Sensor Report

University of Florida
Department of Electrical and Computer Engineering
EEL5666
Intelligent Machines Design Laboratory

James Larson
10-31-02

Table of Contents

Overview

My robot, Broken Arrow, will be able to find a billiard ball and shoot it into a pocket. It will use IR detectors, bump switches, and a CMUcam to navigate the world, find the ball, and locate the pocket.

Sharp GP2D120 IR detectors have been chosen as the IR range-finders used in general navigation, an IR emitter/detector pair will function as a break beam, and bump switches will be used in case the GP2D120 detectors fail to find the wall.

IR Range detectors

For IR range sensors, the GP2D120 IR detectors made by Sharp were chosen. These sensors are very easy to interface, as they are emitters and detectors in one, with a 3-pin Vcc/Gnd/Vo connection. They output an analog signal (Vo), so a more precise distance can be determined than by a digital sensor. The output of these IR sensors are sent to the A-to-D port of the ATMega323 processor via PA0..1.

This sensor was chosen over the Sharp can IR sensors for several reasons. These do not need to be hacked in any way to get them to work. As well, the output voltages range from 0 to 5 volts, much larger than the limited range of the hacked Sharp cans. The drawback to using these is the power consumption – when not needed, the sharp cans could easily be turned off, but the GP2D120's require extra hardware.

These two sensors are facing outward diagonally. This orientation was chosen so that, if needed, a third IR sensor could be added in between these two facing directly forward without interfering signals. Only after extensive experimentation with the pool ball and my robot will I be able to determine if the third IR detector will be needed.

Bump Sensors

I have chosen to use the small push buttons as provided by the IMDL lab for bump sensors. These are configured in a network of three areas, each with two to four switches.

Each area of my robot (front-left, front-right, and back) has a network of parallel bump switches. Having them in parallel allows for more accuracy in the hits. I found that when only three bump sensors were used, there were many areas in which, upon hitting, would not trigger the bump detection. I added more bump sensors to each area in parallel so that, if any one of these switches were thrown, the program would interpret it as part of its region. Although this created a less accurate idea of where the hit took place, this kind of accuracy was not needed in my robot.

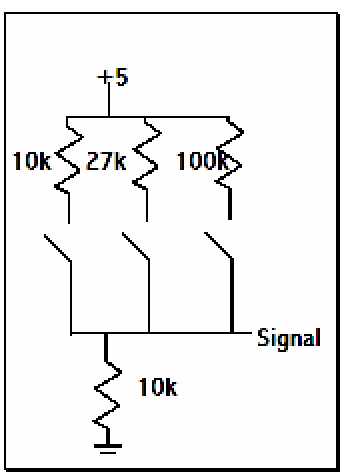

Figure 1

The three regions wires are sent through a voltage divider network as given in class and shown in figure 1. This allows the outputs from the bump switches to be sent to a single analog port instead of several digital ports.

IR Break Beam

An infrared break beam is required to determine if the ball has come close enough to strike. To construct this, I have decided to use a simple IR emitter / phototransistor pair. A circuit diagram is shown in figure 2. This is based off of a circuit created for Cyrus Harrison's summer 2002 IMDL robot.

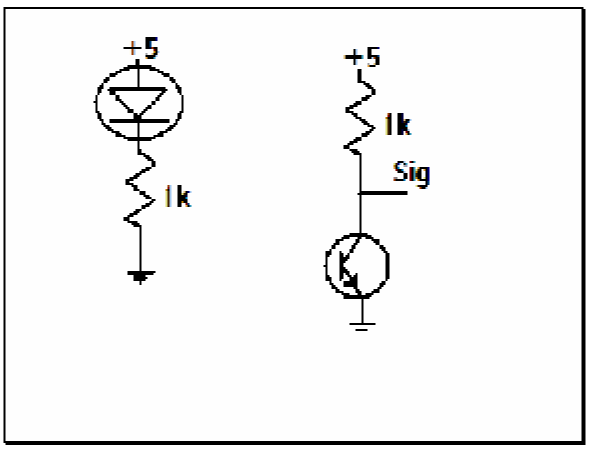

Figure 2

I decided to use shrink tubing to collimate both the IR detector and emitter. With two other IR devices in use on that part of the robot, both needed to be shielded in order to minimize interference.

CMUcam

A CMUcam will be used as the special sensor. Its purpose on the robot is to find the cue ball and the object used as the beacon, which will be placed above where the pocket would be. This has various color manipulation and tracking features that will be used.

The main feature of this camera is its color-tracking feature. Using serial communications, a microcontroller can send the camera various commands to track the centroid of a color region. By tracking the beacon, my robot will be able to find where it needs to shoot and line itself up.

Experimental Data

IR Range detectors

The IR range detectors have a range as specified by the manufacturer to be 4-80 cm. In experiments, the IR sensors were placed at various distances from the wall and the analog values were obtained via the onboard LED's. A graph of this data is shown in figure 3.

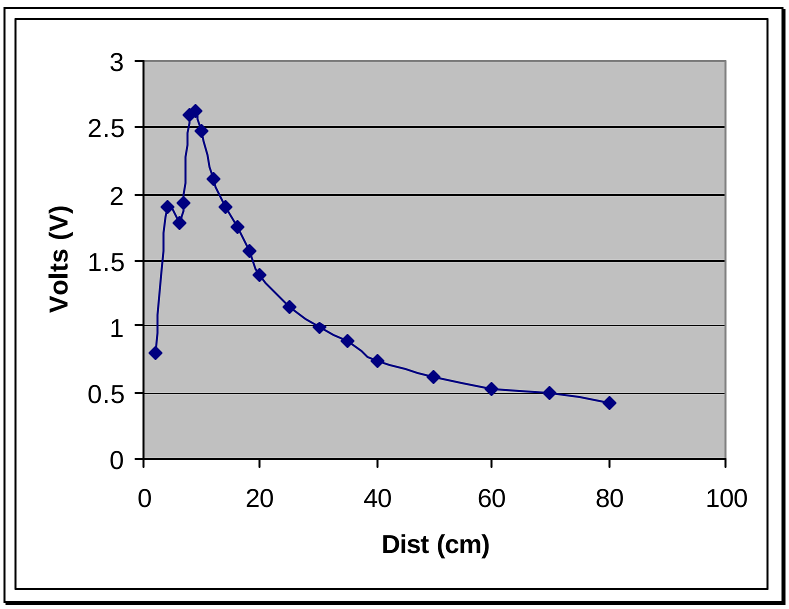

Figure 3. Output values for IR detector at various distances

These values were somewhat surprising. The data sheet says that the IR will work up to 4 cm, but the data is only valid in a useable sense at distances greater than 8 cm. At that point, the IR begins to work in a predictable way (i.e. a lower voltage means a greater distance, etc).

Upon experimenting with various values for a "too-close" threshold, I found that any values greater than 225 (approx. 4.41 V) signify that an obstacle is about to be struck.

This may be lowered in the future, but for my basic obstacle avoidance code, this is the threshold used and it works well.

Bump Switches

In order to find the analog values for each of the bump networks, I decided to find by trial-and-error. I connected one of the bump sensors while leaving the other switches open. I had the output of the LED's show if the switch was connected by lighting if the button is pressed, unlit if it's open. By trying different values on all three bump networks, I found that minimum values of 30, 60, and 120 for each respective bump sensor gave the desired output.

IR break beam

After connecting the IR detector and emitter to each respective circuit, I collimated each and measured voltages using a multimeter. The results were somewhat surprising. The voltage when broken was 5V, but when open, the voltage only dropped to about 4.8V. I coded the analog-to-digital routine and used the LED's to display whether or not the beam was broken and simply tested the analog voltage for 5V (digital 255).

This worked very effectively, but during split seconds, even when the break beam was not closed, the value would equal 255. I will therefore need to count consecutive analog values and, if this value is 5V for a certain number of consecutive conversions, the beam must be broken. The LED's only light for a split second (not even enough time to fully light the LED's), so I would imagine the number of consecutive values will be minimal.

CMUcam

In experiments, I discovered that the CMUcam is not quite as easy to work with that the manufacturers stated. While very few commands are needed, the camera is very weak and is susceptible to IR interference.

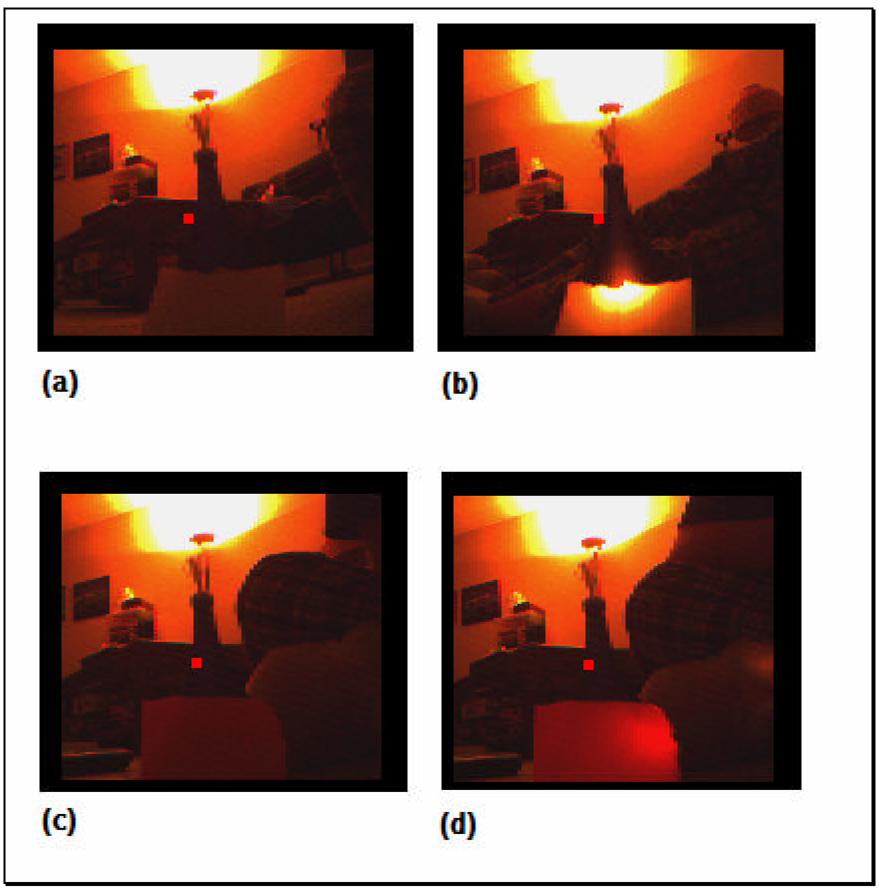

Figure 4

The main drawback I have seen is the tendency of the pictures to have a red tint to them. Figure 4 shows a white piece of paper (Fig 4a, 4b) and a red piece of paper (Fig 4c, 4d) at the bottom of the image. In each, one is lit by a small flashlight, one is left in standard room lighting. As shown, unlit, the red and white papers look very similar to the camera, but lit, they become distinct.

Using the included software, I have been able to track objects, but the tracking is much more accurate while lit. I have tried tracking several different colors, but none have really stuck out as being easier or more difficult to track – sufficient lighting is the key to good image tracking.