ELEC 3908 – Lab 2: Current Flow in the Bipolar Junction Transistor

1 Summary

In this experiment, the ORCAD PSpice circuit simulator will be used to capture data for a bipolar junction transistor. The data will be ported into Matlab, and used to extract the parameters of a common bipolar transistor model, the Ebers-Moll equations. Lastly, a functional neural network will be created and trained, in Matlab to model the BJT input/output relationships.

2 Theory

One of the most commonly used models for the bipolar transistor is a set of relations known as the Ebers-Moll equations. This model is derived by assuming "low-level" injection (i.e. the amount of extra injected minority charge is much less than the inherent majority charge present) in each of the emitter, base and collector regions. Solving the relevant equation in each area and applying boundary conditions at each of the junctions yields three equations, one for each of the emitter, base and collector currents. Although the most general form of the equations contains physical transistor parameters such as the width of the "neutral" base region, a more commonly used form collects these physical parameters into a smaller number of constants. In this form, the Ebers-Moll equations are:

$$I_E = I_{E_S}(e^{qV_{BE}/n_E kT} - 1) - \alpha_R I_{C_S}(e^{qV_{BC}/n_C kT} - 1) \tag{1}$$

$$I_C = \alpha_F I_{E_S}(e^{qV_{BE}/n_E kT} - 1) - I_{C_S}(e^{qV_{BC}/n_C kT} - 1) \tag{2}$$

$$I_B = (1 - \alpha_F) I_{E_S}(e^{qV_{BE}/n_E kT} - 1) + (1 - \alpha_R) I_{C_S}(e^{qV_{BC}/n_C kT} - 1) \tag{3}$$

Where V_{BE}, V_{BC} are the base to emitter and base to collector voltages (V),I_E, I_C, I_B are the emitter, collector and base currents (A), I_{E_S}, I_{C_S} are the emitter and collector saturation currents (A),α_F, α_R are the forward and reverse collector to emitter current ratios and n_E, n_C are emitter and collector ideality (or slope) factors. Note that the collector to emitter current gains α_F and α_R are less than one, and are related to the collector to base current gains β_F and β_R by:

$$\beta_F = \frac{\alpha_F}{1 - \alpha_F}, \qquad \beta_R = \frac{\alpha_R}{1 - \alpha_R} \tag{4}$$

Extraction of the parameters for the model from measured data (or simulated in our case) is in principle quite straightforward. A measurement of I_C , I_B and I_B in the forward and reverse active regions can be used to calculate α_F and α_R simply by calculating the appropriate ratios. Referring to (1), if V_{BC} is set to zero by shorting the collector and base together, then the exponential term involving V_{BC} will be 1, and the second term on the right hand side of (1) will vanish. If the value of V_{BE} is greater than a few times $n_E kT/q$, which is about 26 mV at room temperature so that this will be true for any V_{BE} greater than a very small value, then the exponential will be much greater than 1, and the emitter current will be given approximately by:

$$I_E \approx I_{E_S} e^{qV_{BE}/n_E kT} \tag{5}$$

Thus a measurement of I_E versus V_{BE} at $V_{BC} = 0$, plotted as $ln(I_E)$ versus V_{BE} will give the value of n_E (from the slope) and I_{E_S} (from the intercept).

Using equivalent arguments for equation (2), if V_{BE} is zero, the second term on the right hand side will vanish, and if V_{BC} is at least a few times greater than $n_E kT/q$, then the exponential term will be much greater than 1 and the collector current will be approximately given by:

$$I_C \approx -I_{C_S} e^{qV_{BC}/n_C kT} \tag{6}$$

Thus a measurement of I_C versus V_{BC}, plotted as $ln(I_C)$ versus V_{BC} will give the value of n_C and I_{C_S} .

The accuracy of this model for small terminal voltages is quite good, with the accuracy falling off at higher values due to the breakdown of the low-level injection assumption.

Although mathematically this process is straightforward, in practice the measurements can be difficult to perform accurately. A modern bipolar transistor has a forward collector to base current gain in excess of 300, which implies that for a collector current of 300 μA the base current is only 1μA, which is small enough to require specialized measurement equipment. In addition, the intrinsic series resistance of a current meter causes a voltage drop between the voltage source and the device terminals, necessitating another accurate measurement. However, since this lab is being performed in a simulator, the accuracy of measurement is not a problem. Take this into account while writing your report.

3 Experiment

1. Launch ORCAD PSpice and create a new project, making sure to save it in a known location.
2. Create an NPN transistor. They can be found in Place $\Rightarrow$ PSpice Component $\Rightarrow$ Discrete. See Figure 1.
3. Place DC current sources at the collector and emitter terminals, being sure to follow passive sign convention for the direction. Leave their values at 0A. See Figure 2.
4. Connect the collector and emitter terminals to the base, and add a ground at the base.

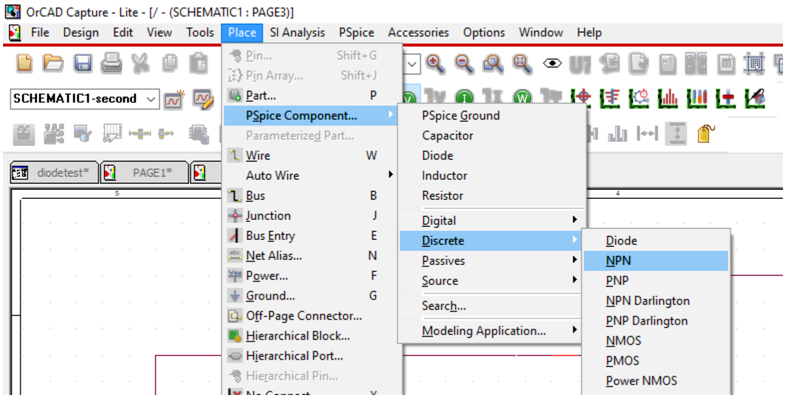


Figure 1: NPN Transistor Location

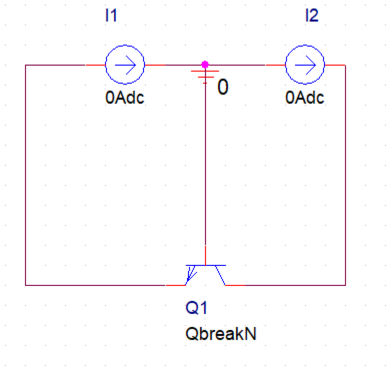


Figure 2: Forward Linear I-V Characteristics Circuit Set Up

3.1 Ebers-Moll Parameters of BC Junction

Perform the measurements necessary to extract the Ebers-Moll parameters. First extract the parameters of the BC junction.

5. Create a new simulation profile. Change the Analysis type to DC Sweep, set the sweep variable to Current Source and indicate the collector current source (I2 as shown in Figure 2). Set the start value to -1μA and the end value to -1mA with an increment of 0.1μA. See Figure 3.
6. Place a set of voltage differential markers at the base and collector pins. See Figure 4.
7. Run the simulation. Change the x - axis to be logarithmic, using the button in the toolbar menu and adjust and label the axis. The graph should be similar to the one in Figure 5.
8. Export the data as a comma separated file. Go to File ⇒ Export ⇒ Comma separated file. Name the file and press OK. This data will be plotted in Matlab for analysis.

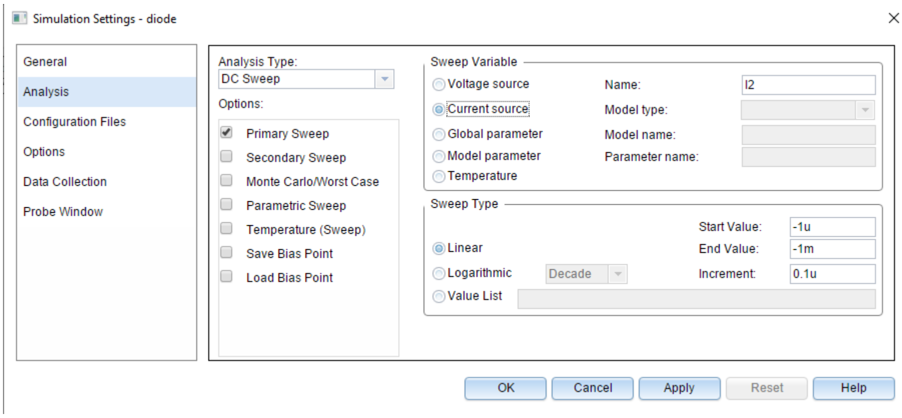


Figure 3: Simulation Settings

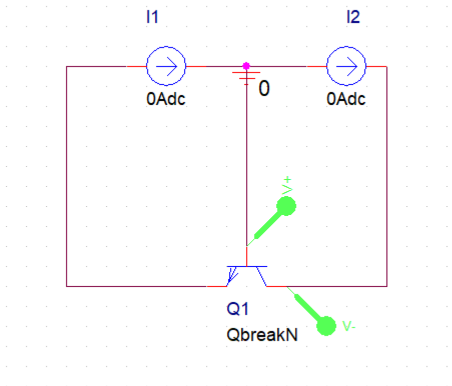


Figure 4: Voltage Indicator Placement for Base - Collector Junction Parameter Extraction

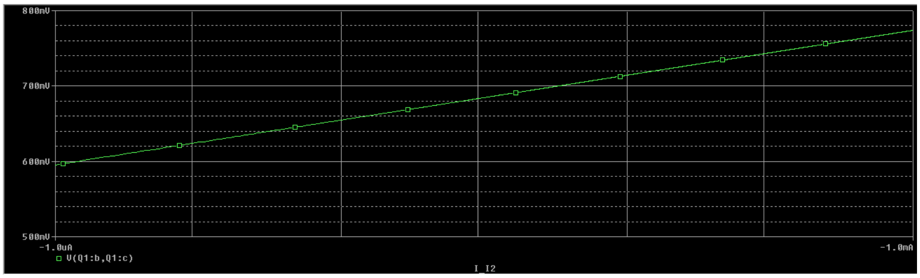


Figure 5: BC Junction Parameter Extraction Graph

3.2 Ebers-Moll Parameters of BE Junction

9. Go back to your circuit, and edit the simulation. Change the current source to the emitter current source(I1 as shown in Figure 4), and set the start value to 1μA and the end value to 1mA with an increment of 0.1μA.

10. Move the voltage differential markers to the base and emitter pins.

11. Run the simulation. Change the x - axis to be logarithmic, using the button in the toolbar menu and adjust and label the axis. Export the data as a comma separated file.

3.3 Forward and Reverse Current Gains

This section will complete the measurements necessary for the parameter extraction. It remains only to measure the forward and reverse current gains.

12. Go back to your circuit and remove the differential voltage markers. Move the ground to the emitter node. Replace the current sources with DC Voltage sources. See Figure 6.

13. Set the emitter voltage source to 0.67V and leave the collector voltage source at 0V.

14. Edit the simulation, change the sweep variable to voltage source, and indicate the collector voltage source (Ex. V1).

15. Set the Starting value to 0V and the end value to 1V with a step of 1mV.

16. Run the simulation, and add a trace of the three currents, the collector current[I(Q1:c)], the base current[I(Q1:b)] and the emitter current[I(Q1:e)], and export the graph for analysis in MatLab. On this graph, once all the currents are added, all the lines will appear flat because of the scale of the graph.

17. Repeat steps 12 - 16 with the BJT in reverse current gain set up. To do this, simply rotate the transistor 180 degrees by selecting it and pressing 'R' on your keyboard twice. Reconnect the terminals. The base should be connected to the same node, and the emitter and collector should now be switched. See Figure 7.

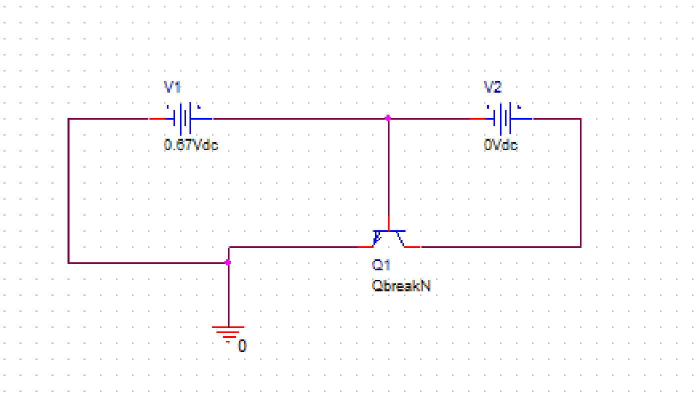


Figure 6: Foreword Current Gain Circuit Set Up

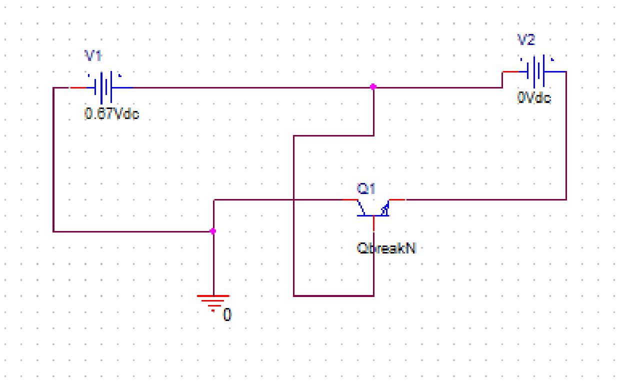


Figure 7: Reverse Current Gain Circuit Set Up

3.4 BJT I-V Characteristics

We now have enough data to do the parameter extraction. Now a typical active region I_C versus V_{CE} at constant I_B plot will be measured.

18. Return to your circuit, remove the emitter voltage source and the current gain markers. Place a DC current source at the base node. See Figure 8.

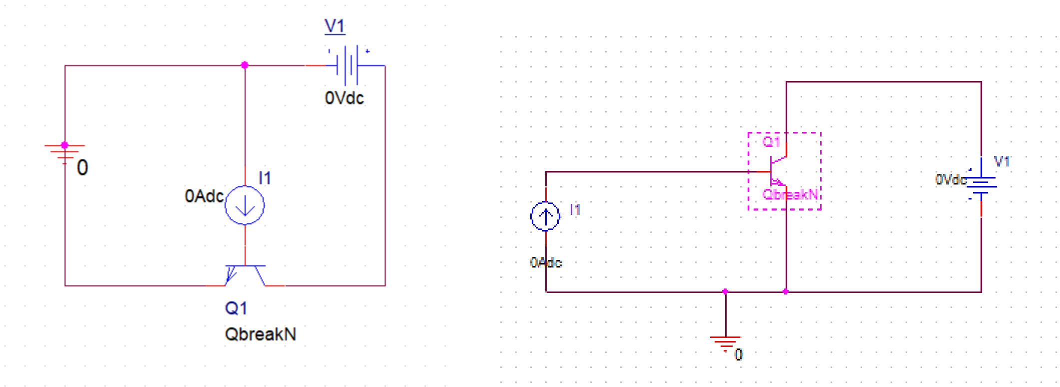
19. Edit the simulation, and set the primary sweep to the collector voltage source. Sweep from 0V - 5V, with a step of 0.1V. Next, under options, add a secondary sweep. Set this sweep to current source, and indicate the base current source. Sweep from 0 - 5μA with a step of 1μA.

20. Add a trace of the collector current [I(Q1:C)]. The graph should resemble figure 9. Export the data as a csv, but when you export, make sure to add the data from the base current source. During the export stage, click on the icon for the base collector source eg. [I(I1)] and then click add.

21. Change the collector voltage source to run from -2 to 5V. Simulate, plot and save the results. For this case you should save in the cvs file the current flowing through I1 (the base current) and the collector current. This will produce a cvs file with 3 columns (V_C, I_b, I_C) from which you will build a neural network model with two inputs (V_C, I_b) and one output (I_C).

22. Edit the NPN model and add the parameter "VAF = 5". Simulate, plot and save the results – make sure you save (V_C, I_b, I_C).

23. Finally, repeat all steps in this section using the PNP transistor. Be sure to REVERSE ALL VOLTAGE AND CURRENT SOURCES.



(a) (b)

Figure 8: I-V Characteristics Set Up a) As described. b) Alternative. Same circuit just differently laid out.

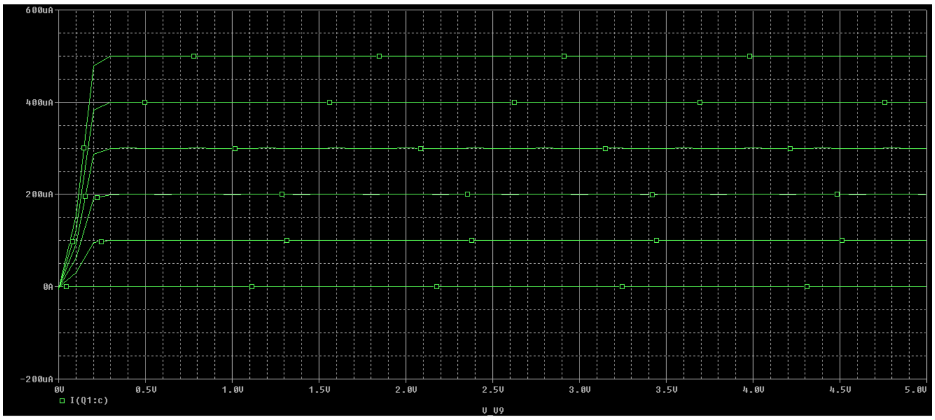


Figure 9: BJT I-V Characteristics Graph

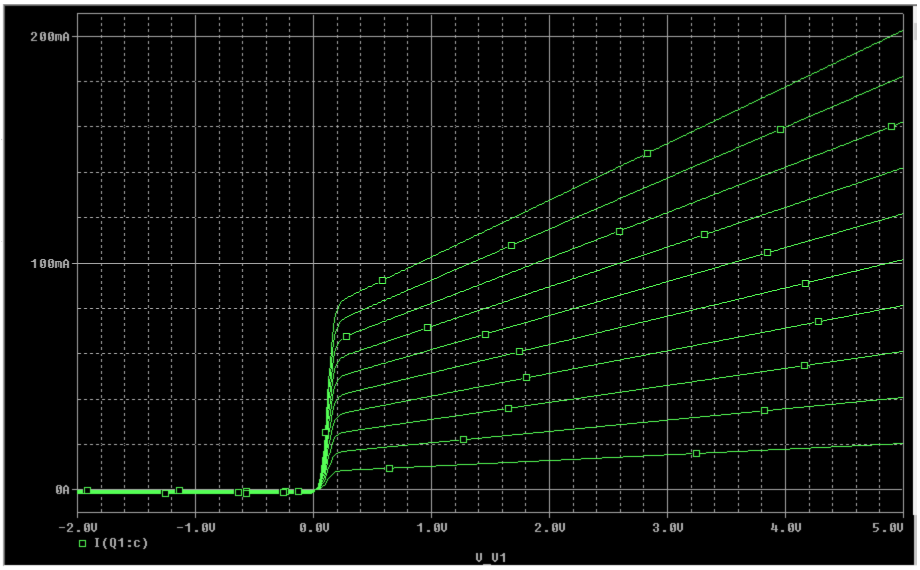


Figure 10: BJT I-V Characteristics Graph with VAF=5

3.5 Plots in MatLab

1. First, go to the file where your project is saved. You should find all of the csv files the you have saved so far. There should be basically 5 NPN files and 5 PNP files(Base Emitter Junction, Base Collector Junction, Foreword Current Gain, Reverse Current Gain and the I-V Characteristics) – plus some from the variations of the model. Move all files to a folder which will contain your saved Matlab scripts and data. This time, you will use the readfile function to import the data to be plotted and analysed.

2. The code in figure 11 will import the data from the .csv file, and as long as it is in the right place, will read the file into a matrix.

3. Use this code for the first two sets of data. Add a parameter extraction, the same way the parameters were extracted in Lab 1. The data does not need to be cut this time as it is linear the whole time. Make a plot of the data as well **Save these graphs for use in your write up and analysis, and present the extracted parameters.**

```
% Initialization steps.
clc;          % Clear the command window.
close all;    % Close all figures
clear;        % Erase all existing variables.
workspace;    % Make sure the workspace panel is showing.
format long; %Allows MATLAB to display all sig figs

Table = readtable('YOURFILENAME.csv');%import and read the data from pspice
y = log(abs(Table{:,1}));    %Takes the absolute value and the natural
                             %logarithm of Column 1 of the table and assigns
                             %it to y
x = Table{:,2}; %Assigns Column 2 of the table to x
LinearCoefficients = polyfit(x, y, 1); %This funtion outputs the linear
                                       %coefficients for a linear function
Slope = LinearCoefficients(1) %Prints the coefficitent values
Yint = LinearCoefficients(2)
equation = {'y = ' Slope 'x +' Yint} ;

figure('Name', 'BC_Junction_Graph')
plot(x, y)
title('Ebers-Moll Parameters of BC Junction')
ylabel('Natural Log of the Base Current (A)')
xlabel('Base - Current Voltage (V)')
text(0.72, -12, equation) %The numbers here are coordinates for where the
                          %text is to be placed on the graph, change them
```

Figure 11: Example Code for Matlab BE and BC Parameter Extraction

4. For the Current Gain data, the code will need to be changed. This time, we are looking to extract the values of α and β. For α, I_C must be divided by I_E, and then the average value of this column must be taken. See the example code in figure 12.

```
% Initialization steps.
clc;          % Clear the command window.
close all;    % Close all figures
clear;        % Erase all existing variables.
workspace;    % Make sure the workspace panel is showing.
format long; %Allows MATLAB to display all sig figs

Table = readtable('YOURFILENAME.csv');%import and read the data from pspice

collector = abs(Table{:,3}); %Must take absolute value to avoid sign error.
emitter = abs(Table{:,4});   %Be careful when assigning your variables
base = abs(Table{:,2});      %your data may be in different columns from
                             %this example set

alpha = mean(collector ./ emitter) %Use ./ for term wise division. The
                                   %mean() function takes the average of
                                   %the column.

beta = mean(collector ./ base)
```

Figure 12: Example Code for Current Gain Parameter Extraction

5. Run this code for all of the current gain data. Be sure to change the equations for the reverse current gain. When the BJT is in the reverse active position, the emitter becomes the collector and vice versa. The equations in the code must be adjusted accordingly. **Save the values of α and β for both the foreword and reverse current gains for use in your analysis.** Double check your answers using equation 4.

6. For the final graph, the data will need to split into chunks to graph it properly. Then a legend will need to be added to the graph to give information about the base current. Note that the code given in figure 13 is incomplete. Make sure your graph has 6 lines, and that all are referenced appropriately in the legend. Take some time to learn some of the Matlab functions to make it look nice.

```
% Initialization steps.
clc;              % Clear the command window.
close all;        % Close all figures
clear;            % Erase all existing variables.
workspace;        % Make sure the workspace panel is showing.
format long;      %Allows MATLAB to display all sig figs
opengl software %Fixes a bug with MATLAB's legend function

Table = readtable('YOURFILENAME.csv');%import and read the data from pspice

Vce = Table{1:51,1}; %Since the voltage repeats every 51 cells,
                     %you only need to make one variable
Ic0 = Table{1:51,2}; %The current changes for each repitition,
                     %so you'll need 6 Ic variables
Ic1 = Table{52:102,2};

figure('Name', 'GRAPH NAME')
hold on             %This will put all the plots on the same graph
plot(Vce, Ic0)
plot(Vce, Ic1)
title('TITLE')
ylabel('LABEL')
xlabel('LABEL')
legend('Ib = 0','Ib = 1uA', 'NumColumns',2) %Play around with the settings
                                            %to make the graph look nice
```

Figure 13: Example code for I-V Characteristics Graph

7. Once you are happy with your graph, **Save it for your write up.**.

4 Data Analysis

4.1 Parameter Extraction

1. We are now in a position to "extract" the Ebers-Moll model parameters from the simulated data.

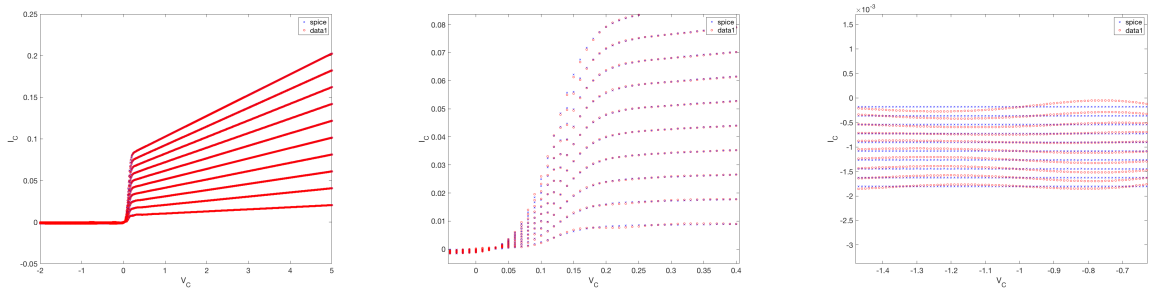
   - From the $ln(I_C)$ versus V_{BC} plot, $\frac{n_C kT}{q}$ may be read off directly as the inverse of the slope of the plotted line($\frac{1}{slope}$). Divide this value by 0.0259 (kT/q at room temperature) to give the value of n_C . The intercept of the line, given directly from the data as Y intercept, is the value of $ln(I_{C_S})$. Calculate the exponential of this value to give I_{C_S} in amps.
   - Similarly, from the $ln(I_E)$ versus V_{BE} plot, $\frac{n_E kT}{q}$ may be read off directly as the inverse of the slope of this plotted line. Divide this value by 0.0259 (kT/q at room temperature) to give the value of n_E . The intercept of the line, given directly from the data as Y intercept, is the value of $ln(I_{E_S})$ .Calculate the exponential of this value to give I_{E_S} in amps.
   - Present the values of α and β, in a table.

2. Verify that the measured values of α_F, β_F, α_R and β_R are related by equation (4).

3. Using the data generated in the Current Gain extraction (section 3.3). Evaluate the collector and base currents for a range of at least 3 or more values of V_{BC} using the values extracted for I_{C_S} ,I_{E_S} ,α_F and β_F, and the known value of V_{BE}. Calculate the error between the simulation and your calculation. Use table 1 to record your results. Show at least one example calculation of each type.

Table 1: Comparison of Results to the Model

V_{BC}	Simulated I_C	Calculated I_C	Neural Net I_C



(a) (b) (c)

Figure 14: Comparison of 20 Neural network with Spice data. a) Full model. b) Near $V_C = 0$. c) Reverse Operation.

4.2 Matlab Neural Net Development

You will build NN models for the NPN only, but for the cases of 1) V_C varying from 0 to 5V with VAF=0 and, 2) V_C varying from -2 to 5V with VAF=5. The basic instructions are as in Lab 1 but you will have two inputs V_C and I_b and one output I_C.

1. Create the input vector by reading the csv file into a Matlab matrix and extracting the first two columns into a 2 by N long matrix.
2. Create the output vector by reading the csv file into a Matlab matrix and extracting the third column.
3. In the Command Window type "nnstart", select Fitting App and click "Next" twice. In the select data page select the Input under Inputs and Output under Outputs accordingly. Also make sure to select Matrix Rows to properly divide the samples for the training. Click "Next".
4. Keep the neurons default and click next, you will go back and modify them later.
5. In the training algorithms, change the training algorithm to Bayesian Regularization as it is better suited for small data sets like this one. Next click "Train and wait for the algorithm to finish training". Then click "Next" twice.
6. Click, "Simple Script" and save the script to your computer. Next run the Simple Script, this should create your neural net in your work-space.
7. In the "Command Window" type some test numbers and record your results. When entering your results into the network, call the function net(x) – where x has two rows of inputs (V_C and I_B). Make sure your values are entered in as matrix rows, and not matrix columns.
8. Compare the NN model to the actual data in a plot.
9. If your result's are way off go back to the neural net GUI and increase the number of neurons in your neural network (watch out too high of a number may freeze or even crash your computer) and repeat the process. Enter the same values as you calculated and record the results in Table such as 1.

4.3 Discussion Question

1. Why are NPN transistors more commonly used than PNP transistors?
2. Explain where the error between the various results might come from given the fact that this lab was performed using a simulator.

5 References

Streetman, B. G., Solid State Electronic Devices, 3rd ed., Sections 7.2.1, 7.5.1.

Puifrey, D. and Tarr, N. G., Introduction to Microelectronic Devices, pp. 336-350, pp. 366-367.

Muller, R. S. and Kamins, T. I., Device Electronics for Integrated Circuits, p. 280.

Sze, S. M., Physics of Semiconductor Devices, p. 140, p. 145.