

UNIVERSITY OF NORTH CAROLINA AT CHARLOTTE

Department of Electrical and Computer Engineering ECGR 4161/5196 Introduction to Robotics

Experiment No. 4 – Edge Detection and Obstacle Avoidance

Overview: The purpose of this experiment is to introduce the student to hierarchy within LabVIEW by programming edge detection and still allowing concurrent obstacle avoidance behavior.

Infra-Red Sensor:



Figure 2.1: Sharp IR Sensor [1]

To accomplish the goal of edge detection a sharp Infra-Red (IR) sensor will be used. Each daNI robotic platform has been equipped with 1 IR sensor. This sensor may be located underneath the front of the robot. “These sensors use triangulation and a small linear CCD array to compute the distance and/or presence of objects in the field of view.”[1] “The basic idea is this: a pulse of IR light is emitted by the emitter.”[1] “This light travels out in the field of view and either hits an object or just keeps on going.”[1] “In the case of no object, the light is never reflected and the reading shows no object.”[1] “If the light reflects off an object, it returns to the detector and creates a triangle between the point of reflection, the emitter, and the detector.”[1]

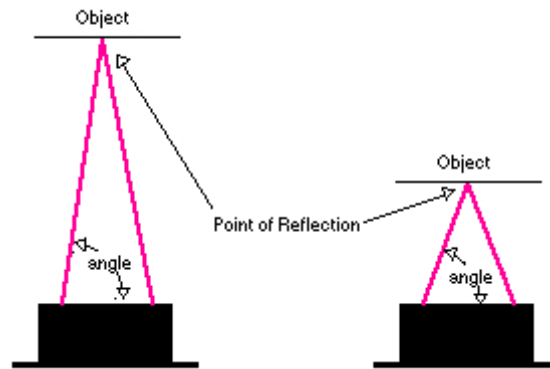


Figure 2.2: IR Sensor Angle of Reflection [1]

“The angles in this triangle vary based on the distance to the object.”[1] “The receiver portion of these new detectors is actually a precision lens that transmits the reflected light onto various portions of the enclosed linear CCD array based on the angle of the triangle described above.”[1] “The CCD array can then determine what angle the reflected light came back at and therefore, it can calculate the distance to the object.”[1] Depending on the distance of the object being reflected off of, the sensor will return a different raw voltage value. The graph in figure 2.3 illustrates this voltage curve. Based on the distance between the IR sensors mounted on daNI, and the surface daNI is placed on, the IR sensor will output a specific voltage. This voltage will be approximately equal to 1.2v for daNI’s current setup. When daNI approaches an edge, the IR sensors voltage will change. If the voltage changes to a value lower than approximately 1.2v the robot should “know” an edge is present and change its course.

Figure 2.3: Analog Output vs. Distance to Reflective Object [2]

Requirements:

- Req. 1 The obstacle avoidance program from lab 1 will be used.
- Req. 2 Two IR sensors will be used.
- Req. 3 A LabVIEW program will be written that will use the IR sensors to detect when the robot is about to fall off of an edge, and reverse and change direction of the robot.
- Req. 4 Robot will operate autonomously.
- Req. 5 Requirements 1 and 3 will be implemented concurrently. (Robot should adopt obstacle avoidance behavior, and only use edge detection code when the IR sensor detects an edge. After robot has reversed and changed direction, robot should continue with obstacle avoidance behavior.)
- Req. 6 Robot should continue running obstacle avoidance behavior after it has detected an edge and changed course.

Connecting a new sensor to the robot:

- Step 1. For this sensor, we want to read the raw analog voltage and compare it to the graph on Figure 2.3. In order to accomplish this, you will want to connect the Vcc pin of the sensor to pin j2 on the FPGA board.
- Step 2. The Vout and ground pin can be connected to one of the digital ground pins on the FPGA board and one of the digital signal lines, or ports as they are seen on LabVIEW.
- Step 3. To obtain the analog voltage in LabVIEW, you must look at port AI0.

Detailed Steps:

- Step 1. Create a new FPGA file and copy all of the program details from the previous FPGA file to this new one.
- Step 2. Add code to read directly from the sensor. Drag the port AI0 and the other analog port from the project tree into the block diagram of the FPGA file.
- Step 3. Simply add an indicator to read from the ports and enclose them separately within their own while loops like how the rest of the code is.
- Step 4. Create a new VI and name it "RoamingWithEdgeDetection." Also open Roaming.vi (we will be copying elements of this code).
- Step 5. Compare the value obtained from the infrared sensors to a value that you wish to use to detect an edge seen from either sensor.
- Step 6. Add in a case for when an edge is detected and when an edge is not detected. Generally when an edge is not detected, the robot will execute the roaming VI.
- Step 7. Copy the coding element "sensor motor angle / Sensor distance" from the roaming.vi code. Paste this element in your RoamingWithEdgeDetection.vi.

The diagram illustrates the navigation system's architecture, divided into three main functional areas:

- Sensing:** This section involves taking distance measurements from an ultrasonic sensor. It includes steps like calculating the next scan angle for the sensor, applying the scan angle to the FPGA, reading the latest sensor distance measurement, and inserting the latest distance measurement and scan angle into the obstacle vectors data.
- Calculation:** This area handles the processing of sensor data. It includes a block labeled "sensor motor angle (rad)" and "sensor distance (m)". A large red arrow points from the "sensor distance (m)" output to the "motor communication" block. Other blocks include "obstacle vectors", "steering frame", and "Frame".
- Actuation:** This section focuses on controlling the motor. It includes steps like retrieving the motor communication setpoint, indexing the motor left and right motor, and applying the motor control.

The diagram also shows various input/output ports and signal flows, such as "motor offset (rad)", "p motors (F)", "Velocities", and "Frame". The overall flow is from Sensing to Calculation to Actuation.

Exercise extreme caution when testing this lab. Even if your code is perfect, sensors are not. We would like to avoid daNI from taking a trip down the stairs.

- [1] <http://www.acroname.com/robotics/info/articles/sharp/sharp.html>
- [2] “Acroname easier robotics: Sharp IR Rangers”, Acroname Inc. 2009

- [1] <http://www.acroname.com/robotics/info/articles/sharp/sharp.html>
- [2] “Acroname easier robotics: Sharp IR Rangers”, Acroname Inc. 2009