilities to develop and implement intelligent systems for many real world problems. Soft Computing is tolerant of imprecision, uncertainty, and partial truth, and it has provided important practical tools for constructing intelligent systems and dealing with human perception [32].

Zadeh states that; "by design, soft computing is pluralistic in nature in the sense that it is a coalition of methodologies which are drawn together be a quest for accommodation with pervasive imprecision of the real world. At this juncture, the principal members of the coalition are fuzzy logic, neuro-computing, evolutionary computing, probabilistic computing, chaotic computing and machine learning. What is important is that members of the coalition are, for the most part, complementary rather than competitive. Furthermore, they are synergic in the sense that, in general, better results can be obtained when they are used in combination rather than in standalone mode" [33]. Fuzzy logic is one of the fundamental soft computing methods.

3.2 Basics of fuzzy logic

Fuzzy logic intends to represent, and reason with, knowledge that is in linguistic or verbal form. The term "fuzzy logic" has been used since the 1960's when L. A. Zadeh published his famous paper on fuzzy set theory [34]. The concept was later expanded to the *theory of approximate reasoning* and the *theory of linguistic logic*. The later is a logic whose true values are expressions of natural language (for example, very young, young, middle age, old). The former is the most often used [35]. A detailed and classic description of the basic fuzzy logic ideas can be found in Zadeh's papers [36-39].

Fuzzy logic deals with perceptions, and *vagueness* is an important attribute of perception. For example [40] within the range of [0*m*, 40*m*] different operators want the robot to travel a large distance. The challenge is determining how different people perceive the concept of *large* and how they agree or disagree about the largeness of a given distance. Three possible answers are as follows:

The first approach is to decide a cutoff distance d. For $x<d$ consider *disagree* and for $x>d$ consider *agree* with the largeness of distance. The graph of agree-disagree is distributed uniformly over the left side of $x = d$ and also over its right side.

In the second approach the *x* interval may be divided into three parts. The first part represents the area that person is in *disagreement* about largeness and the last part is the area that person is in *agreement* on the largeness of *x*. In the middle area, the person is uncertain about largeness of *x* and makes no judgment.

The third person may use what we call the basic idea of fuzzy logic. This person uses a scale for the degree that he/she is in agreement with the largeness of distance. The larger the distance the more in agreement this person is. This depends entirely on the case and sample, as well as on the person's perception. In this case a continuous scale will be used to represent the degree of distance largeness for all *x* values.

These approaches show three different ways of representing one's knowledge about the largeness of distance. The fact that there are three different approaches implies that the concept of *large* is indeed a *vague* concept, meaning that the set of objects it implies have no sharp boundaries [41].

The main motivation for the development of fuzzy logic was to deal with the concept of *vagueness*. It offers a conceptual framework to deal with this concept and to manipulate such variables. Uncertainty and vagueness are two concepts that sometimes are mistaken for each other. The possibility theory addresses uncertainty and not vagueness. Uncertainty comes from the lack of knowledge for the *occurrence* of an *event*. When the event is complete and the results are known, there is no uncertainty. Uncertainty exists when an experiment is to proceed and the results are unknown [35].

Randomness is a specific form of uncertainty. It started as a part of the probability theory by Jakob Bernoulli (1654-1705). The main subject of study was gambling and games. It is used when the subject of study has numerous parameters which make it almost impossible to model with deterministic methods. Quantum mechanics shows that randomness has deep roots in nature. In a simple form "probability can be thought of as a numerical measure of the likelihood that a particular event will occur" [35]. It generally uses a [0,1] scale where values close to 1 represent a high likelihood of occurrence and values close to 0 show the opposite. An example of vagueness was discussed earlier.

3.3 Human perception

Perception is the name given for the process of the organization, interpretation and explanation of the data that reaches the brain from the sense organs. The data reaching the sense organs has no meaning or importance without perception. The information from the senses has to be perceived, in other words explained. We can only decide what kind of a reaction we are going to perform when are able to perceive, or decipher, the raw data that is gathered by our senses [42].

Perception is a vital part of human reasoning. Humans do a variety of physical and mental tasks without any measurements or computations. Some examples of these activities are driving in traffic, parking a car, cooking a meal, playing an instrument and summarizing a story. In fact, our ability to perform these tasks is based on the brain's ability to manipulate perceptions, perceptions of time, distance, force, direction, speed, shape, color, likelihood, intent, truth and other attributes of physical and mental properties [43].

The relationship between perception and action are often discussed in the context of ecological psychology [44]. The new trend in fuzzy logic focuses on the perception processing and preliminary results are emerging [35, 45, 46].

Uncertainty representation is an important point to consider. Initial works mainly focused on probabilistic methods in which Bayesian probabilities were used to represent partial beliefs [47, 48]. These approaches can not distinguish between lack of information and uncertain information, and a probability assignment to a proposition automatically implies an assignment also to its negation [49]. Dempster-Shafer theory of evidential reasoning [50] and fuzzy logic have been used as an alternative [51, 47].

There is an ample amount of available literature on perception, including thousands of papers and books in the areas of psychology, linguistics, philosophy, brain science, and many others [52]. And yet, what is not in existence is a theory in which perceptions are treated as objects of computation. Such a theory is needed to make it possible to conceive, design, and construct systems which have a much higher potential for intelligence than those we have today [43].

Perceptions are fuzzy in the sense that perceived values of variables are not sharply defined, and perceptions are also granular in the sense that perceived values of variables are grouped into granules, with each granule being a clump of points drawn together by difference, similarity, proximity or functionality.

Therefore, the fuzzy logic theory can be used as a departing point for computing propositions and perception-based robot control. This will be done through the use of what is called constraint-centered semantics of natural languages (CSNL) [53, 43, 54].

The principal ideas and assumptions which underlie CSNL can be summarized as follows:

- Perceptions are described by propositions drawn from a natural language.
- A proposition, *p*, may be viewed as an answer to a question.

- A proposition is a carrier of information.
- The meaning of a proposition, p, is represented as a generalized constraint which defines the information conveyed by *p*.
- Meaning-representation is viewed as translation from a language into the generalized constraint language

More needs to be done to develop a computational theory of perceptions applicable in robot control.

3.4 Natural language

The first successful operation of a stored-program electronic computer took place, at the University of Manchester, in June 1948. Within weeks mathematician Alan Turing was drawing up a list of potential uses for this new device: the second and third items were ‘learning of languages’ and ‘translation of languages’. Since the 1960’s computers have been available in universities, text analysis being the subject of interest for many researchers. In the 1970’s, there were great expectations. For example, it was expected to have an automatic translations system very soon but in reality Natural Language Processing (NLP) showed to be more complex.

After what can be called the Chomsky’s decade in the eighties there is a move from a general and domain-independent to a more domain-oriented approach to NLP [55]. As a consequence, instead of pursuing a universal solution for NLP, there was a shift to domain-dependent solutions where the specificity of the semantics of the domain played a major rule.

Language is an important way to convey human perception. Some other imaginable ways to express perception could be art, music, vision, touching, smelling, wealth, power etc.

Natural language can express rules and sequences of commands in very concise way. Natural language uses symbols and syntactic rules and is well suited to interact with robot knowledge represented at the symbolic level. It has been shown that learning in robots is much more effective if it operates at the symbolic level [11, 56].

ROBOT (Harris, 1977) and SHRDLU (Winograd, 1972) are among the first attempts for natural language communication with robots [57-59]. However, both of them are simulated robots and not real ones. SHRDLU allows the user to command a robot that moves in a world made of blocks. Selfridge and Vannoy (1986) present a natural language interface to an assembly robot which allows the user to talk with the system in order to recognize the shapes of objects and put them together to make a more complex component. The knowledge base consists of a set of if-then rules [60].

Some natural language interfaces include both written and spoken forms of natural language. The users of SAM (Brwon et al., 1992) can use a telephone or a keyboard to command the robot. The robot is manipulator arm with six axes and a grip. The user can introduce a description of the objects. This description will later be used to command the robot to perform actions over them [57, 61].

Most of these robots can run a very specific and limited plan. This is a common feature in the majority of systems designed for natural language communication with robots [57].

The difficulty of talking to robots comes from both side of the communication channel. The first problem is that it is human dependent. Different people use natural language in variety of ways and sometimes it is vague. The second problem is the limitations of robots in interact with their environments. This depends on quality of the perception system. Limitation of this system should also be taken into consideration at the natural language interface design stage. The discrepancy between the human perception system and the perception of robot is one of the foreseeable sources of communication problems [62].

3.4.1 Applications of natural language control

The problem of speaking to a machine had indeed been the focus of speculations for a long time before modern technology made it available. However, the style of communication might be different than daily conversation among human. The hypothesis that communication with computers may eventually create a new, precise language does not seem to be surprising [62].

In practical products application of natural language perception-based control might be implemented differently. One application might be object identification. Range sensors such as laser, ultrasonic, and stereovision, the most commonly used sensors in robot navigation, only provide information about the existence of objects in some given positions of space. Many applications require identifying the type of objects that have been detected to make the appropriate decision by robot, considering the fact the object identification problem has not been solved for many real world situations [63].

For example an unmanned combat vehicle may face a variety of environments. If the object recognition algorithm tries to match the description of the detected object with all possible objects, the search would be huge and probably unpractical. But, if a user describes the more probable objects that might be found in a certain mission, object identification would be much easier.

Natural language also can be used for industrial robot programming especially when users are inexperienced and don't know about computer programming. It can replace the current teach pending modules which still use a keyboard interface. Another potential application of natural language perception-based control is in Ergonomics.

Ergonomics studies the human-machine interaction and tries to find a more user friendly interface for man machine communication. Human made systems are getting more complex and sophisticated everyday. At the same time human operators have to interact to and control these systems. They get information from the system status and they perform proper actions. It is possible to reduce complexity of interaction by designing automated systems and by designing an adequate interface between the operator and the system. Design of a friendly interface which decrease complexity of interaction and allow the operators to act in a simple and quick way in every kind of situation is important [57].

Natural language interfaces can be used in a variety of applications and it could help operators to manipulate the system by entering commands in their own language. It can be of great interest as a tool to simplify the work of control center and other machine operators [57]. This is another potential application area for the proposed research.

Probably it is not unrealistic to predict a time that design software packages like AutoCAD are able to design modules by recognizing the verbal description of system from the designer. Combination of such a system with the traditional interfaces could change the human design's ability tremendously. Software industry already is moving in that direction. Now, there are voice recognition word processors and in many packages menu commands can be run by voice.

3.4.2 Methodology

To extract the meaning of a text or speech there are five fundamental steps: orthographic, semantic, statistical, syntactic, and usage analysis [64].

Orthographic Analysis: In this step the units of language, words, will be recognized. In English characters like spaces and tabs separate the words and orthographic analysis is easier than those languages without a clear word boundary.

Semantic Analysis: Different words can represent similar meanings. The goal of this step is to relate all those words to a same meaning. Some techniques like automatic suffix removal or using a domain-specific thesaurus can be applied for semantic analysis. A significant work is required in extracting from the semantic representation of the user's speech to the corresponding robot-executable procedures [11].

Statistical Analysis: The analysis of frequency of word usage is one of the most common methods in NLP [65]. Counting the number of occurrence of a word or words with co-occurrence is an example of statistical analysis.

Syntactic Analysis: Part-of-Speech analysis, for example, can help to disambiguate word sense, and syntactically identified phrases can provide a basis for further statistical analysis.

Usage Analysis: The way in which a document has been used can provide valuable information. For example if an article is about Neural Network and it is published in a manufacturing journal it would be reasonable to infer that the article talks about application of NN in manufacturing.

Natural language processing is a complex task. Interactions of the mentioned analysis present a challenging problem, much higher than the scope and objectives of this project. Several resources explain the complexity of this problem [66-69].

Chapter 4 : Robot Design

"I've been trying for some time to develop a lifestyle that doesn't require my presence."

Gary Trudeau (1948 -)

The Bearcat Cub is an intelligent, autonomous ground vehicle that provides a test-bed system for conducting research on mobile vehicles, sensor systems and intelligent control. The Bearcat Cub was designed for the Intelligent Ground Vehicle Competition (IGVC) and has evolved through the past few years with contributions from all members of the University of Cincinnati Robotics Team [70, 71], where the author has been the team leader. The following is an overview of the design.

4.1 Robot frame

The frame is made of 80/20 aluminum extrusions in order to have a light-weight structure without compromising strength. The junctions are made using small joining strips at the ends or by utilizing corner brackets which sit inside the joints. The advantage of using this frame concept is that it can be easily reshaped if new components are to be added. Stress and weight calculations for the joints were carried out using a safety factor of 125%. A drawing of the basic structure is shown in Figure 4-1.

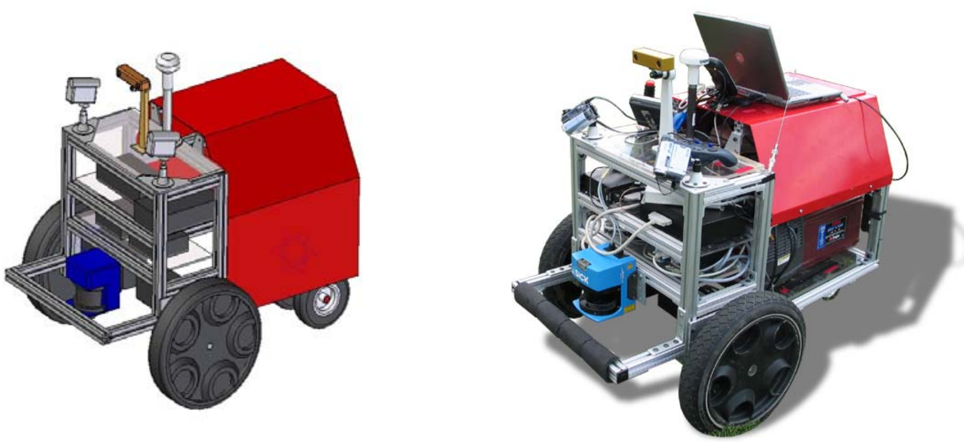


Figure 4-1: Basic robot structure

4.2 Wheels, brakes, motor and gears

The Cub's mechanical system utilizes two types of wheels: two main drive wheels and a dual rear castor wheel. The main drive wheels are 19"-diameter enhanced traction wheels designed by Michelin for the Segway Human Transporter. The rear castor wheel helps improve the stability of the robot during turns such as those with a zero turning radius. This 8", 90 series, dual castor wheel is from Borne & Co. Since the drive wheel size is 19" and the maximum speed of the robot is 5 miles/hour, a frictional coefficient of

0.125 and a gearbox efficiency of 70% have been used to calculate the required gear ratio. A gearbox with a gear ratio of 25:1 was selected and obtained from Segway. The required motor power has been found to be 1.355 hp per motor. Two Pacific Scientific PMA43R-00112-00 2-hp brushless servo motors have been selected for providing power. The gearbox and motors have been selected based on the calculated values. The robot's power system can utilize a maximum of 2 Honda EU-2000i super-quiet generator sets. However, a single generator set has proven to provide 4 ½ hours of continuous power. The advantage of having a generator set in place of batteries is that there is less down-time after losing power, since refueling the generator set is much quicker than recharging a battery.

Mechanical brakes have been designed statically and kinematically in NX 3.0. The brakes are fail-safe in nature: when power is cut from electromagnetic magnets, multiple springs provide nearly 80 pounds of force to the drive wheels. Dynamics calculations and tests have shown that this ensures a distance of 3 feet from top speed to full stop on level ground, in a dry environment.

4.3 Electrical and electronic systems

The electrical systems of the Bearcat Cub consists of a motion controller, 2 amplifiers, 2 DC brushless motors, 2 digital cameras, ISCAN vision processing hardware, Bumblebee stereo vision camera, a laser scanner, GPS unit, digital compass and an emergency stop. All power is provided by a Honda GenSet and/or marine battery. This allows the Bearcat Cub to be outfitted with any set of sensors very easily since there is no need for the user to customize any electronics. The system acts like a hardware equivalent of software plug-and-play. Figure 4-2 below shows the general electronics layout.

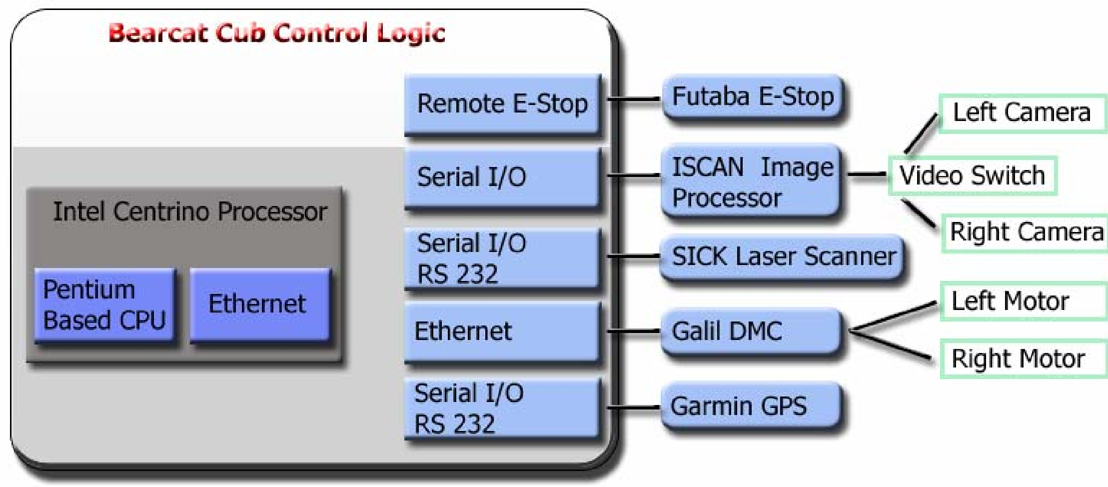


Figure 4-2: Bearcat Cub block diagram

4.4 Emergency E-stop

Safety is of primary importance on the Bearcat Cub. System operation can be halted in 3 separate ways. A Futaba remote control can be used to cut power from all systems via an FM signal capable of transmitting from 65 feet away. Second a manual, large red laboratory standard, emergency power kill switch is located on the back of the Bearcat Cub in case the remote should fail. Also of note the emergency stop is kept from tripping via an active high signal which ensures that, if ever a case arose when power was not delivered to the emergency stop, the system would automatically stop. Finally an abort command can be sent via the 'A' key on our wireless joystick controller. This kills the current process in software allowing a user to check all systems and determine what may have caused a problem without losing system data. Also, the E-Stop is designed with redundant systems incase vibration etc compromises electrical connections.

4.5 DMC motion controller

The Galil DMC 2130 motion control board is the motion controller used for the Bearcat Cub and it is controlled through commands sent via an Ethernet connection from a laptop. Copley amplifiers deliver power to the motors after amplifying the signals they receive from the motion controller. Steering is achieved by applying differential speeds at the right and left wheels. The vision system used for obstacle avoidance sends data to the computer which is processed by the software and then used to generate commands to the motion controller to change the differential speeds of the two motors. The Galil motion controller was chosen because it is web based, has PID and Bode plot tuning software, and is compact and enclosed in a durable package. The controller can accommodate up to 4 axis formats and can control stepper or servo motors on any combination of axes. The Bearcat Cub has the ability to turn about its drive axis effectively performing a Zero Turning Radius (ZTR) pirouette. The block diagram of the system is shown in Figure 4-3.

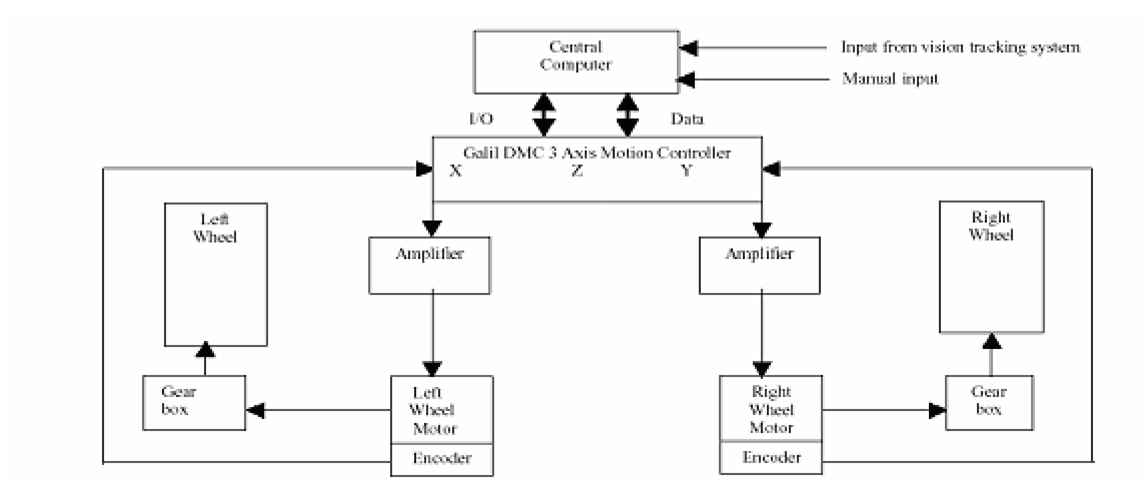


Figure 4-3: Motion control system

4.6 Sensor systems

4.6.1 Laser Measurement System

The Sick LMS 200 scans a 2-dimensional plane of 180 degrees at ½ degree increments and returns obstacle distance measurements for up to 8.191 meters based on laser time of flight. The laser scanner has the capability to scan at a variety of angular ranges and resolutions. The range and resolution of the laser scanner can be changed easily since the system is designed to deal with variable sensory data input.

4.6.2 Vision system

Two video cameras, the right and left cameras, provide the images that are used by the line detection system. The cameras used by the Cub are Sony handy cams. Each camera has its own LCD monitor. The images from the two cameras are fed into a digital video switch using standard Audio-Video cables, and the video switch outputs only one of them at a time. The output is toggled between the right and left cameras based on an input bit from the Galil motion controller which is set and cleared via a command from software. The switch allows the ability to use the output from the other camera if the first camera loses sight of the line. An ISCAN RK447-BMP external image tracker is used which computes the center of the brightest area within a region of the image and returns the image coordinate of the center point at a 30 frame per second rate. Two such points in the image are found, and the corresponding real-world points are computed in software via a linear transformation utilizing camera calibration information. The line detection system then receives the real-world points and uses them to follow the line.

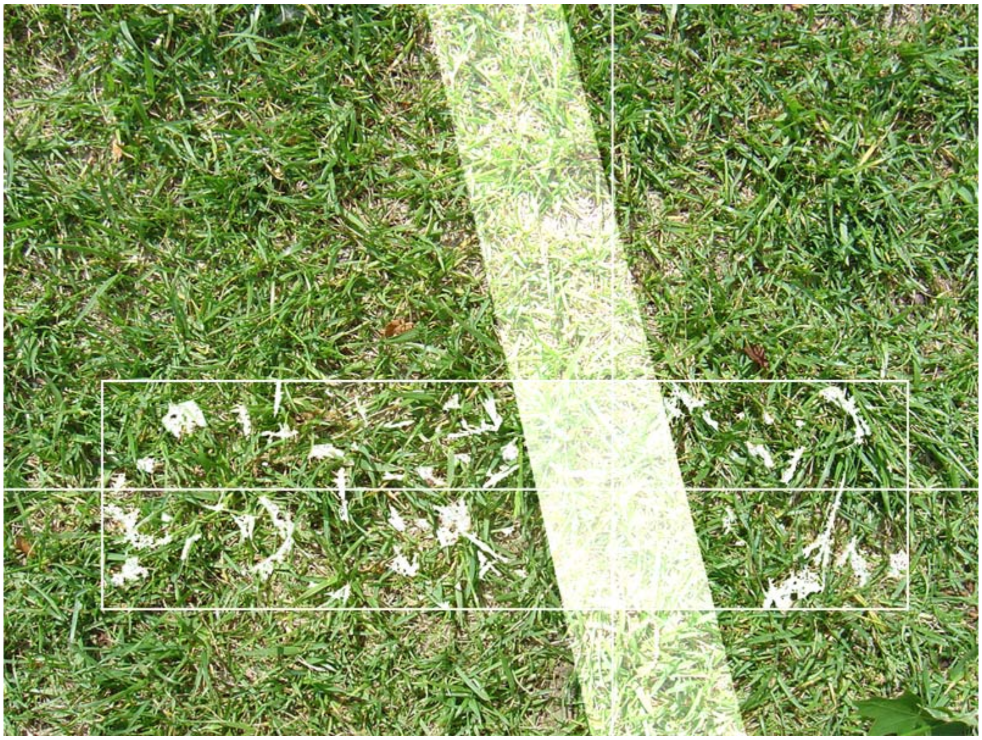


Figure 4-4: The vision processor locating the centroid of a bright region

4.6.3 Global positioning system (GPS)

A commercially available GPS system has been used for the Bearcat Cub. The main criteria for selection are Wide Area Augmentation System (WAAS) capability and embedded navigation features. The Garmin 76 has these requirements and has been selected for implementation. The GPS unit tracks the NAVSTAR GPS constellation of satellites. The signals are received by an antenna and are tracked with 12 parallel channels of L1. C/A code is then down converted to an IF frequency and digitally processed to obtain a full navigation solution of position, velocity, time and heading. The solution is then sent over the serial link via the 9-pin RS 232 connector. The unit communicates with the laptop in NMEA format. Garmin's computations are utilized as much as possible to alleviate Pentium computations.

4.7 Servo motors

The Bearcat Cub uses DC brushless servo motors PMA43R-00112-00 provided by Pacific Scientific. Brushless motors are small and powerful and efficient for servo controls. The servo feedback is provided by encoders mounted on the motor shaft and is used to compute an error signal to the controller. The compensated signal is sent to the motor to turn the robot. The difference between the actual position and position reached is the error signal. This signal is modified by a PID digital filter compensator that is designed for stability and accuracy. Thus, the servo motors are designed to achieving minimal error and maximum accuracy. Having the ability to set PID parameters directly on the controller also allows the Bearcat Cub the flexibility of different controller responses for different environments. For stepper motors, no encoder is present as it sends signals only in steps.

4.8 Computer system

A Dell Latitude D800 laptop is the central processing unit of the Bearcat Cub. It processes data from the laser scanner, GPS, motion control system, digital compass and image processing system. The software has been executed on a Dell laptop running Windows XP. Software has been written in both C++ and C# taking advantage of the .NET Framework where applicable. A user friendly GUI was developed to track the Bearcat Cub's movements and positions. A series of initialization files hold all calibration values and initial values for the system parameters.

4.9 Obstacle detection and avoidance

Obstacles are detected by scanning the data returned from the laser scanner and checking for values less than a user defined maximum obstacle distance. Sensor fusion plays a role, laser data can be verified by data gathered using PointGray's Bumblebee hardware and our in-house vision algorithms. The distance at which the user wishes to detect obstacles is defined in a user GUI. Each set of data returned by the laser scanner is scanned for values less than the maximum obstacle distance, if a smaller value is found then the software continues to check values until a value is found that is greater than the maximum obstacle distance or the end of the data is reached. Any section of data that is found to be less than the maximum obstacle distance is used to create a data structure representing an obstacle that is described by a two angle and distance pairs for the left and right edges.

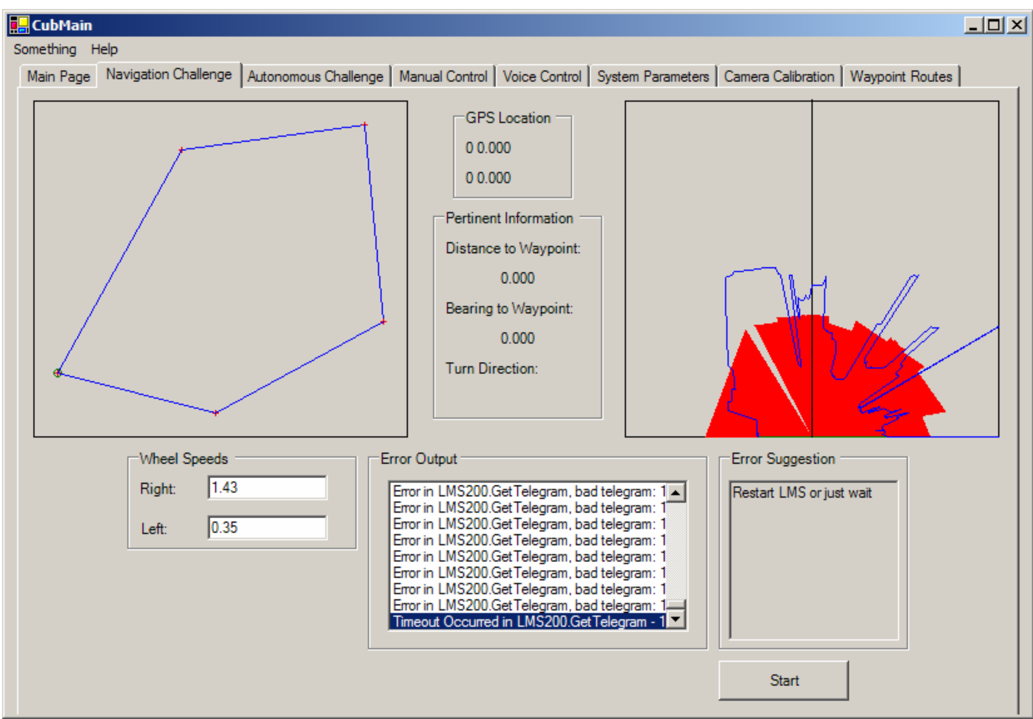


Figure 4-5: Graphical output for the Navigation Challenge [70]

The obstacle avoidance system then widens the edges for each obstacle by half the robot's width plus a specified safe distance, thereby finding the minimum angle the robot must steer to safely avoid hitting the obstacle. The system then throws out the safe angles that overlap, grouping overlapping obstacle regions together. The safe angles that bound the obstacle regions are then compared, and the angle which causes the robot to deviate the least from the desired bearing (i.e. the bearing to waypoint in the Navigation Challenge) is selected. Figure 4-6 shows this process in action, with the blue dot representing the next waypoint.

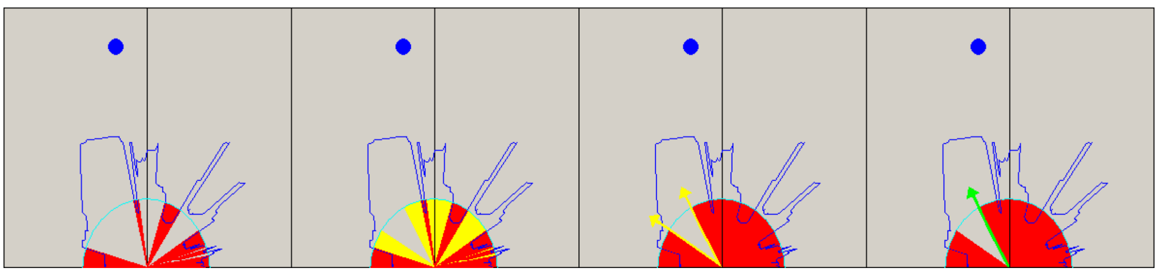


Figure 4-6: Selecting the proper angle to steer through a cluttered environment [70]

4.10 Line following

For the line following competition, the Bearcat Cub has been designed to negotiate an outdoor obstacle course in the minimum time while staying within a 5 mph speed limit and avoiding obstacles. The line following system receives as input a line from the line detection system and a series of obstacles detected by the obstacle detections system. The line is first abstracted as a wall obstacle, and then added to the list of other obstacles. The desired angle to steer the robot through the course is then calculated using the algorithm described in the previous section. Figure 4-7 shows the graphical output from the line following system. On the left, the line (in red) with respect to the robot (in blue) is shown as detected by the line detection system. The right graphic depicts the region to avoid for the line obstacle in yellow, with the regions for the rest of the obstacles in red.

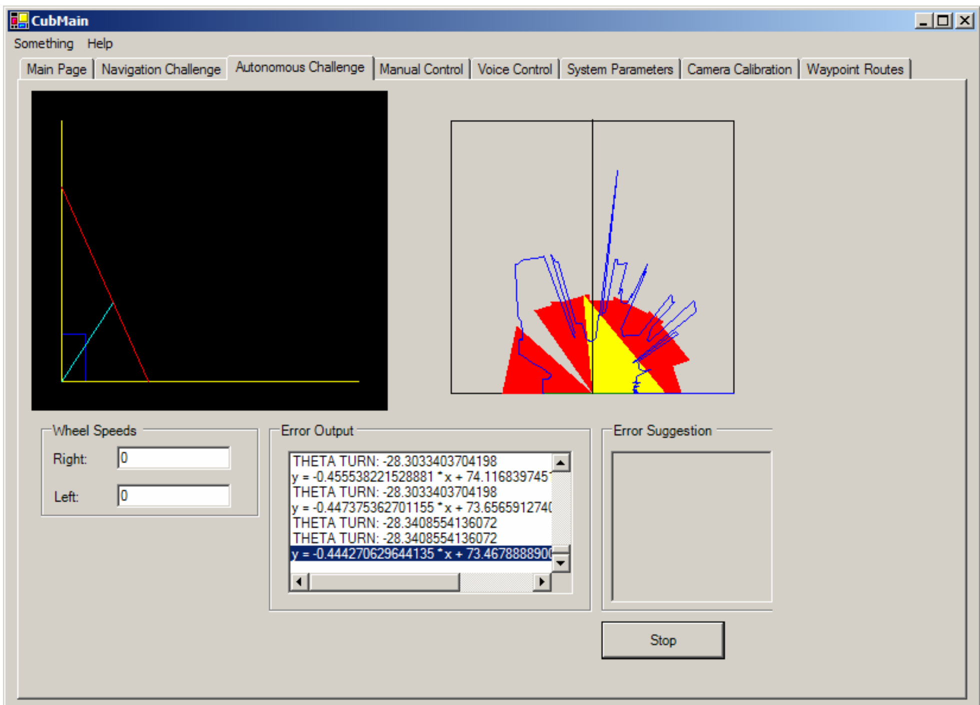


Figure 4-7: Graphical User Interface for Autonomous Challenge [70]

4.11 Waypoint navigation

Global Positioning System (GPS) technology provides the basis for waypoint navigation in the Bearcat Cub. The classical closed feedback control loop was utilized in the modeling of the navigational challenge problem with an input command, feedback signal, error signal, and output transfer function characteristics. The target waypoint destinations are specified as the input command and the feedback signal is provided by the GPS unit based on its position with reference to satellite data. Using the current position co-ordinates and velocity the GPS unit provides bearing, tracking, signal validity and range from the target waypoint to determine the error. The bearing to the waypoint is passed to the obstacle avoidance system, which then determines the best path to the target while avoiding any nearby obstacles and returns the safe bearing. The Waypoint Navigation sends the new bearing to the motion control system, which translates the commands into motor control voltages that steer and propel (right, left or stop) the robot on the course. Once the target range has been reduced to the required tolerance, the robot has reached its target destination waypoint. The process continues for all the waypoints in the input file finally returning the robot to the starting point. A dead reckoning algorithm has been written to compute range and bearing data between GPS updates. While the robot is navigating its route between waypoints, the system graphical user interface (Figure 4-5) displays current information regarding the robots current position, the map of waypoints, the field of view from the obstacle detection system overlaid with red for all areas where the robot cannot go, and any appropriate error or feedback information.

4.12 System integration

The run-time system sensory input consists of two digital cameras, an image processor, a laser scanner system, a GPS unit, and in certain cases a joystick for manual control. The software allows initialization information that is input before the autonomous system begins operation in order to properly calibrate various parts of the system. The output is commands to the motion controller will set the speeds of the two independent drive wheels. The system will use the input information and apply various algorithms to determine the proper course of action, based on the current competition, and output the motion commands to the motion controller.

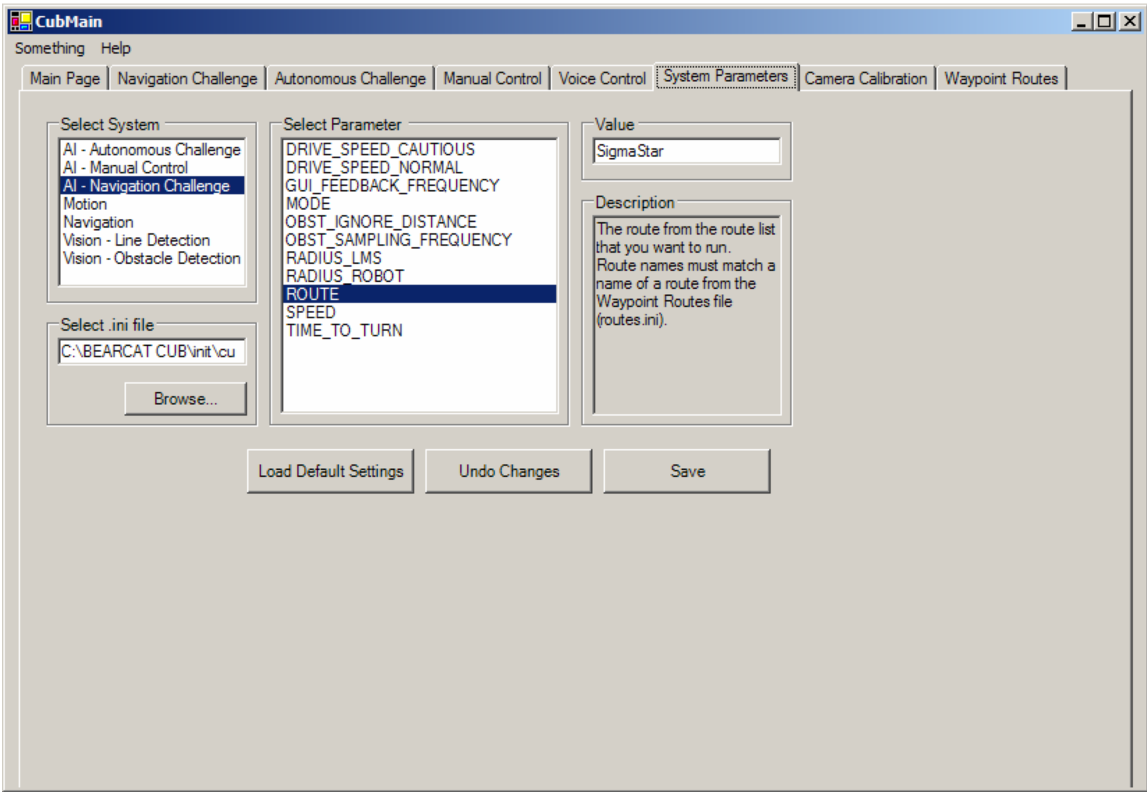


Figure 4-8: Interface for conveniently initializing system parameters [70]

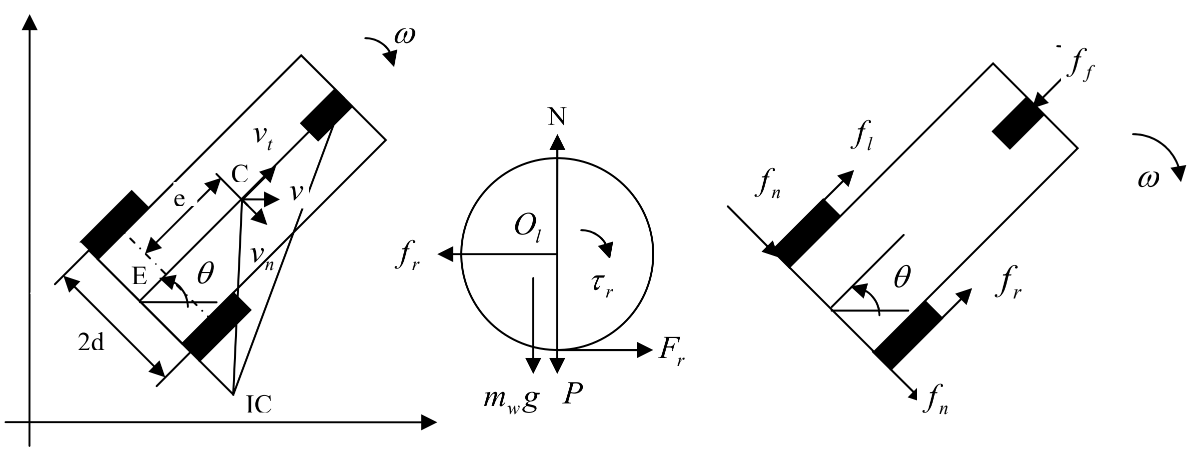
4.13 Safety and reliability

There are four different safety systems built in to stop the robot: manual e-stop, remote control e-stop system capable of stopping the robot from a range of 65 feet, joystick “full-stop” button, with a range of 30 feet and mechanical brakes. Joystick “full-stop” can be used to pause and continue software functionality, while use of the primary e-stop necessitates a manual reboot of the system. The drawbacks to the secondary e-stop are its range and its reliance on the software to be functioning properly in order to work. The primary remote e-stop will work regardless of the state of the software or any other device on the robot and controls engagement of the mechanical brake. A disconnect switch can also cut off all power to the robot. The generators have hazards from both internal combustion and electric powered systems but they come with built in overheat, over power and power surge protection.

Reliability of an autonomous robot can sometimes be difficult to predict. However, we have tried to be as thorough as possible in our testing strategies, and the Cub has performed very well. The emergency stop systems are available to quickly and reliably stop the robot if it starts to misbehave. Front bumpers are present which reduce the impact of physical shocks from reaching the stereo vision and cameras mounted on the robot. All circuits have been color coded to ensure proper reconnection with the black wires used for ground.

4.14 Dynamic model

For the Bearcat Cub robot, a kinematic and dynamic model was derived using the Newton-Euler method by Alhaj Ali et al. [72-76]. "Bearcat Cub structure and dynamic analysis are shown in Figure 4-9":



a. Robot Structure | b. Dynamic analysis for the right wheel | c. Dynamic analysis for the robot

Figure 4-9: Robot dynamic analysis [76]

According to Figure 4-9, the kinematic model with respect to the robot center of gravity (Point C in Fig. 4-9 a.) can be described as follows [76]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ v_n \\ \omega \end{bmatrix} \qquad \text{Eq. (4-1)}$$

Where v_t, v_n, ϖ can be defined in terms of the angular velocity of the robot left wheel ω_l and the angular velocity of the robot right wheel ω_r as follows [76]:

$$\begin{bmatrix} v_t \\ v_n \\ w \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{er}{2d} & \frac{-er}{2d} \\ \frac{r}{2d} & \frac{-r}{2d} \end{bmatrix} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} \qquad \text{Eq. (4-2)}$$

However, Eq. 4-1 can be simplified by utilizing that $v_n = e\omega$ as follows [76]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & e\sin\theta \\ \sin\theta & -e\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega \end{bmatrix} \qquad \text{Eq. (4-3)}$$

The nonholonomic constraint can be obtained directly from Eq. 4-3 as [76]:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = \omega e \qquad \text{Eq. (4-4)}$$

For the center of the wheel axes (Point E in Fig. 4-9 a.) $e = 0$ and hence Eq. 4-4 reduces to [76]:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \qquad \text{Eq. (4-5)}$$

This means that there is no motion in the direction of the wheel axis.

Another constraint for the kinematic model comes from the inertial structure of the robot where the robot's path cannot exceed the minimum turning radius or the maximum curvature [76]:

$$\rho \geq R_{\min imum}\ , \text{or}\quad k \leq K_{\max imum}$$ Eq. (4-6)

From Figure 4-9 b., the Newton-Euler equation for the right wheel can be described as [76]:

$$F_r - f_r = m_w \ddot{x}_r$$ Eq. (4-7)

$$\tau_r - F_r \cdot r = J_w \dot{\omega}_r$$

where:

F_r : is the reaction force applied to the right wheel by the rest of the robot;

f_r : is the friction force between the right wheel and the ground;

m_w : is the mass of the wheel;

τ_r : is the torque acting on the right wheel which provided by the right motor;

r : is the radius of the wheel;

J_w : is the inertia of the wheel.

Note that the Coriolis part had been deleted since it is negligible due to the fact that the wheel inertia is much smaller than the robot inertia.

The dynamic model of the robot can be defined as [76]:

$$M(\xi)\ddot{\xi} + J(\xi,\dot{\xi})\dot{\xi} + F = \tau$$ Eq. (4-8)

where:

$$M(\xi) = \begin{bmatrix} \frac{(mr^2\cos\theta + 2J_0\cos\theta)}{2r} & \frac{(mr^2\sin\theta + 2J_0\sin\theta)}{2r} & \frac{(mr^2 ed\sin^2\theta - mr^2 ed\sin\theta\cos\theta + J_c r^2 + 2J_0 d^2)}{2rd} \\ \frac{(mr^2\cos\theta + 2J_0\cos\theta)}{2r} & \frac{(mr^2\sin\theta + 2J_0\sin\theta)}{2r} & \frac{(mr^2 ed\sin^2\theta - mr^2 ed\sin\theta\cos\theta - J_c r^2 - 2J_0 d^2)}{2rd} \end{bmatrix}$$

$$J(\xi,\dot{\xi}) = \begin{bmatrix} \frac{-J_0\dot{\theta}\sin\theta}{r} & \frac{J_0\dot{\theta}\cos\theta}{r} & \frac{-mre\dot{\theta}\cos\theta(\sin\theta + \cos\theta)}{2} \\ \frac{-J_0\dot{\theta}\sin\theta}{r} & \frac{J_0\dot{\theta}\cos\theta}{r} & \frac{-mre\dot{\theta}\cos\theta(\sin\theta + \cos\theta)}{2} \end{bmatrix}$$

$$F = \begin{bmatrix} \frac{-f_n er}{d} \\ \frac{-f_n er}{d} \end{bmatrix}, \ \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, \ \xi = \begin{bmatrix} x_c \\ y_E \\ \theta \end{bmatrix}$$

To customize the dynamic model for the Bearcat III, the values for $m, r, J_0, e, d, J_c, f_n$ in Eq. 8 are substituted by $m = 306.18kg$, $r = 0.2095m$, $e = 0.338m$, $d = 0.432m$, and J_0, J_c, f_n need to be calculated according to Figure 4-9 for Bearcat III.

The value of the frictional coefficient μ between the ground and the wheel depend of the type of the surface of the ground; for grass, 0.6 is common, while for concrete 0.9 is usually used. Bearcat III usually moves on grass, therefore, 0.6 was used in the calculations. Substituting the parameters for Bearcat III into the normal force equation $f_n = \mu(\frac{1}{3}mg + m_w g)$, f_n is calculated to be 629.45 N.

The moment of inertia for the robot wheel is calculated as follows:

$$J_w = \frac{1}{2} m_t (r_{te}{}^2 - r_{ti}^2) + \frac{1}{2} m_r (r_{re}^2 - r_{ri}^2) = 0.055kgm^2 \qquad \text{Eq. (4-9)}$$

Substituting the value of J_w from Eq. 4-9 for Bearcat III, J_0 is calculated to be $0.274kgm^2$. For more details, refer to Alhaj Ali et al. [7, 75].

Substituting these values into Eq. 4-8, the Bearcat III dynamic model is:

$$M_B(\zeta)\ddot{\zeta} + J_B(\zeta, \dot{\zeta})\dot{\zeta} + G_B(\zeta, \dot{\zeta}, \ddot{\zeta}) = \tau \qquad \text{Eq. (4-10)}$$

where:

$$\zeta = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix}, \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$M_B(\zeta)=\begin{bmatrix}33.454\cos\theta & 33.454\sin\theta & 10.866\sin^2\theta-10.866\sin\theta\cos\theta+11.014\\ 33.454\cos\theta & 33.454\sin\theta & 10.866\sin^2\theta-10.866\sin\theta\cos\theta-11.014\end{bmatrix}$$

$$J_B(\zeta,\dot{\zeta})=\begin{bmatrix}-1.305\dot{\theta}\sin\theta & 1.305\dot{\theta}\cos\theta & -10.866\dot{\theta}\cos\theta(\sin\theta+\cos\theta)\\ -1.305\dot{\theta}\sin\theta & 1.305\dot{\theta}\cos\theta & -10.866\dot{\theta}\cos\theta(\sin\theta+\cos\theta)\end{bmatrix}$$

$$G_B(\zeta,\dot{\zeta},\ddot{\zeta})=\begin{bmatrix}11.308\dot{\theta}^2\sin^2\theta-0.441\dot{\theta}^2\cos^2\theta-11.308\ddot{\theta}\sin\theta\cos\theta-103.422\\ 11.308\dot{\theta}^2\sin^2\theta-0.441\dot{\theta}^2\cos^2\theta-11.308\ddot{\theta}\sin\theta\cos\theta-103.422\end{bmatrix}$$

4.15 Robot calibration

A robot can be thought of as an intelligent connection of perception to action. Implementing this is a formidable task and might take on a wide variety of disciplines, ranging from mechanical logic to microprocessor control to networks of neuron-like gates. Mobile robots pose a unique challenge to artificial intelligence researchers. They are inherently autonomous and they force us to deal with key issues such as uncertainty, reliability and real time response. They also require an integration of mechanical strength, reliable control systems, and sensors for vision and obstacle avoidance. Navigation and mapping are crucial to all robotic systems and are an integral part of autonomous mobile robots. While it is possible for a robot to be mobile and not do mapping and navigation, sophisticated tasks require that a mobile robot build maps and use them to move around. Levitt and Lawton (1990) pose three basic questions that define mobile robot mapping and navigation [77, 78]:

1. Where am I?

2. How do I get to other places from here?

3. Where are the other places relative to me?

Humans receive a large amount of their information through the human vision system, which enables them to adapt quickly to changes in their environment. Vision-based mobile robot guidance has proven difficult for classical machine vision methods because of the diversity and real time constraints inherent in the task. Vision for motion control must always be real time vision. In this context of unmanned guided vehicles, the vision system must enable the robot to perceive changes in its environment while they are occurring and soon enough to react to these changes and make decisions accordingly. For mobile systems, a speed of reaction similar to human beings is desirable. For this, a robot vision system should not introduce a delay of more than 100 ms in reporting an event in the environment or in providing data for some visible motion.

There are several factors, which affect the functioning of the outdoor autonomous systems. Some are variations of road type, appearance variations due to lighting and weather conditions, real time processing constraints and high level reasoning constraints. A general autonomous vehicle should be capable of driving on a variety of road surfaces like grass, concrete, sand, boards etc. The vehicle should function equally well inside, i.e., on a plane surface as well as outside on a varying terrain. The second factor making autonomous driving difficult is the variation in appearance that results from environmental factors. Lighting changes and deep shadows make it difficult for perception systems to pick up important and desired features during daytime driving. The threshold of the perception system has to be adjusted in such a way that the desired features are identified correctly. Any change to the light affects the threshold and performance of the system. Also to be considered is the fact that missing or obscured lane markers make driving difficult for an autonomous system even under favorable lighting conditions. Adequate computer hardware is a key to practical robot vision. Multi-processor systems containing a small number of processing elements, each of them based on a standard microprocessor of moderate performance have been demonstrated to outperform much more expensive computer systems in robot vision applications. Flexibility is a very important factor, including the flexibility of random access pixel data by the processing elements, and flexibility in dynamically concentrating the computing power of the system on those parts of an image containing the most relevant information at any moment. The system should also be flexible in restructuring under software control to match the inherent structure of the vision task. There is always a limited amount of time for processing sensor information. The speed of the front end processing system should be such that the vehicle reacts very quickly to the changes in the environment. For example at 5 miles per hour a vehicle is traveling nearly 7.5 feet per second. A lot can happen in 7.5 feet, like losing track of the lane or straying a significant distance from the lane or colliding with an obstacle if the system does not react accurately or act quickly enough.

To simplify this tedious calibration an artificial neural network can be used. Image processing is used to automatically detect calibration points. Then a back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates. This transformation is the main focus of this study.

The three dimensional (3-D) vision system makes use of 2 CCD cameras and an image-tracking device for the front end processing of the image captured by the camera. The camera reduces the three dimensional world co-ordinate system into two dimensional image co-ordinate system. After getting the information regarding image co-ordinates, at any time, the challenge is to extract three-dimensional information from them. A mathematical as well as geometrical transformation occurs via the camera parameters in transforming a 3-D coordinate system to a 2-D system. If these mathematical and geometrical relations are known, a 3-D coordinate point on a line can be autonomously determined from its corresponding 2-D image point. To establish these mathematical and geometrical relationships, the camera has to be calibrated. This is because if the vision system is well calibrated, accurate measurements of the coordinates of the points on the line with respect to the robot can be made. From these measurements, the orientation of the line with respect to the robot can be computed. With these computations, the next task is to guide the robot. The motion control of the AGV designed has the capability of turning about the center of its drive axis, which is called the zero turning radius feature. It is gaining popularity and expanding commercially in the U.S. mowing market. This feature provides exceptional maneuverability and can make sharp turns possible with relatively greater ease than those without the ZTR (Zero turning radius) feature. Rotating one wheel forward and the other wheel backward generally accomplishes the ZTR function. However in our design we instead vary the speeds of the left and right drive wheels while negotiating a curve. This enables the AGV to make a curved turning path parallel to the track lines.

Calibration of a camera means determining the geometric properties of the imaging process i.e. the transformation that maps a 3-D point, expressed with respect to a reference frame onto its 2-D image whose co-ordinates are expressed in pixel units. This problem has been a major issue in photogrammetry and computer vision for years. The main reason for such interest is that the knowledge of the imaging parameters allows one to relate the image measurements to the spatial structure of the observed scene [79]. The fundamental theorem of robot vision says that manipulation of a point in space *x1* by either a robot manipulator that moves it to another point *x2* or through a camera system that images the point onto a camera sensor at *x2*, is described by the a matrix transformation, which is of the form $X2 = Tx1$. The transformation matrix T describes the first-order effects of translation, rotation, scaling, and projective and perspective projections. Camera calibration is a complex problem because of the following problems:

1. Calibration of internal parameters of a camera, the so-called intrinsic parameters, including the optical and mechanical (geometrical) properties of the camera, such as focal length, lens distortion parameters, the intersection point of the optical axis with the image plane etc. Sometimes the manufacturers supply these parameters but they are usually not accurate enough for computations. Some of them such as focal length vary with adjustments, while some of them such as the lens center are calibrated once and for all depending upon the optical stability of the camera.

2. Estimation of the location of the camera (system) relative to the 3-D world reference system, including rotation and translation between these two systems is required. These are called extrinsic parameters. These parameters are not directly related to the camera itself, but the set up of a camera, which means they have to be calibrated at each set up.

Robert in his paper "Camera Calibration without Feature Extraction" has presented an approach to this problem using a calibration pattern [80]. The approach is different from the classical calibration techniques, which involve extraction of image features and computation of camera coefficients. A classical iterative technique is used to search for the camera parameters that best project 3-D points of a calibration pattern. Li and Lavest have thrown light on some aspects of zoom lens camera calibration [81]. A lot of care has to be taken in the electronic stability of the camera and frame grabber, and the way calibration points are measured and detected in images. In that paper they have addressed some practical aspects of camera calibration, in particular, of a zoom lens system. Through a systematic approach they describe all the keys points that have to be checked in order to obtain accurate calibration results.

Caution is required during calibration. Hong, et al. list two points that should be considered in camera calibration [82]:

1. Reducing the location error of image features as far as possible, by exploiting image processing technique, and

2. Compensating system error by the optimal pattern of approximating residual error of image points, namely the posterior compensation of the system error.

Based on these two points, the calibration process discussed by Weng et al. are of three parts [83]: (1) The direct transformation error approximation camera calibration algorithm; (2) the sub pixel image feature location algorithm combined with the 3D control point field delicate design and fabrication; (3) The precisely movable stage, which provides the reliable means of accuracy checking. Tsai presented an algorithm that decomposes a solution for 12 transformational parameters (nine for rotation and three for translation) into multiple stages by introducing a radial alignment constraint [84]. The radial alignment constraint assumes that the lens distortion occurs only in the radial direction from the optical axis Z of the camera. Using this constraint, six parameters are computed first, and the constraint of the rigid body transformation is used to compute five other parameters.

The remaining parameters are computed by radial lens distortion parameter and estimating it by a nonlinear optimization procedure.

Zhang et. al in "Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints" describe a method for computing the movement of objects as well as that of a mobile robot from a sequence of stereo frames [85]. Stereo frames are obtained at different instants by a stereo rig, when the mobile robot navigates in an unknown environment possibly containing some moving rigid objects.

Zhang et al. present a method for estimating 3D displacements from two stereo frames [86]. It is based upon the hypothesize-and-verify paradigm, which is used to match 3D line segments between the two frames. In order to reduce the complexity of the algorithm, objects are assumed to be rigid. In the experimental sections, the algorithm is shown to work on indoor and natural scenes. "A 3D World Model Builder with a Mobile Robot" - An article written by the same authors describes a system to incrementally build a world model with a mobile robot in an unknown environment [87]. The model is segment-based. A trilocular stereo system is used to build a local map about the environment. A global map is obtained by integrating a sequence of stereo frames taken when the robot navigates in the environment. Luong, et al. in their paper "Motion of an Uncalibrated Stereo Rig: Self-Calibration and Metric Reconstruction" address the problem of self-calibration and metric reconstruction (up to a scale) from one unknown motion of an uncalibrated stereo rig, assuming the coordinates of the principal point of each camera are known [88]. They also present a novel technique for calibrating a binocular stereo rig by using the information from both scenes and classical calibration objects. The calibration provided by the classical methods is only valid for the space near the position of the calibration object. Their technique takes the advantage of the rigidity of the geometry between two cameras. The idea is to first estimate precisely the epipolar geometry, which is valid for a wide range in space from all available matches.

During the execution of a task the vision-system is subject to external influences such as vibrations, thermal expansion etc. which affect and possibly render invalid the initial calibration. Moreover, it is possible that the parameters of the vision-system such as the zoom or the focus are altered intentionally in order to perform specific vision-tasks.

"Self-Maintaining Camera Calibration over Time" by Schenk et al. describes a technique for automatically maintaining calibration of stereovision systems over time without using again any particular calibration apparatus [86]. Worrall, Sullivan and Baker in the paper "A simple, intuitive camera calibration tool for natural images" present an interactive tool for calibrating a camera, suitable for use in outdoor scenes [89]. The motivation for the tool was the need to obtain an approximate calibration for images taken with no explicit calibration data. The method decomposes the calibration parameters into intuitively simple components, and relies on the operator interactively adjusting the parameter settings to achieve a visually acceptable agreement between a rectilinear calibration model and his own perception of the scene.

Most of the previous research deals with the theoretical aspects of zoom lens camera calibration. The intrinsic parameters are obtained by building a pinhole camera model. The research here deals with designing a calibration algorithm keeping in mind the significant practical aspects.

4.15.1 Line following

The objective of the vision system is to make the robot follow a line using a camera. In order to obtain accurate information about the position of the line with respect to the centroid of the robot, the distance and the angle of the line with respect to the centroid of the robot has to be known. The camera system reduces the 3-D information about the obstacle course into 2-D image co-ordinates. In order to obtain a relationship between the two co-ordinate systems, the camera needs to be calibrated. Camera calibration is a process to determine the relationship between a given 3-D coordinate system (world coordinates) and the 2-D image plane a camera perceives (image coordinates). More specifically, it is to determine the camera and lens model parameters that govern the mathematical or geometrical transformation from world coordinates to image coordinates based on the known 3-D control field and its image. The CCD camera maps the line from the 3-D coordinate system to the 2-D image system. Since the process is autonomous, the relationship between the 2-D system and the 3-D system has to be accurately determined so that the robot can be appropriately controlled to follow the line. The objective of this section is to explain the entire calibration process and its significance in this project.

The model of the mobile robot illustrating the transformation between the image and the object is shown in Figure 4-10. The robot is designed to navigate between two lines that are spaced 10 feet apart. The lines are nominally 4 inches wide but are sometimes dashed. This requires a two-camera system design so that when a line is missing, the robot can look to the other side via the second camera. Measurements are referenced to the robot centered as a global coordinate system. For navigation, the cameras must be located with respect to this centroid.

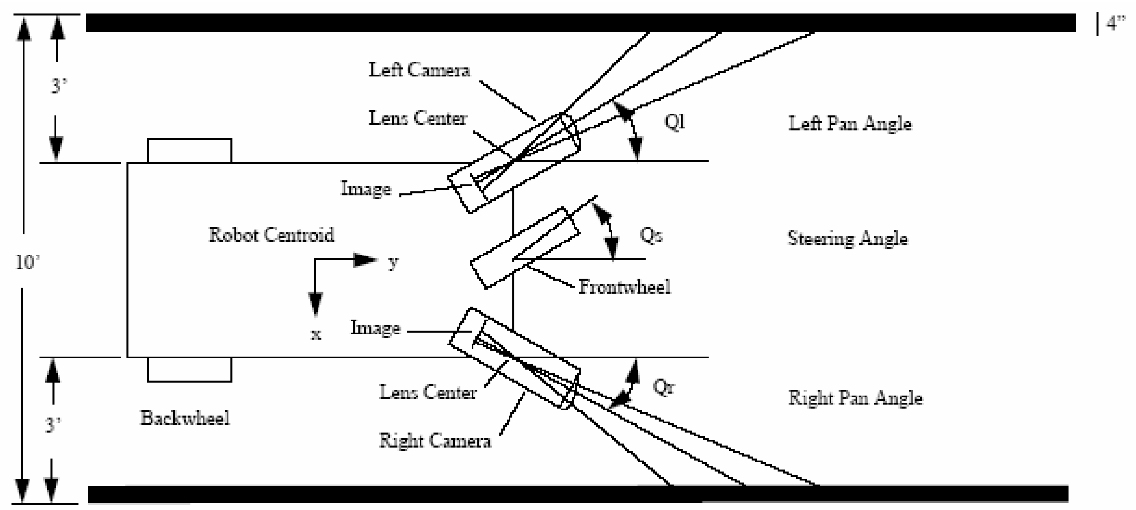


Figure 4-10: Top view model of the robot in the obstacle course

Camera calibration is considered very important in many computer vision problems. Camera calibration in the context of three-dimensional machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters). Camera projection is often modeled with a simple pinhole camera model. In reality, the camera is a much more complicated device, and if it is used as a measurement instrument, a proper calibration procedure should be performed. In order to follow a track, which is separated by two lines, which are 10 ft apart, 2 CCD cameras and an image-tracking device (ISCAN) are used. The ISCAN image tracking system finds the centroid of the darkest or the brightest region in an image and returns the co-ordinates of these points. These are the image coordinates. These coordinates are two-dimensional while the real world co-ordinates are three-dimensional. An algorithm is developed to establish a mathematical and geometrical relationship between the physical three-dimensional (3-D) and its corresponding digitized two-dimensional (2-D) co-ordinates. In an autonomous situation the challenge is to determine 3-D co-ordinates given the image co-ordinates. This is established by what is popularly known as "Calibration" of the camera. The objective is to find any corresponding ground co-ordinate given an image co-ordinate. What makes this the most important and crucial task is that the process of following the line is autonomous and dynamic and hence the relationship between these co-ordinates should be accurately determined. This, in turn, determines how closely the robot follows the line and hence the success of the robot.

The objective was to design a calibration method, which was not only accurate but also is easy and less time consuming. Some calibration methods are very accurate but are extremely time consuming. Two approaches for camera calibration are discussed here.

4.15.2 Matrix method

Camera calibration is a process to determine the relationship between a given 3-D coordinate system (world coordinates) and the 2-D image plane a camera perceives (image coordinates). These co-ordinates are two-dimensional while the real world co-ordinates are three-dimensional.

An algorithm is developed to establish a mathematical and geometrical relationship between the physical three-dimensional (3-D) and its corresponding digitized two-dimensional (2-D) co-ordinates [90-92].

The simple pinhole camera model for a perspective projective transformation is shown in Figure 4-11 and is easily modeled in homogeneous coordinates by the matrix transformation from ground to homogeneous image coordinates.

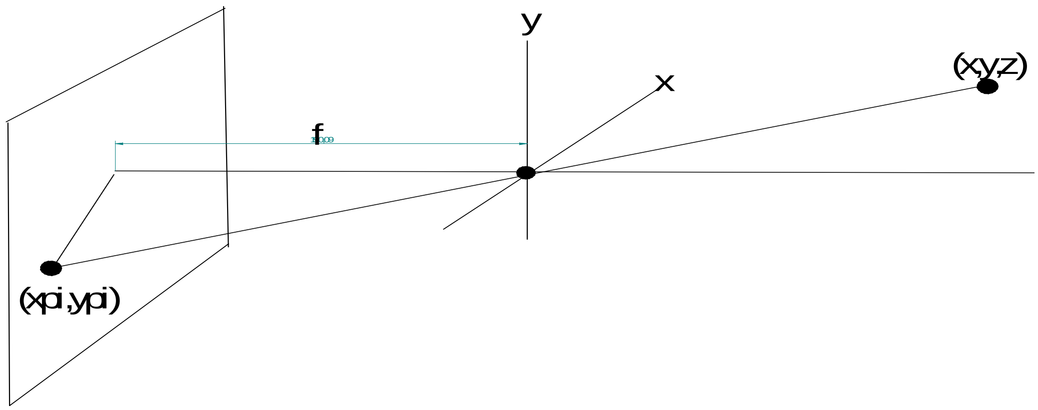


Figure 4-11: Perspective projection onto an image plane

$$\begin{bmatrix} Wx_{PI} \\ Wy_{PI} \\ Wz_{PI} \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} * \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$ Eq. (4-11)

The physical image coordinates are determined by dividing by the scale term W.

$$\left(x_{PI}, y_{PI}, z_{PI}\right) = (Wx_{PI}/W, Wy_{PI}/W, Wz_{PI}/W)$$ Eq. (4-12)

In general the lens center may be translated and rotated with respect to the global coordinate system. This results in a general perspective projective matrix representation as shown. With projection, the image z term and corresponding matrix row can be considered discarded.

$$\begin{bmatrix} Wx_{PI} \\ Wy_{PI} \\ Wz_{PI} \\ W \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} * \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$ Eq. (4-13)

The stereo vision principles approach uses the scaling between the two coordinate systems to determine the relationship between the physical and image coordinates. If the image is projected on the z axis, the model equations relating the two coordinates are described by:

$$Wx_{pi} = A_{11}x_g + A_{12}y_g + A_{13}z_g + A_{14}$$

$$Wy_{pi} = A_{21}x_g + A_{22}y_g + A_{23}z_g + A_{24} \qquad \text{Eq. (4-14)}$$

$$W = A_{31}x_g + A_{32}y_g + A_{33}z_g + A_{34}$$

The coefficients for the above equations may be computed by eliminating the scaling term, W, from the first two equations to obtain Equations 4-15 and 4-16. Then one can utilize matching calibration points to solve for the unknown A coefficients.

$$A_{11}x_g + A_{12}y_g + A_{13}z_g + A_{14} - A_{31}x_g x_{pi} - A_{32}y_g x_{pi} - A_{33_g} z_g x_{pi} = 0 \qquad \text{Eq. (4-15)}$$

$$A_{21}x_g + A_{22}y_g + A_{23}z_g + A_{24} - A_{31}x_g x_{pi} - A_{32}y_g x_{pi} - A_{33_g} z_g x_{pi} = 0 \qquad \text{Eq. (4-16)}$$

Equations 4-15 and 4-16 represent two equations in twelve unknown coefficients, A_{nm}. The coefficients are computed using magic matrix techniques by utilizing six matching calibration points. Since we are dealing with a homogeneous coordinate system, the matrix will include an arbitrary scale factor. If the coefficient A_{34} is set as unity, the resulting transformation matrix will be normalized.

With six calibration data points and $A_{34} = 1$, the following matrix equation was formulated.

$$QA = 0 \qquad \text{Eq. (4-17)}$$

Where

$$Q=\begin{bmatrix} xg1 & yg1 & zg1 & 1 & 0 & 0 & 0 & 0 & -xg1.xPI1 & -yg1.xPI1 & -zg1.xPI1 & -xPI1 \\ 0 & 0 & 0 & 0 & xg1 & yg1 & zg1 & 1 & -xg1.yPI1 & -yg1.yPI1 & -zg1.yPI1 & -yPI1 \\ xg2 & yg2 & zg2 & 1 & 0 & 0 & 0 & 0 & -xg2.xPI2 & -yg2.xPI2 & -zg2.xPI2 & -xPI2 \\ 0 & 0 & 0 & 0 & xg2 & yg2 & zg2 & 1 & -xg2.yPI2 & -yg2.yPI2 & -zg2.yPI2 & -yPI2 \\ xg3 & yg3 & zg3 & 1 & 0 & 0 & 0 & 0 & -xg3.xPI3 & -yg3.xPI3 & -zg3.xPI3 & -xPI3 \\ 0 & 0 & 0 & 0 & xg3 & yg3 & zg3 & 1 & -xg3.yPI3 & -yg3.yPI3 & -zg3.yPI3 & -yPI3 \\ xg4 & yg4 & zg4 & 1 & 0 & 0 & 0 & 0 & -xg4.xPI4 & -yg4.xPI4 & -zg4.xPI4 & -xPI4 \\ 0 & 0 & 0 & 0 & xg4 & yg4 & zg4 & 1 & -xg4.yPI4 & -yg4.yPI4 & -zg4.yPI4 & -yPI4 \\ xg5 & yg5 & zg5 & 1 & 0 & 0 & 0 & 0 & -xg5.xPI5 & -yg5.xPI5 & -zg5.xPI5 & -xPI5 \\ 0 & 0 & 0 & 0 & xg5 & yg5 & zg5 & 1 & -xg5.yPI5 & -yg5.yPI5 & -zg5.yPI5 & -yPI5 \\ xg6 & yg6 & zg6 & 1 & 0 & 0 & 0 & 0 & -xg6.xPI6 & -yg6.xPI6 & -zg6.xPI6 & -xPI6 \\ 0 & 0 & 0 & 0 & xg6 & yg6 & zg6 & 1 & -xg6.yPI6 & -yg6.yPI6 & -zg6.yPI6 & -yPI6 \end{bmatrix} \quad \text{Eq. (4-18)}$$

and

$$A=\left[A_{11},A_{12},A_{13},A_{14},A_{21},A_{22},A_{23},A_{24},A_{31},A_{32},A_{33},A_{34}\right]^{Tr} \quad \text{Eq. (4-19)}$$

There are 12 unknowns in the matrix equation. However, since the matrix equation is homogeneous, there is an arbitrary value or scale. The transformation coefficients can be solved by moving the last column in the matrix Q to the right-hand side and applying the least square regression method. Therefore, one coefficient can be arbitrarily selected leaving 11 coefficients to be determined. Since each image point (x, y) gives two equations, a minimum of five and one half image points could give a solution. A greater number of points permit a least squares solution. After the A coefficients are determined, W is computed and for any image coordinate, x_{pi} and y_{pi}, the corresponding ground coordinates may be computed as shown in the following.

$$Q_1 A=[B]$$
$$\textit{where}$$

$$\begin{bmatrix} xg1 & yg1 & zg1 & 1 & 0 & 0 & 0 & 0 & -xg1.xPI1 & -yg1.xPI1 & -zg1.xPI1 \\ 0 & 0 & 0 & 0 & xg1 & yg1 & zg1 & 1 & -xg1.yPI1 & -yg1.yPI1 & -zg1.yPI1 \\ xg2 & yg2 & zg2 & 1 & 0 & 0 & 0 & 0 & -xg2.xPI2 & -yg2.xPI2 & -zg2.xPI2 \\ 0 & 0 & 0 & 0 & xg2 & yg2 & zg2 & 1 & -xg2.yPI2 & -yg2.yPI2 & -zg2.yPI2 \\ xg3 & yg3 & zg3 & 1 & 0 & 0 & 0 & 0 & -xg3.xPI3 & -yg3.xPI3 & -zg3.xPI3 \\ 0 & 0 & 0 & 0 & xg3 & yg3 & zg3 & 1 & -xg3.yPI3 & -yg3.yPI3 & -zg3.yPI3 \\ xg4 & yg4 & zg4 & 1 & 0 & 0 & 0 & 0 & -xg4.xPI4 & -yg4.xPI4 & -zg4.xPI4 \\ 0 & 0 & 0 & 0 & xg4 & yg4 & zg4 & 1 & -xg4.yPI4 & -yg4.yPI4 & -zg4.yPI4 \\ xg5 & yg5 & zg5 & 1 & 0 & 0 & 0 & 0 & -xg5.xPI5 & -yg5.xPI5 & -zg5.xPI5 \\ 0 & 0 & 0 & 0 & xg5 & yg5 & zg5 & 1 & -xg5.yPI5 & -yg5.yPI5 & -zg5.yPI5 \\ xg6 & yg6 & zg6 & 1 & 0 & 0 & 0 & 0 & -xg6.xPI6 & -yg6.xPI6 & -zg6.xPI6 \\ 0 & 0 & 0 & 0 & xg6 & yg6 & zg6 & 1 & -xg6.yPI6 & -yg6.yPI6 & -zg6.yPI6 \end{bmatrix} * \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \\ A_{14} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{24} \\ A_{31} \\ A_{32} \\ A_{33} \\ \end{bmatrix} = \begin{bmatrix} xPI1 \\ yPI1 \\ xPI2 \\ yPI2 \\ xPI3 \\ yPI3 \\ xPI4 \\ yPI4 \\ xPI5 \\ yPI5 \\ xPI6 \\ yPI6 \end{bmatrix}$$ Eq. (4-20)

To solve for the coefficients, one may use the Moore-Penrose pseudo inverse. First multiply by the transpose of the reduced matrix Q_1.

$$Q_1^T * Q_1 * A = Q_1^T B$$

Then multiply both sides by the inverse of the squart matrix $Q_1^T * Q_1$

$$A = (Q_1^T * Q_1)^{-1} Q_1^T B$$ Eq. (4-21)

Now given the ***A*** matrix coefficients, and the physical image coordinates, one may determine the three dimensional ground coordinates. If this pseudo inverse matrix computation comes out ill conditioned or with a small condition number, another way of doing this computation is needed. For example, the original equations may be rewritten in matrix form as:

$$\begin{bmatrix} W * x_{PI} \\ W * y_{PI} \\ W \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{bmatrix} * \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$ Eq. (4-22)

$$\begin{bmatrix} W * x_{PI} \\ W * y_{PI} \\ W \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} + \begin{bmatrix} A_{14} \\ A_{24} \\ A_{34} \end{bmatrix}$$

or

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} W * x_{PI} - A_{14} \\ W * y_{PI} - A_{24} \\ W - A_{34} \end{bmatrix}$$

or

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = inv(A) * \begin{bmatrix} W * x_{PI} - A_{14} \\ W * y_{PI} - A_{24} \\ W - A_{34} \end{bmatrix} \qquad \text{Eq. (4-23)}$$

Where inv(A) is the inverse of the 3 by 3 A matrix, W is the scaling factor, A_{nm} are coefficients, x_{pi} and y_{pi} are *x* and *y* image coordinates, and x_g, y_g and z_g are the ground coordinates.

So with this formulation, the ground coordinates can be computed with an estimate of the single scale parameter W. Further experimentation is needed to determine if this method is robust [92].

4.15.3 Calibration using Neural Networks

The purpose of this section is to simplify this tedious calibration using an artificial neural network. Image processing is used to automatically detect calibration points. Then a back projection neural algorithm is used to learn the relationships between the image coordinates and three-dimensional coordinates.

Static backpropagation is used to produce an instantaneous mapping of a static (time independent) input to a static output. At the core of all back propagation methods is an application of the chain rule for ordered partial derivatives to calculate the sensitivity that a cost function has with respect to the internal states and weights of a network. In other words, the term backpropagation is used to imply a backward pass of error to each internal node within the network, which is then used to calculate weight gradients for that node. Learning progresses by alternately propagating forward the activations and propagating backward the instantaneous errors.

A Backpropagation network consists of at least three layers of units: an input layer, at least one intermediate hidden layer, and an output layer. Typically, units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. When a backpropagation network is cycled, an input pattern is propagated forward to the output units through the intervening input-to-hidden and hidden-to output weights.

The data used to train the neural network is as follow:

t= [48.856, 56.346, 56.346, 48.831, 52.919, 52.87, 44.55, 47.75, 46.75, 46, 49.75, 40.1, 58, 60, 62, 60, 59, 61, 59.50, 62].
p= [220, 187, 135, 158, 153, 124, 245, 229, 221, 195, 168, 195, 138, 124, 156, 174, 149, 145, 124, 182].

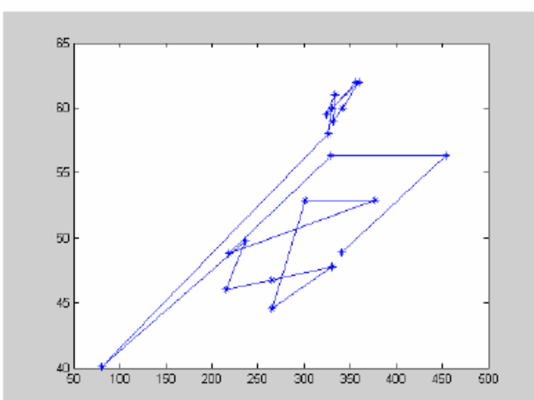


Figure 4-12: Training plot of x coordinate

(Trained values are represented by * and actual values are represented by +)

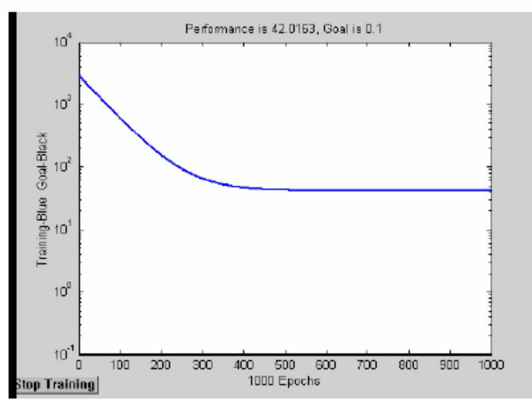


Figure 4-13: Performance plot for x

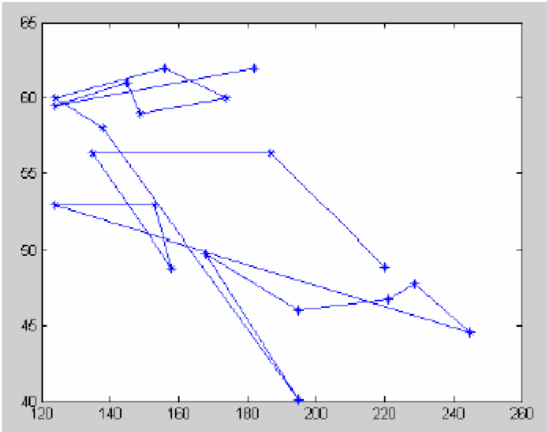


Figure 4-14: Training plot of y coordinate

(Trained values are represented by * and actual values are represented by +)

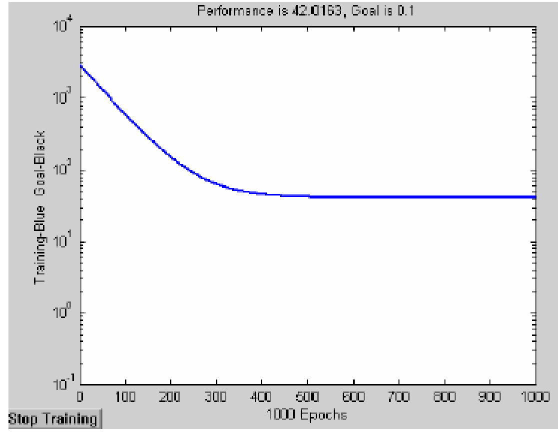


Figure 4-15: Performance plot for y

(Published in reference 78 [78])

4.16 Remote control

Recent progress in Internet capabilities has made it easier to use as a reliable and widely accessible communication framework. Remote control via the Internet is a very young field of research that could have significant applications in the near future. Robotics, manufacturing, traffic control, space exploration, health care, disaster rescue, house cleaning, security, inspection and tele-presence are examples of such applications [93].

Since the first networked device, “Cambridge coffeepot,” appeared on the Internet, a rapid enlargement of the WWW over the past several years has resulted in a growing number of tele-robotics sites and Web accessible devices [94, 95].

Previous researchers have had different approaches to accessibility from the Internet. Availability for public users has been a goal for most projects, but others have focused on special user devices. For example, by 1995, Goldberg et al. had developed a tele-robotic system called the Tele-Garden by which WWW users are able to observe, plant and nurture life in a remote garden [95, 96]. Likewise, Peterson et al. developed a system for tele-pathology by the Internet. This system allows any Internet user to become a consultant for tele-pathology without the acquisition of specialized hardware or software [96].

Rovetta et al. used a mix of communication media for performing tele-surgery in 1995. Their work was based on a special-user access and not a public access Internet [97]. Various other devices have become available over time, such as the Programmable Logic Control for a chemical experiment [98], Microscope [96, 99, 100], Blimp Space Browser, Nuclear Microprobe [101] and Web Camera [102]. In fact, Web cameras are the most common Internet connected devices [94].

The merge of the Internet and manufacturing technologies has resulted in bridging of the gap between engineering technology (such as a rapid prototyping hardware system) and information systems to enable the remote control of engineering resources [103, 104]. Wang et al. presented the concept of an Internet assisted manufacturing system for agile manufacturing practice [105]. In this system, a local user is able to introduce design specifications to a product information system and the Central Network Server can generate complete CAD/CAPP/CAM/CAA files and control the remote FMS or CNC machines to accomplish the whole production process.

Tele-robotics is also an active branch in Internet connected devices. Schiling developed an educational inspection mobile robot for tele-diagnosis of malfunctions, tele-maintenance of machines, and tele-monitoring of remote sites by sensors and tele-operations of remote equipment, including robots [106]. In another experience, Winfield et al. developed a system that can control several robots from a Local Area Network simultaneously [107].

All these systems are not yet commercially available. In fact, the limitation of bandwidth, safety, harmonizing the remote activities and time delays are the major concern of research in this area. In many situations, a human is needed to control these machines. In the existing set up, there are very few tools that offer a remote access to the robot, and its scope is also limited.

Description of the design and development of an interface for remote control of the Bearcat Cub robots via the Internet can be found in [93].

4.17 JAUS standard

The Joint Architecture for Unmanned Systems (JAUS) is a data communication standard targeted toward unmanned systems. The purpose of JAUS is to support the acquisition of Unmanned Systems by providing a mechanism for reducing system life-cycle costs. This is accomplished by providing a framework for technology reuse/insertion. JAUS defines a set of reusable "components" and their interfaces. These reusable components not only reduce the maintenance costs of a system, but also dramatically reduce the development costs of any follow-on system(s). Reuse allows a component developed for one Unmanned System to be readily ported to another Unmanned System or to be easily replaced when technological advances.

Technology insertion is achievable when the architecture is designed to be both modular and scaleable. Components that are deemed necessary for the mission of the Unmanned System may be inserted simply by bundling.

JAUS defines components for all classifications of Unmanned Systems from remote control toward autonomous, regardless of application. As a particular system evolves, the architecture is already in place to support more advanced capabilities [108].

Technical constraints are imposed on JAUS to ensure that the architecture is applicable to the entire domain of Unmanned Systems - now and in the future. The constraints are:

- Platform Independence
- Mission Isolation
- Computer Hardware Independence
- Technology Independence

A simple set of JAUS commands were implemented on Bearcat Cub robot. The commands intended to start the vehicle moving forward in the autonomous mode, stop the vehicle from moving in the autonomous mode, and activate a warning device (horn/light). The JAUS messages were sent over an 802.11g link.

4.18 Robot application case 1: mine clearing

An estimated 100 million landmines which have been planted in more than 60 countries kill or maim thousands of civilians every year. Millions of people live in the vast dangerous areas and are not able to access to basic human services because of landmines' threats. This problem has affected many third world countries and poor nations which are not able to afford high cost solutions. This section tries to present some solutions for the mine clearing. It studies current situation of this crisis as well as state of the art robotics technology for the mine clearing. It also introduces a survey robot which is suitable for the mine clearing applications. The results show that in addition to technical aspects, this problem has many socio-economic issues [109]. Landmines do not distinguish between a soldier, a child or an animal. They can not be aimed and their deadly force is indiscriminant. That's why they are so horrible.

The first generation of mines were pressure-activated and large and used to stop or destroy enemy's vehicles. They could be found and neutralized easily by infantry. As a counter measure, armies developed anti-personnel mines to keep enemy mine clearers away from anti-vehicle mine fields. It is estimated that 75% of all uncleared mines are anti-personnel mines, and this is the category that has created most problems[110].

According to International Campaign to Ban Landmines (ICBL) leading producers and exporters of antipersonnel mines in the past 25 years include China, Italy, the former Soviet Union, and the United States. More than 50 countries have manufactured as many as 200 million antipersonnel landmines in the last 25 years and more than 350 different types of antipersonnel mines exist. Even if no more mines are ever laid, they will continue to maim and kill for years to come. In fact, they kill or injure more than 2000 people a month and with the current mine removal technology it may take about 1000 years to remove all mines if no new mines are buried in the war zones [111].

The 1997 Ottawa treaty bans the use, production, stockpiling, and transfer of antipersonnel landmines. Since the treaty became law, countries may no longer sign it, they must accede. Those countries which have already signed must still ratify it in order to be fully bound by the ban provisions. By the end of 2002, a total of 146 countries had signed the Mine Ban Treaty and 130 had ratified or acceded to it. Since then, 30 million stockpiled mines have been destroyed according to ICBL which monitors the treaty compliance[112].

Landmines have many social and economical impacts which can not be described by simple quantitative measures. Many communities have not been involved in proper clearance activities and have adapted to situation in their own ways. Global Landmine Survey is an international effort to understand the socio-economic impact of landmines and unexploded ordnance (UXO). Without knowing the impacts it is difficult to develop strategies to allocate limited resources to minimize the effect of landmines. Landmine resources compete with other humanitarian activities. The low and decreasing mortality from landmines is often compared to high and soaring mortality from epidemic disease. This has provoked an all-over-nothing debate over the costs and benefits of demining [113]. It is becoming clear that complete clearance is not a feasible solution of the worldwide landmine problem when the size of contaminated area is considered into account. That is why it is essential to understand the social and economical impacts of landmines.

4.18.1 Mine technology

4.18.1.1 Types of mines

The Mine Ban Treaty defines a mine as follow:

Anti-personnel (AP) landmine: "A mine designed to be exploded by the presence, proximity or contact of a person and that will incapacitate, injure or kill one or more persons."

Anti-tank (AT) landmine: An AT mines is a device designed to detonate by more than 100 kilograms of pressure -AT mines cannot distinguish between a tank and tractor.

ICBL categorize mines as follow:

“Blast mines: usually hand-laid on or under the ground or scattered from the air. The explosive force of the mine causes foot, leg, and groin injuries and secondary infections usually result in amputation.

Fragmentation mines: usually are laid on or under the ground and often activated by tripwire or other means. When detonated the explosion projects hundreds of fragments at ballistic speed of up to 50 meters resulting in fragmentation wounds. Some fragmentation mines contain a primary charge to lift the mine above the ground (about 1 to 1.5 meters) before detonating which can injure an adult's abdomen, genitals and take off a child's head.

Plastic mines: Undetectable by metal detectors used by deminers.

Remotely-delivered (R/D) or scatterable mines: Usually disseminated from aircraft, helicopters or artillery. Accurate mapping, recording and marking mines laid in this manner is impossible.

Anti-handling devices: A device intended to protect mine and which activates when an attempt is made to tamper with or otherwise intentionally disturb the mine (Mine Ban Treaty definition).

Self-destruct (S/D) mines: So-called "smart" mines are designed to self-destruct after a designated period of time. If they fail to self-destruct, these mines are also sometimes designed to self-deactivate. There is nothing smart about these mines though - while armed they cannot discriminate between the footfall of a soldier and a civilian.”

Most of mines are plastic or wooden mines with a small metal needle which is hard to detect using the well know metal detectors. Other metal objects in the same minefield create many false alarms. There are other technologies to detect mines. Neutron activation imaging, ion spectroscopy or x-ray tomography which are used for detection of explosive inside the luggage are not practical for mine detection yet. Ground penetrating radars can be used along with the metal detectors. Odor detectors also seem a promising technology for mine detection. Some use dogs to double check a cleared area and sometimes to survey the extent of a minefield before clearance begins. Their main use is to confirm suspected mined areas.

Cost is an important issue in the mine clearing. A clearing cost close to the cost of mine could also decrease the use of mines. It is estimated that even with the traditional demining technology average cost of demining is $800 per mine found [114].

There are also important differences between military and civilian demining efforts. In many military applications speed of operation is more important than the safety of soldiers since the objective is to punch a path though the minefield, with the acceptable losses. This is called “breaching”. In this case typically a tank pushes a heavy demining system and troops follow and a removal of 80% of mines is acceptable [115]. Figure 3 shows an example of such device.

The UN requirement of civilian mine clearing is 99.6%. Simple large rollers are not sufficient to meet the UN requirement. They leave most mines on the side berm they create, where the mine are more difficult to find [115].

Besides, many poor nations and civilian groups are not able to afford high cost military solutions. To be practical in large scale demining efforts the cost of demining system should be less $10,000 in mass production. This is some kind of threshold, suggested by some researchers [114]. This cost is mainly influenced by sensors.

4.18.1.2 Mine-clearing technology

The current mine clearing technology reflects a varied and diverse approach to diffuse anti-personnel land mines. They range from the old fashioned sniffer dogs to highly sophisticated polarized infrared technology. The costs of landmine clearing using sophisticated techniques are prohibitive for poor third world countries which have the majority of the dormant mines. The relatively primitive technique of detecting mines using a trained sniffer dogs and a trained deminer has a high human costs. It is estimated that for every 2000 mine cleared there is a fatal human error. The training required for personnel to disarm mine is even more complicated by the fact that there are almost thousands types and makes of anti-personnel mines. For example during the last days of Persian gulf war in 1991 they were many different kinds of mines were used like MK-118, Blu-77b, Blu-97, M-42 and 46, Blu-61-a-b, Blu-63-b, 86-b, Blu-91-b, Blu-92-b and bluga [116].

None of the technologies available seem in fact capable of reaching, in a very large number of situations, good enough detection while maintaining a low false alarm rate. Rather, each one will probably have to find, if it exists, a specific area of applicability, determined by technological as well as economical or even social factors, and possibly other sensors to work with using some form of sensor fusion. The need for a better exchange of information between the specialists in each category is obvious, using options such as data sharing on the Internet [117].

The following table lists the current technologies available or are in the process of being developed. These technologies can be leveraged to find the 'best of breed' which works for most mine clearing scenarios [117].

Table 4-1: Mine detection technologies

Sensor technology	Maturity	Cost and Complexity
Passive infrared	Near	Medium
Active infrared	Near	Medium
Polarized infrared	Near	Medium
Passive electro-optical	Near	Medium
Multi-hyperspectral	Far	High
Passive mm-wave	Far	High
mm-Wave radar	Near	High
Ground penetrating radar	Near	Medium
Ultra-wideband radar	Far	High
Active acoustic	Mid	Medium
Active seismic	Mid	Medium
Magnetic field sensing	Near	Medium
Metal detection	Available	Low
Neutron activation analysis	Near	High
Charged particle detection	Far	High
Nuclear quadrupole reason.	Far	High
Chemical sensing	Mid	High
Biosensors	Far	High
Dogs	Available	Medium

There are different approaches to detecting mines. Robots which can be equipped with different kinds of sensors and actuators depending on the mines that are being cleared seem to be a realistic option. The costs of these robots are reasonable if we consider the lives that can saved.

4.18.2 Examples of robots

The robots that are available in the market for detection and clearing mines are very different in their approach. Here we list a few which represents some of the typical approaches for mine clearing robots.

Pemex –BE: is a lightweight 2-wheels robots developed as a first cross-country test vehicle for searching anti-personnel mines as shown in Figure 4-16. The sensors are located inside a half-sphere which acts as a third supporting point. It weights less than 16 kg and can easily be dismantled and carried out as hand luggage. It is battery operated with autonomy of 60 minutes and can move at a speed of up to 6 km/h[118].

Advantages:

- The cheaper of all the robots
- Easiest to navigate across difficult terrain
- The very light weight robot

Disadvantages:

- The not safe for the operator.
- No sophisticated sensors

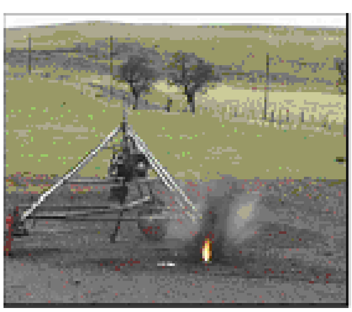


Figure 4-16. Pemex –BE

Figure 4-17. Dervish

Dervish:

The Dervish, shown in Figure 4-17, is a remote-controlled vehicle designed to detonate anti-personnel mines with charge weights up to 250 gm, equivalent to the largest size of anti-personnel mine. The Dervish detects and detonates anti-personnel mines by mimicking the ground loading of a human foot. It sweeps a path, 5 meters wide, covering ground at intervals of only 3cm [119].

Advantages:

- It can detonate the landmines
- Extremely safe for the operator
- It is very easy to use

Disadvantages:

- No sophisticated sensors
- Difficult to navigate

Figure 4-18: ILDP system

ILDP: The ILDP system consists of a teleoperated vehicle carrying three scanning sensors which operate while the system is in motion; a metal detector array (MMD) based on electromagnetic induction (EMI), an infrared imager (IR), ground penetrating radar

(GPR), and a confirmatory sensor which requires the system to be stationary and near a target of interest, consisting of a thermal neutron analysis (TNA) detector.[5]

Advantages:

- Highly sophisticated sensors
- Fastest land mine clearing robot
- Highly safe for the operator

Disadvantages:

- Training the operator is expensive

Figure 4-19: SHADOW DEMINER

Shadow Deminer is a robot capable of traversing an anti-personnel minefield carrying mine detecting sensors or video cameras. The robot is able to traverse rugged terrain and degrade gracefully in the event of damage. The Shadow Deminer for an eight-legged vehicle with emergent walking behavior using pneumatic actuators and local materials where possible. These factors contribute to the simplicity of the basic vehicle and low cost if destroyed.

Advantages:

- Highly efficient sensors
- Can climb inclines.
- High resolution area

Disadvantages:

- High cost of maintenance
- High initial investment

There are different ways that robot could help human in mine detection and mine clearing. Small autonomous vehicles equipped with different sensors could scan an area and determine the contaminated area. This phase when is done manually is very dangerous because deminers are working faster and talking more risks in compare to systematic search [120]. Once the polluted area or the actual location of a mine was specified then the systematic search and neutralizing process can begin. Even a robot can go to a pre-specified location by avoiding obstacles and place a detonator or some chemical to destroy the mine.

A light-weight small autonomous robot is an option for the mine clearing. Such robot could be cheap enough in mass production for many humanitarian applications. It should carry small weight and size sensors (which is still an unsolved problem). There are major subsystems for the robot.

Landmines are great treats to lives of millions of people and no perfect solution exists. In this section several state of the art mine clearing methods were investigated and some mine clearing robots were introduced. It also current status of international mine clearing activities were presented. The survey robot, which was developed by the authors, will be explained briefly and its applicability in mine clearing will be discussed. It can be concluded that much more research and development is needed to solve the global crisis of landmines.

4.19 Robot application case 2: soil sampling survey robot

A survey robot was developed by Peter Cao, Masoud Ghaffari, and Ernie Hall at the Center for Robotics Research at the University of Cincinnati which can be modified for mine clearing purpose. The robot has several subsystems. The survey robot shown in Figure 4-20 is equipped with the GPS navigation system with supervised remote control ability.

The overall function of the robot is to carry the soil sampling device to a targeted waypoint. Stop and let the soil sampling device sample the soil, and then send back sample data to the remote base.

The computer communicates with the soil sampling tube via two RS232 ports. There are two motion units in the robot. The robot is guided by the GPS receivers with a bounded error of approximately 10 feet. On the robot navigation side, both wheels rotate a that navigate the robot from one spot to the next; on the soil sampling unit side, the linear actuator pushes the penetrometer down for soil sampling and lift up after that. The sampling unit could be equipped with chemical sensors or it the whole sensors could be replaced by other mine detecting sensors. The original robot platform is a Friendly Robot lawnmower which cost only $500. The GPS system and the sensors will add an additional cost.

The proposed sensor system for soil sampling was constructed and fully verified the concept of sample soil properties with autonomous mobile robot is feasible. The soil sampling system was demonstrated to the air force officers at Hurlburt Air force Base in November 2002. The robot finished designated tasks on site with a better than expected accuracy. This is the first time an autonomous soil sampling sensor system was successfully integrated with a GPS guided mobile robot. The performance of this robot verified the concept that robot can take place of personnel for the soil sampling operation in unstructured environments. However, more needs to be done to add mine detecting sensors and proof the concept in a real mine field.

Figure 4-20: Survey robot

Soil sampling can be labor-intensive, time consuming and even dangerous for humans. For example, soil strength sampling requires human labor in collecting samples and doing tests on-site in an unstructured environment or in a laboratory [121]. One important goal of this research is to determine the resistance strength of the soil sub-grade. The California Bearing Ratio (CBR) test is a way of quantifying the soil strength factor. A soil's CBR value is an index of its resistance to shearing under a standard load compared to the shearing resistance of a standard material (crushed limestone) subjected to the same load. The CBR is the basis for determining the thickness of soil and aggregate layers used in the design of roads and airfields in the theater of operations [122]. The bearing capacity of a soil is its ability to support loads that may be applied to it by an engineering structure, such as a building, a pavement on a highway, or a runway in an airport and the moving loads that may be carried thereon. A soil with insufficient bearing capacity to support the loads applied on it would fail by shear, resulting in the structure moving or sinking into the ground. Bearing capacity is directly related to the allowable load that may be safely placed on a soil [122].

The CBR test is a simple penetration test developed to evaluate the strength of soil subgrades. The CBR test is standardized so we are able to rank soil strengths according to their CBR values: the stronger the subgrade, the higher the CBR reading; conversely, the softer the sub-grade, the lower the CBR reading. The CBR test consists of causing a plunger of a standard penetrometer to penetrate into a soil sample. The CBR test can be done in the laboratory, or in the field. Although it is most appropriate for fine-grained soils, CBR can also be used to characterize aggregates for road base applications [123].

The Global Positioning System (GPS) navigation gives the robot adaptive navigation ability. That is, starting from any point with any initial orientation, the robot can navigate to the targeted point. Even if the robot motion system is not very accurate, its motion errors can be compensated by continuous adjustments from GPS guidance. The proposed navigation controller allows the robot to interact with the environment much more accurately, much faster, and much more reliably and allows the robot to exhibit complex behaviors while taking into account multiple goals. For example, avoid nearby obstacles while performing long-distance navigation, or navigation to a long distance targeting point.

Figure 4-21 shows different views of the soil sampling robot. The soil sampling core is contained in an aluminum tube, which is mounted onto the bottom of the robot. The tube functions as sample core protector and force withstander. When the linear actuator pushes the penetrometer into the ground, the linear actuator force is transferred to the aluminum bottom of the robot.

The penetrometer is connected with the load cell, and the load cell with the actuator. The actuator, when it functions, pushes the penetrometer into the ground at a constant speed. Therefore we can record the soil penetration force as a function of penetration depth.

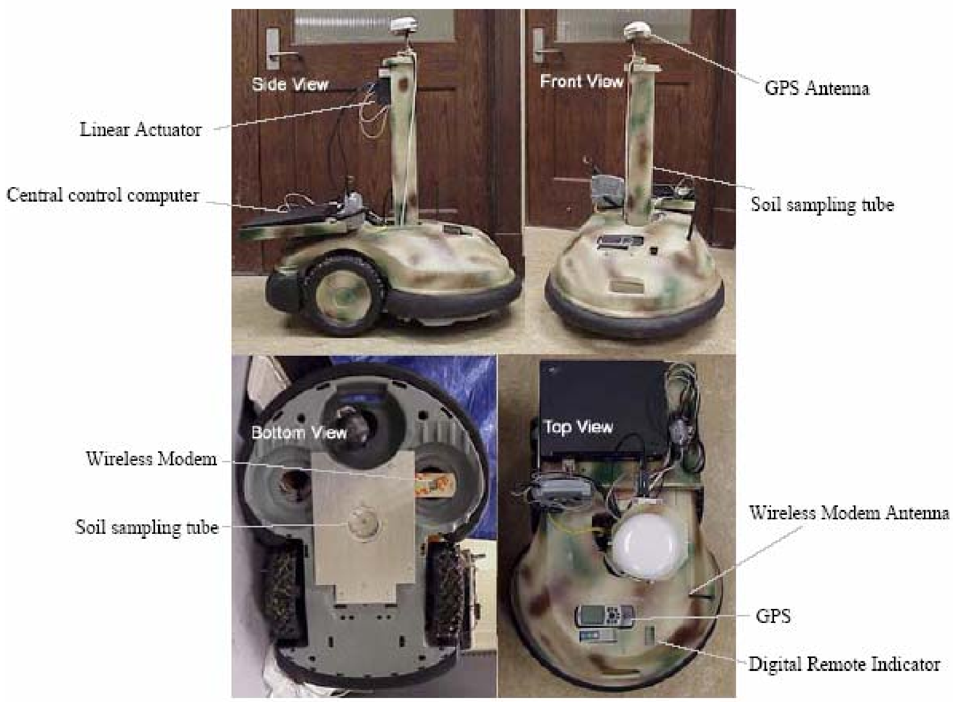


Figure 4-21: Survey robot structure [122]

4.20 Robot application case 3: snow accumulation prevention robot

Snow is a major problem in many cities across the United States. Snow cleaning is a tedious and time/labor consuming task. When there is significant amount of snow, let’s say more than 2”, it is economical to utilize heavy snow cleaning equipment and remove the snow. Dealing with a small amount of snow is a different challenge. When people wake up and see 1” of snow in their driveway they wish they had a way to prevent the snow from accumulating in the first place.

Masoud Ghaffari along with Jay Lee and Mark McCrate designed and patented a snow prevention accumulation robot to answer such need. The can be preprogrammed and, with no human intervention, it can clear a predetermined area with its snow blower and prevent further accumulation with its salt spreader. The design considers the following criteria:

- The work space is limited to a perimeter; a driveway or a parking lot
- The robot should scan the whole area
- An economical and affordable solution is desired
- The amount of accumulated snow is less than 1”
- Snow prevention and cleaning, and not snow shoveling, is desired

Figure 4-22 shows the robot cleaning a driveway.

Figure 4-22: Snow accumulation prevention robot

Chapter 5 : Perception Modeling

"If the human brain were so simple we could understand it, we would be so simple we couldn't."
Lyall Watson (1939-)

According to Wikipedia[1] "perception is one of the oldest fields within scientific psychology, and there are correspondingly many theories about its underlying processes. The oldest quantitative law in psychology is the Weber-Fechner law, which quantifies the relationship between the intensity of physical stimuli and their perceptual effects. It was the study of perception that gave rise to the Gestalt school of psychology, with its emphasis on holistic approaches."

---

[1] www.wikipedia.org

5.1 Approach I: Natural language perception-based control

Information which is conveyed by propositions drawn from a natural language will be said to be perception-based [53]. Natural language perception-based control (NLPC) can be defined as "*perceiving information about the dynamic environment by interpreting the natural language and reacting accordingly*".

In the NLPC, perceptions are not dealt with directly. Instead, NLPC deals with the descriptions of perceptions expressed in the natural language. Therefore, propositions in a natural language play the role of surrogates of perceptions. In this way, manipulation of perceptions is reduced to a familiar process, manipulation of propositions expressed in a natural language [43].

Table 5-1: Comparison of measurement and perception-based information

Information	Data	Example
Measurement-based	Numerical	There is a obstacle 20.2 feet away
Perception-based	Linguistic	There is a ramp in front

The problem is how to compute on perceptions and use it for robot control. To be realistic, the proposed model applies some assumptions to restrict the scope of project.

- Application of computing theory of perceptions is limited to the robot control.
- The robot's operating environment is limited to what has been defined for the international ground vehicle competition (IGVC) navigation course. It is a semi-structured environment with lines and obstacles (see Figure 5-1).
- Natural language processing is limited to simple propositions related to the robot navigation.

- The robot works on the semi-supervised mode by receiving the feedback from the environment.
- Less precision, which is an intrinsic part of perception, in exchange to the lower cost and complexity of sensory system, is accepted.

The robot control implementation has two phases. The first phase is the instructional mode. The commands were given to the robot and robot followed the instructions based on what operator perceives. Table 5-2 shows some examples of these commands.

Table 5-2: Example of instructional control commands

What	**Where**	**How much**
MOVE	LEFT	A LITTLE
MOVE	FORWARD	UNTIL SEE OBSTACLE/LINE DISAPEAR
GO TO	OBSTACLE	UNTIL VERY CLOSE/CLOSE
FOLLOW	KNOWN ROUTE	UNTIL it is DEFFERNT
LOOK	From RIGHT CAMERA	UNTIL SEE A LINE/OBSTACLE
CONTINUE	In AUTOMODE	
STOP	HERE	WITHIN 1 FOOT

The second mode is the declarative mode. In this mode the environment was described to the robot with simple propositions. The robot should make its movement decisions based on what is described to it. Propositions are limited to what is expected in the international ground vehicle competition course. Table 5-3 gives an idea about some of these propositions.

Table 5-3: Examples of propositions in the declarative mode

Proposition	**Possible action**
There is an obstacle in the front left	Move a little bit to right
Left line is disappearing	Switch to the right camera for line following
The obstacle is very close in front	Make a big turn
There is a obstacle in front close to the left line	Turn to right and then left

5.1.1 Model context

The international ground vehicle competition course was used as a test-bed. In the navigation challenge of this contest there are white lines to follow, which sometimes disappear, and barrels to avoid. This course declared to the system by an operator and the robot was supposed to navigate through the path. Figure 5-1 shows the competition course. Figure 5-2 is the picture of an unmanned ground vehicle demonstrated in the 11th IGVC in Detroit and taken by the author.

Figure 5-1: IGVC competition course

Figure 5-2: An UGV by General Dynamics

5.1.2 Perception modeling and creative control

Sherry Liao et al. developed a creative control model as shown in Figure 5-3 [6, 124]. The architecture is proposed according to the creative learning theory [125]. In this proposed diagram, there are three important components: task control center, criteria (critic) knowledge database, and learning system. Adaptive critic learning method is a part of the creative learning algorithm. However, creative learning with decision-making capabilities is beyond the adaptive critic learning. The most important characteristics of the creative learning structure are: (1) Brain-like decision-making task control center, entails the capability of human brain decision-making; (2) Dynamic criteria database integrated into the critic-action framework, makes the adaptive critic controller reconfigurable and enables the flexibility of the network framework; (3) Multiple criteria, multi-layered structure; (4) Modeled and forecasted critic modules result in faster training network.

It is assumed that we can use a kinematic model of a mobile robot to provide a simulated experience to construct a value function in the critic network and to design a kinematic based controller for the action network. Furthermore, the kinematic and dynamic models may also be used to construct a model-based action in the framework of the adaptive critic-action approach. In this algorithm, we build a criteria (critic) database to generalize the critic network and its training process. It is especially critical when the operation of mobile robots is in an unstructured environment. Another component in the diagram is the utility function for a tracking problem (error measurement). A creative controller is designed to integrate the domain knowledge and task control center into the adaptive critic controller. It needs to be a well-defined structure such as in the autonomous mobile robot application as the test-bed for the creative controller.

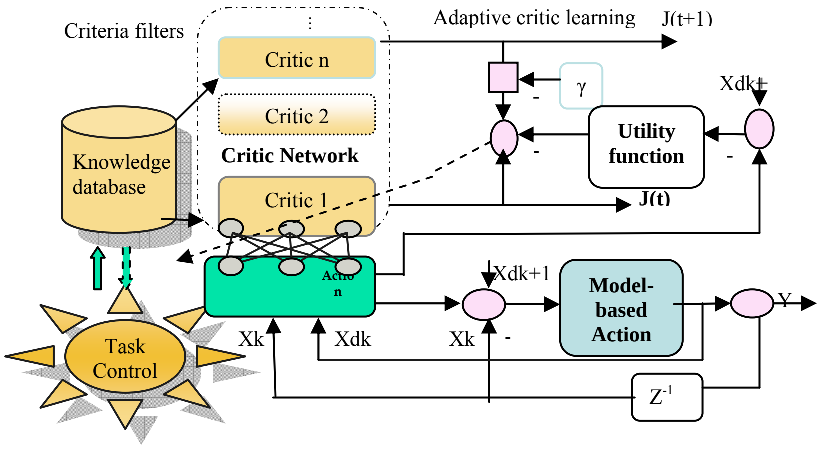


Figure 5-3: Proposed CL Algorithm Architecture [6]

5.1.3 Adaptive critic control

Adaptive critic (AC) control theory is a component of creative learning theory. Werbos summarized recent accomplishments in neurocontrol as a "brain-like" intelligent system. It should contain at least three major general-purpose adaptive components: (1) an Action or Motor system, (2) an "Emotional" or "Evaluation" system or "Critic" and (3) an "Expectations" or "System Identification" component [126].

"Critic" serves as a model of the external environment to be controlled; solving an optimal control problem over time may be classified as adaptive critic designs (ACD). ACD is a large family of designs which learn to perform utility maximization over time. In dynamic programming, normally the user provides the function $U(\underline{X}(t), \underline{u}(t))$, an interest rate r**,** and a stochastic model. Then the analyst tries to solve for another function $J(\underline{X}(t))$, so as to satisfy some form of the Bellman equation shown in Eq. (5-1) that underlies dynamic programming [3]:

$$J(R(t)) = \max_{u(t)}(U(R(t),u(t)) + < J(R(t+1)) >)/(1+r)$$ Eq. (5-1)

where "<>" denotes expected value.

In principle, any problem in decision or control theory can be classified as an optimization problem. Many ACDs solve the problem by approximating the function J. The most popular methods to estimate J in ACDs are heuristic dynamic programming (HDP), Dual Heuristic Programming (DHP) and Globalized DHP (GDHP) [126, 127]. HDP and its ACD form have a critic network that estimates the function J (cost-to-go or strategic utility function) in the Bellman equation of dynamic programming, presented as follows:

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k)$$ Eq. (5-2)

Where γ is a discount factor (0<γ<1), and U(.) is the utility function or local cost. An alternative approach referred to as Dual Heuristic Programming (DHP) has been proposed. Here, the critic network approximates the derivatives of the future cost with respect to the state variable. It has been proved that DHP is capable of generating smoother derivatives and has shown improved performance when compared to HDP [128, 129]. Werbos first proposed the idea of how to do GDHP[130]. Training the critic network in GDHP utilizes an error measure which is a combination of the error measures of HDP and DHP.

5.1.4 Task Control Center (TCC)

The task control center (TCC) can build task-level control systems for the creative learning system as shown in Figure 5-4 [125]. By "task-level", we mean the integration and coordination of perception, planning and real-time control to achieve a given set of goals (tasks). TCC provides a general task control framework, and it is intended to be used to control a wide variety of tasks and permits responsive actions based on mission commands, on interactions with other robots. Although TCC has no built-in control functions for particular tasks (such as robot path planning algorithms), it provides control functions, such as task decomposition, monitoring, and resource management, that are common to many applications. The particular task built-in rules in the dynamic database matches the constraints on particular control schemes or sub-tasks or environment allocated by TCC. The task control center acts as a decision-making system. It integrates domain knowledge or criteria into the database of the adaptive learning system. Task control architecture for mobile robots provides a variety of control constructs that are commonly needed in mobile robot applications, and other autonomous mobile systems. Integrating TCC with adaptive critic learning system and interacting with the dynamic database, the creative learning system could provide both task-level and real-time control or learning within a single architectural framework.

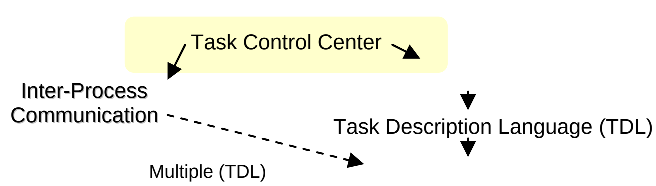


Figure 5-4: The structure of task control center

5.1.5 Perception-based task control center

In the presented model the TCC acts as the highest level of decision making. It processes perceptions and the sensory information. This is the area that the human brain shows its extraordinary abilities.

The problem can be explained as follow. There are a collection of propositions expressed in the natural language about the robot environment and the goals. These propositions could come from an online operator or in the fully autonomous mode they could be stored in the robot or sensed and perceived from sensory information. From these propositions we wish to infer proper tasks for the TCC. In the perception-based module the answers could be in the form of natural language.

The perception-based task control center should follow a human-like logic. Consider an example suggested by Kubota et al. about how human drive [4]. In driving a car when the road is wide and without obstacle, drivers pay attention to the far area ahead and speed up. However, in a crowded situation they pay attention to near area, and should slow down. In this case the relationship between the attention range and speed has been changed dynamically based on the facing environment state. It shows how human brain approaches the problem differently.

A similar approach was taken by Stanford team who won the 2005 DARPA Grand Challenge. "*After the competition, Thrun reflected that one of the key advantages of his Stanford team's Stanley robot, which won the race and the $2 million, was its vision-based speed switch. Stanley uses a simple but powerful form of machine learning to hit the gas whenever it spots a smooth road extending into the distance*" [131].

Figure 5-5 shows the flow of information in the perception-based task control center. First, sensory data from environment will be collected and processed to information. In the IGVC context and Bearcat Cub robot there are GPS, laser, digital cameras, and stereo vision cameras that collect information from the environment. Then collected information will be perceived.

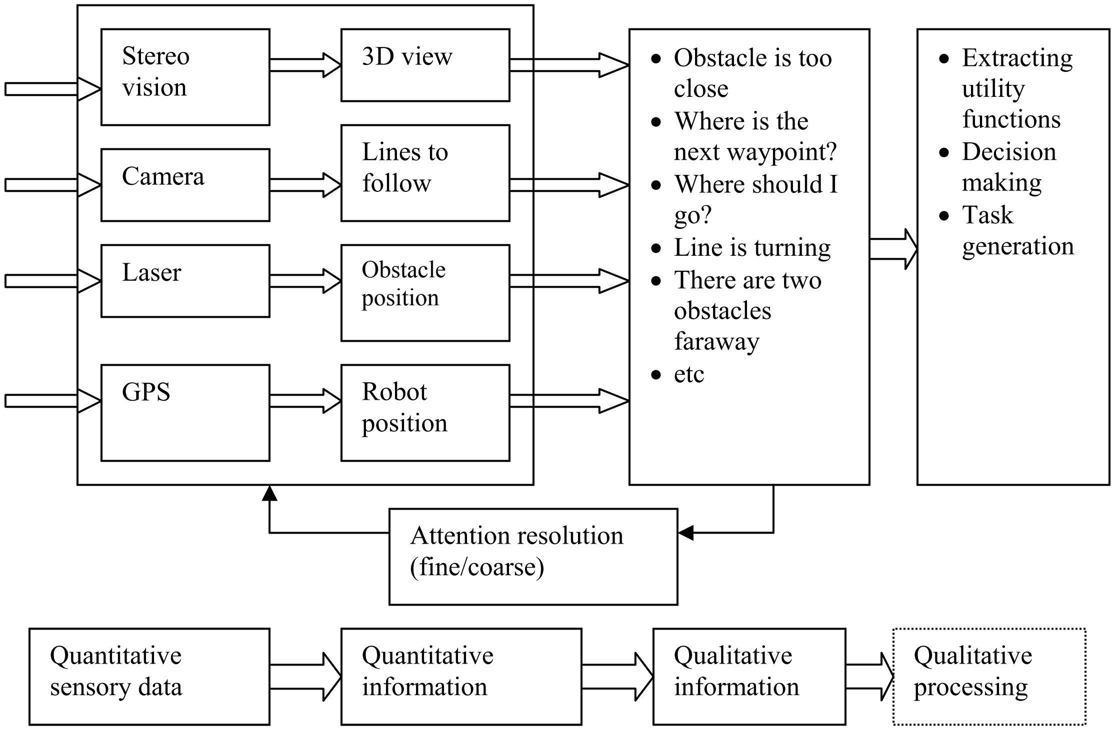


Figure 5-5: Flow of information in perception system

A human-like perception is the one that differentiates between coarse and fine perceptions. In the coarse perception, a broader view is the subject of attention. In an analogy with Bellman equations, coarse perception is an estimation of the function J (cost-to-go or strategic utility function). The fine perception is when the system tries to solve a local problem. For example when there is an immediate obstacle in front of the robot, the main priority or subject of attention is avoiding that obstacle. That is analogous to the local utility function or U in Bellman equation. Table 5-4 represents examples of sensing to action process.

Table 5-4: Example of sensing to action process

Coarse Sensing	Coarse Perception	Fine Sensing	Fine Perception	Task Selection	Criteria	Action – Iteration
Look around	Where am I	Zoom in on certain areas	What is that	What can I do	How well can I do it	Do it
	Navigation challenge start	Load waypoints	Navigation challenge	Go to next waypoint	Error of location	Go to waypoint
	Autonomous challenge start	Look for lines	Find lines and obstacles	Go down path and avoid obstacles	Track error and ticket error	Go along path

5.1.6 Implementation methodologies

Among soft computing methodologies, which were explained in chapter 3, fuzzy theory is a preferred choice to implement the proposed perception framework. Fuzzy theory provides linguistic representation such as 'close', 'far', and 'safe'. Fuzzy system implements mapping from its input space to output space by a number of fuzzy if-then rules. In the autonomous navigation problem, fuzzy system enables to deal with incompleteness and uncertainty of information which comes from the environment [132].

Figure 5-6 shows a basic fuzzy approach for the task control center. The input of this system is sensory data collected from the environment. The output is suggested tasks that will be entered to the dynamic knowledge database module.

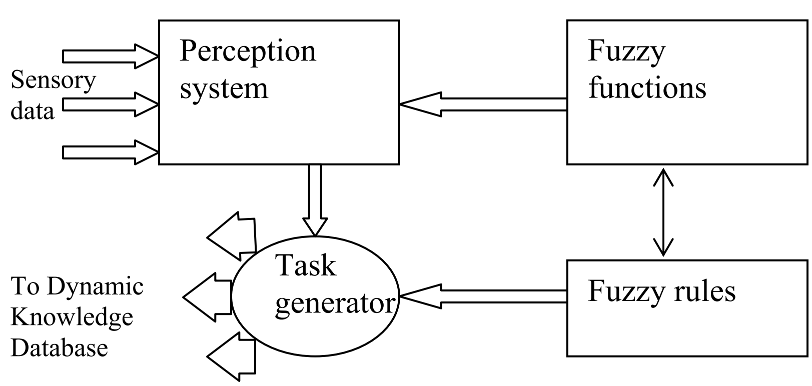


Figure 5-6: A basic fuzzy model

Fuzzy rules will be used to generate tasks. In a basic fuzzy inference system the input variables are perceptual information and the output is the desired task. When *n* is the number of perceptual information, *r* the number of fuzzy rules, and *s* that of output tasks, a general inference rule can be written as:

if $\boldsymbol{p_1}$ is $\boldsymbol{A_{l,1}}$ and $\boldsymbol{p_2}$ is $\boldsymbol{A_{l,2}}$ … and $\boldsymbol{p_n}$ is $\boldsymbol{A_{l,n}}$ ***then*** $\boldsymbol{y_1}$ is $\boldsymbol{w_{l,1}}$ and $\boldsymbol{y_2}$ is $\boldsymbol{w_{l,2}}$ … $\boldsymbol{y_s}$ is $\boldsymbol{w_{l,s}}$

where $\boldsymbol{p_i}$ is perceptual input information and $\boldsymbol{y_j}$ is task output (i=1,2,…,n. ; j=1,2,…,s) and $\boldsymbol{A_{l,i}}$ is the membership function (l=1,2,…,r).

5.1.6.1 Neuro-Fuzzy methodology

Fuzzy systems are used when expert knowledge about the process is available, while artificial neural networks are useful when enough process data are available or measurable. Both systems are capable of dealing with non-linear problems. The main difference is that neural systems are treated in a numeric quantitative manner, whereas fuzzy systems are capable of symbolic qualitative processing.

Another perception-based implementation approach for TCC is neuro-fuzzy methodology. Neuro-fuzzy combines the features of both fuzzy theory and artificial neural networks. Hayashi and Buckley proved that 1) any rule-based fuzzy system may be approximated by a neural net and 2) any neural net (feedforward, multilayered) may be approximated by a rule-based fuzzy system [133]. This kind of equivalence between fuzzy rule-based systems and neural networks is also studied by others [133-135].

Neuro-fuzzy computing enables one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in data-rich environments into the system.

The modeling of imprecise and qualitative knowledge as well as the transmission of uncertainty are possible through the use of fuzzy logic.

Neuro-fuzzy hybridization is done broadly in two ways: a neural network equipped with the capability of handling fuzzy information (*fuzzy-neural network* (FNN)) and a fuzzy system augmented by neural networks to enhance some of its characteristics like flexibility, speed, and adaptability (*neural-fuzzy system* (NFS)) [136, 137, 135]. For the TCS model, a neural-fuzzy system is proposed.

A neural-fuzzy system (NFS) is designed to realize the process of fuzzy reasoning, where the connection weights of the network correspond to the parameters of fuzzy reasoning [138, 135]. Using the backpropagation-type learning algorithms, the NFS can identify fuzzy rules and learn membership functions of the fuzzy reasoning. To goal is to design neural networks guided by fuzzy logic formalism to implement fuzzy logic and fuzzy decision-making, and to realize membership functions representing fuzzy sets. A neuro-fuzzy system is shown in Figure 5-7.

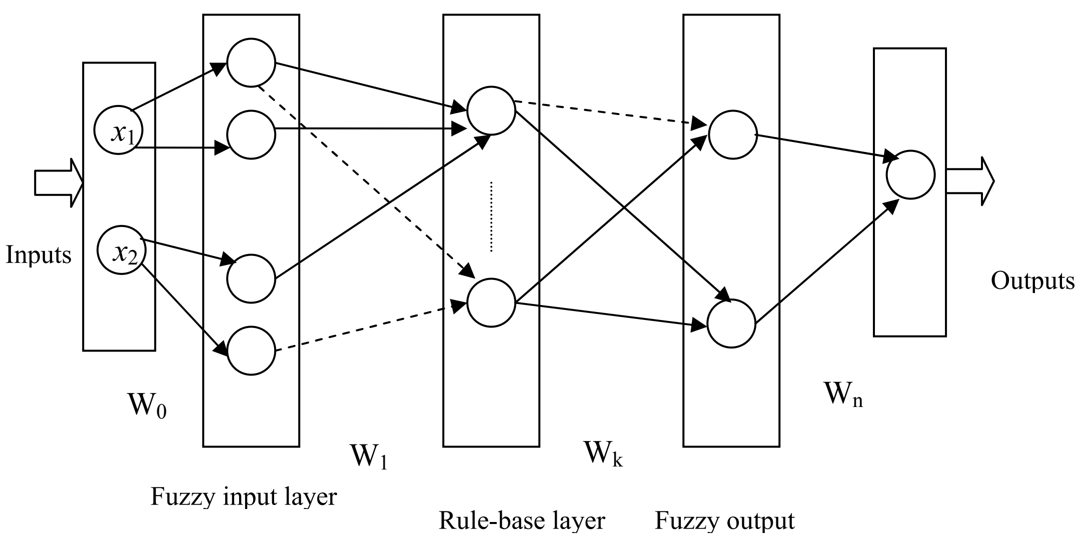


Figure 5-7: Structure of a neuro-fuzzy system

5.2 Approach II: estimation-based perception modeling

In robot navigation building an exact world model in a dynamic environment is not feasible or practical in many situations. Estimation theory provides foundations and tools to deal with *uncertainty* that is inherent of such models. Estimation theory has been known and studied in control theory for a while. Weiner, who is a pioneer in the area of Cybernetics, and Kolmogoro, with his familiar statistical test, independently developed the foundation of estimation theory [139, 140]. Kalman filter that was introduced in 1960 was a major breakthrough in navigation theory [141, 142]. Kalman introduced a recursive digital algorithm for integrating navigation sensor data to achieve optimal overall system performance.

Application of estimation theory in mobile robot modeling has been inspired by several papers by Smith [143], Durrant-Whyte [144], and Faugeras [145]. Brooks and Chatila also published ad-hoc techniques for manipulation of uncertainty [146, 147]. Crowley has applied estimation theory to model uncertainty of sensor data and perception for mobile robots [148-150]. A more theoretical and detailed approach toward estimation theory and it application in tracking and navigation has been introduced by Bar-Shalom et al. [151].

Estimation theory and Bayesian approach have been applied more extensively in vision than dynamic world modeling.

The perception modeling is based on some assumptions.

1. Perceptions are described by propositions.
2. A proposition is a carrier of information.
3. The meaning of a proposition, P, is represented as a generalized constraint which defines the information conveyed by P.

4. A proposition, P, include a set of properties and associations based on spatial positions.
5. Each proposition comes with a confidence factor.

The model presented in Figure 5-8 includes several steps. In the first step information from surrounding environment will be perceived. Sensory information should be converted to a language that lends itself to the estimation theory operations. The *uncertainty* is an inherent part of the robot dynamic environment and also subject of estimation theory. Therefore, each *proposition*, that conveys perceived information, would have a *confidence value* that represents the uncertainty.

The next step is to estimate navigation variables and parameters based on perceived information. These values will be compared by previous information and will be fed to the robot model in 'matching and modeling' stage. Finally, navigation decisions will be made and process will be repeated and updated.

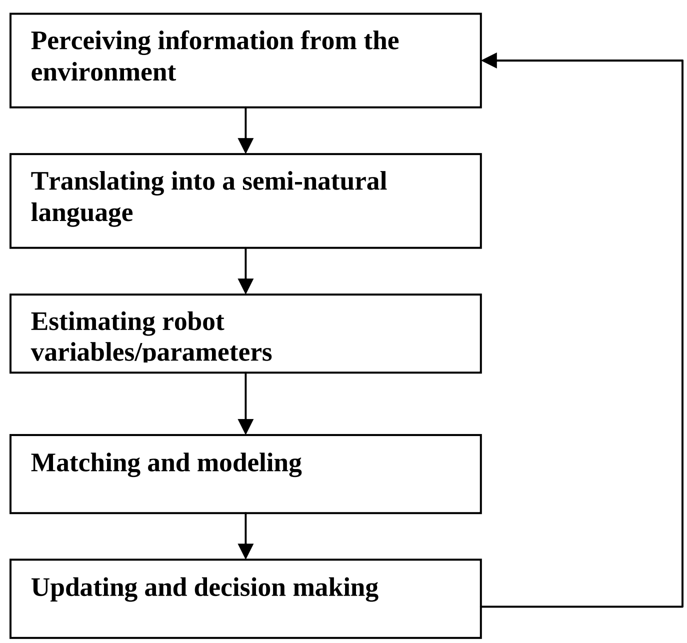


Figure 5-8: Estimation-based model

When a proposition is converted to a numerical form, there are well defined techniques to investigate the model further [149].

In this case the Kalman filter prediction equations provides the means for predicting the state of the model, the Mahalanobis Distance provides a simple measure for matching, and the Kalman filter update equations provide the mechanism to update the property estimates in the model.

A dynamic world model, $M(t)$, is a list of propositions which describe the "state" of a part of the world at an instant in time t.

$$\text{Model: } M(t) \equiv \{P_1(t), P_2(t), \ldots, P_m(t)\} \qquad \text{Eq. (5-3)}$$

Each $P_i(t)$ describes a part of the world model and includes conjunction of estimated properties, $\hat{X}_i(t)$, and a confidence factor $CF_i(t)$.

$$\text{Proposition: } P_i(t) \equiv \{\hat{X}_i(t), CF_i(t)\} \qquad \text{Eq. (5-4)}$$

Newer observations have a lower confidence factor. When an observation confirms its previous ones, a higher confidence factor will be assigned to that particular segment. If no observation of the segments occurs in a few cycles, it will be considered as noise and will be removed from the model. When an observation becomes confident, it will be kept in the model for several cycles, even if it is missed from observations. The number of cycle depends on the system application.

Consider the IGVC robot example mentioned in the previous section. Detection of a line is an observation with a confidence factor. When observations of points on the line are repeated, a higher confidence factor will be assigned to the line detection proposition. Those observations will remain in the model even if they are not repeated for several cycles. In this case, the robot may have moved further and the line is no longer in its field of view. Such information will remain in the model as long as they are relevant and useful for navigation. Observation of random bright points – in IGVC example – which will not be repeated will be removed for the model as noise.

A proposition can represent an estimate of a subsystem, or a part of the world, with association among N properties of a vector, $\hat{X}(t)$.

$$\hat{X}(t) \equiv \{\hat{x}_1(t), \hat{x}_2(t), \ldots, \hat{x}_n(t)\} \qquad \text{Eq. (5-5)}$$

The actual state of the world, $X(t)$, in unknown but it is estimated by an observation process and observation vector of $Y(t)$. The observation process comes with a random noise, $N(t)$[149].

$$Y(t) = X(t) + N(t) \qquad \text{Eq. (5-6)}$$

$X(t)$ is not known but its estimate, $\hat{X}(t)$, can be calculated from observations. At each cycle, combination of a predicted observation $\hat{Y}(t)$ and an actual observation $Y(t)$ will provide an estimate value, $\hat{X}(t)$. The difference between $Y(t)$ and $\hat{Y}(t)$ will be used to update the estimate $\hat{X}(t)$ in the following way.

Estimates of uncertainty for $\hat{X}(t)$ and $Y(t)$ are needed for this process. The uncertainty can be seen as deviation between the estimated and actual vectors of the world, $\hat{X}(t)$ and $X(t)$. The expected value of deviation is approximated by a covariance matrix $\hat{C}(t)$ which represents the square of the expected difference between the estimate and the actual world state.

$$\hat{C}(t) \equiv E\left\{\left[X(t) - \hat{X}(t)\right]\left[X(t) - \hat{X}(t)\right]^T\right\} \qquad \text{Eq. (5-7)}$$

The uncertainty estimate provides two crucial roles:

1) It provides the tolerance bounds for matching observations to predictions, and

2) It provides the relative strength of prediction and observation when calculating a new estimate.

Because $\hat{C}(t)$ determines the tolerance for matching, system performance will degrade rapidly if we under-estimate $\hat{C}(t)$. On the other hand, overestimating $\hat{C}(t)$ may increase the computing time for finding a match [149].

In the next phase of modeling, the value for $X^*(t+\Delta t)$ will be predicted based on the estimated vector $\hat{X}(t)$. That corresponds to calculation of predicted uncertainty, $C^*(t+\Delta t)$, based on estimated uncertainty $\hat{C}(t)$. Temporal derivatives of the $\hat{X}(t)$ properties and covariance's between the properties and their derivatives will be used for such prediction. The estimated derivatives can be considered as properties of the vector $\hat{X}(t)$.

5.2.1 The first order prediction

In the first order prediction only the first temporal derivative is estimated. For the higher number of properties and the higher order of derivatives the same procedure can apply. Also in this case the time variable *t* is continuous and the time interval, ΔT, can vary.

The derivatives can be added to the proposition vector $X(t)$. Therefore, if there are N properties in $X(t)$, the vector will include 2N elements that are N properties and N first derivatives. However, the observation vector $Y(t)$ includes only N elements.

To predict the next value, $x^*(t+\Delta T)$, of the property $\hat{x}(t)$ of the vector $\hat{X}(t)$, an estimation of a first order temporal derivative, $\hat{x}'(t)$, is needed.

$$\hat{x}'(t) = \frac{\partial \hat{x}(t)}{\partial t}$$

A Taylor series can be used to predict the change in $X(t)$. In the case of first order prediction, all higher order terms are represented by the random vector $V(t)$, approximated by its estimate $\hat{V}(t)$. The mean for $V(t)$ is assumed to be zero, in most cases, and its variance is represented by $Q(t)$.

$$Q(t) = E\{V(t)V(t)^T\} \qquad \text{Eq. (5-8)}$$

Therefore, the prediction of a property can be summarized as:

$$x^*(t+\Delta T) = \hat{x}(t) + \frac{\partial \hat{x}(t)}{\partial t}\Delta T + \hat{V}(t) \qquad \text{Eq. (5-9)}$$

Consider the case that there are two properties $x_1(t)$ and $x_2(t)$ for the proposition $\hat{X}(t)$

.

$$\hat{X}(t) = \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_1'\ (t) \\ \hat{x}_2(t) \\ \hat{x}_2'(t) \end{bmatrix}$$

In the matrix form the prediction can be written as:

$$X^*(t+\Delta T) := \varphi \hat{X}(t) + V(t)$$

Where φ is:

$$\varphi = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With the prediction of $X^*(t+\Delta T)$ there is an uncertainty that can be calculated based on the covariance between each property, $\hat{x}(t)$, and its derivative. That uncertainty, $\hat{Q}(x)$, can model the effect of other derivatives. The second prediction equation is [149]:

$$C_x^*(t+\Delta T) := \varphi^T \hat{C}_x(t)\varphi + \hat{Q}_x(t) \qquad \text{Eq. (5-10)}$$

5.3 Approach III: spatial knowledge modeling for autonomous challenge

5.3.1 Spatial knowledge

Humans use spatial relationships to describe their environment and to navigate, for example, a pothole or to veer around a desk and pass through a doorway. Recent cognitive models suggest that people use these types of spatial knowledge to perform many daily tasks. They also emphasize in importance of spatial knowledge and how it develops [18, 152, 153].

Spatial cognition includes acquisition, organization, use, and revision of knowledge about spatial environments [154]. Natural language descriptions of spatial situations can be viewed as the linguistic image of mental/internal representations of these situations. In particular, this concerns the partial correspondence between the spatial inventory of natural language and the 'cognitive ontology' of space. In this framework, the following problem areas require attention (among others): Which cognitive entities can we assume to exist in the system of natural language (dimensionality, shape, orientation, etc.)?

The spatial cognition priority program is particularly oriented towards cognitively oriented sub areas of computer science / artificial intelligence, psychology, linguistics, anthropology, and philosophy which are concerned with complex behavior in dealing with physical space.

Different forms and representations of spatial information can be identified in systems navigating in complex surroundings. One of the most common distinctions in spatial navigation research concerns the difference between landmark, route, and survey knowledge of an environment [154]. In human navigation three distinctive terms should be defined, landmarks, routes and survey knowledge.

Landmark: A landmark is a unique object at fixed location. It could be a visual object, odor, sound, or a tactile percept. A landmark is a decision making point. It could be a confirmation for continuing the previous pattern and decision or it could result to a new decision.

Route: A route corresponds to a sequence of objects or events as experienced during navigation (e.g. tunnels, trails, roads, corridors). Sequences can either be continuous or discrete. Examples of objects are pictures and movements, and examples of events are decisions like left or right turns.

Survey knowledge: Survey knowledge is a navigation environment model that contains routes and landmarks. A map is an example of the survey knowledge.

Information in route knowledge is accessed sequentially as an ordered list of locations. Survey knowledge in the other hand is considered as an integrated model of navigation environment. It enables the inference of spatial relationship between the arbitrary pairs of locations. In a set theory approach landmark, route, and survey knowledge can be related with a subset relationship as shown in Eq. (5-10).

$$landmark \subset route \subset surveyknowledge \qquad \text{Eq. (5-10)}$$

Route knowledge can be acquired in different ways. Exposure to a route can lead to a series of connections. This route knowledge can be used in similar situations. For example driving in a US city downtown may familiarize a driver with a pattern that can be used in similar situations.

The study of route and survey knowledge has received a great deal of attention in spatial cognition research [154].

5.3.2 Implementation of spatial knowledge model

The University of Cincinnati robot team has designed and constructed a robot, the Bearcat Cub as shown in Figure 5-9, for the Intelligent Ground Vehicle Competition, the DARPA Grand Challenge, and many other potential applications. The Bearcat Cub is an intelligent, autonomous ground vehicle that provides a test-bed system for conducting research on mobile vehicles, sensor systems and intelligent control.

The Bearcat Cub was used as a test bed to implement the human-like spatial knowledge model in a robot. The robot has two cameras for line following, a laser scanner and a stereo vision system for obstacle detection and spatial modeling, a Global

Positioning System (GPS) for navigation. It has different modes of run including manual control, autonomous challenge that includes line following and obstacle avoidance, voice control, and GPS navigation. It also utilizes a hybrid power system.

A model similar to what was explained in the human navigation modeling was implemented in the bearcat cub robot. A 200 meter long, 8 meter wide *route* was marked by flags. The route had some sharp turns and some obstacles were placed randomly. Five points were marked by GPS as *landmarks*. The robot was supposed to stay in the route and reach the landmarks.

Several tests were conducted and the robot finished the course successfully. Since the GPS accuracy was limited to 10 feet, in some runs the robot reached to a certain distance of the *landmarks.* To offset the error, the results of each run were used to update the GPS coordinates of the landmarks. This corresponds to the idea of *survey knowledge* in the human navigation model.

A laser scanner was used to detect the obstacles. A remotely controlled toy car was driven in the robot route to create a moving obstacle. The robot was able to avoid the stationary and moving obstacles successfully.

Humans drive and navigate differently and there is no unique path. However, the idea of smooth driving and avoiding sudden movements is common in human navigation. A similar approach was used in the robot navigation. One example of such approach is shown in Figure 5-10.

The robot detects a path that could be a line, a wall or any other indicator of a need for changing the direction. The bearcat cub has two powered wheels and a caster wheel as shown in Figure. 5-9. To change the direction, the wheels should move with different velocities. Equation 5-11 represents the speed of each wheel for a robot with the width of *w* and the turn angle of *θ.* The *t* is the interesting part of the equation. It is the expected time for a turn. By its nature *t* is a fuzzy variable. It was used to add a human-like feature to this experiment. A table of expected values of *t* based on human perception for different values of *θ* and velocities was used. This way the robot was able to avoid obstacles, reach *landmarks*, and follow the *route* smoothly.

Figure 5-9: The Bearcat Cub robot

From two points, p_1 and p_2, along the line, the robot detects the orientation of the line with respect to the robot. The appropriate steering angle, θ, is calculated to make the robot parallel to the line. In addition, a certain hugging distance, h, must also be maintained with the midpoint of the two points on the line.

This architecture is influenced by psychological models of human navigation as explained. It consist three levels of *landmark*, *route*, and *survey knowledge*. Humans use this kind of spatial knowledge to navigate

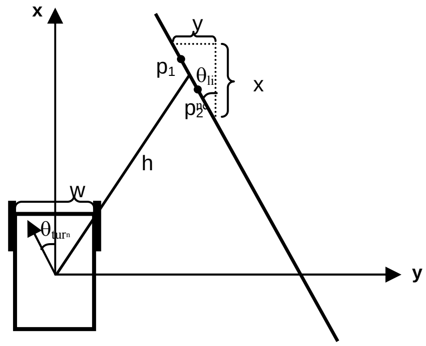


Figure 5-10: Top view of the robot and its steering angle θ

$$V_{left} = V_{center} - (\theta wt/2) \qquad \text{Eq. (5-11)}$$

$$V_{right} = V_{center} + (\theta wt/2)$$

Chapter 6 : Perception Optimization

"Do what you know and perception is converted into character."

Ralph Waldo Emerson (1803 - 1882)

In this section, DARPA Grand Challenge, DARPA Urban Challenge, and Intelligent Ground Vehicle Competition (IGVC) will be used interchangeably as examples for the foundation of perception optimization theory. This theory tries to apply an optimization approach to formulate the design process, as well as navigation algorithm, of a robot using human perception.

6.1 DARPA Urban Challenge problem

The Defense Advanced Research Projects Agency (DARPA) plans to hold its third Grand Challenge competition in 2007, which will feature autonomous ground vehicles executing simulated military supply missions safely and effectively in a mock urban area[2].

There are several missions that each vehicle needs to complete. These are examples according to DARPA:

Mission 1: Complete a mission defined by an ordered series of checkpoints in a complex route network. The vehicle will have 5 minutes to process a mission description before attempting the course.

Mission 2: Interpret static lane markings (e.g., white and yellow lines) provided with the route network definition file and behave in accordance with applicable traffic laws and conventions.

Mission 3: Exhibit context-dependent speed control to ensure safe operation, including adherence to speed limits.

Mission 4: Exhibit safe-following behavior when approaching other vehicles from behind in a traffic lane. This includes maintaining a safe-following distance.

Mission 5: Exhibit safe check-and-go behavior when pulling around a stopped vehicle, pulling out of a parking spot, moving through intersections, and in situations where collision is possible.

Mission 6: Stay on the road and in a legal and appropriate travel lane while en route, around sharp turns, through intersections, and while passing. The route network definition file will specify the GPS coordinates of the stop signs.

[2] http://www.darpa.mil/grandchallenge/index.asp

6.2 Problem formulation

The contest missions can be divided into smaller problems. The question is how to formulate the designer expertise and perception in this process with an optimization approach. The concept of Quality Function Deployment (QFD) will be used for problem formulation.

Quality Function Deployment is a decision-making tool that has been used to collect the voice of expert and human perception in product and service development, brand marketing, and product management. QFD was originally developed by Yoji Akao and Shigeru Mizuno in the 1960s. The first published article was in 1966 by Oshiumi of Bridgestone Tire. In the last 20 years this technique has been embraced by US companies and is being implemented in Six Sigma and ISO procedures [155].

Table 6-1 represents relationship between missions (WHATs) and criteria (HOWs). The symbol in each cell represents perception of affinity between the mission and the criterion. This measure could come from the expert, survey, literature, and so on.

Table 6-1: Criteria matrix

What\How	Importance	Avoid obstacle	Detect line	Minimize distance to waypoint	Smooth move	Keep distance from line
Mission 1	VH	VH	M	L	L	M
Mission 2	H	M	VH	L	L	H
Mission 3	VH	H	VH	VH	M	M
Mission 4	VH	VH	H	VH	M	H
Mission 5	VH	H	VH	H	N/A	H
Relative weight		0.19	0.20	0.17	0.28	0.16

The second column is the importance of each mission. The numbers in this column would be the same if missions are equally weighted. Numbers 1-5 were assigned to the symbols and after considering the importance of each mission, the relative weight of each criterion was calculated as shown in the last row.

To capture human perception in the robot design and formulate the optimization problem, the second iteration of QFD has been shown in Table 6-2. HOWs (i.e. the criteria) in the first table have been converted to WHATs in the second table. Therefore, what is called *mission* in this table is different from that in Table 6-1. To achieve lower-level missions (e.g. avoiding obstacles) different sensory systems are required. Table 6-2 shows the relative importance of each sensor. The last row shows the relative weight of each sensor in the overall design. Fields in Tables 6-1 and 6-2 are just examples of what could be explored in this process.

Table 6-2: Sensor matrix

What\How	Importance	GPS	Laser	Stereo vision	Camera	Compass
Avoid obstacle	VH	L	VH	VH	M	L
Detect line	VH	N/A	N/A	VH	VH	N/A
Minimize distance to waypoint	H	VH	M	M	M	H
Move smoothly	M	L	M	L	L	H
Keep distance from line	VH	N/A	L	H	VH	H
Relative weight		0.11	0.17	0.27	0.26	0.18

More iterations of QFD can be performed to formulate experts' perceptions and clarify different aspects of robot development.

6.3 Perceptual state

Sensors collect information from a robot's environment. Then the robot needs to *decide* the next *action* based on the *perceived* information. This section focuses on perception optimization. Figure 6-1 shows this process. How to act and navigate in an optimal fashion, after the decision has been made, is a problem that has been studied by others [6, 7].

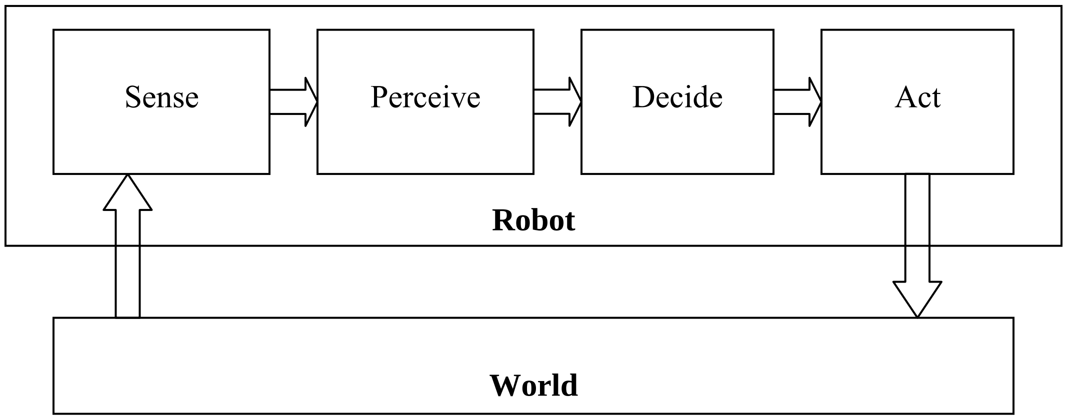


Figure 6-1: World-robot interaction

The subject of interest is that which was called *criteria* in Table 6-1 and *missions* in Table 6-2 (relative to their roles in each phase). This can be defined as *perceptual states* of the *agent*.

Agent: An agent is a system that perceives its environment, acts on it, and pursues its agenda to change what it will sense in the future. In this context, the words *agent* and *robot* will be used interchangeably.

Perceptual state: The state of an agent which represents the criteria of a higher level system. A comparison between Tables 6-1 and 6-2 provides an example for this definition. Criteria such as 'avoid obstacle', 'detect line', and 'move smoothly' are examples of perceptual states of the agent.

6.4 Analogy with human perception

The perceptual state of the agent is quite similar to human state of mind. Consider the daily activities of a student as an example. Studying for next week's exam is a concern- a state of mind- for the first two hours. The student may continue studying (or think about studying) until he/she switches to the next state. If there is a strong reward/punishment for staying in the current state, he/she tends not to change the state (continue studying). If the current state is satisfied (ready for the exam), the student most likely will move to the next state (e.g. chatting online). A good (rational) student is one who tries to maximize his rewards over a period of time. He/she may plan his daily activities accordingly. The good student has a sense of reward, or satisfaction, for different activities over a period of time. Choosing the right sequence of activities (different states of mind), and adjusting those activities according to information collected from the environment, is the way the student can optimize his/her rewards.

Perception modeling and optimization for robot navigation would follow a similar path. Consider the IGVC example illustrated in Figure 6-2.

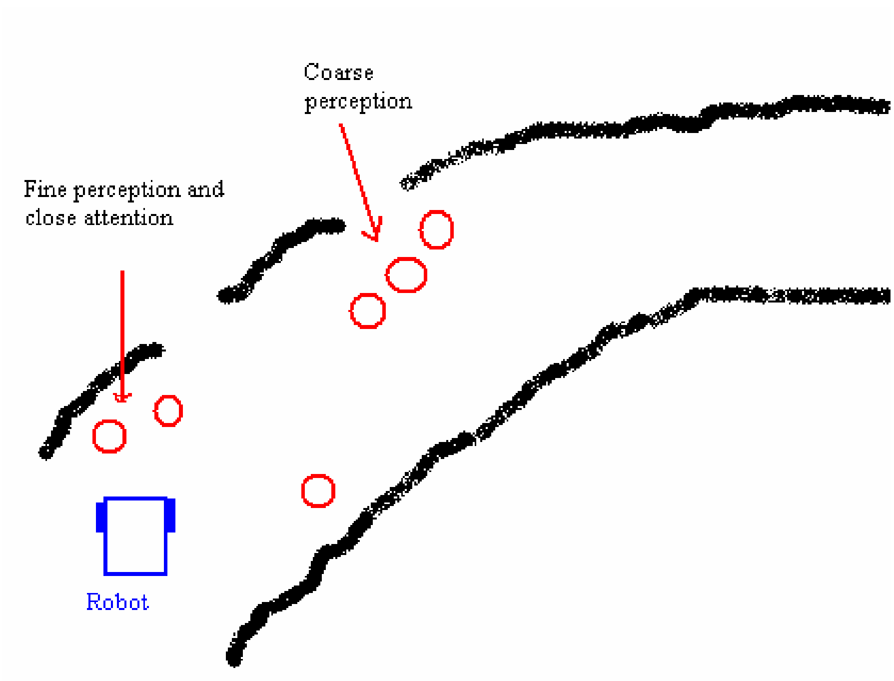


Figure 6-2: The IGVC course

The robot *perceptual state* is avoiding immediate obstacles in front of it. Simultaneously, other perceptual states are in the robot's horizon: 'following lines', 'moving smoothly', or 'moving fast' are other possible states. Based on sensory information, the robot may stay in 'avoiding obstacle' mode or may switch to another state. In each perceptual state a different action may be taken. Sensory information provides the result of each action and how appropriate they have been. Figure 6-3 is an example of this process.

State	Avoid obstacle	Avoid obstacle	Follow line	Go to waypoint
Action	Turn left	Turn left	Turn right slightly	Accelerate
	T1	T2	T3	T4

Time

Figure 6-3: State-action over time

tries to avoid obstacles. Then it perceives it more rewarding to go to the state of 'follow line'. The question is: *how* should the agent switch to different perceptual states to maximize its reward.

6.5 Markov decision process

A Markov decision process (MDP) can model this problem. For simplification, a discrete time model has been chosen. However, it is possible to use continuous models as well [156, 130].

In the discrete time model at each time step t = 0, 1, 2, 3, … the robot decides to update its state and chooses the next action. The following terminology will be used according to Sutton and Barto [157]:

- $s_t \in S$, where s_t is the state at step t and S is the set of possible states

- $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t and a_t is an action
- $r_t \in R$, where r_t is reward received after taking action a_t
- π_t, where $\pi_t(s,a)$ is the probability of $a_t = a$ if $s_t = s$. Deciding which action to take is called *policy*.
- γ, discount factor that reduces the weight of future rewards in compare to current one
- $V^*(s_t)$, the optimum state-value function
- V^π, the state-value function for policy π
- Q^π, the *action-value function for policy* π .

At each time step, a reward coming from sensors will be given to the robot from its environment. This approach is called reinforcement learning. The goal of the robot is to maximize the *total* amount of reward, not the immediate reward, it receives. In a formal way it can be written as:

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \qquad \text{Eq. (6-1)}$$

If the robot is in state s and takes action a, then the *transition probability* of each possible next state, s' , is:

$$P_{ss'}^a = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \qquad \text{Eq. (6-2)}$$

And the expected value of the next reward is:

$$R_{ss'}^{a} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$$ Eq. (6-3)

A policy, π , is a mapping from each state, $s \in S$, and action, $a \in A(s)$, to the probability $\pi(s,a)$ of taking action a when in state s. The *value* of a state s under a policy π , denoted $V^{\pi}(s)$, is the expected return when starting in s and following π thereafter. $V^{\pi}(s)$ can be defined as:

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\}$$ Eq. (6-4)

The value of taking action a in state s under a policy π , denoted $Q^{\pi}(s,a)$, as the expected return starting from s, taking action a, and thus following policy π :

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\}$$

A policy π is superior than policy π' if its expected return is greater than or equal to that of π'. That means $\pi \geq \pi'$, if and only if $V^{\pi}(s) \geq V^{\pi'}(s)$ for all $s \in S$. The optimum policy is the one that is better than or equal to all other policies. All the optimum policies can be shown by π^*. They have the same state-value function V^*.

$$V^*(s) = \max_{\pi} V^{\pi}(s) \text{ , for all } s \in S$$

The optimal action-value function, Q^*, is also the same for all optimum policies.

$$Q^*(s,a) = \max_{\pi} V^{\pi}(s,a) \text{, for all } s \in S \text{ and } a \in A(s)$$

This function gives the expected return when the robot takes action a in state s following an optimal policy. It can be rewritten as:

$$Q^*(s,a) = E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \qquad \text{Eq. (6-5)}$$

$$V^*(s) = \max_a E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\}$$

And

$$V^*(s) = \max_a \sum_{s'} P_{ss'}^a \left[R_{ss'}^a + \gamma V^*(s')\right] \qquad \text{Eq. (6-6)}$$

$$Q^*(s,a) = \sum_{s'} P_{ss'}^a \left[R_{ss'}^a + \gamma \max_{a'} Q^*(s',a')\right]$$

Dynamic programming (DP) methodologies can be used to find an optimum solution for this problem. It is assumed that the robot has a starting point and a target which makes the problem a finite Markov decision process. Also, the discrete action space is assumed to be an approximation of continuous possible robot actions in each state.

The main idea of dynamic programming is to use value functions to plan the search for better policies. Reinforcement learning provides update rules for improving approximations of the desired value functions. The first step is to compute the value function for each state. This is called policy evaluation or prediction problem in DP literature [157]. Following the previous equations for V^π :

$$V^\pi(s) = E_\pi\{R_t | s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\}$$

The value function for policy π can be rewritten as:

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P_{ss'}^{a} \left[R_{ss'}^{a} + \gamma V^{\pi}(s') \right] \qquad \text{Eq. (6-7)}$$

when $\pi(s,a)$ is the probability of taking action a in state s under policy π, with the expectation that policy π will be followed and $\gamma < 1$.

Iterative and recursive methods are common in solving DP problems. If V_0, V_1, V_2 … are a sequence of approximate value functions, each mapping state space to reward space, given the initial approximation V_0 is chosen arbitrarily, then each successive approximation is obtained using the Bellman equation for V^{π} as an update rule:

$$V_{k+1}(s) = E_{\pi}\left\{ r_{t+1} + \gamma V_k(s_{t+1}) \middle| s_t = s \right\}$$

$$V_{k+1}(s) = \sum_{a} \pi(s,a) \sum_{s'} P_{ss'}^{a} \left[R_{ss'}^{a} + \gamma V_k(s') \right] \qquad \text{Eq. (6-8)}$$

It can be shown that the sequence $\{V_k\}$ will converge to V^{π} as $k \to \infty$[157]. This algorithm is called *iterative policy evaluation*.

6.6 Policy improvement

The reason for calculating the value function is to find better policies. Consider V^{π} is the value function for an arbitrary policy π. To be sure that there is no better policy in state s, or if there is a need to change to a new policy, action a from s can be chosen with the assumption that policy π will be continued thereafter. The value for this new policy is:

$$Q^{\pi}(s,a) = E_{\pi}\left\{ r_{t+1} + \gamma V^{\pi}(s_{t+1}) \middle| s_t = s, a_t = a \right\}$$

$$Q^{\pi}(s,a) = \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V^{\pi}(s') \right]$$

If this is greater than $V^{\pi}(s)$, then it is expected to select a every time s is encountered.

6.7 Perceptual states in DARPA Grand Challenge case

The output of each sensor indicates a reward for reaching a particular state. Some sensors are more relevant to the state than others. For example, when the robot is in 'avoid obstacle' state, information collected from its laser scanner is more important than that from its GPS. To calculate the final reward for this state, each sensor has a different voting right (weight). In other words, the perceived value of information from each sensor is different. To organize the voting regime according to the experts' perception, the QFD approach explained in Table 6-2 can be used.

The number of possible actions, states, transition probabilities, and rewards depends on the specific problem and mission. To clarify the perception optimization approach consider the DARPA Grand Challenge example. Gibbs reports the following story [131]:

> "In a ***mobile*** office set up near the starting chutes 13 route editors, three speed setters, three managers, a statistician and a strategist waited for the DARPA CD. Within minutes of its arrival, a "preplanning" system that the team had built with help from Science Applications International Corporation, a major defense contractor, began overlaying the race area with imagery drawn from a 1.8-terabyte database containing three-foot-resolution satellite and aerial photographs, digital-elevation models and laser-scanned road profiles gathered during nearly 3,000 miles of reconnaissance driving in the Mojave.
>
> The system automatically created initial routes for Sandstorm and H1ghlander, the team's two racers, by converting every vertex to a curve, calculating a safe speed around each curve, and knocking the highest allowable speeds down to limits derived from months of desert trials at the Nevada Automotive Testing Center. The software then divided the course and the initial route into segments, and the manager assigned one segment to each race editor.
>
> Flipping among imagery, topographic maps and reconnaissance scans, the editors tweaked the route to take tight turns the way a race driver would and to shy away from cliff edges. They marked "slow" any sections near gates, washouts and underpasses; segments on paved roads and dry lake beds were assigned "warp speed."

> The managers repeatedly reassigned segments so that at least four pairs of eyes reviewed each part of the route. Meanwhile, in a back room, team leaders pored over histograms of projected speeds and estimates of elapsed time.
>
> The software then divided the course and the initial route into segments, and the manager assigned one segment to each race editor.
>
> Flipping among imagery, topographic maps and reconnaissance scans, the editors tweaked the route to take tight turns the way a race driver would and to shy away from cliff edges. They marked "slow" any sections near gates, washouts and underpasses; segments on paved roads and dry lake beds were assigned "warp speed."”

The DARPA Grand Challenge was taken on October 8th, 2005. Two hours before the race a CD containing 2,935 GPS waypoints, speed limits, and width of corridors was given to each team. Some teams used satellite imagery and reconnaissance information of the course to plan a strategy for each segment. Each segment was assigned to a team of experts to define features such as ‘slow’, ‘speed up’, and ‘stay away from cliff edges’.

These features are examples of *perceptual states.* Moving from one *state* to another is a critical decision that needs to be *optimized*. Previous information about the route, such as GPS data and images, can provide *transition probabilities* for moving among states.

No map is entirely up-to-date and accurate. Information collected from sensors, along with previous knowledge of route, will provide *reward* signals and will be a source of correction in *policy* selection.

The DARPA Grand Challenge report reveals an important characteristic of robot navigation that generally has not been emphasized in the literature: integration of human perception and strategy with robot navigation algorithms. The perception theory presented here is equipped with a strategy planning tool, QFD, which links strategy and planning stages with dynamic programming optimization tools. The main rationale behind this approach is optimization theory as applied to robot design and development.

Chapter 7 : Conclusion

"An author is a fool who, not content with boring those he lives with, insists on boring future generations."

Charles de Montesquieu (1689 - 1755)

7.1 Summary and contribution

In the case of robot navigation, when it is possible to build an accurate map of the environment, following an approximately optimal path is achievable. However, in a dynamic situation, when obstacles are not stationary or when enough information about the environment is not available, traditional path planning approaches are not sufficient. That is where other methodologies, e.g. the perception-based control, play an important role.

In this dissertation, several prototype robots were studied. An unmanned ground vehicle, the Bearcat Cub, a soil sampling survey robot, and a snow prevention robot were introduced. Potential applications of these robots in environments such as mine fields were discussed as well.

The problem of perception-based control was formulated from several points of view. The creative control framework was expanded to include the perception-based task control center with fuzzy and neuro-fuzzy elements. This adds qualitative reasoning to the dynamic programming optimality approach.

A statistical model was used to model the uncertainty that is an inherent part of perception. In addition, the spatial knowledge approach was used to model autonomous navigation of the Bearcat Cub robot. Applying an optimization approach to model human perception in the design process, as well as navigation algorithm of the robot, is another contribution of this research.

7.2 Future work

A solid theory for computation of perception is currently missing. Soft computing methodologies, and specifically fuzzy theory, are slowly moving in this direction. A multidisciplinary approach that combines the strengths of current methodologies- e.g. neural networks, genetic algorithm, fuzzy logic, machine learning, statistical theory, and optimization theory- appears to be a promising research direction.

Most research on verbal communication with robots has mainly focused on issuing commands, such as activating pre-programmed procedures using a limited vocabulary. These procedures directly convert the voice commands to measurements without computing perceptions. Mimicking human perception and to some degree perception computing, can be investigated much further.

In the area of robot design, home robotics also has great potential. There are increasing numbers of products such as vacuum cleaners, pet or toy robots, and lawnmowers. Different aspects of home robotics can be studied in future work.

In addition, autonomous navigation of vehicles is an open-ended problem that requires sophisticated methodologies. Implementation of the previously-discussed creative control framework in a real product is another suggested future task.

Finally, using an industrial engineering approach to model all stages of design for unmanned vehicles in a systematic fashion provides interesting future avenues for research.

References

1. Chatterjee, R., Matsuno, F., "Use of single side reflex for autonomous navigation of mobile robots in unknown environments," *Robotics and Autonomous Systems,* 35, pp. 77-96, 2001.

2. Cahlink, G. Making Sacrifices for the Future. http://www.govexec.com/top200/03top/top03s4.htm

3. Oriolo, G., Ulivi, G., Vendittelli, M., "Fuzzy Maps: A New Tool for Mobile Robot Perception and Planning," *Journal of Robotic Systems,* 14, (3), pp. 179-197, 1996.

4. Kubota, N., et al., "Learning of mobile robots using perception-based genetic algorithm," *Measurement,* 29, pp. 237-248, 2001.

5. Arkin, R. C., *Behavior-based robotics*. MIT Press: Cambridge, MA, 1999.

6. Liao, X. Creative Learning for Intelligent Robots. University of Cincinnati, Cincinnati, 2005.

7. Alhaj Ali, S. Technologies for Autonomous Navigation in Unstructured Outdoor Environments. University of Cincinnati, Cincinnati, 2004.

8. Boden, M., "AI's Half-Century," *AI Magazine,* 16, (2), pp. 69-99, 1995.

9. Duro, R. J., Santos, J., Becerra, J. A., Some approaches for reusing behavior based robot cognitive architectures obtained through evolution. In *Biologically inspired robot behavior engineering*, vol. 109 pp. 239-260, Duro, R. J., Santos, J., Grana, M., Physica-Verlag: New York, 2003.

10. Barber, H. M., Skarmeta, A. G., "A Framework for Defining and Learning Fuzzy Behaviors for Autonomous Mobile Robots," *International Journal of Intelligent Systems,* 17, pp. 1-20, 2002.

11. Lauria, S., et al., "Mobile robot programming using natural language," *Robotics and Autonomous Systems,* 38, pp. 171-181, 2002.

12. Pratihar, D. K., Deb, K., Ghosh, A., "A genetic-fuzzy approach for mobile robot navigation among moving obstacles," *International Journal of Approximate Reasoning,* 20, pp. 145-172, 1999.

13. Al-Khatib, M., Saade, J. J., "An effcient data-driven fuzzy approach to the motion planning problem of a mobile robot," *Fuzzy Sets and Systems,* 134, pp. 65-82, 2003.

14. Tunstel, E., Oliveira, M. A. A. d., Berman, S., "Fuzzy Behavior Hierarchies for Multi-Robot Control," *International Journal of Intelligent Systems,* 17, pp. 449-470, 2002.

15. Seraji, H., "Fuzzy Traversability Index:A New Concept for Terrain-Based Navigation," *Journal of Robotic Systems,* 17, (2), pp. 75-91, 2000.

16. Li, W., Ma, C., Wahl, F. M., "A neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments," *Fuzzy Sets and System,* 87, pp. 133-140, 1997.

17. Michaud, F., Lachiver, G., Dinh, C. T. L., "Architectural Methodology Based on Intentional Configuration of Behaviors," *Computational Intelligence,* 17, (1), pp. 132-156, 2001.

18. Skubic, M., et al. "Using Spatial Language in a Human-Robot Dialog," *IEEE International Conference on Robotics &Automation*, pp. 4143-4148, Washington, DC, 2002.

19. Skubic, M., et al. "Generating linguistic spatial descriptions from sonar readings using the histogram of forces," *IEEE International Conference on Robotics & Automation*, vol. 1, pp. 485-490 Seoul, Korea, 2001.

20. Skubic, M., et al. "Extracting navigation states from a hand-drawn map," *IEEE International Conference on Robotics & Automation*, vol. 1, pp. 259-264 Seoul, Korea, 2001.

21. Ghaffari, M., Hall, E. L. "Robotics and nature, from primitive creatures to human intelligence," *SPIE Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, vol. 5608, pp. 169-176, Philadelphia, 2004.

22. Adams, B., et al., "Humanoid Robots: A New Kind of Tool," *IEEE Intelligent Systems,* 15, (4), pp. 25-31, 2000.

23. Hall, E. L., Hall, B. C., *Robotics, A User Friendly Introduction*. Holt, Rinehart and Winston: 1985.

24. Halme, A., Luksch, T., Ylonen, S., "Biomimicing motion control of the WorkPartner robot," *Industrial Robot: An International Journal,* 31, (2), pp. 209–217, 2004.

25. Dario, P., Guglielmelli, E., Laschi, C., "Humanoids and Personal Robots: Design and Experiments," *Journal of Robotic Systems,* 18, (12), pp. 673–690, 2001.

26. Zlatev, J., "The epigenesis of meaning in human beings and possbly in robots," *Lund University Cognitive Studies,* 79, 1999.

27. Fong, T., Nourbakhsh, I., Dautenhahn, K., "A survey of socially interactive robots," *Robotics and Autonomous Systems,* 42, pp. 143–166, 2003.

28. Dautenhahn, K., "Getting to know each other - artificial social intelligence for autonomous robots," *Robotics and Autonomous Systems,* 16, pp. 333-356, 1995.

29. Dautenhahan, K., Billiard, A. "Bringging up robots or the psychology of socially intelligent robots: From theory to implementation," *Autonomus Agents*, 1999.

30. Kanda, T., et al., "Effects of Observation of Robot–Robot Communication on Human–Robot Communication," *Electronics and Communications in Japan,* 87, (5), pp. 691-700, 2004.

31. Clark, J. E., et al. "Biomimetic Design and Fabrication of a Hexapedal Running Robot," *IEEE International Conference on Robotics and Automation*, 2001.

32. Zhou, C. Soft Computing for Intelligent Robotic Systems. http://www.dia.fi.upm.es/doctorado/temasavanzados/anteriores/0102/4.htm

33. Zadeh, L. A., "Applied Soft Computing – Foreword," *Applied Soft Computing,* 1, pp. 1-2, 2001.

34. Zadeh, L. A., "Fuzzy Sets," *Information and Control,* 8, pp. 338-353, 1965.

35. Novak, V., Perfilieva, I., Mockor, J., *Mathematical Principles of Fuzzy Logic*. Kluwer Academic Publishers: Boston, 1999.

36. Zadeh, L. A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - I," *Information Sciences,* 8, pp. 199-249, 1975.

37. Zadeh, L. A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - II," *Information Sciences,* 8, pp. 301-357, 1975.

38. Zadeh, L. A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - III," *Information Sciences,* 9, pp. 43-80, 1975.

39. Zadeh, L. A., "Fuzzy logic and approximate reasoning," *Synthese,* 30, pp. 407-428, 1975.

40. Kochen, M., Application of Fuzzy Sets in Psychology. In *Fuzzy Sets and their Application to Cognitive and Decision Processes*, pp. 395-408, Zadeh, L. A., et al., Academic Press: New York, 1975.

41. Driankov, D., A Reminder on Fuzzy Logic. In *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, Driankov, D., Saffiotti, A., Physica-Verlag Heidelberg: New York, 2001.

42. Ghaffari, M., Ali, S. A., Hall, E. L., "A Perception –Based Approach toward Robot Control by Natural Language," *Intelligent Engineering Systems through Artificial Neural Networks,* 14, pp. 391-396, 2004.

43. Zadeh, L. A. "A new direction in fuzzy logic-toward automation of reasoning with perceptions," *Fuzzy Systems, IEEE International*, vol. 1, pp. 1-5, IEEE, 22-25 Aug,1999.

44. Kubota, N., et al., "Fuzzy and Neural Computing for Communication of a Partner Robot," *Journal of Multi-Valued Logic & Soft Computing,* 9, pp. 221-239, 2003.

45. Dvorak, A., Novak, V., "Formal theories and linguistic descriptions," *Fuzzy Sets and Systems,* 143, pp. 169-188, 2004.

46. Novak, V., Perfilieva, I., "On the Semantics of Perception-Based Fuzzy Logic Deduction," *International Journal of Intelligent Systems,* 19, pp. 1007–1031, 2004.

47. Gasos, J., Rosetti, A., "Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments," *Fuzzy Sets and System,* 107, pp. 1-24, 1999.

48. Elfes, A., "Sonar-based real world mapping and navigation," *IEEE Journal of Robotics and Automation,* 3, pp. 249-265, 1987.

49. Tirumalai, A., Schunck, B., Jain, R., "Evidential reasoning for building environment maps," *IEEE Transactions on Systems, Man and Cybernetics,* 25, (1), pp. 10-20, 1995.

50. Kim, H., Swain, P., "Evidential reasoning approach to multi source data classification in remote sensing," *IEEE Transactions on Systems, Man and Cybernetics,* 25, (8), pp. 1257-1265, 1995.

51. Saffiotti, A., Wesley, L., Perception-based self localization using fuzzy locations. In *Reasoning with Uncertainty in Robotics, Lecture Notes in Artificial Intelligence*, pp. 368-385, Dorst, L., Springer: Berlin, 1996.

52. Carruthers, P., Chamberlain, A., *Evolution and the human mind, modularity, language and meta-cognition*. Cambridge University: Cambridge, 2000.

53. Zadeh, L. A., "Toward a perception-based theory of probabilistic reasoning with imprecise probabilities," *Journal of Statistical Planning and Inference,* 105, pp. 233-264, 2002.

54. Zadeh, L. A., "From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 46, (1), pp. 105 -119, 1999.

55. Baud, R., "Present and future trends with NLP," *International Journal of Medical Informatics,* 52, pp. 133-139, 1998.

56. Cangelosi, A., Harnad, S., "The adaptive advantage of symbolic theft oversensorimotor toil: Grounding language in perceptual categories," *Evolution Communication,* 4, (1) 2002.

57. Romano, J. M. G., et al., "A generic natural language interface for task planning application to a mobile robot," *Control Engineering Practice,* 8, pp. 1119-1133, 2000.

58. Harris, L. R., "ROBOT: a high performance natural language processor for data base query," *ACM SIGART Newsletter,* 61, pp. 39-40, 1977.

59. Winograd, T., *Understanding natural language*. Academic Press: New York, 1972.

60. Selfridge, M., Vannoy, W., "A natural language interface to a robot assembly system," *IEEE Journal of Robotics and Automation,* 2, (3), pp. 167-171, 1986.

61. Brown, M. K., Buntschuh, B. M., Wilpon, J. G., "A perceptive spoken language understanding robot," *IEEE Transactions on Systems, Man and Cybernetics,* 22, (6), pp. 1390-1402, 1992.

62. Marciniak, J., Vetulani, Z. y., "Ontology of Spatial Concepts in a Natural Language Interface for a Mobile Robot," *Applied Intelligence,* 17, pp. 271-274, 2002.

63. Gasos, J., Integrating linguistic descriptions and sensor observations for the navigation of autonomous robots. In *Fuzzy logic Techniques for autonomous vehicle navigation*, vol. 61 pp. 313-339, Driankov, D., Saffiotti, A., Physica-Verlag: New York, 2001.

64. Cimino, J., "Knowledge-based approaches to maintenance of a large controlled medical terminology," *International Journal of Medical Informatics Association,* 1, pp. 35-50, 1994.

65. Feldman, R., Dagan, I., Hirsh, H., "Mining Text Using Keyword Distributions," *Journal of Intelligent Information Systems,* 10, pp. 281-300, 1998.

66. Teich, E., *Systemic Functional Grammar in Natural Language Generation*. Cassell: New York, 1999.

67. Oostendrop, H., Mul, S., Cognitive Aspects of Electronic Text Processing. In *Advances in Discourse Process*, Ablex: New Jersey, 1996.

68. Sampson, G., *Evolutionary Language Understanding*. Cassell: New York, 1996.

69. Cummins, R., Cummins, D., *Minds, Brains, and Computers: The Foundation of Cognitive Science, An Anthology*. Blackwell: Malden, MA, 2000.

70. Gaylor, J., et al. *Senior Design Report: Bearcat Cub*; University of Cincinnati: Cincinnati, 2005; p^pp.

71. *The Bearcat Cub design report for 14th Intelligent Ground Vehicle Competition*; Univeristy of Cincinnati Robotics Team: Cincinnati, 2006; p^pp.

72. Alhaj Ali, S. M. Technologies for Autonomous Navigation in Unstructured Outdoor Environments. PhD, University of Cincinnati, Cincinnati, OH, 2003.

73. Alhaj Ali, S. M., et al., Dynamic Simulation of Computed-Torque Controllers for a Wheeled Mobile Robot Autonomous Navigation in Outdoor Environments. In *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 13 pp. 511-516, ASME: New York, 2003.

74. Rajagopalan, R., "A generic kinematic formulation for wheeled mobile robots," *Journal of Robotic Systems,* 14, (2), pp. 77-91, 1997.

75. Ghaffari, M., et al., "Design of an Unmanned Ground Vehicle, Bearcat III, Theory and Practice," *Journal of Robotic Systems,* 21, (9) 2004.

76. Wu, W., Chen, H., Woo, P., "Time optimal path planning for a wheeled mobile robot," *Journal of Robotic Systems,* 17, (11), pp. 585-591, 2000.

77. Levitt, T. S., Lawton, D. T., "Qualitative Navigation for Mobile Robots," *Artificial Intelligence,* 44, pp. 305-360, 1990.

78. Sethuramasamyraja, B., Ghaffari, M., Hall, E. "Automatic Calibration and Neural Networks for Robot Guidance," *SPIE Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, vol. 5267, pp. 137-144, Providence, RI, 2003.

79. Thorpe, C. E., *Vision and navigation: the Carnegie Mellon Navlab*. Kluwer Academic Publishers: Boston, 1990.

80. Robert, L. "Calibration Without Feature Extraction," *International Conference on Pattern Recognition*, pp. 704-706, Computer Society Press, Jerusalem, 1994.

81. Li, M. X., Lavest, J. M., "Some Aspects of Zoom-Lens Camera Calibration," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 1105-1110, 1996.

82. Hong, F., Baozong, Y., "An Accurate and Practical Camera Calibration System for 3D Computer Vision," *Chinese Journal of Electronics,* 1, (1), pp. 63-71, 1991.

83. Weng, J., Cohen, P., Herniou, M., "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14, (10), pp. 965-980, 1992.

84. Tsai, R. Y., "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf Cameras and Lenses," *IEEE Transaction on Robotics and Automation,* 3, (4), pp. 323- 344, 1987.

85. Zhang, Z., Faugeras, O., Ayache, N. "Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints," *International Conference on Computer Vision*, pp. 177-186, 1988.

86. Zhang, Z., et al., "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry," *Artificial Intelligence Journal,* 78, pp. 87-119, 1995.

87. Zhang, Z., Faugeras, O. D., "A 3D World Model Builder with a Mobile Robot," *International Journal of Robotics Research,* 11, (4), pp. 269-285, 1992.

88. Zhang, Z., Luong, Q. T., Faugeras, O., "Motion of an Uncalibrated Stereo Rig: Self-Calibration and Metric Reconstruction," *IEEE Transaction on Robotics and Automation,* 12, (1), pp. 103-113, 1996.

89. Worrall, A. D., Sullivan, G. D., Baker, K. D. "A simple, intuitive camera calibration tool for natural images," *4th British Machine Vision Conference*, Berkshire, UK, 1994.

90. Hall, E. L. Robot vision course notes. University of Cincinnati, Cincinnati, 2005.

91. Parasnis, S. S. Four Point Calibration and a Comparison of Optical Modeling and Neural Networks for Robot Guidance. University of Cincinnati, Cincinnati, 1999.

92. Hall, E., et al., "A Comparison of Two and Three Dimensional Imaging," *to be published in SPIE Intelligent Robots and Computer Vision : Algorithms, Techniques, and Active Vision*, 2006.

93. Ghaffari, M., et al. "Internet-based Control for the Intelligent Unmanned Ground Vehicle: Bearcat Cub," *SPIE Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, vol. 5267, pp. 90-97, Providence, 2003.

94. Devices Connected to the Internet. http://dir.yahoo.com/Computers_and_Internet/Internet/Devices_Connected_to_the_Internet/

95. Paulos, E., Canny, J., "Ubiquitous tele-embodiment: applications and implications," *International Journal of Human-Computer Studies,* 46, pp. 861-877, 1997.

96. Petersen, I., "Telepathology by the Internet," *Journal of Pathology,* 191, pp. 8-14, 2000.

97. You, S., "A low-cost internet-based telerobotic system for access to remote laboratories," *Artificial Intelligence in Engineering,* 15, pp. 265-279, 2001.

98. Frederik, O., "Remote Monitoring and Control of Electrochemical Experiments via the Internet Using “Intelligent Agent” Software," *Electroanalysis,* 11, (14), pp. 1027-1032, 1999.

99. Mansfield, J., "Development of a System to Provide Full, Real-time Remote Control of a Scanning Electron Microscope across the Second Generation Internet: The Teaching SME," *Microscopy and Microanalysis,* 6, pp. 31-41, 2000.

100. Baur, C., "Robotic nanomanipulation with a scanning probe microscope in a networked computing environment," *Journal of Vacuum Society,* 15, (4), pp. 1577-1580, 1997.

101. Churms, C., "The remote control of nuclear microprobes over the Internet," *Nuclear Instruments and Methods in Physics Research B,* 158, pp. 124-128, 1999.

102. Solina, B. P. F., "User interface for video observation over the internet," *Journal of Network and Computer Applications,* 21, pp. 219-237, 1998.

103. Wang, C., "Implementation of remote robot manufacturing over Internet," *Computers in Industry,* 45, pp. 215-229, 2001.

104. Tay, F., "Distributed rapid prototyping – a framework for Internet prototyping and manufacturing," *Integrated Manufacturing System,* 12, (6), pp. 409-415, 2001.

105. Wang, Z., "Architecture for Agile Manufacturing and Its Interface with Computer Integrated Manufacturing," *Journal of Material Processing Technology,* 61, pp. 99-103, 1996.

106. Schilling, K., "Telediagnosis and teleinspection potential of telematic techniques," *Advances in Engineering Software,* 31, pp. 875-879, 2000.

107. Winfield, Holland, O., "The application of wireless local area network technology to the control of mobile robots," *Microprocessors and Microsystems,* 23, pp. 597-607, 2000.

108. *The Joint Architecture for Unmanned Systems: Reference Architecture Specification*. Ed 3.2, 2004.

109. Ghaffari, M., et al. "Mines and human casualties, a robotics approach toward mine clearing," *SPIE Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, vol. 5608, pp. 306-312, 2004.

110. Dincerler, V., "Landmines and Mine Clearance Technologies," *International Secretariat*, 1995.

111. Huang, Q.-J., Nonami, K., "Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control," *Mechatronics,* 13, pp. 773-790, 2003.

112. "Annual Report 2002," *International Campaign to Ban Landmines*, 2002.

113. Benini, A., Moulton, L. H., Conley, C. E., "Landmines and Local Community Adaptation," *Journal of Contingencies and Crisis Management,* 10, (2), pp. 82-94, 2002.

114. Nicoud, J. D., Machler, P., "Robots for Anti-personnel Mine Search," *Control Eng Practice,* 4, (4), pp. 493-498, 1996.

115. Nicoud, J. D., "Vehicles and robots for humanitarian demining," *Industrial Robot,* 24, (2), pp. 164-168, 1997.

116. Khuraibet, A. M., "Nine years after the invasion of Kuwait: the impacts of the Iraqi left-over ordnance," *The Environmentalist,* 19, pp. 361-368, 1999.

117. Bruschini, C., Gros, B. "A survey of current sensor technology research for the detection of landmines," *International Workshop on Sustainable Humanitarian Demining*, Zagreb, Croatia, 1997.

118. Tunstel, E., Howard, A., Seraji, H., "Rule-based reasoning and neural network perception for safe off-road robot mobility," *Expert Systems,* 19, (4), pp. 191-200, 2002.

119. Maaref, H., Barret, C., "Sensor-based navigation of a mobile robot in an indoor environment," *Robotics and Autonomous Systems,* 38, pp. 1-18, 2002.

120. Colona, E., et al., "An integrated robotic system for antipersonnel mines detection," *Control Engineering Practice,* 10, pp. 1283–1291, 2002.

121. Ghaffari, M., Cao, P., Hall, E. "Autonomous Techniques for Soil Sampling in Hazardous Fields," *International Conference on Human Impacts on Soil Quality Attributes in Arid and Semiarid Regions*, pp. 159-163, Isfahan, 2005.

122. Cao, P. Autonomous Runway Soil Survey System with the Fusion of Global and Local Navigation Mechanism. PhD University of Cincinnati, Cincinnati, 2004.

123. Hausmann, M. R., *Engineering Principles of Ground Modification*. McGraw-Hill Publishing Company: New York, 1990.

124. Ghaffari, M., Liao, S., Hall, E. L. "A Framework for the Natural Language Perception-based Creative Control of Unmanned Ground Vehicles," *SPIE Defense and Security Symposium* vol. 5422, pp. 218-226, Orlando (Kissimmee), 2004.

125. Liao, X., et al. "Creative Control for Intelligent Autonomous Mobile Robots," *Intelligent Engineering Systems Through Artificial Neural Networks, ANNIE*, vol. 13, St. Louis, MI, 2003.

126. Werbos, P. "Optimal Neurocontrol: Practical Benefits, New Results and Biological Evidence," *Wescon Conference Record*, pp. 580-585, 1995.

127. Prokhorov, D., D.Wunsch, "Adaptive Critic Designs," *Neural Networks,* 8, (5), pp. 997-1007, 1997.

128. Lendaris, G. G., Shannon, T. T., Rustan, A. "A Comparison of Training Algorithms for DHP Adaptive Critic Neurocontrol," *Neural Networks, 1999. IJCNN '99. International Joint Conference on,*, vol. 4, pp. 2265 -2270, 1999.

129. Venayagamoorthy, G. K., Harley, R. G., Wunsch, D. C., "Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator," *IEEE Transactions on Neural Networks,* 13, (3), pp. 764-773, May 2002.

130. Werbos, P. J., Approximate Dynamic Programming for Real-Time Control and Neural Modeling. In *Handbook of Intelligent Control*, pp. 493--525, White, A. D., Sofge, D. A., Van Nostrand Reinhold: 1992.

131. Gibbs, W., "Innovations from a Robot Rally," *Scientific American,* 294, (1), pp. 64-71, 2006.

132. Hashimoto, S., Kojima, F., Kubota, N. "Perceptual System for A Mobile Robot under A Dynamic Environment," *IEEE International Symposium on Computational Intelliieoce in Robotics and Automation*, pp. 747-752, Japan, 2003.

133. Hayashi, Y., Buckley, J., "Approximations between fuzzy expert systems and neural networks," *International Journal of Approximate Reasoning* 10, pp. 63–73, 1994.

134. Benitez, J. M., Castro, J. L., Requena, I., "Are artificial neural networks black boxes?," *IEEE Transaction on Neural Networks* 18, pp. 1156–1164, 1997.

135. Mitra, S., Hayashi, Y., "Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework," *IEEE Transactions on Neural Networks,* 11, (3), pp. 746-768, 2000.

136. Pal, S. K., Mitra, S., *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*. Wiley: New York, 1999.

137. Lin, C. T., Lee, C. S. G., *Neural Fuzzy Systems—A Neuro–Fuzzy Synergism to Intelligent Systems*. Prentice-Hall: Englewood Cliffs, NJ, 1996.

138. Nauck, D., Klawonn, F., Kruse, R., *Foundations of Neuro–Fuzzy Systems*. Wiley: Chichester, U.K. , 1997.

139. Weiner, N., *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. John Wiley and Sons: New York, 1949.

140. Kolmogorov, A., "Interpolation and Extrapolation of Stationary Random Sequences," *Bulletin of the Academy of Sciences of the USSR* 5, 1941.

141. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering,* 82, pp. 35-45 1960.

142. Kalman, R. L., Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Transactions of the ASME–Journal of Basic Engineering,* 83, pp. 95-107, 1961.

143. Smith, R. C., Cheeseman, P., "On the Estimation and Representation of Spatial Uncertainty," *International Journal of Robotics Research,* 5, (4), pp. 56-68, 1987.

144. Durrant-White, H. F., "Consistent Integration and Propagation of Disparate Sensor Observations," *International Journal of Robotics Research,* 6, (3), pp. 3-24, 1987.

145. Faugeras, O., Ayache, N., Faverjon, B. "Building Visual Maps by Combining Noisey Stereo Measurements," *IEEE International Conference on Robotics and Automation*, San Francisco, CA, 1986.

146. Brooks, R. "Visual Map Making for a Mobile Robot," *IEEE International Conference on Robotics and Automation*, St. Louis, 1985.

147. Chatila, R., Laumond, J. P. "Position Referencing and Consistent World Modeling for Mobile Robots," *IEEE International Conference on Robotics and Automation*, St. Louis, 1985.

148. Crowley, J. L. "Asynchronous Control of Orientation and Displacement in a Robot Vehicle," *IEEE Conference on Robotics and Automation*, Scottsdale, AZ, 1989.

149. Crowley, J. L. "Mathematical Foundations of Navigation and Perception For an Autonomous Mobile Robot," *International Workshop on Reasoning with Uncertainty in Robotics*, pp. 9-51, University of Amsterdam, The Netherlands, 1995.

150. Crowley, J. L., Wallner, F., Schiele, B. "Position Estimation Using Principal Components of Range Data," *IEEE International Conference on Robotics & Automation*, pp. 3121-3128, Leuven, Belgium, 1998.

151. Bar-Shalom, Y., Li, X. R., Kirubarajan, T., *Estimation with applications to tracking and navigation* Wiley: New York, 2001.

152. Ghaffari, M., Hall, E. L., "Perception and Navigation Modeling in Humans and Intelligent Systems," *Intelligent Engineering Systems through Artificial Neural Networks,* 15, pp. 43-49, 2005.

153. Previc, F. H., "The Neuropsychology of 3-D Space," *Psychological Bulletin,* 124 (2), pp. 123-164, 1998.

154. Werner, S., et al. "Spatial Cognition: The Role of Landmark, Route, and Survey Knowledge in Human and Robot Navigation," *Informatik 97*, pp. 4150, Springer, 1997.

155. Denney, R., *Succeeding with Use Cases: Working Smart to Deliver Quality*. Addison-Wesley Boston, MA, 2005.

156. Bertsekas, D. P., *Dynamic Programming and Optimal Control* Athena Scientific: Nashua, NH, 2005.

157. Sutton, R. S., Barto, A. G., *Reinforcement Learning: An Introduction*. MIT Press: Cambridge, MA, 1998.

158. Genaidy, A. M., Karwowski, W., Shoaf, C., "The fundamentals of work system compatibility theory: An integrated approach to optimization of human performance at work " *Theoretical Issues in Ergonomics Sciences,* 3, pp. 346-368, 2002.

159. Ghaffari, M., Genaidy, A. "A Neural Network Model for Human Performance Optimization in the Workplace," *2nd International IEEE EMBS Conference on Neural Engineering*, Washington DC, 2005.

160. Wallace, W., et al., "Assessing the compatibility of work system factors through an integrative model: A case study," *International Journal of Occupational Safety and Ergonomics,* 9, pp. 25-33, 2003.

Appendix A: A case study

Human perception and performance optimization

Human perception is the subject of study in many fields such as psychology, biology, management, and engineering, in addition to intelligent systems and robotics. As one could imagine, getting the optimum results from the '*machine*' and '*human*' is a major topic of study in some of these fields. 'How to improve robot's performance by studying human perception' was the main question in previous sections. In this section, perception and intelligent system's techniques will be used to study human performance. A neural network will estimate the optimal conditions that an employee should work in based on the job demands.

In a series of studies by Genaidy et al. the concept of *work compatibility* (WC) and its relation with *work energizers* (WE) and *work demands* (WD) have been studies [158-160]. Work-related factors are classified into two major categories depending upon their impact on human performance. Work demands (e.g. making decision, work conflict) are forces with negative impact in the sense of energy replenishment. Work energizers (e.g. financial incentives, social recognition) are factors with positive impact on the flow of energy in the human engine. The work compatibility is an integrated work design criterion that improves different aspects of human performance in a workplace.

The magnitude of work energizer, work demand, or work compatibility is each described by five linguistic levels, that is, very low (VL), low (L), moderate (M), high (H), and very high (VH) [160]. For mathematical derivation of WC, we express these levels by numerical numbers ranging from 1 to 5 corresponding to the five linguistic levels, respectively.

Work compatibility is written in a matrix form as $[WC_{ij}]$ where *i* and *j* correspond to the respective levels of *WD* (row) and *WE* (column) taking values from 1 to 5. For example, WC_{23} is the work compatibility that corresponds to *WD*=2 and *WE*=3 as demonstrated in (1) and (2).

Neural Network Model

As mentioned work compatibility matrix represents WC as a function of the WD and WE. For each entry, the row and column numbers are values of WD and WE respectively. This provides 25 training data for the following neural network.

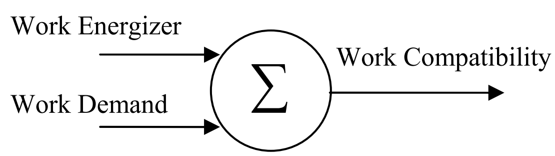


Figure A-1: Mathematical concept of work compatibility (WC)

Table A-1: Linguistic values of work compatibility[160]

	WE				
WD	Very low	Low	Moderate	High	Very high
Very low	L	L	L	L	L
Low	L	M	M	M	M
Moderate	L	M	H	H	H
High	VL	L	M	VH	H
Very high	VL	VL	L	M	M

$$[WC] = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 3 & 3 & 3 \\ 2 & 3 & 4 & 4 & 4 \\ 1 & 2 & 3 & 5 & 4 \\ 1 & 1 & 2 & 3 & 3 \end{bmatrix} \quad \text{or} \quad [WC] = \begin{bmatrix} L & L & L & L & L \\ L & M & M & M & M \\ L & M & H & H & H \\ VL & L & M & VH & H \\ VL & VL & L & M & M \end{bmatrix}$$

A generalized regression neural network was constructed. This is a kind of radial basis network that is often used for function approximation.

Matlab^{T} code for constructing and training the network is as follow:

```
P = [1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5;
1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5];
T = [2 2 2 2 2 2 3 3 3 3 2 3 4 4 4 1 2 3 5 4 1 1 2 3 3];
Data2 = [ 2  2  2  2 2;  2  3 3 3 3; 2 3 4 4 4; 1 2 3 5 4;  1 1 2 3 3];
spread = 0.5;
net = newgrnn(P,T, spread);
Y = sim(net,P);
```

P is the input matrix which corresponds to row and column numbers is matrix (1) that are WD and WE respectively. T is the target vector that represents values of WC for each given WE and WD based on table A-1. The spread parameter determines the smoothness of function. The larger spread will result in a smoother function in exchange to a higher sum square error. To fit data closely a spread smaller than the typical distance between input vectors is appropriate. For this problem spread=0.5 was chosen that resulted Figure A-2. For this function sum square error is 0.1975. It means that the neural network approximates work compatibility as a function of work demand and work energizer well. This function (and corresponding network) represents the expert knowledge for work compatibility for different values of work demand and work energizer. Figure A-2 illustrates WC as a function of WD and WE.

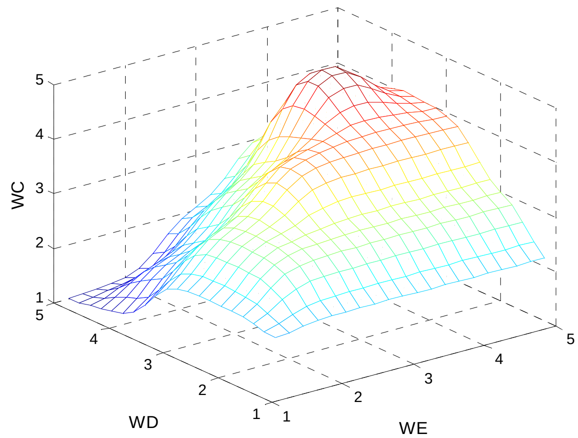


Figure A-2 Approximated WC function

This function shows where the maximum work compatibility is located. The rate of change around the local and global optima is extractable as well. A proper combination of WD and WE should be chosen to avoid low compatibility as well as high rate of change (danger of dramatic change). More discussion about this graph can be found in reference [159].

Index