PERCEPTUAL NAVIGATION FOR SEMI-AUTONOMOUS WHEELCHAIR OPERATIONS

by

HAJIME UCHIYAMA

(Under the direction of Walter D. Potter)

ABSTRACT

The objective of this research is to develop a semi-autonomous architecture for an intelligent wheelchair, "a wheelchair-navigation agent," that comprises perceptual and navigational capabilities by means of computer science, robotics, and sensor technology. The perceptual navigation system consists of a computer, a collection of sensors (e.g., ultrasonic sensors and a CCD camera), and a novel man-machine tactile interface (*Vibrotactile Glove*). A customized behavior-based control architecture is employed to implement the perceptual and navigational assistance (*perceptual/navigational behaviors*). The wheelchair-navigation agent aims at improving independent mobility of multi-disabled individuals, and this thesis presents integration of sensory information and human-machine interaction.

INDEX WORDS: Artificial Intelligence, Assistive Technology, Computer Vision, Robotics

PERCEPTUAL NAVIGATION FOR SEMI-AUTONOMOUS WHEELCHAIR OPERATIONS

by

HAJIME UCHIYAMA

B.Eng., Hokkaido University, Japan, 1992

M.Eng., Faculty of Engineering Hokkaido University, Japan, 1994

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2008

PERCEPTUAL NAVIGATION FOR SEMI-AUTONOMOUS WHEELCHAIR OPERATIONS

by

HAJIME UCHIYAMA

Approved:

Major Professor: Walter D. Potter

Committee: Michael A. Covington
Ron McClendon

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2008

DEDICATION

To my family, my friends, and those who gave me inspiration and encouragement.

ACKNOWLEDGMENTS

I would like to thank Dr. Potter, my thesis adviser, for allowing me to conduct the wheelchair project with appropriate advice and long lasting support. His extravagant patience and encouragement should be worth noted.

I would like to thank my committee: Dr. Covington and Dr. McClendon. Specifically much work of designing and implementing our own microcontroller for the Vibrotactile Glove could not have come to exist without Covington's knowledge and skill in micro-electronics.

Much thanks to encouragement from my current co-worker, Jeremy Tarver, Robert Eunice, and the previous co-worker Yuki Ono. Without their supports, I could not have done this far at all.

I thank countless people who either directly or indirectly has given me inspiration and encouragement, and specifically Julia, who originally visited the AI center to seek a help and gave us a new perspective and much feedback during the early stage of the project.

Finally to my father, mother, and sister, I could not more appreciate how much you have supported me for such long years both financially and emotionally, and my love is always with you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this chapter, I give an overview of intelligent robots in assistive technology and survey of previous work followed by the descriptions of semi-autonomous wheelchairs and our motivation. After the statement of our objectives, the design concept of this project is introduced.

1.1 ROBOTICS AND ASSISTIVE TECHNOLOGY

Recent advances in robotic technologies have extended the range of robot applications enormously. Robots are now used in industrial settings for handling and welding, in hospitals for assisting surgeons and nurses, in consumer households for vacuuming or for entertaining, and in more countless fields for performing variety of tasks. The World Robotics 2005 survey predicts that service robots will routinely assist eldery and physically challenged people by the end of 2010, and have large potential for future applications not only within the service industries, but also in our rapidly aging society as future caregivers (United Nations, 2005). While conventional industrial robots are given clearly defined tasks within known environments, service robots must deal with dynamically changing environments that contain a certain degree of unknown parameters. Given goals of the service robots would vary from clear to vague depending on their fields and purposes of employment.

Studies have shown that individuals with disabilities, regardless of their age, benefit substantially from access to independent mobility, including electrically powered wheelchairs (Douglass and Ryan, 1987; Paulsson and Christoffersen, 1989; Verburg et al., 1991). Independent mobility increases vocational or educational opportunities, reduces dependence on caregivers, and promotes

a feeling of self-reliance and self-esteem. Unfortunately, some individuals are unable to independently operate a powered wheelchair due to some combination of motor, sensory, or perceptual impairments. With the aim of assisting those individuals, a number of studies have been conducted in the field of assistive technology which combines robotics and artificial intelligence to develop an intelligent wheelchair system. In robotics, the intelligent wheelchair system is classified as a robotized wheelchair, a subcategory of service robots.

However, almost no robotized wheelchairs are currently on the market, and few are expected in the near future (United Nations, 2005). This may mean that the market for robotized wheelchairs is practically in the stage of infancy; most of the applications mainly serve research-based purposes rather than end customers. The research into robotized wheelchairs is challenging and related to a number of other fields, such as mechatronics, micro-electronics, robotics, computer vision, human-computer interface (HCI), artificial intelligence, assistive technology, psychology, and physiology, all of which are already large fields of research.

1.1.1 Previous Work

The platforms of robotized wheelchairs in research are mostly power wheelchairs; some try to achieve autonomous control over the wheelchairs, and others aim for semi-autonomous control. We briefly survey some selected robotized wheelchair systems.

- Applied AI Systems (TAO Project): Applied AI Systems, Inc. developed the robotic wheelchair prototype, TAO series, conducted by Gomi and Griffith (1998). The goal of the TAO project is to create an add-on system for any standard powered wheelchair that will provide a high level of autonomy. The current model presented to the market is TAO-7 and is mostly used for research and development purposes.[1]

- Kiss Institute of Practical Robotics (Tin Man Project): The Tin Man project is aimed at the development of a low-cost robotic wheelchair to aid people with impaired mobility while

[1] Information is available at `http://www.aai.ca`

focusing on a more user-friendly interface and cost issues (Miller and Slack, 1995). The latest version, Tin Man II, is a robotized wheelchair which exhibits manual navigation with obstacle avoidance override, autonomous control, and manual mode. The Tin Man system was also used as a base platform for the Wheelesley project at MIT (see below).

- MIT Artificial Intelligence Lab (Wheelesley): The goal of the Wheelesley project is the creation of a complete robotic wheelchair system to be used by users unable to manipulate a standard motorized wheelchair. This project aims to establish the system for both indoor and outdoor navigation with automatically switching navigation modes. Wheelesley was developed based on the Tin Man model and interacts with the seated user via a graphical user interface (Yanco, 1998).

- Northeastern University, Deictically controlled wheelchair: The goal of this project is to create a "gopher" robot that can be given commands easily and accurately for the handicapped (Crisman and Cleary, 1998). The robot will retrieve objects in the environment autonomously via commands from users, which are communicated deictically.

- University of Edinburgh (Smart Wheelchair): The CALL Centre of the University of Edinburgh has developed a wheelchair intended to be used by children who do not have the physical, perceptual or cognitive abilities to control an ordinary powered mobility aid (Odor and Watson, 1994).[2] The Smart Wheelchair exhibits collision detection followed by a maneuvering action, follows line tracks laid along the floor, and communicates with the user via a speech synthesizer or other feedback system.

- University of Michigan (NavChair): The NavChair system intends to provide mobility support to users who are unable to drive standard motorized wheelchairs, and this is one of the most successful systems in the '90s (Levine et al., 1999). The NavChair uses several navigation modes and switches them automatically. The navigation modes are as follows: general obstacle avoidance, door passage, and wall following.

[2]Information is available at `http://callcentre.education.ed.ac.uk/`

While there has been much research to develop robotized wheelchairs, very few are sold commercially such as Applied AI and CALL Centre, and none of those systems are intended to be used outside of a research lab or a training facility.

1.1.2 SEMI-AUTONOMOUS WHEELCHAIRS

With autonomous control, the system probes the environment, makes decisions, and fully controls the mobility of the wheelchair, which leaves the user totally dependent upon the equipment. While some users may be comfortable with an autonomous wheelchair transportation system, others want to be more involved with the process. It is essential for them to feel in control, while being responsible for both decision-making and motion rather than being a passenger.

A semi-autonomous (SA) wheelchair system comprises hardware equipment similar to autonomous wheelchairs, but the purpose of the system development focuses on maximizing the level of autonomy of users. Thus, the magnitude of assistance is designed to provide just as much as the user really needs. Generally, the type of assistance for a user varies from one to another; therefore, the specification of an SA wheelchair system needs to be determined based on the user's physical capabilities.

A number of research projects have developed various types of semi-autonomous wheelchairs. Borgolte et al. (1998) developed a robotized wheelchair featuring omnidirectional maneuverability to provide an intuitive semi-autonomous control system to assist people with severe or multiple disabilities. Argyros et al. (2002) presented a robotic wheelchair with a semi-autonomous navigation system comprising a panoramic vision system to support the users who have limited motor control of the upper extremities. The Mobile Internet Connected Assistant (MICA) project (Rönnbäck et al., 2006) designed a semi-autonomous wheelchair system in which the user either locally or remotely controls the chair by means of head-held sensors. Despite using a slightly different platform, a power-assisted manual wheelchair,[3] Simpson et al. (2005) demonstrated the first prototype

[3]Based on a manual wheelchair, the rear wheel hubs are replaced with motorized ones that magnify and/or adjust the propulsive force provided by the user.

which attempts to provide the users who have visual impairments with some basic navigation tasks such as collision avoidance.

However, none of these systems provide or enhance perceputual capabilities for the seated user who has severe visual impairment.

1.1.3 MOTIVATION

The goal of our wheelcahir project[4] is to increase the independent mobility of power wheelchair users with severe visual impairment by providing perceptual and navigational capabilities. We assume our typical SA wheelchair user is tactilely and auditorily competent with fine motor control of the upper extremities. In fact, our research efforts have been influenced by a former student who has exactly the disabilities we are mentioning. A collaborative discussion with the student at the initial stage of the project enabled us to elucidate such problems of interest as: collision avoidance, human and obstacle detection, drop-off avoidance, portal navigation, textual-information acquisition, and indoor navigation.

Many large universities provide on-campus curb-to-curb van transportation service to persons with mobility, visual, and other health-related impairments; however, typically no official care attendant service is provided inside a building. Therefore, our focus is primarily on the situations in which the wheelchair users need to be on their own, independently maneuvering in indoor circumstances.

1.2 DESIGN CONCEPT

Since the SA wheelchair system needs to be individualized to the user, our design principle is based upon modularity and flexibility. Further, due to the nature of the problems described in Section 1.1.3 which must be resolved quickly and handled concurrently, the *behavior-based control*

[4]Although this thesis solely comprises the outcome of my own work unless noted otherwise, others have involved with the wheelchair project at the AI center, and we share certain perspectives and the goal. Therefore, the first-person pronoun may be used either in plural or in singular depending on the context.

(BBC) architecture (Arkin, 1998; Brooks, 1991b; Matarić, 1992) is suitable for the base architecture of our design. Utilizing a customized BBC architecture, *behaviors* are defined in order to accomplish given tasks. Behaviors emerge from an agent-environment interaction based on a number of loosely tied processes that run asynchronously and respond to problems in a parallel manner.

In our project, those behaviors are realized to accomplish their tasks through user-machine cooperation. The robotic wheelchair can be realized as a *wheelchair-navigation agent* which consists of two cooperating sub-agents: the *perceptual agent* and the *navigational agent*. The perceptual agent acquires the state of the environment through sensors (*Perceptual Behaviors*), and the navigational agent interprets the state and passes the appropriate information to the user (*Navigational Behaviors*). Finally the user manipulates the wheelchair in combination with his/her own perception and judgment (*Manipulative Behaviors*).

Since we intend to leave maximum control on the user side, the degree of mobility assistance from the wheelchair-navigation agent is set to minimal; that is, the system does not directly intervene in controlling the wheelchair at all. We also assume that our target users are able to generate an internal representation (*cognitive map*) of the familiar environment; that is, they have some knowledge of the starting or current location and the destination as well as some landmarks in between. Based on such assumptions, assisting local navigation, including obstacle avoidance, and landmark matching are more immediately needed tasks than more complicated tasks, such as planning the whole path and navigating autonomously. The main focus of our project is on pursuing perceptual and navigational assistance, thus *semi-autonomous control*.

The base wheelchair is a standard motorized wheelchair (Figure 1.1) that consists of two front caster wheels, two rear motorized wheels, a battery pack, and a controller (joystick). The perceptual navigation system needs to comprise a computer, a collection of sensors, and man-machine interfaces.

Figure 1.1: The base motorized wheelchair (Invacare Nutron® R-32; the photo image retrieved at `http://asburymedical.com`).

1.3 CHAPTER OVERVIEW

This section provides an overview of the rest of the chapters in this thesis.

In Chapter 2, I describe the two types of the behaviors, perceptual and navigational behaviors, in more detail as well as addressing some issues in the BBC architecture. Chapter 3 introduces our behavior design and hardware design. The two custom components: the *Behavior Cell* and *Behavior Network* are presented to define behaviors for wheelchair operations. The hardware design explains the sensing modules and interface module. Chapter 4 presents the implementation of each layer: the behaviors, the communication link between host computer and micro-controller, sensing modules, and interface modules. In Chapter 5, I demonstrate the experimental results and discuss them in detail. Finally in Chapter 6, I conclude the discussion and suggest directions.

CHAPTER 2

BEHAVIORS

In this chapter, I introduce three types of behaviors that are specifically defined for motorized wheelchair operations, namely, perceptual, navigational, and manipulative behaviors. I also address some commonly known issues in designing the Behavior-based control architecture.

2.1 PERCEPTUAL/NAVIGATIONAL BEHAVIORS

The wheelchair-navigation agent is responsible for the perceptual/navigational behaviors that will aid the user's limited perception. Let us consider some of the requirements for the perceptual behaviors.

First, all of the perceptual/navigational behaviors have to be reactive and robust. They must deal with a real, complex, and dynamic environment. Any delay or failure of perceptual responses is unacceptable because it may create severe, possibly fatal, consequences for the user.

Second, each navigational behavior unit stimulates human perception through its man-machine interface (an effecter). Each of the signals from the effecter should be intuitive to learn, distinctive from others, and robust against outside noise. We assume that our target users are competent in both tactile and auditory perception; therefore, the output configuration from the navigational behaviors should convey information (guiding or warning signals) to the user through either tactile or auditory feedback. The details of the interface will be discussed in Section 3.3.

Third, the perceptual/navigational behaviors should be performed independently in a parallel manner. Since each behavior is defined unitarily, a behavior is responsible for achieving a particular task, based on the sensory inputs that are relevant to its internal process. However, this does not mean we should restrict ourselves to designing only primitive behaviors. Capability of interaction

between the behaviors must be provided; some outputs from a behavior may or may not be inputs of another behavior.

Finally, each perceptual/navigational behavior should be designed incrementally (modularly) according to its purpose (goal). Some behaviors should be able to consist of other subordinate behaviors, while others may be simple in structure, nearly hard-wired to the sensors. An arbitration process is required in order for the wheelchair-navigation agent to accommodate the multiple outputs from the various behaviors. Individual behaviors are prioritized based on pre-determined criteria, such as emergency, significance, or user needs.

2.2 MANIPULATIVE BEHAVIORS

The user of the motorized wheelchair is responsible for controlling the actual wheelchair movements. Existing hardware on the motorized wheelchair interprets the user's commands through the joystick and activates the two direct current (DC) motors driving the rear wheels (one motor for each wheel).

The basic scheme for driving the wheelchair is to push the joystick toward the desired direction and to release the joystick to slow down or to stop. The joystick housing is located at the front of the right armrest, and the joystick is equipped with 360 degrees of mobility. The joystick is spring-loaded and automatically returns to the upright (neutral) position when released. Pushing the joystick in a given direction causes the chair to move in that direction (Figure 2.1). The joystick has proportional drive control, meaning that the further it is pushed from the neutral position, the faster the wheelchair moves. To release the joystick causes the wheelchair to slow to a stop. The wheelchair has automatic speed and direction compensation.

Manipulative behaviors will be achieved such that the user of the wheelchair recognizes the state of the environment and manipulates the wheelchair with a single command, such as stopping the wheels, or with a sequence of commands. The recognition of the environment is carried out through devices using tactile (and/or auditory) feedback and through the user's own available senses, such as hearing the background noises and/or acquiring the haptic information via a white

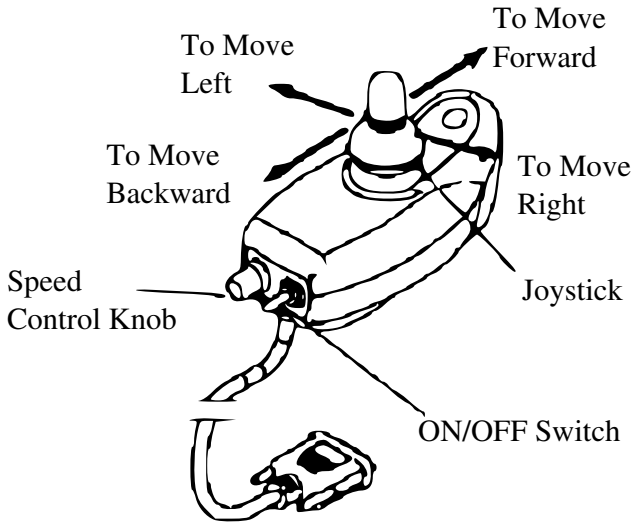


Figure 2.1: Schematic diagram of the joystick and wheelchair operation (**?**).

cane.[1] The seated user would also have to arbitrate if there were a conflict between perceptual behaviors (provided by the wheelchair-navigation agent) and her/his own senses.

2.3 CONTROL ARCHITECTURE ISSUES IN ROBOTICS

In order for the wheelchair navigation to function in real, complex, and unpredictable environments, the perceptual/navigational behaviors must perform reasonably and timely. Reference to discussions of robot control architectures may advocate designing the wheelchair navigation system.

Deliberative architectures that rely on a centralized world model operate by collecting all available sensory data, generating a complete model of its static environment, planning an optimal series of actions based on that model, and executing that plan (Nilsson, 1984; Moravec, 1988, 1989; Laird and Rosenbloom, 1990). However, in such a sense-plan-act paradigm uncertainty in sensing and changes in the environment require frequent re-planning, the cost of which may impede achievement of the goals.

[1] Pranghofer (1996) suggests the seated user to use a slightly longer carbon-fiber cane than one used by an ambulatory person of the same height.

Behavior-Based approaches (Arkin, 1998; Brooks, 1991a, 1992; Matarić, 1991, 1992) provide a framework of architecture on which perceptual processing is distributed across multiple independent agents (behaviors). Each behavior, operating asynchronously and concurrently, is responsible for a particular task based on the sensory data that is relevant to its process.

The Subsumption Architecture (Brooks, 1986, 1991b) represents behaviors as separate layers and organizes them in a bottom-up manner with a pre-prioritized control framework. Behaviors (layers) work on individual goals, and the lowest (most primitive) layer is defined as an augmented finite state machine (AFSM). The Subsumption Architecture arbitrates among behaviors based on explicitly assigning priorities to each behavior; the output from a behavior with the highest priority is predominant and the rest are ignored. Such priority-based arbitration may be effective for choosing among incompatible outputs; however, due to the absence of the ability to store internal state dynamically, this priority-based arbitration is incapable of either providing any decisions (outputs) that satisfy multiple goals simultaneously or solving complex problems that contain temporal sequences.

The Behavior-Based Control (BBC) architecture is based upon the Subsumption Architecture with a capability of containing an internal state and constructing a flexible control framework. Such a capability enables the system to employ various forms of distributed representations implemented within the behavior networks (Matarić, 1997). An abstract behavior representation also allows the BBC architecture to generate and maintain complex plan-like strategies without redesign and recompilation processes (Nicolescu and Matarić, 2000b,a). Yet having such advantages of flexibility, the methodology of behavior arbitration is one of the challenges of the BBC architecture.

In order to provide a multiple-goal-oriented decision in the BBC architecture, it is essential for a behavior arbitration mechanism to obtain command fusion functionalities. Command fusion combines the commands (outputs) from individual behaviors so that the decisions may satisfy multiple goals while preserving the reactivity and modularity of the system. Several command fusion approaches have been proposed by combining with a variety of algorithms: the activating spreading schema, centralized voting system from distributed behaviors, fuzzy logic control, immunological

network evolved by a genetic algorithm, hybrid automata, and neural network (Maes, 1989; Payton et al., 1992; Yen and Pfluger, 1995; Ishiguro et al., 1996; Kondo et al., 1998; Egerstedt, 2000; Na and Oh, 2003).

CHAPTER 3

DESIGN

In this section, I present an overview of the behavior-based architecture with *Behavior Cell* and *Behavior Network* components, and the hardware design, which consists of a sensor system and feedback module.

3.1 BEHAVIOR ARCHITECTURE OVERVIEW

3.1.1 BEHAVIOR CELL

I present a unit of the behavioral structure, a *Behavior Cell*, which is based upon the BBC architecture with an extended input/output feature. A Behavior Cell consists of an input/output (I/O) component, a behavioral function component, and an internal storage component (Figure 3.1). It structurally resembles an artificial neuron; however, it has a logical gate in addition to widely extended functions such that the innervation link between cells can run by both Boolean and numeric means.

A Behavior Cell does not have to employ all components; it may or may not consist of the behavioral function and the internal storage components depending upon what features it needs. Behavior cells communicate with other Behavior Cells, sensors, and effectors through their I/O components.

The I/O component consists of a subset of the following eight I/O ports: Port-EB, excitatory inputs; Port-IB, inhibitory inputs; Port-DI, sensory/behavioral inputs; Port-RS, a reset signal; Port-IS, an innervation signal; Port-AB, an activation output; Port-EO, an effect output; and Port-AO, actuator outputs. The excitatory and inhibitory inputs are linked to the corresponding behaviors' activation output ports. When any activation (inhibition) conditions are met, the behavior

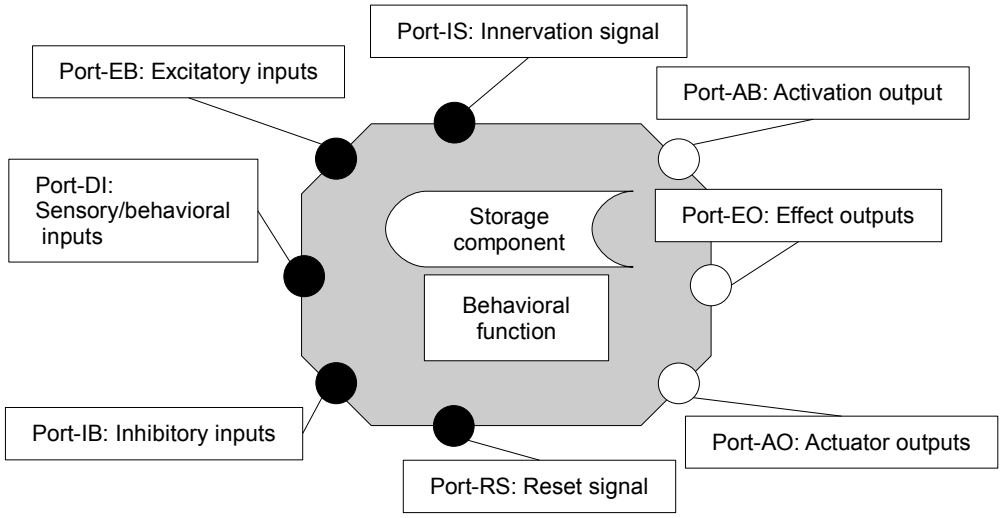


Figure 3.1: The basic structure of a Behavior Cell.

is activated (deactivated). This architecture allows both Port-EB and Port-IB to specify activation (inhibition) conditions by using logical expressions (Boolean algebraic functions), such as $Activation = (Activation1 \wedge Activation2) \vee \neg Activation3$.

Port-DI takes various types of data inputs from sensors or other behaviors (effect outputs). When Port-IS receives an innervation signal from outside or from Port-RS, the behavior checks or sends its inputs and outputs. If Port-RS receives a reset signal from the system, the behavior will clear all dynamic contents of the storage component and activate Port-IS. Port-AB contains an activation value (binary) that is linked to the value of Port-EB. Port-EO contains an effect value that is derived from the behavioral function. If the behavior is connected to its effector(s), Port-AO sends actuator outputs to them. The generic features of each port are summarized in Table 3.1.

The behavioral function component provides a flexible activation/computation functionality. The function can be various types, such as algebraic sum, sigmoid, Gaussian, and look-up-table, as well as a simple by-pass function (e.g. a direct link between inputs and outputs). More complicated functionalities, such as fuzzy logic inference operators or artificial neural networks, could also be implemented, but I do not discuss them in this thesis.

Table 3.1: The generic features of input/output ports of a Behavior Cell

Port ID	Name	Feature
Port-EB	Excitatory Inputs	Binary value is derived from a predefined logical expression and the connected input values.
Port-IB	Inhibitory Inputs	Binary value is derived from a predefined logical expression and the connected inputs values.
Port-DI	Sensory/Behavioral Inputs	Input data from sensors and/or other behaviors. A various types of the data (including arrays) are allowed.
Port-IS	Innervation Signal	Innervates the Behavior Cell.
Port-RS	Reset Signal	Clear temporal memory.
Port-AB	Activation Output	Binary value linked to Port-EB and Port-IB
Port-EO	Effect Outputs	Outputs (the result of computation)
Port-AO	Actuator Outputs	Action control to effectors.

The storage component provides a storing capability of the current state onto its dynamic data, which enables the behavior to achieve goals that contain temporal sequences. It may also contain internal static data which all instantiated behaviors can share and refer to, as well as individual constant data that a behavior utilizes as permanent reference information, such as threshold values or look-up tables.

The activation/computation process performed by a Behavior Cell is as follows:

1. When Initialization/Reset input (Port-RS) is activated, it refreshes the internal dynamic memory and innerves Innervation Input (Port-IS).

2. When Innervation Input receives a signal, check the value of Effect Inputs (Port-EB). If true, set Activation Output (Port-AB) value to 1 (true) and go to the next step, otherwise return.

3. Check the value of Inhibitory Inputs (Port-IB) to see whether the behavior is inhibited. If false, go to the next step, otherwise set Activation Output (Port-AB) to 0 (false) and return.

4. *In case of using Port-EO:* Using the information from Sensors/Behavior Inputs (Port-DI), derive the return value from the behavioral function and write this value to Effect Output (Port-EO) and return. Store the necessary data in the internal memory if so designed.

5. *In case of using Port-AO:* Similar to (4), derive the return action commands from the behavioral function and send the commands to the effectors via Action Outputs (Port-AO) and return. Store the necessary data in the internal memory if so designed.

3.1.2 BEHAVIOR NETWORK

Similar to other Behavior-Based architectures (for instance, Nicolescu and Matarić, 2002), my approach also enables behaviors to consist of other behaviors. Such behaviors are implemented based on a subset of corresponding behaviors, thus represented as a *Behavior Network*. The types of relationships between a Behavior Network and its corresponding behaviors may vary; they can be hierarchical or interdependent. In a Behavior Network, behaviors communicate with each other through their port-to-port links, and precondition dependence characterizes the links; thus, the activation of a behavior is dependent on its pre-conditional links.

An individual Behavior Cell can connect to multiple Behavior Cells, and similarly, multiple Behavior Cells can be linked to a single Behavior Cell. This multiple-connectivity allows a Behavior Cell to be a member of multiple Behavior Networks, which makes component-like behaviors, such as interface to the sensors, reusable. Containing multiple task-oriented and reactive behaviors (*functional behaviors*) enables a Behavior Network to accomplish various tasks, such as command arbitration, learning, and planning, while asynchronously performing tasks within the distributed architecture. The bottom line of the Behavior-Based philosophy, the distributed architecture, is preserved such that the behaviors are relatively simple, organized into modules, and performed in a parallel fashion.

A Behavior Network must also behave as structurally equivalent as a Behavior Cell when observed from outside. In order to do so, each Behavior Network must contain a specific type of Behavior Cell (*I/O cell*), which accomplishes the tasks related to input/output communication and activation sequence inside of the Behavior Network. Figure 3.2 depicts a generic Behavior Network that consists of I/O cells and functional behaviors.

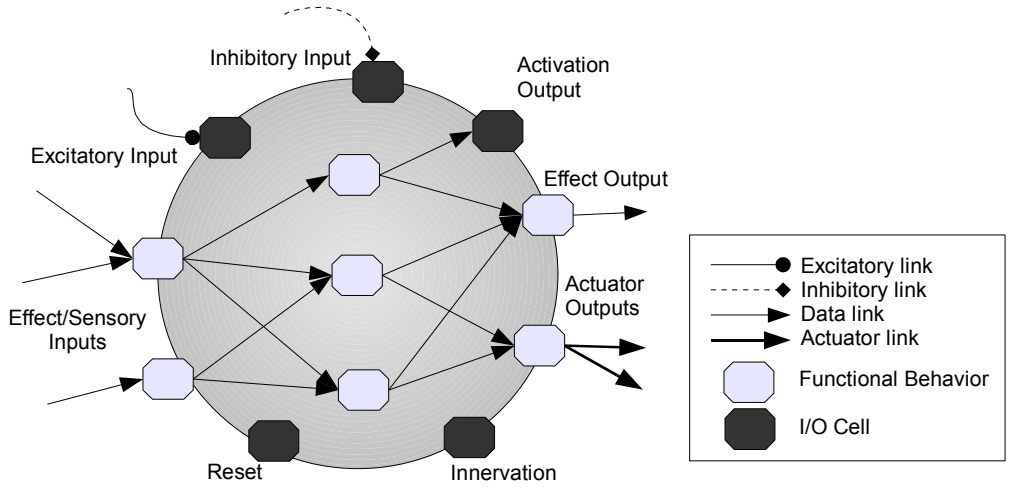
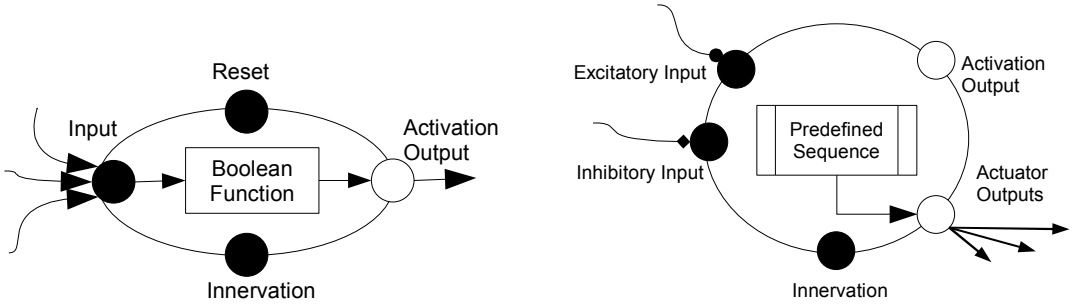


Figure 3.2: Schematic diagram of a Behavior Network.

The I/O cells are categorized in two types: Boolean I/O cells that exchange Boolean signals and activation I/O cells that control sequential activation in the Behavior Network. Figure 3.3a illustrates a generic Boolean I/O cell that consists of Port-EB (excitatory inputs), Port-RS (reset signal), Port-IS (innervation signal), Port-AB (activation output), and a Boolean algebraic function. The Boolean algebraic function can employ various symbolic logic expressions, and the result from the function is directly connected to the activation output (Port-AB).

Activation I/O cells are responsible for the sequence of innervation/reset functionalities in the Behavior Network. An activation I/O cell consists of an excitatory input, an inhibitory input, an innervation input, an activation output, action outputs, and a storage component that contains a predefined activation sequence of the behaviors (Figure 3.3b). The activation I/O cell innerves the reset/innervation ports of the behaviors that belong to the Behavior Network according to the sequence stored on the storage component.

The functional behaviors deal with data and/or actuator communication. Connections between functional behaviors consist of excitatory links (between Port-AB and Port-EB), inhibitory links (between Port-AB and Port-IB), (sensory and/or behavioral) data links (between Port-EO and Port-DI), and actuator links (between Port-AO and effectors). A functional behavior in a Behavior Network can also be another Behavior Network.



(a) Boolean I/O cell (b) Activation I/O cell

Figure 3.3: Schematic diagrams of Boolean I/O cell (a) and Activation I/O cell (b).

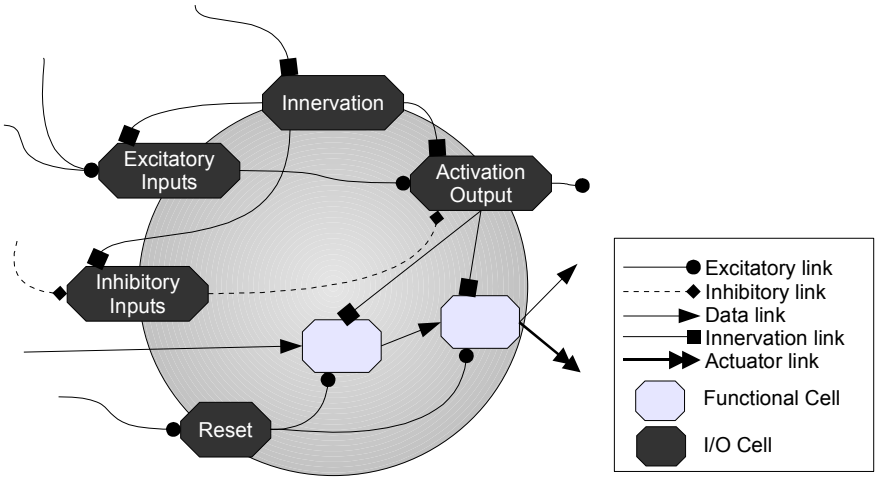


Figure 3.4: Schematic diagram of I/O cell links in a Behavior Network.

However, the activation process of a Behavior Network differs from the one of a Behavior Cell. An Innervation cell is first innerved and it innervates the Excitatory Inputs cell, Inhibitory Inputs cell, and Activation Output cell in this order. If the value of Port-AO (activation output) of an Activation Output cell is true, it will innervate the functional behaviors in a predefined order otherwise the whole process will return; thus the Behavior Network will be deactivated (Figure 3.4).

3.2 SENSING MODULES

In this research, we design and employ a sensor system which consists of two types of modules: a stationary ranging module and a motorized vision module. The stationary ranging module is attached to the wheelchair and constantly measures distances to any objects that are within the scope of the sensors. The motorized vision module is designed to have the ability to pan/tilt and to acquire depth data as well as scenery images, so that the perceptual behaviors can utilize more detailed information.

3.2.1 STATIONARY RANGING MODULE

Employing ultrasonic range sensors (sonar sensors) for the stationary ranging module appears appropriate because sonar sensors commonly used in robotics research have an effective range (typically between a few centimeters and 6 m) that is suitable for indoor navigation. Sonar sensors are also inexpensive and easy to install. Devantech SRF08 was chosen for its cost-perfomance and size efficiency. The specification of the SRF08 and its beam pattern are shown in Table 3.2 and Figure 3.5, respectively.[1] According to the specification, the ultrasound wave propagation profile (beam width) of an SRF08 is roughly 45°.

It is common to employ an array of sensors in order to obtain a wide coverage area around a mobile robot (specifically for omni-directional robots). However, the wheelchair we employed is kinetically non-holonomic;[2] that is, although it could physically move laterally, it is not equipped with the mechanism to control such movement. Based on the kinetic property of the wheelchair and the beam pattern of an SRF08, six sonar sensors are attached to the front, and two sonar sensors are placed to the rear, as illustrated in Figure 3.6a such that the front side of the wheelchair has more coverage. Figure 3.6b depicts the sonar installation at Front Right and Front-side Right location.

The eight sonar sensors circumferentially attached to the wheelchair are coupled into four opposing pairs: Front Left/Rear Right, Front Right/Rear Left, Front-Side Right/Side Left, and Side

[1] Retrieved at `http://www.acroname.com`

[2] The controllable degrees of freedom is less than the total degrees of freedom.

Table 3.2: Ultrasonic sensor SRF08: Specification

Voltage	5 V
Current	30 mA (typical)
Frequency	40 kHz
Maximum Range	6 m (10 feet)
Minimum Range	3 cm (approx. one inch)
Sensitivity	Detect a 3cm diameter stick at > 2 m
Input Trigger	10 µs Min. TTL level pulse
Echo Pulse	Positive TTL level signal, width proportional to range.
Weight	0.4 Oz.
Size	1.75” w x 0.625” h x 0.5” d

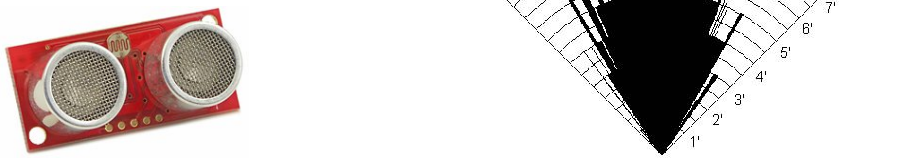


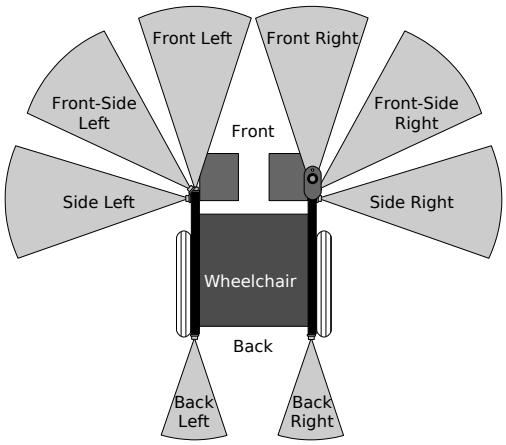
Figure 3.5: Devantech SRF08 (left) and its beam pattern (right).

Right/Front-Side Left. The sensor controller is implemented to fire pairs at an appropriate interval to minimize the crosstalk problem.

3.2.2 Motorized Vision Module

Since typical ultrasonic sensors only measure the distance to the closest object within range, certain information (e.g. the precise location and orientation of a door-opening) is difficult to obtain from them.

Applying an array of sensors is one way to tackle this problem; however, one would need a special treatment to avoid an erroneous phenomenon, *crosstalk*, in which one sonar receives



(a) Schematic diagram of coverage areas of sonar sensors.

(b) Sonar sensors installation on Front Right/Front-side Right photo: Michael A. Covington.

Figure 3.6: Installation of the stationary ranging modules.

the ultrasound waves emitted from another sonar. Some attempts to address this issue have been proposed by alternating firing patterns (Borenstein and Koren, 1995) or employing double pulsing patterns (Kleeman, 1999). While these techniques might work well do detect a small object within a relatively open space, it would still be problematic to use a sonar array within a very narrow area (such as a hall way inside the AI center in the University of Georgia) simply because all objects are too close to the wheelchair.

In contrast, a vision based ranging system, specifically a laser rangefinder system, is suitable to acquire such geometric information because of its accuracy in determining distance within a reasonable amount of time.

LASER RANGING MODULE

A variety of commercial laser rangefinders are availble with ease of installation; however, there are a few disadvantages: Some have limited scanning range or area (viewing angles), some are too large to mount on wheelchairs, and all of the pre-made laser rangefinders are currently quite expensive.

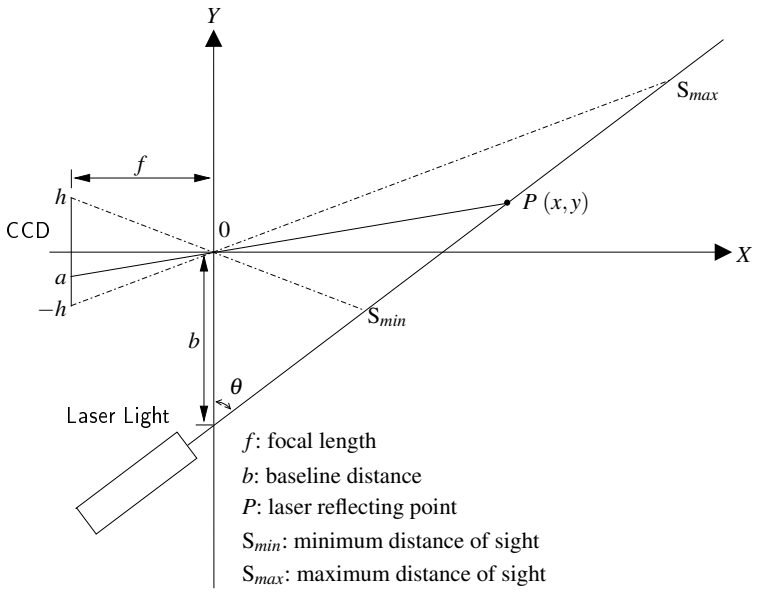


Figure 3.7: Camera-centered triangulation geometry (adapted from Jain et al. (1995)).

An alternate way to resolve the problem is to create a custom laser rangefinder. With an inexpensive camera and a laser pointer, a simple laser rangefinder can be configured by utilizing the triangulation principle and image processing methodology (range imaging using structured lighting). Figure 3.7 illustrates that the coordinate (x,y) of the laser reflecting point P can be calculated from observed point a in the acquired image, the focal length (f), the baseline distance (b) between the camera center and the laser light source, and the projection angle θ by the following formula:

$$(x,y) = \frac{b}{f\cot\theta - a}(f,a). \tag{3.1}$$

The minimum and maximum distances (depths) that can be measured by this method are illustrated as S_{min} and S_{max} respectively.

Our approach is to employ a CCD camera (a webcam) as a low cost optical sensor in combination with a laser line generator emitting a horizontal laser line such that depth mapping is extended from a single point to a horizontal cross-sectional image in a view.

An advantage of using a custom-made laser rangefinder is that we are able to use the camera for other purposes as well. A few examples of computer vision techniques that can be useful for

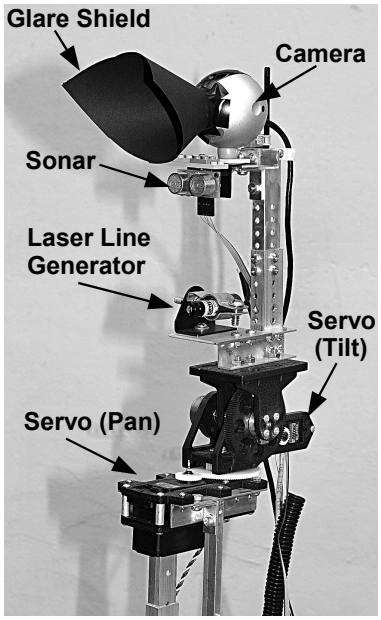


Figure 3.8: The motorized vision module prototype (photo: Michael A. Covington).

wheelchair operations include door frame detection, human-face detection, and textual information extraction.

A typical inexpensive CCD camera generally has a limited field of view which is not enough to cover the surrounding environment of the wheelchair. To overcome this drawback, we designed the laser range finder to be motorized by mounting it on a pan/tilt servo device.

Figure 3.8 illustrates the prototype of the motorized vision module. It consists of an off-the-shelf webcam (Logitech QuickCam® Pro 4000), a laser line generator module (distributed by World Technology Inc.), a pan/tilt module (ServoCity™ SPG785 and SPT400), and a microcontroller (Rowley Associates Ltd. CrossFire® LPC-2138) that accepts signals from the host computer (PC) and controls the servo motors and the laser module. A hardware specification list is shown in Table 3.3.

ADD-ON SONAR

Although this module is vision based, we also add a sonar sensor (Devantech SRF08) to it for two reasons. The first reason is to acquire the depth information in a complementary manner when the

Table 3.3: Hardware specification of the motorized vision module.

Hardware Equipment	Product name: Specification
Camera	Logitech Quickcam® Pro 4000 Optical Sensor Type: CCD Image Capture Resolution: 640 x 480 (Video) Frame Rate: 15 fps (640 x 480) Effective Focal Length: 4.5 mm Field of View: 42° Interface: USB
Optical filter	Celestron™ LPR-94126A
Laser module	Laser line generator (Class II, output power: 10 mW, wavelength: 635 nm, fan angle: 45°)
Pan/tilt system	(Pan) SPG785 Gearbox: Hitech HS-785HB (0–420°) (Tilt) SPT400 Tilt System: Hitech HS-425BB (0–135°)
Sensor Micro controller	Rowley Associates Ltd. CrossFire® LPC-2138 Evaluation kit

laser module is not in use. The wavelength of the laser light is 635 nm (visible red), so continuous usage of the laser module would distract other people and attract unwanted attention. Therefore, the laser module is activated only when it is really needed.

The other reason is to make the module incorporate more redundancy by acquiring the information from the different sensory channels when both sensors are in use. One technical difficulty of our laser rangefinder is that the laser rangefinder would not be able to find correct distance if the target object were made of highly transparent or reflective materials such as glassware or mirrors. Comparing the depth data from both sensory channels enables the system to detect such occasions.

OTHER VISION PROCESSING

When the vision module is not in the laser ranging mode, it is also possible to utilize the acquired scenery images for other vision processing. In this thesis, we only use basic machine vision techniques; however, this design would allow us to exploit a variety of computer vision algorithms to achieve more sophisticated tasks.

3.3 INTERFACE MODULES

Communication between the user and the SA wheelchair via a versatile and robust man-machine interface plays an important role in achieving successful transportation activities. For those who have visual impairment, the sensory channels available for communication are limited to audible or tactile perception.

At first glance, auditory communication such as voice navigation or warning beepers may seem to be the easiest way to relay information to users. However, making noise might be socially inappropriate in certain situations. The auditory signals could also be overwhelmed by strong background noise in the environment (such as a fire alarm). Furthermore, additional sound sources are undesirable with respect to preserving the user's resource of attention since people with severe visual impairment rely extensively on their auditory system to collect environmental information.

Human tactile perception is robust and suitable for multimodal sensing (van Erp and van Veen, 2004). While people with blindness already use their haptic sensory system (e.g. using a cane), the somatic sensory system of another area of skin can still be used. Since the user passively receives most information, an effective form of tactual information should comprise vibrations (*vibrotactile signals*).

The outputs from the perceptual behaviors should consist of various kinds of vibrotactile signals conveying information via human tactile perception. Previous work evaluating a variety of tactile feedback systems has shown substantial potential (Geldard, 1975; Gemperle et al., 2001; Tan and Pentland, 1997; Tan et al., 2000; Traylor and Tan, 2002; Zelek et al., 2003). For example, Tan et al. (2000) have derived several benefits by studying the *cutaneous sensory saltation* in order to develop a haptic interface. The cutaneous sensory saltation is a tactile illusion discovered in the 1970's in the Cutaneous Research Laboratory at Princeton University (the word "saltation" is Latin for "jumping") (Geldard and Sherrick, 1972; Geldard, 1975). When a series of tactile stimuli are presented to different loci with a short interstimuli interval, perceived locations are shifted and distributed between the first and last stimulator sites. Although the occurence of this illusion is robust, the nature of mechanism of the cutaneous saltation is still under debate (Kilgard and Merzenich,

1995; Eimer and Forster, 2005). The sensory saltation may provide directional information that requires an otherwise highly intuitive grasp. A vibrotactile display can be established by relatively simple hardware configurations, such as an array of tactors, and can be perceived on many body sites, such as the fingertips and the back (Cholewiak and Collins, 2000).

3.3.1 VIBROTACTILE GLOVE

It is concluded that the vibrotactile display should reasonably be located on the hand because we have defined that our users can operate the SA wheelchair with little or no problems using their hand. In contrast, the fingertips would not be a convenient place for using a vibrotactile display because many individuals with severe visual impairment already use their hands to control a joystick, hold a cane, or read Braille signs using their fingertips. Further, a display mounted on the back seat would not be ideal because the users would frequently need to adjust themselves to put their back against the seat.

Although the palm side of the hand is much more sensitive in tactile perception than the back side of the hand, volar flexion during the joystick operations makes it difficult to design a robust vibrotactile display which guarantees consistent pressure of a vibrotactor to the skin. In contrast, the back side of the hand maintains its flatness better during joystick operation; thus, the back side of the hand has been chosen as a locus of the vibrotactor.

Since our purpose is to convey translated visuospatial information to the user, an array of vibrotactors should obviously be employed rather than a single vibrotactor. The choice of an appropriate size of array depends upon the area of the locus, the size of each vibrotactor, and the tactile sensitivity of the corresponding cutaneous area.

Naturally, a suitable form of carrying the vibrotactile display is a glove (*Vibrotactile Glove*). The Vibrotactile Glove consists of the vibrotactile array inside the glove so as to face the back side of the hand and a motor-array controller, which generates and sends signals to the vibrotactors.

3.3.2 VIBROTACTILE ARRAY

Figure 3.9a shows the vibrotactile array which consists of vibrotactors, each of which is directly mapped to correspond to the ultrasonic sensors as shown in Figure 3.6a. I used an ordinary baseball glove and cut it on the little-finger side to make it easier to install the vibrotactor array. Each vibrotactor contains a miniature vibrating electric disk motor attached with a connector pin (Figure 3.9b) and is attached inside of the glove using Velcro®. The vibrating motor is thin, light-weight, and characterized by an off-centered weight causing lateral vibration when the specified voltage (3 V) is applied. It is also fully self-contained in a housing shell, thus no moving parts are visible. A vibrotactor generates a vibrotactile stimulus consisting of a rapid vibration lateral to the skin surface. Although the terminals of the vibrotactors are fully sealed by epoxy adhesive coating, finger cots are attached on the surface that make contact with human skin in order to maintain constant contact pressure between the vibrotactors and the skin surface and to minimize the effects of perspiration, body oils, hand cream, and so forth.

3.3.3 MOTOR-ARRAY CONTROLLER

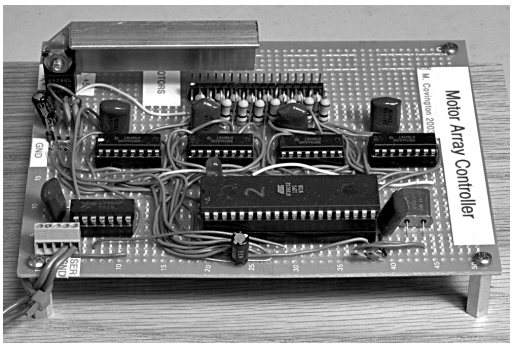
We implemented a motor-array controller on a single-board computer to control an array of vibrating disk motors. The controller receives control signals from a PC by a serial port and executes various patterns of vibration in an array of up to 16 motors. Each time the motor-array controller receives a control signal from the host computer, a vibration pattern is generated by an assembly-language program running on a CPU of the controller.

The prototype of the motor-array controller is shown in Figure 3.9c. The CPU is an Atmel AT89C52, a flash-memory derivative of the Intel 8051 (MCS/51) with the same instruction set. Serial input is received through an MC1489A or equivalent RS-232 line receiver. Two of the CPU's output ports (P0 and P1) are coupled to the motors through SN754410 motor control ICs.

Although the specification of the motor-array controller was determined by me, actual board design and implementation including programming in assembly language to run Atmel AT89C52

(a) The layout of the vibrotactor array on the back side of the glove.

(b) The vibrotactor consisting of a vibrating disk motor and connector pin.



(c) The prototype of the motor-array controller.

Figure 3.9: The tactile feedback module (photo: Michael A. Covington).

were solely done by Dr. Covington. The circuit diagram and pinouts diagram of the motor-array controller are shown in Appendix A.

3.4 System Architecture

An overview of the system architecture is depicted in Figure 3.10 (left). It illusrates a layered structure which constructs the hardware layer (camera, servos, laser module, and the Vibrotactile Glove), the microcontroller unit (MCU) layer for controlling sensors and the Vibrotactile motor array, the communication layer, and the application layer. Both the camera and the sensor controller are connected to the host computer with USB cables. The motor-array controller receives control codes from the host computer via a serial port. In the diagram, the thin arrows represent the flow of data, and the thick arrows represent the flow of actuator commands. The overview of our SA wheelchair prototype is shown in Figure 3.10 (right).

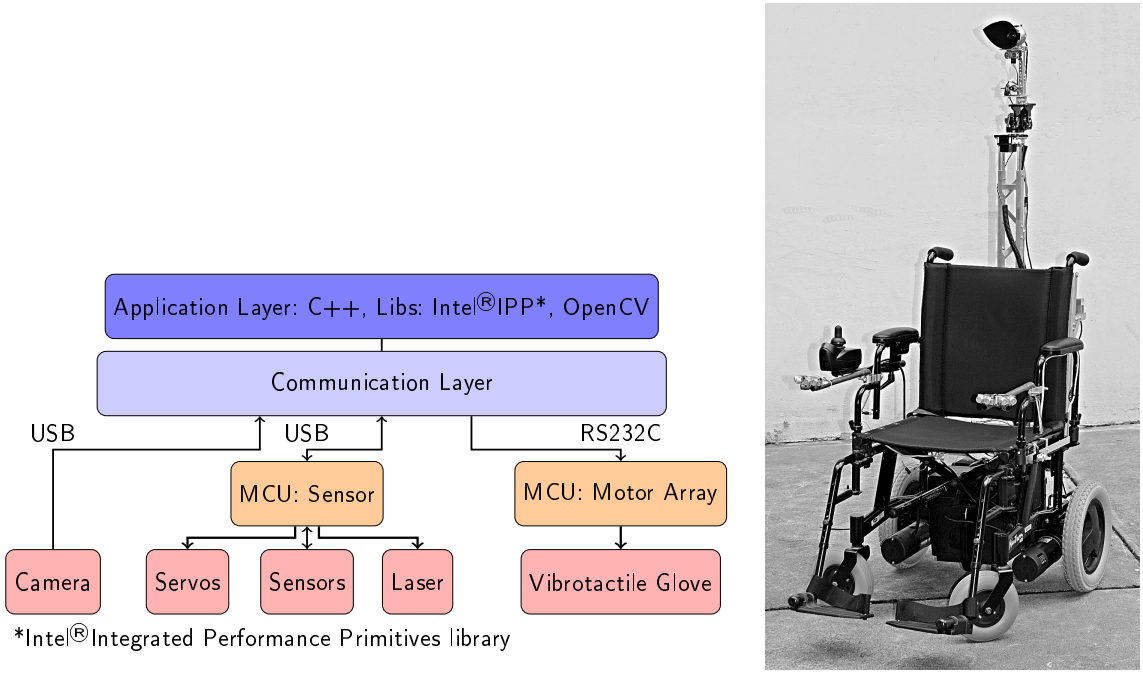


Figure 3.10: System architecture (left) and the SA wheelchair prototype (right, photo: Michael A. Covington).

CHAPTER 4

IMPLEMENTATION

In this chapter, I describe the implementation of the hardware modules and behaviors.

4.1 IMAGE ANALYSIS TO DETECT LASER REFLECTION LINE

This section describes the primarily image analysis to detect the laser reflecting line for the motorized vision module.

4.1.1 LINE ANALYSIS

An initial attempt to apply general edge detection methods, such as Sobel, Canny, and Laplacian masks, over the whole image did not succeed. This is because too many irrelevant points are detected, and the process to eliminate the irrelevant points will cost much computation for using many computer vision processes. Seeking a computationally cheap method, the following domain specific constraints are applied to the vision processing.

1. Each column contains at most one point that reflects the laser light (limit one point per column).

2. The point is red (or looks reddish to human eyes) with sharp edge.

3. Each point consists of a few to several pixels with continuous level of red intensity.

At first glance, it may appear that the simple thresholding of red intensity would be a solution; however, it is not the case because the absolute level of red intensity of the points differs depending on the background, reflections from other light sources, and the distance from the camera.

Figure 4.1: Sampling image array selection

Figure 4.1 illustrates some samples of pixel arrays (100 x 1 pixels) selected from various regions that include the red laser reflecting points over various background levels. Each sample attaches a letter on top of it, denoting D: dark, V: vague, B: bright, N: noisy, and P: plain background respectively.

From Figure 4.2 through Figure 4.6, the histograms of the color channels (i.e., the intensity of Red, Green, and Blue, and the degree of Hue[1]) of the sample pixel arrays are shown as well as the transposed images above them, with illustrating down arrows that locate the laser reflecting points determined by human eyes. In the histograms, Red, Green, and Blue ranging from 0 to 255 use the left Y (Y1) axis, and Hue ranging from 0 to 360° uses the right Y (Y2) axis. The X axis denotes the location of the pixel in the image.

DARK BACKGROUND WITH STRONG REFLECTION (FIGURE 4.2)

This is an ideal condition, and only the laser reflecting point reveals acute peak intensity of Red and high degree of Hue, and other parts show nearly zero because of its dark background.

[1]One of the three main attributes of perceived color, in addition to lightness and chroma or colorfulness. Hue characterizes a color described with names such as “red,” “yellow,” etc.

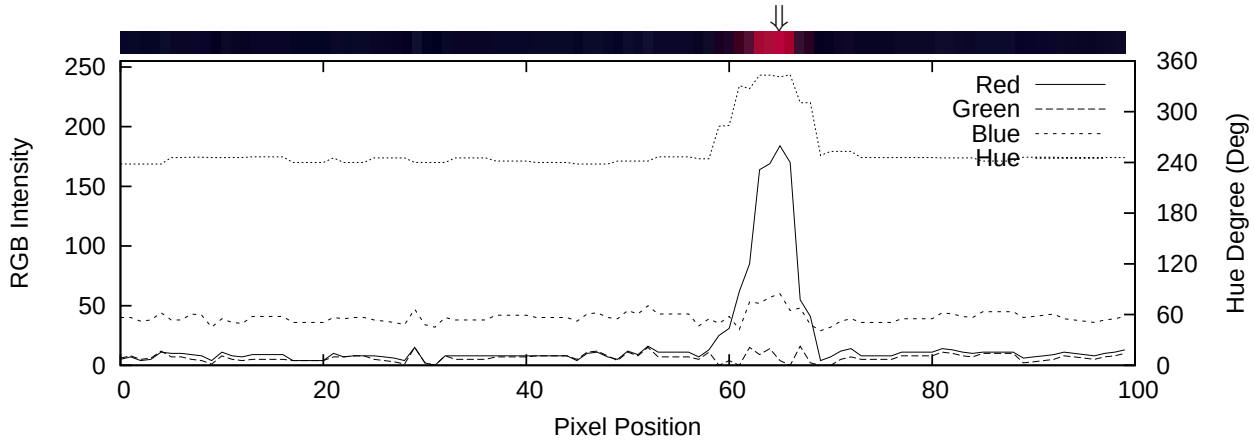


Figure 4.2: Color channels in the dark region (D).

VAGUE REGION (FIGURE 4.3)

All pixels are very dark, and the peak of Red on the reflecting point is also quite vague even for human eyes although the peak Hue is distinctive. Note that the intensity of Blue is stronger than Red at most of the pixels because of the condition of the background (perhaps a bluish object is placed in the shelf).

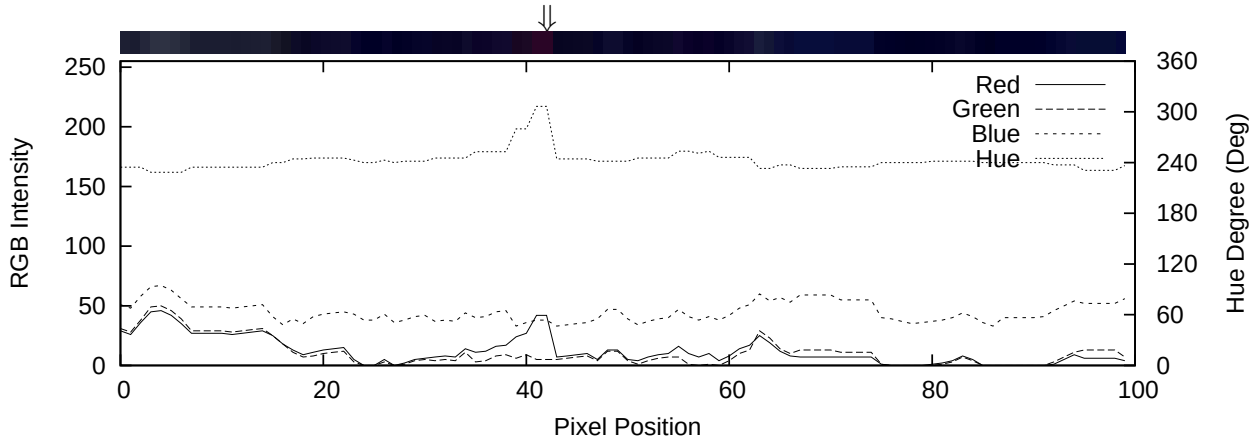


Figure 4.3: Color channels in the vague region (V)

BRIGHT REGION (FIGURE 4.4)

While all RGB color channels maintain high intensity, the reflecting point is clear because of the higher peak of Red and the low peak of Hue. Note that the degree of "*red*" defined by Hue will increase if the value (degree) comes close to either 0° or 360°.

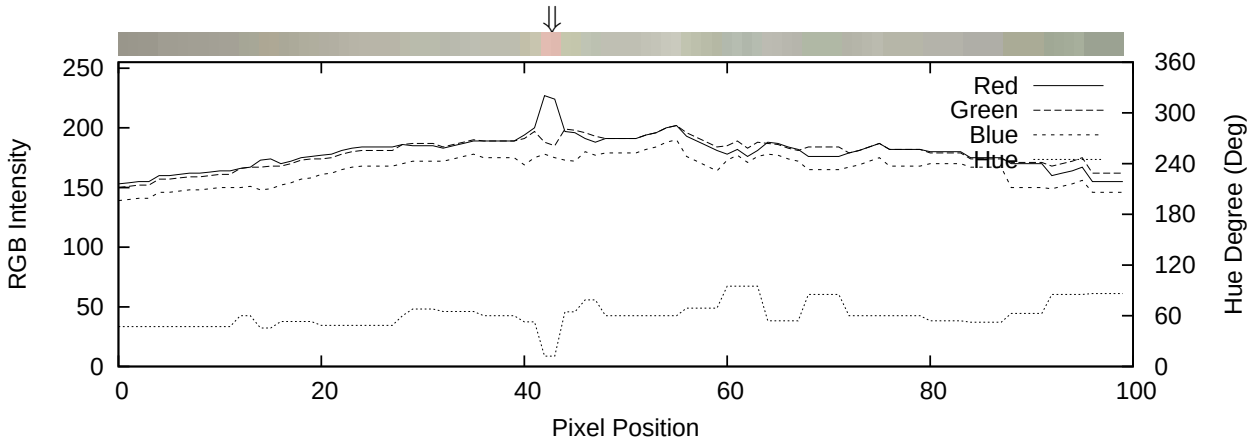


Figure 4.4: Color channels in the bright region (B)

NOISY BACKGROUND (FIGURE 4.5)

This has a noisy background, and it is most challenging to identify the reflecting point by observing the value of the color channels than others. However, it seems the pixels on the reflecting point have larger intensity difference between Red and Green than other parts, which is also consistent with the other figures. The degree of Hue is also a revealing high peak.

PLANE (FIGURE 4.6)

The plane region has similar color distribution to the one of bright regions in RGB colorspace. The Hue shows a high degree of value at the reflecting point although there are some pixels on which Hue cannot be defined.

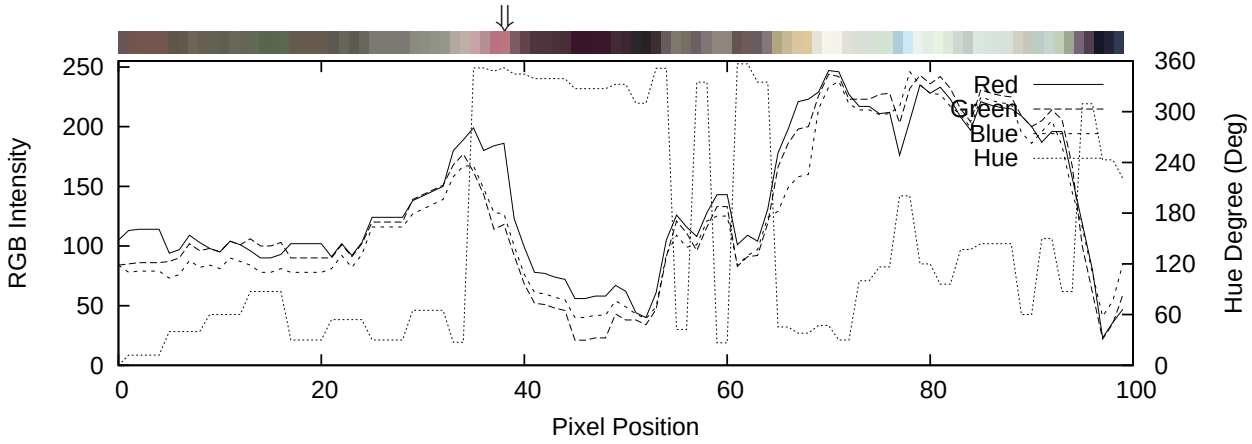


Figure 4.5: Color channels in the noisy region (N)

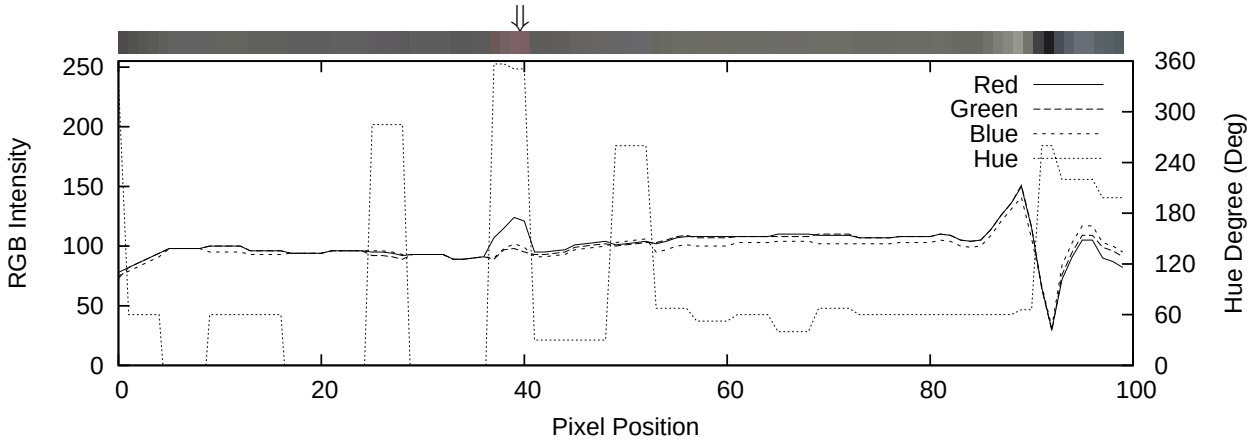


Figure 4.6: Color channels in the plane region (P)

4.1.2 COLUMN SEGMENTATION METHOD—DETECTING EDGE POINT PER COLUMN

Close inspection of the color channels of those regions suggests that the following algorithm (I call it *column segmentation*) appears to work well to detect the laser reflecting red line. Column segmentation consists of the following processes: creating an intermediate image (obtain *Irg*), finding the local minimum of the second derivative of *Irg* (d^2Irg_{min}), checking the redness of the local minimum, and confirming the continuity of the R value at the local minimum. The histograms of

Irg, d^2Irg, and *Redness* of each region are shown in Figure 4.7 through Figure 4.11. The overview of the process of column segmentation is illustrated in Algorithm 1.

Creating an intermediate image: Intensity level subtracted Green from Red (*Irg*) in RGB colorspace in order to reduce the noise of the background. *Irg* is given by the following function.

$$Irg = \begin{cases} Ir - Ig, & \text{if } Ir \geq Ig \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where *Ir* and *Ig* are the intensity of Red and Green respectively.

Find the local minimum of the second derivative of *Irg*: In order to find a single pixel that represents the reflecting point, I use the local minimum of the second derivative of *Irg* (denoted as d^2Irg) as an indicator. The convolution mask, $G_x : \begin{pmatrix} 1 & -2 & 1 \end{pmatrix}$ is applied, and the operation for each pixel will be:

$$\begin{aligned} d^2Irg &= \frac{\partial^2 Irg}{\partial^2 x} \\ &= Irg_{i-1} - 2Irg_i + Irg_{i+1}. \end{aligned}$$

Check the *Redness* of the point: Instead of using Hue degree (0°to 360°), I defined a simplified function, *Redness*, in which the degree of "redness" is distributed between 0 (no red) and 1 (completely red). *Redness* is calculated by the following function. The point is valid if *Redness* is greater than or equal to 0.5.

$$Redness = \begin{cases} 1 - \dfrac{Hue}{120}, & \text{if } Hue \leq 120 \\ 1 - \dfrac{Hue - 240}{120}, & \text{if } Hue \geq 240 \cdot \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Confirm the continuity of Red intensity *redGradient*: When the candidate point is identified, make sure whether the Red intensity level in neighboring pixels is continuous and not within

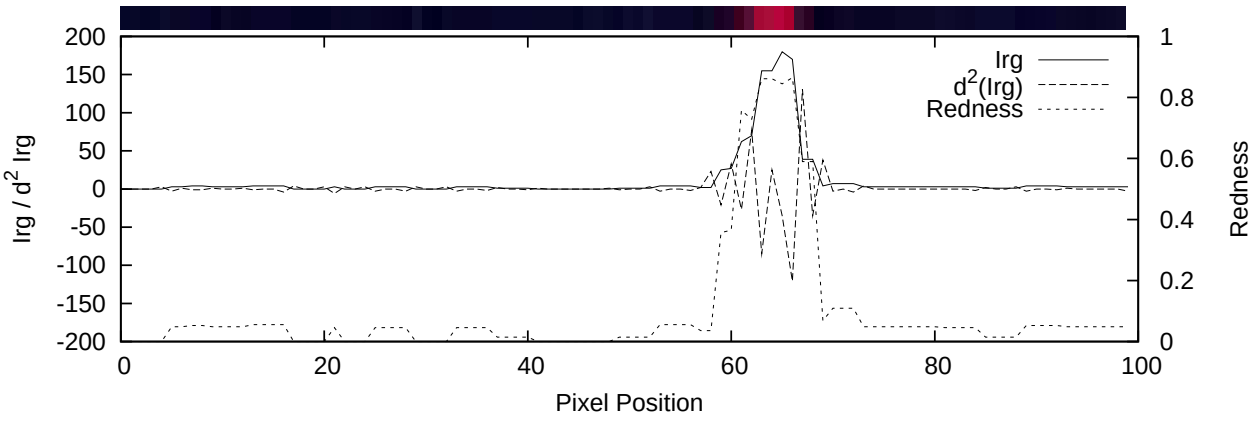


Figure 4.7: Irg, d^2Irg, and Redness in the dark region.

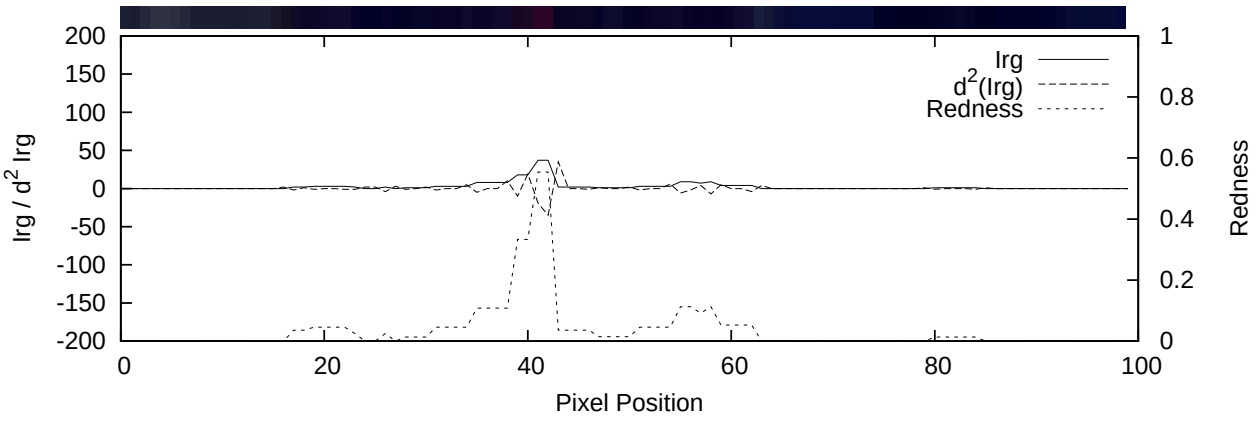


Figure 4.8: Irg, d^2Irg, and Redness in the vague region.

a gradual change. The degree of gradient of Red intensity (*redGradient*) is measured by formula (4.3). Only if the redGradient is less than or equal to 0.15, is the point valid.

$$redGradient_i = \frac{|Irg_{i-1} - Irg_i|}{255} \tag{4.3}$$

4.1.3 Performance of Column Segmentation Method

Figure 4.12 illustrates the detected points using the column segmentation method, which overlapped onto the original image. Column segmentation is simple and can effectively detect the laser

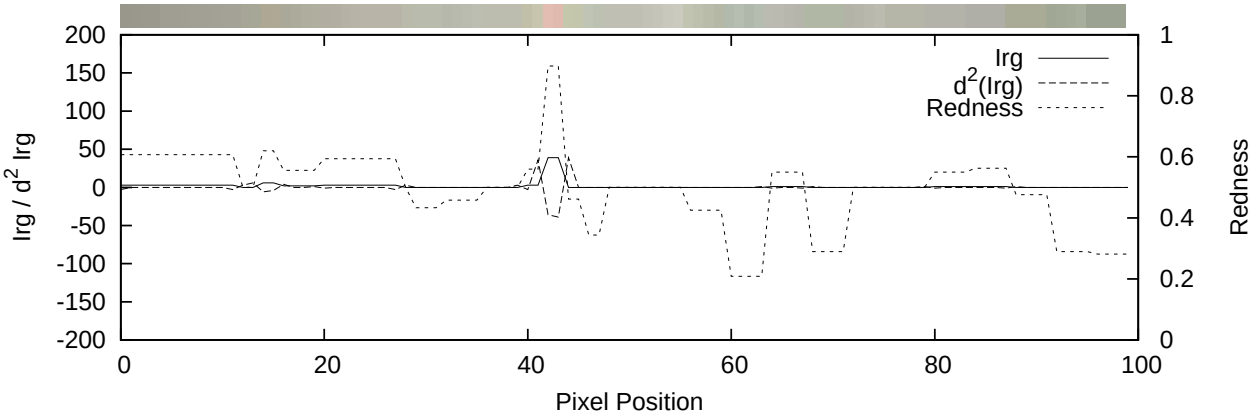


Figure 4.9: Irg, d^2Irg, and Redness in the bright region.

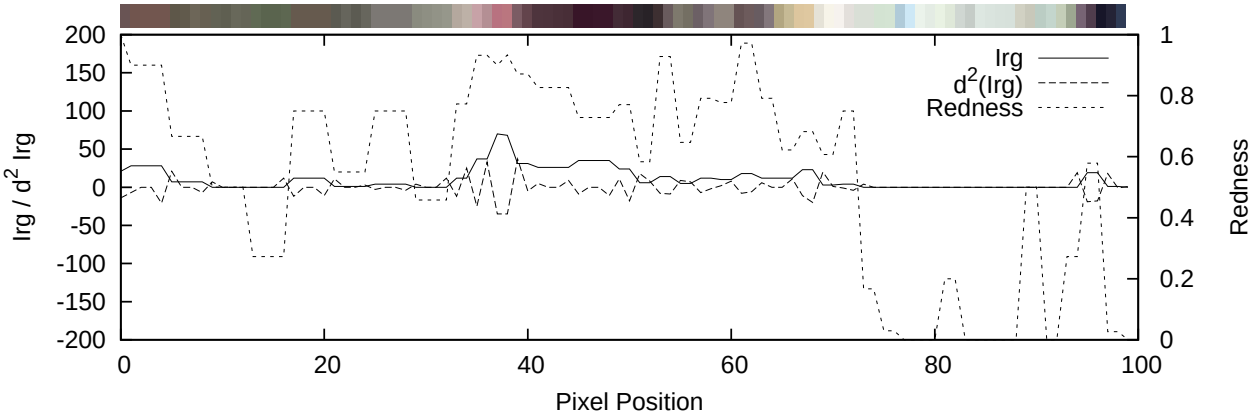


Figure 4.10: Irg, d^2Irg, and Redness in the noisy region.

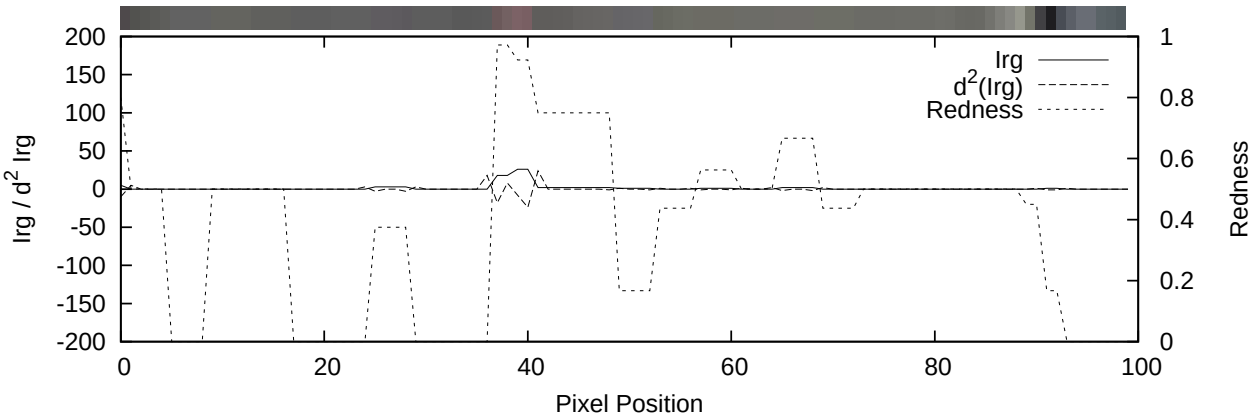


Figure 4.11: Irg, d^2Irg, and Redness in the plane region.

```
input  : A RGB image I(size w × h), each element in I contains RGB values (Ir, Ig, Ib)
output: A float array PixValue(size w) and an integer array PixPosition(size w)
1  G = ( 1  −2  1 );
2  for i ← 1 to w do
3      PixValue(i) ← 0;
4      PixPosition(i) ← 0;
5      for j ← 1 to h do
6          conv ← G ⋆ (Ir[i, j] − Ig[i, j]);
7          if conv > PixValue(i) then
8              if Redness([i, j]) ≥ 0.5 and RedGradient([i, j]) ≤ 0.15 then
9                  PixValue(i) ← conv;
10                 PixPosition(i) ← j;
11             end
12         end
13     end
14 end
15 return PixValue, PixPosition;
```

Algorithm 1: Column segmentation to detect laser reflecting points

reflecting points in the image. Compared to other conventional edge detecting methods, it has some advantages because of its domain specific constraints.

However, column segmentation also has a few drawbacks. For example, when there is no appropriate point in the column because of occlusion of the laser light, it tries to find one anyway and will result in marking an irrelevant pixel. Such marked points are basically noise, and further work is needed to reduce them. Column segmentation also lacks the capability to interconnect proximate points marked in neighboring columns.

Those drawbacks have been later resolved and implemented as described in Section 4.2.7.

4.2 Behavior Implementation

The essential goal of the perceptual behaviors is to provide navigational and/or warning guidance to the user through a tactile interface. The perceptual behaviors are designed for indoor usage with partial domain knowledge, such as destination and portal information. A set of typical spec-

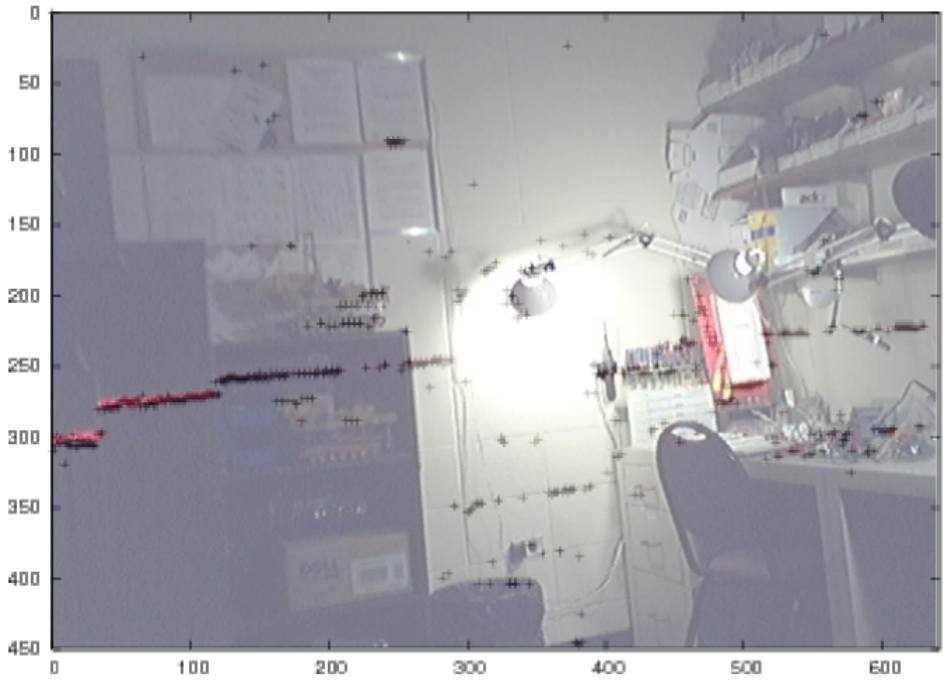


Figure 4.12: Detected points overlapped onto the original image

ifications of the goals/tasks that should be performed by the wheelchair-navigation agent are as follows:

> While roaming without any specific instructions from the user, the wheelchair-navigation agent should mainly (1) notify of obstacles; however, under crowded circumstances, (2) navigate the orientation of an available free space to the user. Simultaneously (3) search for a landmark (e.g., a doorway) from the image, and if a doorway is found, (4) assist the user to pass through it.

From the user's perspective, the first specification, notifying of obstacles, simply constructs a perceptual support, almost direct translation from ranging information into a vibrotactile representation. This specification, however, will almost certainly fail to serve the purpose when the number of obstacles increases because of the *masking effect*[2] (Verrillo, 1983). In addition, it is far

[2]When multiple vibrotactile stimuli simultaneously present to the skin, one stimulus decreases the detectability of another.

more effective to navigate the user to an available space (orientation) when in a crowd or a narrow corridor.

The second specification, navigating a free-space, serves in a complementary manner to notifying of obstacles since if there are many free areas available, it is also difficult for the wheelchair-navigation agent to determine the best orientation to navigate. By these specifications, the user will be either notified of obstacles or navigated to the recommended orientation one way or another.

The third specification, searching for a landmark, although it needs to be concurrently performed with other specifications, is not brought to the user's attention until a landmark is found (e.g., a doorway). This specification mainly relies on sensory input from the motorized vision module and machine/computer vision techniques performed on the host computer in order to accomplish more complicated tasks.

Once the doorway is confirmed as a valid portal to go through, the fourth specification is designed to perform a plan-like navigation task from the current location to the destination (the other side of the door) based on the acquired geometric information and the result of the user output (wheelchair movement) while taking into account dynamic changes in the environment such as obstacles.

To accomplish these tasks, the perceptual/navigational behaviors of interest in our project are constructed: namely, two reactive behaviors, `Obstacle notification` and `Free-space finding`; a task-oriented behavior, `Doorway navigation`; a command arbitration behavior, `Navigation command manager`; and `Sensor command manager` which manages the commands for the sensor-related effectors (servo motors and the laser line generator).

Figure 4.13 illustrates the schematic diagram of the perceptual/navigational behaviors accompanied with the sensor-effector systems. The thin arrows represent data links, and the thick arrows represent actuator links. Other port-to-port links are not shown in this diagram.

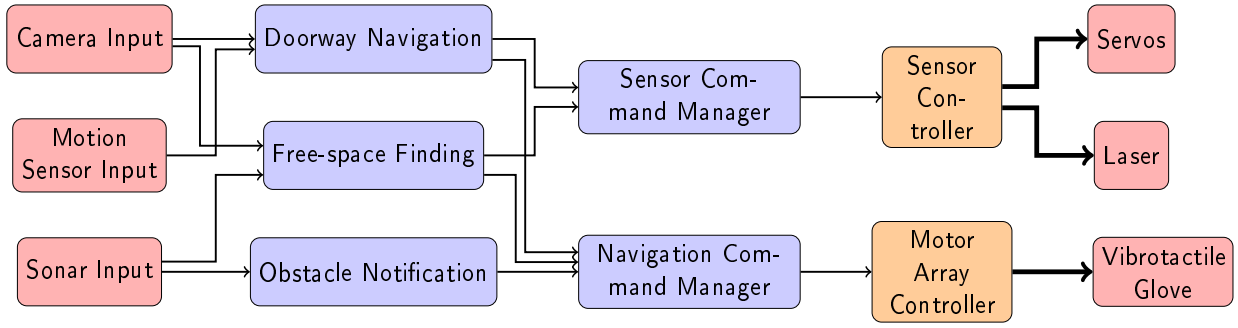


Figure 4.13: Schematic diagram of the perceptual and navigational behaviors (thin arrows: data links, thick arrows: actuator links).

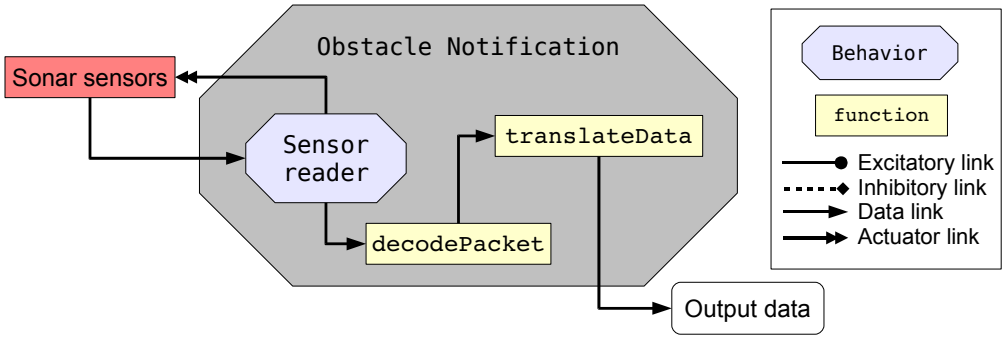


Figure 4.14: Schematic diagram of the `Obstacle notification` behavior.

4.2.1 Obstacle notification

The `Obstacle notification` behavior is a reactive behavior which takes sensor readings from the ultrasonic sensors and directly maps into vibrotactile representations. It consists of one subordinate behavior, `Sonar reader`, and two internal functions: `decodePacket` and `translateData` (Figure 4.14). The `Sonar reader` behavior is a component-like behavior which reads ranging data from the ultrasonic sensors (Section 4.2.2). The `decodePacket` function decodes the data packet from the `Sonar reader` behavior into an array of ranging data. The `translateData` function interprets a numeric value of ranging data into a discrete representation which expresses the distance of an obstacle within the scope of each sensor.

4.2.2 SONAR READER

The `Sonar reader` behavior is implemented as a firmware program that runs on the sensor microcontroller (CrossFire® LPC-2138). It controls the timing of activation/waiting of the paired sonar sensors, reads the registry value of each sensor, bundles the sensor readings into a data packet consisting of a series of hexadecimal codes, and sends the packet to the host computer. The `Sonar reader` behavior is a default task on the sensor controller, therefore the sensor reading cycle runs continuously while the power is on. On each read cycle, the sonar readings are bundled into a data packet and sent to the host computer via a USB cable. The specification of the `Sonar reader` behavior was designed by the author, and the implementation was done by Robert Eunice.

4.2.3 FREE-SPACE FINDING

The `Free-space finding` behavior is, by default, another reactive behavior relating to the sensor readings, but its task is different from the `Obstacle notification` behavior. Instead of notifying of obstacles, it reports a list of orientations in which no immediate obstacles are found. This behavior is particularly effective to find a way out when the SA wheelchair is surrounded by many obstacles. It consists of an internal function `decodePacket` and three subordinate behaviors: `Sonar reader`, `Range finder`, and `Way finder` (Figure 4.15). The `Free-space finding` behavior primarily relies on sonar readings from the `Sonar reader` behavior, and utilizes the `Way finder` behavior to find a list of obstacle-free orientations. If the `Way finder` behavior determines that the sonar readings do not provide enough information, it sends a request to the `Sensor command manager` behavior to move the pan servo motor and invokes the `Range finder` behavior to acquire more accurate depth information.

4.2.4 WAY FINDER

The `Way finder` consists of three internal functions: `findDirection`, `checkValidation`, and `searchAllDirection` (Figure 4.16). The `findDirection` function receives ranging data, seeks obstacle-free orientations, and sends the result to the `checkValidation` function. The

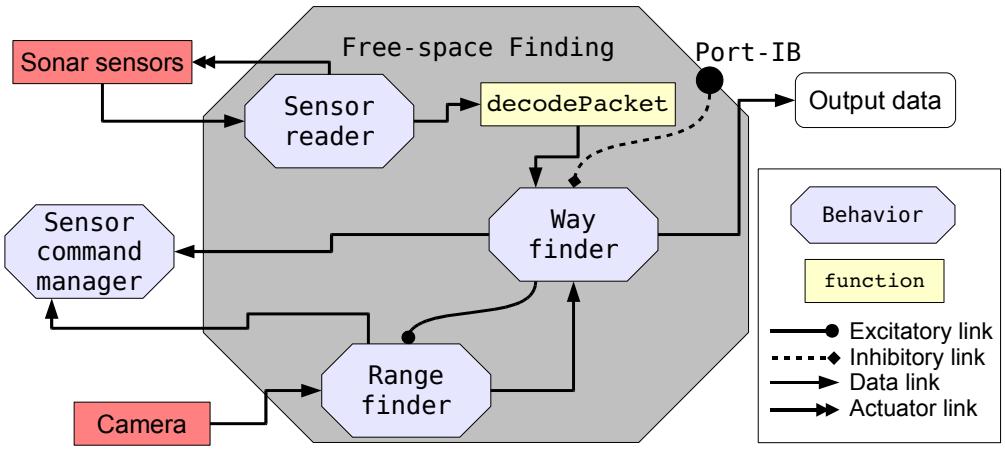


Figure 4.15: Schematic diagram of the `Free-space finding` behavior.

`checkValidation` function then evaluates the result, and if the result meets the preset criteria, it will be sent as an output data to the linked behavior, `Navigation command manager`.

If the result does not satisfies the criteria, that means, if no obstacle-free orientations are found from sonar ranging data, and if Port-IB (an inhibitory input) is activated, the null value will be sent as an output data. Port-IB will be activated if the `Doorway navigation` behavior finds a possible door, and the `Range finder` behavior in the `Doorway navigation` behavior has been in use.

If an obstacle-free orientation is not found from the sonar ranging data and Port-IB is not activated, the `searchAllDirection` function is invoked, and it activates the `Sensor command manager` behavior to request to move the pan servo motor at a prefixed camera angle and also activates the `Range finder` behavior. The `findGap` function then receives the result from the `Range finder` behavior and the current camera angle from the `Sensor command manager` behavior, and seeks an obstacle-free orientation from the array of laser ranging data. The result of the `findGap` function is sent to the `checkValidation` function to be evaluated, and the loop that involves moving the camera with an angle increment and measuring the distance continues until the satisfactory result is found.

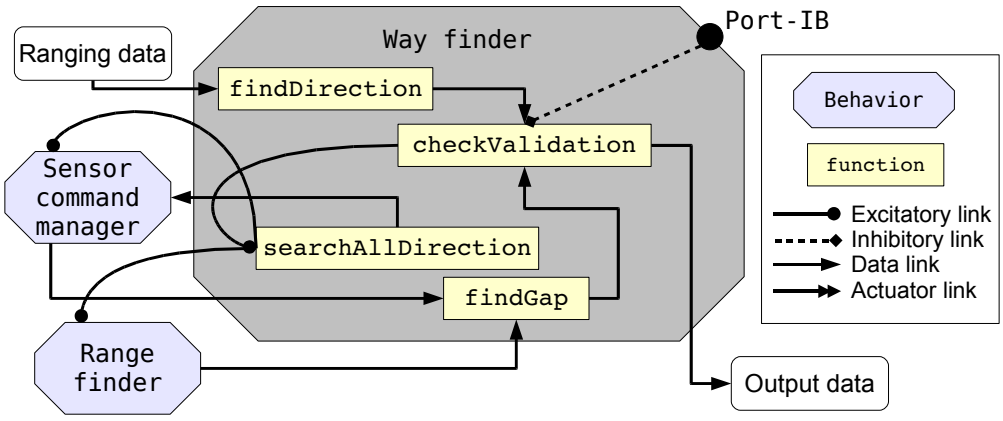


Figure 4.16: Schematic diagram of the `Way finder` behavior.

If the `findGap` function cannot find an obstacle-free orientation in any orientation, that means, the wheelchair is surrounded by obstacles with very short distance in all directions, it then sends an emergency stop code as an output result.

4.2.5 DOORWAY NAVIGATION

The task of passing through a doorway, in general, comprises several subtasks addressing dynamic determination of a maneuvering trajectory and decision making which relies on high level cognitive processes. Some previous studies demonstrate the creation of a local map in order to generate a maneuvering trajectory for an autonomous vehicle (Patel et al., 2002; Surmann et al., 2003). Such maneuvering trajectories typically shape a smooth curvature as illustrated in Figure 4.17. However, generating such trajectories requires much finer angular guidance than the one that the vibrotactile signals can convery.

My guidance approach utilizes the *Pivotal Point/Zone*, an intermediate goal point (area) from which the wheelchair can straightforwardly move toward the doorway, to create a *Pivotal-zone trajectory* that consists of a series of simplified wheelchair movements. In contrast with the maneuvering trajectory in Figure 4.17, the Pivotal-zone trajectory in Figure 4.18 consists of distinctly separated components: a series of nearly straight lines and a sharp turning point. By splitting the trajectory into those components, my approach enables each step of the wheelchair movement to be

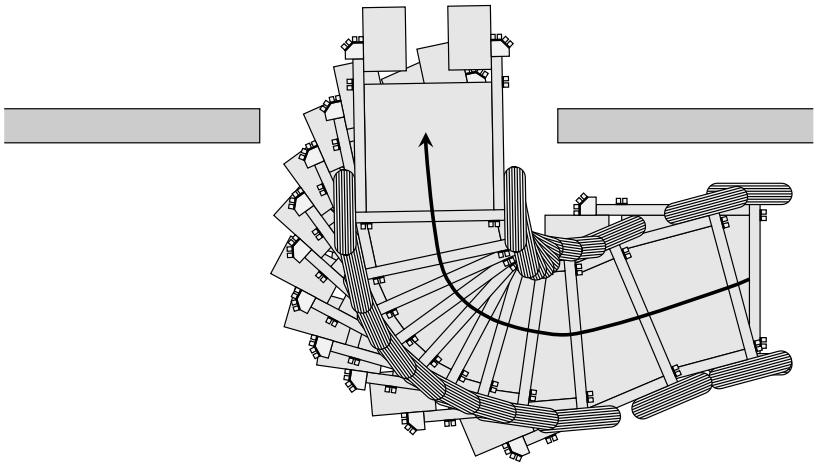


Figure 4.17: Schematic diagram of a maneuvering trajectory for passing through a doorway.

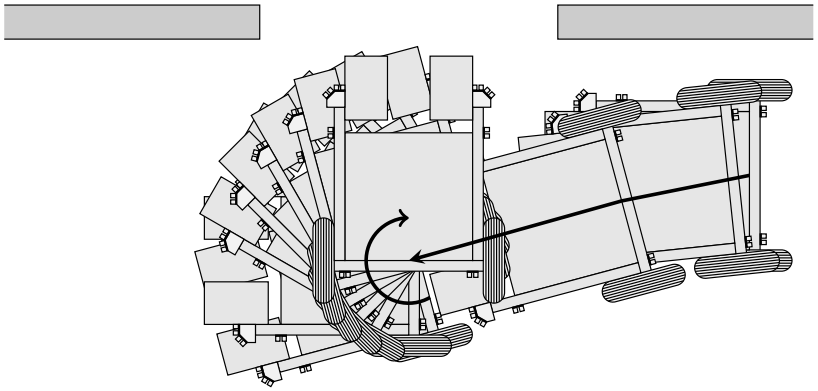


Figure 4.18: Schematic diagram of a Pivotal-zone trajectory for passing through a doorway.

translatable into a simple and intuitive vibrotactile representation, such as "*go front-left*," "*swivel clockwise*," or "*go forward*."

Figure 4.19 illustrates the `Doorway navigation` behavior which comprises the following behaviors: `Doorway detection`, `Range finder`, `Door validation`, `Object tracker`, and `Path planner`. The `Doorway detection` behavior searches for a doorframe from the image stream, and if it finds a candidate, the `Range finder` and `Door validation` behaviors are invoked to confirm the validation of the candidate. Once the doorway is confirmed, the `Object tracker` behavior

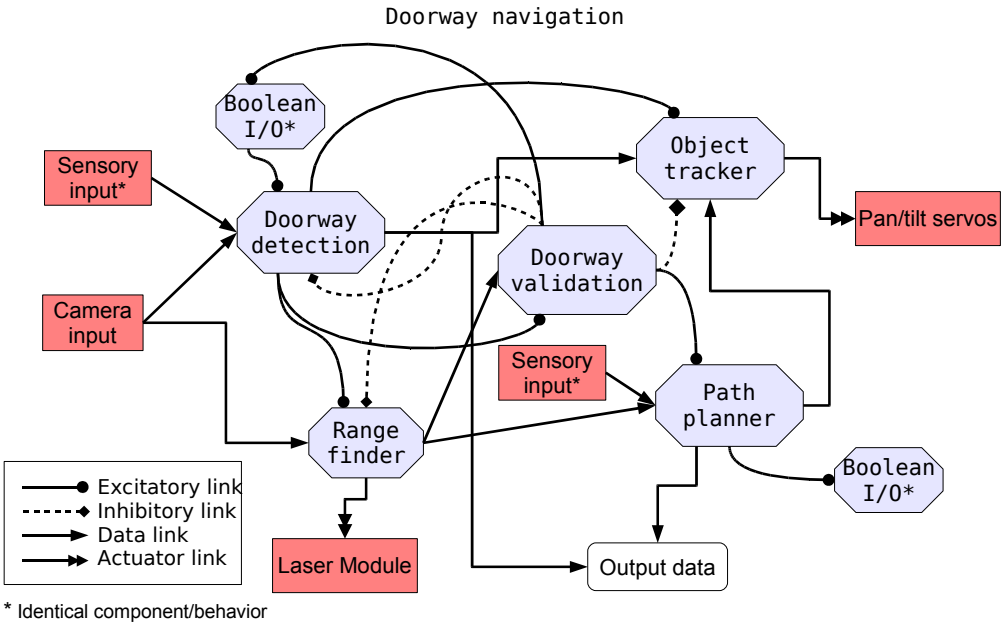


Figure 4.19: Schematic diagram of `Doorway navigation` Behavior Network.

traces the door in order to fixate the camera to the doorway. In the mean time, the `Path planner` behavior is invoked to undertake a repetitive process of localizing the wheelchair, adjusting the path to the sub-goal, and sending a signal indicating the orientation to the user, until the wheelchair reaches the sub-goal (pivotal point). At the pivotal point, the user is guided to swivel until the wheelchair straightforwardly faces the doorway and then to move forward.

For simplification, we treat the doorway as a portal with a binary attribute of either opened or closed. If the door were only a few inches open, in which the `Door validation` behavior would not be able to determine an opened door because of little discrepancy in the laser reflection image, the `Doorway navigation` behavior would conclude that the door navigation task for that door is not available. Similarly, if the door were open but slightly a hindrance for the wheelchair to go through the doorway without actually touching the door, the user would need to push it to make more space. We also ignore the case that a door cannot be fully open but only up to a certain angle because such information would never be available unless the user has *a priori* knowledge about the door or actually tries to push it.

4.2.6 DOORWAY DETECTION

The `Doorway detection` behavior constitutes a computer-vision based algorithm to classify doorways. Doorways are classified based on the presence of the door jambs and the lintel. These are represented in the image as two strong vertical lines connected by a horizontal line. First, the image from the camera is converted to greyscale and reduced to half its original size. Edges are extracted by using a custom convolution filter which combines the accurate edge localization of the Laplacian kernel with the directional bias of a Sobel filter to improve detection of vertical lines.

The detection of vertical lines is achieved by applying the Probabilistic Hough Transform (Kiryati et al., 1991) to the edge image. The technique searches a proportion α ($5\% < \alpha < 15\%$) of randomly selected edge pixels in the image to detect line segments. The vertical line segments are passed to an algorithm that merges line segments deemed close enough. The search for the lintel is limited to the region between the top of the image and the top of the lowest vertical line. Within this sub-region, edge extraction is performed by means of a second-order partial derivative. Horizontal line segments are also detected by the Probabilistic Hough Transformation, examined to ensure they are sufficiently horizontal, and then passed to the merging algorithm. After both sets of line segments are collected, a matching algorithm is performed to determine combinations of two vertical lines connected by a horizontal line that construct a doorframe.

If the matching criteria for a doorframe is met, the `Range finder` behavior (Section 4.2.7) is invoked to measure the distance to the door which enables a validation process to increase accuracy and reduce false positives. Once the doorframe is validated, the dimensions of the door candidate are obtained and sent to the linked behaviors.

The `Doorway detection` behavior has been designed and implemented solely by Jeremy Tarver who worked on the same project.

4.2.7 RANGE FINDER

The `Range finder` behavior also constitutes a computer-vision based algorithm to detect a laser line. The laser detection algorithm seeks each laser point per column while eliminating noise and

erroneous detection. Utilizing the column segmentation method described in Section 4.1.2, an acquired image is first decomposed into its constituent RGB color channels, and an intermediate image is created by subtracting the green channel from the red. The algorithm then seeks a candidate pixel that represents the laser point by means of a second-order differential for each column.

The 8 neighboring pixels of each candidate are examined to eliminate single outlier pixels. After outlier points are eliminated, the resulting points are collapsed into a one-dimensional array representing the height in the original image where each point was found. Based on camera calibrations, these points are then mapped into distances and sent to the linked behaviors.

The `Range finder` behavior was proposed and partially designed by the author, and has been implemented by Jeremy Tarver. In both `Doorway detection` and `Range finder` behaviors, we utilized an open source computer vision library, OpenCV, and Intel® Integrated Performance Primitives for real-time image processing.

4.2.8 PATH PLANNER

As briefly described in Section 4.2.5, the `Path planner` behavior is responsible for path-planning that consists of localizing the wheelchair, creating a path to the goal point, sending a signal indicating the orientation to the goal point, and repeating those steps until the wheelchair reaches the Pivotal Zone. At the Pivotal Zone, the user is guided to swivel until the wheelchair straightforwardly faces the doorway and then to move forward.

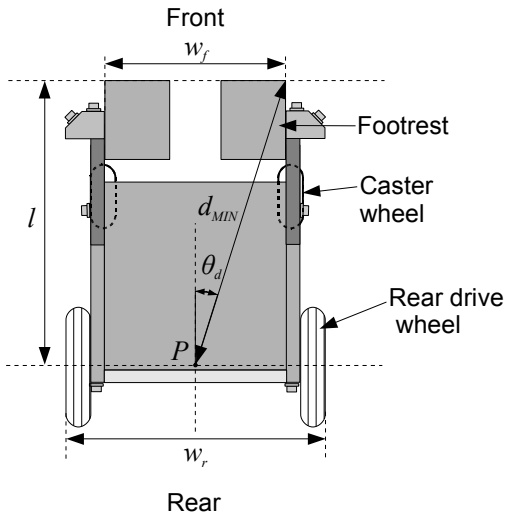
In this section, I discuss the Pivotal Zone strategy in detail and present an algorithm of creating a guidance path that is suitable for the vibrotactile navigation of the wheelchair. In the following sections, I clarify the drive configuration of our wheelchair, define the coordinate system and geometric modeling, present the principle of guidance generation by a series of vector manipulations, and exhibit the flow of the `Path planner` behavior.

WHEELCHAIR DRIVE CONFIGURATION

In order for the `Path planner` behavior to undertake a series of these tasks, the drive configuration of the wheelchair must first be realized. As for the type of wheelchair that we employed, it is concisely explained in the final report (1993–1998) of the Rehabilitation Engineering Research Center (RERC) on Wheelchair Mobility (Hobson and Brubaker, 1998, page 11):

> The ability of a powered wheelchair user to maneuver in tight spaces is closely related to the chair's drive and steering configuration. The most common drive configuration, differential rear wheel drive, consists of fixed and driven rear wheels with front caster wheels. Direction changes are made by individually varying the speeds of the rear wheels. In this configuration the point about which the wheelchair pivots lies on the line perpendicular and running through the center of the rear wheels. The minimum turning radius is achieved when the pivot point is directly between the rear wheels. The minimum space required to turn the wheelchair is then determined by the maximum distance from that point to any other point on the wheelchair. This is usually the front corner of the footrests or the user's feet.

For further discussion, the dimensions of the wheelchair are denoted as follows: l, the length between the edge of the footrest and the center of the rear wheel; w_f, the width of the wheelchair at the footrest; w_r, the width of the wheelchair at the rear wheel; P, the pivot point for the minimum turning radius; d_{MIN}, the distance between P and the front corner of the footrest; and θ_d, the angle between the center line of the wheelchair perpendicular of the rear axle (rear-to-front direction) and the line segment that connects P and the front corner of the footrest (Figure 4.20). The actual measurements of the dimensions are: l, 814 mm; w_f, 460 mm; and w_r, 630 mm, respectively, and d_{MIN} is given by the Pythagorean theorem ($\sqrt{l^2 + (w_f/2)^2} \simeq 846$ mm). The angle θ_d is obtained by the inverse trigonometric function of tangent; thus, $\theta_d = \tan^{-1}(230/814) = 15.78$ (degrees). I have added an extra 200 mm to d_{MIN} to obtain d (1046 mm) in which the portion of foot overhanging on the footrest is considered. Also, a 50 mm of margin is added to w_r to obtain the modified width,



Measurements	
l	814 mm
w_f	460 mm
w_r	630 mm
Side margin	50 mm
w_{rm}	680 mm
d_{MIN}	Approx. 846 mm
d	1046 mm
θ_d	15.78°

Figure 4.20: Dimensions of the wheelchair (Invacare® Nutron R32).

w_{rm}. The modified width of the wheelchair w_{rm}, and the distance between P and the edge of the user's feet d, are the two major constraints to determine the path to go through the doorway.

COORDINATE SYSTEM AND GEOMETRIC MODELING

The wheelchair-centric coordinate frame is placed with the origin at P, the X axis on the line from rear to front of the wheechair, and the Y axis on the line from right to left of the wheelchair. The polar coordinate system is defined as an angle $\varphi = 0$ at the front of the wheelchair, and $\varphi = \pi$ or $-\pi$ at the rear of the wheelchair. The angle values in the first or second quadrants of the Cartesian coordinate system are positive, and the ones in the third and forth quadrants are negative. Similarly, counterclockwise and clockwise angular movements are defined positive and negative, respectively.

Figure 4.21 illustrates the wheelchair-centric coordinate system where the wheelchair is located near a doorway. J_1 and J_2 denote both sides of the door jambs, and their polar coordinates are specified as $J_1\ (r_1, \theta_1)$ and $J_2\ (r_2, \theta_2)$,

where

r_1 denotes distance between P and J_1;

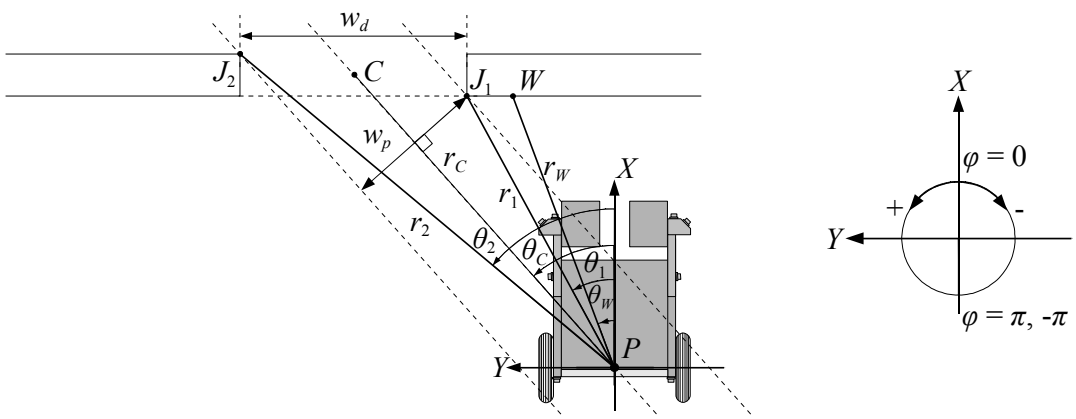


Figure 4.21: Wheelchair-centric coordinate system.

r_2, distance between P and J_2;

θ_1, angle from the $X(+)$ axis to the segment $\overline{PJ_1}$;

θ_2, angle from the $X(+)$ axis to the segment $\overline{PJ_2}$;

and $|\theta_1| < |\theta_2|$.

As an auxiliary reference point, W (r_W, θ_W) is measured at an arbitrary point on the wall near J_1. Measurement of W is needed to calculate the width of the door opening w_d at the initial stage and to obtain the vector component of the orthogonal approach line to the doorway when the camera is too close to the door to obtain both door jambs in a single field of view. Note that abovementioned denotations: r_1, r_2, r_W, θ_1, θ_2, and θ_W are the observed values from the laser ranging data.

The goal point C is defined in the middle of the doorway, and when the pivot point of the wheelchair P reaches C, the task of the `Path planner` behavior is accomplished. In the polar coodinate system, C is specified as C (r_C, θ_C), where r_C denotes the distance between P and C; and θ_C, the angle between the angular origin of the coordinate system and C.

Having defined those properties of the coordinate system, let us proceed to discuss the geometric modeling of the doorway navigation. First, the width of the door w_d can be obtained by a series of relatively simple vector manipulations. Let $\mathbf{j}_1$, $\mathbf{j}_2$, and $\mathbf{w}$ denote the vectors from P to J_1, J_2, and W respectively. As depicted in Figure 4.22, w_d can be obtained by acquiring the projection of the

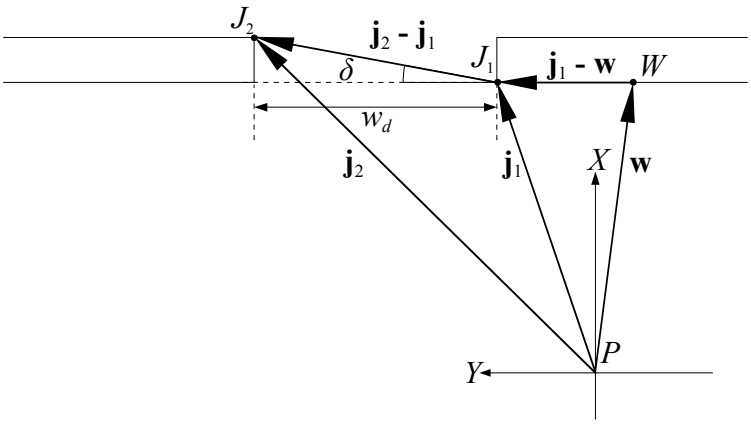


Figure 4.22: Schematic diagram of vector expression to obtain w_d.

vector $\mathbf{j}_2 - \mathbf{j}_1$ to a line segment or vector parallel to the wall. Let δ denote the angle between the reference vector $\mathbf{j}_1 - \mathbf{w}$ and $\mathbf{j}_2 - \mathbf{j}_1$. The projection of $\mathbf{j}_2 - \mathbf{j}_1$ is then written as $|\mathbf{j}_2 - \mathbf{j}_1| \cos\delta$ and obtained by using their inner product as follows:

$$\begin{aligned} w_d &= |\mathbf{j}_2 - \mathbf{j}_1| \cos\delta \\ &= \frac{(\mathbf{j}_2 - \mathbf{j}_1) \cdot (\mathbf{j}_1 - \mathbf{w})}{|\mathbf{j}_1 - \mathbf{w}|}. \end{aligned} \tag{4.4}$$

If $\mathbf{j}$ is expressed as $\mathbf{j}(r,\theta)$ in the polar coordinate system, its X and Y components in the Cartesian coordinate system $(\mathbf{j}_X, \mathbf{j}_Y)$ will be expressed as $(r\cos\theta, r\sin\theta)$; thus, the X and Y components of $\mathbf{j}_2 - \mathbf{j}_1$ are:

$$\mathbf{j}_2 - \mathbf{j}_1 = \begin{pmatrix} r_2\cos\theta_2 - r_1\cos\theta_1 \\ r_2\sin\theta_2 - r_1\sin\theta_1 \end{pmatrix}.$$

Similarly, the X and Y components of $\mathbf{j}_1 - \mathbf{w}$ are:

$$\mathbf{j}_1 - \mathbf{w} = \begin{pmatrix} r_1\cos\theta_1 - r_W\cos\theta_W \\ r_1\sin\theta_1 - r_W\sin\theta_W \end{pmatrix}. \tag{4.5}$$

Thus, their inner product can be obtained by the following operation:

$$\begin{aligned} (\mathbf{j}_2 - \mathbf{j}_1) \cdot (\mathbf{j}_1 - \mathbf{w}) &= (r_2\cos\theta_2 - r_1\cos\theta_1, r_2\sin\theta_2 - r_1\sin\theta_1) \begin{pmatrix} r_1\cos\theta_1 - r_W\cos\theta_W \\ r_1\sin\theta_1 - r_W\sin\theta_W \end{pmatrix} \\ &= r_1 r_2\cos(\theta_1 - \theta_2) + r_1 r_W\cos(\theta_1 - \theta_W) - r_1^2. \end{aligned} \tag{4.6}$$

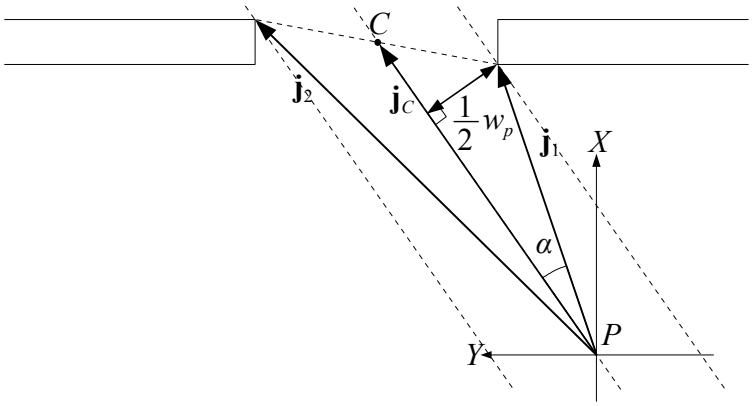


Figure 4.23: Schematic diagram of vector expression to obtain w_p.

By the Pythagorean theorem, $|\mathbf{j}_1 - \mathbf{w}|$ is calculated as follows:

$$\begin{aligned} |\mathbf{j}_1 - \mathbf{w}| &= \sqrt{(r_1\cos\theta_1 - r_W\cos\theta_W)^2 + (r_1\sin\theta_1 - r_W\sin\theta_W)^2} \\ &= \sqrt{r_1^2 - 2r_1r_W\cos(\theta_1 - \theta_W) + r_W^2}. \end{aligned} \tag{4.7}$$

By Equation 4.6 and Equation 4.7, Equation 4.4 is solved as follows:

$$\begin{aligned} w_d &= |\mathbf{j}_2 - \mathbf{j}_1|\cos\delta \\ &= \frac{(\mathbf{j}_2 - \mathbf{j}_1)\cdot(\mathbf{j}_1 - \mathbf{w})}{|\mathbf{j}_1 - \mathbf{w}|} \\ &= \frac{r_1r_2\cos(\theta_1 - \theta_2) + r_1r_W\cos(\theta_1 - \theta_W) - r_1^2}{\sqrt{r_1^2 - 2r_1r_W\cos(\theta_1 - \theta_W) + r_W^2}}. \end{aligned} \tag{4.8}$$

When the wheelchair is placed off from the orthogonal approach line to the door entrance, such as illustrated in Figure 4.21, the available width of the door opening with respect to the wheelchair w_p will be the projection of the original width w_d. The calculation of w_p can also be achieved in a similar manner. In addition to the case of calculating w_d, let $\mathbf{j}_C$ denote the vector from P to C. As depicted in Figure 4.23, w_p can be simply written by using a vector component:

$$\begin{aligned} w_p &= 2|\mathbf{j}_1|\sin\alpha \\ &= 2r_1\sin\alpha. \end{aligned} \tag{4.9}$$

where α denotes the angle between $\mathbf{j}_1$ and $\mathbf{j}_C$ ($0 \le \alpha < \pi/2$).

By using the inner product of $\mathbf{j}_1$ and $\mathbf{j}_C$, $\cos\alpha$ is obtained by the following equation:

$$\cos\alpha = \frac{\mathbf{j}_1 \cdot \mathbf{j}_C}{|\mathbf{j}_1||\mathbf{j}_C|} = \frac{\mathbf{j}_1 \cdot \mathbf{j}_C}{r_1 r_C}. \tag{4.10}$$

In the Cartesian coordinate system, the x and y components of $\mathbf{j}_1$ and $\mathbf{j}_2$ are specified as follows:

$$\mathbf{j}_1 = \begin{pmatrix} r_1\cos\theta_1 \\ r_1\sin\theta_1 \end{pmatrix} \quad \text{and} \quad \mathbf{j}_2 = \begin{pmatrix} r_2\cos\theta_2 \\ r_2\sin\theta_2 \end{pmatrix};$$

therefore, vector $\mathbf{j}_C$ in the Cartesian coordinate system is also specified as:

$$\begin{aligned} \mathbf{j}_C &= \frac{1}{2}(\mathbf{j}_2 - \mathbf{j}_1) \\ &= \frac{1}{2}\begin{pmatrix} r_2\cos\theta_2 - r_1\cos\theta_1 \\ r_2\sin\theta_2 - r_1\sin\theta_1 \end{pmatrix}. \end{aligned} \tag{4.11}$$

Thus, the inner product of $\mathbf{j}_1$ and $\mathbf{j}_C$ by the Cartesian coordinate components is obtained by the following equation:

$$\begin{aligned} \mathbf{j}_1 \cdot \mathbf{j}_C &= \frac{1}{2}(r_1\cos\theta_1, r_1\sin\theta_1)\begin{pmatrix} r_2\cos\theta_2 - r_1\cos\theta_1 \\ r_2\sin\theta_2 - r_1\sin\theta_1 \end{pmatrix} \\ &= \frac{1}{2}r_1\{r_2\cos(\theta_1-\theta_2) - r_1\}. \end{aligned} \tag{4.12}$$

By the Pythagorean theorem, r_C is calculated as follows:

$$\begin{aligned} r_C &= |\mathbf{j}_C| \\ &= \sqrt{\left\{\frac{1}{2}(r_2\cos\theta_2 - r_1\cos\theta_1)\right\}^2 + \left\{\frac{1}{2}(r_2\sin\theta_2 - r_1\sin\theta_1)\right\}^2} \\ &= \frac{1}{2}\sqrt{r_2^2 - 2r_1r_2\cos(\theta_1-\theta_2) + r_1^2}. \end{aligned} \tag{4.13}$$

From Equation 4.10, Equation 4.12, and Equation 4.13, α is obtained by the inverse trigonometric function of cosine:

$$\alpha = \cos^{-1}\left(\frac{r_2\cos(\theta_1-\theta_2) - r_1}{\sqrt{r_2^2 - 2r_1r_2\cos(\theta_1-\theta_2) + r_1^2}}\right). \tag{4.14}$$

Thus, the value of θ_C yields the following cases:

$$\theta_C = \begin{cases} \theta_1 + \alpha & \text{if } \theta_2 > 0 \\ \theta_1 - \alpha & \text{if } \theta_2 < 0 \end{cases}. \tag{4.15}$$

GUIDANCE GENERATION

The principle of the guidance generation consists of three steps: judgment of whether the wheelchair is in the Pivotal Zone, creation of a direction to navigate the wheelchair to the Pivotal Zone, and navigation to turn the wheelchair until it faces the doorway followed by the command of "go-forward."

Figure 4.24 illustrates the flowchart of the `Path planner` behavior in detail. The flow of the guiding process consists of four levels, each of which contains one or more discriminants and a loop back process. Before the first discriminant at Stage 0 (also denoted as Dis. 0 in Figure 4.24), necessary geometric information of the door: the localization of the three points (J_1, J_2, and W) is obtained from the laser ranging data provided by the `Range finder` behavior.

Stage 0: At Stage 0, the width of the door is examined to determine whether the doorway is wide enough for the wheelchair to go through. If w_d is smaller than w_{rm}, the `Path planner` behavior halts and sends the deactivation signal to the Boolean I/O cell of its parent, the `Doorway navigation` behavior.

It might seem awkward to set the Discriminant 0 because the validation of the door must have been confirmed by the `Door validation` behavior before the `Path planner` behavior is invoked. However, in order to accommodate a dynamic environment and to prepare for a possible misjudgment of the `Door validation` behavior, this discriminant is needed as a safeguard to keep this behavior from falling into an infinite loop seeking a non-existent path. Also note that the `Path planner` behavior cannot be linked to activate the `Door validation` behavior to avoid another infinite loop of activating one behavior to another.

Stage 1: The projected width of the door opening w_p is examined to compare with w_{rm} at Stage 1 (Dis. 1). If $w_p \geq w_{rm}$, the process goes to the next stage, otherwise, a path orientation to adjust the position of the wheelchair is created. The path orientation is then converted into the vibrotactile

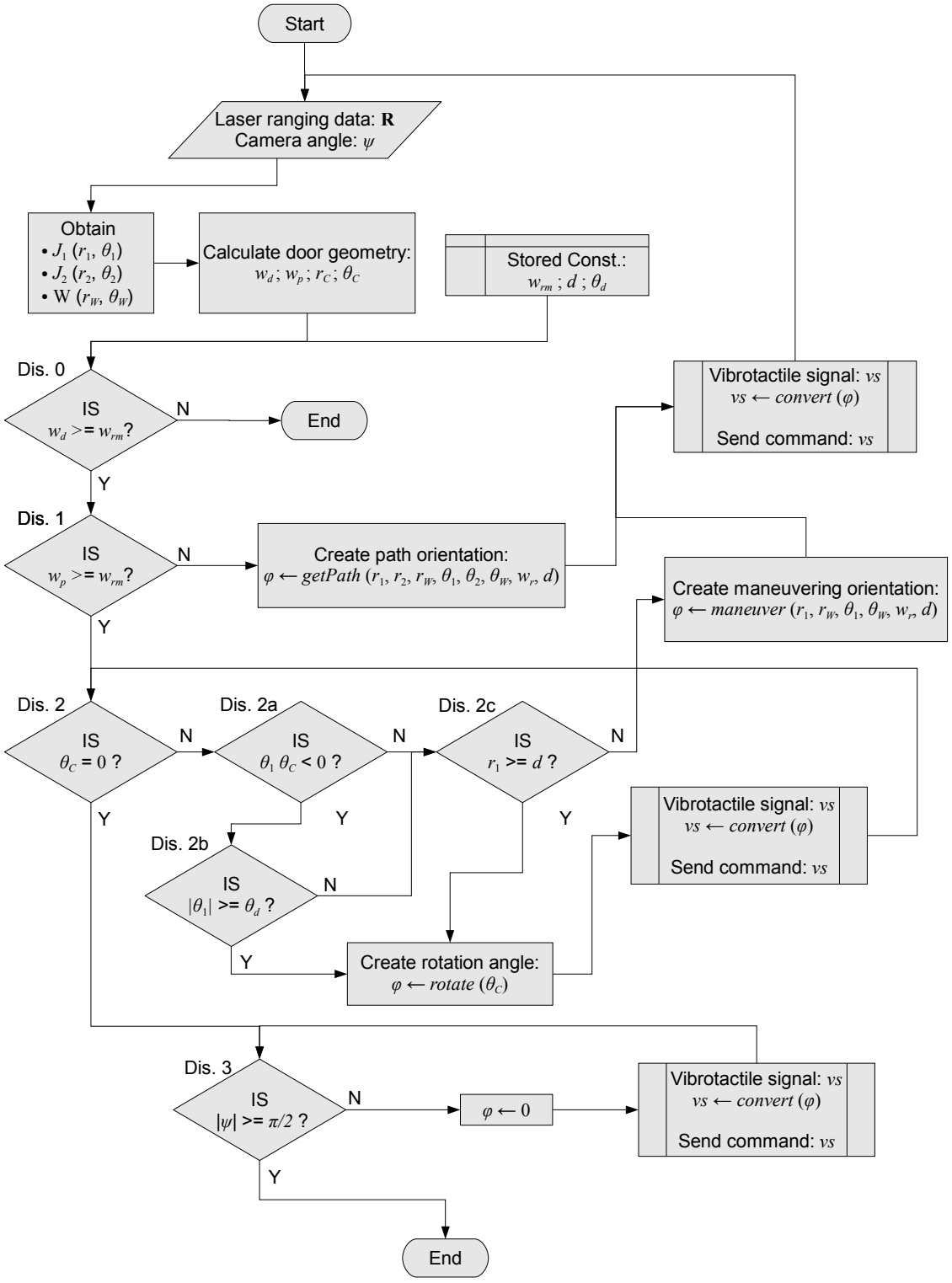


Figure 4.24: The flowchart of the `Path planner` behavior.

signal to convey to the user, and the process flow goes back to the starting step. The detail of creating the path orientation will be discussed later in this section (**Path Adjustment** in page. 58).

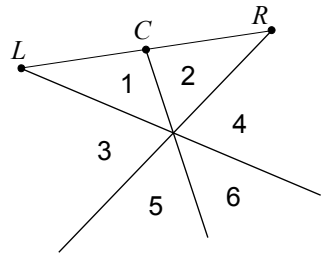
Stage 2: At Stage 2, the guiding process handles the angular adjustment. There are four discriminants at this stage: Dis. 2, Dis. 2a, Dis. 2b, and Dis. 2c. The first discriminant (Dis. 2) of them examines whether θ_C is zero. If it is, this means that the wheelchair straightly faces the goal point (the center of the door opening); therefore, the process proceeds to the next level because no angular adjustment is needed.

If θ_C is non-zero, the second discriminant at this level (Dis. 2a) examines the sign of the multiplication of θ_1 and θ_C. If this value is smaller than zero, the angular origin of the wheelchair (the $X(+)$ axis of the chair) is between the two door jambs.

Discriminant 2a may seem unintuitive; however, the validity of this discriminant can be easily clarified by examining the sign of each angle and comparing their absolute values as shown in Figure 4.25. Note that the properties of angles (θ_L, θ_C, and θ_L) in each angle field (from 1 to 6) show where the angular origin of the wheelchair is in that field. For instance, consider the angular origin of the wheelchair in Field 2, from the angular origin, L and C are counterclockwise located, and R, clockwise; therefore, their angle signs are positive, positive, and negative, respectively. R is closer to the angular origin than L, so θ_R becomes θ_1. Since the signs of θ_R and θ_C are different, the multiplication of these yields a negative. Similarly one can examine Field 1, and these two are the only cases in which the value of the multiplication $\theta_1\theta_C$ yields a negative.

If $\theta_1\theta_C < 0$ is true, the next discriminant (Dis. 2b) examines whether the absolute value of θ_1 is greater than or equal to θ_d, the angle between the angular origin of the wheelchair and the front corner of the footrest. If this is the case, the angular origin is close to facing the center of the doorway, and the both door jambs will not interfere upon turning the wheelchair regardless of the value of r_1. Thus, the `Path planner` behavior generates a simple rotational command for the user.

If either Discriminant 2a or 2b turns out false, the next Discriminant 2c examines whether r_1 is greater than d, the distance between P and the user's foot. If this is true, the process also generates



Field#	Signs			θ_1	Absolute value comparison
	θ_L	θ_C	θ_R		
1	+	−	−	θ_L	$\lvert\theta_L\rvert < \lvert\theta_R\rvert$
2	+	+	−	θ_R	$\lvert\theta_R\rvert < \lvert\theta_L\rvert$
3	−	−	−	θ_L	$\lvert\theta_L\rvert < \lvert\theta_C\rvert < \lvert\theta_R\rvert$
4	+	+	+	θ_R	$\lvert\theta_R\rvert < \lvert\theta_C\rvert < \lvert\theta_L\rvert$
5	−	−	+	θ_L	$\lvert\theta_L\rvert < \lvert\theta_R\rvert$
6	−	+	+	θ_R	$\lvert\theta_R\rvert < \lvert\theta_L\rvert$

Figure 4.25: Comparison of sign and absolute value properties in angle fields.

a simple rotational command. If $r_1 < d$, the `Path planner` behavior creates a maneuvering orientation to make more space between the wheelchair and the nearest jamb, or to generate a command to navigate the user to turn alternative direction.

Stage 3: This is the final stage of the `Path planner` behavior. At this stage, the wheelchair is on the process of passing through the doorway, and the only task is to realize whether the wheelchair reaches the goal point. Since the camera is controlled by the `Object tracker` behavior to trace either one of the door jambs, if the camera angle ψ becomes greater than or equals to $\frac{\pi}{2}$, the whole task is completed. Otherwise, the `Path planner` behavior navigates the user to keep going forward.

Path adjustment: Generally, the wheelchair may not always face parallel or perpendicular to the doorway or a wall. Let X' denote the direction along with the orthogonal approach line to enter the doorway, and Y', from right to left with respect to face the doorway (Figure 4.26).

The adjustment of the path is to create a vector $\mathbf{n}$ which is obtained by adding two vectors: $\mathbf{c}_p$ and $\mathbf{w}_s$, as the following vector equation:

$$\mathbf{n} = \mathbf{c}_p + \mathbf{w}_s \tag{4.16}$$

where $\mathbf{c}_p$ is a vector with its direction along with the Y' axis and the size of the projection of $\mathbf{j}_C$ to Y', and $\mathbf{w}_s$ has a direction component from J_1 to orthogonal of $\mathbf{j}_C$ with the size of $w_{rm} - w_p$ if w_{rm} is greater than w_p, otherwise zero.

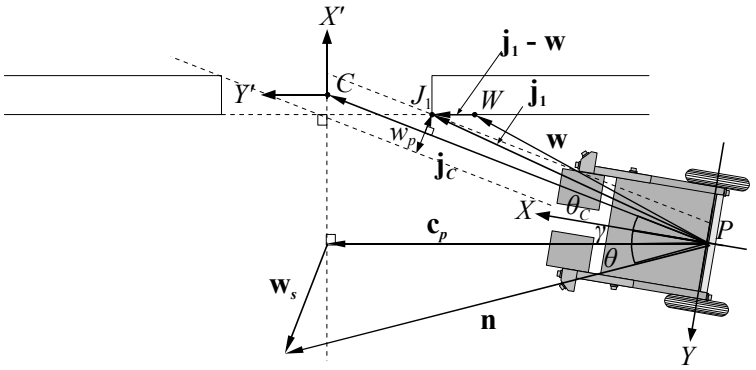


Figure 4.26: Schematic diagram of vector expressions for path adjustment.

Let γ denote the adjustment angle between the X axis and direction Y' in Figure 4.26. The adjustment angle γ can be obtained by utilizing the reference vector $\mathbf{j}_1 - \mathbf{w}$ as follows:

$$\cos\gamma = \frac{(\mathbf{j}_1 - \mathbf{w})_X}{|\mathbf{j}_1 - \mathbf{w}|} \tag{4.17}$$

where $(\mathbf{j}_1 - \mathbf{w})_X$ denotes the X component of the vector $\mathbf{j}_1 - \mathbf{w}$. The vector $\mathbf{j}_1 - \mathbf{w}$ was already specified in the Cartesian coordinate system by Equation 4.5 as:

$$\mathbf{j}_1 - \mathbf{w} = \begin{pmatrix} r_1\cos\theta_1 - r_w\cos\theta_w \\ r_1\sin\theta_1 - r_w\sin\theta_w \end{pmatrix}.$$

The length of $\mathbf{j}_1 - \mathbf{w}$ was also obtained by Equation 4.7 as:

$$|\mathbf{j}_1 - \mathbf{w}| = \sqrt{r_1^2 - 2r_1r_w\cos(\theta_1 - \theta_w) + r_w^2}.$$

Thus, γ is obtained by the inverse trigonometric function of cosine as:

$$\gamma = \cos^{-1}\left(\frac{r_1\cos\theta_1 - r_w\cos\theta_w}{\sqrt{r_1^2 - 2r_1r_w\cos(\theta_1 - \theta_w) + r_w^2}}\right). \tag{4.18}$$

In the polar system, $\mathbf{c}_p$ is expressed as $\mathbf{c}_p(|\mathbf{j}_C|\cos\beta, \gamma)$,

where

$$\beta = \begin{cases} |\theta_C| + |\gamma| & \text{if } \theta_C \cdot \gamma > 0 \\ |\theta_C| - |\gamma| & \text{if } \theta_C \cdot \gamma < 0 \end{cases}. \tag{4.19}$$

Recalling $|\mathbf{j}_C| = r_C$, which was obtained in Equation 4.13 as:

$$r_C = \frac{1}{2}\sqrt{r_2^2 - 2r_1 r_2 \cos(\theta_1 - \theta_2) + r_1^2},$$

the X and Y components of $\mathbf{c}_p$ are then expressed as:

$$\mathbf{c}_p = \begin{pmatrix} r_C \cos\beta \cos\gamma \\ r_C \cos\beta \sin\gamma \end{pmatrix}. \tag{4.20}$$

The other vector $\mathbf{w}_s$ in the polar coordinate system is denoted as $\mathbf{w}_s(w_s, \eta)$, thus, it can be expressed in the Cartesian coordinate system as

$$\mathbf{w}_s = \begin{pmatrix} w_s \cos\eta \\ w_s \sin\eta \end{pmatrix} \tag{4.21}$$

where

$$w_s = \begin{cases} w_{rm} - w_p & \text{if } w_{rm} > w_p \\ 0 & \text{otherwise} \end{cases}$$

and

$$\eta = \begin{cases} \theta_C + \dfrac{\pi}{2} & \text{if } \theta_C < 0 \\ \theta_C - \dfrac{\pi}{2} & \text{if } \theta_C > 0 \end{cases}.$$

Finally, the X and Y components of the vector $\mathbf{n}$ are obtained as:

$$\mathbf{n} = \begin{pmatrix} r_C \cos\beta \cos\gamma + w_s \cos\eta \\ r_C \cos\beta \sin\gamma + w_s \sin\eta \end{pmatrix}. \tag{4.22}$$

The vector $\mathbf{n}$ is characterized to be attracted to the center line between the door jambs of the doorway and to be repulsive for the edges of the door jambs. As the wheelchair closely proceeds to the center line, the size of the attraction component of the vector decreases. If the wheelchair comes too close to the wall near the doorway, the repulsion component of the vector will increase, which means the vector navigates the user to take a greater approach angle to the door.

Angular adjustment: Angular adjustment essentially navigates the user to turn (swivel) the wheelchair to straightforwardly face the doorway. In many occasions, the value of θ_C ($0 \leq |\theta_C| \leq \pi$) is given to generate a rotational command, however, such rotation may cause the wheelchair to hit an obstacle in some occasions.

Figure 4.27 illustrates an example of such instances. From the pivot point P of the wheelchair at Stage 2, the available width w_p is wide enough, but the transition to Stage 3 by turning clockwise will make the wheelchair hit the wall because the distance between the wall and wheelchair is less than the minimum turning radius d.

In the angular adjustment process, the `Path planner` behavior determines whether a constraint angle area exists based on the sensory input. If the original turning violates the constraint area, it examines whether the alternative angle area is available. If the alternative angle area yields a non-constraint area; for example, as illustrated in Figure 4.27, it will generate the command to swivel counterclockwise even though such turning takes more than an 180° (π in Radian) rotation. If the alternative angle area is also a constraint area, the process goes back to the path adjustment stage. The alternative adjustment angle θ'_C is given as follows:

$$\theta'_C = \begin{cases} \theta_C + 2\pi & \text{if } \theta_C < 0 \\ \theta_C - 2\pi & \text{if } \theta_C > 0 \\ 0 & \text{otherwise} \end{cases} \cdot \tag{4.23}$$

4.2.9 SENSOR COMMAND MANAGER

The `Sensor command manager` behavior accepts input commands from the `Doorway navigation` and the `Free-space finding` behaviors, more specifically, the `Object tracker` and `Range finder` behaviors in the `Doorway navigation` behavior, and the `Way finder` and `Range finder` behaviors in the `Free-space finding` behavior. When the `Object tracker` and `Range finder` behaviors in the `Doorway navigation` behavior are active, a request of the `Range finder` behavior in the `Free-space finding` behavior to move the servo motors is suppressed.

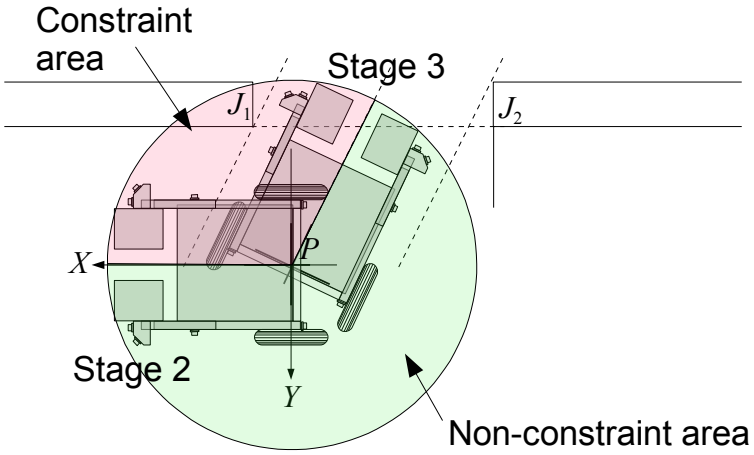


Figure 4.27: Schematic diagram of constraint/non-constraint areas of angular adjustment.

4.2.10 NAVIGATION COMMAND MANAGER

The `Navigation command manager` behavior accepts navigation commands from the `Doorway navigation`, `Free-space finding`, and `Obstacle notification` behaviors, arbitrates the commands, and generates a vibrotactile representation. It consists of four internal functions: `gateSignals`, `selectSignal`, `encodePacket`, and `mergeCommands` (Figure 4.28). In this section, I use the following abbreviations to represent the commands from the aforementioned behaviors: $Command_{door}$ denotes the command from the `Doorway navigation` behavior; $Command_{free}$, the `Free-space finding` behavior; and $Command_{obs}$, the `Obstacle notification` behavior.

The `gateSignals` function accepts $Command_{free}$, which will be one of the following: a list of obstacle-free orientations, the "stop-the-chair" signal (no free-space is found), or a null code (the `Doorway navigation` behavior is active). The "stop-the-chair" signal is sent to the `selectSignal` function, and the null code is sent to the `mergeCommands` function. If $Command_{free}$ consists of a list of obstacle-free orientations, it will be sent to both.

By default, $Command_{obs}$ is sent to the `encodePacket` function to generate a series of control codes (a vibrotactile representation) to be passed to the motor-array controller. However, when more than two immediate obstacles are found in $Command_{obs}$, the `selectSignal` function chooses

the most appropriate orientation from the list of obstacle-free orientations in $Command_{free}$. The selection of the orientation in $Command_{free}$ is performed based on a preset criterion, such as to choose the most obstacle-free orientation.

When the `Doorway navigation` behavior is activated, both $Command_{door}$ and $Command_{free}$ are sent to the `mergeCommands` function to find an orientation that satisfies both constraints. If $Command_{free}$ consists of a null code, the `mergeCommands` function simply chooses $Command_{door}$ to be sent to the `encodePacket` function. If $Command_{free}$ consists of a list of obstacle-free orientations, the `mergeCommands` function tries to find the closest orientation to the one from $Command_{door}$. Once the `mergeCommands` function issues a new command, the `encodePacket` function only takes that command and ignores $Command_{obs}$.

The `encodePacket` function generates a 36-byte control code packet which is executed by the `Vibrotactile motor controller` behavior to generate a vibrotactile signal to the Vibrotactile Glove. If the vibrotactile signal to be presented is directional guidance or the "stop-the-chair" signal, the `encodePacket` function consults a look-up table that defines a series of preset control codes. If the vibrotactile signal needs to notify of information about obstacles (spatial information), the `encodePacket` function generates a 36-byte control code based on the spatial information. The reason for not using another look-up table for spatial information is the efficiency of development. The number of control codes depends on the combination of the location and distance of one or two obstacles, therefore, it can easily be counted up to several hundred based on the current configuration of spatial representation as described in Section 4.3.2. The details of the signal generation are discussed in Section 4.3.

4.2.11 VIBROTACTILE MOTOR CONTROLLER

The `Vibrotactile motor controller` receives the output signal from the `Navigation command manager` behavior and generates the actual vibration pattern for the Vibrotactile Glove. This behavior is indeed a firmware component which is implemented on the motor-array microcontroller

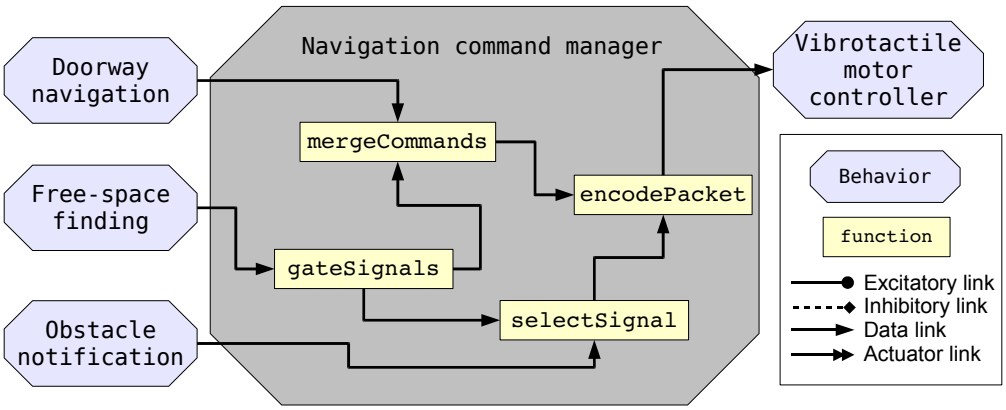


Figure 4.28: Schematic diagram of the `Navigation command manager` behavior.

(Section 3.3.3).[3] Each vibrotactile signal pattern consists of 12 phases and is repeated until another pattern or the stopping signal is issued. A phase comprises three byte codes, which represents the status of the upper motors (motor 0 to 7), the status of the lower motors (motor 8 to 15), and the duration of the phase, respectively. Any combination of the motors can be active per phase. The duration of a phase can be set from 3 ms to the maximum of 765 ms incremented by three milliseconds.

4.3 VIBROTACTILE SIGNAL PATTERN

All the necessary information to operate the SA wheelchair must be symbolized as a set of signals, so that the wheelchair-navigation agent is able to transmit the signals to the vibrotactile display to generate vibrotactile signal patterns. The vibrotactile signal patterns for the SA wheelchair operation are categorized into three groups: warning signal, spatial representation, and directional guidance.

Warning Signal When an emergency situation occurs in which the SA wheelchair determines that it is too dangerous to drive the wheelchair or finds any sort of system malfunctioning, the warning signal is invoked, and the user must stop the wheelchair movement.

[3]The specification of the `Vibrotactile motor controller` behavior was requested by the author, and this component was designed and implemented solely by Dr. Covington.

Spatial Representation The spatial representation is translated from visuospatial information to help the user to "sense" the orientation and distance of an obstacle.

Directional Guidance The directional guidance is designed to navigate the wheelchair user in the desired direction (e.g. "go-forward", "turn-left").

The vibrotactile display consists of nine vibrating motors (*vibrotactors*). A vibrotactor generates an individual pulse and its characteristic is determined by controlling the duration of pulses and the interpulse interval. A sequential pulse consists of a sequence of individual pulses generated by more than one vibrating motor that indicates the desired direction.

4.3.1 Warning Signal

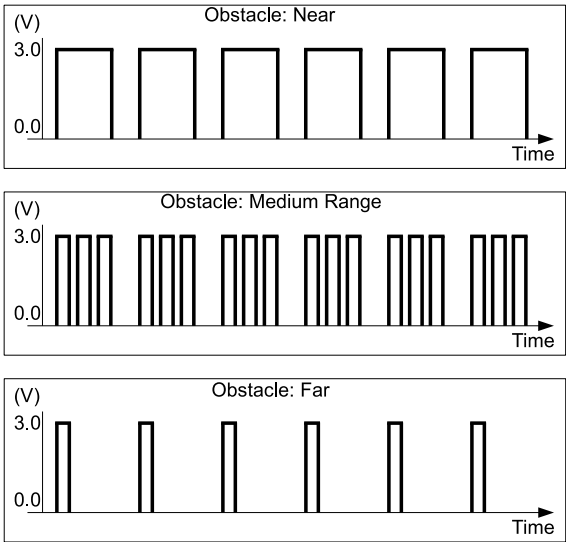
The warning message does not need to contain any visuospatial information; thus, all vibrotactors are designed to repeat a long duration pulse simultaneously.

4.3.2 Spatial Representation

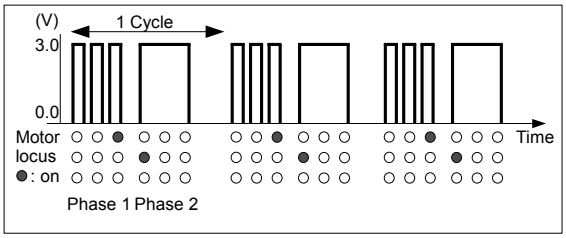
The orientation and distance of an obstacle needs to be discretely mapped into the vibrotactile display due to its limited spatial resolution. The orientation of an object is mapped into the locus of the corresponding peripheral vibrotactors, each of which represents approximately 45 degrees of the surrounding area of the chair. The degree of distance is set into three levels: "near" denotes the range from approximately 0.1 m to 0.6 m from the wheelchair; "medium," 0.6 to 1.2 m; and "far," 1.2 to 1.8 m. These thresholds are primarily experimental, and will be adjusted during the field test.

The magnitude of each level is implemented by the density of the vibration pulse. Repetition of a single short vibration pulse represents the "far" range; three short pulses, the "medium" range; and a long pulse, the "near" range (Figure 4.29a).

As briefly mentioned in Section 4.2, the presence of multiple stimuli in vibrotactile perception can cause the masking effect which decreases the detectability of stimuli. In the primarily



(a) Examples of signal patterns for notifying of an obstacle in different ranges (top: obstacle is near, middle: obstacle is in medium range, bottom: obstacle is far).



(b) An example of a signal pattern for notifying of two obstacles (north-east: medium range, west: near).

Figure 4.29: Schematic diagrams of obstacle signal patterns.

experiment, this phenomenon also occurred. When two stimuli are simultaneously presented, one cannot distinguish the loci of stimuli if they are either different patterns or presented in nearby loci. When more than two stimuli are simultaneously presented, one cannot distinguish the loci of stimuli regardless of their closeness or patterns.

However, limiting to only a single obstacle to be presented is undesirable. To solve this dilemma by compromise, I configured the signal generation mechanism such that one repetitive cycle contains multiple phases, and a unique stimulus is permitted to present in each phase. Figure 4.29b illustrates an example of a signal pattern to notify of two obstacles at the north-east and west of the wheelchair. Containing more phases per cycle is possible; however, the duration of one cycle increases as more phases are included, which would make the display “slow” (typically one phase takes approximately one second).

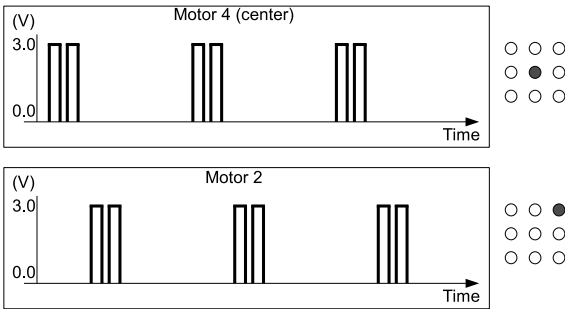


Figure 4.30: An example of the pulse pattern to indicate direction (“north-east”). The dark circle in the array represents the activated tactor.

4.3.3 DIRECTIONAL GUIDANCE

The directional guidance behavior is designed to generate successively activating vibrotactors with a short interstimuli period. The center vibrotactor is first invoked with two short pulses followed by another two short pulses in the desired orientation. I set the number of short pulses to two in order to avoid confusion with spatial representation.

Figure 4.30 shows a sample pattern of the direction “go-forward-right” (or “north-east”) by using two tactors: tactor 4 and tactor 2, in sequence. Applying the carefully chosen interstimuli intervals between the tactors (20 to 200 ms (Geldard, 1985)), it is suggested that the user would feel on their hand a single sensation in the designated direction rather than two separate stimuli. In the preliminary experiment, I found that the appropriate interstimuli interval for the Vibrotactile Glove is between 10 and 30 ms.

4.3.4 VIBROTACTILE SIGNAL PATTERN DEVELOPMENT TOOL

Manually writing a command to generate a vibrotactile signal pattern is possible, however, it would be quite tedious and inefficient work to manually develop a variety of patterns, each of which consists of 36-byte codes. To reduce such a burden, I have implemented a development tool that

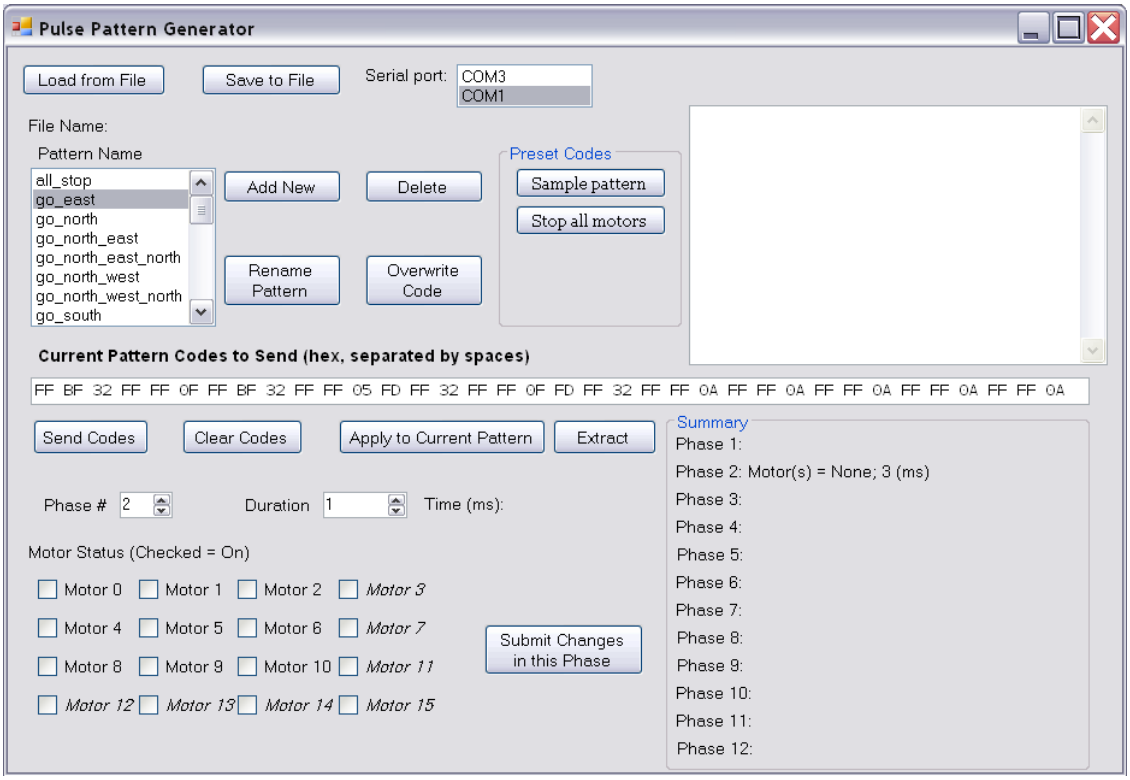


Figure 4.31: Screenshot of the vibrotactile signal development tool.

enables the user (the developer of signal patterns) to create vibrotactile signal patterns more intuitively and efficiently.[4]

Figure 4.31 shows a screenshot of the development tool. The user can generate a signal pattern in the graphical interface form and send it to the motor-array controller via a serial port. The details of a signal pattern can be edited in the interface form and saved it onto a text file. Previously created and saved patterns can also be loaded from a text file in the interface form.

[4]The original prototype of this software was implemented by Dr. Covington. The original version is capable of sending codes but does not have a capability to generate signal patterns, nor to save/load signal patterns to/from a text file.

CHAPTER 5

RESULTS

In this chapter, the first section concerns the effect of the parameters of vibrotactile signals over the qualitative performance of the Vibrotactile Glove. The second section reports the current status of other subprojects and presents the road-map toward the integration of the whole system.

5.1 EFFECT OF VIBROTACTILE SIGNAL PARAMETERS

5.1.1 OBJECTIVES

The purpose of this experiment is to study the effect of basic characteristics of signal parameters to generate vibrotactile patterns. Based on the hardware specification of our motor-array microcontoller, the signal parameters control the length of the duration of either the top or the bottom of the vibrotactile pulses (Figure 5.1). In order to make this tactile display practical, appropriate signal parameters need to be configured in order to make the vibrotactile patterns concise and noticeable. Specifically, the following three characteristics need to be clarified:

- Recognizable shortest duration of pulse (DP) and inter repetition period (IRP) consisting of three pulses
- Appropriate lengths of DP, IRP, and inter stimuli period (ISP) that make human tactile perception feel a directional vibrotactile stimulus along the two tactors (or that make human tactile perception feel the “sensory saltation”)
- Appropriate length of the inter cycle period (ICP) which makes clear distinction from the IRP and ISP

where DP denotes the duration period of a single pulse; IRP, the duration period between pulses on the same tactor; ISP, the duration period between the first stimulus at tactor 4 (center) and the subsequent stimulus at another tactor; and ICP, the duration between the end of the last pulse in a cycle and the beginning of the first pulse in the next cycle.

5.1.2 METHODS

STIMULI AND PROCEDURES

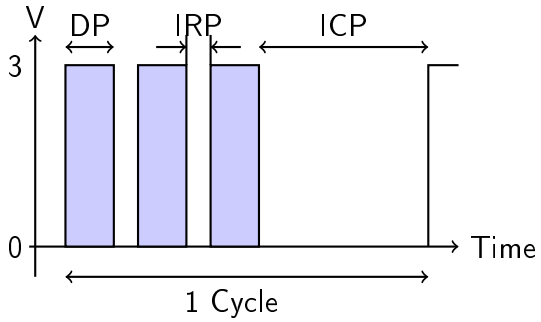
EXPERIMENT 1: The effect of the DP and the IRP for spatial representation was qualitatively evaluated by examining a combination of different lengths of duration, ranging from 50 to 200 milliseconds with approximately 50 ms increments for each of them.

The primary criterion is how human tactile perception can correctly distinguish the number of pulses. The qualitative measures (literal expressions) are defined as follows: No signal, Almost no signal, Almost one pulse, Barely countable, Countable, and Countable with clear localization. The ICP was set to approximately 350 ms during this evaluation.

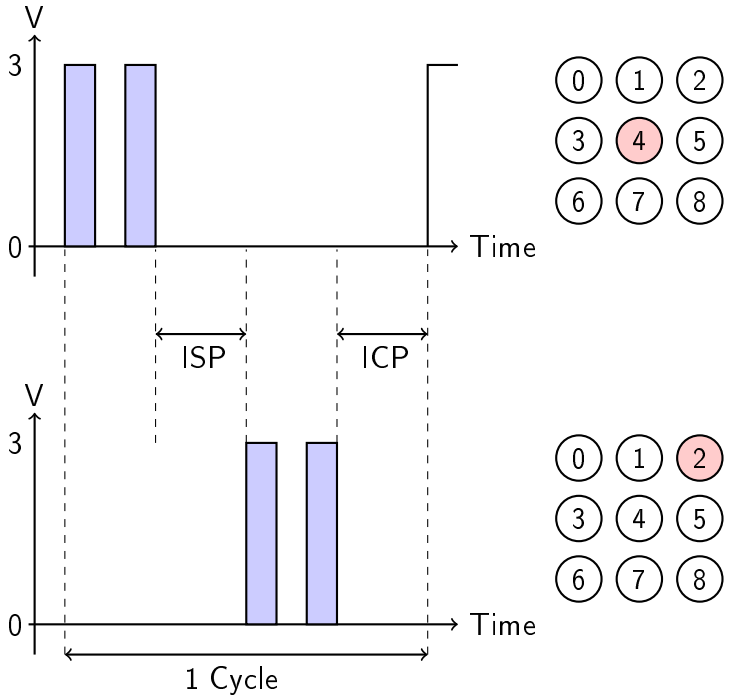
EXPERIMENT 2: The effect of the ICP for spatial representation was qualitatively evaluated by changing the duration ranging from 150 to 350 milliseconds with approximately 50 ms increments.

The main criterion is whether the end of the cycle can be intuitively grasped, and its qualitative measures are defined as No distinction, Vague distinction, and Clear distinction. Based on the results of Experiment 1, the DP was fixed to 200 milliseconds, and the IRP was set to two different levels: 100 and 150 milliseconds.

EXPERIMENT 3: The effect of the DP, IRP, and ISP for directional guidance was qualitatively evaluated. The reason that I employed another set of experiments to determine the DP and IRP parameters was because the criteria for the repetitive signals between spatial representation and directional guidance are quite different from each other. For spatial representation, clear discernment of the number of pulses per cycle is crucial because it represents the distance from the wheelchair to a detected obstacle. In contrast, for directional guidance, to distinguish the actual number of



(a) Spatial representation signal.



(b) Directional guidance signal.

Figure 5.1: Parameters of vibrotactile signal pattern.

pulses per cycle is not so important as long as it feels somewhat different from the signal patterns for spatial representation. More importantly, the DP/IRP characteristic accompanied with the ISP for directional guidance needs to be tuned to produce a cutaneous illusion that presents a "stream" of vibrotactile signals along a designated direction.

The levels of stimuli were set to 15, 60, and 105 milliseconds for each parameter, which makes 27 combinations. The major criteria were measured by three aspects: signal orientation, signal continuity, and signal strength, each of which is assigned the scaling values of 1 (poor), 2 (mediocre), 3 (good), and 4 (excellent). Signal orientation describes how clearly human tactile perception recognizes the locations of both tactors. Signal continuity indicates how continuously the two stimuli are presented in tactile perception. A high scaling value is given if the two stimuli give human tactile perception an illusion of an elongated single stimuli along the two tactors instead of two independent stimuli. Signal strength indicates the magnitude of the presented stimulus in tactile perception, and the scaling values are given proportionally to the magnitude.

5.1.3 RESULTS

RESULT 1: The result of Experiment 1 is shown in Table 5.1. Firstly, regardless the length of the IRP, if the DP is set to 50 milliseconds, human tactile perception could not effectively recognize the signals. Since a very small amount of motor vibration was visually and auditorily confirmed, this may be indicating that 50 milliseconds is not long enough for the vibrating motor to produce a vibration that human tactile perception can sense. There always be a rise time for an electric motor to start rotating stably, so the effective rotation time is even shorter than 50 ms.

From 100 ms to 200 ms for the length of the DP, the longer the length of the DP and/or of the IRP, the better the detectability of the number of pulses per cycle becomes. Specifically when the DP is 200 ms, the detectability of signal orientation is also improved.

RESULT 2: Table 5.2 shows the result of Experiment 2. Under the condition in which the DP and the IRP are set to 200 and 150 milliseconds, respectively, the detectability of the end of cycle became distinctive when the ICP was set 250 milliseconds. In the case of NP/IRP = 200/100, the

Table 5.1: Effect of the DP and IRP to distinguish three pulses per cycle for spatial representation (ICP $\approx$ 350 ms).

	DP (ms)			
IRP (ms)	50	100	150	200
50	No signal	Almost one pulse	Barely countable	Barely countable
100	Almost no signal	Barely countable	Countable	Countable
150	No signal	Countable	Countable	Countable with clear localization
200	No signal	Countable	Countable	Countable with clear localization

Table 5.2: Effect of the ICP to distinguish the end of cycle for spatial representation (NP = 200 ms, IRP = 150, 100 ms).

	DP (ms) – IRP (ms)	
ICP (ms)	200–150	200–100
150	No distinction	Vague distinction
200	Vague distinction	Clear distinction
250	Clear distinction	Clear distinction
300	Clear distinction	Clear distinction
350	Clear distinction	Clear distinction

border of clear distinction lowers to 200 milliseconds. This difference indicates that the ICP should better to be set to 100 milliseconds longer than the IRP.

RESULT 3: Table 5.3 shows the result of Experiment 3. The overall evaluation values vary depending on the combination of parameters; however, the effect of the length of DP nearly directly contributes to the evaluation of signal strength, which also supports the result of Experiment 1. The effect of the IRP is not very clear; specifically, the effects of both 15 and 60 milliseconds mostly seem identical when other conditions are set within the ranges that yield high overall scores. However, the condition of IRP = 105 ms tends to decrease the values of signal continuity. As for the

ISP, the condition of ISP = 15 ms consistently contributes better scores specifically for signal continuity. The signals which provide excellent signal continuity also give a cutaneous illusion that the vibration runs along the two tactors; however, the occurrence of the sensory saltation is inconclusive.

As a side note, I noticed another potential of the directional guidance to include another quantitative attribute such as magnitude, speed, or distance. Comparison between Test # 13 (DP:150, IRP:60, ISP:15) and # 22 (DP:150, IRP:60, ISP:15) revealed that for human tactile perception, the signal of # 22 feels somewhat "slower" than # 13, while both of them indicate the same direction and yield nearly equal scoring values for signal orientation and signal continuity.

5.1.4 SUMMARY

Upon summarizing the results, I should make a note that these experiments have been undertaken not by formal psychophysiological methods but rather by an informal screening evaluation process. Therefore, a quantitative conclusion cannot be derived and the evaluation values are bias-prone. Also, for someone sighted, the sensitivity of the skin may be significantly different from a person with severe visual impairment.

Having said that, I believe that the Vibrotactile Glove has good potential to convey necessary information, and the next stage of research might be actual human subject tests. Here is a list of my findings:

- The shortest unit of a single vibrotactile pulse for any purposes should be at least 100 milliseconds. Using pulses shorter than 100 ms may yield weak, unnoticeable, and thus impractical vibrotactile signals.

- For spatial representation, longer duration period (DP) of a pulse and longer inter repetition period (IRP) make a vibrotactile signal clear and comprehensive. The drawback of using long DP and IRP is that it will also make the signal "slow" and difficult to convey much

Table 5.3: Effect of DP, IRP, and ISP for directional guidance (ICP = 150 ms).

#	DP (ms)	IRP (ms)	ISP (ms)	Location†	Continuity†	Strength†	Overall††
1	100	15	15	3	2	1	6
2	100	15	60	3	2	1	6
3	100	15	105	3	1	1	5
4	100	60	15	3	3	2	8
5	100	60	60	3	2	1	6
6	100	60	105	3	1	1	5
7	100	105	15	1	2	1	4
8	100	105	60	2	2	1	5
9	100	105	105	2	1	1	4
10	150	15	15	4	4	3	11
11	150	15	60	4	4	3	11
12	150	15	105	2	3	2	7
13	150	60	15	3	4	3	10
14	150	60	60	3	3	3	9
15	150	60	105	3	2	3	8
16	150	105	15	3	3	3	9
17	150	105	60	3	2	3	8
18	150	105	105	3	1	3	7
19	200	15	15	2	3	4	9
20	200	15	60	2	3	4	9
21	200	15	105	2	3	4	9
22	200	60	15	4	3	4	11
23	200	60	60	4	2	4	10
24	200	60	105	4	2	4	10
25	200	105	15	2	2	4	8
26	200	105	60	2	2	4	8
27	200	105	105	2	2	4	8

† 1 (poor), 2 (mediocre), 3 (good), 4 (excellent)

†† Overall score = Location score + Continuity score + Strength score

information within a short cycle. Currently I find that the period between 100 and 200 milliseconds is appropriate for the DP, and the period between 100 and 150 milliseconds is suitable for the IRP.

- Inter cycle period (ICP) should be typically set to 100 milliseconds longer than the IRP.
- For directional guidance, the appropriate range of DP is between 100 and 200 milliseconds; the range of IRP, 15 and 60 ms; and the ISP, approximately 15 ms. The signals which provide excellent signal continuity also give a cutaneous tactile illusion that the vibration runs between the two tactors; however, the occurrence of the sensory saltation is inconclusive.
- As a side note, if other parameters for directional guidance are set within appropriate ranges, a longer DP generates a cutaneous illusion that the signal runs "slowly" compare to the signal with the short DP. This characteristic could be used to represent another quantitative attribute such as distance to the goal or desireable speed accompanied with the directional guidance.

5.2 CURRENT STATUS

Our SA wheelchair project is currently a work in progress. Several coworkers have also been involved with the development of other parts of the project, and integration needs all parts to be completed. As for the computer vision related behaviors, the `Doorway detection` behavior is complete, but the `Range finder` behavior still needs more work. The `Sonar reader` behavior has been nearly completed but is somewhat buggy. Currently we are working on fixing the bug. The `Path planner` behavior was just conceived and needs to be implemented to function along with the related behaviors.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter presents the conclusion of this thesis and thoughts on the development of the system with some suggestions for the future work.

6.1 CONCLUSION

Our SA wheelchair prototype which consists of a stationary ranging module, a motorized vision module, and a tactile feedback module has been presented. The goal of our SA wheelchair project is to provide perceptual and navigational capabilities to wheelchair-bound users with severe vision impairment.

Our SA wheelchair is particularly characterized by the two modules: the motorized vision module and the tactile feedback module. The motorized vision module consists of lowcost components but has exhibited versatile capabilities including laser ranging and other computer-vision techniques such as doorway detection. I have presented a novel tactile feedback module, a Vibrotactile Glove, which consists of an array of vibrotactors and conveys various information such as presence of obstacles and recommended orientation to the users without interference of their auditory perception.

A custom BBC architecture consisting of Behavior Cells and Behavior Networks components has been proposed to address the problems when the users with vision impairment need to operate wheelchairs. In this thesis, I have presented the schemes for the user to avoid obstacles and to go through a narrow doorway. However, these are only a small subset of the problems that exist, and there are a great number of problems challenging researchers. Particularly, drop-off avoidance,

distinction between humans and stationary objects, and local mapping capability can be the next topics of interest.

6.2 FUTURE WORK

In this section, aside from the rest of the work needed to be completed (Section 5.2), I summarize some machine/computer vision techniques other than depth measurement for possible future work. The operations contain image processing, feature extraction, stereoscopic vision, and feature tracking. Short descriptions of those operations are summarized as follows (Jain et al., 1995; Davies, 2005):

Image Processing: The purpose of image processing is to eliminate unnecessary details and enhance the area of interest in the image for computational efficiency. In this project, we would specifically focus on the shape (points and lines) of the objects.

Feature Extraction: In feature extraction the area of interest is retrieved from a given image, such as a doorframe and a corridor intersection. This is a cognitive operation, so we will provide heuristic schemata that should help to find such feature points and lines within a reasonable amount of time. In this project we would specifically aim to identify a line with a specified inclination with the edge of the doorway or with the corner of the corridor.

Stereoscopic Vision: The purpose of stereoscopic vision is to obtain the accurate spatial relationship between the target object and the wheelchair. If any point in the scene is visible in both cameras, it will be projected to a pair of points (a *conjugate pair*) in the two cameras. Based on the displacement between the positions of the points (*disparity*) and the known camera factors (such as the distance between the two cameras and the focal length of the camera), the distance and angle of the point would be calculated. The stereoscopic vision approach generally includes a correspondence problem (how to identify a conjugate pair). In this project, we would attempt to implement a behavior employing the stereo vision technique that only correlates the features that are retrieved by the feature extraction operation.

Feature Tracking: Coupled with a servomechanism, a behavior employing the feature tracking operation would work as an autonomous agent. This operation enables the behavior to select the feature of interest from the candidate features provided from the feature extraction operation and track the horizontal movements of the feature frame by frame. Once a feature is "locked on," it sends the angle data that rotates the servomotor for the counter-movement of the feature. Feature tracking aims to lock on the target in the center of the view. If the feature moved too fast to follow timely, the amount of the counter-movement could be predicted based on differential equations.

BIBLIOGRAPHY

Argyros, A. A., Georgiadis, P., Trahanias, P., and Tsakiris, D. P. (2002). Semi-autonomous navigation of a robotic wheelchair. *Journal of Intelligent and Robotic Systems*, 34:315–329.

Arkin, R. C. (1998). *Behavior-Based Robots*. The MIT Press, Cambridge, Massachusetts.

Borenstein, J. and Koren, Y. (1995). Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 11(1):132–138.

Borgolte, U., Hoyer, H., Bühler, C., Heck, H., and Hoelper, R. (1998). Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22(3):233–253.

Brooks, R. A. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA(2):14–23.

Brooks, R. A. (1991a). Integrated Systems Based on Behaviors. *SIGART Bulletin*, 2(4):46–50.

Brooks, R. A. (1991b). Intelligence without Representation. *Artificial Intelligence*, 47:139–159.

Brooks, R. A. (1992). How to Build Complete Creatures Rather than Isolated Cognitive Simulators. In Vanlehn, K., editor, *Architectures for Intelligence*, pages 225–239. Lawrence Erlbaum Associates, Hillsdale, NJ.

Cholewiak, R. W. and Collins, A. A. (2000). The generation of vibrotactile patterns on a linear array: Influences of body site, space, and time. *Perception & Psychophysics*, 62(6):1220–1235.

Crisman, J. D. and Cleary, M. E. (1998). Progress on the deictically controlled wheelchair. In Mittal, V. O., Yanco, H. A., Aronis, J. M., and Simpson, R. C., editors, *Assistive Technology and Artificial Intelligence*, volume 1458 of *Lecture Notes in Computer Science*, pages 137–149. Springer.

Davies, E. R. (2005). *Machine Vision: Theory, Algorithms, Practicalities, Third Edition*. Morgan Kaufmann.

Douglass, J. and Ryan, M. (1987). A pre-school severely disabled boy and his powered wheelchair: A case study. *Child Care, Health Development*, 13:303–309.

Egerstedt, M. (2000). Behavior based robotics using hybrid automata. In Lynch, N. A. and Krogh, B. H., editors, *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Proceedings*, volume 1790 of *Lecture Notes in Computer Science*, pages 103–116, Pittsburgh, PA, USA. Springer-Verlag. March 23-25.

Eimer, M. and Forster, B. (2005). Cutaneous saltation within and across arms: A new measure of the saltation illusion in somatosensation. *Perception and Psychophysics*, 67(3):458–468.

Geldard, F. and Sherrick, C. E. (1972). The cutaneous "rabbit": A perceptual illusion. *Science*, 178:178–179.

Geldard, F. A. (1975). *Sensory Saltation: Metastability in the Perceptual World*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Geldard, F. A. (1985). The mutability of time and space on the skin. *Journal of the Acoustical Society of America*, 77:233–237.

Gemperle, F., Ota, N., and Siewiorek, D. P. (2001). Design of a wearable tactile display. In *Fifth International Symposium on Wearable Computers (ISWC '01)*, pages 5–12. IEEE Computer Society.

Gomi, T. and Griffith, A. (1998). Developing intelligent wheelchairs for the handicapped. In Mittal, V. O., Yanco, H. A., Aronis, J. M., and Simpson, R. C., editors, *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, volume 1458 of *Lecture Notes in Computer Science*, pages 150–178, New York. Springer.

Hobson, D. A. and Brubaker, C. E. (1998). Final report: Summary of activities aug. 1, 1993–nov. 30, 1998. Technical report, Rehabilitation Engineering Research Center on Wheelchair Mobility, University of Pittsburgh.

Ishiguro, A., Kondo, T., Watanabe, Y., and Uchikawa, Y. (1996). Immunoid: An immunological approach to decentralized behavoir arbitration of autonomous mobile robots. In Voigt, H.-M., Ebeling, W., Rechenberger, I., and Schwefel, H.-P., editors, *PPSN*, volume 1141 of *Lecture Notes in Computer Science*, pages 666–675. Springer.

Jain, R. C., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. McGraw-Hill.

Kilgard, M. P. and Merzenich, M. M. (1995). Anticipated stimuli across skin. *Nature*, 373:663.

Kiryati, N., Eldar, Y., and Bruckstein, A. (1991). A probabilistic Hough transform. *Pattern Recognition*, 24:303–316.

Kleeman, L. (1999). Fast and accurate sonar trackers using double pulse coding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, Korea*, pages 1185–1190.

Kondo, T., Ishiguro, A., Watanabe, Y., Shirai, Y., and Uchikawa, Y. (1998). Evolutionary construction of an immune network-based behavior arbitration mechanism for autonomous mobile robots. *Electrical Engineering in Japan*, 123(3):1–10.

Laird, J. E. and Rosenbloom, P. S. (1990). Integrating, execution, planning, and learning in Soar for external environments. In Dietterich, T. S. W., editor, *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 1022–1029, Hynes Convention Centre. MIT Press.

Levine, S. P., Bell, D. A., Jaros, L. A., Simpson, R. C., Koren, Y., Member, S., and Borenstein, J. (1999). The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7:443–451.

Maes, P. (1989). The dynamics of action selection. In *Proceedings of IJCAI-89*, pages 991–997, Detroit, Michigan.

Matarić, M. J. (1991). Behavioral synergy without explicit integration. *SIGART Bulletin*, 2(4):130–133.

Matarić, M. J. (1992). Behavior-based control: Main properties and implications. In *IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, pages 46–54, Nice, France.

Matarić, M. J. (1997). Behavior-based control: examples from navigation, learning, and group behavior. *Journal of Experimental and Theoritical Artificial Intelligence*, 9(2-3):323–336.

Miller, D. P. and Slack, M. G. (1995). Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2(1):77–88.

Moravec, H. (1989). Autonomous Mobile Robot Research Using the HErmies-III Robot. In *IORS International Conference on Intelligent Robot and System*, Tsukuba, Japan.

Moravec, H. P. (1988). Sensor Fusion in Certainty Grids for Mobile Robots. *AI Magazine*, 9(2):61–74.

Na, Y.-K. and Oh, S.-Y. (2003). Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification. *Autonomous Robots*, 15(2):193–206.

Nicolescu, M. N. and Matarić, M. (2000a). Extending behaivor-based systems capabilities using an abstract behavior representation.

Nicolescu, M. N. and Matarić, M. J. (2000b). Deriving and using abstract representation in behavior-based systems. In *AAAI/IAAI*, page 1087. AAAI Press / The MIT Press.

Nicolescu, M. N. and Matarić, M. J. (2002). A hierarchical architecture for behavior-based robots. In Gini, M., Ishida, T., Castelfranchi, C., and Johnson, W. L., editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*. ACM Press.

Nilsson, N. (1984). Shakey the robot. Tech Note 323, AI Center, SRI International.

Odor, J. and Watson, M. (1994). Learning through smart wheelchairs: A formative evaluation of the effective use of the call centre's smart wheelchairs as part of children's emerging mobility, communication, education and personal development. Technical report, Final Report to The Nuffield Foundation and the Scottish Office Education Department.

Patel, S., Jung, S.-H., Ostrowski, J. P., Rao, R., and Taylor, C. J. (2002). Sensor based door navigation for a nonholonomic vehicle. In *IEEE International Conference on Robotics & Automation*, pages 3081–3086, Washington, DC.

Paulsson, K. and Christoffersen, M. (1989). Psychosocial aspects of technical aids - how does independent mobility affect the psychosocial and intellectual development of children with physical difficulties? In *Proc. 2nd Int. Conf. of the Rehabilitation Engineering Society of North America, Ottawa*, pages 282–285.

Payton, D. W., Keirsey, D., Kimble, D. M., Krozel, J., and Rosenblatt, J. K. (1992). Do whatever works: A robust approach to fault-tolerant autonomous control. *Applied Intelligence*, 2:225–250.

Pranghofer, M. (1996). Wheels and white canes tips for helping blind wheelchair users. *Braille Monitor*, 39(3):122–124.

Rönnbäck, S., Piekkari, J., Hyyppä, K., Berglund, T., and Koskinen, S. (2006). A semi-autonomous wheelchair towards user-centered design. In Miesenberger, K., Klaus, J., Zagler, W. L., and Karshmer, A. I., editors, *ICCHP*, volume 4061 of *Lecture Notes in Computer Science*, pages 701–708. Springer.

Simpson, R., LoPresti, E., Hayashi, S., Guo, S., Ding, D., Ammer, W., Sharma, V., and Cooper, R. (2005). A prototype power assist wheelchair that provides for obstacle detection and avoidance for those with visual impairments. *Journal of NeuroEngineering and Rehabilitation*, 2(1):30.

Surmann, H., Nüchter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198.

Tan, H., Lim, A., and Traylor, R. (2000). A psychophysical study of sensory saltation with an open response paradigm. In *Ninth (9th) International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*.

Tan, H. Z. and Pentland, A. (1997). Tactual displays for wearable computing. In *The International Symposium on Wearable Computers*, pages 84–89.

Traylor, R. and Tan, H. Z. (2002). Development of a wearable haptic display for situation awareness in altered-gravity environment: Some initial findings. In *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 159–164.

United Nations (2005). *World Robotics 2005*. United Nations Publication, Geneva.

van Erp, J. B. and van Veen, H. A. (2004). Vibrotactile in-vehicle navigation system. *Transportation Research Part F: Traffic Psychology and Behaviour*, 7(Issues 4-5):247–256.

Verburg, G., Balfour, L., Snell, E., and Naumann, S. (1991). Mobility training in the home and school environment for persons with developmental delay. In *Final Report to Ontario Mental Health Foundation and Ministry of Community and Social Services' Research and Program Evaluation Unit*.

Verrillo, R. T. (1983). Vibrotactile masking: Effects of one- and two-site stimulation. *Perception and Psychophysics*, 33(4):379–387.

Yanco, H. A. (1998). Wheelesley: A robotic wheelchair system: Indoor navigation and user interface. *Lecture Notes in Computer Science*, 1458:256.

Yen, J. and Pfluger, N. (1995). A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation. *IEEE Trans. on Systems, Man, and Cybernetics*, 25(6):971–978.

Zelek, J. S., Bromley, S., Asmar, D., and Thompson, D. (2003). A haptic glove as a tactile-vision sensory substitution for wayfinding. *Journal of Visual Impairment & Blindness*, 97(10):1–24.

APPENDIX A

MOTOR-ARRAY CIRCUIT DIAGRAM

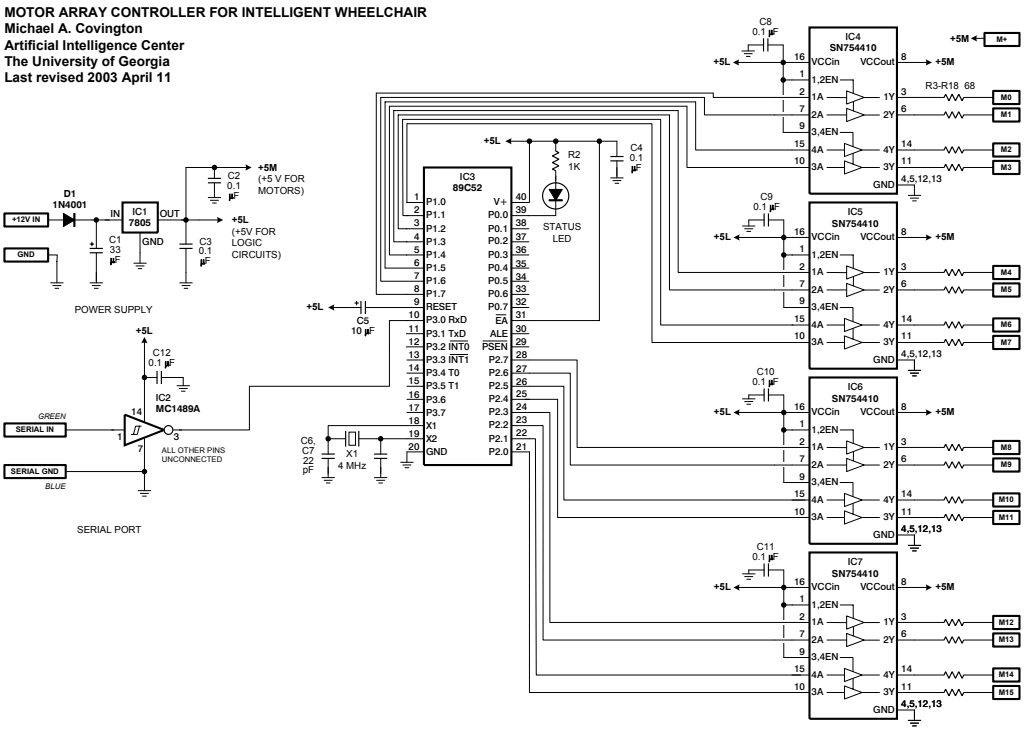


Figure A.1: Circuit diagram of the motor-array controller (designed by Michael A. Covington).

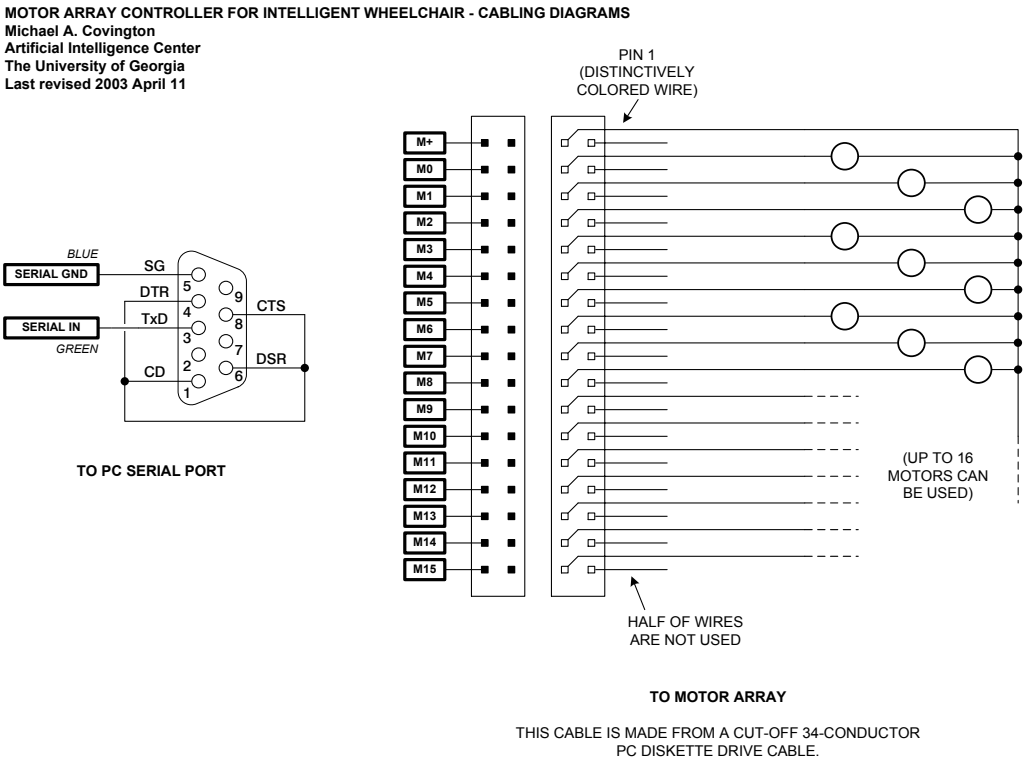


Figure A.2: Pinouts diagram of the motor-array controller (designed by Michael A. Covington).