

1. INTRODUCTION

This report describes the results of experiments on the current-versus-voltage characteristics of an LED. This report also describes the design of a simple circuit to drive the LED.

Sections 2 and 3 present results of two methods of fitting the current-versus-voltage characteristics of the LED to a Shockley law model of the diode: linear regression and the downhill simplex method.

Section 4 presents the calculation of a suitable resistor value to be used in series with the LED based on the approximation that the voltage drop across the LED is constant.

2. LINEAR REGRESSION MODEL FOR LED

2.1 An LED's voltage and current were measured with a simple experimental setup consisting of a 12 Volt power supply across a potentiometer and an LED in series. By adjusting the potentiometer, the voltage across and current through the LED were varied. Table I lists results measured with a digital current/voltage meter in the laboratory.

TABLE I
LED MEASUREMENTS

voltage (V)	current (mA)
1.30	0.6
1.35	2.2
1.40	13.6
1.41	21.1

2.2 We can use the data in Table I to fit a Shockley's law model describing the relationship between current and voltage for diodes:

$$i = I_s (e^{v/V_T} - 1) \quad (1)$$

where

- i ≡ current through diode in Amps
- v ≡ voltage across diode in Volts
- I_s ≡ reverse saturation current in Amps
- V_T ≡ thermal voltage = kT/q in Volts ≈ 26 mV at room temperature
- k ≡ Boltzmann constant = $1.38 \cdot 10^{-23}$ J/°K
- T ≡ temperature °K (293 °K = 68°F, 300°K = 80.6°F)
- q ≡ electronic charge = $1.602 \cdot 10^{-19}$ C

Because we wish to use linear regression as our first method of modeling the diode, we reduce Equation (1) to a linear approximation. Our first step is to ignore the -1 term in (1).

$$i = I_s e^{v/V_T} \quad (2)$$

Our second step is to take the natural log of both sides of (2):

$$\ln(i) = \ln(I_s) + \frac{v}{V_T} \quad (3)$$

Table II lists the values of $\ln(i)$ versus v , and Fig. 1 (in Section 2.3) shows a plot of this data.

TABLE II
LED MEASUREMENTS

voltage (V)	current (mA)
1.30	0.6
1.35	2.2
1.40	13.6
1.41	21.1

2.3 We observe that (3) has a linear form:

$$\ln(i) = a_0 + a_1 v \quad (4)$$

where

$$\begin{aligned} a_0 &\equiv \ln(I_s) \\ a_1 &\equiv 1/V_T = 38.462 \text{ V}^{-1} \end{aligned}$$

The Matlab™ script "polyfit_diode.m" in Appendix A finds the values of a_0 and a_1 that give the best fit, (in the least-squares sense), of the data in Table I to the line described by (4). Tables III and IV show the numerical results, and Fig. 1 shows a plot of the results. The differences between measured and modeled values are on the order of 10 percent.

We also note that the least-squares (or linear regression) fit gives a value of $a_1 = 32.461 \text{ V}^{-1}$ compared to $a_1 = 38.462 \text{ V}^{-1}$. This might arise from experimental error or leaving out the additive term of -1 in our analysis.

TABLE III
LED LINEAR REGRESSION PARAMETERS

parameter	value
a_0	-49.732
a_1	32.461

TABLE IV
PREDICTED LED MEASUREMENTS

voltage (V)	current (mA)
1.30	0.5
1.35	2.7
1.40	13.7
1.41	19.0

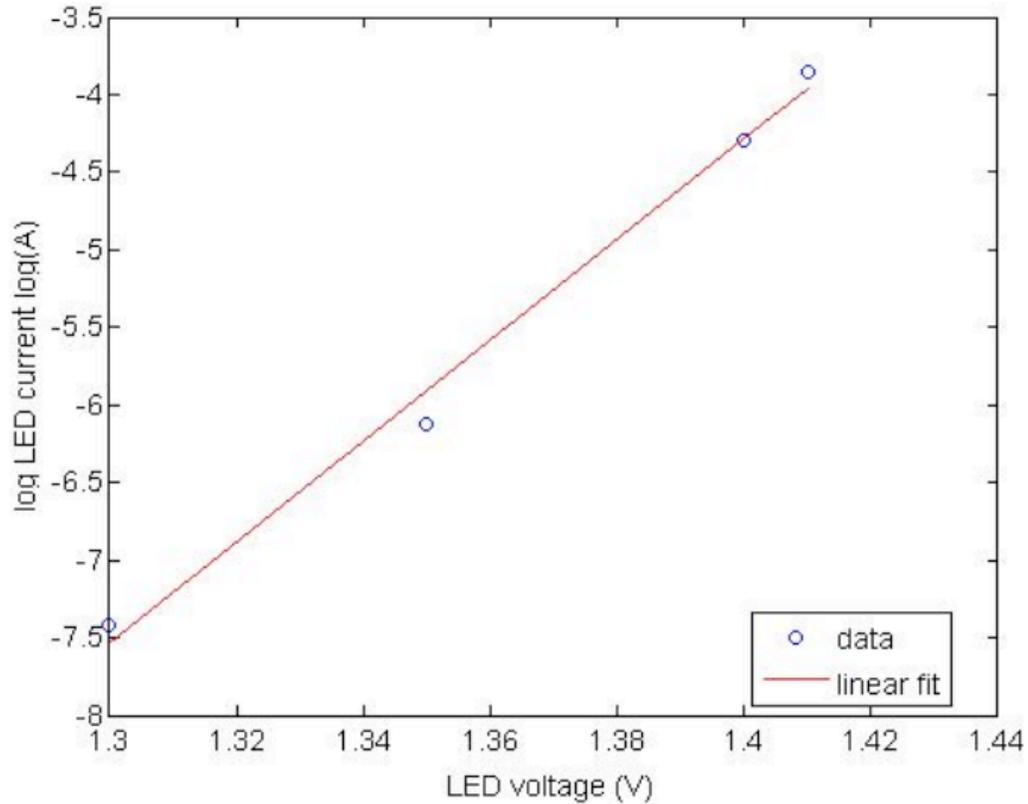


Fig. 1. Linear Regression fit of $\ln(i)$ vs V for LED.

3. "fminsearch" NONLINEAR OPTIMIZATION MODEL FOR LED

[Continue the report in the above fashion. Include figures where appropriate]

4. LED DRIVER CIRCUIT DESIGN

[Derive the driver resistor value here. Include a circuit diagram as Fig. 2.]

5. CONCLUSION

The parameters for the LED Shockley law model, (found by linear regression and the Matlab™ "fminsearch" function), differ in accuracy by approximately five percent. The modeling errors for the linear regression (or liner least-squares) fit are on the order of ten percent. The modeling errors for the fminsearch (or downhill simplex) fit are on the order of five percent. The fminsearch approximation may be more accurate because it directly models the nonlinear Shockley law. Since the two models differ only slightly, we obtain nearly the same resistor value of 2 kΩ for the driver circuit shown in Fig. 2 in both cases.

APPENDIX A

```
% script file polyfit_diode.m
%
% Sets arrays of measured data and uses polyfit function
% to find best values for a linear regression model of
% ln(i) vs v.
%
% The motivation for the linear regression model is the
% Shockley's Law diode model:
%  $i = I_s (\exp(v/vT) - 1)$ 
% where  $I_s$  = reverse saturation current
%  $vT = kT/q$  is thermal voltage
%  $T$  = temperature in degrees K

% Define physical constants.
k = 1.38e-23; % J/degK Boltzmann constant
q = 1.602e-19; % C electron charge

% Create arrays containing measured data.
v_diode = [1.3 1.35 1.4 1.41];
i_diode = [0.0006 0.0022 0.0136 0.0211];

x=v_diode;
y=log([0.6e-3, 2.2e-3, 13.6e-3, 21.1e-3]);

a = polyfit(x,y,1);

% Calculate model predictions for the measured voltages.
pred_i_diode = a(2) + a(1) * v_diode;

% Plot the measured and model values on the same plot.
axes('FontSize',14)
% hold off
plot(v_diode, y, 'bo')
hold on
xlabel('LED voltage (V)','FontSize',14);
ylabel('log LED current log(A)','FontSize',14);
plot(v_diode, pred_i_diode, 'r-');
legend('data','linear fit','Location','Best');
% hold off
```

APPENDIX B

```
% script file fminsearch_diode.m
%
% Sets arrays of measured data and uses fmins function
% to find best values of reverse saturation current, Is,
% and temperature, T, for a Shockley's Law diode model:
%  $i = I_s (\exp(v/vT) - 1)$  where  $vT = kT/q$  is thermal voltage.

% Declare global variables for measured data so they are
% accessible inside tot_sq_err function.
global v_diode
global i_diode

% Define physical constants.
k = 1.38e-23; % J/degK Boltzmann constant
q = 1.602e-19; % C electron charge

% Create arrays containing measured data.
v_diode = [1.3 1.35 1.4 1.41];
i_diode = [0.0006 0.0022 0.0136 0.0211];

% Find optimal Is and T by calling fmins function.
% Matlab requires that Is and T be placed together in
% one array. We call this array x:
%  $x = [I_s \ T]$ 
% We must specify initial values, which we call x0, where
% fmins will start its search.
% We use the linear regression solution as our starting guess.
% (Note: if the initial guess is too far off, fmins may fail.)
x0 = [2.5e-22 358.0]; % x0 = [Is T]

% fmins returns the optimal array, x.
% Note that we also must have a function called tot_sq_err
% that fmins can call as tot_sq_err(x) with x as argument.
% The name of the function is our choice, but we must tell
% fmins what the name is. This function is defined in a file
% named tot_sq_err.m, and we must set the path appropriately
% (under FILE menu) so Matlab can find this file.
% tot_sq_err must return a scalar value of the total model
% squared error when the model parameters are given by x.
x = fminsearch('tot_sq_err', x0);

% We extract Is and T from x as the first and second entries.
Is = x(1)
T = x(2)

% Done with the approximation. Now create a plot.
% Calculate model predictions for the measured voltages.
pred_i_diode = [];
for ind = 1 : length(v_diode)
    p_i_diode = Is * (exp(v_diode(ind)/(k*T/q)) - 1);
    pred_i_diode = [pred_i_diode p_i_diode];
end
```

```
% Print out predicted i values.
pred_i_diode = pred_i_diode

% Plot the measured and model values on the same plot.
axes('FontSize',14)
hold off
plot(v_diode, i_diode, 'bo')
hold on
xlabel('LED voltage (V)','FontSize',14);
ylabel('LED current (A)','FontSize',14);
plot(v_diode, pred_i_diode, 'r-')
legend('data','fminsearch fit','Location','Best');
hold off
```

APPENDIX C

```
function tot = tot_sq_err(x)
% Calculate total squared error between diode model and
% measured data.
%
% Requires globally defined arrays:
% v_diode = array of measured voltages for data points
% i_diode = array of measured currents for data points
% Example:
% If the data pts as ordered pairs of (voltage, current) were:
% (0.0, 0.13), (0.1, 0.24), (0.2, 0.37).
% The corresponding arrays would be:
% v_diode = [0.0, 0.1, 0.2];
% i_diode = [0.13, 0.24, 0.37];

global v_diode
global i_diode

% Extract Is and T from array x that is function argument.
Is = x(1);
T = x(2);

% Define physical constants.
k = 1.38e-23; % J/degK Boltzmann constant
q = 1.602e-19; % C electron charge

% Clear the total squared error variable.
tot = 0;

% Add up the squared errors for all the data points.
for ind = 1 : length(v_diode)

    % Calculate the diode current from the model.
    i_model = Is * (exp(v_diode(ind) / (k*T/q)) - 1);

    % Square the error in the model prediction for this data point.
    i_err_sq = (i_diode(ind) - i_model)^2;

    % Add the squared error to the running sum.
    tot = tot + i_err_sq;
end

return
```