**Lisa Anthony**
**Robot Building Lab**
**Grad Project: Preliminary Plan and Outline**
**Version 2**
**Due November 28, 2001**

# 1 Task

I propose to investigate the accuracy of the sonar sensors we have been using in class; many groups received readings which did not make sense, given certain assumptions that were made about the operation of these sensors. To attain a more reliable sonar sensor would be both interesting and practical for future use of these sensors.

## 1.1 Problems to Solve

In the graphs of the elementary in-class experiments conducted with the sonar sensors, several problems or inconsistencies are apparent:

- Surveys of *sonar_closeup()* often return spike values far below the appropriate range. Also, values returned by *sonar_closeup()* sometimes explicably drop in the middle of a survey, but then recover to the appropriate range.
- Surveys of both *sonar_closeup()* and *sonar_sample()* often return spikey values instead of a smooth line.
- In most cases, *sonar_closeup()* and *sonar_sample()* return identical patterns of readings, but not at identical values, which is inexplicable given our current assumptions.
- In many cases, when sweeping over an obstacle with either *sonar_closeup()* and *sonar_sample()*, the readings at the beginning and end of the range of readings which correspond to the obstacle's location, are much higher than in the middle, resulting in a U-shaped curve.

Other issue(s) which are notable:

- In order to interpret and verify values from the sonar survey, it is necessary to have a complete and accurate map of the world, to ensure that what may seem like "erroneous" readings are not actually obstacles which you did not realize were interfering.

# 2 Hardware

The only hardware required for this project is a sonar sensor mounted on a servo motor attached to a HandyBoard, in similar configurations to that used in class. No drive train or other sensors are needed. The experiments will be conducted in the same manner used in class: the sonar/servo/HandyBoard combination takes readings and sends them across a serial line to a PC.

# 3 Software

Each experiment will require some type of control program which sets up how the sonar takes the readings, obtains these readings and stores them, transmitting them to a PC, where they will then be graphed, using a program like Microsoft Excel.

The end result of this project will be a "filter" function to add to the sonar library: this function will mask from its users the procedure (determined by experimentation) used to obtain the most accurate sonar reading possible. Within the function will be incorporated the points learned in the experiments (discussed in Section 4) concerning the validity of the *sonar_closeup()* or *sonar_sample()* methods in certain situations, whether and how to survey the environment, and whether and when to adjust sonar position using the servo motor.

The outputs or computed values of this function will be three quantities: a distance representing how far away the closest point of the closest obstacle is; a length representing the estimated size of the obstacle in the robot's current field of view; and an angle representing the approximate skew of the object away from the robot. These three values will be 0 if there was no obstacle within the detectable range.

# 4 Experiments

## 4.1 Controlled Variables

There are many factors which affect the sonar's accuracy; it will be necessary to run experiments to determine how large this effect is and how this effect can be controlled and accounted for. Most notably, level of battery power, movement of the robot itself using its DC motors, interference caused by taking sonar readings too close in time together, movement of the sonar atop the servo motor, and orientation or distance of the obstacle(s) may have effects on the results and must be controlled during experimentation. It will not be possible to test for all possible combinations of the above variables; rather, a decision tree will be constructed to determine which are the most salient for our purposes.

Note: The true maximum range of the sonars is not extremely important in our robot world, since an obstacle more than 10' or 20' away is not that urgent in avoidance routines (although it will cause the need for more exploring to get as detailed a picture of the world). The timeout value is located in the **sonar.c** library and, as long as our assumption of the 20' maximum is less than the distance sound can travel a round trip in that timeout interval, the experiments will not be compromised.

## 4.2 Experimental Conditions

In the experiments conducted, it will be necessary to keep careful records of the world map, so that the obtained data can be later analyzed in the context of the experimental setting. This will help determine the types of patterns to look for in the "filter" function to be designed based on these experiments.

In all of the below experiments, versions will be run using both sonar sampling methods (*sonar_closeup()* and *sonar_sample()*. This is to attempt to determine why or when one should be trusted over the other. This requires two separate runs of each set of experiment conditions, to ensure that using the methods interchangeably are not interfering with each other.

## 4.3 Preliminary Experiments

- **AC vs (Random) Battery Power** Running the robot on AC power is not practical or even feasible in the real world; however, there is no reliably accurate way to determine level of battery power when running experiments. Some preliminary experiments will be necessary in order to determine if it is safe to assume that AC vs. battery power has or has no effect on sonar readings. If it is determined that there is no effect, further experiments can be run that do not take this variable into account. On the other hand, if it *does* have an effect, the later experiments will have to be run under both conditions (with the level of battery power being a very rough estimate based on time since last charging).

- **DC Motors Interfering** In the same manner, using the sonar sensors with our robots is often in tandem with path-planning or map-building or some other task which requires the robot be moving around in the world and taking sweeping surveys using the servo motor; these also could cause noise in the sonar readings. It is not too strict to require the sonar samples to be taken while the robot is standing still; therefore, all experiments will be run while the robot's motors are off and the robot is stationary within the world. If time permits, experiments can be done to determine the effect of the motors on the sonar by hooking up a motor which is not driving any wheels to the HandyBoard and comparing the readings when the motors are running or not running.

- **Sonar Reading Wait Interval** The third condition is interference that may be caused by taking sonar readings too close together in time, without allowing for any "signal die-down" time. Some wait time between sonar readings is not impractical, especially since it is likely to be in the range of a fraction of a second. The minimum wait time must be determined for use in later experiments. To figure out what values to test, a "binary search" strategy will be used: if a higher number still doesn't work, increase the estimate; if it is too high, decrease to about halfway between the initial two estimates; continue in this way until values stabilize.

- **Servo-Sonar Wait Interval** Related to this is the potential effect of the motion of the servo motor. It is not feasible to require the robots never to use the servo motors to take sweeping sonar surveys. Therefore, some minimum time interval to use as a "pad" between servo movement and sonar reading must be experimentally determined, to allow us to compensate for the possibility of servo wire noise. To do this, several conditions must be tested: first, behavior with no waiting time between sonar reading and servo motion; then, some preliminary hypothesis of an upper limit for wait time will be tested (the value obtained via the previously discussed experiment might be a good starting point). After this, experimental values to test will be determined based on the same "binary search" style algorithm discussed above.

### 4.4 Obstacle Experiments

- **Shape: Flat Edge, Convex Corner, Concave Corner** To determine the ability of the sonar to tell when an obstacle begins and ends (specifically addressing the U-shaped curve issue), experiments will have to be run with obstacles that appear to have different shapes in the sonar's field of view. Three conditions must be tested: the obstacle is a flat edge across the field of view; the robot is facing into a convex corner; and the robot is facing into a concave corner. Servo sweeping *will* be used in these experiments. One key to look for in these data are sharp spikes at either end of a flat, low range of readings, indicating the edges of a visible obstacle.

- **Orientation: Perpendicular, Angled Away, Angled Toward** To determine the ability of the sonar to determine the angle of displacement of obstacles, experiments will have to be run with obstacles these three different orientations: the obstacle is perpendicular to the 90-degree normal of the sonar's field of view (not angled); the obstacle is at a positive (+) 45-degree angle to the 90-degree normal (angled to the right); and the obstacle is at a negative (-) 45-degree angle to the 90-degree normal (angled to the left). Servo sweeping *will* be used in these experiments. One key we hope to see in these data is a gradual, linear increase/decrease in sonar readings over a given interval for angled obstacles.

Other variables which may affect the results of these experiments or might be interesting to study at a future date include varying lengths of obstacles (i.e., whether they fill up the sonar's field of view or are extremely short); the presence of multiple obstacles in the same field of view; more subtle angle differences than the 0, +45 and -45 degrees used above; and the effect of obstacle material or surface characteristics on sonar readings and the reflection of sound.

## 5 Analysis of Results

To determine the reliability of the wrapper function eventually produced via the data patterns obtained from the experiments, testing experiments will have to be run. In the analysis experiments, similar test cases to the preliminary research experiments will be used; the world map will be carefully documented, and the function's actual sonar readings will be compared to predicted readings based on the model of the sonar's accuracy.

## 6 Final Points

Variability among sonars is a possible confounding factor to the application of these results to future uses of the sonars with our robots. Running these experiments with every sonar sensor available is not feasible, however. At the completion of the experimental analysis, a few simple test cases will be designed that future teams can run easily to determine if their sonar sensors have the same relative behaviors as the one used to generate this suite of data and write the "filter" function. If it is found that future sonars do not reliably reproduce the expected results for these test cases, this entire suite of calibration experiments will unfortunately have to be run for each sonar. We would like to avoid this scenario at all costs. It may be possible for future teams to write some sort of simple "translation" method to analogize their sonar readings with the ones returned by the "filter" function.