Mobile Robot Navigation by Wall Following Using a Rotating Ultrasonic Scanner

Martin B. Holder, Mohan M. Trivedi, and Suresh B. Marapane

Electrical and Computer Engineering Department
University of California at San Diego, La Jolla, CA 92093-0407
Email: trivedi@ece.ucsd.edu

Abstract

This paper presents a wall following algorithm for a mobile robot. The robot uses range information provided by a single rotating ultrasonic transducer. Experimental evaluations have shown that the algorithm is capable of navigating and dealing with a variety of wall structures typically encountered in an indoor environment.

1. Introduction

Autonomous navigation involves sensing, obstacle detection and avoidance and path planning. Ultrasonic based navigation is more appropriate in indoor environments [1, 4]. In the case of indoor environments autonomous navigation can be accomplished by wall-following [2, 3]. In this, the robot navigates alongside a wall at a pre-determined distance. Wall following can be thought of as being a low level reflexive behavior which when coupled with a more intelligent behavior can accomplish a useful high level task.

In this paper we describe a robust wall following algorithm that utilizes a single ultrasonic transducer mounted on a motor that provides radar type range information. The wall following algorithm is implemented on a distributed multiprocessor mobile robot SMAR-T (Small Mobile Autonomous Robotic Testbed.)

2. SMAR-T system overview

The present research is conducted as a part of our activities in cooperative robotic systems [9, 6, 8]. SMAR-T is a distributed multiprocessor mobile robot. SMAR-T is designed specifically as an experimental testbed for research in cooperative robotic systems. In one of the first successful demonstration of such a cooperative robotic team involving SMAR-T is a convoying system, where SMAR-T has succeeded in following a larger mobile robot autonomously [8].

The base of SMAR-T is of a circular footprint with an 18 inch diameter. The robot weighs approximately 35 pounds standing approximately 1 foot in height. A two wheel differential stepper motor drive system allows the robot to perform pivotal movements. The drive wheels are driven with a 2:1 gear ratio to account for a total 250 oz.in. of torque delivered to the wheel from each motor. SMAR-T's drive system is capable of attaining speeds up to 1 ft/sec. Figure 1 shows the mobile robot SMAR-T.

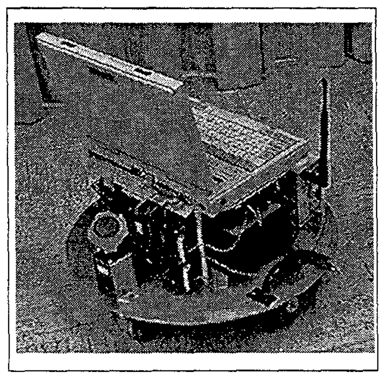

Figure 1. *SMAR-T*

The control system of SMAR-T is comprised of two layers. The high level control is considered to be derived from the host system, a 486 laptop PC, from only a programming perspective utilizing the C language. Lower level, more specific control will come from the dedicated subsystems which are programmed in Motorola assembly and include hardware dependent on the dedicated task. A means of intra-communication was developed between the high level host and lower level subsystems. The completed system is shown in Figure 2.

The PC host system interacts with the robot's various subsystems through RS232 communications. Each subsystem has a dedicated 68HC11 micro-controller unit (MCU), which provides for a parallel, distributed, control ability.

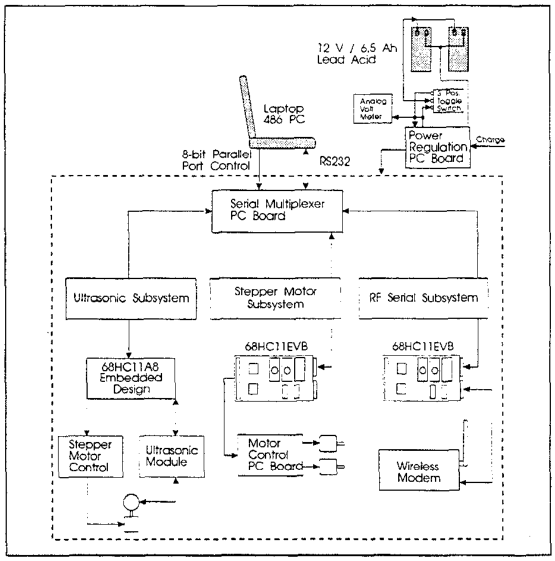

Figure 2. *SMAR-T's Electrical Control System.*

The dedicated MCU's can perform various tasks without intervention from the PC, freeing the workload of the PC. Since the laptop PC has only one serial port, and the need to communicate with three different subsystems exists, an RS232 serial multiplexer was designed. The multiplexer is controlled by the host PC's parallel port which selects the subsystem the host wishes to communicate with. An attempt was made in the hardware design and assembly programming of each subsystem to maximize its versatility.

3. Rotating ultrasonic scanner subsystem

SMAR-T uses an ultrasonic transducer for acquiring range information. The ultrasonic subsystem consists of a single ultrasonic transducer mounted on a stepper motor. This subsystem is capable of generating radar-type range information of its surroundings. The granularity and the angular range of operation are all controllable via a host processor.

The ultrasonic transducer subsystem is an embedded system based on a single 68HC11A1 MCU. High level control of the subsystem is achieved via an RS232 serial interface. Figure 3 shows a diagram of the ultrasonic subsystem. The ultrasonic subsystem is shown in Figure 4. The various hardware/firmware issues associated with the embedded system and high level control of it are described in this section. More discussion of the system can be found in reference to [5].

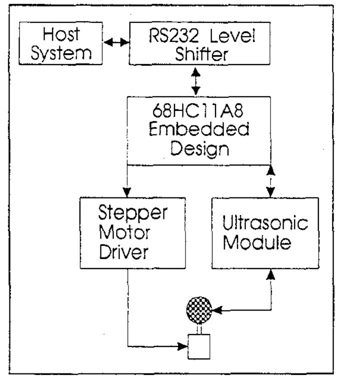

Figure 3. *Ultrasonic subsystem.*

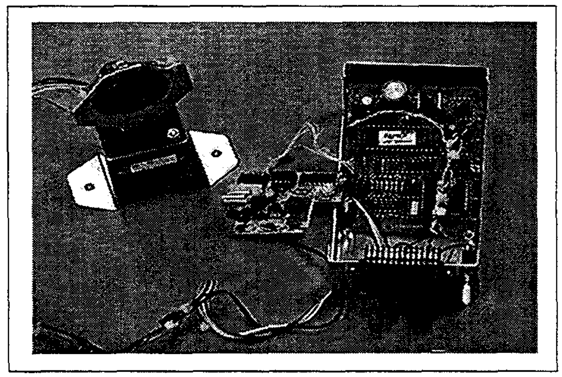

Figure 4. *Ultrasonic transducer subsystem.*

3.1. Ultrasonic module for ranging

This module contains the electronics to transmit (fire) and receive (listen) ultrasonic pulses. The module consists of two TI chips (TL851 and TL852), step-up transformer, and control circuitry. The TL851 and TL852 work in tandem with one another. One is used for the transmission of the sound pulses, the other for receiving. Typically, a pulse train of sixteen pulses at a frequency of 49.4 kilohertz is issued to excite the transducer thus providing the bursts of sound. The actual transmission's amplitude for each sound pulse reaches three hundred volts. After a given time delay to allow for the transducer to settle from initial ringing, the receiver portion begins to listen for incoming sound through a bandpass filter centered at 49.4 kHz with 5 kHz lobes. The receiver has the capability to alter its gain or sensitivity to enable it to hear the pulses which travel further distances and are attenuated as a result. When a given pulse is detected (threshold tripped) the time differential is noticed and distance of flight can be calculated.

3.2. Motor control for panning ultrasonic transducer

A small 12.5 oz.in. stepper motor is used to rotate the ultrasonic transducer, and the MC3479 chip is used to drive the motor directly. The MC3479 is capable of directly driving stepper motors with a maximum current rating of 350mA per coil [7]. Internal logic keeps track of the current pulse sequence and for the given inputs, provides internal logic to generate the next correct pulse sequence.

All stepper motors operate via sequencing which details the manner in which the coils are given particular polarities for proper motor advancement. The MC3479 handles this process internally. Sequence parameters differ depending on the operating mode (full/half step) of the stepper motor. The particular stepper motor used is capable of rotating 1.8°/pulse in full step operation thus 0.9°/pulse in half step operation. Due to the light load of a single ultrasonic transducer, no problems associated with moving such a load by the stepper motor are noticed.

3.3. Embedded system: hardware and firmware

The focus of the ultrasonic subsystem is based on the 68HC11A1 MCU, which is manufactured by Motorola, and capable of sixteen-bit addressing with an eight-bit data bus. The bus speed of the MCU is 2 MHz, which is derived from a divide by four 8 MHz clock. The chip is very versatile, thus enabling the programmer a multitude of features such as interrupts, serial communication, and I/O ports. Primary interests for this MCU are based on its timing capabilities along with its ability to handle serial transmissions. The timing capabilities are useful for providing crucial interrupt driven timed responses to enable accurate timing/control for the ultrasonic module. The ability to handle serial transmissions is useful in driven subsystem with simple commands to select the control actions of interest.

The MCU is programmed to utilize a total of three timer interrupts for firing and listening of the transducer on the range module. Two timer output compare (TOCX) interrupts and one timer input capture (TIC1) interrupt will be utilized to time the sound waves. Once a command has been issued to fire the transducer and the free running counter reaches zero, a TOC2 interrupt will be generated. This interrupt will fire the transducer and enable an additional TOC3 interrupt to occur 1msec later. The 1 msec delay allows the transducer to settle from ringing before listening for returned sound waves. Once the 1 msec time period has expired, the TOC3 interrupt occurs and sets the BINH bit high on the module indicating the module should now listen for returned sound waves. The interrupt service routine also sets the TIC1 interrupt to occur when a low to high transition is seen on its dedicated pin. This pin is directly connected to the ECHO output from the module which indicates the sound wave has been received by setting its corresponding pin high. The MCU will then notice such a change and will note the ending time period. For echoes not received by the module, a time-out condition is noticed and accordingly.

Ultrasonic serial protocol/command set

All serial transmissions received by the MCU are polled in software, and the MCU utilizes its **SCI** (Serial Communications Interface) for such transmissions. Interrupts are not necessary for the MCU's serial system. The robotic system is setup in a master-slave format. The MCU will react based only on the commands sent by the host computer. At no time will the subsystem send bytes to the host system without the host system's knowledge of such incoming data. The protocol begins with the host computer system sending a start of transmission byte, ASCII 'E'. The next byte sent is the command byte. A total of seven command bytes are used, and since the command set is short, commands are represented by ASCII '1' through '7'. The last value sent by the host system is the checksum value which is a single, unsigned eight-bit value. The checksum value is the unsigned ASCII addition of 'E' and the command byte. If properly sent, the command will be serviced by the MCU, which will return data if appropriate along with a checksum. The **SYNC** command is a special case and exception to this rule. For commands with no data to be sent back, the MCU will send the received checksum, if valid.

3.4. High level control

The high level control of the subsystem is achieved via a host processor which connects to the subsystem through an RS232 serial interface. The high level commands include those for selecting the sampling angles, total range of pan, and firing of the transducer. The command set is tabulated in Table 1.

The function calls handle all serial communications for the host system to properly communicate with the subsystem's MCU. All routines likewise make the proper serial connections of the host system to the subsystem via a hardware serial multiplexer.

4. Wall Following Algorithm

The ability to follow walls sounds simplistic, but is by no means a trivial task for a mobile robot. A multitude of decisions must be made based on current ultrasonic sensor readings. The decisions made will dictate how both motors will be driven to prevent the robot from possible collisions, yet maintain some desired distance from the wall. The sensor readings can be faulty or misleading and can lead to wrong

Table 1. *Ultrasonic Subsystem High Level Command Set.*

function	Meaning
son_cwdir	CW Direction
son_ccwdir	CCW direction
son_fullst	Set pan motor for full step operation (1.8deg/step)
son_halfst	Set pan motor for half step operation (0.9deg/step)
son_pulse	Send pulse to rotate pan motor in it's current configuration
fire	Fire transducer and gather timing information
son_sync	Synchronize on-board RS232 communications with host PC
son_reset	Reset subsystem and acquire transducer angle information
sonar_angle	Move transducer to a specific angle of interest
son_scan	Scan particular region of interest at specified resolution
clr_scan_values	Clear sonar array containing range information at specified angles

decisions and ultimately poor wall-following ability. The sensor's angular position relative to the given object and reflections of the sound waves from the object play an important role in the inaccuracy of such readings. A means of disregarding such readings improves the robustness of the robot in handling such a task.

The wall-following algorithm utilizes two regions of sensory data to determine the closeness of surrounding objects. Figure 5 illustrates the two regions. The **R1** region provides the ability for the robot to look ahead to avoid upcoming collisions on the robot's right side. **R1** (region one) includes a total of 7 sonar readings separated by 9 degree increments to cover a total angular area of 54 degrees, and the reading which gives the lowest value is used in the decision process. **R2** represents the region in which the robot searches for non-avoidable obstacles. **R2** (region two) includes a total of 3 sonar readings separated by 5.4 degree increments to cover a total angular area of 10.8 degrees, The lowest reading is the value used in the decision process. **R1** and **R2** regions make the decision process robust by making the decision based upon a range of data rather than one single measurement.

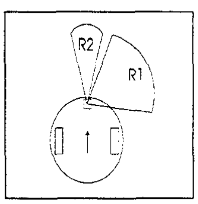

Figure 5. *Sensory regions used for wall-following.*

The algorithm for wall following is passed two important parameters, a distance and the maximum motor speed. The distance parameter (**DIST**), details the closeness by which the robot will attempt to hug the wall. The maximum motor speed parameter restricts the robot's motors from reaching speeds beyond this value. Both parameters are important and affect the ability of the robot to perform its wall following task.

The lower the distance parameter, the more probable it is that the robot will enter various openings along a wall such as open doors. This is due to the 30 degree cone associated with the ultrasonic transducer and the **R1** region over which the transducer is scanned. For a larger distance or further wall following approach, a more broad range of sensory data is given, due to the cone. This range could be larger than the opening in question and thus prevent the robot from entering.

Figure 6 illustrates the flowchart of the algorithm. For cases in which the robot is following a wall and the wall's reference suddenly disappears, indicating a possible corner, the robot will begin a hard turn towards the right. Figure 7 shows such a scenario and the movement profiles the robot will undergo. In such a condition, the speed deviation is extremely high; and the right wheel of the robot will approach zero while the left wheel will approach maximum speed. This will continue until successive sonar scans reveal the corner of the wall, at which time the motor speeds will once again be adjusted, which continues until the robot once again converges to the distance parameter between itself and the wall.

5. Experimental Results

Extensive set of experiments have been conducted to evaluate the performance of the algorithm. These evaluations have shown that the algorithm is capable of navigating and dealing with a variety of wall structures typically encountered in an indoor environment. One such experiment is presented in this paper.

In this experiment a trash can and box are placed alongside the wall to demonstrate the wall follower. Three experiments were made at wall following distances of 4, 12, and 18 inches. The results of the the various test runs are shown in Figure 8. In the case of 4 and 12 inch distances

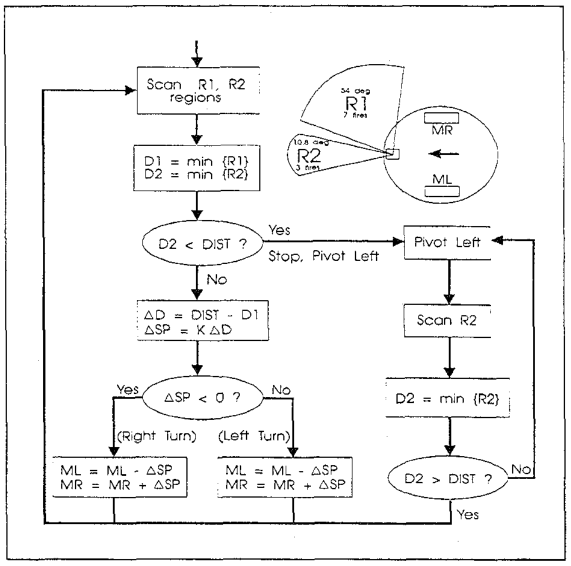

Figure 6. *SMAR-T's Wall Following Algorithm.*

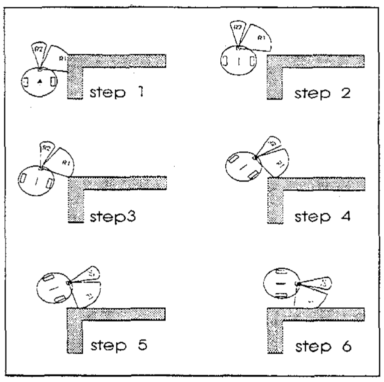

Figure 7. *Movement profiles at a corner.*

the robot was able to maneuver successfully around the contours of the obstacles and enter the open doorway. In the 18 inch distance, SMAR-T likewise was able to avoid the obstacles yet did not enter the open doorway. The open door was detected by the robot's R2 region thus triggering the exception case (non-avoidable obstacle) of the algorithm. This occurred due to the open door being closer to the robot than the specified wall following distance. In this situation, the robot stopped and began a pivotal left turn until clear from the door at which time the robot once again resumed its normal wall following.

SMAR-T's wall following algorithm has proven to be very robust. Numerous situations have been tested with extremely robust results. In each of the three test runs, SMAR-T was able to maintain the given distances and maneuver around the contours of the obstacles.

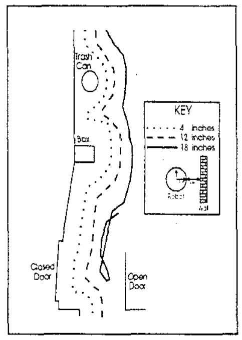

Figure 8. *Three experimental trials*

6. Concluding Remarks

In this paper we described a robust wall follower. The wall follower utilizes range information provided by a rotating ultrasonic transducer. The algorithm was implemented on a distributed multiprocessor mobile robot SMAR-T. Extensive set of experiments have been conducted to evaluate the performance of the algorithm. These evaluations have shown that the algorithm is capable of navigating and dealing with a variety of wall structures typically encountered in an indoor environment.

References

[1] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. SMC*, 19(5):1179–1187, September 1989.

[2] R. A. Brooks. Elephants don't play chess. In *Proceedings of the 1990 IROS*, North-Holland, June 1990.

[3] G. Giralt. *NATO ASI Series Robotics and Artificial Intelligence*, chapter Mobile Robots. Springer-Verlag, New York, NY, 1984.

[4] C. Gourley and M. Trivedi. Sensor based obstacle avoidance and mapping for fast mobile robots. In *Proceedings of IEEE ICRA*, pages 1306–1311, San Diego, CA, May 1994.

[5] M. B. Holder. Design and implementation of an integrated multi-processor mobile robot for experimental research in cooperative robotics. Master's thesis, Univ. of Tennessee, 1994.

[6] S. Marapane, M. Holder, and M. Trivedi. Motion control of cooperative robotic teams through visual observation and fuzzy logic control. In *ICRA*, Minneapolis, Minnesota, Apr. 1996.

[7] Motorola Semiconductor Corporation. *Motorola Linear Interface Devices*, 1988.

[8] K. C. Ng and M. M. Trivedi. Multirobot convoying using neuro fuzzy control. In *ICPR*, Vienna, Austria, Aug. 1996.

[9] M. Trivedi. Intelligent robots: control and cooperation. In *SPIE*, pages 139–142, Orlando, Florida, Apr. 1995.