Simon Fraser University
Burnaby, BC
V5A 1S6
trac-tech@sfu.ca

March 9, 1999

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 370 Project Gerbil Design Specification

Dr. Andrew Rawicz,

The attached document, *Project Gerbil Design Specification*, outlines the proposed design for our project. The Gerbil's core purpose is to autonomously navigate and map a room, utilizing a positioning and sensory system. This project would allow for various applications. One such application would be the cleaning of hospital floors, which is the purpose we will explore.

Contained in this document is the proposed system design, as well as the specifications of the various system components. These components include the positioning, motor control, and sensory systems, and the system processing.

TRAC Technologies comprises of four intelligent, hard-working, and innovative third-year-engineering students -- Celina Tinio, Robbie Grue, Andrew McPherson, and Travis Hammond. If you have any questions or concerns regarding our functional specification, please contact Celina Tinio at 872-3750, or by e-mail at mtinio@sfu.ca.

Sincerely,

Celina Tinio
TRAC Technology

Enclosure: *Project Gerbil Design Specification*

Project Gerbil Design Specification

Submitted by: **TRAC Technology**
Celina Tinio, Travis Hammond, Robbie Grue, Andrew Mcpherson
trac-tech@sfu.ca

Contact: **Celina Tinio**
School of Engineering Science
Simon Fraser University
mtinio@sfu.ca

Submitted to: **Andrew Rawicz**
School of Engineering Science
Simon Fraser University

Steve Whitmore
School of Engineering Science
Simon Fraser University

Date: March 9, 1999

Executive Summary

The ability of machines to interact with their environments in a similar fashion to humans is of immeasurable value in today's ever-changing society. Our project, the Gerbil, is an attempt to design a small robot that will have the ability to navigate around an area, detect any objects it encounters, and maintain a current map of these objects. Among many other things, such a robot will allow areas such as hospitals to be cleaned of biohazards and other wastes, which could pose health risks to human counterparts.

The Gerbil will navigate using a positioning and object sensory system, whose output signals will be manipulated by a microprocessor. The processor will then be responsible for directing the robot's movements and creating and updating the map of the area as needed.

This document describes the proposed functionality for the Gerbil project, along with some of the design choices being presented to us in regards to the positioning and sensory systems.

Table of Contents

List of Figures

List of Tables

1. Introduction

One of the inexplicable obsessions in the history of robotics is the desire to replicate the functionality of the human. Humans have many aspects which designers have found very difficult to duplicate in the realm of machines. Certain aspects, such as object recognition and navigation, have an incredible potential when combined with those advantages, which a machine has over a human.

One example of this is in the area of biohazardous waste disposal. Such substances pose a substantial threat to those people who must clean them, while they have little or no effect on a machine. Therefore, a machine would be better suited to the handling of such tasks, and would be ideal if the recognition and navigation aspects of a human could be combined with its resistance to disease. This is what the Gerbil project attempts to achieve.

The purpose of this document is to describe the required design for the Gerbil. The intended audience for this document is Dr. Andrew Rawicz, Mr. Steve Whitmore, TRAC Technology product designers, and external design consultants.

2. Sensory System

The sensory system will determine if the robot is in physical contact with an object. It must also allow the robot to follow the perimeter of the wall or object while remaining in contact with it. To solve this problem, we will use a horizontal wheel as a bumper that surrounds the robot. Figure 1 illustrates the design.

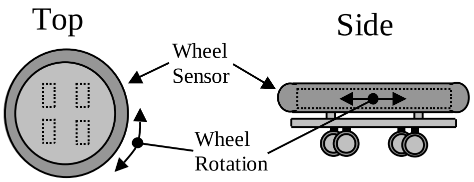

Figure 1 - Wheel Bumper Sensor

The wheel will be able to turn so that the robot can follow an object by constantly remaining in contact with it. The sensory system will also be able to monitor from which direction a force is being applied to the wheel. This will be done using four spring-loaded sensors that will be mounted on the front and back and on each side. They will come into contact with the wheel through bearings such that the wheel is allowed to turn. The output of each sensor will then be fed directly to the system processor. Figure 2 shows the location of the sensors on the wheel bumper.

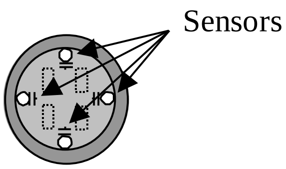

Figure 2 - Object Sensor Location

The sensory system will also generate an interrupt to the processor when the robot touches an object. The interrupt will be created by sending the OR of all the switch outputs to an interrupt pin on the processor. Figure 3 illustrates the circuit design of sensory system.

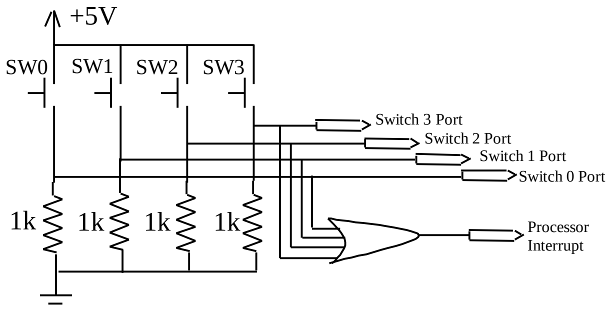

Figure 3 – Circuit Design for Object Sensors

3. System Processor

3.1 Summary of System Responses

The system will need to be controllable (through the use of microprocessors) in order to perform any useful task. Therefore, the system must be able to respond appropriately for both controlled and environment inputs and conditions. Table 1 shows the applicable inputs / output requirements along with the suitable system behaviour.

Requirement - *Internal*	**System Behaviour**
Robot Movement	The robot's movement must be controlled by the System Processor.
Positioning	To determine the robot's current location, position signals must be deciphered and stored.
Pathing	The robot must be capable of arriving at a target position.
Object Sensory	Once an object is sensed, such as a wall or table, the robot must take appropriate action.

Table 1 - Input and Output System Behaviour

3.2 PIC Microprocessor Overview

The required system responses shown in Table 1 will be controlled by two PIC processors (16F84 and 16C774). These processors were chosen because of their simplicity (RISC), excellent supporting documentation and example modules, and the high availability of parts and simulators. The I/O pins and physical layout of a 16F84 and 16C774 PIC are illustrated below in Figure 4 and 5.

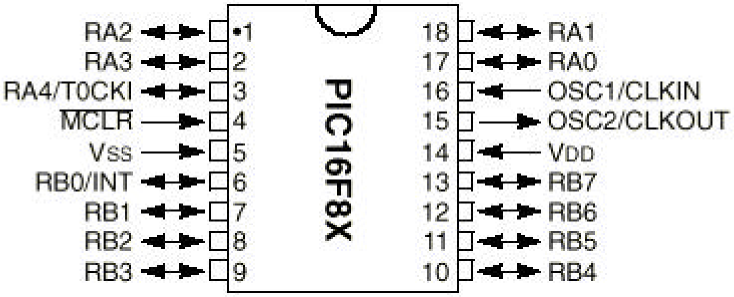

Figure 4 - PIC 16F84 Microcontroller

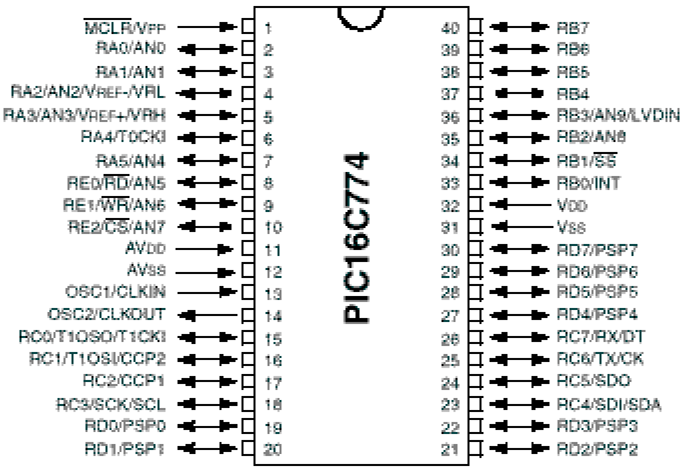

Figure 5 - PIC 16C774 Microcontroller

The PIC 16C774 will be used as the main processing centre of the robot. The PIC 16F84 will be used as the controller of the positioning system. The main processor will be interfaced directly to the sensory system and the motor control system. The main processor will also use a parallel interface to communicate with the positioning controller. The main processor will interface serially to an RS-232 IC responsible for map transfer and will also interface serially with EEPROM.

3.3 Positioning Algorithms

3.3.1 Positioning Introduction

Because the core functionality for our Gerbil project is to be able to clean a floor , a proper positioning algorithm must be established. This algorithm will dictate the robot around the room in such a manner that the floor is covered completely, yet efficiently. To accomplish this task, the robot needs to know it's current position, as well as the next target position. The positioning algorithm will now be discussed.

3.3.2 Position Distances

As stated in the positioning introduction, the robot requires to know it's current location, along with the next desired location (target). Once the target is reached, a new target will be assessed and the process begins again. This procedure will direct the robot throughout the room, and is discussed further in Section 1.4, "Pathing Algorithms".

Determining the robot's position can be based upon many different measurements (wheel rotations, etc). We chose to use a counter algorithm which will count the delay time associated with a sound wave signal being sent from external beacons. This method is adequate in that we do not require the physical location of the robot, and thus a symbolic position will do.

This counter system is demonstrated below in Figure 6.

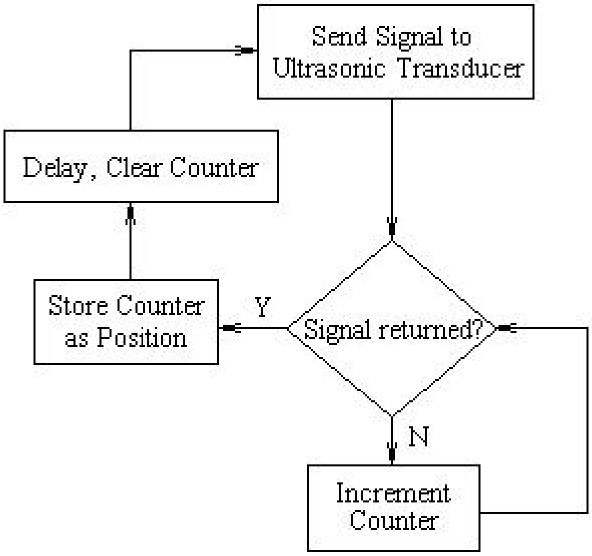

Figure 6 - Position Determining Flow Diagram

We will use ultrasonic transducers (discussed in Section 5, "XXX") to create the delay signal being returned to the robot. This is because the clock frequency inside the PIC microprocessors is much faster than that of the returning sound signal. Therefore, the further the robot is from the reference transducer beacons, the larger the counter (i.e. position) value. Thus, this counter will be used to represent the position "distance".

We must, however, specify the maximum size counter to be used. We felt that a 16-bit counter (00 - FF HEX) will be sufficiently large for our purposes. This means, however, that we must ensure that the counter does not run too high and overflow back to "00". Therefore, a delay must be placed on the counter such that "FF" (maximum allowable counter value) at least one room width away. Our project will use a maximum room size of 5m x 5m. Figure 7 illustrates the counter distance system from the two ultrasonic beacons.

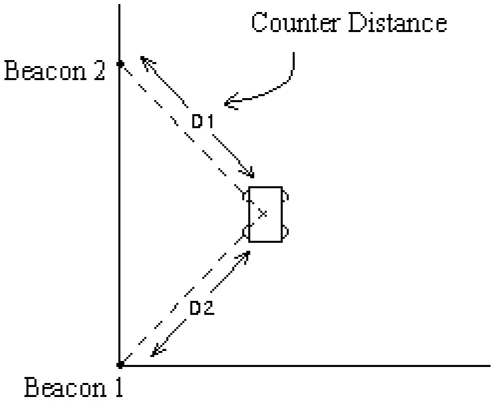

Figure 7 - Beacon-Robot Counter Distance

3.3.3 Counter Distance into X-Y Position

Positioning, however, is based on a coordinate system. Therefore, the counter distances from the previous section must be converted into X-Y coordinates to be of any use. Figure 8 below shows the beacon distances with respect to the X-Y coordinates.

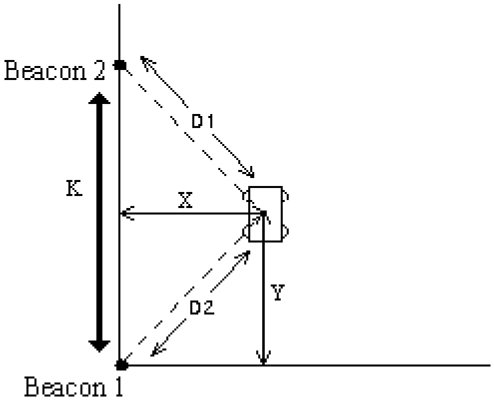

Figure 8 - Beacon Distance to X-Y Coordinates

From Figure 8 above, we obtain the following X-Y formulas:

$$y = \frac{d_2^{\,2} + k^2 - d_1^{\,2}}{2k}$$

$$x = \sqrt{d_2^{\,2} - y^2}$$

Because square root cannot be easily calculated on a RISC microprocessor, an approximation for the X position must be made. By observing Figure 9, we can easily find the nearest whole number for X. As well, the error induced from this approximation will be negligible, as our position counter will be ≥ 20.

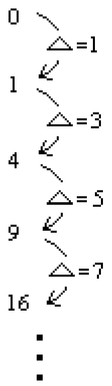

Figure 9 - Approximating Square Root

3.4 Pathing Techniques

3.4.1 Pathing Introduction

The pathing algorithm will be used to dictate the robot throughout the room. Ideally, each available coordinate (not blocked by an object) will be covered once and once only. Because there are many ways of implementing a pathing algorithm, an appropriate method must be designed with both algorithm simplicity and ability in mind.

The main idea of a pathing algorithm is to determine what the next robot location should be based on the current location, as well as the previously visited coordinates. To determine the target position, wall and object locations (i.e. unreachable coordinates), as well previously visited coordinates, must be processed. The task of proceeding to a target destination will be performed using a feedback loop, as shown in Figure 10.

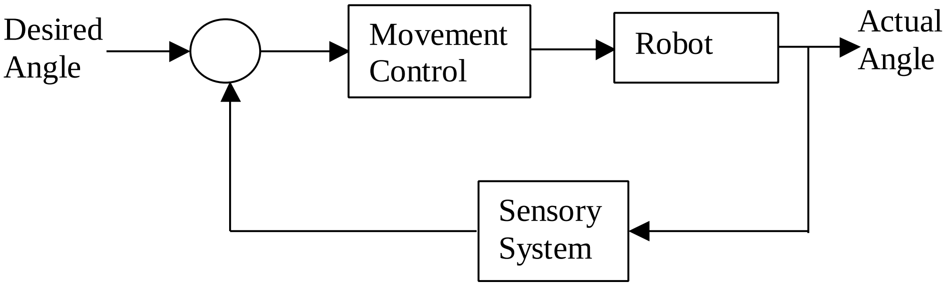

Figure 10 - Target Destination Feedback Loop

3.4.2 Coordinate Memory Storage

In order for the coordinate information to be used, external memory must be interfaced into our PIC processor. Therefore, the manner in which the coordinates are stored into this memory must be determined.

One such method is to store each visited coordinate sequentially into memory. Figure 11 gives an example of this approach.

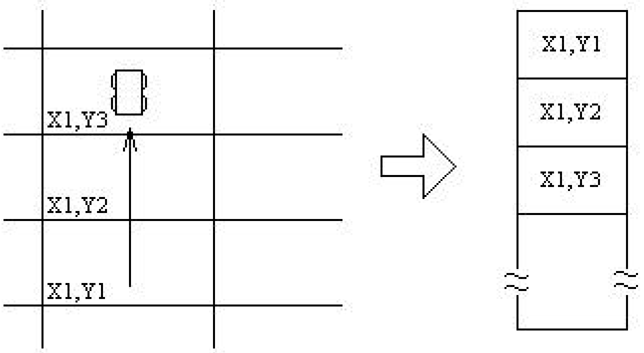

Figure 11 - Encountered Coordinates into Memory

This method, however, does not give any information about previous locations (without looking through the entire memory stack). Only the previous position is readily available.

To obtain a better storage method, we can first see that not every position requires storage. We could reduce the resolution of our distance/coordinate ratio by not storing every counter value. Alternatively, we can see that only the positions in which a wall was encountered needs to be stored for a map to be recreated. This drastically reduces the number of positions requiring storage. This wall-contact positioning storage system is shown below in Figure 12.

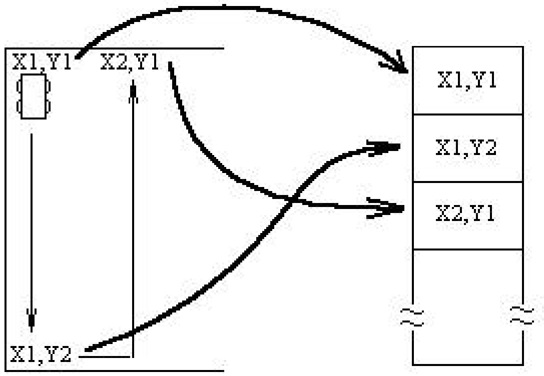

Figure 12 - Encountered Walls into Memory

3.4.3 Pathing Techniques

The robot can attempt to cover the room in a variety of ways. One such method is to use a “zamboni” approach. This involves circling the outer wall of the room and slowly decreasing it’s radius until it reaches the center. An example of the zamboni method is shown below in Figure 13.

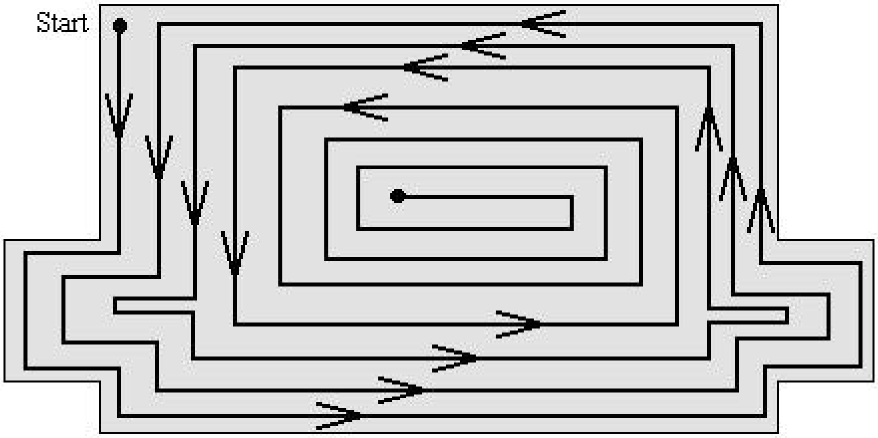

Figure 13 - Zamboni Method

The idea behind the zamboni method is to generate an initial path around the room, generate a smaller path, and proceed on the new smaller path. The new current path will then become the “initial” path as before, and the process will begin again. This process will continue until the center of the room is reached.

Another approach is to employ a “lawn-mower” method. This technique involves the robot going straight until an object is hit, then incrementing the robot’s position and proceeding back. An example of this approach is now shown in Figure 14.

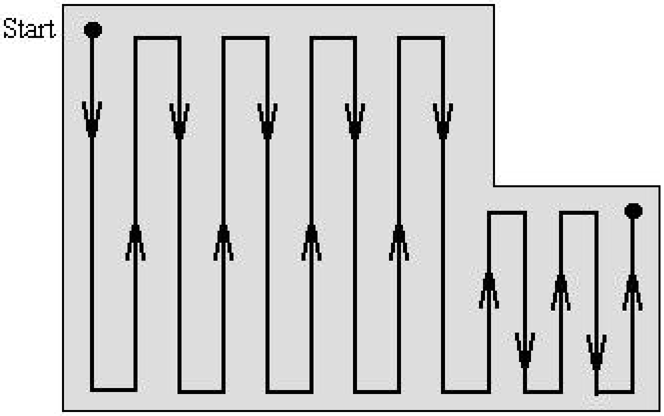

Figure 14 - Lawn-Mower Method

3.5 Map Transfer to PC

The Robot will have the ability to transfer information about its environment to a PC. This will be accomplished through a serial data link to the PC. The System Processor will interface to an RS-232 translating IC which will handshake to a PC via a RS-232 cable

3.6 Object Sensory Control

The sensors provide the direction of a sensed obstruction. Once an obstruction has been detected, an interrupt will be sent to the processor, as depicted in Figure 15.

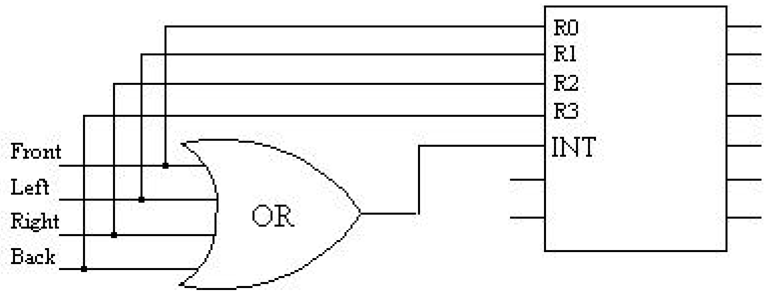

Figure 15 - Wall Sensory Inputs and Interrupt

At this point the robot will determine whether the object has already been encountered by consulting the onboard map. If the object has already been encountered, which is usually a case of overshoot from the previously desired destination, the robot will simply continue with the same operation. However, if the robot encounters a new object, or encounters the wall for the first time, a different course of events will take place. The robot will attempt to follow the perimeter of the wall or object. It will use a simple algorithm to do so. The robot will continue to move and attempt to remain in contact with the object as it does so. If the sensors indicate that the robot is no longer in contact with the object, the robot will react by turning towards the object. If, instead, the sensors indicate that the object is now more in front of the robot (i.e. a corner has been encountered), the robot will turn away from the obstruction.

4. Motor Control System

One of the most essential features, which the Gerbil will need, is the ability to move. Without that, all of the other features are pointless. As shown below in Figure 16, there are two components to the Motor Control System, namely, the motors and the drivers.

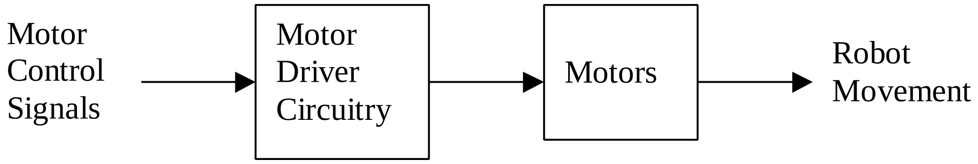

Figure 16 - Motor Control Block Diagram

4.1 DC vs. Stepping Motors

There are two possible choices for the motors. The first choice is using DC motors. These motors operate by applying a current to coil of wire that is placed between two permanent magnets. This current creates a magnetic field, which is opposite to the permanent magnets, causing the coil to be rotated. DC motors have the advantage of being able to supply a greater torque by increasing the applied current, which is one of the more important qualities a motor must have. The disadvantage of the DC motor lies in its control. Because of the nature of these motors, it is difficult to precisely control them without some sort of closed loop configuration to provide feedback on its position and/or speed. Some sort of sensor is required, such as a tachometer, whose output then needs to be integrated into the control system. This ends up being overly complicated and expensive.

The second choice is to use stepping motors, which is what we will be doing. A stepping motor is constructed with the permanent magnet as the part that rotates, while the electromagnets remain stationary. The permanent magnet is caused to rotate by applying current to different electromagnets at different times, in such a way as to cause the permanent magnet to always need to move a short distance to reach an opposing magnetic pole (this is where the “step” comes in). Stepping motors have basically the opposite advantages and disadvantages present in the DC motors. They are much easier to control, and this control can be precisely accomplished in an open loop configuration. However, stepping motors are unable to supply the high torque available with DC motors, but we feel that the torque supplied by these motors will be sufficient for our purposes.

4.2 Unipolar vs. Bipolar

There are two main types of stepping motors, bipolar and unipolar. Both motors operate on the same basic principles outlined above. The main difference, however, lies in how the windings in the motors are wired. This wiring has an effect on two important characteristics, control and torque.

In a unipolar motor, each of the two windings will have a center tap, as shown below in Figure 17. For both windings, the numbered center taps are attached to the supply voltage, and the lettered taps are attached to ground through a switch. The rotation of the rotor would be accomplished by switching on winding 1a, then 2a, then 1b, then 2b, then 1a again, and so on.

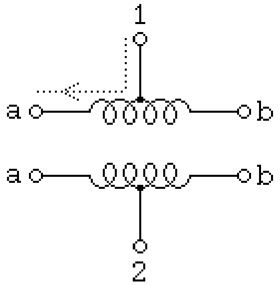

Figure 17 - Unipolar Motor Windings

This type of winding results in the motor being very easy to control, because all that is required is a switch for each of the four letter taps and some sort of current limiting. The disadvantage of a unipolar motor is a result of the thinner wires in the windings, since each winding is effectively actually two opposite windings. These thinner wires result in a high resistance in the coils. This resistance effectively limits the maximum current, which may be applied to the motor, because of the power loss in the stator windings. This power loss will directly affect the maximum temperature rise of the motor, which cannot exceed a certain limit. This limiting of the current reduces the overall torque available in the motor.

The windings in a bipolar motor are wired more simply than in the unipolar case. As can be seen below in Figure 18, these windings lack the center tap. Applying a current to the first winding, with 1a at positive supply and 1b at ground initiates the movement of the motor. The current is then applied to the second winding, with 2a at positive supply and 2b at ground. The current is then applied to the first winding again, but this time 1b is put to positive supply, and 1a is put to ground. Finally, the current is applied to the second winding, with 2b at positive supply and 2a at ground.

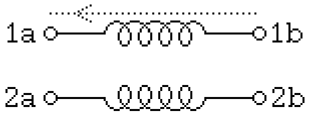

Figure 18 - Bipolar Motor Windings

The omission of the center tap results in two things. First, the control circuit for a bipolar motor is more complex, because the direction of the current through the windings must be able to be reversed. Therefore, some sort of H-bridge circuitry must be used, in addition to the control circuitry needed for the unipolar motor. However, unlike the unipolar motor windings, there is only one actual coil per windings, which allows the wire used to have a greater cross section. This increased wire size results in less resistance, which effectively means that a greater torque can be achieved from a bipolar motor that is otherwise identical to a unipolar counterpart, because a greater amount of current can be applied to the windings.

The trade-off is essentially then between control simplicity and available torque. As an H-bridge is not an extremely difficult thing to design, and taking into account that many stepping motor driver ICs come with these H-bridges built in, it would seem that the bipolar motors would be a better choice. At this time, we are using unipolar motors, simply because of availability. We are in the process of obtaining some bipolar motors at this time, and hope that we can get them in time to incorporate them into our design.

4.3 Stepping Motor Driver

Despite the gain in ease of control over DC motors, controlling stepping motors still is somewhat tedious due to the need to activate the different switches at very specific times, the actual construction of the electronic switches, and the current limiting circuitry needed. Fortunately, nearly all types of stepping motors now have IC drivers available, which include all of the things mentioned above, and in the case of bipolar drivers, they will typically contain H-bridges as well. For this section, we will focus on the control chip being used on the unipolar motors we now have, which are essentially the same as those for the bipolar case, only without the H-bridge.

Figure 19 below shows a typical driver IC and how it would be connected to the stepping motor.

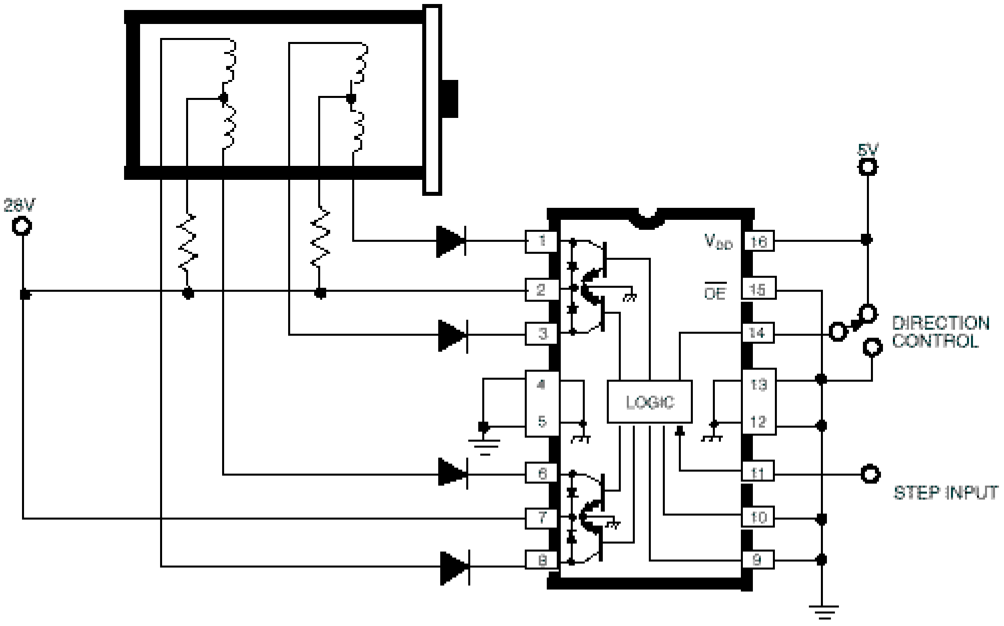

Figure 19 - Unipolar Stepping Motor Driver IC

Most of these pins are self-explanatory, and so only a few of the functions will be explained here. Pin 15 on the chip is the output enable, which allows the motor to be disabled if this is set at a logic high. The direction control on Pin 14 simply determines the rotation sequence of the outputs. The only other inputs of interest are Pins 9, 10, and 11. Pin 11, the step input, turns the motors on, and for each high-to-low transition of the square wave input, advances the motor one “step”. Thus, the frequency of this input determines the speed of the motor, which certain minimum and maximum limits.

Pins 9 and 10 are of more interest in respect to the design of the control system. These two pins allow for the choice between three types of sequences: one phase, two phase, and half-step drive sequences. Figure 20 below shows a one phase drive sequence. Such a sequence only powers one of the windings at any given time, and so it uses less power than any of the other drive sequences.

```
Winding 1a 1000100010001000100010001
Winding 1b 0010001000100010001000100
Winding 2a 0100010001000100010001000
Winding 2b 0001000100010001000100010
                          time --->
```

Figure 20 - One Phase Drive Sequence

The next type of sequence, shown below in Figure 21, is the two phase drive sequence. In this sequence, the next step winding is powered while the previous one is still on, so that at any given time, two windings are powered. Note that the two halves of each winding are never energized at the same time. This sequence obviously uses about twice the power used by the one phase drive sequence, but results in a greater torque (approximately 1.4 times that of the one phase drive sequence).

```
Winding 1a 1100110011001100110011001
Winding 1b 0011001100110011001100110
Winding 2a 0110011001100110011001100
Winding 2b 1001100110011001100110011
                     time --->
```

Figure 21 - Two Phase Drive Sequence

The final drive sequence available with this type of driver is the half-step sequence. Figure 22 below illustrates this drive sequence. Half-stepping is basically a combination of the one phase and two phase drive sequences. Half-stepping is used to increase the position resolution of the steps, i.e. reduce the step angle by a factor of 2 and increase the number of steps per revolution by a factor of 2. This results in greater precision. Half-stepping has two main disadvantages. First, it uses twice as many clock pulses to complete a revolution as the other two drive sequences, which means the clock frequency must be twice as great. Second, the half-step drive sequence only produces about half the torque of a two phase drive sequence.

```
Winding 1a 11000001110000011100000111
Winding 1b 00011100000111000001110000
Winding 2a 01110000011100000111000001
Winding 2b 00000111000001110000011100
                     time --->
```

Figure 22 - Half-Step Drive Sequence

We have no real need for such an accurate step angle, since most typically available stepping motors have step angles of approximately 1.8 degrees or 200 steps per revolution. Therefore, the half-step function is unnecessary. In addition, it is more probable that we will be needing a greater torque, since that is the area in which stepping motors are inferior to DC motors, more than we will be short of power. We will thus be using the two phase drive sequence, which is configured as shown in Figure 19.

5. Positioning System

The robot's positioning system consists of an active beacon system using two pre-placed beacons and a receiver onboard the robot. The system will use ultrasonic pulses and the Time-of-Flight (TOF) method to determine the location of the robot.

5.1 Ultrasonic Waves

Ultrasonic waves are sound waves that are outside the audible frequency range of a human. Since the device's possible uses include cleaning hospital rooms, the use of radio waves could cause interference with hospital equipment. As well, certain hospitals prohibit the operation of devices which may cause interference inside the building. Thus, we have chosen to use ultrasonic waves as a safe and unobtrusive signal.

5.2 Overview of Robot-Beacon Interaction

There will be a specific interaction between the robot and the beacons, defined as follows.

1. Upon request by the System Processor, the robot's position system will begin by sending an Electro-Magnetic signal of specified wavelength to one beacon. The system will also start a timer.
2. The beacon will receive the EM Signal and will immediately send an ultrasonic signal.
3. The robot's position system will receive the ultrasonic signal and will record the time of arrival.
4. The process will be repeated for all beacons.
5. The position system will use this information to send the proper command to the system processor.

5.3 Beacon Side of the Positioning System

A block diagram of the beacon side of the positioning system is shown in Figure 23 below.

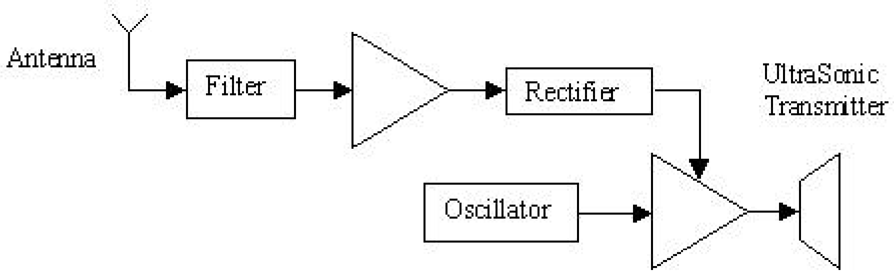

Figure 23: Positioning System – Beacon Side

The antenna receives the EM signal emitted by the onboard transducer. The received signal then passes through a filter to remove any noise and to ensure that it is in fact a valid signal. The signal is then transformed into a DC signal via a rectifier circuit. The DC signal acts as a control signal to a buffer, allowing the oscillator to trigger an ultrasonic pulse.

5.4 Robot Side of the Positioning System

Figure 24 shows the portion of the positioning system that is onboard the robot.

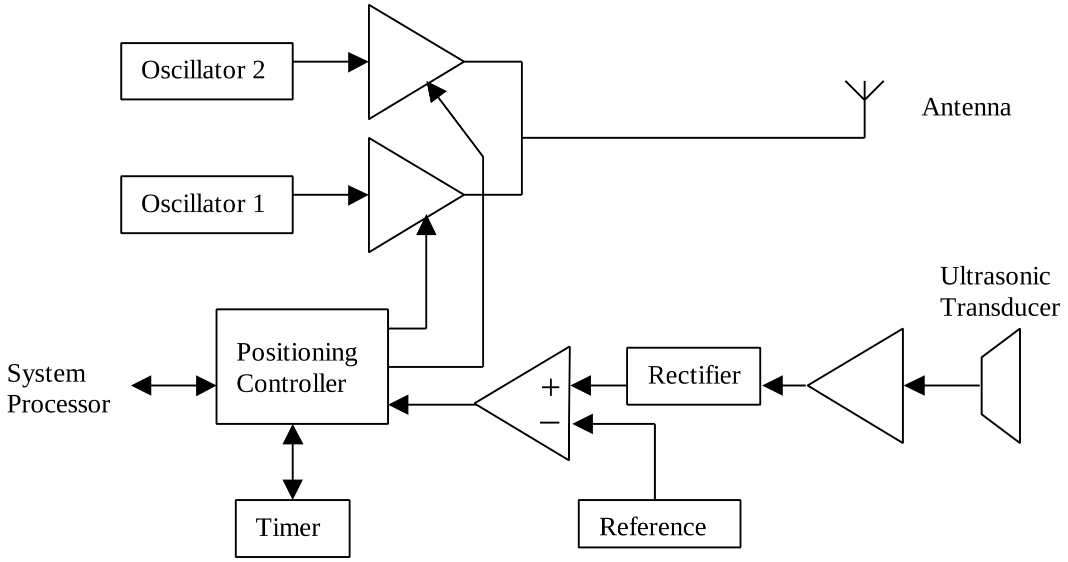

Figure 24: Positioning System -- Robot Side

The Robot Side initiates by sending an Electro-Magnetic signal to a specific beacon. This is done by the Positioning Controller, which drives the appropriate pin high, enabling the correct buffer.

The ultrasonic pulses emitted by the poled beacon are received by an onboard sensor. The output of the ultrasonic transducer is amplified, rectified and compared to a reference voltage before being outputted to the position controller. If the position controller determines that the ultrasonic pulse has been received, it records the timer information and immediately poles the next beacon.

The Time-of-Flight (TOF), that is the time that the signal takes to reach the robot sensor, is then used as a measure of the robot's distance from the beacon. The Positioning Controller will be responsible for conditioning the time of flight information for input to the system processor.

5.5 System Constraints

The beacons will be pre-placed in the room as shown in Figure 25. One beacon (beacon A) will be placed in a corner and the other (beacon B) will be placed a pre-defined distance away from the corner beacon along a chosen wall. By placing beacon A in a corner, we reduce the range for which it must detect and transmit signals from 360° to 90° degrees. Although it would be optimal to have beacon B in a corner as well, we have decided to place it a pre-defined distance away from beacon A. The pre-defined distance simplifies the calculation of the robot's position in an X-Y coordinate plane. Since rooms are of varying length, beacon B cannot always be placed in a corner while maintaining the same pre-defined distance between beacons. Nonetheless, the range of sensitivity necessary for beacon B is reduced from 360° to a maximum of 180°. The transmitter/receiver onboard the Gerbil, on the other hand, will have the full 360° range of sensitivity. Since the robot will be facing the beacons at various angles, the onboard transducer must be able to both emit and detect pulses from any direction.

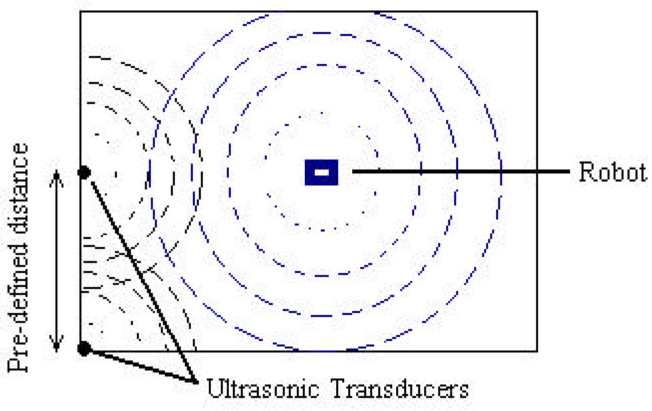

Figure 25: Beacon System

Secondly, ultrasonic transducers typically have a sensing envelope of roughly 30° as shown in Figure 26.

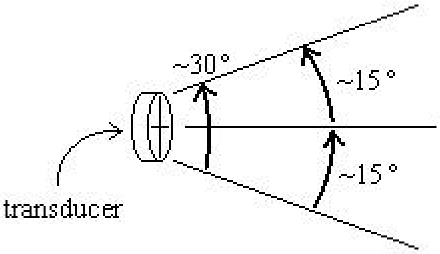

Figure 26: Transducer Sensitivity Range

Therefore, to achieve a range of 90° and 360° we designed each beacon and sensor to be a group of 3 to 12 transducers set one next to the other, as shown in Figure 27.

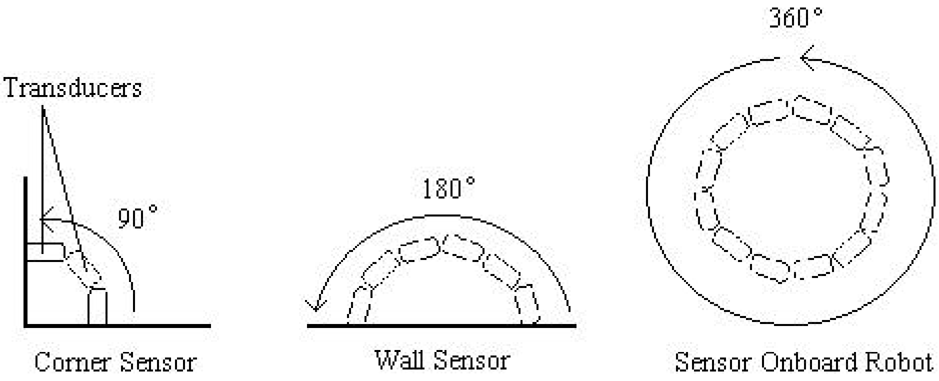

Figure 27: Corner, Wall, and Onboard Sensors

Thirdly, all parts of the room must be in view of the two beacons. If one or both beacons are blocked from the robot sensor range, the onboard receiver will not be able to detect the ultrasonic pulses from the beacons, and the robot will be unable to calculate its position. Figure 28 shows a room that meets these requirements and a second room that does not meet the requirements.

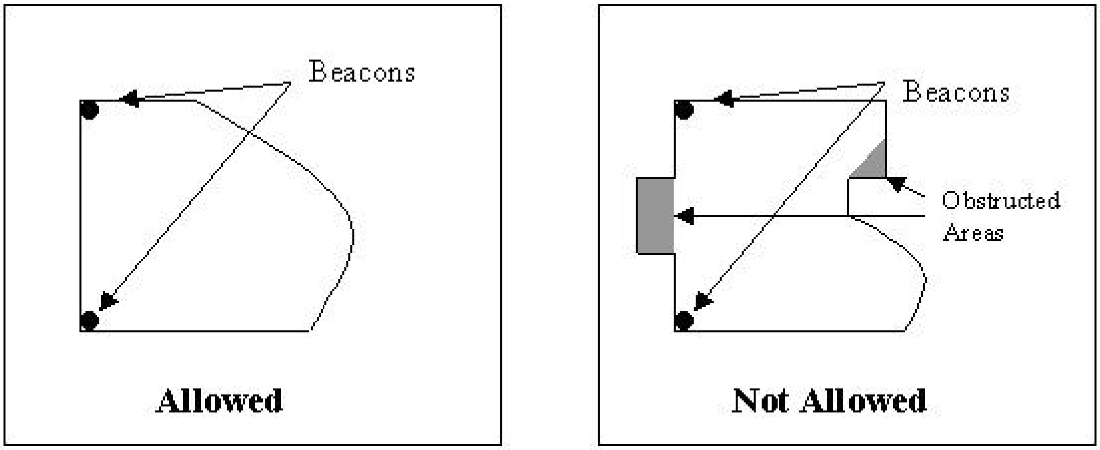

Figure 28: Room Restrictions

5.6 Potential Problems with a TOF System

The use of ultrasonic waves may cause problems due to the fact that the propagation speed of ultrasonic waves is affected by changes in temperature and humidity. For example, a change of 30° F may cause an error of 0.3 meters for a measured distance of 10 meters[1]. However, because the robot will be operating indoors, we do not expect any major changes in temperature and humidity that may affect the robot's accuracy.

---

[1] Example taken from Navigating Mobile Robots Systems and Techniques by Johann Borenstein, H.R. Everett, and Liqiang Feng, 1996, p.66

Testing

Positioning System

The Positioning System will be tested as follows:

1. The PIC will be loaded with a special test program that does only the algorithms associated with the positioning system.
2. The robot will be placed at various locations with relation to the beacons.
3. The output of the micro-controller, which will be a binary representation of the measured distances, will be checked with a probe at each location.

Sensory System

The Sensory system will be tested as follows: The wheel bumper sensor will be pushed at various angles and the input to the micro-controller will be tested with a probe.

Robot Control System

The Robot Control System will be tested in the following ways:

1. Robot will be loaded will increasing increments of weight to determine the maximum allowable weight of components.
2. Robot will be given various test inputs to determine frequency needed for desired speeds.
3. Robot will be given various test inputs to determine how well it responds to commands to turn left and right, and go forwards and backwards, and also to stop.

System Processor

The System Processor will be tested in the following ways:

1. Mapping will be tested by placing the robot in different locations, and verifying that the correct map is stored in memory.
2. Verify that once the map has been generated, the robot performs the implemented pathing algorithm.

References

[1] David Lee *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot*, Cambridge University Press, Cambridge;1996

[2] Microchip Technology Inc, *PIC16C77X* (Data Sheet);1999

[3] Microchip Technology Inc, *PIC16F8X* (Data Sheet);1999

[4] SGS Thomson Micorelectronics, *Stepper Motor Driving* (Application Note); 1995

[5] Allegro Microsystems Inc, *5804 BiMOS Unipolar Stepper-Motor Translator/Driver* (Data Sheet); 1998

[6] Douglas W. Jones, *Control of Stepping Motors, A Tutorial*, http://www.cs.uiowa.edu/~jones/step ;1998