A Minimal Infrared Obstacle Detection Scheme [*]

C. Ronald Kube
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
kube@cs.ualberta.ca

October 23, 1998

Introduction

Obstacle detection is often the first perceptual system we add to our mobile robots. After all, once the robot is moving around, constantly bumping into things doesn't make a very impressive demonstration of the robot's intellect [1]. A simple solution is a contact sensor in the form of a bump switch or whisker, but this approach to obstacle detection is not very impressive and reminiscent of a blindfolded C3PO of Star Wars fame. An alternate solution is near-proximity sensing, possible with an infrared emitter/detector pair. These optoelectronic infrared components are inexpensive, easy to use and widely available from several manufacturers.

In this article we'll look at two such devices, the Siemens SFH484 infrared emitter and its matching detector, the BP103B infrared phototransistor. To demonstrate its use, we'll build a simple minimal obstacle detection system using just two sensors and show how to gather and plot data from them. Then we'll configure the two sensors for a simple obstacle avoidance system and implement it on a reactive robot, reminiscent of Braitenberg's Vehicles [1], by connecting the sensor's output to a 68HC11. Finally, we'll show how this simple strategy has been used to create obstacle avoidance behaviors in our system of 10 mobile robots, and we'll discuss some possible enhancements and alternate uses for the sensors.

Infrared Emitters and Detectors

Infrared radiation is a radiant energy longer than visible red with wavelengths between 770 and 1500 nanometers. Two optoelectronic devices that make use of this energy are the Infrared-Emitting Diode (IRED), and the Infrared Phototransistor. Two types of IRED radiant-energy sources are Gallium Arsenide (GaAs) and Gallium Aluminum Arsenide (GaAlAs), which emit in the 940 nm and 820 nm portion of the near-infrared spectrum respectively. Infrared phototransistors are simply transistors designed to be responsive to this radiant energy and are typically silicon bipolar NPN types without a base terminal. Figure 1 shows the relative spectral characteristics of the human eye, a Silicon Phototransistor, two types of Infrared Light Emitting Diodes, and a Tungsten light source.

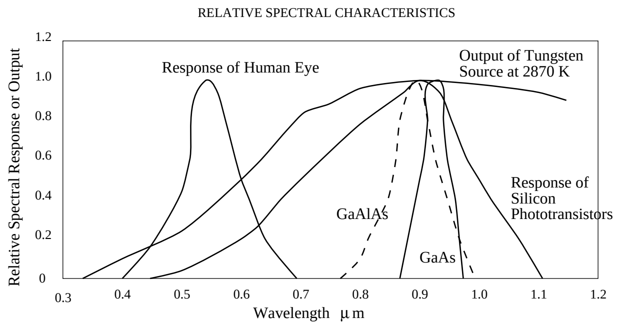

Figure 1: The relative spectral characteristics of the human eye, a tungsten light source and a silicon phototransitor (adapted from [3]).

To help you make sense of the data sheets that describe these components let's consider a few important terms:

Acceptance Angle is measured from the optical

---

[*] A previous version of this article appeared in The Robotics Practitioner, spring 1996.

[1] If you're trying to generates laughs for the observers (or tears for the creators) then let your robot roam without obstacle avoidance.

axis and is the maximum angle at which a phototransistor will detect a travelling infrared ray.

Beam Angle is the total angle between the half intensity points of an IRED's emitted energy.

Radiant Flux or Power output usually quoted for IREDs is the time rate of flow of radiant energy.

Photocurrent is the current that flows when a photosensitive device is exposed to radiant energy.

IRED emitter's typically come as GaAs, originally developed in the 1960's, or GaAlAs. Comparing Gallium Aluminum Arsenide with the older Gallium Arsenide we note that GaAlAs have [2]:

- 1.5 to 2 times the power output than GaAs for the same current;
- Peak wavelength of 880nm versus 940nm;
- Forward voltage (V_f) is slightly higher;
- Fewer failures, with failure defined as a 50 percent drop in power output.

The Siemens SFH484 is a very high power GaAlAs infrared emitter, with a power output of 20 mW at a forward current of 100 mA, and a beam angle of 16 degrees. The matching phototransistor is the BP103B-4 with a photocurrent $\geq$ 6.3 mA and an acceptance angle of 50 degrees.

For our purpose, we will configure the emitter/detector pair to measure the amount of reflected IR energy from obstacles in front of the robot. And we would like to correlate the reflected reading to a distance from the sensor to the obstacle. This will depend, among other things, on the emitter's radiant flux, the distance to the object, the nature of the object's surface reflectance, and the light current of the detector.

Implementation and Testing

The circuit shown in Figure 2 will allow you to connect up to four IR emitter/detector pairs. Table 1 lists the components used to construct the circuit. Header $J1$ is connected to an output pin on the 6811 microcontroller (Port B pin 0 for example) and will be used to turn the IREDs on and off; the extra pin is used to daisy-chain the enable signal to a second board. Header $J2$ is connected to the A/D input port (Port E) on the 6811. The mechanical layout shown for the circuit makes use of male strip headers to connect the IR pairs to $J3$ shown in Figure 3. Remember to install jumpers in all unused LED positions. Power and ground is brought in on $J4$ and a shunt is normally installed in $JP1$ and removed when adding an ammeter to adjust the IREDs forward current.

The emitter/detector pair can be mounted using $T1-3/4$ LED holders in 1/4 inch holes spaced 1/2 inch apart on a small piece of angle aluminum. Figure 3 shows three techniques for mounting IR pairs. In the bottom photo wires are soldered directly to the component with hot-glue added for mechanical strength. The middle photo shows the IR pair mounted to a small circuit board with a four position header. The top photo uses a plug/socket combination to connect to the component. Both the IRED and phototransistor are soldered to a two position machine pin plug, with the matching socket soldered to the data cable used to bring the signal to the control board. Each method allows the sensor to be easily positioned anywhere on the robot.

To test the circuit connect one sensor pair to LED1/T1 watching for correct polarity. Remember to install jumpers in the unused IRED position (LED2). Connect an ammeter between $JP1$ pins 1 and 2, add power and enable $FET1$ by connecting $J1$ to +5. Adjust $R5$ to read 90 - 100 mA. Turn the power off and remove the ammeter, reinstalling the jumper in $JP1$. Connect an ammeter between $JP2$ pins 1 and 2, add power and enable $FET1$ by connecting $J1$ to +5. Adjust $R7$ to read 90 - 100 mA.

Measure the output voltage on pin 1 of $J2$; with the power on, enable set to +5 and nothing in front of the sensor, you should read a low voltage (around 0.5 VDC) in a dimly lit room. As you move your hand in front of the sensor the voltage should rise and continue to rise as you approach, reading about 4.9 VDC at a distance of two inches from the sensor. With these tests complete we are now ready to perform a few simple experiments.

Experimenting

Before writing an algorithm that uses the sensor's data it would help to get a feel for what the sensor output actually looks like. Towards this end, we'll plot both the voltage versus angle of a target passing in front of the sensor on a semicircular arc, and the voltage versus distance to a target for two different lighting conditions.

Figure 4 shows the setup for the first experiment, in which we'll determine the approximate acceptance angle of the phototransistor in a configuration using re-

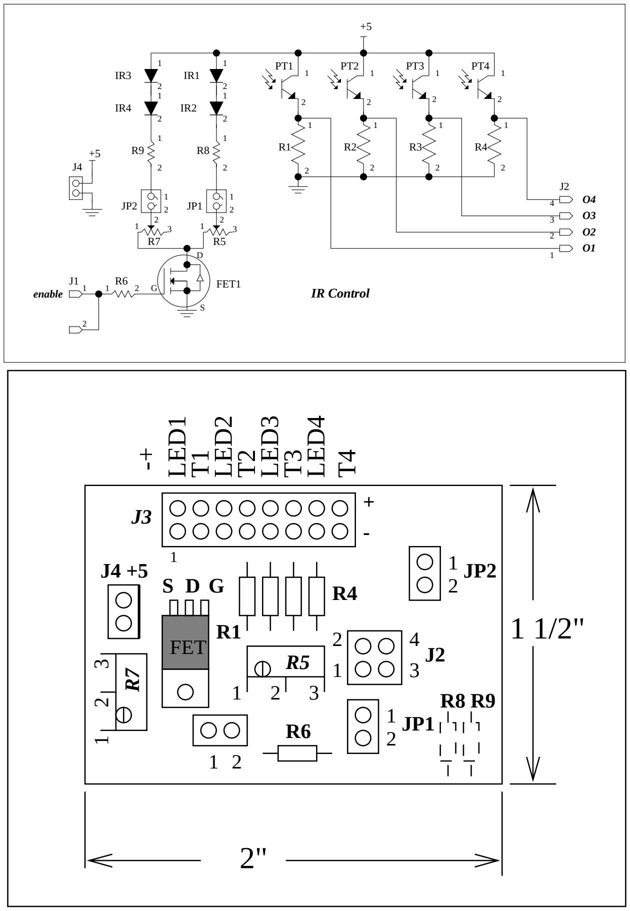

Figure 2: The schematic and mechanical layout of the IR obstacle detection module. The pot is used to adjust the forward current of the IR LEDs when switched ON by the FET. The analog voltage output from the IR phototransistors are sent to Port E (A/D input) of the 68HC11 where they are converted to digital values and compared with threshold functions.

flected energy. The emitter and detector are mounted 1/2 inch apart and facing a small one inch square white target placed at the same height as the sensor and at a distance of five inches. The target is moved on a semicircle in 10 degree increments and the voltage at the output of the phototransistor ($J2_1$) recorded. Figure 5 shows the results plotted for two lighting conditions, a dimly lit room and one with sunlight from a nearby window. Note the effect sunlight has on the overall readings. Since sunlight has natural infrared radiation it serves to increase the amount of noise the phototransistor sees.

One method to help separate signal from noise, is to compare the output voltage of the phototransistor with the IRED switched ON to the output voltage with the IRED switched OFF. The "OFF" reading from the phototransistor is ambient light and if the reading is the same as the "ON" reading then chances are there is no obstacle to reflect infrared energy[2]. If the readings are different, within a small error bar, then an obstacle has been detected.

Part Ref.	*Qty.*	*Description*
FET1	1	IRF510 Motorola N-channel TMOS Power FET TO-220AB
LED1-4	4	Siemens SFH 484 Infrared Emitter
T1-4	4	Siemens BP 103B-4 Phototransistor
R1-4	4	Resistor 100K 1/4 Watt
R5,R7	2	Trimmer Potentiometer 100 Ohms
R6	1	Resistor 10K 1/4 Watt
R8,R8	2	Resistor 15 Ohms 1/4 Watt
JP1-2, J1	3	Single Row Strip Header 2 contacts Male Straight
J2	1	Dual Row Strip Header 4 contacts Male Straight
J3	1	Dual Row Strip Header 16 contacts Male Straight
J4	1	Molex 0.100" Header 2 contacts Male Straight polarizing wall

Table 1: Parts list for the IR obstacle detection module. Any commonly available IR pair may be substituted, just remember to adjust the forward LED current and measure the angle of acceptance of the phototransistor.

The next experiment measures the output voltage as a function of distance to a white target. Record the output voltage of the phototransistor (with the IRED ON) as you move a white paper envelope along a tape measure towards the sensor. Also try the same experiment with different targets and varying lighting conditions, then plot your results as voltage versus distance to target. Figure 6 shows an example using a white paper envelope with results for both the IRED ON and OFF conditions. In Figure 7 a comparison is made with another phototransistor commonly available at Radio Shack.

Now that we have a feel for the type of data we can expect from our sensor, the next step is to design a configuration that will allow us to turn obstacle detection into avoidance. We'll test our obstacle detection scheme on a simple reactive robot that uses two wheel motors for steering and propulsion. By using a random-walk, two avoidance behaviours and a simple model for motion, the robot reliably wanders a cluttered room with relative ease using just two sensors.

---

[2] Although this is also true of objects whose colour or surface characteristic doesn't reflect infrared, making them IR invisible

A Simple Model for Motion

When we build robots that autonomously roam their environment, we create a mapping between perception and action that forms the basis of how the robot behaves for any given stimulus. This mapping might be fixed in the case of a reactive robot or adaptive if some form of learning is used. Obstacle avoidance represents one such perception-to-action mapping necessary in any successful navigation scheme. To use our just created obstacle sensors we'll need to decide what the robot should do when an obstacle is detected. This will depend very much on the type of motion the robot is capable of generating.

In a charming rendition of imaginary robotics[3], Valentino Braitenberg creates a set of 14 increasingly complex robots based on simple stimulus-response behaviours [1]. Braitenberg's second robot is capable of moving using a pair of left and right wheel motors. Both the direction and speed of each motor can be changed, resulting in a simple differentially steered platform on which to mount an assortment of stimulus specific sensors. Behaviour is created by "wiring" left and right sensors to either the left or right wheel motors of these hypothetical machines. To create a light-seeking robot the left and right light sensors are cross-connected to the left and right wheel motors, such that when a stimulus appears on the left side it causes the right wheel motor to turn forward causing the robot to turn left towards the light source.

For our purpose here, we'll use the same simple model for motion and create a set of actions which result in the robot's movement in small fixed increments. The robot pictured in Figure 9 is capable of the moving using the following motion commands: *forward, left-turn, left-rotate, right-turn, right-rotate, stop* and *idle*. The *idle* output simply means the behaviour is not competing for control of the robot's actuators. Since our plan is to use just two forward pointing sensors we won't include *backward* in our motion command set. All motion is in discrete increments of 1/8 inch linear or 2° angular movement. Thus to get the robot to move forward a constant stream of *forward* commands are issued on each iteration of the perception/action loop. We can create a WANDER behaviour that moves the robot forward and occassionally issues a random left or right turn.

On the perception or input side of our model we'll divide our obstacles into near (≤ 5 inches) or far (> 5 but ≤ 10 inches) by thresholding the sensor's output to be used for CONTACT and AVOID behaviours shown in Figure 6. Triggering the thresholds will cause the robot to rotate or turn away from the obstacle depending on which threshold was tripped. Table 2 lists the stimulus-response mappings used to create our two avoidance behaviours. A "1" in the *Stimulus* column indicates a threshold voltage has been reached by either the robot's left or right obstacle sensor. If both sensors detect an obstacle we arbitrarily choose to go right to avoid it.

Obstacle Sensor to Actuator Mapping

Stimulus		Obstacle Avoidance Behaviours	
L	R	AVOID	CONTACT
0	0	*idle*	*idle*
0	1	*left-turn*	*left-rotate*
1	0	*right-turn*	*right-rotate*
1	1	*right-turn*	*right-rotate*

Table 2: The stimulus-response mappings for obstacle avoidance behaviours. Obstacle sensor data is thresholded to produce boolean outputs, which are divided into Left and Right stimuli to match the left and right division of the actuator model. The "idle" output means the behaviour doesn't compete for actuator control.

The minimum straight-on detection range can be varied by the careful positioning of the left and right obstacle sensors. Figures 8 and 9 show how the left and right obstacle sensors are mounted and oriented on our circular robot. The sensors are pointed so that their field-of-views overlap at approximately 24 inches from the robot, allowing for their diverging sensing cones to detect head-on collisions with another similar robot. To generate collision free motion a WANDER behaviour is added to the CONTACT and AVOID behaviours and processed in a prioritized loop with each behaviour issuing its own motion commands. The controller is then tested in a cluttered room and the sensors orientation adjusted until collision free navigation is achieved.

To test the obstacle detection scheme in a more dynamic environment, 10 identical robots were initially placed together and switched on. Obstacle avoidance causes them to disperse while trying to maintain the minimum obstacle to robot distance in the forward looking view (see Figure 10).

Of course not all obstacles will reflect infrared energy in an amount detectable by the phototransistor. Having said that, your obstacle avoidance system should also include contact sensors so that in the event of a hard collision your robot can still recover, possibly with an apologetic "oh sorry I didn't see you."

---

[3] Required reading for any serious roboticist.

Future Enhancements

A more omni-directional obstacle sensor can be had by either adding additional IR sensor pairs or by rotating a single sensor on a servo or stepper motor. For the later to be useful, you would also need to know the position the sensor is pointing when a reading is taken. Depending on the angular resolution you wish to obtain, you may avoid having to add a slotted encoder wheel to obtain motor position by using the step count when using a stepper motor, or the pulse width input to position the servo motor. In each case the angular resolution of your sensor readings should be greater than twice the motor's positional error.

For example, if you command the servo, with a ± 2 degree position error, to a position of 5 degrees, then it may actually be anywhere from 3 - 7 degrees. In order for your sensor to "see" a stimulus source at 5 degrees the sensor's acceptance angle should be at least 4 degrees (or ± 2 degrees from center); so that if the position, say for example, is at 3 degrees then the sensor will detect a stimulus source radiating at an angle from 1 - 5 degrees.

Since the phototransistor's spectral sensitivity is usually as low as 400 nanometers (see Figure 1) they can also be used to track other light sources. A simple game of Cops 'N Robbers, played in our mobile robotics course two years ago, is possible by adding a bright light to the tops of two robots. Program the Cop to chase the light by using a seek-light behaviour, and the Robber should avoid the Cop by using a seek-dark behaviour. Videotape the game and send us a copy and we'll send you a University of Alberta robot team shirt for your effort.

References

[1] Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*, MIT Press, Cambridge, MA.

[2] Optoelectronics Infrared Products Catalog E26, Honeywell, 1993.

[3] The Optoelectronics Data Book for Design Engineers, Fifth Edition, Texas Instruments Inc. 1978.

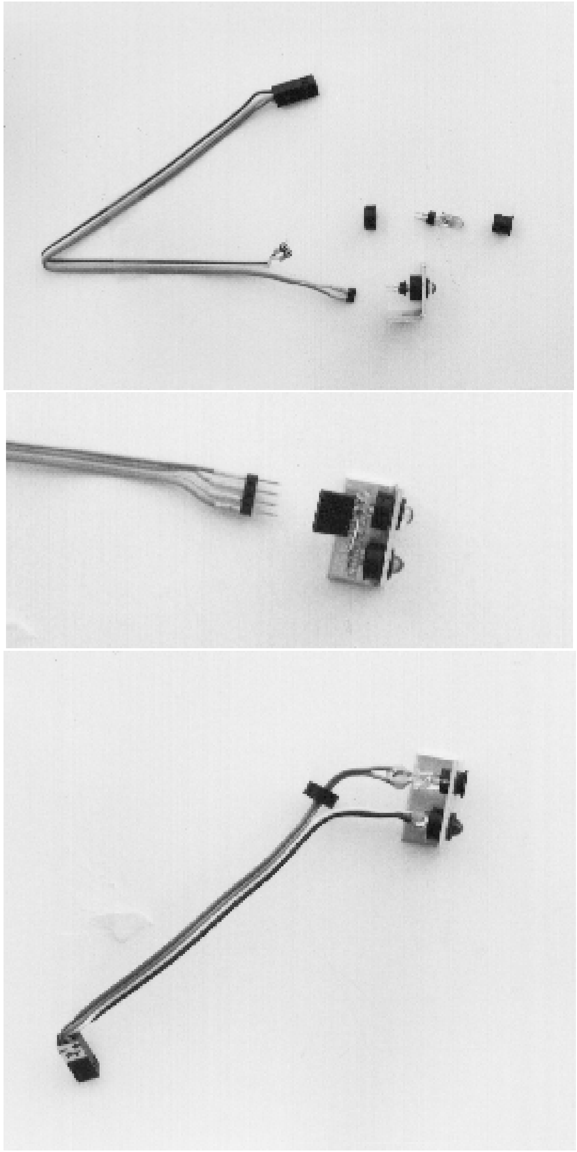

Figure 3: Shown are three techniques for connecting the IR sensor. The top photo uses a plug/socket connection. The middle photo solders both the IR LED and phototransistor to a small circuit board with female header. The bottom photo shows how hot-glue can be used to strengthen a wire to component solder connection.

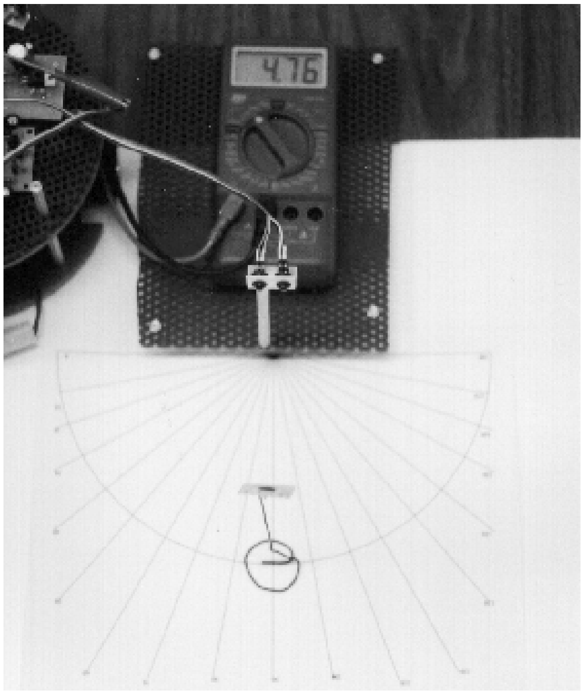

Figure 4: Once you've mounted the emitter/detector you can determine its angle of acceptance by making a voltage as a function of angle plot. Place a small target at the height of the sensor and record the output voltage as you move the target along a semicircle.

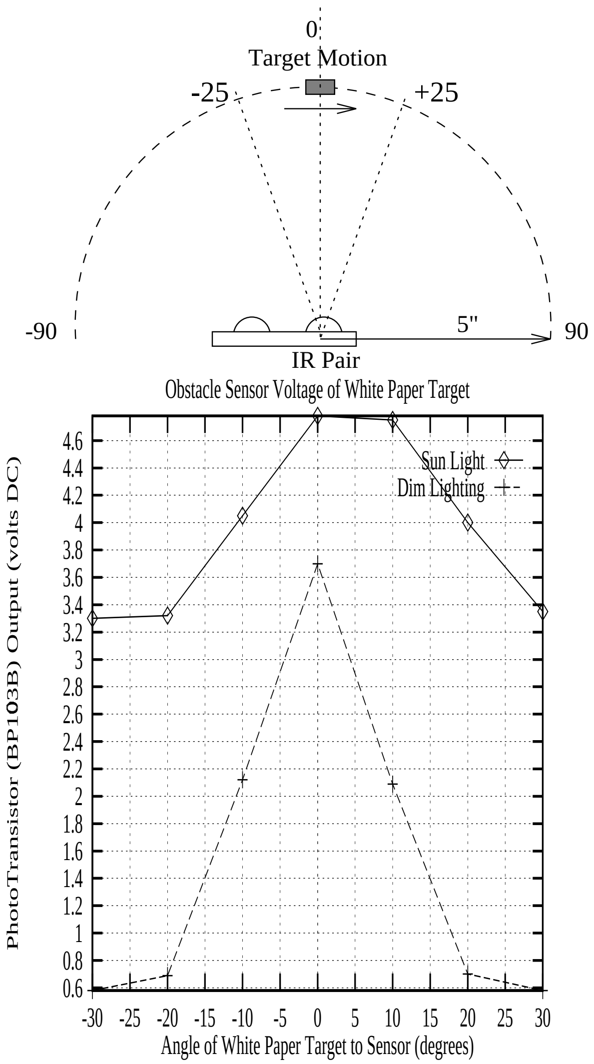

Figure 5: The output voltage of a Siemens BP103B phototransistor as a function of angle to a small 1" by 1" white target. Shown are two room lighting conditions: Sun Light (which contains lots of IR noise) and a dimly lit room with low ambient IR noise.

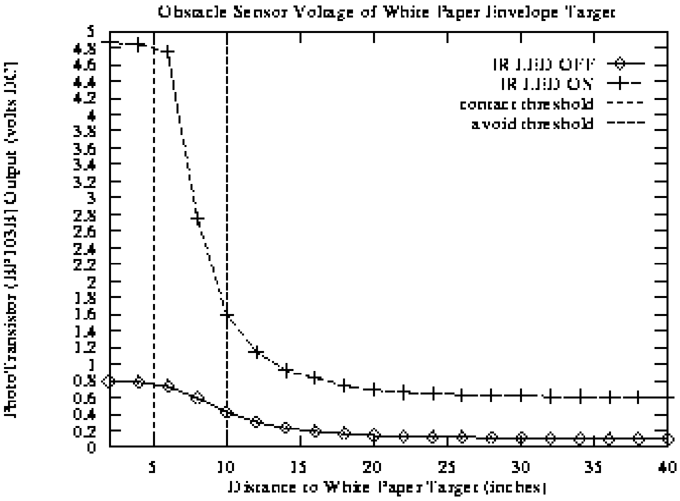

Figure 6: The output voltage of a Siemens BP103B phototransistor as a function of distance to a white target. Shown is the output of the phototransistor with the IR LED (a Siemens SFH484) on and off. Ambient IR noise can be eliminated by subtracting the "OFF" from the "ON" voltage.

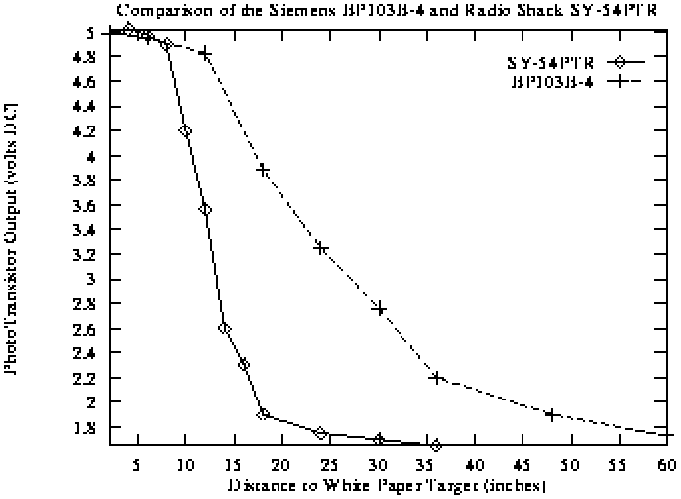

Figure 7: Compared are the output voltages of a Radio Shack SY-54PTR and a Siemens BP103B phototransistor as a function of distance to a white target.

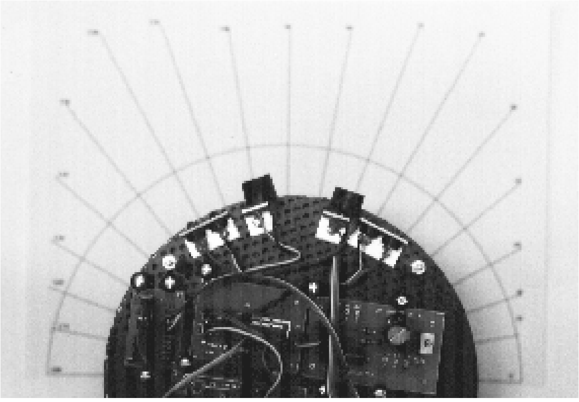

Figure 8: For a minimal obstacle detection system, mount a left/right pair of IR sensors pointing at an angle such that their field-of-view (angle of acceptance) overlaps at the minimum straight-on detection range.

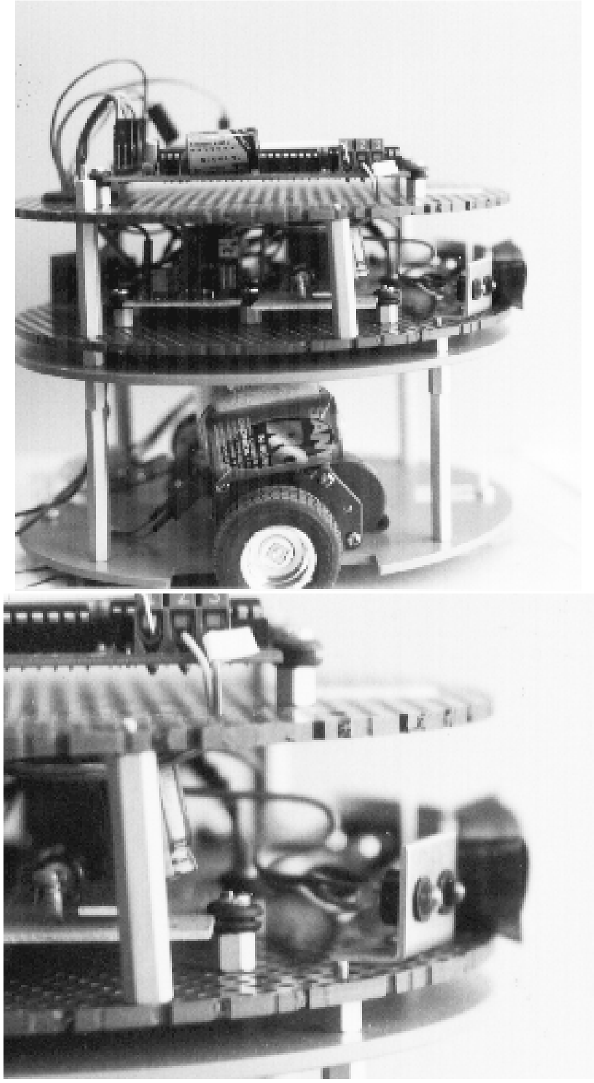

Figure 9: The IR LED/phototransistor pair can be mounted in 1/4" hole using a standard LED mounting holder on an aluminum bracket as shown.

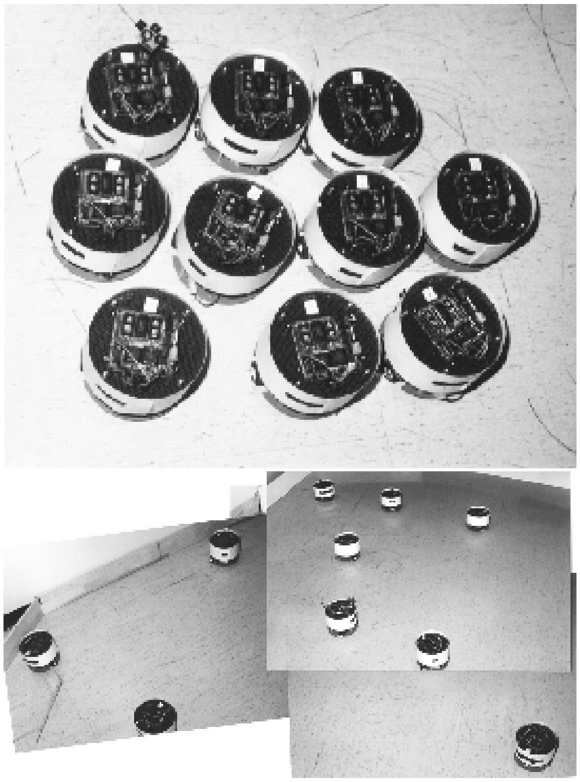

Figure 10: By adjusting the obstacle avoidance thresholds, you affect the distance the robot maintains from detected obstacles. Shown here are the start and end positions for 10 robots trying to maintain a minimum avoidance distance. These robots only detect obstacles in the forward direction using two sensors.