Experiment #10 Target Acquisition and Approach

Students should now have a basic library of motion functions and initialization for global variables. We are now prepared to perform actual machine intelligence programming.

The first applicable task is to acquire the angular and distance position of a nearby target, turn to face the target, accurately measure the distance to the target and move forward to a position 2 inches in front of the target. The process requires a calibrated mid-range IR sensor, a varsIntTurn function, analogRead function, and decision making supporting software written by the user.

A priori assumptions; the robot is within 18 inches of a target, the target is within a known angular scanning range, the target is compliant with size and shape specifications, and the sensor and robot are calibrated. All assumptions can be made true by proper preparation and maneuvering prior to scanning

Stage 1 – Scanning for target

The first task is to rotate the robot and simultaneously read and process the IR sensor data in real time. This can be done with the following programming structure.

varsIntTurn(rotl, 90, 500, 1,0, rightmotors, leftmotors):

```
while(steps > 0) {
        // read sensor
        // process data
        //  store results
                }
```

The read sensor is the standard analogRead(A0); command. The process data consists of testing whether the data is above or below the 18 inch threshold from the sensor calibration. If the data is above the threshold and the prior reading was below the threshold, you have detected a target edge. You must record the angular position by reading the steps variable. The second edge can similarly be detected as a reading below threshold following a reading above threshold. The width of the target is the difference between the steps readings from the leading edge to the trailing edge. The angular position of the center of the target relative to the ending position of the turn is the average position of the two edges. All intermediate and final calculations need to be stored in unique variables.

Stage 2 – Turning to target

Using the result from Stage 1, calculate the turn angle and execute a rotate right turn. Remember that the angle data from Stage 1 is in steps format and the calling variable for the turn functions is calibrated in degrees. Convert from steps to degrees by dividing by steps_per_degree.

Stage 3 – Hard distance fix

Call the delay(100) function to allow the sensor voltage to stabilize. Read and average 500 samples of the sensor. Apply the calibration equation to convert the average sensor reading to true distance.

Stage 4 – Move to target

Subtract 2 inches from the hard fix and move to the target by the resulting distance using the mov function.

Assignment

Write a standalone function to perform all 4 stages of the process. Be sure to declare all variables. Use float variables for all distance and angular calculations. Compile and debug. Test with professor.

Variables

1. long firstEdge, secondEdge, width, center
2. float centerAngle, distToTarget;
3. constant float calScale, calPower;
4. long sum = 0;