Problem Set 1 — 2.086 Spring 2012

Released: *Tuesday, 14 February 2012, at 4 PM.*

Due: *Tuesday, 28 February 2012, at 2:30 PM by hardcopy at the beginning of class.*

A Robot in a Room

It is critical for a robot to be able to determine the layout of its environment and also its own position within the environment for purposes of navigation and also task completion. In this problem set we consider a simple scenario: a robot first scans the environment to determine the shape (boundaries) of a room and its position within this room; the robot then calculates the area of the room in order to plan for tasks. (These tasks could take a variety of forms such as cleaning the surface or perhaps inspecting the surface for the presence of flaws or contaminants.)

In Figure 1 we depict the actual robot equipped with an Infrared (IR) Range-Finder which will serve to determine distance (to obstacles) and hence room shape. (In fact, the robot is equipped with several Range-Finders, but we will consider here data obtained from a single sensor.) We shall describe the IR Range-Finder in more detail below. Note the IR Range-Finder is mounted on a turret which rotates relative to the robot and hence the robot can scan the surroundings while remaining stationary.

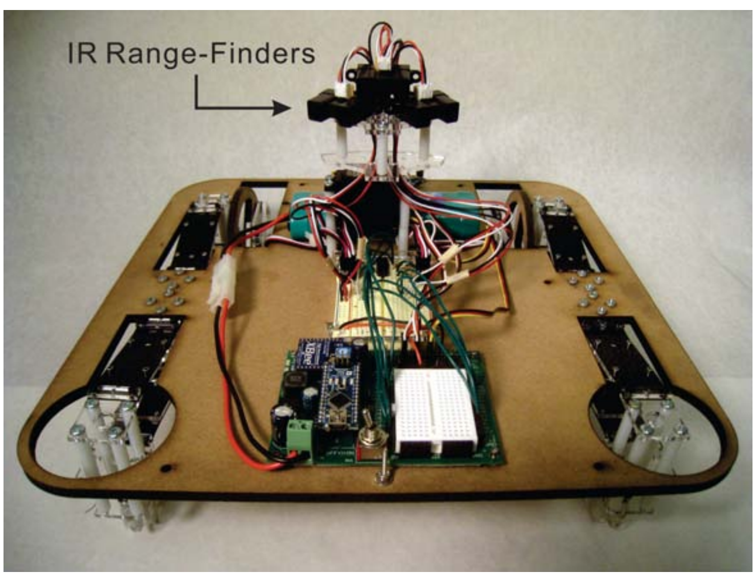


Figure 1: Robot equipped with an Infrared (IR) Range-Finder.

In Figure 2 we show a schematic of the robot in the room. We are concerned only with the two-dimensional plane, and hence we may describe the room boundary as $(X(\theta), Y(\theta))$ for $0 \leq \theta < 2\pi$. Note the coordinate system is centered on the robot IR Range-Finder and furthermore we assume that $\theta = 0$ is aligned with the positive x direction. (The front of the robot is referenced to $\theta = 90°$.)

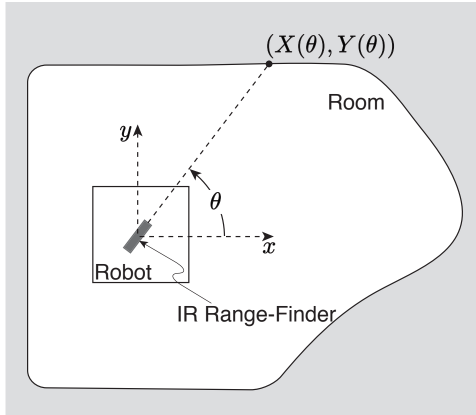
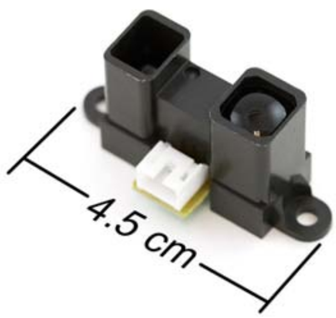
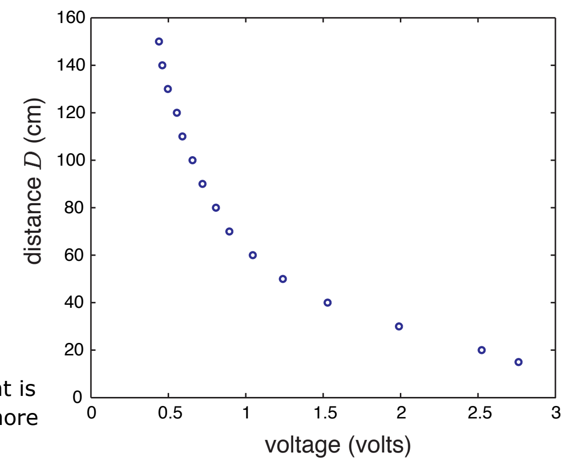


Figure 2: Schematic of the robot in the room.



(a) The Sharp IR distance sensor



(b) Calibration curve

Figure 3: (a) Sharp GP2Y0A02YK0F infrared distance sensor and (b) associated calibration curve.

Infrared Range-Finder

Infrared range-finders are relatively inexpensive distance sensors which nevertheless achieve reasonable performance speed and accuracy relative to much more expensive approaches.

Our robot is equipped with the Sharp BP2Y0A02YK0F IR distance sensor shown in Figure 3(a). The Sharp sensor uses triangulation to calculate distance from measurements of the angle of incidence of IR beams reflected from the distant surface of interest back to a receiving position sensitive device (PSD) mounted on the transducer. We show in Figure 3(b) the calibration data: the independently determined distance D (measured in cm) to a surface *versus* the voltage recorded by the transducer. Note this calibration data is discrete we perforce consider only a finite number of positions for our calibration wall and hence we will need to use interpolation in order to obtain accurate voltage-to-distance conversion for the actual room data. (Note we expect D to vary inversely with voltage, however we ask you to feign ignorance of this dependence for the purposes of this particular problem set.)

Instructions

You should download from the website the file `PSet1.mat`.

This file, which you should `load` into your workspace, contains single-index row arrays `RefDistance` and `RefVoltage` which contain the calibration data of Figure 3(b).

This file also contains single-index row arrays `RoomVoltage` and `RoomTheta_Deg` for the actual room data: `RoomVoltage(i)` is the voltage measured at angle `RoomTheta_Deg(i)`, for $i = 1, \ldots, K \equiv$ `length(RoomTheta_Deg)`; note the angles are distinct and positive and ordered in strictly increasing (counter-clockwise) fashion. Note that `RoomDistance` and `RoomTheta_Deg` do not wraparound we *do* include $\theta = 0$ but do *not* include $\theta = 360°$ (the last angle is less than and distinct from $360°$).

The `RoomTheta_Deg` data is in degrees rather than radians, so you should convert to radians a new vector `RoomTheta` promptly to avoid any subsequent errors.

Room Shape

1. (10 pts) Plot `RefDistance` *versus* `RefVoltage`. Make sure that in all figures in all problem sets you label your axes, add a plot title, indicate all units as appropriate, and include a legend (whenever there is more than a single curve in a figure).

   The deliverable here is the plot.

2. (15 pts) Write a script which converts a single-index row array of test voltages `TestVoltage` to a single-index row array of distances `TestDistance` based on piecewise linear interpolation with respect to the distance-voltage calibration data of `RefDistance`, `RefVoltage`. (The first lines of your script should load the calibration data and define the single-index row array `TestVoltage`.) You should find Exercise 4. of MATLAB Exercises_Recitation 2 quite useful in this regard.

   The deliverables here are

   (i) the MATLAB script (copied and pasted into your problem set document), and

   (ii) a table which compares the output of your script for `TestDistance` with by-hand calculation of the same quantity for the particular case of `TestVoltage = [1,1.5,2]`.

   The purpose of the latter is of course to confirm that your code is performing correctly.

3. (10 pts) Now run your script with `TestVoltage = RoomVoltage` to determine `RoomDistance`. Then process the array `RoomDistance` transforming from polar coordinates to Cartesian coordinates to obtain $\widehat{X}(\theta_i), \widehat{Y}(\theta_i)$, $1 \le i \le K$; $\widehat{X}(\theta_i), \widehat{Y}(\theta_i)$, $1 \le i \le K$, is our experimental/data-based approximation to the actual room boundary $X(\theta_i)$, $Y(\theta_i)$, $1 \le i \le K$. You should store $\widehat{X}(\theta_i)$ and $\widehat{Y}(\theta_i)$ as two separate single-index MATLAB row arrays `Xvec` and `Yvec`, respectively, each of `length` K.

   The deliverable here is a plot of the room shape: the room boundary, as described by $\widehat{X}(\theta_i), \widehat{Y}(\theta_i)$, $1 \le i \le K$. You should find the standard `plot` routine up to the job; please remember to label the axes and provide a title.

4. (10 pts) There are many sources of error in the distance measurement (and hence in $(\widehat{X}(\theta), \widehat{Y}(\theta))$ relative to $(X(\theta), Y(\theta))$: spurious scattering/reflections in particular near corners or rapid variations in the room shape (which should be apparent in your plot); the resolution of the PSD transducer; the error in the linear interpolation of the calibration data.

   Here we analyze briefly the error due to the PSD transducer. We know from independent sources that the measurement error in the voltage is $\pm E_V = \pm 0.02$ volts. We wish to estimate the induced error in the distance, $\pm E_D$ cm, for distances $D \approx 30$ cm and also $D \approx 130$ cm.

   The deliverables here are, for each of the two distances requested (approximately 30 cm and approximately 130 cm),

   (i) an estimate for the error in the distance, $\pm E_D$ cm, induced by the measurement error in the voltage, $\pm E_V = \pm 0.02$ volts;

   (ii) the derivation and justification of your error estimate.

Room Area

We know that the area of the room, A^{room}, can be evaluated as the line integral around the boundary of the room,

$$A^{\text{room}} = -\oint_{\text{Room Boundary}} Y(\theta)\, dX(\theta)\ . \tag{1}$$

In performing the line integral we proceed in counter-clockwise fashion such that (thanks to the minus sign out front) positive dx and negative dx conspire to give a positive area. The formula Eq. (1), and related formulas, can be deduced from standard integral theorems of vector calculus.

5. (10 pts) We now wish to approximate Eq. (1): we invoke experimental measurements of distance to deduce the room shape and then apply the trapezoidal rule to obtain

$$\widehat{A}_h^{\text{room}} = -\sum_{i=1}^{K} c_i\, \widehat{Y}(\theta_i)\ . \tag{2}$$

   Note that the hat indicates the approximation of the boundary from measurements/interpolation and then the subscript h refers to the trapezoidal rule approximation of the integral.

   The deliverable here is expressions for c_i, $1 \le i \le K$. *Hints*: For $2 \le i \le K-1$, the c_i will depend at most on $\widehat{X}(\theta_{i-1})$, $\widehat{X}(\theta_i)$, and $\widehat{X}(\theta_{i+1})$; c_1 will depend at most on $\widehat{X}(\theta_K)$, $\widehat{X}(\theta_1)$, and $\widehat{X}(\theta_2)$, and c_K will depend at most on $\widehat{X}(\theta_{K-1})$, $\widehat{X}(\theta_K)$, and $\widehat{X}(\theta_1)$. Recall that i increasing corresponds to counter-clockwise rotation, as desired.

6. (15 pts) Develop a MATLAB script which implements your trapezoidal rule approximation.

   The deliverable here is the MATLAB script (copied and pasted into your problem set document).

7. (10 pts) To test your integration code, replace (temporarily) in your script the actual room data with "synthetic room data": replace `Xvec` with `[1,1,-1,-1,1]`, `Yvec` with `[0,1,1,-1,-1]` and K with 5.

   The deliverables here are

   (i) the answer your code should provide *and* your simple derivation of this result (you should not need to actually execute your integration script by hand), and

   (ii) the answer your code actually does provide.

   The purpose here is of course to test that your code performs correctly. Do not just say "it works": provide the numbers requested.

8. (10 pts) Now apply your integration code to the actual room data.

   The deliverables here are

   (i) the value for $\widehat{A}_h^{\text{room}}$ predicted by your script, and

   (ii) your "guess" for the room area based on your plot of Question 3, eyeball estimates for dimensions, and a simple area derivation. (*Always* perform a "sanity check" as a good way to avoid blunders.)

   Please make sure to always present numerical answers within a full sentence which clearly indicates what the number actually represents in terms of well-defined variables; also provide units as appropriate.

9. (10 pts) Inspection of your room geometry plot of Question 3 suggests that $\widehat{X}(\theta_i), \widehat{Y}(\theta_i)$, $1 \leq i \leq K$, is certainly a bit noisy in particular near the corners. Do you think this noise will have a large, medium, or small effect on the accuracy of $\widehat{A}_h^{\text{room}}$ and why? You may assume that the noise oscillates relatively regularly between positive (an overestimation of the true distance) and negative (an underestimation of the true distance), even though this is not quite true for our experimental data.

   The deliverable here is an essay of no more than three sentences.

2.086 Numerical Computation for Mechanical Engineers
Fall 2012