Design Objective

This Experiment was to replicate a Monte-Carlo Method. A Monte-Carlo method is a compter simulation where a production process is analyzed and the probability of a certain outcome is determined based on a large calculated dataset. In this experiment there were three different cases all with the same sample size and testing for the same outcome. However, the three cases were testing different situations in which the outcome wa possible. The manufacturer produced 100 Ohm resistors but did so by having: one single 100 Ohm resistor, 4 400 Ohm resistors in parallel, or 10 1,000 Ohm resistors in parallel. The manufacture produces their resistors with a 5% tolerance. They wanted to use the Monte-Carlo method to calculate the probability that their resistors are within a 1% tolerance.

Test Procedure

* Calculate 10,000 simulated resistor values using a random number generator
* Determine the number of resistors with a 1% error or less
* Determine the probability based upon the number of resistors that have 1% tolerance or less

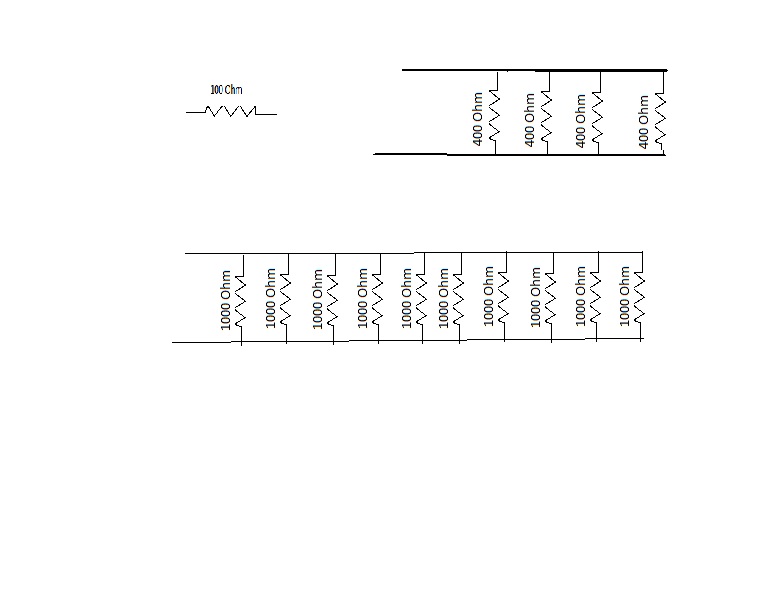
Circuit Schematic

Figure 3. 10 1000 Ohm Resistors in Parallel

Figure 2. 4 400 Ohm Resistors in Parallel

Figure 1. 100 Ohm Resistor

Matlab Code

disp('case 1')

R=[];

count = 0;

for i=1:1:10000

R(i) = (95 + rand(1)\*10);

if R(i)<=101 && R(i) >= 99

count = count + 1;

end

end

P = (count\*100)/i

subplot(2,2,1);

hist(R),title('Monte Carlo Case 1');

xlabel ('Resistance in Ohms'), ylabel('Number of Resistors');

disp('case 2')

for i = 1:1:10000

y(i) = 1/(380 + 40 \* rand(1)) + 1/(380 + 40 \* rand(1)) + 1/(380 + 40 \* rand(1)) + 1/(380 + 40 \* rand(1));

R(i) = 1/y(i);

end

subplot(2,2,2);

hist(R),title('Monte Carlo Case 2');

xlabel ('Resistance in Ohms'), ylabel('Number of Resistors');

total = 0;

for i=1:1:10000

if(R(i) >= 99 && R(i) <= 101)

total = total +1;

end

end

probability = total \* 100 / 10000

disp('Case 3')

for i = 1:1:10000

y(i) = 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1)) + 1/(950 + 100 \* rand(1));

R(i) = 1/y(i);

end

subplot(2,2,3);

hist(R),title('Monte Carlo Case 3');

xlabel ('Resistance in Ohms'), ylabel('Number of Resistors');

total = 0;

for i=1:1:10000

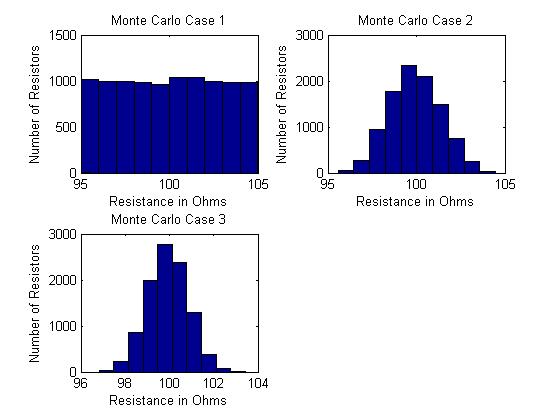
if(R(i) >= 99 && R(i) <= 101)

total = total +1;

end

end

probability = total \* 100 / 10000



Graphs

Figure 4. The three output Graphs of the Matlab Code.

Measured Data

Case 1

P = 20.0500

Case 2

probability = 49.4700

Case 3

probability = 72.4000

Discussion

The results of the data showed that the most accurate overall of the three cases was the last case where there were 10 1000 ohm resistors in parallel. The second most was the 4 400 ohm resistors and the least accurate was the single resistor. This can be attributed to the probability that all 10 resistors or all 4 resistors would all be in the higher range of the tolerance level. This would cause the resistors that are farther from the desired value to be compensated for by the ones that are closer. This is shown in Figure 4 in the histograms created by Matlab.

Overall this lab was easy to follow and very straight forward, the only thing I would change would be some of the wording on the instructions right in the beginning they were a little confusing at first and had to be reread. Once you understood what was going on however the lab was easy to do. The sample code really helped with the understanding of the directions.

Experiment 3

Monte-Carlo Method

Monday October 1st , 2012

**Francis Murray**

Ashley Cottrell

John Norman

EE 222-010