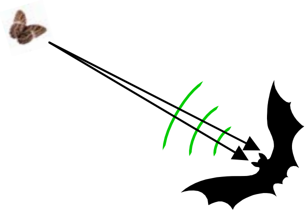
Multi-echo Integration for Sonar-based Vehicle Navigation

Gaurav Singal, Marshal Miller, Matthew Cheely, Timothy Horiuchi

MERIT 2004

CSSL

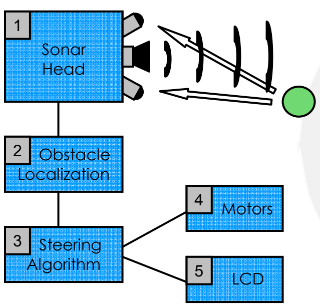
Introduction



Bats use echolocation to navigate dark, complicated environments in search of food. By emitting high frequency sound waves and analyzing the echoes in both ears, they can accurately localize obstacles and targets.

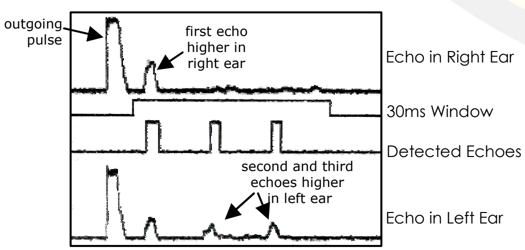
Our goal was to create a robotic vehicle which could use echolocation to navigate through obstacles. Such technology can be directly applied to unmanned vehicles. Unmanned aerial vehicles, for example, cannot presently navigate through forests due to the limited capabilities of current sensors, including vision and infrared.

Overview



1. The sonar head emits a 40kHz sound pulse, which travels to nearby objects and reflects back to the microphones.
2. The echo envelopes are sent to an obstacle localization algorithm, which determines the angle and distance of the object.
3. This positional information is sent to a steering algorithm which finds the clearest path.
4. Motor commands are sent to the robot, which steers toward the chosen path.
5. Display information is output to the LCD.

Echoes/ILD

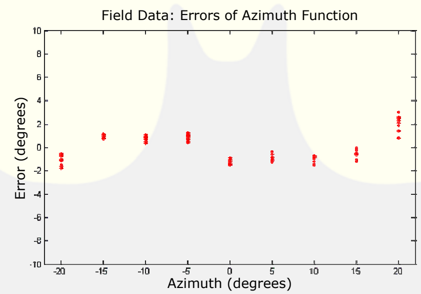


The sonar head "listens" for 30 ms after emitting a pulse. Echoes are detected using de-bounced level thresholds. The location of echo source can be determined by using information encoded in the echoes. Distance is computed by measuring the time-of-flight of the sound. Direction is determined by analyzing the Interaural Level Difference (ILD), the difference in echo strengths in each ear. For example, an object on the right will register as a stronger echo in the right ear. With bats, this effect is caused by the bat's head, which produces a "shadowing" effect. In our setup this property results from the two microphones being angled outward.

Obstacle Localization

We used echo information to determine the azimuth (horizontal angle) of and range to the target. We positioned an object at 60 different known positions and recorded the echoes returned to each ear. This data was used to perform regression analysis to find a function to compute azimuth using the integrals of the echoes in each ear. We found that a simple function using the log of the ratio-of-integrals provided accurate results.

$$azimuth = \alpha \ln\left(\frac{\int LeftEcho}{\int RightEcho}\right) + \beta$$



Steering Algorithm

The positions of the obstacles are sent to an algorithm called "Openspace," which finds an open direction for the robot to move toward:

1. The algorithm starts with a set of bins, each representing a potential direction. Each direction is initialized as "open," with preference given to the current heading (Fig 1.1).
2. Detected obstacles cause a decrease in the "openness" of neighboring directions (Fig 1.2).
3. A triangle centered at each obstacle is subtracted from the a priori distribution. The properties of the triangle depend on the distance to the obstacle – closer obstacles cause a deeper and wider triangle to be subtracted (Fig 1.3).

Once all obstacles are taken into account, the robot is told to move in the most open direction, i.e. the bin with the highest value.

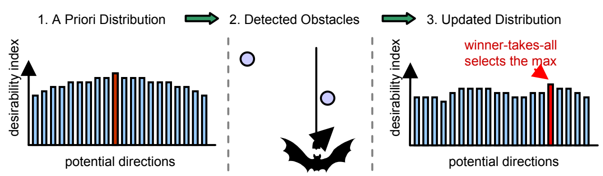
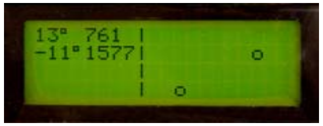


Fig 1.1 Fig 1.2 Fig 1.3

Implementation and Output

All of the processing is done on two PIC microcontrollers. One PIC is responsible for converting ILD information from echoes into obstacle locations, while the other executes the Openspace algorithm. Obstacle information is passed from the azimuth PIC to the Openspace PIC via serial communication. The use of two PICs and parallel processing allows for real-time control of the robot.



LCD Showing Detected Obstacles: Left side shows Azimuth(degrees), range(mm) Right side shows internal world representation

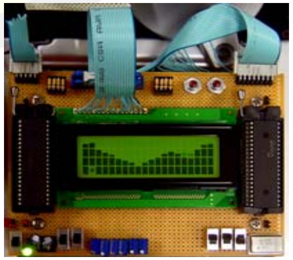
LCD Showing Openspace Distribution: Each column is an Openspace bin with height proportional to desirability of that direction

Motor Control

The Openspace algorithm outputs differential drive commands. Two 6-bit digital commands are sent in parallel to the robot, setting the speed of each wheel. The wheel ratios are configured to drive the robot at a constant tangential velocity. Since Openspace simply outputs motor commands, the system can be adapted to any vehicle.

Results

The robot we built is capable of locating obstacles to within two degrees of accuracy at a distance of four meters. Moreover, the robot is capable of detecting numerous objects in its path with a single ping. The Openspace algorithm allows the robot to not only react to immediate obstacles but also account for more distant threats. The robot successfully navigated through a field of poles in repeated trials. The robot was also able to navigate through more realistic terrain, avoiding trees on a mulch bed.



Acknowledgements

This work was supported by the University of Maryland MERIT 2004 Program and by a grant from the Air Force Office of Scientific Research (FA95500410188).