Ioannis M. Velgakis
150167271

EEE119 Component Tolerance Analysis Using Matlab

Introduction

An undefined principle that was always governing the entrepreneur world, since its establishment as a social structure in the early ages, was the maximization of the total revenue from a given product. For a company that its main product is somehow related to some electrical or electronic components, this main principle is translated to the reduction of failed components or the maximization of the yield (Appendix I) due to the components tolerance effect.

The primary purpose of this lab report is to perform components tolerance analysis based on two models, a) the extreme value analysis model and b) the Monte Carlo analysis (simulation) model, on three simple circuits: a potential divider (Figure 1), a low pass filter (Figure 2) and a 2-bit R–2R ladder DAC (Figure 3). An individual engineer nowadays has to understand the significance of a component tolerance analysis, especially in the design stage of a product, in order not only to deeply perceive the subjected variation of the components, but also to be able to predict the worst case scenario and to be able to design a backup plan or an alternative to avoid a possibly fatal outcome (airplane system's failure).

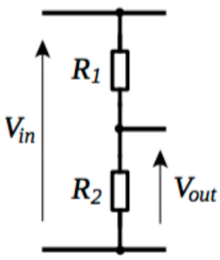

Figure 1: Potential Divider[1]

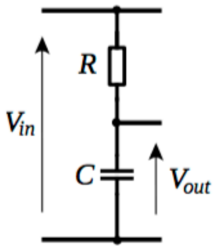

Figure 2: Low Pass Filter[1]

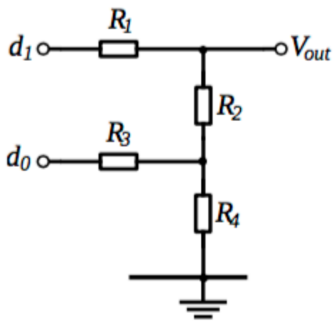

Figure 3 2-bit: R – 2R ladder DAC [1]

In the following, a brief description and explanation of the analysis models that will be used is given.

a) Extreme Value Analysis Theory Model[1][2]

As mentioned above in every electrical component a variation effect appears. In other words, in a sample of x manufactured electrical components from the same patch their values will be fluctuating in an interval between the mean value (δ) and the extreme values (δ-da, δ+da), where da is the modulus of the perturbation in the sample. The distribution of the components' values inside the interval and whether the modulus of the extreme values is wide or not (i.e. the manufactured components' values are closer to the median or not) depends on the manufacturing process. The extreme value analysis theory model (EVA) compares the operation and the efficiency of a system that operates with all its components to their nominal (median δ) value with a system in which all the components have their extreme tolerance values ($\delta \pm da$). Although this model can be easily used to analyze the majority of the systems it is not used widely because as the systems are getting bigger and are consisted of lots of components this analysis becomes chaotic and it is easy to extract any data at all.

b) Monte Carlo Analysis (simulation)[3],[5]

The Monte Carlo analysis is a method of solving various problems using computational statistic mathematics by constructing for each problem a random process with parameters equal to the required quantities of the individual problem. More specifically, for the components' tolerance estimates, thousands of identical circuits are being simulated at once using for the parameters (component's values) a set of values specified by each component's tolerance. These random sets of values are being retrieved from an interval created by the component's tolerance and the probability of each set is defined by the type of distribution that each set of values follow. Two types of distribution are used in this analysis: the uniform distribution (Figure 4) and the Gaussian (normal) probability distribution (Figure 5) (Appendix I).

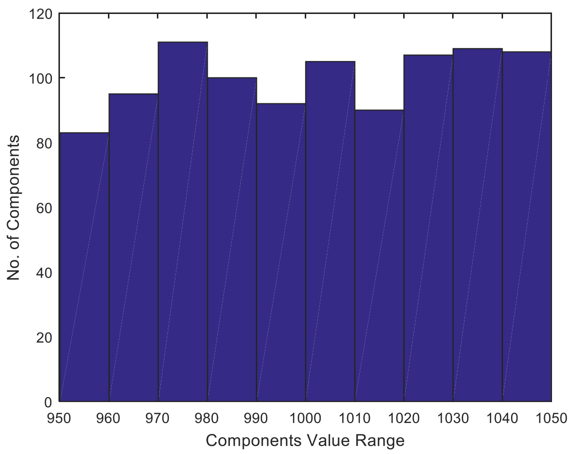

Figure 4: Uniform Distribution Example

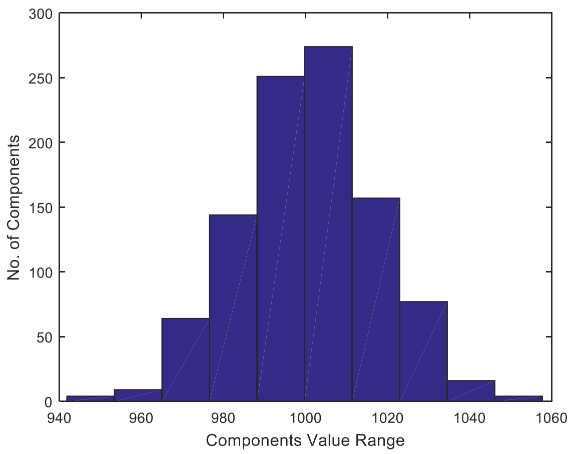

Figure 5: Gaussian (normal) Probability Distribution example

Results

The results section is divided in three sub-sections: (i) the Potential divider, (ii) the Low – Pass filter, and (iii) the 2-bit R-2R ladder DAC. In each section there is a brief description of the circuits function and the tolerance analysis using both the methods discussed before. All the formulae used are included in Appendix I, and also, all the matlab[4],[10] programming code in Appendix II.

(i) Potential Divider[6]

The Potential (Voltage) divider (Figure 1) is a very important basic circuit with many practical applications. It allows to supply a circuit with a voltage different from that of an available power source. The particular potential divider that we are going to examine has the following specifications: $R_1 = 3k\Omega \quad , R_2 = 1k\Omega \, , voltage\ division\ ratio\ (gain) = G = 0.25 \pm 2\%$ or 0.25±0.005 in decimal values.

Also, resistors R_1 and R_2 are assumed to have the same tolerance rating, i.e. $|T_{R1}| = |T_{R2}| = |T_R|$. In the following we will provide the results of the gain's tolerance analysis using extreme value analysis (EVA) and Monte Carlo simulation.

Using extreme value analysis[7] we derive an expression for the tolerance of the voltage division ratio , T_G, featuring the component tolerance , T_R.

From the circuit of Figure 1 we calculate the gain of the potential divider

$$G = \frac{V_{out}}{V_{in}} = \frac{iR_2}{i(R_1 + R_2)} = \frac{R_2}{R_1 + R_2} \qquad (1.1)$$

Plugging in the given nominal values R_1=3kΩ and R_2=1kΩ, we find G=0.25 as expected. Next, we can derive an expression for the gain tolerance, T_G, in the extreme value limit, by rewriting eq. (4.1a) as follows,

$$G(1 \pm T_G) = \frac{R_2(1 \pm \Delta R_2)}{R_1(1 \pm T_{R_1}) + R_2(1 \pm T_{R_2})} = \frac{R_2(1 \pm T_R)}{R_1(1 \pm T_R) + R_2(1 \pm T_R)}$$

Plugging in the values given, we calculate the following extreme values for the gain, assuming $T_{R1} = T_{R2} = T_R$, tabulated in Table 1.1a below.

Table 1.1

Resistor tolerance T_R	**Combinations**	**G**	**Gain tolerance T_G**
0.01	R_1+T_R; R_2+T_R	0.250	
0.01	R_1+T_R; R_2- T_R	0.246	
0.01	R_1-T_R; R_2+ T_R	0.254	
0.01	R_1-T_R; R_2- T_R	0.250	0.0038 or 1.5%
0.01333	R_1+T_R; R_2+T_R	0.250	
0.01333	R_1+T_R; R_2- T_R	0.245	
0.01333	R_1-T_R; R_2+ T_R	0.255	
0.01333	R_1-T_R; R_2- T_R	0.250	0.005 or 2%
0.02	R_1+T_R; R_2+T_R	0.250	
0.02	R_1+T_R; R_2- T_R	0.243	
0.02	R_1- T_R; R_2+T_R	0.258	
0.02	R_1- T_R; R_2- T_R	0.250	0.0075 or 3%
0.05	R_1+T_R; R_2+T_R	0.250	
0.05	R_1+T_R; R_2- T_R	0.232	
0.05	R_1-T_R; R_2+ T_R	0.269	
0.05	R_1- T_R; R_2- T_R	0.250	0.0188 or 7.5%
0.10	R_1+T_R; R_2+T_R	0.250	
0.10	R_1+T_R; R_2- T_R	0.214	
0.10	R_1-T_R; R_2+ T_R	0.289	
0.10	R_1- T_R; R_2- T_R	0.250	0.0376 or 15%

Looking up at this table, we observe that the gain tolerance increases in an analogous fashion with the resistor tolerance and that for higher values of T_R=1.33%, the circuit fails to meet the specification limit of 2%.

On the other hand, remembering that tolerance and errors are almost the same quantities, we apply the well defined calculus for the error propagation theory[8] here.

Let the gain be given by the relation, $G = f(R_1, R_2) = \frac{R_2}{R_1+R_2}$, then the variance of the function δf (which is similar to the total differential df) according to the error theory is given by,

$$\delta f = \frac{\partial f}{\partial R_1}\delta R_1 + \frac{\partial f}{\partial R_2}\delta R_2 = -\frac{R_2}{(R_1+R_2)^2}\delta R_1 + \frac{R_1}{(R_1+R_2)^2}\delta R_2$$

The standard deviation σ is related to the function variance by: $\sigma = \sqrt{(\delta f)^2}$,

where $(\delta f)^2 = (\frac{\partial f}{\partial R_1}\delta R_1 + \frac{\partial f}{\partial R_2}\delta R_2)^2 = \left(\frac{\partial f}{\partial R_1}\delta R_1\right)^2 + \left(\frac{\partial f}{\partial R_1}\delta R_1\right)^2 + 2\frac{\partial f}{\partial R_1}\delta R_1\frac{\partial f}{\partial R_2}\delta R_2$.

The last term is usually small, so it can be discarded, therefore we have,

$$(\delta f)^2 = \left(\frac{\partial f}{\partial R_1}\delta R_1\right)^2 + \left(\frac{\partial f}{\partial R_1}\delta R_1\right)^2 = (-\frac{R_2}{(R_1+R_2)^2}\delta R_1)^2 + (\frac{R_1}{(R_1+R_2)^2}\delta R_2)^2 .$$

Now the variance of any quantity, for example a resistance R is: $\delta R = R_{nom}T_R$, where T_R is the tolerance related to the quantity R and R_{nom} is its nominal value. Then the previous equation is written, keeping in mind that : $\delta G = G_{nom}T_G$ where $G_{nom} = \frac{R_2}{R_1+R_2} = 0.25$,

$$G_{nom}T_G = \sqrt{(\delta f)^2} = \sqrt{(\frac{R_2}{(R_1+R_2)^2}R_1 T_{R_1})^2 + (\frac{R_1}{(R_1+R_2)^2}R_2 T_2)^2} ,$$

and after simplifications we have,

$$\frac{R_2}{R_1+R_2}T_G = \sqrt{\frac{(R_1R_2)^2}{(R_1+R_2)^4}(T_{R_1}^2 + T_{R_2}^2)}$$

Finally, the gain tolerance is equal to,

$$T_G = \frac{R_1}{R_1+R_2}\sqrt{T_{R_1}^2 + T_{R_2}^2} . \qquad (1.2)$$

Using eq. (1.2), we determined the following values tabulated in table 1.2 for the voltage division ratio tolerance with the potential divider resistors having a tolerance of (i) T_R=±1%, (ii) T_R=±2%, (iii) T_R=±5% and (iv) T_R=±10%, where R_1=3kΩ and R_2=1kΩ with $|T_{R1}|=|T_{R2}|=|T_R|$.

Table 1.2

Resistor tolerance T_R	Gain tolerance T_G
1%	0.35%
2%	0.71%
5%	1.77%
5.66%	2.00%
10%	3.54%

These tolerance ratings meet the specification limit of $T_G=\pm2\%$ up to a resistor tolerance of $T_R\approx5.66\%$, while the extreme value scheme gave a tighter resistor tolerance limit of $T_R\approx1.33\%$.

Following in this section we run a Monte Carlo simulation, in order to determine the effects of the resistor tolerance on yield. The Matlab program code used for this task is listed at the end of this report in Appendix II. A histogram of the voltage division ratio for a resistor tolerance $T_R=5\%$ is shown in Figure 6.1.1 For the rest resistor tolerances cases, the histograms are similar, so in the interests of saving space not all of them were included here.

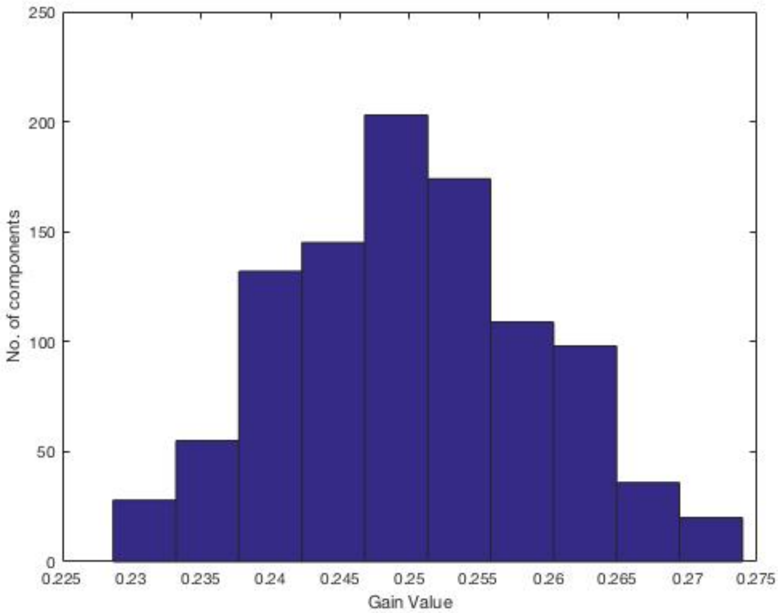

Figure 6.1.1 : Histogram of gain, from a uniform Monte Carlo simulation for T_R=5%.

We have run the simulation for all the resistor tolerances asked. In the table 1.3 below we have tabulated the yield gains for each tolerance value for a run of 1000 circuits and in Figure 6.1.2 below we presented a plot of tolerance vs gain yield.

Table 1.3

Resistor tolerance T_R	**Yield**
0.01	1.000
0.02	0.796
0.05	0.402
0.10	0.225

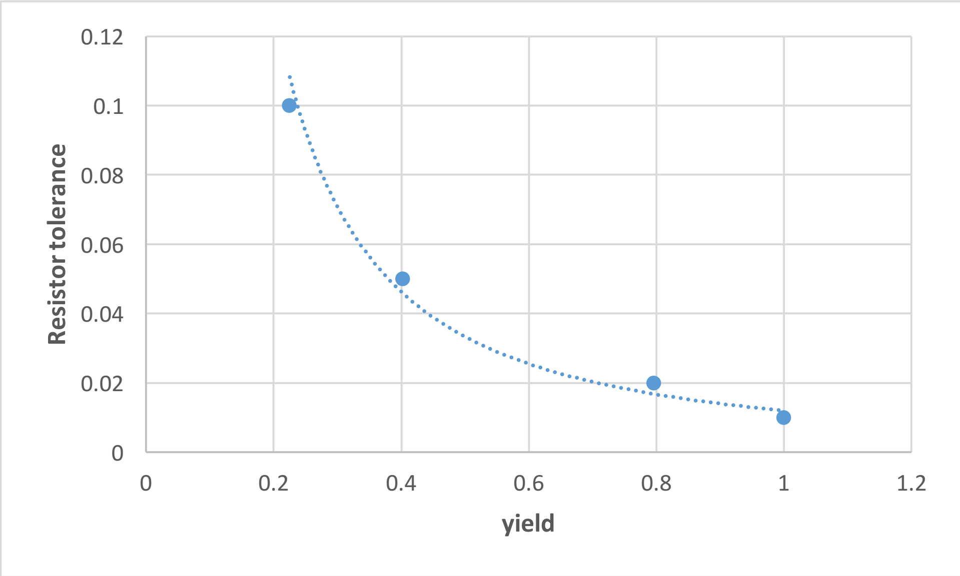

Figure 6.1.2: A plot of resistor tolerance vs yield for the potential divider circuit

According to our calculations, eq. (1.2) will give, $T_G = \frac{R_1}{R_1+R_2}\sqrt{2}\cdot T_R \Rightarrow T_R = T_G\frac{R_1+R_2}{\sqrt{2}\cdot R_1}$ and for T_G=2%, this relation will give a value for the most possible resistor tolerance at T_R=1.9%. On the other hand, using the extreme limits scheme, we found that the entrance value of T_R=1.33% was the higher tested value for the circuit to ensure the specification of 2% (see table 1.1)

Finally, we performed a using a Monte Carlo run of 1,000 circuit using a uniform probability distribution to determine the gain, for the given nominal values R_1=3kΩ and R_2=1kΩ, and a resistor tolerance T_R=2.744% (we have tried many different values T_R to reach the desired tolerance limit of the gain T_G=2%). In the following Figure 6.1.3, the frequency of gain values for this Monte Carlo run it is shown. The curve seems to follow a Gaussian distribution. From this simulation we found G_{mean}=0.2502 with a tolerance T_G≈0.02=2%.

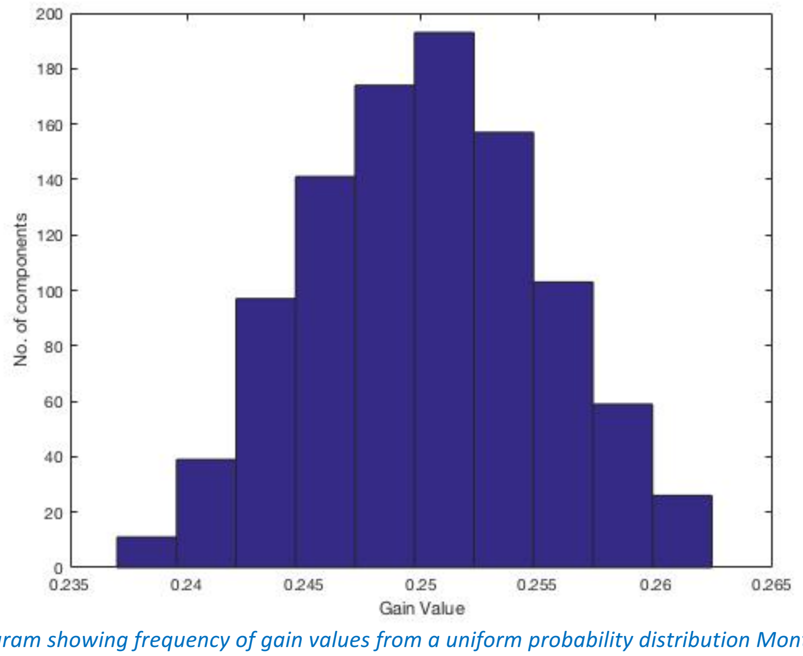

Figure 6.1.3: Histogram showing frequency of gain values from a uniform probability distribution Monte Carlo simulation

The results from this Monte Carlo analysis overestimates the values received by the other two methods of the extreme values analysis and of the statistical analysis, eq. (1.2).

(ii) Low Pass Filter[9]

A low pass filter by definition is a circuit offering passage to signals of low frequency or of frequency below the corner frequency. For the circuit that we are going to analyze (Figure 2) the corner frequency $f_c = 1kHz \pm 5\%, where\ f_c = \frac{1}{2\pi RC}$ and the component values are specified: $R = 1.6k\Omega\ and\ C = 100nF \pm 10\%$.

Using EVA we derive an expression for the tolerance in corner frequency, T_{fc}, in terms of the resistor component tolerance, T_R.
From the circuit of Figure 2 we calculate the gain of the low pass filter,

$$G = \frac{V_{out}}{V_{in}} = \frac{iX_C}{i(R+X_C)} = \frac{X_C}{R+X_C} = \frac{R\frac{1}{j\omega C}}{R+\frac{1}{j\omega C}} = \frac{R}{1+Rj\omega C} = R\frac{1}{1+j\frac{\omega}{\omega_0}} \qquad (2.1)$$

where $\omega_0 = \frac{1}{RC} = \frac{1}{1.6\times10^3\Omega\times100\times10^{-9}F} = 6.25\times10^3 c/sec = 6.25kc/sec$, then the corner frequency is: $f_c = \frac{\omega_0}{2\pi} = \frac{6.25}{2\pi}kHz = 1kHz$. Equation (2.1) has the form of the high-pass transfer function.

Next, we can derive an expression for the corner frequency tolerance, T_{fc}, using the extreme value limit, by manipulating the equation

$$f_c = \frac{\omega_0}{2\pi} = \frac{1}{2\pi RC} = \frac{1}{2\pi}\cdot\frac{1}{R(1\pm T_R)\cdot C(1\pm T_C)} \qquad (2.2)$$

Plugging in the given elements values, we calculate the following extreme values for the corner frequency, posted in the following table 2.1, for the component values R=1.6kΩ and C=100nF, T_C=10%. The specified limit for the tolerance of the corner frequency is T_{fc}=±5%.

Table 2.1

Resistor tolerance T_R	Combinations	Corner frequency f_c [kHz]	Corner frequency tolerance T_{fc}
0.01	R+T_R ; C+T_R	0.895	
0.01	R+T_R ; C- T_C	1.094	
0.01	R-T_R ; C+ T_C	0.913	
0.01	R-T_R ; C- T_C	1.116	11%
0.02	R+T_R ; C+T_C	0.887	
0.02	R+T_R ; C- T_C	1.084	
0.02	R-T_R ; C+T_C	0.923	
0.02	R-T_R ; C- T_C	1.128	12%
0.05	R+T_R ; C+T_C	0.861	
0.05	R+T_R ; C- T_C	1.053	
0.05	R-T_R ; C+ T_C	0.952	
0.05	R-T_R ; C- T_C	1.163	15%
0.10	R+T_R ; C+T_C	0.822	
0.10	R+T_R ; C- T_C	1.005	

0.10	R-T_R ; C+ T_C	1.005	
0.10	R-T_R ; C- T_C	1.228	20%

It is obvious that none of there results meets the specified limit for the tolerance of the corner frequency of T_{fc}=±5%.

In another approach, we apply the trivial procedure for the error propagation, in order to derive an expression for the tolerance of the corner frequency, T_{fc}

Let the corner frequency be given by the relation, $f_c = f(R, C) = \frac{1}{2\pi RC}$, then the variance of the function δf (which is similar to the total differential df) is given by,

$$\delta f = \frac{\partial f}{\partial R}\delta R + \frac{\partial f}{\partial C}\delta C = -\frac{1}{2\pi R^2 C}\delta R - \frac{1}{2\pi R C^2}\delta C$$

The standard deviation σ is related to the function variance by: $\sigma = \sqrt{(\delta f)^2}$,

where $(\delta f)^2 = (\frac{\partial f}{\partial R}\delta R + \frac{\partial f}{\partial C}\delta C)^2 = \left(\frac{\partial f}{\partial R}\delta R\right)^2 + \left(\frac{\partial f}{\partial C}\delta C\right)^2 + 2\frac{\partial f}{\partial R}\delta R\frac{\partial f}{\partial C}\delta C$.

The last term is usually small, so it can be discarded, therefore we are left with,

$$(\delta f)^2 = \left(\frac{\partial f}{\partial R}\delta R\right)^2 + \left(\frac{\partial f}{\partial C}\delta C\right)^2 = (-\frac{1}{2\pi R^2 C}\delta R)^2 + (\frac{1}{2\pi R C^2}\delta C)^2$$
$$= (\frac{1}{2\pi RC})^2\left((\frac{\delta R}{R})^2 + (\frac{\delta C}{C})^2\right) = f_c^2\left((\frac{\delta R}{R})^2 + (\frac{\delta C}{C})^2\right)$$

Now the variance of any quantity, for example a resistance R is equal to: $\delta R = R_{nom}T_R$, where T_R is the tolerance related to the quantity R and R_{nom} is its nominal value. Then the previous equation can be written, keeping in mind that: $\delta f_c = f_{cnom}T_{f_c}$ where $f_{cnom} = \frac{1}{2\pi RC}$

$$f_{cnom}T_{f_c} = \sqrt{(\delta f)^2} = f_c\sqrt{T_R^2 + T_C^2},$$

and after simplifications we have,

$$T_{f_c} = \sqrt{T_R^2 + T_C^2} \qquad (2.3)$$

Using eq. (2.3), we calculated the values tabulated in table 2.2 for the tolerance in the corner frequency T_{fc}, with a tolerance in the resistor of (i) T_R=±1%, (ii) T_R=±2%, (iii) T_R=±5% and (iv) T_R=±10%, and a tolerance in the capacitor of T_C=±10%.

Table 2.2

Resistor tolerance T_R	tolerance in the corner frequency T_{fc}
1%	0.100 or 10%
2%	0.102 or 10%
5%	0.112 or 11%
10%	0.141 or 14%

Again, none of there results meets the specified limit of the corner frequency tolerance of T_{fc}=±5%!

In this section we run a Gaussian distribution Monte Carlo simulation, to determine the effects of the resistor tolerance on yield. A histogram of the voltage division ratio for a resistor tolerance T_R=10% is shown in Figure 6.2.1. The specified tolerance in the corner frequency is given at T_{fc}=±5%. The tolerance in the corner frequency for this run was estimated at T_{fc}=1.3%. For the rest resistor tolerances cases, the histograms are similar, so in the interests of saving space we didn't include all of them here.

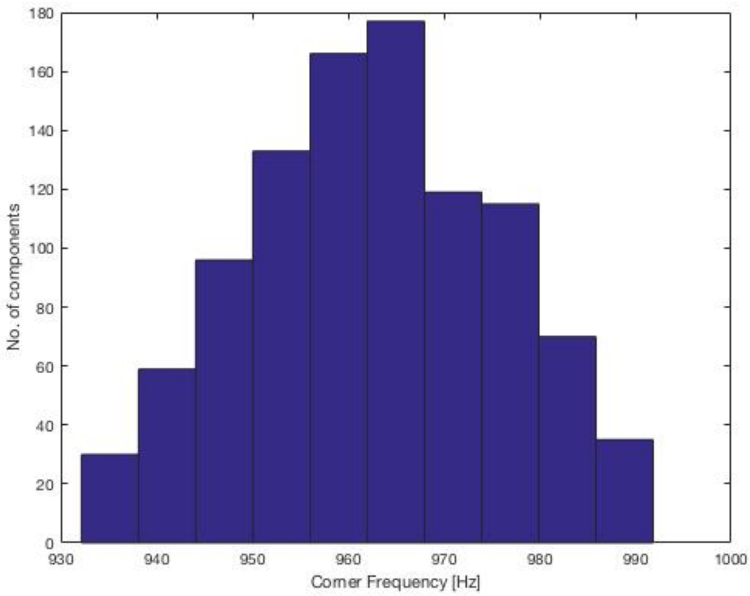

Figure 6.2.1: Histogram of the corner frequency from a Gaussian distribution Monte Carlo simulation for a tolerance in the resistor of T_R=10%.

We have run the simulation for all the resistor tolerances asked (and more). In table 2.3 below we have tabulated the yield gains for each tolerance value for a run of 1000 circuits and in figure 6.2.2 below, we presented a plot of tolerance in the resistor .vs. yield gain.

Table 2.3

Resistor tolerance T_R	Yield
1%	1.000
2%	1.000
5%	1.000
6%	0.998
8%	0.962
9%	0.936
10%	0.908
12%	0.828
15%	0.708

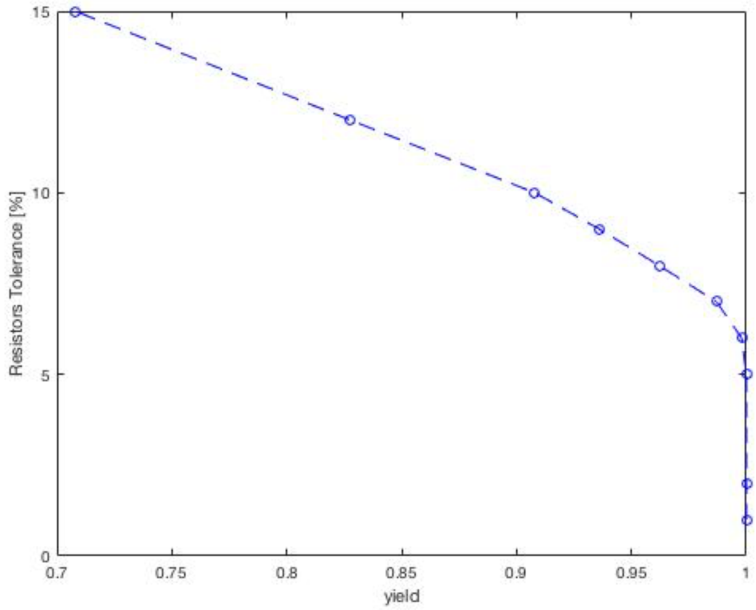

Figure 6.2.2: A plot of tolerance vs yield gain for the corner frequency

Reorganizing eq. (2.3), we can calculate the capacitor tolerance: $T_C = \sqrt{T_{fc}^2 - T_R^2}$; plugging in the values T_{fc}=5% and T_R=±2%, we get T_C=4.6%.

In addition, we run a 1,000-sample uniform distribution Monte Carlo simulation to determine the capacitor tolerance; for f_c=1kHz ±5% and R=1.6kΩ, this simulation gave these result: for the resistor tolerance value of T_R=±2%, the capacitor tolerance was found equal to T_C=3% with a mean capacitance value of C=99.8nF, in other words, our expression from the first part of the EVA, overestimates considerably the largest possible capacitor tolerance value T_C over 50%.
We run a Gaussian distribution Monte Carlo simulation of 1,000 circuits to test the results from the previous Part. This simulation gave for T_R=2%, a mean capacitance value of C=98.8nF with a capacitor tolerance equal to T_C≈0.3% which is one order of magnitude lower than the one found from the Gaussian distribution Monte Carlo simulation run in the previous Part with same parameters.

(iii) 2-bit R-2R ladder DAC[11]

Resistor ladder networks provide a simple, inexpensive way to perform digital to analog conversion (DAC). The superiority of the 2-bit R-2R ladder type in comparison with all the other DAC types is its inherent accuracy and ease of manufacturing. As shown in Figure 6.3.1 the R-2R ladder DAC converts a digital code $d_1 d_0$ into an analogue voltage V_{out}. Both data lines are connected to a microprocessor that operates with defined 0 V to 5 V. The nominal value of the resistors is: $R_1 = R_3 = R_4 = 2R = 20k\Omega \; and \; R_2 = R = 10k\Omega$.

In the following we will derive an expression for the output voltage, V_{out}, as a function of the digital code, d_1, d_0.

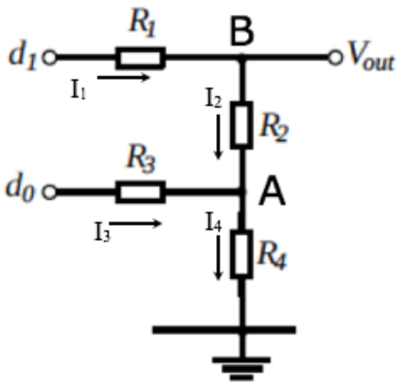

Figure 6.3.1: 2-bit R-2R ladder DAC

We apply the Kirchhoff's 1st law at nodes A and B (Nodal Analysis) as shown in Figure 6.3.1. If d_0 and d_1 are the potential values of the data lines d_0 and d_1, correspondingly, then we have,

Node A: $I_3 + I_2 = I_4 \quad \Rightarrow \quad \frac{d_0 - V_A}{R_3} + \frac{V_B - V_A}{R_2} = \frac{V_A}{R_4}$ (3.1)

$\Rightarrow \quad d_0 + 2V_B - 4V_A = 0$ (3.2)

where we have used the given resistors relation: R_1=R_3=R_4=2R and R2=R. Similarly,

Node B: $I_1 = I_2 \quad \Rightarrow \quad \frac{d_1 - V_B}{R_1} = \frac{V_B - V_A}{R_2}$ (3.3)

$\Rightarrow \quad 2V_A = 3V_B - d_1$ (3.4)

Eliminating V_A between eqs. (3.2) and (3.4), we receive,

$$V_{out} \equiv V_B = \frac{d_0}{4} + \frac{d_1}{2} \quad (3.5)$$

As being defined at the beginning, data line d_0 (LSB) operates at the voltage level 0V and data line d_1 (MSB) at the voltage level V_{ref}=5V. If we associate the bit number d_i with the voltage $d_i = V_{ref} a_i$, then eq. (3.5) can be put in the form,

$$V_{out} \equiv V_B = V_{ref}\left(\frac{a_0}{4} + \frac{a_1}{2}\right) \qquad (3.6)$$

where (a_0, a_1) is the digital input (for the 2-bit ladder DAC).

The full scale output voltage is $V_{out}[FS] = V_{ref}\left(\frac{2^2-1}{2^2}\right) = 5V\left(\frac{2^2-1}{2^2}\right) = 3.75V$ and the resolution $1LSB = \frac{V_{ref}}{2^2} = \frac{5V}{4} = 1.25V$.

Now for the output voltage taking in consideration that the resistors are exposed to a variation due to a given tolerance ($R_i \rightarrow R_i(1 \pm T_R)$) we have to eliminate the relation between V_{out} and R_1,...R_4, so we start out from eqs. (3.1) and (3.3),

$$\frac{d_0 - V_A}{R_3} + \frac{V_B - V_A}{R_2} = \frac{V_A}{R_4}$$
$$\frac{d_1 - V_B}{R_1} = \frac{V_B - V_A}{R_2}$$

and by eliminating V_A between these equations, finally we receive,

$$V_{out} \equiv V_B = \frac{d_1\left(\frac{1}{R_1}+\frac{R_2}{R_1R_3}+\frac{R_2}{R_1R_4}\right)+d_0\frac{1}{R_3}}{\frac{1}{R_1}+\frac{1}{R_3}+\frac{1}{R_4}+\frac{R_2}{R_1R_3}+\frac{R_2}{R_1R_4}} \qquad (3.7)$$

With the substitution $R_1=R_3=R_4=2R$ and $R_2=R$, this equation reproduces eq. (3.5). In the above expression it is understandable that each resistance value should be replaced by $R_i \rightarrow R_i(1 \pm T_{Ri})$, which it's been done in the Monte Carlo runs of the following part.

In the following Figure 6.3.2 we present a histogram of the mean values the output voltage of a DAC as a function of the input code with (i) ideal resistors (T_{Ri}=0%) and (ii) a run of 1,000 DACs with a uniform distribution Monte Carlo simulation for T_{Ri}=10%. The output voltage tolerance was found to be close to 5% with increasing trends going to lower input codes.

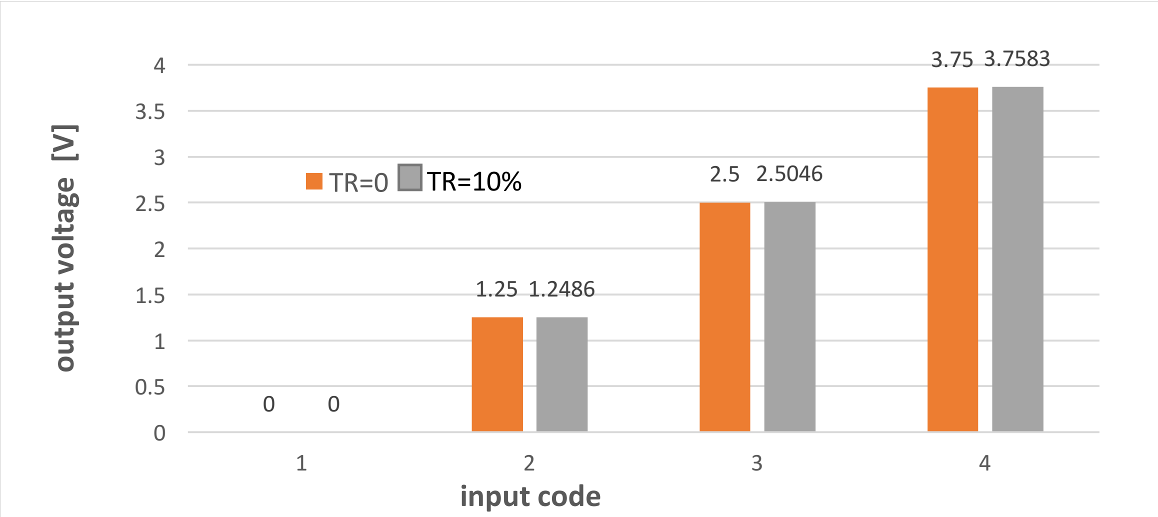

Figure 6.3.2: Histogram of the output voltage from a run of 1,000 2-bit R-2R ladder DACs versus the input code (i) with ideal resistors (orange color), (ii) with a resistor tolerance of T_R=10% (gray color).

The input code convention we used is given in the following table (table 3.1).

Table 3.1

Input Code (decimal)	Input Code (binary)	
	a_1	a_0
1	0	0
2	0	1
3	1	0
4	1	1

Following a Monte Carlo Simulation for the relationship between the Gain error (Appendix I) and the resistor's tolerance value. In order to get of the Gain error at full-scale, we run 1,000 DACs with a uniform distribution Monte Carlo simulation with the following resistor tolerance values: (i) T_{Ri}=±1%, (ii) T_{Ri} =±2%, (iii) T_{Ri} =±5% and (iv) T_{Ri} =±10%. We use the Gain error at full-scale. The full scale output voltage V_{out}[FS] and the output voltage for the LSB input code, 1LSB, have been calculated in a previous part, specifically we found, $V_{out}[FS] = 3.75V$ and $1LSB = 1.25V$.

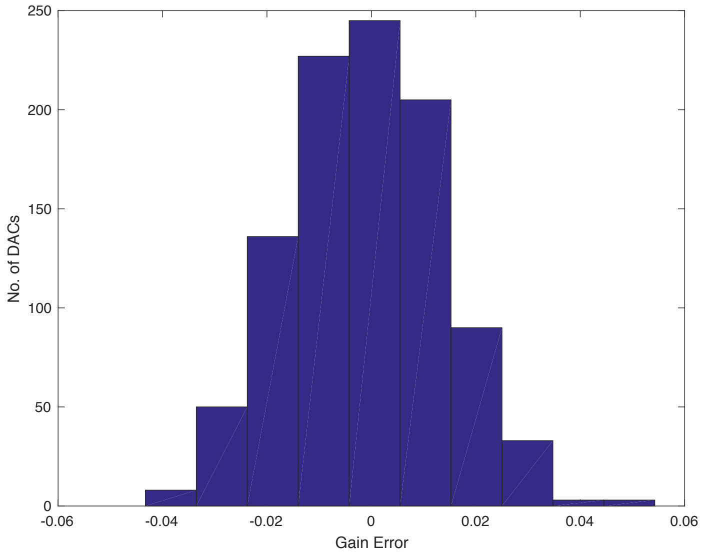

Figure 6.3.3: Histogram of the Gain error of a 2-bit R-2R ladder DAC for T_R=1%

The above histogram of Gain error of a uniform distribution Monte Carlo simulation of 1,000 DACs with a resistor tolerance value of T_R=±1%. The rest of resistor tolerances gave similar histograms, therefore for saving space we didn't include all of them here. In the table below (table 3.2) we have tabulated the maximum and minimum values of Gain error for each resistor tolerance value, plus their mean values.

Table 3.2

Tolerance T_R	Min Gain error value [V]	Max Gain error value [V]	Gain error mean value [V]
1%	-0.0433	0.0543	-9.2e-4
2%	-0.0821	0.0909	-0.0020
5%	-0.2297	0.2082	0.0012
10%	-0.4826	0.4571	-0.0038

Below using Monte Carlo simulation we calculated the DNL (Appendix I), running 1,000 DACs with uniform resistor distribution with tolerances: (i) T_{Ri}=±1%, (ii) T_{Ri} =±2%, (iii) T_{Ri} =±5% and (iv) T_{Ri} =±10%. All the DAC outputs have been evaluated and the maximum (i.e. the worst) value is been used.

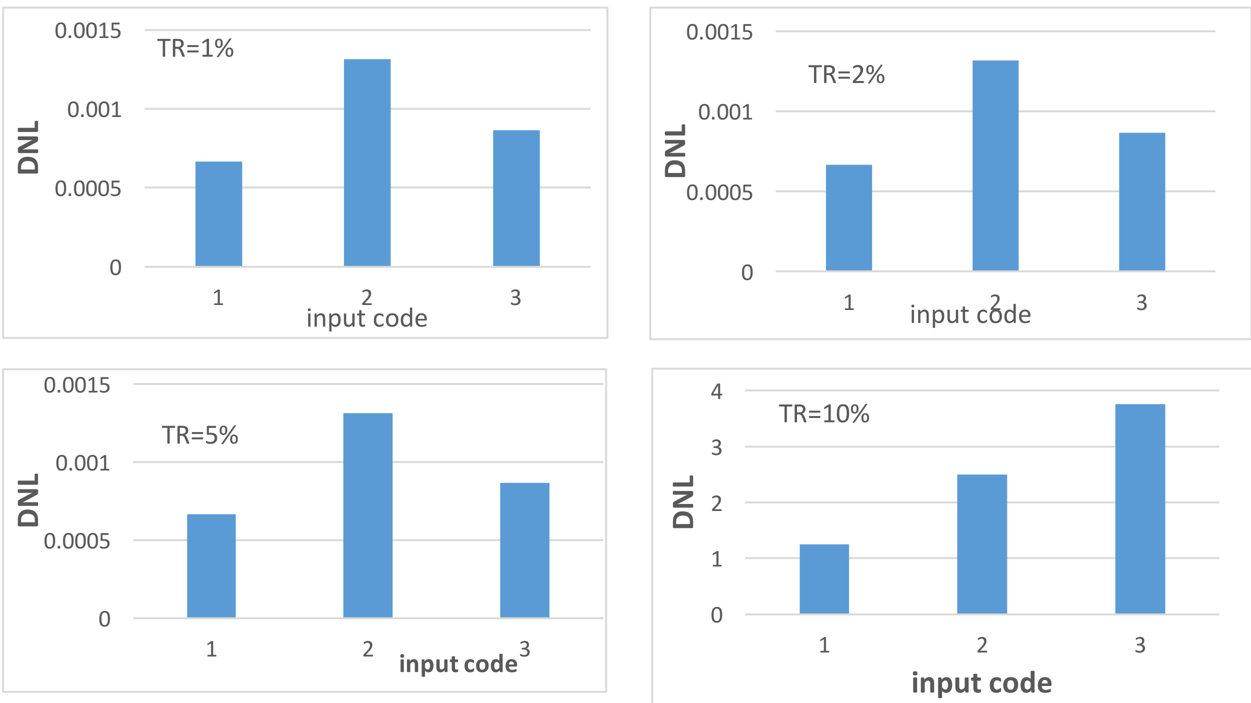

Figure 6.3.4: Histograms of the DNL for the resistor tolerances posted

We have posted in Figure 6.3.4 above, the histograms of DNL from a uniform distribution Monte Carlo simulation of 1,000 DACs with the following resistor tolerances: (i) T_{Ri}=±1%, (ii) T_{Ri} =±2%, (iii) T_{Ri} =±5% and (iv) T_{Ri} =±10%. The horizontal scale actually doesn't correspond to the "input code", but to the differences of consecutive input codes. In the table below (table 3.3) we have tabulated the maximum and minimum values of DNL for each resistor tolerance value we tested.

Table 3.3

Tolerance T_R	DNL min	DNL max
1%	2.6e-4	5.9e-4
2%	2.3e-4	6.5e-4
5%	2.2e-5	0.0016
10%	6.4e-4	0.0047

We have plotted in Figure 6.3.5 below, the maximum values of Gain error and the maximum values of DNL, from a uniform distribution Monte Carlo simulation of 1,000 DACs as a function of the resistor tolerances.

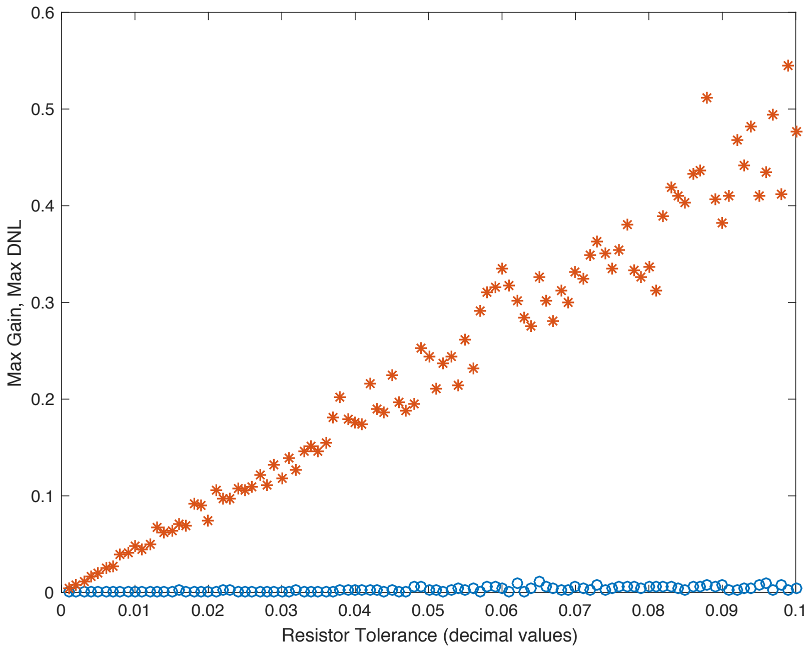

Figure 6.3.5: A graph of the maximum values of Gain error (orange stars) and the maximum values of DNL (blue open circles) from a run of 1,000 DACs versus the resistor tolerance obeying a uniform distribution

Since the data we plotted differ by a scale of 10, we replotted alone the maximum values of DNL in Figure 6.3.6 to show more details, in order to gain more insights of these results.

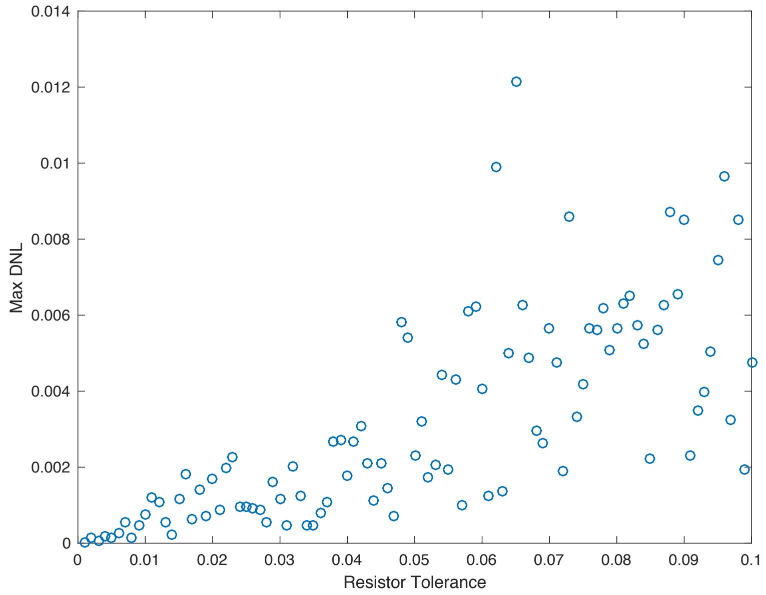

Figure 6.3.6: A graph of the maximum values of DNL from a run of 1,000 DACs versus the resistor tolerance obeying a uniform distribution

In addition, we run the code to calculate the maximum value of resistor tolerance that ensures DNL < 1/2LSB and Gain error < 0.1LSB. Actually, for the condition "DNL < 1/2LSB" to be valid, we calculated a resistor tolerance threshold at T_R =0.8142%. On the other hand, for the condition "Gain error < 0.1LSB" to be valid, we calculated a resistor tolerance threshold at T_R =0.0280%. Therefore, the second limit, T_R =0.0280%, will fulfill both criteria.

Conclusions

We have contacted a thorough investigation to our strength of some critical and novel problems as well as new concepts & methods in electronics & in system engineering, in general, with which this assignment is dealing with. Our findings were interpreted in the individual sections to the best of our knowledge and understanding.

In this assignment we dealt with the problems of potential divider circuits, circuits of low pass filters and 2-bit R-2R ladder DACs where we applied the principles of extreme value analysis and the Monte Carlo analysis on the propagation of component tolerances to our results and the restrictions in order to meet the specifications of a product. Furthermore, we investigated the yield of the circuits tested on account of the component tolerances. We used the Monte Carlo simulation tool to check further our results. There were mismatches of our results from the different methods produced, which were accounted on the running principle of the individual methods.

References

[1] Foster, Martin, Mr. *Component Tolerance Analysis Using Matlab*. Working paper. N.p.: n.p., n.d. Print.

[2] Bonaventure, Cletus K., Capt. "APPLICATIONS OF EXTREME VALUE THEORY IN THE RELIABILITY ANALYSIS OF NON-ELECTRONIC COMPONENTS." Thesis. Air University, US Air Forces, 1960. Web. 12 May 2016. <http://www.dtic.mil/dtic/tr/fulltext/u2/675545.pdf>.

[3] Amelin, Mikael. *On Monte Carlo Simulation and Analysis of Electricity Markets*. Thesis. KTH University, 2004. Stockholm: Universitetsservice US-AB, 2004. Web. 12 May 2016. <https://www.diva-portal.org/smash/get/diva2:8243/FULLTEXT01.pdf>.

[4] *Matlab*. Vers. R2015b. Natick, MA: MathWorks, 2015. Computer software.

[5] Shreider, Yu. A. *Method of Statistical Testing: Monte Carlo Method*. 1st ed. N.p.: Elsevier, 1964. Print.

[6] "Voltage Divider." *Http://hyperphysics.phy-astr.gsu.edu*. N.p., n.d. Web. 15 May 2016. <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/voldiv.html>.

[7] B eirlant, Jan, Jozef L. Teugels, and Petra Vynckier. *Practical Analysis of Extreme Values*. 1st ed. Leuven (Belgium): Leuven UP, 1996. Print.

[8] Coles, Stuart. *An Introduction to Statistical Modeling of Extreme Values*. 3rd ed. Bristol: Springer-Verlaf London Limited, 2004. Print.

[9] "Low-pass Filters." Web. 16 May 2016. <http://www.allaboutcircuits.com/textbook/alternating-current/chpt-8/low-pass-filters/>.

[10] Hunt, Brian R., Kevin R. Coombes, Ronald J. Lipsman, and Jonathan M. Rosenberg. *A Guide to MATLAB®: For Beginners and Experienced Users ; Updated for MTLAB® 7 and Simulink® 6*. Cambridge: Cambridge U, 2007. Print.

[11] Horowitz, Paul. *The Art of Electronics*. 3rd ed. N.p.: n.p., n.d. Print.

Appendices

Appendix I – Formulae

- Yield as a measure in manufacturing process is defined as:

$$yield = \frac{No. of\ products\ meeting\ specification}{Total\ no. of\ products\ manufactured}$$

- Gaussian (normal) probability Distribution is given by:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2\right)$$

- The DAC's output voltage as a function of the input digital code:

$$V_{out} = V_{ref}\left(\frac{a_1}{2} + \frac{a_0}{4}\right)$$

- Full scale Gain error for a 2-bit R-2R ladder DAC:

$$Gain\ error\ (LSB) = \frac{V_{out}[d_1 d_0 = 11] - V_{out}[FS]}{1LSB}$$

- Differential non-linearity (DNL):

$$DNL = max\left|\frac{V_{out}[i+1] - V_{out}[i] - 1LSB}{1LSB}\right|$$

Appendix II – Matlab Program Code

Uniform Distribution:

```
% define nominal and tolerance parameters
Rnom=1e3;
dR=0.05;
%loop to generate 1000 resistors
for ind=1:1000
% determine individual component value
R(ind)=Rnom*(1+2*(rand-0.5)*dR);
end
%plot a histogram
hist(R);
```

Gaussian (normal) Probability Distribution:

```
% define nominal and tolerance parameters
Rnom=1e3;
dR=0.05;
%loop to generate 1000 resistors
for ind=1:1000
% determine individual component value
R(ind)=Rnom*(1+dR/3*randn);
end
%plot a histogram
hist(R);
```

Potential Divider:

```
% A2  The Matlab program code for a uniform probability distribution Monte
Carlo simulation
% to determine the gain calculation of a potential divider circuit
% define nominal and tolerance parameters
% R1=3k? and R2=1k?.
Rnom1=3e3;
Rnom2=1e3;
dR=0.01;

%loop to generate 1000 resistor pairs R1, R2
for ind=1:1000
% determine individual component value
G(ind)=Rnom2*(1+2*(rand-0.5)*dR)/( Rnom1*(1+2*(rand-0.5)*dR)+
Rnom2*(1+2*(rand-0.5)*dR));
end

%plot a histogram
hist(G);

Gnom=Rnom2/(Rnom1+ Rnom2);
Gmean = mean(G);
% standard deviation
s = std(G);
% tolerance TG
TG=s/Gnom;


% A2a ? Supplement Matlab program code of routine A2
```

```
% to determine the yield of a potential divider circuit
% define tolerance parameters
dGnom=0.02;

%loop to determine the yield (how many circuits meet the specification)
n=0;
for ind=1:1000
% determine how many meet the specification
dG= abs((G(ind)-Gnom)/Gnom);
if (dG <= dGnom) n=n+1;
end
end

yield=n/1000;
```

Low pass filter:

```
% A3 ? The Matlab program code for a Gaussian distribution Monte Carlo
simulation
% to determine the corner frequency fc=1kHz ?5% where fc=1/(2? RC) of a
potential divider
% define nominal and tolerance parameters
% component values R=1.6k? and C=100nF ?10%.
Rnom=1.6e3;
Cnom=100e-9;
dR=0.01;
dC=0.1;

%loop to generate 1000 resistor pairs R1, R2
ntot=1000;
for ind=1:ntot
% determine individual component value
            f(ind)=1/( 2*pi*(Rnom*(1+dR/3*randn)*Cnom*(1+dC/3*randn)));
end

%plot a histogram
hist(f);

fnom=1/(2*pi*Rnom*Cnom);
fmean = mean(f);
% standard deviation
s = std(f);
% tolerance Tfc
Tfc=s/fnom;



% A3a ? Supplement Matlab program code of routine A3
% to determine the yield of a potential divider circuit
% define tolerance parameters
dfnom=0.05;

%loop to determine the yield (how many circuits meet the specification)
n=0;
for ind=1:ntot
% determine how many meet the specification
df= abs((f(ind)-fnom)/fnom);
if (df <= dfnom) n=n+1;
end
end
```

```
yield=n/ntot;
```

2-bit R-2R ladder DAC:

```
% A4 – The Matlab program code for a Gaussian distribution Monte Carlo simulation
% to calculate the output voltage of a DAC for various input codes with uniform distribution
% of resistors of various tolerances
% define nominal and tolerance parameters
R1=20e3;
R2=10e3;
R3=20e3;
R4=20e3;
Vref=5;
LSB=1.25;
LSBhalf=LSB/2;
% (a0, a1) is the digital input

dR=0.0;

%Vout2(1)=0;
for ind2=1:1500
  dR=dR+0.001;
  for ind1=1:3
  a0=mod(ind1,2);
  a1=round((ind1-a0)/2);
   if a1 == 2;
   a1=0;
   end
d1=Vref*a1;
d0=Vref*a0;
Voutnom=(d1*(1/R1 + R2/(R1*R3) + R2/(R1*R4) ) + d0/R3 )/(1/R1 +1/R3 + 1/R4 + R2/(R1*R3) + R2/(R1*R4) );
  for ind=1:1000
  % determine individual component value
   Vout(ind)=(d1*(1/(R1*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-
0.5)*dR)*R3*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-
0.5)*dR)))+d0/(R3*(1+2*(rand-0.5)*dR)))/(1/(R1*(1+2*(rand-0.5)*dR))+1/(R3*(1+2*(rand-
0.5)*dR))+1/(R4*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R3*(1+2*(rand-
0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-0.5)*dR)));
  end
  Voutmean = mean(Vout);
% standard deviation
s = std(Vout);
% tolerance TVout
  TVout=s/Voutnom;
  Vout2(ind1)=Voutmean;
  end

%plot a histogram
%hist(Vout2);

A(1)=0;
for i=1:3
  A(i+1)=Vout2(i);
end
```

```
for j=1:3
   DNL(j)=abs((A(j+1)-A(j)-LSB)/LSB);
end

    maxxDNL=max(DNL);
    DNLmax(ind2)=maxxDNL;
    DNLmin=min(DNL);
    if maxxDNL > LSBhalf
      return % Stop the ind2 for loop
    else
    savedR=dR;
    end
end
%plot a histogram
%hist(DNL);
```

% A5 – The Matlab program code for a Gaussian distribution Monte Carlo simulation
% to calculate the Gain error at full-scale,
% for various input codes with a uniform distribution of resistors of various tolerances
% define nominal and tolerance parameters

```
R1=20e3;
R2=10e3;
R3=20e3;
R4=20e3;
dR=0.0;
Vref=5;

% start up the counter for the gain yield loop
n=0;
%for ind1=1:1
% (a0, a1) is the digital input
%   a0=mod(ind1,2);
%   a1=round((ind1-a0)/2);
%     if a1 == 2;
%     a1=0;
%     end
for ind2=1:1500
   dR=dR+0.001;
a0=1;
a1=1;
d1=Vref*a1;
d0=Vref*a0;
Voutnom=(d1*(1/R1 + R2/(R1*R3) + R2/(R1*R4) ) + d0/R3 )/(1/R1 +1/R3 + 1/R4 + R2/(R1*R3) + R2/(R1*R4) );
VoutFS=3.75;
LSB=1.25;
LSBtenth=0.1*LSB;
Gainnom=(Voutnom-VoutFS)/LSB;
for ind=1:1000
  % determine individual component value
  Voutind=(d1*(1/(R1*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-
0.5)*dR)*R3*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-
0.5)*dR)))+d0/(R3*(1+2*(rand-0.5)*dR)))/(1/(R1*(1+2*(rand-0.5)*dR))+1/(R3*(1+2*(rand-
```

```
0.5)*dR))+1/(R4*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R3*(1+2*(rand-
0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-0.5)*dR)));
Gain(ind)=(Voutind-VoutFS)/LSB;
end
  Gainmean = mean(Gain);
% standard deviation
s = std(Gain);
% tolerance TGain
  TGain=s/Gainnom;
%  Vout2(ind1)=Voutmean;
% end

%plot a histogram
hist(Gain);
%find the maximum and minimum values of Gain error for each resistor tolerance value
    maxxGain=max(Gain);
 maxGain(ind2)=maxxGain;
 minGain=min(Gain);
    if maxxGain > LSBtenth
      return % Stop the ind2 for loop
    else
    savedR=dR;
    saveindex2=ind2;
    end
end
```

% A6 – The Matlab program code for a Gaussian distribution Monte Carlo simulation
% to calculate the DNL for various input codes with uniform distribution of resistors and
% various tolerances.
% define nominal and tolerance parameters

```
R1=20e3;
R2=10e3;
R3=20e3;
R4=20e3;
dR=0.01;
Vref=5;

% start up the counter for the gain yield loop
n=0;
%for ind1=1:1
% (a0, a1) is the digital input
a0=1;
a1=1;
%    a0=mod(ind1,2);
%    a1=round((ind1-a0)/2);
%      if a1 == 2;
%      a1=0;
%      end
d1=Vref*a1;
d0=Vref*a0;
Voutnom=(d1*(1/R1 + R2/(R1*R3) + R2/(R1*R4) ) + d0/R3 )/(1/R1 +1/R3 + 1/R4 + R2/(R1*R3) + R2/(R1*R4) );
VoutFS=3.75;
LSB=1.25;
for ind=1:1000
```

```
  % determine individual component value
  Voutind=(d1*(1/(R1*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-
0.5)*dR)*R3*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-
0.5)*dR)))+d0/(R3*(1+2*(rand-0.5)*dR)))/(1/(R1*(1+2*(rand-0.5)*dR))+1/(R3*(1+2*(rand-
0.5)*dR))+1/(R4*(1+2*(rand-0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R3*(1+2*(rand-
0.5)*dR))+(R2*(1+2*(rand-0.5)*dR))/(R1*(1+2*(rand-0.5)*dR)*R4*(1+2*(rand-0.5)*dR)));
Vout(ind)=(Voutind-VoutFS)/LSB;
  end
  Voutmean = mean(Vout);
% standard deviation
s = std(Vout);
% tolerance TVout
  TVout=s/Voutnom;
%  Vout2(ind1)=Voutmean;
% end

%plot a histogram
hist(Vout);
%find the maximum and minimum values of Gain error for each resistor tolerance value
maxVout=max(Vout);
minVout=min(Vout);
```