Autonomous Robots, 2, 77–88 (1995)

Design and Testing of a Low-Cost Robotic Wheelchair Prototype

DAVID P. MILLER
MITRE Corporation, MS Z421, 7525 Colshire, McLean, VA 22102
dpmiller@mitre.org

MARC G. SLACK
MITRE Corporation, McLean, VA 22102
slack@mitre.org

Abstract. Many people who are mobility impaired are, for a variety of reasons, incapable of using an ordinary wheelchair. In some instances, a power wheelchair also cannot be used, usually because of the difficulty the person has in controlling it (often due to additional disabilities). This paper describes two low-cost robotic wheelchair prototypes that assist the operator of the chair in avoiding obstacles, going to pre-designated places, and maneuvering through doorways and other narrow or crowded areas. These systems can be interfaced to a variety of input devices, and can give the operator as much or as little moment by moment control of the chair as they wish. This paper describes both systems, the evolution from one system to another, and the lessons learned.

Keywords: assistive robotics, behavior control, navigation, low-cost robots

1 Introduction

Semi-autonomous mobile robots have traditionally been directed at performing delivery, surveillance, or similar tasks. As part of an assistive technology system, mobile robots can play a very different role. This paper describes the design and testing of some prototype robotic wheelchairs. Unlike most other robotic systems, wheelchairs are almost constantly getting new navigation input from their users. This both simplifies and complicates the navigation process. This application revolves around human interface and cost issues, two areas that are of lesser importance in more traditional robotic applications.

The following section will motivate the need for work on robotic wheelchairs. Subsequent sections will describe the hardware and software used to develop each of two wheelchair prototypes: *Tin Man I & II*. The testing and lessons learned for each robot will also be described. We will close with some observations on where this work fits in within the greater scheme of robotics.

2 The Need for Semi-Autonomous Power Wheelchairs

Power wheelchairs are traditionally used by people who do not have the upper body strength and dexterity to operate a manual wheelchair. However, operation of a power wheelchair can still be a difficult and demanding task for many such individuals. The operator must be able to accurately sense their environment, recognize hazards, and be able to translate their mobility desires into continuous joystick commands for the chair.

A variety of user interfaces have been created to aid people, in using power wheelchairs, that lack the ability to operate a traditional joystick. In most instances, this involves repositioning the joystick and adding a mechanical attachment to the end of the joystick so that it may be operated by a person's elbow, chin, or tongue. In some cases, an eye tracker is used, or options are flashed one at a time on a display, and the operator makes their selection by pressing a single switch, blinking, or altering their breathing pattern. But in all these instances, the command options are basically the same: move forward or backwards, turn left or right.

For most operators who cannot use the traditional joystick, and even for many who can, operating the chair is a tedious, unnatural and stressful activity. Their limited bandwidth of interaction with the chair limits the speed at which they can safely travel, and often puts them in situations that are hazardous for themselves and for the objects in their environment. Additionally, many potential power-wheelchair users have limited visual acuity, or must be seated in a way that limits their forward vision. None of the traditional interfaces address these vision related problems.

The Tin Man chairs, as described throughout this paper, use a variety of sensors and a microprocessor to aid the user in the operation of the wheelchair. Tin Man can automatically sense most obstacles and maneuver about them. It can maintain a heading without user input, and can navigate along corridors and through doorways. Enhancements currently being implemented allow the user to input a map of their home or office, and direct the chair by simply specifying the desired room or location. The features of Tin Man allow the user to control the chair using much less frequent interaction than with traditional power chairs, and get improved performance over traditional chairs. The robots greatly improve the utility of the chair for people who cannot issue continuous and accurate heading and speed commands to the chair. Perhaps most important, Tin Man was designed to be inexpensive (Miller & Grant, 1994; Miller & Slack, 1994). Unlike most other work on robotic wheelchairs, which use custom mechanics (e.g., Gelin et al., 1993) Tin Man is built on top of a commercial power wheelchair. Both Tin Man prototypes could be produced and sold at prices well within the price range of normal power wheelchairs.

3 System Design of Tin Man I

Tin Man I is built on top of a commercial wheelchair (see Fig. 1) from *Vector Wheelchair Corporation.* Tin Man I has no electrical interface between the chair's controls and the robot's computer. Instead, there is a mechanical interface. The control computer controls two servo motors that are mechanically linked to the standard joystick that comes with the chair (see Fig. 2). The user enters their commands through an input device (usually another joystick). The commands and sensory data are processed by a commercial micro-controller based around the Motorola 68HC11 processor (see Fig. 3). The micro-controller then commands the servo motors which move the main joystick on the chair. The joystick position is read by a standard wheelchair controller which sends control signals to the two drive motors.

Tin Man I has five types of sensors:

- Drive motor encoders;
- Contact sensors;
- IR proximity sensors;
- Sonar range finders;
- Fluxgate compass.

The drive motor encoders, after gearing, deliver a resolution of 6.725 tics per inch. With the encoder resolution and the robot's wheel separation, theoretically the robot's orientation can be known to a resolution better than 0.01 radians. Unfortunately, because of the width of the drive wheels, slippage, wheel distortion, etc., it appears that the robot is only able to turn within ±10% of the commanded amount. As a result, dead reckoning errors can grow quickly.

Fig. 1. Tin Man I.

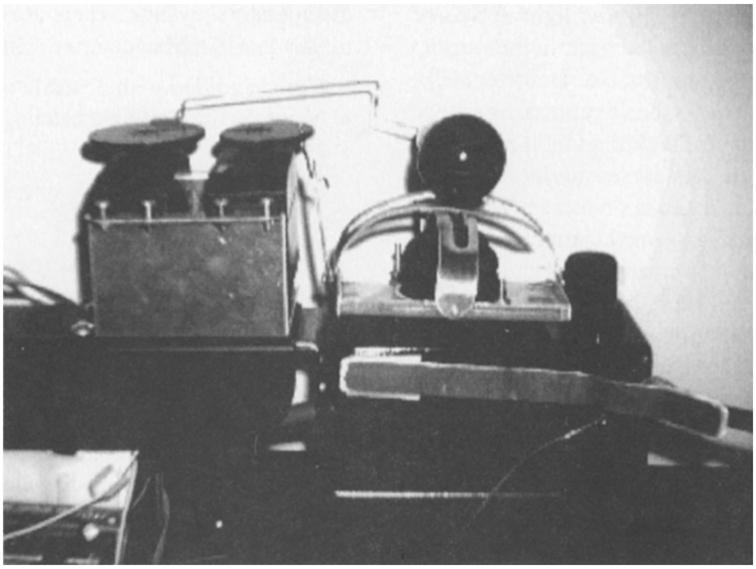

Fig. 2. Mechanical interface to joystick.

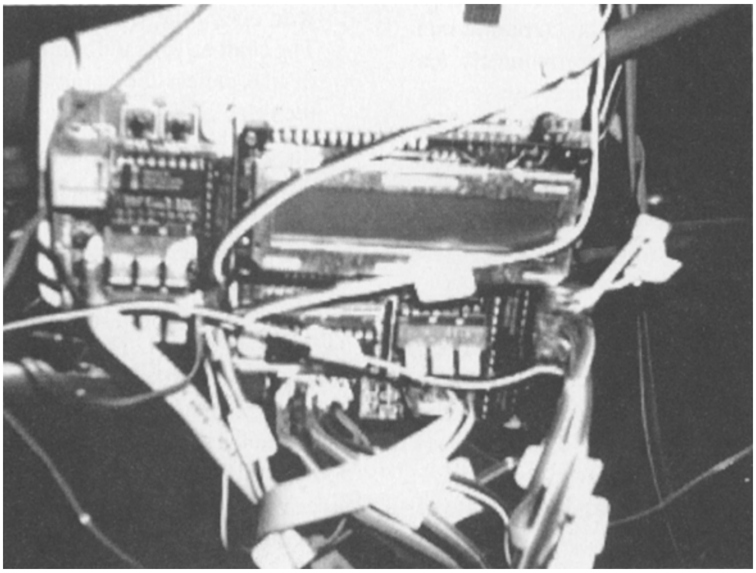

Fig. 3. The Tin Man I micro-controller.

There are eight contact sensors on the robot which are used as whiskers. Each sensor is made from a resistive strip approximately ten centimeters in length. As the strip is bent, its resistance changes, and the degree of the bend can be calculated from the current flow through the strip. Two of the strips are mounted on each side of the robot, one in front of the wheel and the other in front of the armrest. The remaining four sensors are mounted on the front. These sensors are enclosed by a sheet of foam rubber. The foam fills the gaps between the sensors. If the foam contacts an obstacle, its shape is distorted causing the sensing strips to bend.

There are four IR proximity sensors distributed evenly along the front and sides of the robot. These sensors emit a coded beam of infrared light. If an object is nearby, the light is reflected back to the sensor, indicating the presence of an obstacle. Unfortunately, these sensors are very albedo sensitive, making detection of dark objects more difficult than light objects.

To compensate for the albedo sensitivity of the IR sensors, there are six sonar range finders on Tin Man. Each sonar has a resolution of one centimeter, a minimum range of thirty-five centimeters, and a maximum range of five meters. It takes each sensor approximately two-hundred milliseconds from the time it is activated until it settles on a reading. Due to port limitations on the 6811 processor, all of the sonars are timeshared into the same timing port using a simple round robin scheme. Each sonar can be activated or deactivated in software, and only the active sonars are polled. Because of these limitations, if all the sonars are active, it can take over one second between readings from a specific sonar.

The fluxgate compass is a standard compass meant to be used in an automobile. The coils that control the display are directly wired to two of the analog to digital ports on the micro-controller. The computer can distinguish changes in heading of approximately ten degrees. While not adequate for accurately traversing long, open distances, this is sufficient resolution for navigating along streets and in building corridors where the environment can help keep the robot on course.

3.1 Software Design for Tin Man I

The software for Tin Man I is written in IC: an interactive, multi-tasking dialect of the C language. Each sensor type has its own asynchronous process which monitors those sensors. With the exception of the sonars, every sensor is polled at least 5 Hz. The maximum safe speed of the chair is governed by this sensor refresh rate combined with the deceleration rate of the chair.

All of the sonars are multiplexed through a single port and into a single timing register. It takes several ultrasonic pulses to ensure a reliable distance reading from the sonar, and from the time the first pulse starts, till the last echo returns, a single sonar controls the timing register. A single sonar can be read at 3–5 Hz. Most modes of the robot use at least three active sonars leading to an update rate of approximately 1 Hz.

In the manual operation mode, the operator gives their input through a joystick. The micro-controller reads the joystick and issues servo-motor commands to cause the chair's joystick to copy the movements of the operators joystick. There are three semi- automatic modes that Tin Man can run. They are:

- Human guided with obstacle override;
- Move forward along a heading; and
- Move to X, Y.

In all three modes, the same priority scheme holds true:

1. If a contact sensor reads true, the chair moves away from the point of contact;
2. If a proximity sensor reads true (and contact sensors do not) then the chair turns away from the direction of the sensor reading true (if both front sensors read true then the chair will back up, if both side sensors read true then the chair will go straight, slowly);
3. If a sonar senses an obstacle less than 60 cm away in front or behind then the chair will not move forward or backward. If a sonar senses an obstacle less than 1 m away, then the chair will turn away from the direction of the obstacle;
4. The robot follows the designated heading or towards the designated way point, unless this conflicts with one of the sensor rules listed above;
5. The chair follows the commands from the user input device, unless the commands conflict with one of the rules above.

When operating in the obstacle override mode, the chair follows the user's instructions except when a nearby obstacle is detected. When an obstacle is detected, the chair will modify its heading, following a safe heading that is as close as possible to the heading being input by the user. If the user puts in a stop, the chair will stop. This is probably the most common mode to run the chair. It is especially useful when training someone to use a power chair. It is also helpful when maneuvering in tight spaces or through narrow doorways. For an operator with slow reflexes or limited perception, this mode allows the chair to be operated at a speed much faster than would otherwise be safe. In all cases, it greatly reduces the risk of impact with an obstacle, and the severity of an impact should one occur.

The move forward along a heading mode is most useful for someone who has a very limited amount of bandwidth for input to the chair. The chair can be spun until the desired heading is reached. When at the desired heading, the chair moves forward, avoiding or maneuvering about obstacles as needed. If the chair is pointed in the general direction of a doorway, it will autonomously maneuver through the doorway. If pointed down a hallway, the chair will continue down the hallway until blocked. The only control needed by the user is to: put the chair into this mode; designate the proper heading; tell the chair when to stop. Currently all three commands are executed by pressing a button at the desired time, but they could as easily be commanded by monitoring eye blinks or a sip/puff controller.

The move to X, Y position mode allows the user to specify a specific position in absolute coordinates for the chair to go to. A heading to the desired point is calculated, the chair turns to that heading and then moves forward much as in the previous mode. Obstacles are avoided, and after each deviation, the chair heads straight for the goal location. This mode is meant to be used only in situations where there is a mostly clear path towards the goal location. To go to locations that involve going around corners, down corridors, etc., it is best to input a series of locations representing waypoints for the robot to follow. This mode of operation is intended for our future work at task level control of the robot discussed in section 5.

3.2 *Testing of the First Prototype*

Tin Man I was run extensively over a limited set of indoor environments. Testing of Tin Man I was done primarily by the authors[1]. The environments were "prepared" in that hazards that were known to be beyond the robot's capabilities had been removed. In particular, the environment contained no: drop-offs; obstacles less than 10 cm in height; low-overhangs and no vertical obstacles less than 10 cm in cross-section.

This testing showed that the robot's behavior tended to be governed by the sonars, in open areas, and by the contact sensors in tight spaces. The IRs played very little role, except when going down a hallway; where the IRs tended to keep the robot following the wall. Reliance on the contact sensors in tight spaces (e.g., doorways) was sometimes problematic. Mounting contact sensors so they could be triggered in any direction, protect the frame of the chair, and not occasionally have the sensors scraped off the robot, proved to be very difficult, especially along the side of the robot.

The IR sensors were built from discrete components including the "Sharp Box" commonly used in robot kits. While the authors had used the same design for IR sensors with great success on small robots, these sensors were not very effective on this robot. The reason for the inadequate performance is simply that the sensor's emitter range and focus are insufficient for the size, speed, and deceleration rate of the chair.

Despite the relatively slow speed of operation (due to the slow sonar sample rate) and the occasional destruction of a contact sensor, Tin Man I gathered a great deal of interest. We found the response, especially from representatives of the potential user community, to be very encouraging.

4 Design of Tin Man II

In building Tin Man II, we wanted to: decrease reliance on contact sensors; modify the user interface to reflect the needs of the user community; increase the operating speed. We also wanted to create a system that was better packaged, so that it could be more thoroughly tested by users; without fear of accidental rewiring.

4.1 *Tin Man II Hardware*

Like the first wheelchair, Tin Man II is built on top of *Vector Wheelchair Corp.* wheelchair. This time a model was chosen with more cutouts on the side panels. These extra openings simplified wiring and sensor mounting (see Fig. 4). We also changed processors, moving to a *Motorola* 68332 on a *Vesta Technologies Inc.* board. A supplemental board was also used to handle digital to analog and analog to digital conversions.

With the new digital to analog converters, we were able to electrically interface the micro-controller to the chair's wheelchair controller. The joystick was interfaced to the computer through the analog to digital ports. The joystick box was also used to house an override switch and status lights (see Fig. 5).

To reduce the dependence on contact sensors, Tin Man II uses twelve IR proximity sensors made by *SunX*. These sensors are highly directional, have an adjustable range, and a maximum range of about 1.5 meters when reflecting against a standard reflectance gray card. The performance against off-white walls is similar. The IRs are placed so that the activation pattern is different when the edge of the chair just clears an obstacle, versus when an obstacle is too close to the side to turn in that direction. As a result, the chair can be made to navigate through doors and around most obstacles relying solely on the IR sensors.

Seven sonars are used on Tin Man II. These units are of the same type as that used on Tin Man I, however on Tin Man II each sonar has its own dedicated port. Therefore, all the sonars can be read at approximately 4 Hz. These sonars have approximately a 7° dispersion angle and, due to the improved processing, an increased maximum range of fifteen meters. They give excellent readings off of perpendicular walls, but are subject to

Fig. 4. Tin Man II.

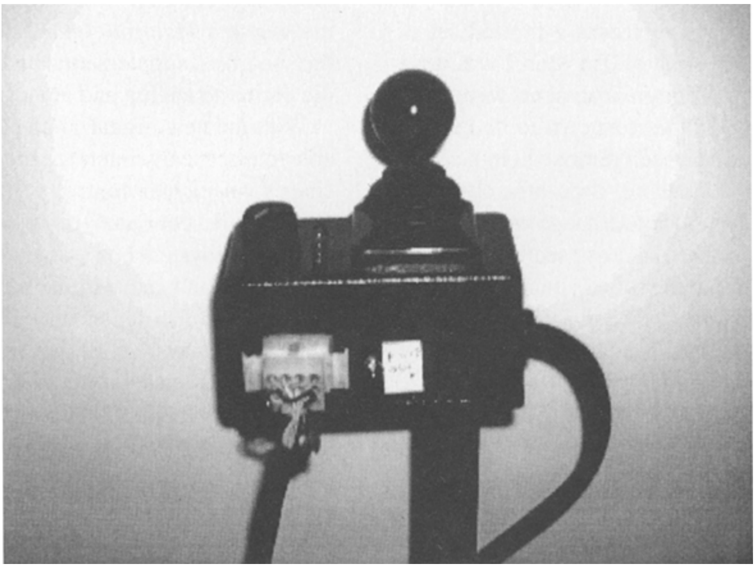

Fig. 5. Tin Man II joystick with override and lights.

specularity when aimed at a significant angle off the wall's normal. They are also subject to signal absorption when the sonar ping encounters a soft surface such as clothing. In Tin Man I, rather than being the primary sensor modality, the sonars act as backup to the IRs.

Tin Man II has a front contact bumper. This bumper is constructed from PVC pipe, and has mounted to it a folded aluminum sheet. Inside the aluminum sheet are several pressure switches. When contact is made, the aluminum is compressed and the switch is activated. The entire front of the bumper is coated with a thick sheet of stiff foam rubber to distribute impacts and reduce denting to the aluminum and scratching on the object being impacted (see Fig. 4). The bumper switches can discriminate between left, center and right side impacts.

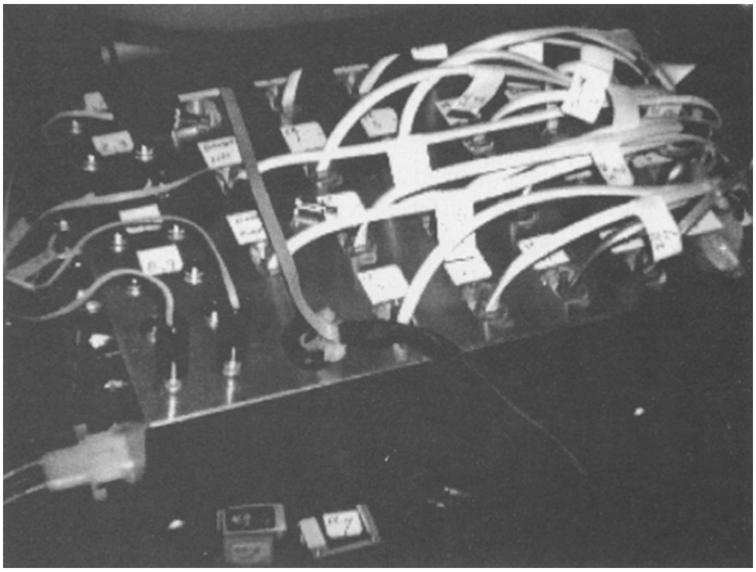

Fig. 6. Tin Man II electronics box.

The robot also has wheel encoders, of the same type as on Tin Man I. There are also two large paddle buttons mounted on the sides of the seat. These buttons are used as an emergency stop, and also as an interface for the user. The chair's operational mode may be switched by using the buttons.

In Tin Man II, all the electronics are packaged in a single box that has thirty-two ports using standard RJ-11 and mini-plug receptacles. These are used for supplying power and signal lines to each of the sensors (see Fig. 6). The resulting system greatly simplifies reconfiguring the robot's sensors.

4.2 Software Design for Tin Man II

The software for Tin Man II is, in basic functionality, the same as for Tin Man I. The code is written in ANSI C and cross complied using GCC on a Sun SPARC station. The robot operates in four basic modes:

- Human guided with obstacle override;
- Turn avoiding obstacles;
- Move forward avoiding obstacles; and
- Manual mode.

The first mode (Human guided with obstacle override) is identical to that mode in Tin Man I. The only differences are that the operator can use the original factory joystick on Tin Man II, and that the safe operating speed is more than twice as fast as it was on Tin Man I.

The next two modes can be controlled by the operator using only the two large paddle buttons. There is a green button and a red button. There are also three green lights and a red status light on the joystick box. The red button will stop the chair and turn on the red light on the joystick box. Pressing the red button a second time will change the chair's operating mode. The current mode is indicated by the blinking green light (the light blinks to indicate that the processor is alive and well). The modes cycle round-robin between human guided; turn; and move forward. Once the desired mode has been selected, the green button will enable that mode (and turn off the red status light). In human guided mode, activation means that the chair will respond (when safe) to the commands issued through the joystick.

In turn mode, operation means that the chair will start to turn clockwise. If the green button is held in the chair will turn counter clockwise. The speed with which the chair turns depends on the proximity of obstacles around the chair.

In move forward mode, the green button will start the chair moving forward. The chair will continue moving forward until the red button is pressed or an obstacle is encountered. When an obstacle is encountered the chair will move around it or back up away from it and turn slightly. Holding in the green button will cause the chair to move backwards until an obstacle is detected. Using just the turn and move forward modes, it is quite easy to move through a building without ever using the joystick.

The final mode is manual operation. This is activated using a toggle switch on the side of the joystick box. This switch effectively bypasses the 68332 processor. Commands issued through the joystick are interpreted directly by the chair's motor controller, just as when the chair came from the manufacturer. The 68332 does not have to be operational to use this mode.

4.3 *Testing the Second Prototype*

Two types of testing were performed on Tin Man II: qualitative tests were performed in a number of settings; a series of quantitative tests were done over a specific test course. This section describes these tests.

Qualitative Testing. Tin Man II was tested extensively by the authors and numerous others (none of which were mobility impaired) in an office environment. Tin Man II was found to be about twice as fast and more robust in obstacle avoidance and doorway navigation than was Tin Man I. A new user could operate the chair with only a minute of instruction. Hallway navigation in particular was robust and could be done at the speed of a fast walk (about 5 mph).

The chair was then taken to the Marriott hotel in downtown Atlanta where the President's Commission for the Employment of People with Disabilities was holding their annual conference. The hotel provided some physical challenges for the robot that are not commonly found in office environments. The conference also provided a chance for wheelchair users to review and test the robot.

The hotel's basic decoration theme was: dark wood, metallic gold trim, and Plexiglas walled walkways. The dark wood, of course, did a poor job of reflecting the IRs. The gold trim, on the other hand, was visible to the IRs at a distance of well over two meters. The result was that the chair would come very close to some walls and doorways, but give ash trays, planters, and posts an excessively wide passage. The clear walled walkways did not reflect the IRs at all, and had to be navigated almost strictly by sonar. By reducing the operating speed, these could be negotiated successfully, though not particularly smoothly or quickly.

All in all, the robot was able to handle the challenges posed by the hotel, including going through some doorways that were not considered wheelchair accessible. However, this experience did point out that it would be useful to have some way to dynamically set and store different bias sets for the various sensors. For example, the elevator floor boards were lined with gold trim. Therefore the chair avoided going to the back of the elevator, and sometimes had to be switched to manual override to allow room for other passengers to board. Being able to turn down the sensitivity of the IRs for the elevators would have been handy. However, navigation through doors, which were made of a dark wood, would have benefitted from a higher sensitivity setting on the IRs.

The members of the wheelchair user community who saw or tried the chair were in general quite favorable. They would like to see the chair made faster, or slightly more autonomous (i.e., they either want to get around quicker, or be able to concentrate on other things while they are getting there). Several people also pointed out that it would be useful for the chair to have some sort of obstacle detection display visible to the user. When the chair would not turn right, they wanted to know where the chair thought there was an obstacle. This would be especially useful for users who have limited head movement capability.

Quantitative Testing. Two aspects of performance were measured for the quantitative testing: speed and user interaction. User action was measured in three components: number of joystick moves; time-critical button pushes; and time-independent button pushes. The range of the joystick was broken into the center position, and nine forward, reverse, left and right settings. This yields 361 possible joystick settings. The joystick was considered to be in a particular setting if the user held it in the same position for at least one hundred milliseconds. Positions held for a shorter period were ignored.

A time-critical button pushing event was when a button was pushed to stop an action (stopping a turn or forward or backwards motion). Non time-critical button pushings were used to start a movement, or to switch between modes of chair operation.

The test course consisted of having an operator run the chair from the center of a room, through a doorway, down a hall to an elevator, pushing the elevator call button, going down a hall, into a room, and parking at a desk. The course length was fifty meters, though the distance traveled by the chairs tended to be a couple meters longer, depending on the run. The average results are shown in Table 1.

Each test subject did both manual and human guided runs. Operators were told to try and minimize joystick movements in both runs. The standard technique

Table 1. Operator input in different modes on test course.

Mode	Joy stick moves	Time critical buttons	Non-critical buttons	Time
Manual	81.3	0	0	1:33.9
Human guided	50.5	0	1	1:40.5
Auto forward /turn	0	8.3	19.7	2:22.3

used was to move the joystick to an end position (e.g., full forward) and leave it there making turn corrections as needed. In practice runs it was shown that joystick counts almost an order of magnitude higher were possible if the joystick was not held against one of the stops. The conscious moves of the joystick were considerably lower than the counts shown in the Table, and a longer timing constant may be more appropriate. However, in all runs, whether the operator or the computer was counting joystick moves, the manual mode required 50% more moves than the human guided system using automatic obstacle avoidance, while the time performance penalty was less than 10%.

Using a combination of the auto forward and auto turn modes, an operator typically had to make only eight real-time inputs. The input was pushing the stop button. The other nineteen button pushes were to select between turn and forward modes, select the direction of movement, and to start moving. This mode of operation was noticeably slower than the other modes. But for many people who would not be able to operate a real-time input, the greatly reduced bandwidth could be of considerable benefit. Typically the operator turned the chair towards the door and had it go forward in through the door and into the hall. Once it was in the hallway the chair would automatically align with the hallway. About half the time this was the correct alignment. The rest of the time the operator had to turn the chair around and then have it go forward. Another turn adjustment usually had to be made to move over to the side of the hall where the elevator button was located. A similar set of commands was issued to get the robot back to the room and in front of the desk. This mode was slower because the chair was idle while the operator was switching modes, and the chair turned much more slowly when in crowded areas.

5 Current Work

In order to make Tin Man a more friendly system for the mobility impaired, the current interface is being redesigned to be more intuitive and situational in nature. While the current interface that runs the user's commands through a joystick will be retained for users who desire more direct and continuous control of the chair, the new interface allowing users to direct the chair with discrete commands is being constructed. The command level interface will allow users to direct the robot to move to specific locations within a building (e.g., "go to the kitchen") or to guide the robot through a series of finer grain commands (e.g., "exit the room," "turn to the left," "follow the hall," "stop at the next door on the left"). In recurring situations the robot will "learn" the sequence of commands necessary for accomplishing the task while preserving the user's ability to guide the robot through unfamiliar environments where finer grained commands are necessary.

These enhancements to the interface have grown out of the work being done in developing an architecture for autonomous and supervised intelligent robots (Miller, 1990; Gat, 1991; Slack, 1992; Elsaesser & Slack, 1994; Yu, Slack & Miller, 1994). A premise of the work is that accomplishing new tasks is largely the process of executing a sequence of smaller tasks for which solutions are known. For example, while you may not have visited a particular building before, maneuvering in the building uses capabilities previously learned and used in other buildings: maneuvering in hallways, passing through doors, entering/exiting elevators, etc. It is this observation and supporting technology which we are transitioning to the Tin Man project, in order to create a task level user interface to Tin Man.

We are constructing a set of robotic skills to supplement these capabilities (obstacle avoidance while following user commands, following a heading, or going to a specific point) with the following skills in order to form a more sophisticated user command language:

Stop: Immediately stops the chair and drives the motors to maintain the current position. This will allow the chair to maintain its position even on hills.

Backup: The chair would retrace its previous movements up to some limit or till stopped by the user. This allows the user to quickly and easily return to a previous location or room. This would be accomplished by recording way points every time the chair changed its heading significantly and then automatically performing a series of X, Y moves to the list of way points, in reverse order.

Backtracking: This is similar to the backup command, however the chair will turn around and be driven in a forward direction.

Wall Following: The chair would align itself to the wall (selected by the user) and move along that wall at a constant distance (while avoiding obstacles) until terminated by the user. This would be implemented by servoing (when no other obstacles were closer) to a preset distance on the side sonars.

Follow Hall: This is a variant of the follow wall which assumes that there to be two walls to follow simultaneously.

Count Doors/Intersections: This will use the chair's side looking sonars and the encoder readings to detect discontinuities in the walls while the robot is in either the follow hall or follow wall mode. This information is used to form task termination conditions (e.g., follow hall until the third door/intersection on the right).

Pass through door: This skill assumes that there is a door located approximately in front of the robot. It then uses the sensors to move through the passage stopping when the chair is fully through the doorway.

Exit room: This is a more subtle operation. If the chair has a memory of entering the current room, it will attempt to exit the room by moving back to the point where the chair had entered the room and then execute the pass through door command. However, if the chair has no memory of the location of the room's exit or which of several exits that the chair should use in exiting the room, it will isolate the direction of the exit by querying the user. The form of the user input will depend on the user's interface with the chair. The form of a user's input into the system is dependent upon their capabilities and the sophistication of the information interface between the user and the chair.

Docking: The chair would approach an object in front, slow down and stop at first contact. If the object was a table or a desk, the chair would slow and then stop when it was a prespecified distance under the object.

Automated Safing: These functions would prevent the chair from moving too quickly over bumpy surfaces or going over terrain that might cause tipover. Both functions could be implemented using roll and pitch "3-position" sensors.

These low-level skills are being implemented on the microcontroller we've installed on the chair. In order to accomplish the sequencing and planning aspects of this work, we are interfacing the microcontroller with an *Apple Powerbook.* The Powerbook runs the task sequencing (Firby, 1989) and planning processes which are currently implemented in Lisp. The sequencer communicates with the microcontroller turning on and off the skills depending on the context and task at hand. The microcontroller sends asynchronous signals back to the sequencer, allowing the sequencer to identify the success or failure of the current task. The sequencer configures the microcontroller for the current task and the microcontroller notifies the sequencer of changes in the state of the world. This allows the two systems to remain synchronized while executing asynchronously.

The Powerbook also serves as the basis for the user's interface to the system. It is used for displaying maps, querying the user for information, and for acquiring commands from the user. While the nature of the interface will change from user to user, the fact that the Powerbook is a general purpose computer capable of multimedia allows for a great deal of flexibility in the way that information is gathered from, and presented to, the user.

With the introduction of these autonomous skills it becomes possible to direct the chair to carry out sequences of commands which work together to perform the more complicated tasks of moving to specific locations in the world (e.g., "take me to the kitchen"). The interface we are developing allows the user to install new software modules to improve or extend the chair's capabilities. For example, initially we are directing our efforts at the control software necessary for maneuvering around a standard office building. Later we will be working at extending the system to handle household situations and then extending the system further to operate in a city sidewalk environment. At each stage, a user would simply have to install the software upgrades (and perhaps some additional sensors) to their system in order to have their chair perform in these newly supported environments.

6 Related Work and Contribution

The basic capabilities of the Tin Man robots are not new or unique to the field of mobile robots. In large part, the control systems for these robots are taken directly from the system described in (Miller et al., 1992). In that robot, *Rocky III*, the control system guided a prototype planetary rover through natural terrain. While the environments and the mobility systems are different, the basic navigation strategy is the same. The algorithms for defining the low-level skills that the robot must exhibit, and the strategy for sequencing those skills is identical in the Tin Man robots and Rocky III.

The idea of mixing control between a users input through a joystick, and the navigation system of a reactive robot is also established in the literature. Connell (Connell & Viola, 1990) describes a robot called *Mr. Ed* which can be ridden by a person. While the point of Connell's system was to make a robot that behaved more like a horse than a wheelchair, the concept is similar.

In the late '80s, a semi-robotic wheelchair (Nisbet, Odar & Loudon, 1988) was used to help motivate children confined to a wheelchair. In (Bell et al., 1993), Bell describes an ongoing research program that has produced a robotic wheelchair with capabilities similar to that of the Tin Man chairs. Mori (Mori, Kotani & Kiyohiro, 1994) has steadily been improving his "Hitomi" travel assistant. This system is not really a wheelchair (though it is built out of one) but is more an aid for people who are blind. None the less, the sensing capabilities and navigation are similar to those needed for robotic wheelchairs. Mori's system also has real-time vision which allows it to sense oncoming cars from sufficient distance to take evasive action.

The contribution of the Tin Man robots is more one of methodology than capability. Each of the Tin Man robots was conceived, designed, built and programmed in about one work month. All of the hardware in both robots (with the exception of the box where the connectors were mounted) is commercial hardware which is readily available. The wheelchair, controller, computer and sensors are all mass-produced consumer products; they are reliable and relatively inexpensive. The software, while customized for this application, uses control strategies and programming techniques well known in the literature.

The user interface for Tin Man is simple to use. The tests in section 6 show that this interface requires substantially less operator interaction, and less operator precision then in standard wheelchair interfaces. Yet, the button arrangement is not appropriate for many people. People with multiple disabilities require a variety of specialized interfaces; the specifics depend on the individual's motor and communications capabilities. But the interface that has been demonstrated has the two critical features that work well with almost all people, whatever their capabilities or limitations:

1. the interface is low-bandwidth,
2. the interface is not time critical.

The result is that the chair has been operated by people aged 6 to sixty with only minimal instruction[2]. The digital lines that connect the user control buttons could easily be connected to chin switches, a voice activated switch, head scanner, or any number of other specialized interface devices that are appropriate for a specific user. These devices are also commercially available and could be easily integrated.

The importance of this work is not in the component technologies, or that this particular implementation can help many people with disabilities (though we believe this is true). The Tin Man robots are demonstrations that robot technology, both hardware and software, has matured to the point where a specialized system can be put together quickly and inexpensively, and then operated successfully by technologically naive users. For the discipline of robotics to have an impact on the world, and for there to be continued interest in robotics research, it is necessary that this feat be repeated many times.

7 Conclusions

We have constructed two prototype robotic wheelchairs that are capable of maneuvering through a wide variety of typical environments without collision. The chairs take direction from the human user in a variety of forms ranging from direct control to destination specification. This type of chair should prove useful to persons with mobility impairment and limited visual acuity, spasticity, diminished fine motor control or any condition that makes it difficult for them to independently operate a normal power wheelchair. There has also been interest expressed in using the chair by other wheelchair users who believe that the autonomous functions demonstrated by the Tin Man chairs would be more generally useful.

Some of the more unusual aspects of this project are that the equipment and parts are all readily available and off the shelf, including much of the software. The cost for the modifications represent only a slight increase in cost over a normal power wheelchair. Tin Man is an existence proof that robotic aides for the mobility impaired do not have to be prohibitively expensive.

Acknowledgments

The authors would like to thank Anne Wright, Randy Sargent, Mark Westling, and Mike Wessler for their help in developing Tin Man. This work was supported in part by The American Association for Artificial Intelligence, Apple Computer Corporation, The Jet Propulsion Laboratory under a grant from their

Director's Discretionary Fund, The KISS Institute for Practical Robotics, The MIT AI Laboratory, The MITRE Corporation, Power Concepts Incorporated, and Vector Wheelchair Corporation.

Notes

1. Neither of the authors are, at present, mobility impaired.
2. Instruction consisted of a two-minute explanation of what the buttons did, and what each status light meant.

References

Bell, D.A., Levine, S.P., Koren, Y., Jaros, L.A., and Borenstein, J. 1993. Shared Control of the NavChair Obstacle Avoiding Wheelchair. In *Proceedings of RESNA'93.*

Connell, J. and Viola, P. 1990. Cooperative Control of a Semi-Autonomous Mobile Robot. In *Proceedings of the 1990 IEEE Conference on Automation and Robotics*, pp. 118–1121.

Elsaesser, C. and Slack, M.G. 1994. Planning with Sequences of Situated Skills. In *Proceedings of the AAIA/NASA Conference on Intelligent Robots in Field, Factory, Service and Space.*

Firby, R.J. 1989. Adaptive Execution in Complex Dynamic Worlds. *Ph.D. thesis*, Yale University Department of Computer Science.

Gat, E. 1991. Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots. Ph.D. thesis, Virginia Polytechnic Institute Department of Computer Science.

Gelin, R., Detriche, J.M., Lambert, J.P., and Malblanc, P. 1993. The Sprint of Coach. In *Proceedings of the '93 International Conference on Advanced Robotics.*

Miller, D.P. 1990. Rover navigation through behavior modification. In *Proceedings of the NASA Conference on Spacecraft Operations Automation and Robotics.*

Miller, D.P., Desai, R.S., Gat, E., Ivlev, R., and Loch, J. 1992. Reactive Navigation Through Rough Terrain: Experimental Results. In *Proceedings of the 1992 National Conference on Artificial Intelligence*, SanJose, CA, pp. 823–828.

Miller, D.P. and Grant, E. 1994. A Robot Wheelchair. In *Proceedings of the AAIA/NASA Conference on Intelligent Robots in Field, Factory, Service and Space.*

Miller, D.P. and Slack, M.G. 1994. Increasing Access With a Low-Cost Robotic Wheelchair. In *Proceedings of the Intelligent Robots and Systems Conference, IROS'94*, Munich, Germany, pp. 1663–1667.

Mori, H., Kotani, S., and Kiyohiro, N. 1994. A Robotic Travel Aid HITOMI. In *Proceedings of the Intelligent Robots and Systems Conference, IROS 94*, Munich, Germany, pp. 1716–1723.

Nisbet, P.D., Odor, J.P., and Loudon, I.R. 1988. The CALL Centre Smart Wheelchair. In *Proceedings of the First International Workshop on Advanced Robotics for Medical Health Care*, Ottawa, Canada.

Slack, M.G. 1992. Sequencing formally defined reactions for robotic activity: Integrating RAPS and GAPPS. In *Proceedings of the SPIE Conference on Sensor Fusion.*

Yu, S., Slack, M.G., and Miller, D.P. 1994. A streamlined software environment for situated skills. In *Proceedings of the AAIA/NASA Conference on Intelligent Robots in Field, Factory, Service and Space.*

David P. Miller is presently a Principal Scientist at the MITRE Corporation where he carries out research on autonomous mobile robots. He is also a Director of KISS Institute for Practical Robotics and is co-chair of the Robotics Resources and Manufacturing Department of the International Space University. His research interests are in control architectures for intelligent agents, assistive robotics and robotics and technology education. He received his Bachelor's degree in Astronomy from Wesleyan University in 1981 and his Ph.D. in Computer Science from Yale University in 1985.

Marc G. Slack is currently a Lead Scientist at the MITRE Corporation where conducts research on intelligent agents. He is a member of the board of the KISS Institute for Practical Robotics. His research interests are in spatial navigation, robot architectures and network agents. He received his BS in 1985, his MS in 1987 and his Ph.D. in 1990, all in computer science from Virginia Tech.