EEL 5666 IMDL

Summer 2004

Written Report #2

Sonar Range Finder SRF04

Instructor: Dr Arroyo, Dr Schwartz

Student: Tsang-Wei Huang

Date: 07/06/2004

Introduction

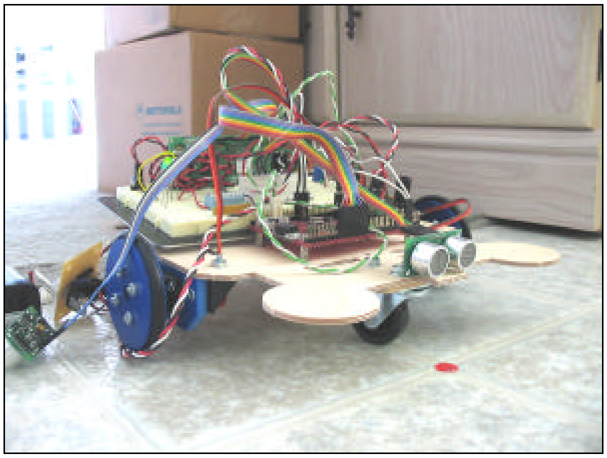
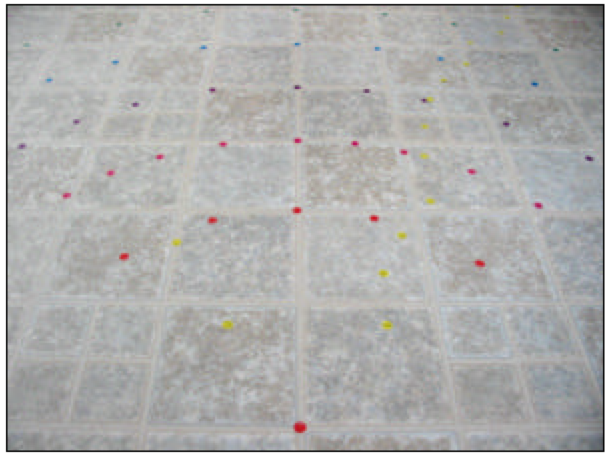
My robot involves measuring the distance between the shooting arm and the hoop. The Devantech SRF04 Ultrasonic Range Finder is best suited for this purpose. It is more accurate than infrared and the range is much longer (approximately 3cm to 3m). Not only does the robot need to know the distance, it also needs to center itself in order to line up with the hoop. By using two such sensors, centering a specific target in an open area can be achieved. This report presents some testing data gathered in a real experiment, compared with the data provided by the manufacturer. It also includes some diagrams showing how the arrangement of the two sensors can affect the reading. The application for this system for my robot will be discussed in the end, and the limitation will also be discussed.

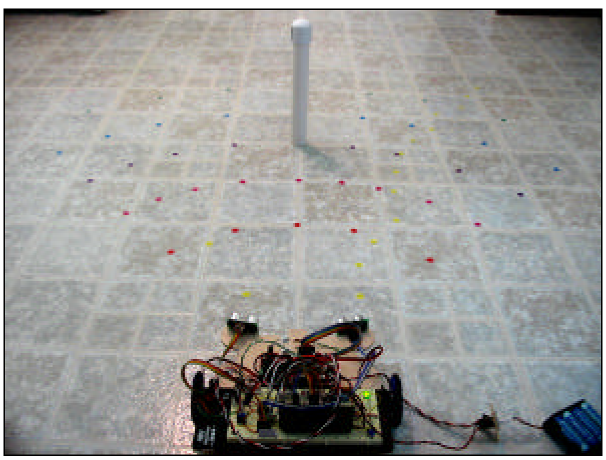
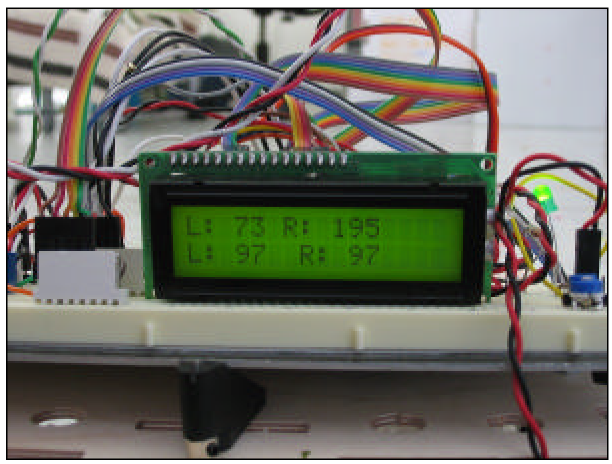
Theory of Operation

The sensors transmits a pulse of ultrasonic waves that travels at the speed of sound, which can be calculated by V = 331.4 + 0.6 Tc m/s, where Tc is the temperature in Celsius. In room temperature, 25 degree Celsius, sound travels at approximately 346.5 meters per second, or 13.64 inches per millisecond, or 1.14 feet per second. The sensors pulses for a brief moment after the transmission to avoid echo. The sound reflects from any object, if there’s any, in its path and is detected by the receiving module in the sensor. By calculating how much time is elapsed between the transmission and the echo, the distance can be determined.

Experiment Setup

The experiment setting is the dining area in my apartment. I used small stickers to mark certain important points. The object to be detected is a PVC pipe with one inch diameter. The following pictures show the experiment in progress.

Experiment Data

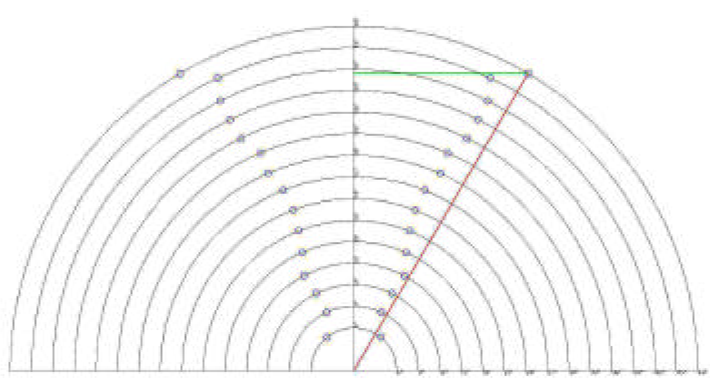
distance	reading	range	angle(rad)	angle(deg)
6	16	3.750	0.675	38.682
9	25	3.825	0.439	25.151
12	33	5.250	0.453	25.944
15	43	7.000	0.486	27.818
18	52	7.250	0.415	23.752
21	60	7.750	0.378	21.657
24	68	8.500	0.362	20.742
27	77	9.875	0.374	21.453
30	85	12.000	0.412	23.578
33	94	13.000	0.405	23.200
36	102	15.750	0.453	25.944
39	109	17.250	0.458	26.251
42	118	18.625	0.459	26.324
45	126	19.000	0.436	24.975
48	137	24.250	0.530	30.345

Distance: how far away the PVC pipe is placed from the origin – the red line in the diagram.

Reading: the reading from the SRF sensor for the specific distance. Details on how the reading is obtained will be presented in later sections.

Range: how far way from the last detectable spot to the center axis – the green line.

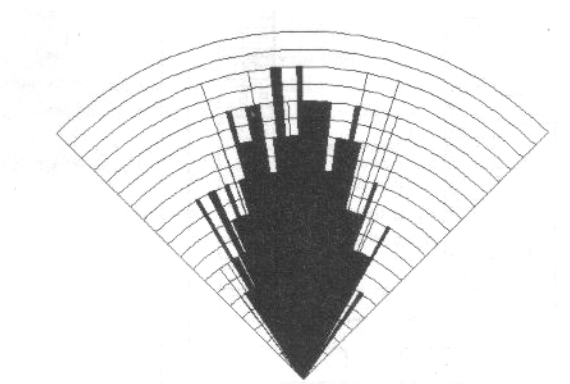
Angle: ASIN (range/distance), the angle specified by the red line and the center axis



In the diagram above, each line represents a 6” increment.

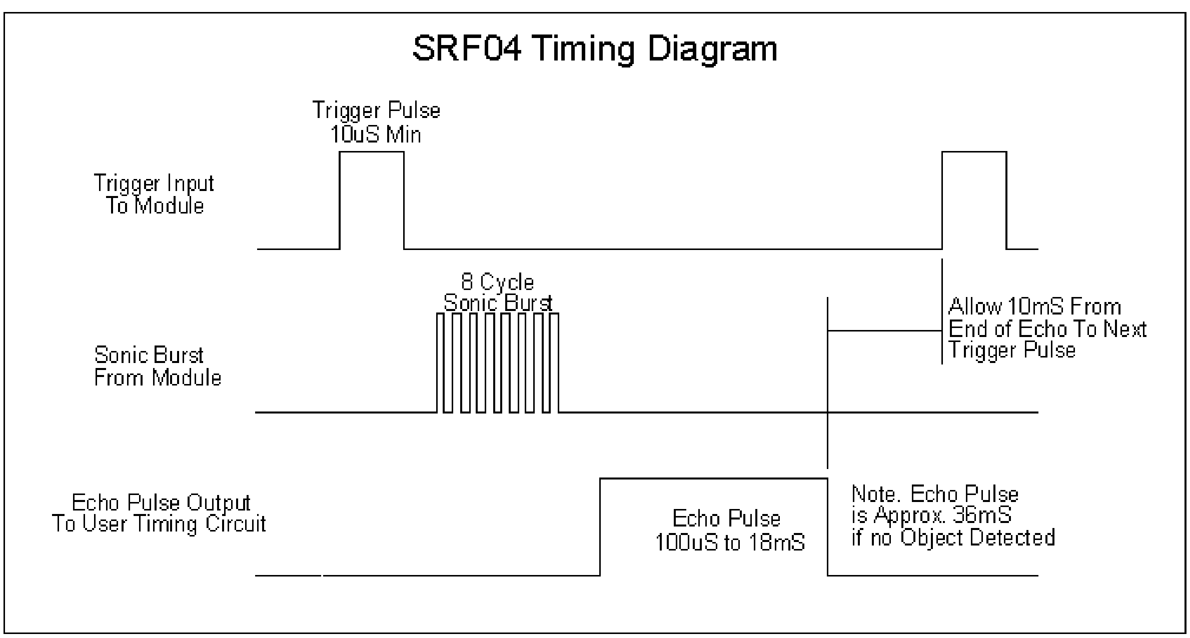
The readings from the table above yield an average of 25.7 degrees wide, cone shaped pattern. The beam pattern specified in the datasheet is 30 degrees, which isn't too far away from the actual readings. One thing that was observed during the process, the furthest spots detectable by the sensor tend to change in each trial, sometimes they can be up to 6 inches apart when the object is placed more than 3 feet away. This really complicated the experiment because I couldn't make up my mind as to where to mark the spot.

The beam pattern obtained by Acroname is the following:



The radial lines indicate 6" distance increments.

Timing Diagram



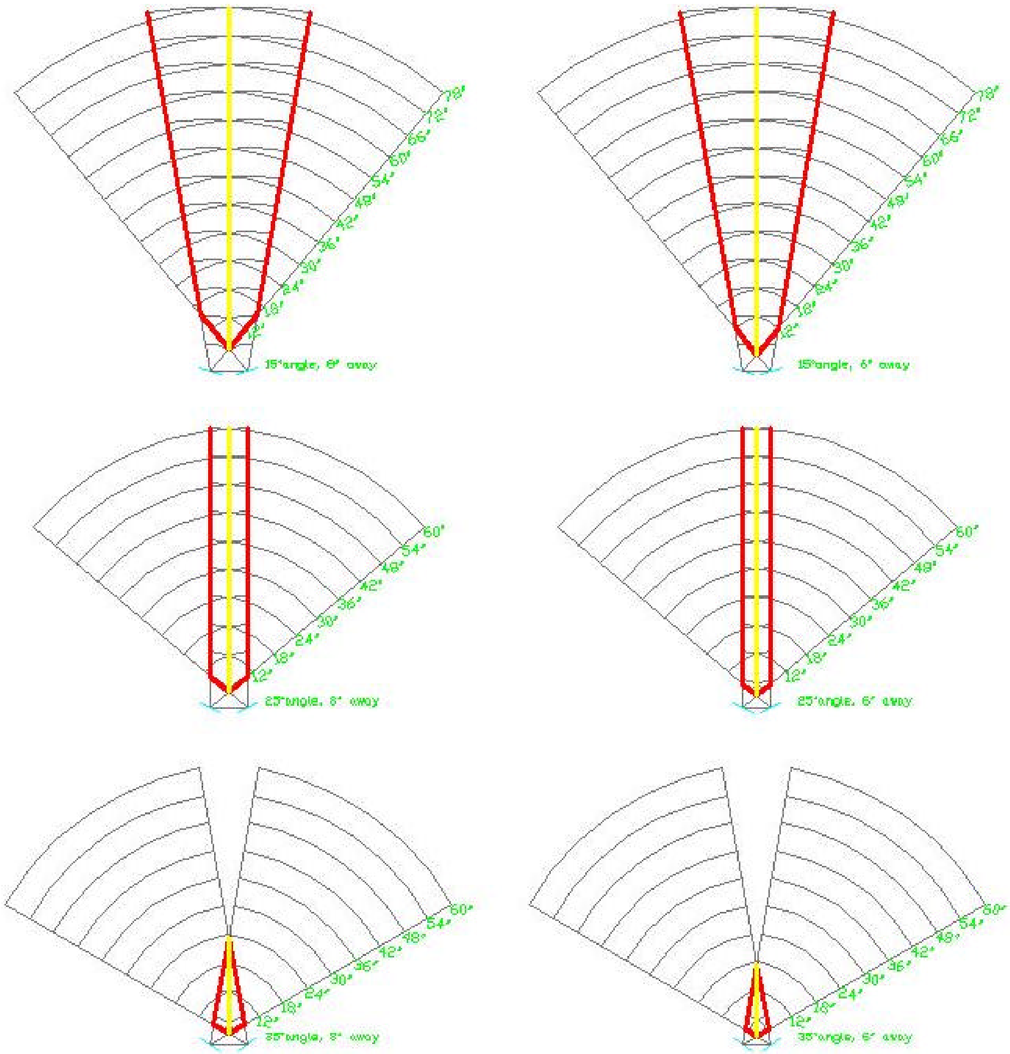
Software Implementation

The approach is suggested by Will. First a timer is setup to trigger an interrupt every 50 us. In the interrupt service routine, a state variable is checked and updated to create a state machine that acts according to the timing diagram above. There are four states:

- SRF_State_init: The initialization state, the sensor is started when the state variable is set to this state.
- SRF_State_wait: After the trigger input is set to high for one cycle (50 us), the system enters the wait state. It stays in this state until the output line is high.
- SRF_State_echo: The state when the echo line is high, every time the system enters the interrupt handler, it checks the echo line, if it's still high, a counter is incremented, once the echo line is low, the reading is completed and the counter contains some value proportional to the distance.
- SRF_State_idle: The state when the sensor is inactive.

SRF Sensor Application

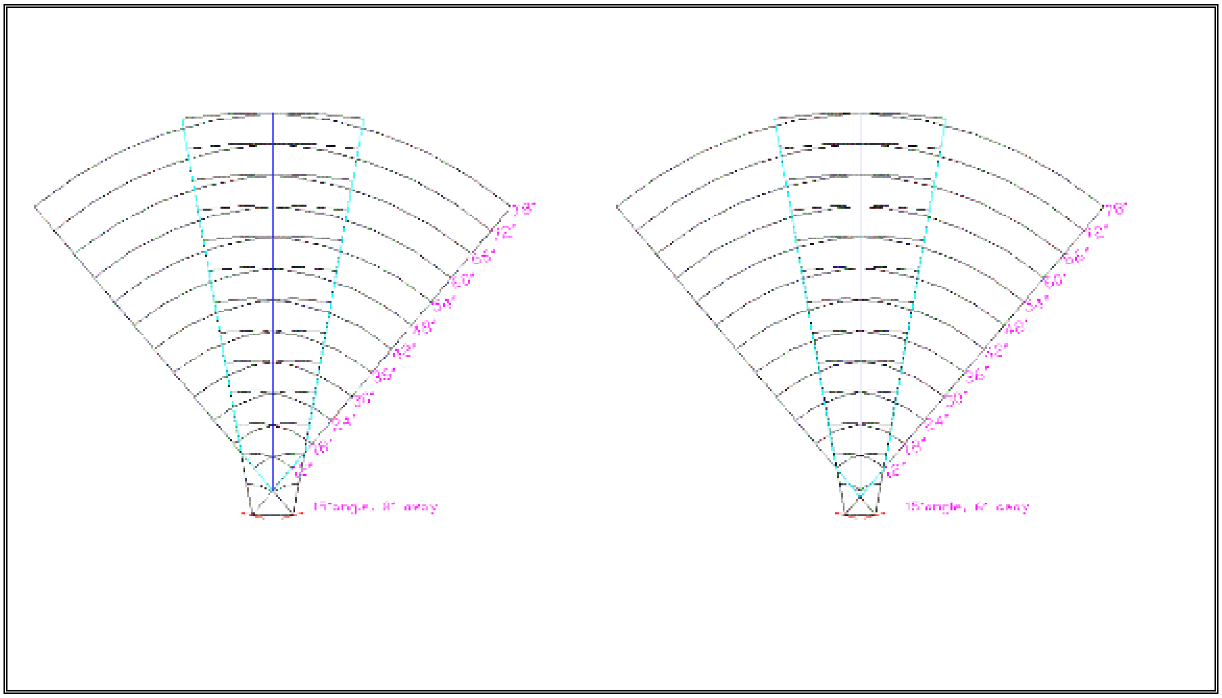
To make use of the sensor, I place two sensors 6 inches apart from each other, and turn them toward each other, creating a cross-eyed vision. The sensors are triggered one after another, with at least 10 milliseconds in between to avoid picking up previous transmissions. Beam patterns overlap and if both sensors report a pair of matching values, the object can be assumed to have the same distance away from both sensors, which indicates it's at the center of the robot. A more detailed diagram is presented below:



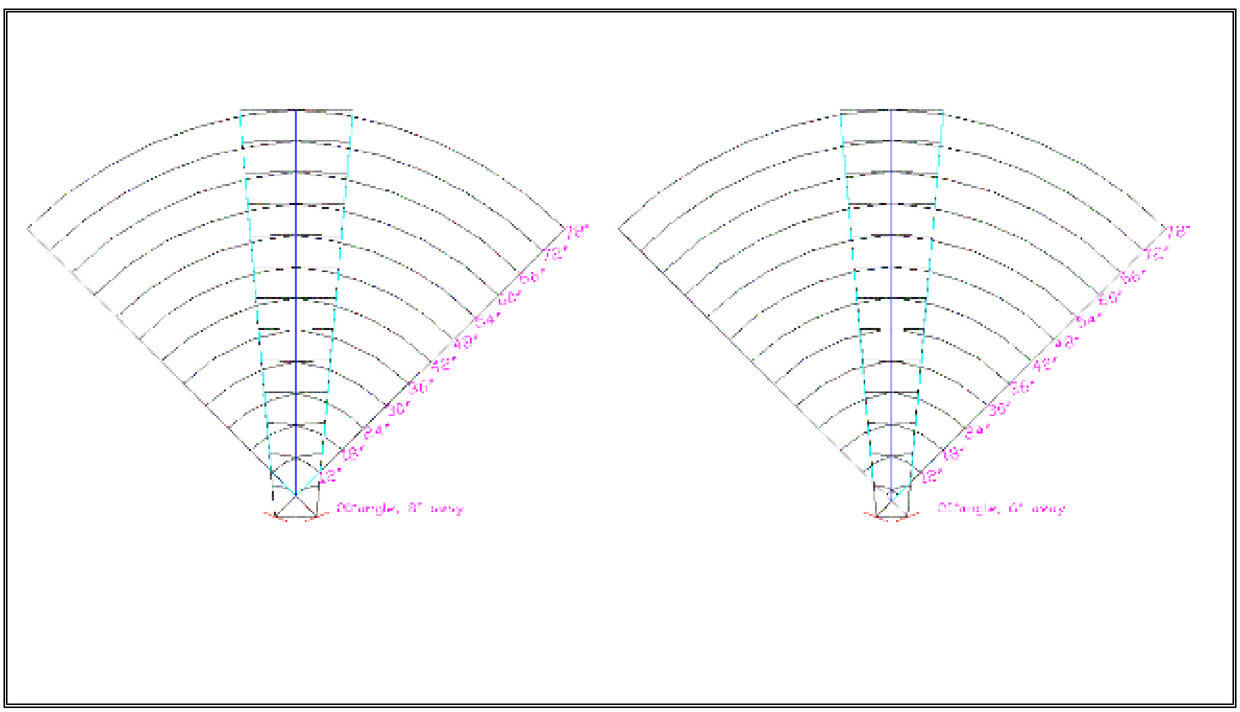
The redial lines indicate the range for each sensor, and the red lines are the boundaries of the overlapped regions. The following diagrams will show a total of 8 different arrangements of the sensors.

For the following diagrams the ones on the left show two sensors 8” apart, the ones on the right 6” apart.

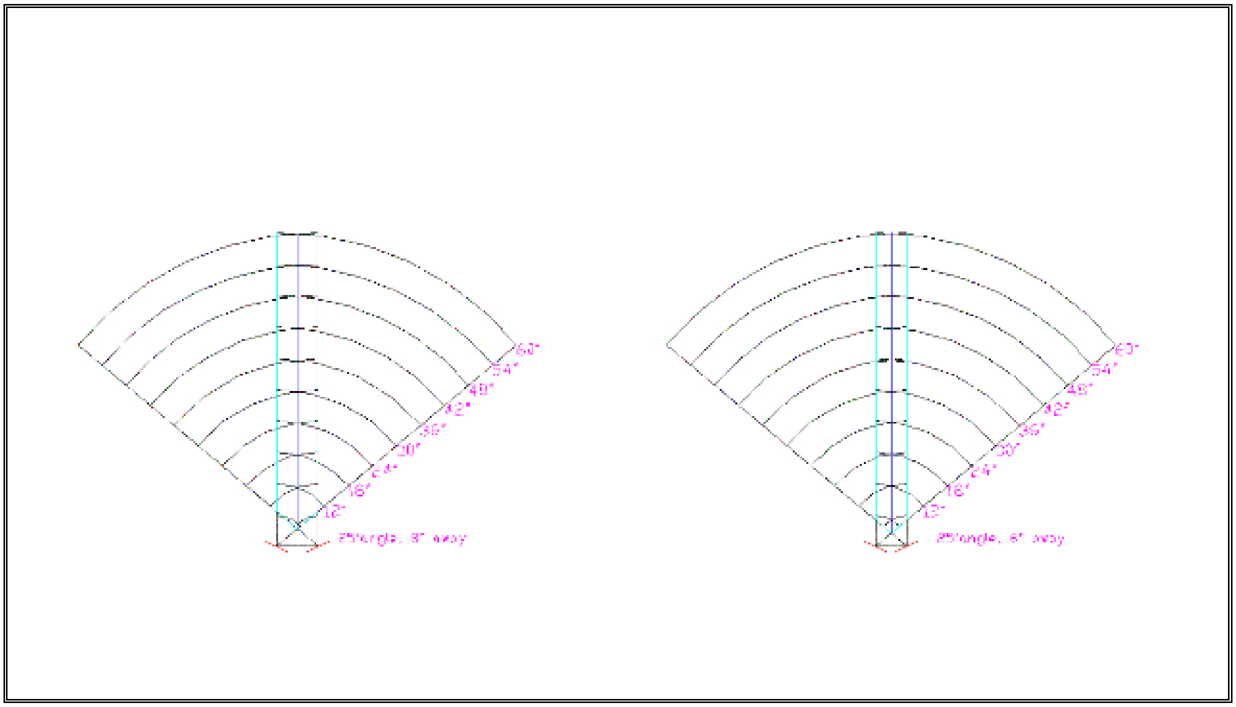
Sensors are turned 15 degrees towards the center.



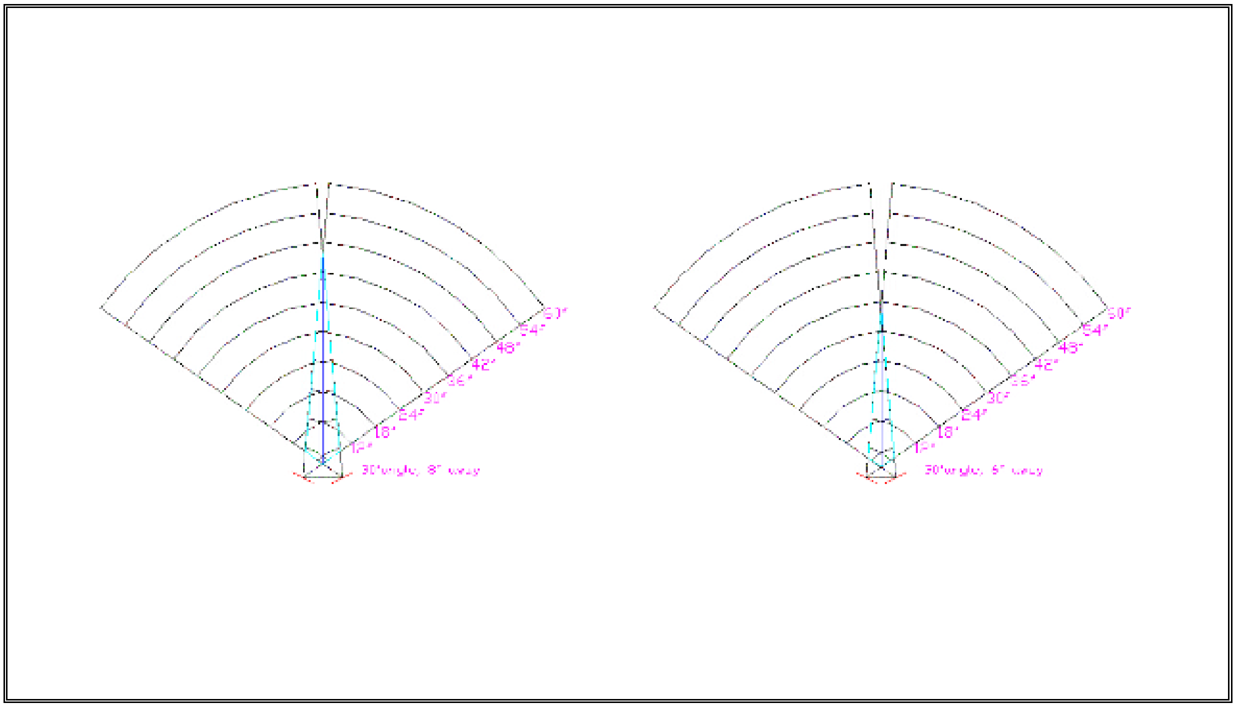
Sensors are turned 20 degrees towards the center.



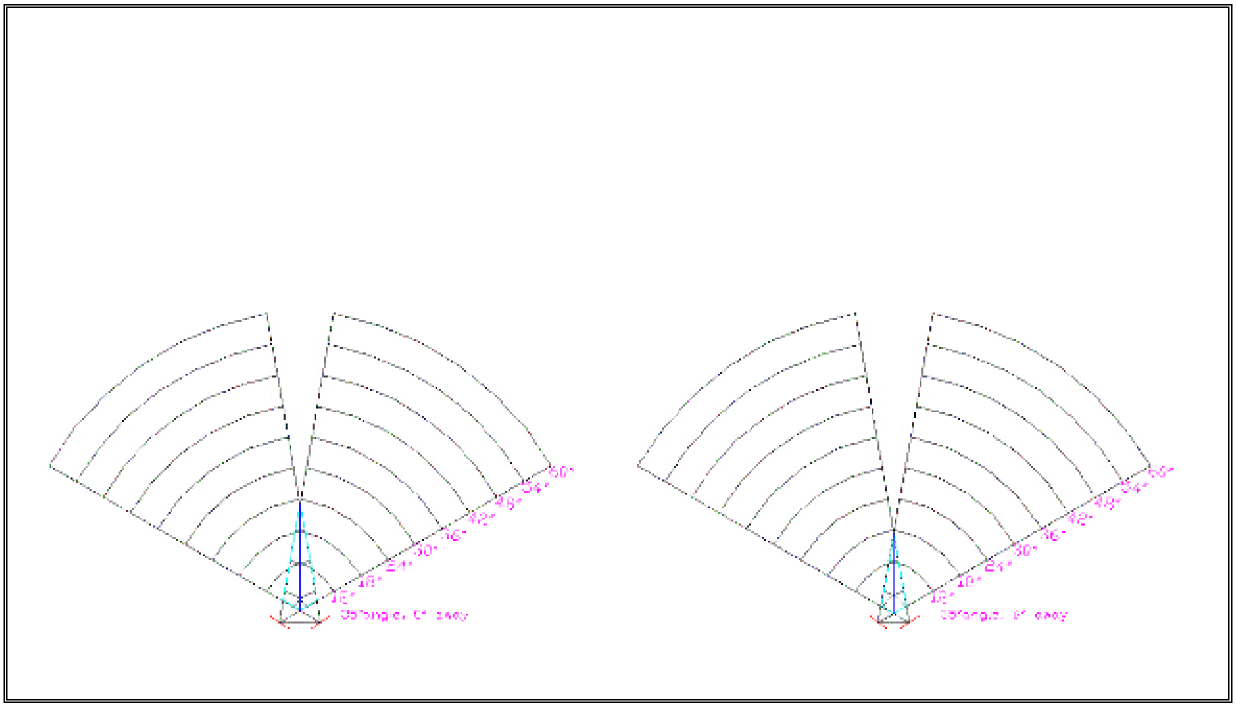
Sensors are turned 25 degrees towards the center.



Sensors are turned 30 degrees towards the center.



Sensors are turned 35 degrees towards the center.



Conclusion

When the two sensors are turned at a small angle (15 to 20 degrees), the beams overlap and the region extends to the maximum range. As the angle gets bigger to about 25 degrees, the inner boundaries became almost parallel. This is, in theory, the optimal arrangement since the error margin can be maintained at a fixed value, regardless of the distance. When the angle gets above 30 degrees, the boundaries intersect and create a zone in which objects can be detected by either sensor. Beyond this zone, neither sensor can detect any objects, even it is directly in front of the robot. This kind of arrangement isn't completely bad, because it can be used to do obstacle avoidance, since it has a wider view than the previous arrangements and can "see" more on the side of the robot. A better approach is to make the angles flexible and controllable by software, so the angle can be adjusted for different purposes.