Peppy™ Sonar Project Final Report

ILLINOIS INSTITUTE OF TECHNOLOGY
Christopher L. Jones
Omar Zrien
7/29/04

Abstract – As robots develop, so must their ability to interact with their environment. Multiple methods of robot interaction, ranging from laser scanning to video pattern identification, are in development world wide. Regardless of the method, the objective remains to develop a system that will better allow a robot to manipulate its surroundings with less user input. This paper details multiple approaches to sonar driven object identification. The primary focus of the project is to develop the simplest possible functioning example and to lay a foundation for future research into object identification. In this report, several theories are developed, problems with each are discussed, solutions and alternatives are suggested and the current project deliverables are presented.

INTRODUCTION

Current practical implementations of sonar are limited but can be found in products like Polaroid Cameras, which utilize sonar to track the range of a picture. Similar range finding technology can be found in robotic systems throughout the educational field. Software like the PYRO Robotics Module[1] are used at the undergraduate level and involve students in robotics projects using, among other things, straight line sonar sensors, usually mounted on all sides of a robot, in order to identify obstacles around the robot.

The use of sonar systems is made most practical because of its cost-effective nature. Sonar also requires little processing power for the existing applications and it covers a larger field of view (FOV) in a much smaller time interval than other methods.

The objective of the Peppy™ Sonar Project is to develop the simplest possible sonar object identification and acquisition system and incorporate it into the Peppy™ robot. The system is designed to search for and identify an object and proceed to retrieve the object for the user.

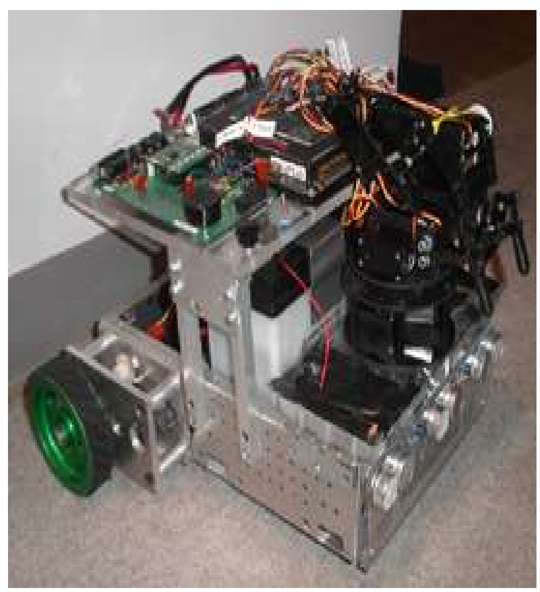


Figure 1.1 The Peppy™ Robot with the current sonar system. The robot was redesigned to accommodate the new system and provide room for future expansion.

[1] An introduction to PYRO Robotics Software can be found at: http://emergent.brynmawr.edu/wiki/index.cgi/PyroModulePythonIntro

1. SONAR BASICS

The sonar system is designed to send and receive an ultrasonic pulse and, based on analysis of that pulse, determine the coordinates of a target object. As the sonar sends a pulse it records the times associated with the disturbances that are received. Each disturbance represents a potential object, and the associated time is used in conjunction with the speed of sound to determine the distance to the object:

$$\frac{\text{Time}}{2} \times \text{Speed of Sound} = \text{Distance}$$

The sound wave must travel to the object and the echo must travel back, therefore the recorded time is twice the time it takes for the wave to travel to the object and must be divided by two.

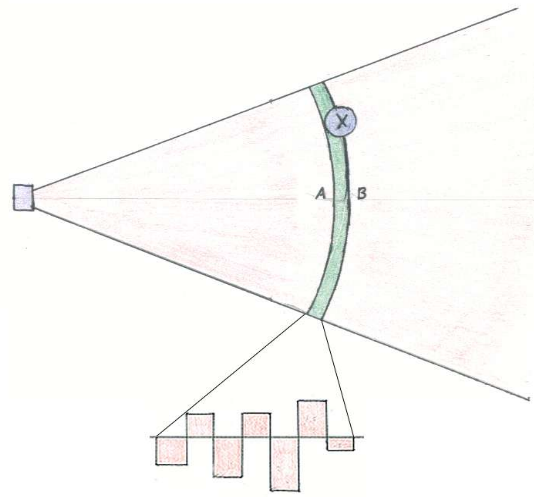


Figure 1.2 Radius Measurement: Detection of object X within a sensor's FOV between radii A and B. The sensor sees only surfaces facing it. Square wave feedback from sonar sensors represents reflections heard between A and B respectively.

The distance is shown in figure 1.2 by radii A and B. These radii correspond to the leading and trailing edge respectively, of the echo analog signal. Signal manipulation is discussed further in section 3D, Triangulation.

Object X in figure 1.2 reflects the sonar pulse from all surfaces facing the sensor. There is no way for the sensor to view the side of object X that faces away. Further reflection patterns are discussed in section 3B, Two Dimensional Mapping.

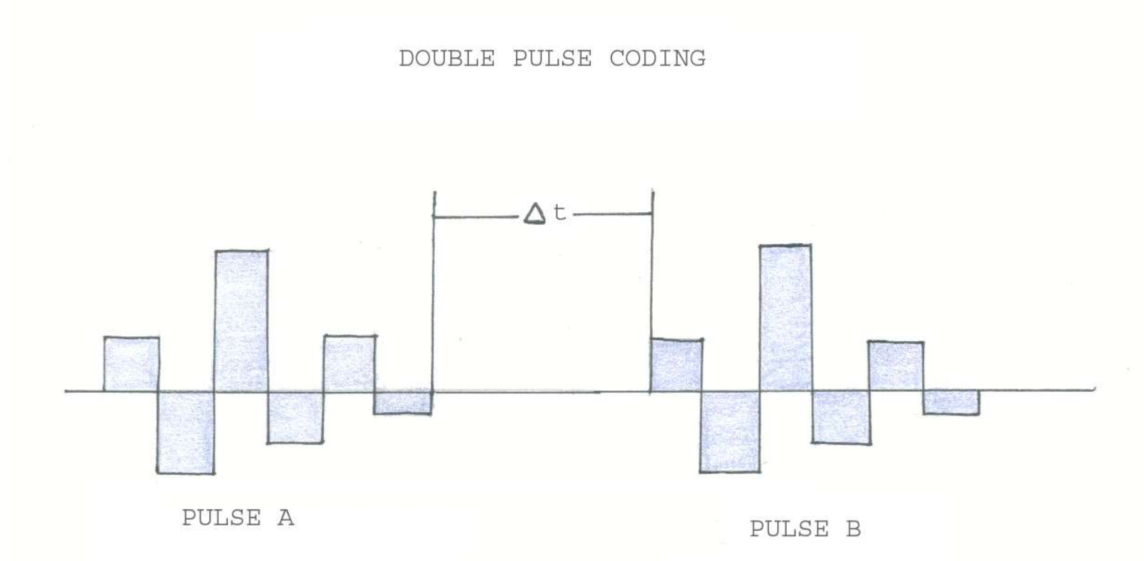


Figure 2.2 Double Pulse Coding: Pulse A and B are identical signals output by the transmitter. The time difference Δt is a random spacing unique to each transmitter/receiver pair

.

2. THE SIGNAL

In order to avoid interference when multiple sensors are employed at the same time, a method called "double-pulse coding" (Figure 2.2), developed by Professor Lindsay Kleeman of Australia's Monash University, can be employed. This method of sending and receiving sonar signals involves transmitting two pulses, a random unit of time apart, and listening for echo patterns that are the same unit of time apart. Each transmitter and receiver pair uses different randomized pulse spacing. The use of double-pulse coding prevents the sonar receivers from recording echoes from the wrong transmitter. At this point, Double-Pulse coding has not been implemented into the Peppy™ robot because our signal sending is done in series rather than in parallel. The use of parallel pulses, or much faster series pulses, would only be required if many sensor pairs were used.

The circuit, constantly receiving data from the sensors, must store the data for future use. Data storage is entirely dependant on the method of analysis. The Labview, signal analysis program could prove a powerful tool in the study of ultrasonic echo waveforms, as well as a powerful wave manipulation tool. Labview has the power to serially input data, a process supported by the existing sonar circuit, and manipulate the data to analyze, calculate, and store any necessary object, wave and position data. Labview however, runs on a computer and requires a connection to the robot.

A simple solution would be to incorporate a wireless infrared serial communication system to relay data from the robot to a computer. Such a system would require proximity but would allow analysis to be conducted on the computer without a teather.

Another solution, applicable after wave analysis has been explored and the methods have solidified, would be to add a more powerful processor to the robot that could be programmed and utilized for the analysis. The most convenient method of data storage, and the most advanced, would be the use of an ALFAT chip. The ALFAT chip allows for the manipulation of the FAT32 file system found on many hard drives and removable storage devices. The ability to store data on such media would prove a powerful tool for the development of the robot.

3. OBJECT IDNETIFICATION

The sonar system, identifies objects based on analysis of an analog signal (Fig. 2.1). Each sensor will have read an analog signal full of echo patterns from all objects in its FOV. In order to identify an object, the proper analog disturbance must be correlated between multiple sensors. This proves to be the most difficult and cumbersome part of the sonar identification system. Such calculation however, is the next step in the advancement of sonar technology. This section will discuss the limitations of a sonar system and methods for identifying objects and correlating sensors.

A. LIMITATIONS

Because of the nature of an analog wave, there is an inherent limitation seen when two objects exist at the same radius from the sensor (Figure 3.1). The two overlapping objects are heard as a single pulse (Figure 3.2). The problem is not quite as fateful as it may seem. When multiple sensors are employed, the same objects can not exist at the same distance from all sensor pairs. The program can take advantage of this and identify jumbled waveforms based on information from other sensor pairs or, if enough alternate sensors exist with good data, it can ignore the jumbled waves and use the good data.

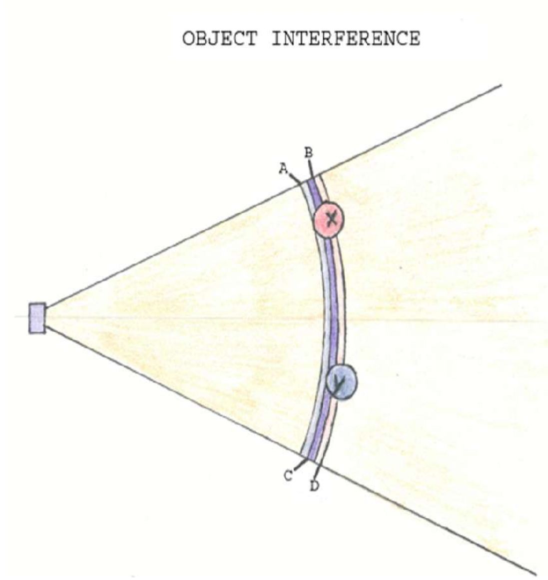


Figure 3.1 Object Interference: Objects X heard at radii A and B, and object Y heard at radii C and D, overlap and look like a single object between radii A and D.

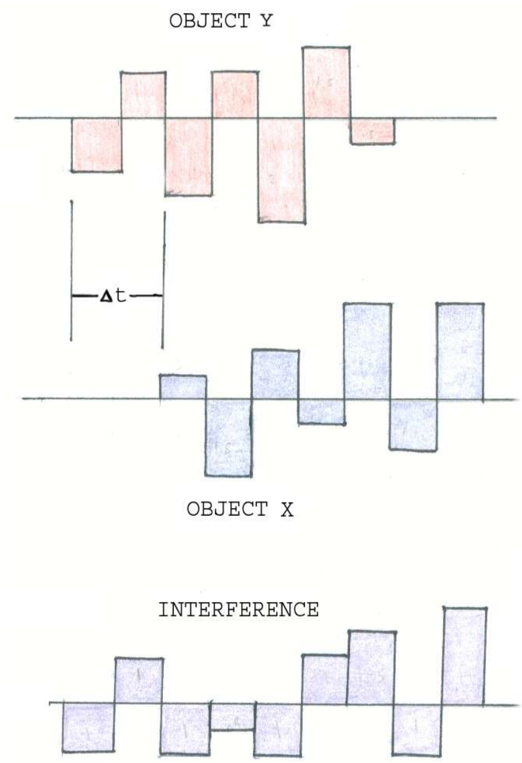


Figure 3.2 Signal Interference: Wave X is heard between radii A and B, and wave Y is heard from radii C and D. The sensor only hears the Interference pattern made from the overlap of wave X and Y and the objects are therefore indistinguishable.

Another probable solution to the object interference problem, and one that may also be used in conjunction with the previous solution, is correlation with video pattern recognition or laser scanning to identify potential jumbled targets.

B. 2 DIMENSIONAL MAPPING

Navigation is an integral part of robot-environment interaction. One method of navigation is two dimensional mapping, a method where a flat, digital image of a surrounding environment is constructed and used to define robot position limitations. The robot would maintain a current position inside the map which could be utilized to set various gains limiting motor velocity as the robot nears obstacles.

Two dimensional mapping is based on recognizing signal patterns that correlate with standard boundary features in an environment. This method of applying sonar deals exclusively with wall features, identifying flat and sloping surfaces, outside and inside corners, and other edges. Research into this method of identification can be conducted using labview to capture analog waves. Comparing those captured waves may help to identify patterns for recognition.

Typically applicable to mapping rooms and large objects, this method is not easily applicable if there is an abundance of small objects, as it is possible to overlook small objects below the sonar field of view (FOV) or just too small for the sensors to perceive. Small objects also have the potential to band together to look like larger obstacles, thus inhibiting robot motion where it otherwise would be clear to move.

Professor Lindsay Kleeman of Australia's Monash University, has conducted extensive research into two dimensional mapping. He has developed mobile computing systems that track around a room while updating a two dimensional map and avoiding obstacles. His research can be found at: http://www.ecse.monash.edu.au/centres/irrc/LKPubs/ and may prove beneficial for future developments in the Peppy™ Project. Efforts were made this semester to contact the professor but no reply was ever received.

C. OBJECT UNIQUE ECHOES

One method to correlate sensor data could prove intrinsic in every observed echo. Similar to wave recognition for 2D mapping, such a method could be utilized if every object possesses a unique echo pattern as recorded by the sonar system. The echo waveform would be dependant on the shape of the object as well as the objects density. Similar to transfer of energy on a string or in wind instruments, the characteristics of the echo may be found to be dependant on the object as every object possesses an inherent sonic absorbance. The use of such an identification method would allow the robot to not only distinguish between a volleyball and a medicine bottle, but also differentiate a can of soup from a can of soda, nearly indistinguishable by shape alone.

With this method there exists a dependency on orientation as it has been observed that objects without cylindrical symmetry reflect drastically different echoes as orientation varies. Some similarity may still exist in these waveforms but at this point, object identification solely based on echo characteristics does not seem probable; though future research may indicate otherwise.

D. TRIANGULATION

The next advancement in object acquisition, after the target has been identified, is to determine its position relative to the robot. Any coordinate system may be employed for this purpose but, as a result of Peppy's™ two wheel mode of steering, cylindrical coordinates have been deemed the most applicable. Peppy's™ design includes two encoders, one in each transmission. Each encoder reports pulses, observed as they rotate, to the controller. These pulses can be used to determine the distance each wheel has traveled and, based on the difference between the wheels, the angle of the robot relative to its starting point, or its objective.

In order to determine the position of an object, three or more times, Δt, are required. The sensors should be mounted in a front-facing plane to simplify correlation of sensors. The array could be mounted on a servo, but the angle would need to be taken into account when pulsing and recording data, and the speed of oscillation would need to be small enough to ignore the rotation, unless it is compensated for in the calculations.

(INSERT CALCULATIONS 3.1)

In order to make these calculations, we are assuming that an object can be reduced to a point, or a series of points, that can be calculated separately. This can be done by matching waveforms and using a time at a consistent point within the waveform.

The use of the leading edge of a waveform, the start of the analog signal, would calculate the position of the front of the object for each sensor. Each sensor would pick up the surface closest to it first, but this could be taken into account with the size of the object, or ignored for small, cylindrical objects.

Figure 3.2 demonstrates the triangulation of a small, cylindrical object using the leading edge of the observed echo. In the diagram, "D" is the actual distance to object A and "d" represents the distance to the object as the system triangulates it via leading edge calculation.

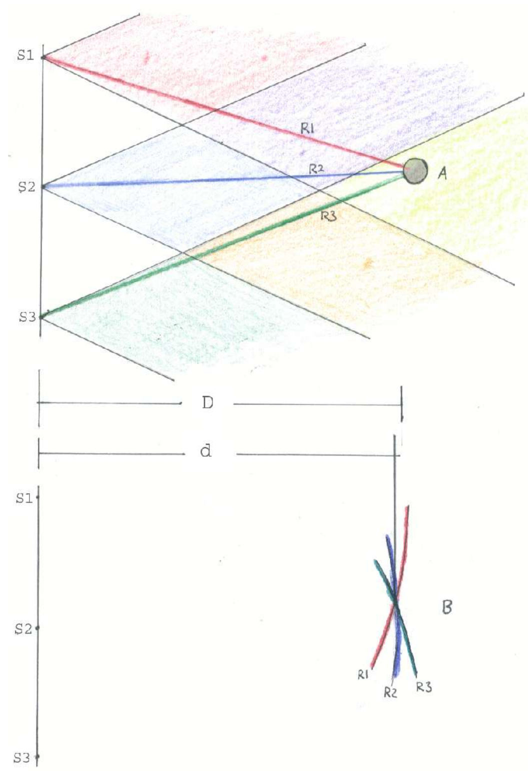


Figure 3.2 Leading Edge Correlation: The difference between D and d is small.

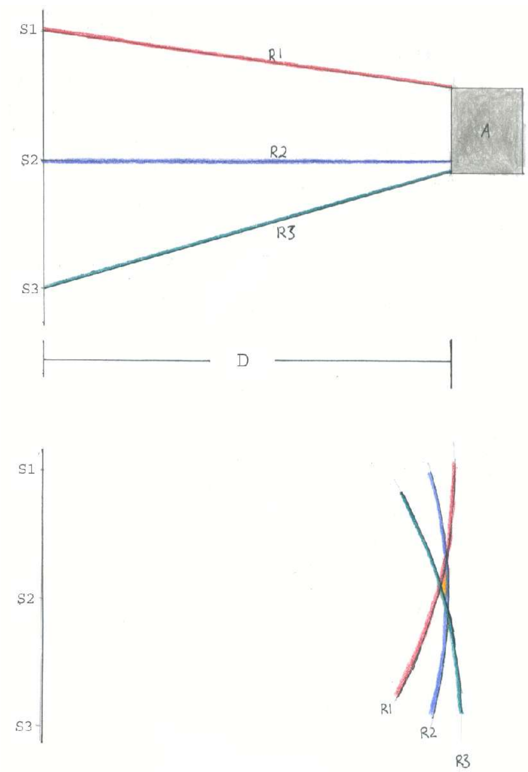


Figure3.3 Leading Edge Ambiguity: Ambiguous results from the three calculated radii result in an object that is impossible to triangulate.

Asymmetrical and larger objects (Figure 3.3) prove more difficult to triangulate. Each sensor observes a drastically different point as its leading edge and, as a result, it is impossible to triangulate the object. Further wave analysis would be required to enhance the accuracy of triangulation for objects of arbitrary shape.

With the difficulties encountered using mathematical triangulation, it proves to be an impractical method for object locating. The method of triangulation remains valid, though the previously discussed implementation is impractical. If the same method could be implemented with multiple points, at any time the data is available and also incorporate object identification, it would be much more efficient and practical. The next section discusses a theoretical construction of such a system.

E. MATRIX ENVIRONMENT OBSTRUCTION SENSING

The Matrix Environment Obstruction Sensing (MEOS) system keeps track of where each individual sensor hears an obstacle and steadily verifies objects as more sensors hear obstructions at the same location. To accomplish this, the system maintains a matrix that represents the environment, each cell representing a volume in space. The resolution of the environment is controlled by its size and the number of cells used. This matrix can also be expanded and contracted within the system depending on the size of the environment and the required level of interaction.

The sensors on the robot are continually scanning for obstacles. At any instant in time, there exists a calculable cone where the sensor has the potential of hearing an obstruction (see Figure 3.4). When an obstruction is heard, it is registered in the form of an analog disturbance. As a sensor picks up a disturbance, the time at the start and end are recorded and the radii can be calculated as previously discussed.

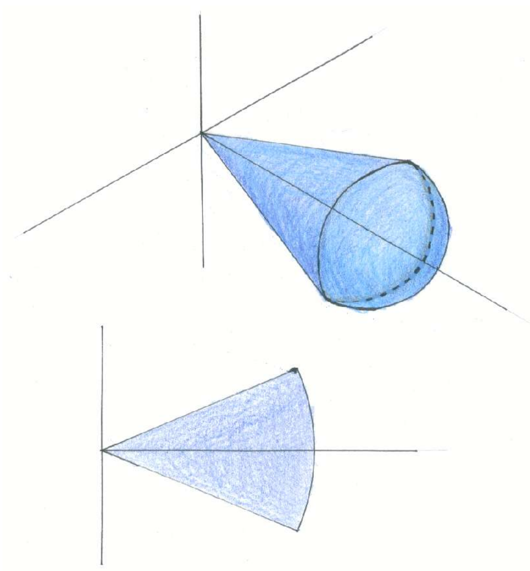


Figure 3.4 Potential Cone:

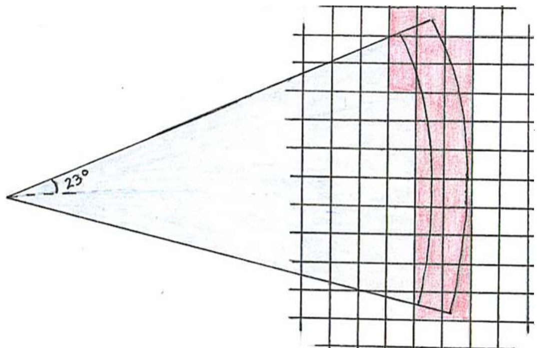
In order for the MEOS system to apply the received sensor data to the developing matrix environment, it is necessary for the system to know exactly where the sensors are relative to the environment. The robot has the capability to track its movement within the environment via the existing encoders, and external measurements can be programmed into the robot to determine sensor location or a template of known orientation can be used to allow the robot to calculate settings without programmed input. Because the system then knows the location of its sensors in the environment, it can also calculate the matrix equivalent of the potential cone. The MEOS system then uses the calculated radii for the disturbances and flags all cells between them inside the cone of potential (see Figure 3.5 It would be prudent, in this case, to be generous with the definition of 'inside' and flag pixels that are close, but not inside the detected radii.

Figure 3.5 Flagged Cone of Potential: The sonar sensors detect an object within the curved boundaries. Those boundaries, when transferred to the MEOS matrix will result in the flagging of all intersecting pixels (red above).

Within the matrix environment, objects can be determined to be solid as multiple sensors observe obstacles at the same radius. As a second sensor detects a disturbance, its cone and radii band are flagged and area that overlaps with cells previously flagged are now twice flagged. The number of sensors required to determine a solid obstruction can be determined based on the number of available sensors and trial and error observation of the system. The same method can be used to determine empty space; the clearing of flags when multiple other sensors hear nothing at the same location.

As an object becomes solidified, it will form at first, as a shell. The shell exists because multiple sensors detect the face of the object pointing toward them. In order to identify the entirety of an object, the robot must circle to the opposite side and make observations since the robot cannot infer what may be there.

Sonar observations alone will not, in most cases, be capable of fully blocking in an object. It becomes the responsibility of the MEOS system to extrude and complete the accurate solidification of an object. The base of an object for example, can be extruded to the floor because the robot knows the distance to the object and its own height from the floor. This method is applicable after symmetry has been identified and may not be valid if near-ground features are pivotal in object identification. Further solidification of the object body may or may not prove necessary depending on the method of identification. If leeway is given in template matching between the target



objects. An object template is pre-recorded object characteristics stored in a matrix in the same manner that an object is identified. The robot associates a command “noun” with each template and when that noun is referenced the robot compares observations to that template to identify the target object.

The concept of an environment may also be expanded to allow extensive, unique interaction within various environments. It is feasible to use separate memory hardware for various environments storing target object information for unique environments on separate cards from more common objects. Swapping or switching between cards would allow the robot to store a multitude of data for a specific environment and not be required to sort through that data while in other environments when it would not be necessary. Using separate sets of data for each environment would also allow for separate interaction sequences. Tasks like turning on lights that vary from one environment to the next would be more convenient if you didn’t have to specify “how” to the robot.

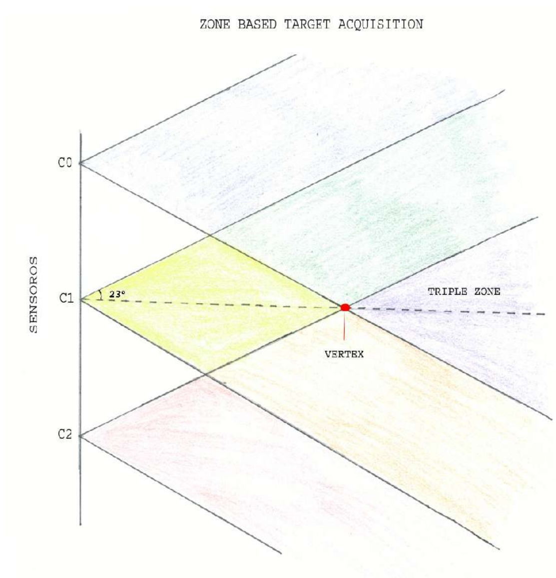


Figure 4.1 Tracking Zones: Each zone can be seen by a different combination of sensors. The vertex is a point of calculable distance in front of the sensor plane.

4. CURRENT IMPLEMENTATION

A. ZONE BASED OBJECT TRACKING

The Peppy™ robot is currently equipped with a fully functioning zone based object tracking system. This system allows the robot to track the nearest object, acquire its exact position relative to the robot, and further manipulate it for the user. This method is simple, but demonstrates the capabilities of sonar technology.

The ultra-sonic transceivers are mounted to the front of the robot in the same plane. Each sensor has a solid FOV that overlaps with the other sensors in front of the robot. The overlapping fields of view create six zones in front of the robot, each zone identified by the sensors that can hear obstructions in them (Figure 4.1). The robot identifies these zones as the sonar circuit returns which sensors hear obstructions. Built-in timers limit the waiting time for hearing an echo and control the length of the sensors FOV.

As the robot identifies which zone the target object is in, it responds with various gains unique to each zone to turn the robot toward the object and position the object in the triple zone, where all three sensors detect the disturbance. A rough distance is then calculated to the object and the robot moves toward the target, the goal is to position the robot at the vertex of the triple zone. Once the object is positioned at the vertex of the triple zone, because the distance between sensors and the angle of their FOV is constant, the object's position is known to the robot. From that point, the robot can proceed to manipulate the object as the user specifies.

There is no actual identification taking place, the robot is programmed to track the first visible object and is therefore limited in use.

B. FOLLOW MODE

Though limited in its ability to track specific objects, the current implementation has led to the development of a unique and useful robot interaction tool; the leash.

The efficient tracking method currently in use on Peppy™ can be utilized to allow the robot to follow a hand held leash. This leash consists of an object at the end of a stick that is to be held in front of the robot. When the user speaks the command "Peppy follow," the robot will proceed to identify the leash and track it. Peppy™ will however, limit itself to maintaining the object at the vertex of the system. As the user moves the leash, Peppy™ responds using the previously mentioned gains to move along with the leash.

(picture of leash in front of robot)

Appendix A: Sonar Synopsis

Intro into PICs

The sonar unit works independently from the FRC driven robot. It consists of an 8-bit Microchip processor commonly known as a “PIC”. This processor was chosen because of its ease of use and inexpensive price. In fact the processors we used were free samples from Microchip's web site, www.microchip.com. The processor we chose is the 18F458, a processor commonly used by both experienced hobbyists and new programmers. We used Microchip's ICD 2 to program the processor. This programmer is considered top of the line as it gives more debugging ability to the user than any other programmer currently out. The ICD 2 costs approximately $200, but there exists many PIC programmers on the market for those seeking a more inexpensive product. More information can found from the 18f458 datasheet, which can also be found at Microchip’s web site.

The compiler we used is called the C18, a free demo can be downloaded at Microchip’s web site, as well as the Integrated Development Environment (IDE) called MPLAB v6. The compiler download will also install documentation on software libraries and specifics on the compiler.

Concept

The sonar unit is used to track the proximity of close objects. This is done by emitting a sonic pulse and then receiving the same pulse after it has reflected off a nearby object. Knowing the speed of sound and the time difference between sending and receiving the sonic pulse distance to an object can be calculated, i.e. $v=d/t$. By constantly reading in three different transmitter/receiver pairs, our robot can track moving objects to follow, such as a “leash”, or it can be used to retrieve objects.

Specifics

The processor sequentially emits 25kH pulses into each transmitter. The corresponding receiver is then enabled. The firmware waits approximately 16 milliseconds for a response from the corresponding receiver, otherwise it times out. The receiver signal is amplified in the audio amplifiers and then sent into the 10-bit analog to digital converters in the PIC. This is how magnitude of the wave is recorded by the PIC. By observation it has been noted that when no sonar pulse is “heard” the ADCs return a value of approximately 300. When a pulse is received the ADCs returns an oscillating value of about 200 and 800. The time difference is found by utilizing the PIC’s timer. A timer interrupt is enabled which increases a count variable every 16 microseconds. The value of this variable is recorded right before the sonic pulse is sent, and after a pulse is received.

Explanation of Circuit

The USART header on the circuit provides TTL level communication to the First Robot Controller on the robot. The blue wire in the photograph provided is the TX (Transmit) and the orange wire is the RX (Receive). Although the sonar unit doesn’t receive any data for zone tracking, future models may require the capability so it was included in the circuit design. Below the USART is the Communication jumper. Place a bridge between the top two pins and the PIC will send data in 5 byte packets. The first two bytes have a value of hexadecimal FF, used to synchronize data, the following three bytes each hold the time difference recorded by the PIC for each corresponding transmitter/receiver pair. The time differences are measured in an arbitrary time unit where one unit equates to 16 microseconds. In other words, multiply the time differences by 16 to find the difference in microseconds. This may be done for purposes of calculating absolute distances. For debugging purposes, the lower two pins of the header can be bridged (DO NOT bridge all three pins). This will cause the PIC to emit data in ASCII format following the RS-232 protocol, which allows users to directly observe the time differences instead of sending the data into another TTL level device. The data can be received through the db-9 serial port interface. The windows program hypertrm, can be used to see the data; the properties of the COM connection is 19200 bits per second, 8 data bits, no parity, one stop bit and no flow control. A more sophisticated program can also be used such as MATLAB. The Transmitter header is below the Comm header. The top pin provides a common ground to the transmitters, and the other three pins provides the 25kH pulse to each transmitter. The Receiver header connects to the receivers, the top two pins go to the first receiver, middle two to the second, and bottom two to the third. The program port below the PIC is wired (right to left) into the PIC’s MCLR pin, power, ground, pin RB7 (PGD) and pin RB6 (PGC). This port allows in-circuit programming, or flashing the firmware without having to separate the processor from the circuit. Above the top audio amplifier is the power header. In the photograph provided the red wire marks the positive header, and the green marks the ground.

Circuit Theory

Once the 25kH pulse is emitted from the PIC and into a transmitter the firmware then waits for a signal. If an object successfully reflects the sonar wave back into the corresponding receiver the signal is sent into an audio amplifier. According to the audio amplifier's datasheet the gain should be approximately 200 due to the capacitor across pins 1 and 8. The output of the audio amplifiers is then sent into the ADCs of the PIC.

Materials List

18F458 40-pin PIC processor
Quantity: 1
Purpose: The "brain" of the circuit where the firmware is stored and executed.
Place of purchase: Free samples were taken from www.microchip.com, bulk purchases can also be bought at this site

LM386N 8-pin Audio Amplifiers
Quantity: 3
Purpose: Amplify the received signal
Place of purchase: www.digikey.com or RadioShack

4.7uF Electrolytic Capacitors
Quantity: 2
Purpose: Decouplers, help prevent spontaneous reset of processor due to variances in supply voltage.
Place of purchase: www.digikey.com or RadioShack

22 pF ceramic capacitors
Quantity: 2
Purpose: Smooths the oscillating signal the crystal outputs
Place of purchase: www.digikey.com or RadioShack

470 nF polyester capacitor
Quantity: 3
Purpose: Connects to pin 1 and pin 8 of the Audio Amplifiers, according to the datasheet this increase the gain to 200
Place of purchase: www.digikey.com or RadioShack

16.000 M Htz Crystal
Quantity: 1
Purpose: PIC uses this as a timing reference
Place of purchase: www.digikey.com or RadioShack

10 kΩ Resister
Quantity: 1
Purpose: pullup on MCLR pin on PIC, necessary to execute firmware
Place of purchase: www.digikey.com or RadioShack

MAX233 IC
Quantity: 1
Purpose: Converts TTL logic communication from PIC into RS-232 which is what PCs use.
Place of purchase: www.digikey.com or RadioShack, free samples at http://www.maxim-ic.com

DB9 female port
Quantity: 1
Purpose: Physical interface for PCs

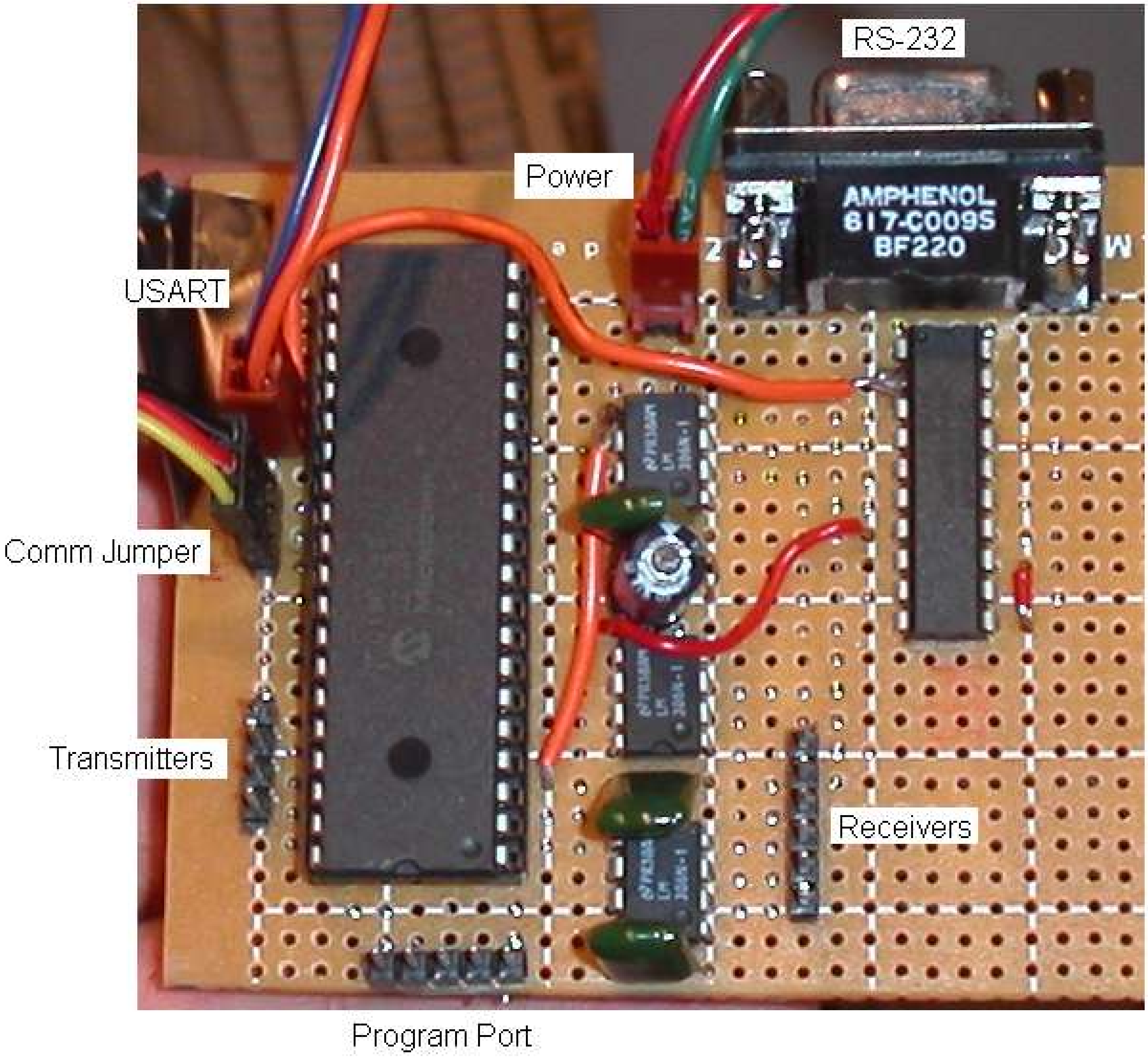
Place of purchase: www.digikey.com or RadioShack
25kH Ultrasonic 25kHz Transducers
Quantity: 6
Purpose: A 25kH pulse is periodically sent to three transducers, the other three are used as receivers similar to microphones except these transducers will only "hear" 25kH sound waves. This simplifies the filtering which is normally done by an op-amp circuit, requiring more circuit space and hardware.
Place of purchase: www.sparkfun.com

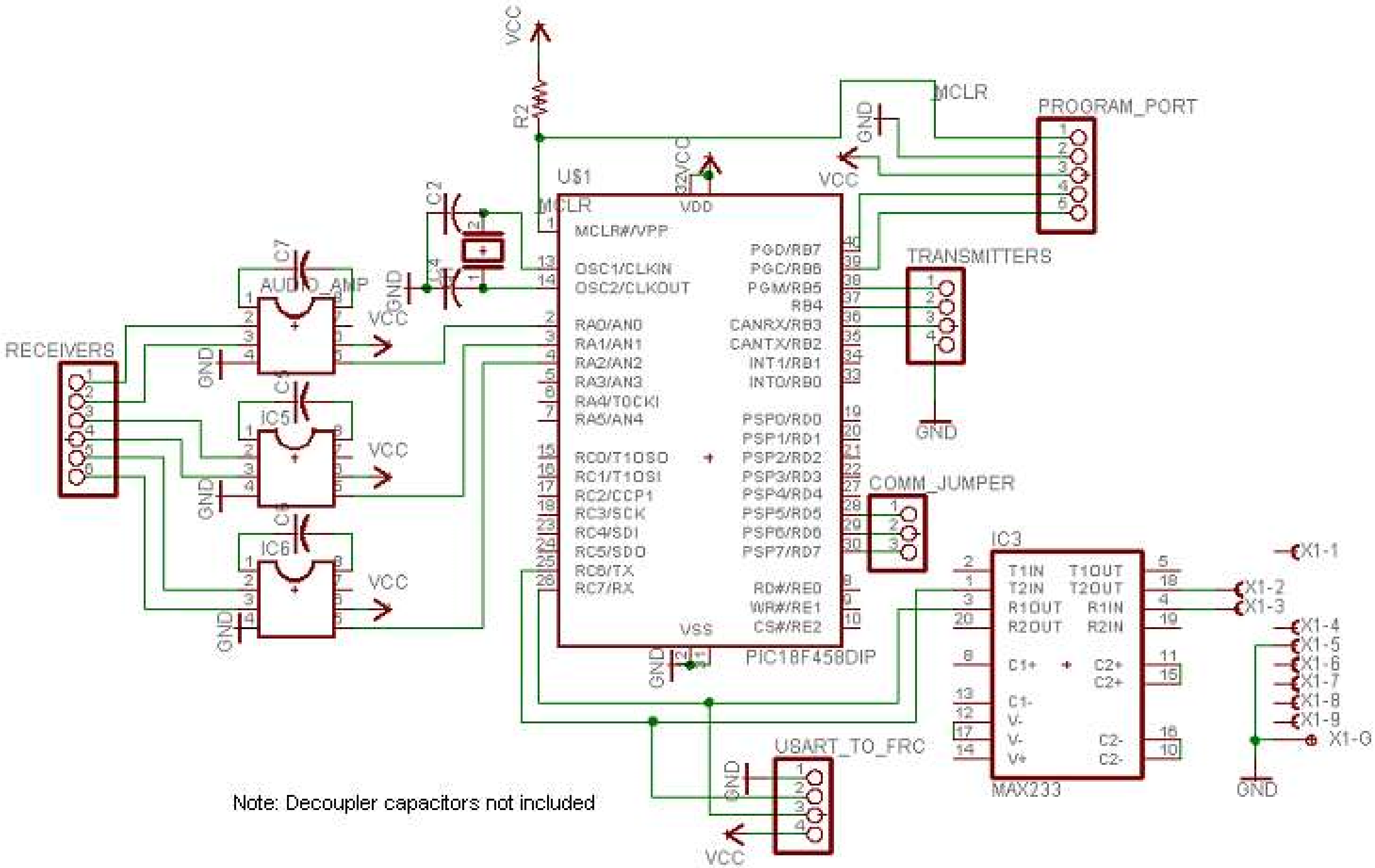
40 Pin IC Socket
Quantity: 1
Purpose: Allows PIC to easily disconnected or replaced
Place of purchase: RadioShack

PC Board
Quantity: 1
Place of purchase: RadioShack

Materials recommended for future projects involving sonar unit:

Oscilloscope capable of accurately measuring frequencies (25kH worst case).
Soldering iron.
Circuit Design software (we used Eagle Layout Editor 4.11r2, free demo available at http://www.cadsoftusa.com/
PIC Programmer (we used Microchip's ICD 2)
Regulated power supply.


RS-232
Power
AMPHENOL
817-C0095
BF220
USART
Comm Jumper
Transmitters
Receivers
Program Port



RECEIVERS
AUDIO_AMP
IC5
IC6
C7
C5
C6
C2
C4
R2
VCC
GND
U$1
MCLR
MCLR#/VPP
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2
RA3/AN3
RA4/T0CKI
RA5/AN4
RC0/T1OSO
RC1/T1OSI
RC2/CCP1
RC3/SCK
RC4/SDI
RC5/SDO
RC6/TX
RC7/RX
VDD
VSS
PGD/RB7
PGC/RB6
PGM/RB5
RB4
CANRX/RB3
CANTX/RB2
INT1/RB1
INT0/RB0
PSP0/RD0
PSP1/RD1
PSP2/RD2
PSP3/RD3
PSP4/RD4
PSP5/RD5
PSP6/RD6
PSP7/RD7
RD#/RE0
WR#/RE1
CS#/RE2
PIC18F458DIP
PROGRAM_PORT
TRANSMITTERS
COMM_JUMPER
USART_TO_FRC
IC3
T1IN
T2IN
R1OUT
R2OUT
C1+
C1-
V-
V+
T1OUT
T2OUT
R1IN
R2IN
C2+
C2-
MAX233
X1-1
X1-2
X1-3
X1-4
X1-5
X1-6
X1-7
X1-8
X1-9
X1-G
Note: Decoupler capacitors not included