Statistical Analysis and Yield Calculations

(Part 2)

Dr. José Ernesto Rayas-Sánchez

1

Outline

- Definition of Monte-Carlo analysis
- Formulating Monte-Carlo analysis
- Generating random outcomes
- Practical Monte-Carlo analysis
- Estimating the yield
- Practical yield predictions
- Estimating the required number of outcomes

Dr. J. E. Rayas-Sánchez

2

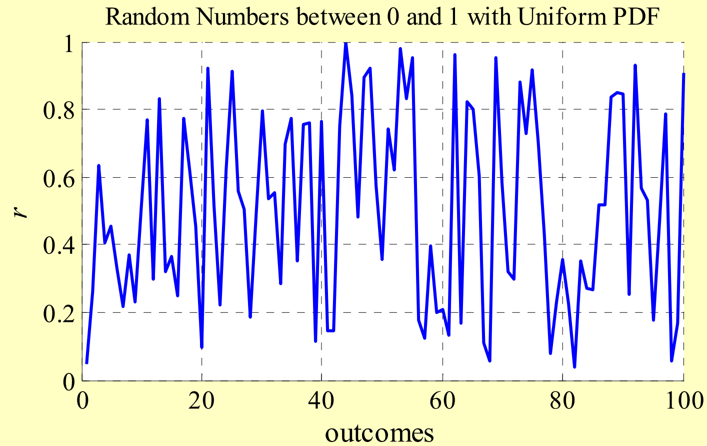
Monte-Carlo Analysis of Circuits

- It consists of the repeated simulation of topologically identical networks with different random selections of element values according to their specific tolerances and probability distribution functions

Formulating Monte-Carlo Analysis

- $\boldsymbol{y} \in \Re^t$ contains the t parameters of the circuit that are subject to statistical fluctuations
- Circuit responses are $\boldsymbol{R}(\boldsymbol{y}) \in \Re^r$, where r is the number of responses of interest
- Random outcomes around a nominal design $\boldsymbol{y}^{\text{nom}}$ are calculated as $\boldsymbol{y}_j = \boldsymbol{y}^{\text{nom}} + \Delta\boldsymbol{y}_j$, for $j = 1,\ldots N,$ where N is the number of outcomes, and $\Delta\boldsymbol{y}_j \in \Re^t$ is a random variation for the j-th outcome
- $\Delta\boldsymbol{y}_j$ are calculated according to the tolerance τ_i and the probability distribution function p_i of the i-th element y_i
- The Monte Carlo analysis around $\boldsymbol{y}^{\text{nom}}$ consists of simulating $\boldsymbol{R}(\boldsymbol{y}_j)$ for $j = 1, \ldots N$

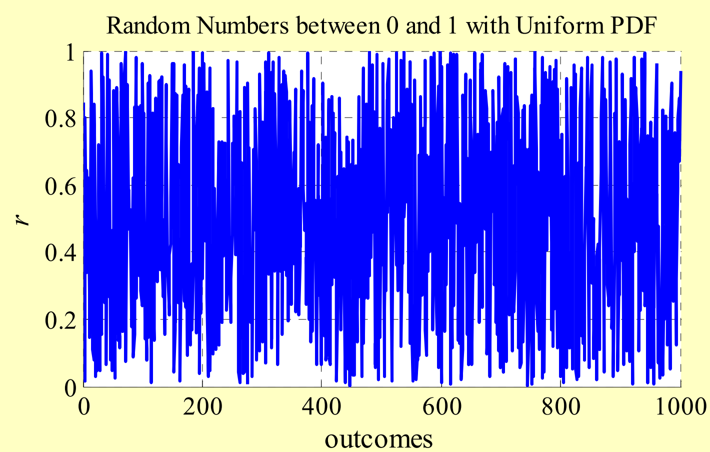
Generating Random Outcomes – Uniform PDF



$$\boldsymbol{y}, \boldsymbol{y}^{\text{nom}}, \Delta\boldsymbol{y}_j \in \Re^t \qquad \boldsymbol{y}_j = \boldsymbol{y}^{\text{nom}} + \Delta\boldsymbol{y}_j \text{, for } j = 1 \ldots N$$

$$(\Delta y_i)_j = y_i \tau_i (2r_j - 1) \text{, for } i = 1 \ldots t$$

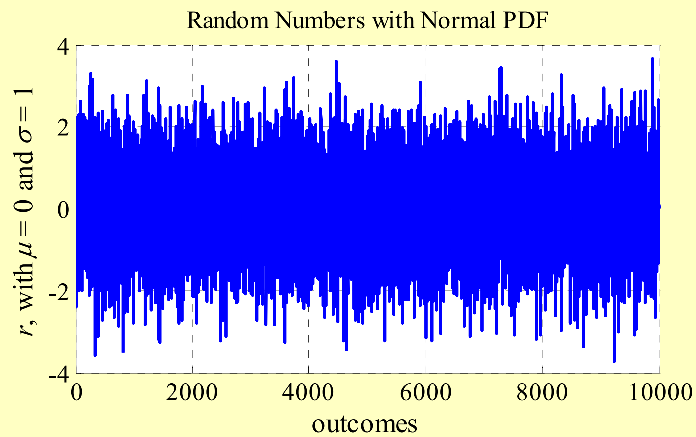
Generating Random Outcomes – Uniform PDF



$$\boldsymbol{y}, \boldsymbol{y}^{\text{nom}}, \Delta\boldsymbol{y}_j \in \Re^t \qquad \boldsymbol{y}_j = \boldsymbol{y}^{\text{nom}} + \Delta\boldsymbol{y}_j \text{, for } j = 1 \ldots N$$

$$(\Delta y_i)_j = y_i \tau_i (2r_j - 1) \text{, for } i = 1 \ldots t$$

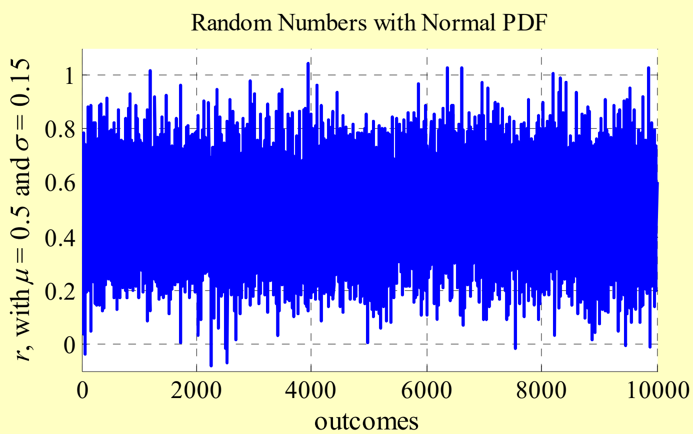
Random Numbers with Normal PDF



μ : average
σ : standard deviation
σ^2 : variance

Dr. J. E. Rayas-Sánchez 7

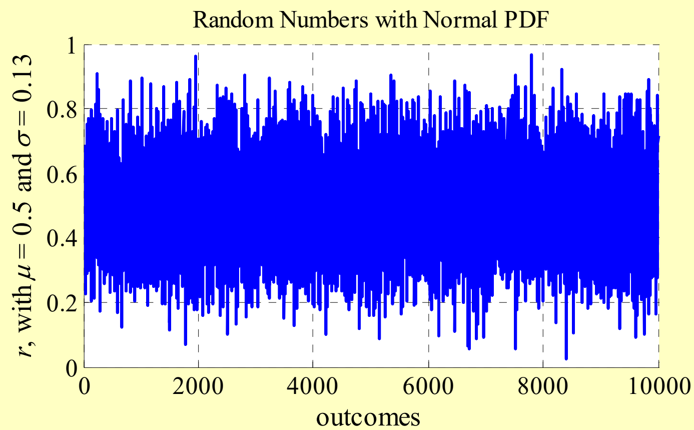
Random Numbers with Normal PDF (cont.)



μ : average
σ : standard deviation
σ^2 : variance

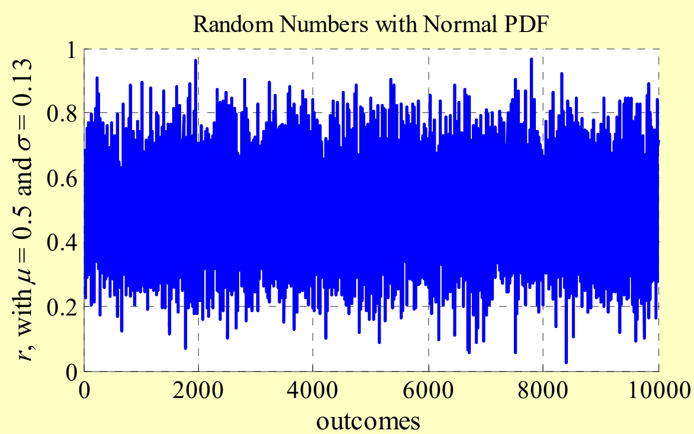
Dr. J. E. Rayas-Sánchez 8

Random Numbers with Normal PDF (cont.)



μ : average
σ : standard deviation
σ^2 : variance

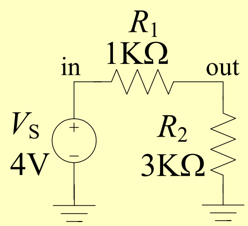
Generating Random Outcomes – Normal PDF



$\boldsymbol{y}, \boldsymbol{y}^{\text{nom}}, \Delta\boldsymbol{y}_j \in \Re^t$ $\qquad$ $\boldsymbol{y}_j = \boldsymbol{y}^{\text{nom}} + \Delta\boldsymbol{y}_j$, for $j = 1 \ldots N$

$$(\Delta y_i)_j = y_i \tau_i (2r_j - 1) \text{, for } i = 1 \ldots t$$

Monte-Carlo Analysis – Example



$$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$$

$$\boldsymbol{y}^{nom} = [4V \quad 1K\Omega \quad 3K\Omega]^T$$

$$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$$

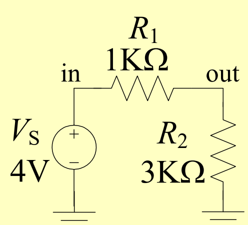
$$\boldsymbol{R}(\boldsymbol{y}) = [V_{in} \quad V_{out} \quad I_{V_S}]^T$$

$$\boldsymbol{R}(\boldsymbol{y}^{nom}) = [4V \quad 3V \quad 1mA]^T$$

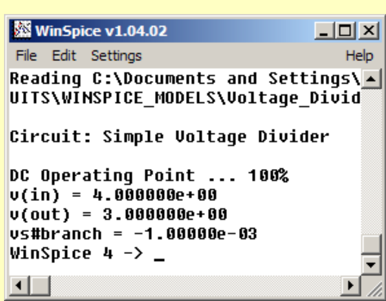
Perform a Monte-Carlo analysis for V_{out} around $\boldsymbol{y}^{nom}$, where

$$V_{out}^{nom} = V_{out}(\boldsymbol{y}^{nom}) = 3V$$

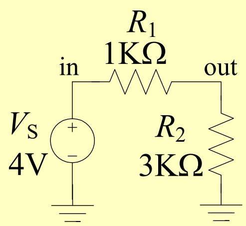
Monte-Carlo Analysis – Example (cont.)



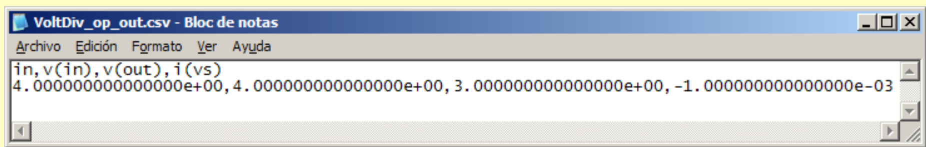

```
Simple Voltage Divider
*-----------------------
Vs   in   0   DC 4V
R1   in   out 1K
R2   out  0   3K

.control
op
print v(in) v(out) i(Vs)
.endc
.end
```



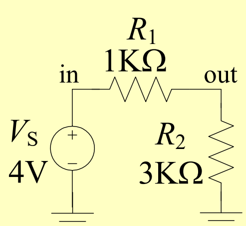
Monte-Carlo Analysis – Example (cont.)




```
Simple Voltage Divider
*-----------------------
Vs   in  0   DC 4V
R1   in  out 1K
R2   out 0   3K

.control
op
write VoltDiv_op_out.csv v(in) v(out) i(Vs)
.endc
.end
```

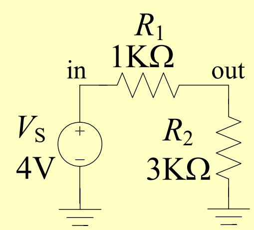


Monte-Carlo Analysis – Example (cont.)



After generating a parameterized Matlab file (`VoltDiv_SPICE1`) to drive the SPICE simulation…

Monte-Carlo Analysis – Example (cont.)

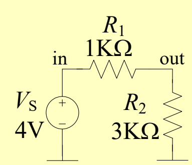


Assuming a uniform PDF for all y_i with $i = 1 \ldots t$

```
% Nominal Design
Vs = 4; R1 = 1e3; R2 = 3e3;
Ynom = [Vs R1 R2];

% Calculate Responses of Interest at Nominal Design
[psi,R] = VoltDiv_SPICE1(Ynom);
OPresp = R{1};
Vout_nom = OPresp(:,3);

% Define Outcomes and Tolerances
N = 1000; % Number of outcomes.
tauVs = 0.03; % Tolerance for Vs.
tauR1 = 0.05; % Tolerance for R1.
tauR2 = 0.1;  % Tolerance for R2.
tau = [tauVs tauR1 tauR2]; % Vector of tolerances.

% Generate Random Outcomes with Uniform PDF
Y = zeros(N,length(Ynom)); % Matrix to store outcomes.
for j = 1:N
    Y(j,1) = Ynom(1)*(1+tau(1)*(2*rand-1));
    Y(j,2) = Ynom(2)*(1+tau(2)*(2*rand-1));
    Y(j,3) = Ynom(3)*(1+tau(3)*(2*rand-1));
end

% Calculate Responses of Interest at Random Outcomes
Vout_rand = zeros(1,N);
for j = 1:N
    [psi,R] = VoltDiv_SPICE1(Y(j,:));
    OPresp = R{1};
    Vout_rand(j) = OPresp(:,3);
end

% Plotting Results ...
```

Dr. J. E. Rayas-Sánchez 15

Monte-Carlo Analysis – Example (cont.)

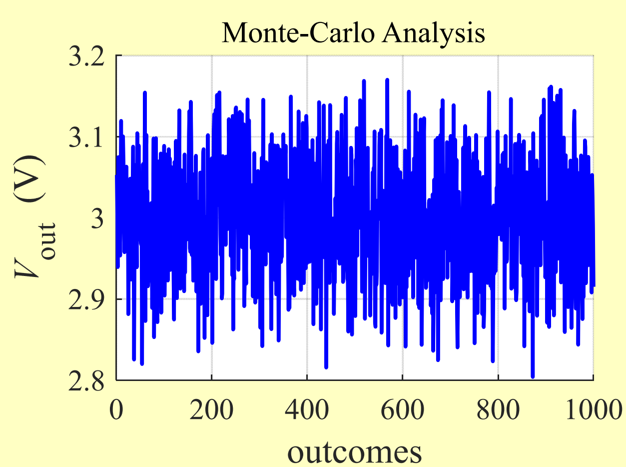


$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

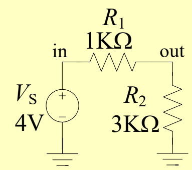
$V_{out}^{nom} = 3V$

Using $N = 1000$, uniform PDF



Dr. J. E. Rayas-Sánchez 16

Monte-Carlo Analysis – Example (cont.)

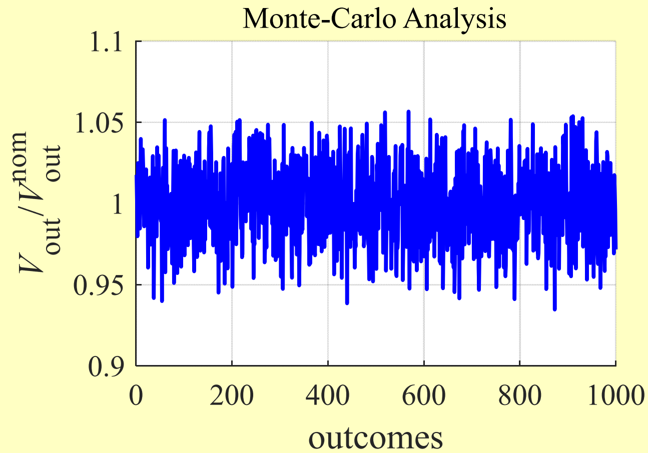


$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3\text{V}$

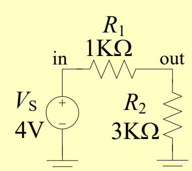
Using $N = 1000$, uniform PDF



Dr. J. E. Rayas-Sánchez

17

Monte-Carlo Analysis – Example (cont.)

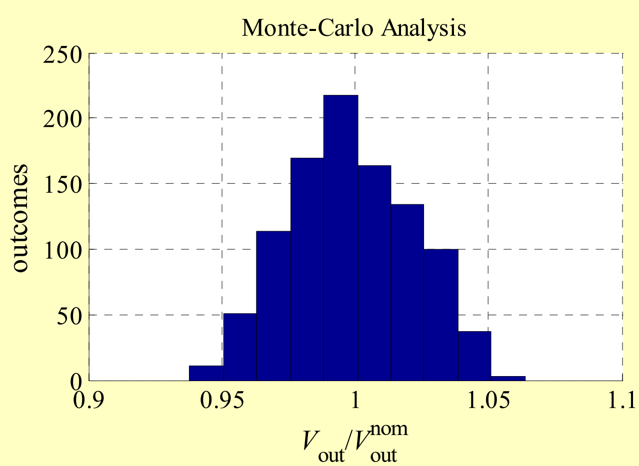


$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3\text{V}$

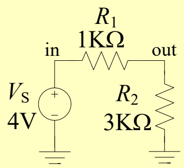
Using $N = 1000$, uniform PDF



Dr. J. E. Rayas-Sánchez

18

Monte-Carlo Analysis – Example (cont.)

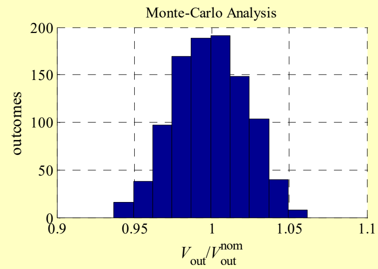


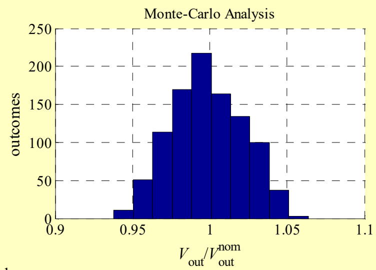
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

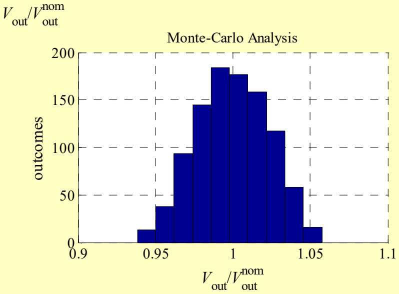
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

3 MC analysis with $N = 1000$ (uniform PDF)

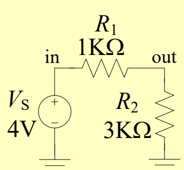






Dr. J. E. Rayas-Sánchez
19

Monte-Carlo Analysis – Example (cont.)

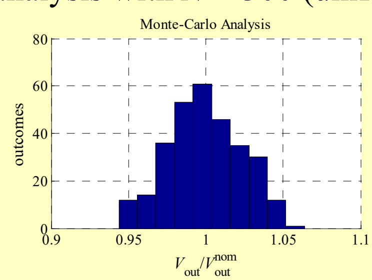


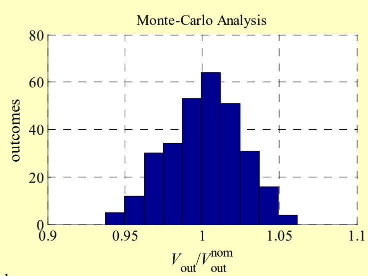
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

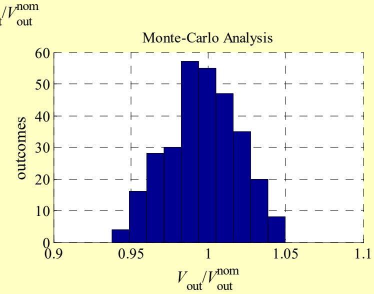
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

3 MC analysis with $N = 300$ (uniform PDF)

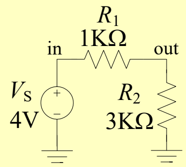






Dr. J. E. Rayas-Sánchez
20

Monte-Carlo Analysis – Example (cont.)

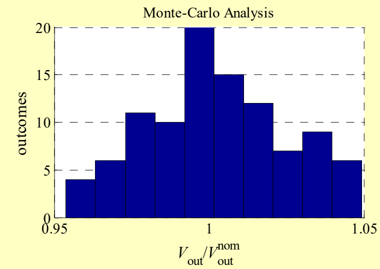


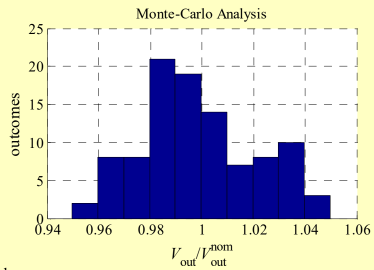
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

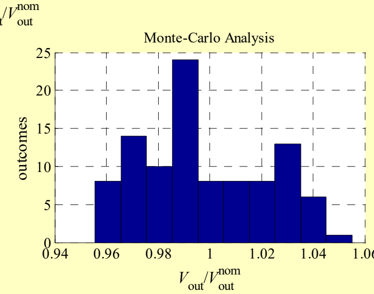
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

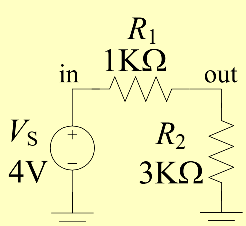
3 MC analysis with $N = 100$ (uniform PDF)







Monte-Carlo Analysis – Example (cont.)

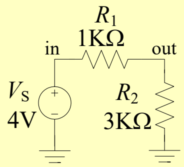


Assuming a normal PDF for all y_i with $i = 1 \ldots t$

```
% Nominal Design
Vs = 4; R1 = 1e3; R2 = 3e3;
Ynom = [Vs R1 R2];

% Calculate Responses of Interest at Nominal Design
[psi,R] = VoltDiv_SPICE1(Ynom);
OPresp = R{1};
Vout_nom = OPresp(:,3);

% Define Outcomes and Tolerances
N = 1000; % Number of outcomes.
tauVs = 0.03; % Tolerance for Vs.
tauR1 = 0.05; % Tolerance for R1.
tauR2 = 0.1;  % Tolerance for R2.
tau = [tauVs tauR1 tauR2]; % Vector of tolerances.

% Generate Random Outcomes with Normal (Gaussian) PDF
Y = zeros(N,length(Ynom)); % Matrix to store outcomes.
for j = 1:N
    Y(j,1) = Ynom(1)*(1+tau(1)*(2*(0.5+0.13*randn)-1));
    Y(j,2) = Ynom(2)*(1+tau(2)*(2*(0.5+0.13*randn)-1));
    Y(j,3) = Ynom(3)*(1+tau(3)*(2*(0.5+0.13*randn)-1));
end

% Calculate Responses of Interest at Random Outcomes
Vout_rand = zeros(1,N);
for j = 1:N
    [psi,R] = VoltDiv_SPICE1(Y(j,:));
    OPresp = R{1};
    Vout_rand(j) = OPresp(:,3);
end

% Plotting Results ...
```

$\mu = 0.5$
$\sigma = 0.13$

Monte-Carlo Analysis – Example (cont.)

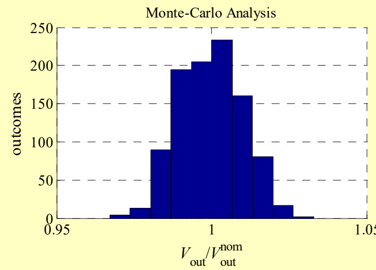


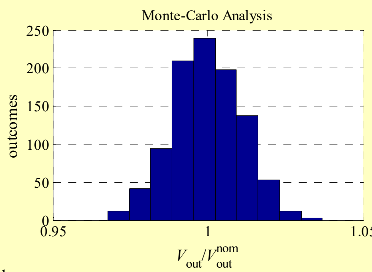
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

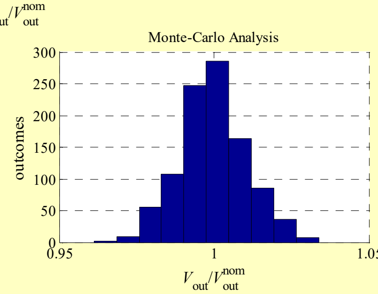
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

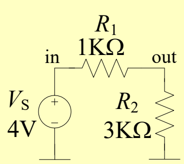
3 MC analysis with $N = 1000$ (normal PDF)







Monte-Carlo Analysis – Example (cont.)

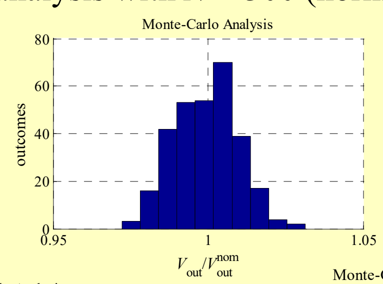


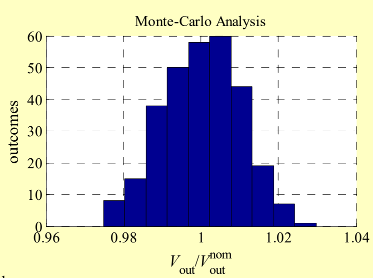
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

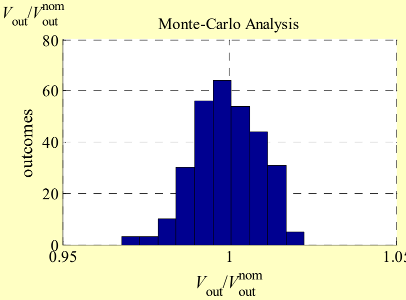
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

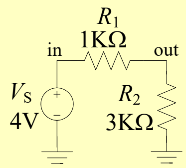
3 MC analysis with $N = 300$ (normal PDF)







Monte-Carlo Analysis – Example (cont.)

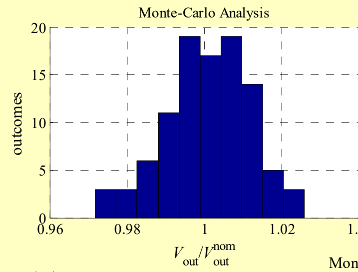


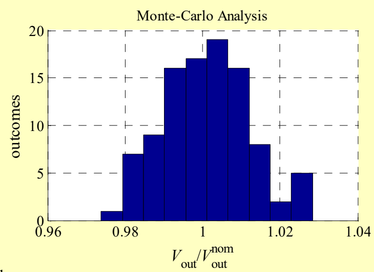
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

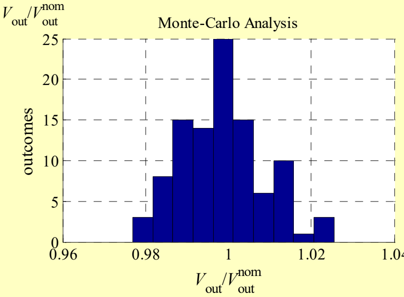
$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

$V_{out}^{nom} = 3V$

3 MC analysis with $N = 100$ (normal PDF)

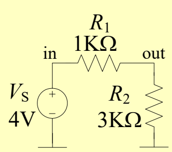






Dr. J. E. Rayas-Sánchez 25

Monte-Carlo Analysis – Example (cont.)



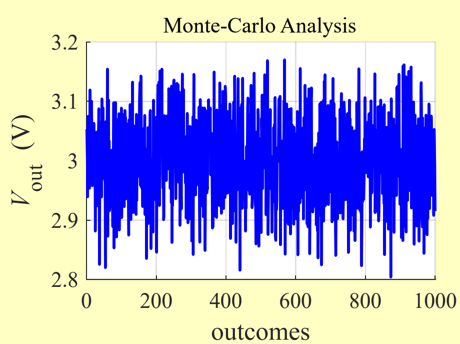
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

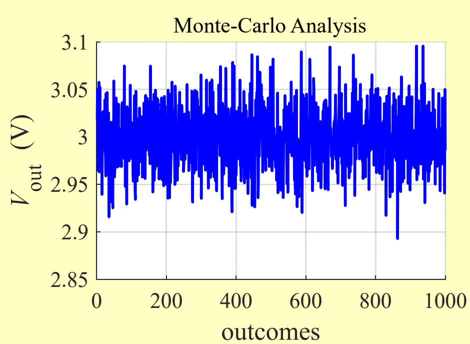
$V_{out}^{nom} = 3V$

MC analysis with $N = 1000$

Uniform PDF

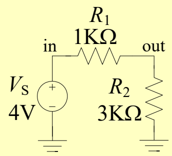


Normal PDF



Dr. J. E. Rayas-Sánchez 26

Monte-Carlo Analysis – Example (cont.)



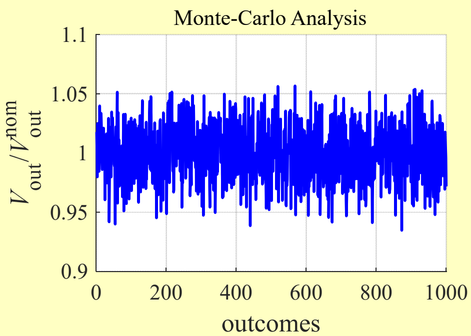
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

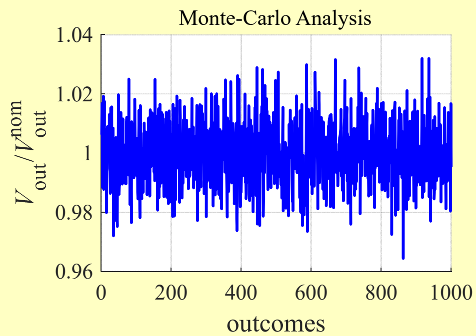
$V_{out}^{nom} = 3V$

MC analysis with $N = 1000$

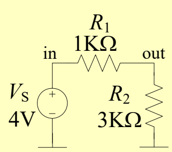
Uniform PDF



Normal PDF



Monte-Carlo Analysis – Example (cont.)



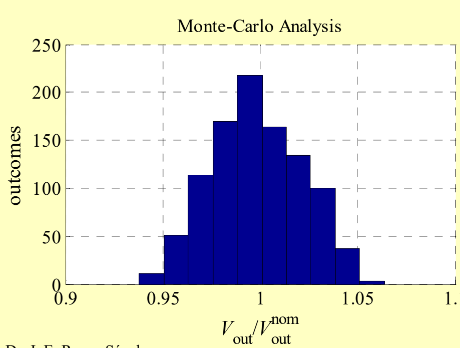
$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$

$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$

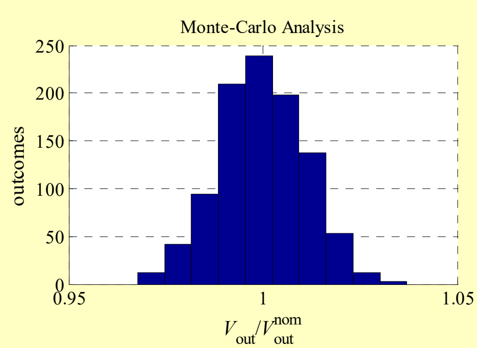
$V_{out}^{nom} = 3V$

MC analysis with $N = 1000$

Uniform PDF



Normal PDF



Yield Definitions

- The probability that a manufactured unit (outcome) will satisfy all its design specifications
- The ratio of the number of manufactured units which pass performance testing to the total number of units manufactured (in the limit, when the number of units tends to infinity)

Dr. J. E. Rayas-Sánchez 29

Estimating the Yield

- Using a minimax objective function U,

$$U(\boldsymbol{y}) = \max\{\ldots e_k(\boldsymbol{y})\ldots\}$$

- Vector $\boldsymbol{e}(\boldsymbol{y})$ contains all the error functions with respect to the design specifications
- $\boldsymbol{e}(\boldsymbol{y})$ is formulated such that
  - all the design specifications are satisfied by the j-th outcome $\boldsymbol{y}_j$ when all the elements in $\boldsymbol{e}(\boldsymbol{y}_j)$ are negative
  - any element $e_k(\boldsymbol{y}_j)$ in $\boldsymbol{e}(\boldsymbol{y}_j)$ with a positive value implies that the j-th outcome $\boldsymbol{y}_j$ is violating some design specification

Dr. J. E. Rayas-Sánchez 30

Estimating the Yield (cont.)

- Each outcome has an acceptance index defined by

$$I_a(\boldsymbol{y}_j) = \begin{cases} 1, & \text{if } U(\boldsymbol{y}_j) \le 0 \\ 0, & \text{if } U(\boldsymbol{y}_j) > 0 \end{cases}$$

- If N is sufficiently large for statistical significance, the yield Y at the nominal point $\boldsymbol{y}^{\text{nom}}$ can be approximated by using

$$Y(\boldsymbol{y}^{\text{nom}}) \approx \frac{1}{N}\sum_{j=1}^{N} I_a(\boldsymbol{y}^{\text{nom}} + \Delta\boldsymbol{y}_j) = \frac{1}{N}\sum_{j=1}^{N} I_a(\boldsymbol{y}_j)$$

Minimax Objective Function – Simple Errors

$$U(\boldsymbol{y}) = \max\{\ldots e_k(\boldsymbol{y})\ldots\}$$

$$\text{where} \quad e_k(\boldsymbol{y}) = \begin{cases} R_k(\boldsymbol{y}) - S_k^{\text{ub}} & \text{for all } k \in I^{\text{ub}} \\ S_k^{\text{lb}} - R_k(\boldsymbol{y}) & \text{for all } k \in I^{\text{lb}} \\ \left(| R_k(\boldsymbol{y}) - S_k^{\text{eq}} | / \Delta S_k^{\text{eq}}\right) - 1 & \text{for all } k \in I^{\text{eq}} \end{cases}$$

- $R_k(\boldsymbol{y})$ is the k-th model response at outcome $\boldsymbol{y}$
- $S_k^{\text{ub}} \ge 0$ and $S_k^{\text{lb}} \ge 0$ are upper and lower bound specs, and S_k^{eq} are equality specs with tolerance ΔS_k^{eq}
- I^{ub}, I^{lb} and I^{eq} are index sets (not necessarily disjoint)

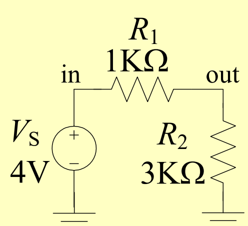
Minimax Objective Function – Normalized Errors

$$U(\boldsymbol{y}) = \max\{\ldots e_k(\boldsymbol{y})\ldots\}$$

$$\text{where} \quad e_k(\boldsymbol{y}) = \begin{cases} \dfrac{R_k(\boldsymbol{y})}{S_k^{\text{ub}} + \varepsilon} - 1 & \text{for all } k \in I^{\text{ub}} \\ 1 - \dfrac{R_k(\boldsymbol{y})}{S_k^{\text{lb}} + \varepsilon} & \text{for all } k \in I^{\text{lb}} \\ \dfrac{|R_k(\boldsymbol{y}) - S_k^{\text{eq}}|}{\Delta S_k^{\text{eq}}} - 1 & \text{for all } k \in I^{\text{eq}} \end{cases}$$

- $R_k(\boldsymbol{y})$ is the k-th model response at outcome $\boldsymbol{y}$
- $S_k^{\text{ub}} \geq 0$ and $S_k^{\text{lb}} \geq 0$ are upper and lower bound specs, and S_k^{eq} are equality specs with tolerance ΔS_k^{eq}
- I^{ub}, I^{lb} and I^{eq} are index sets (not necessarily disjoint)
- ε is an arbitrary small positive number

Yield Estimation – Example



$$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$$

$$\boldsymbol{y}^{\text{nom}} = [4\text{V} \quad 1\text{K}\Omega \quad 3\text{K}\Omega]^T$$

$$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$$

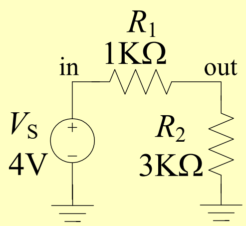
$$\boldsymbol{R}(\boldsymbol{y}) = [V_{\text{in}} \quad V_{\text{out}} \quad I_{V_S}]^T$$

$$\boldsymbol{R}(\boldsymbol{y}^{\text{nom}}) = [4\text{V} \quad 3\text{V} \quad 1\text{mA}]^T$$

Estimate the yield for the following design specifications:

$$2.9\text{V} \leq V_{\text{out}} \leq 3.1\text{V}$$

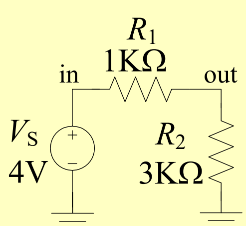
Yield Estimation – Example (cont.)



Obtain circuit responses from a Monte-Carlo analysis

```
% Nominal Design
Vs = 4; R1 = 1e3; R2 = 3e3;
Ynom = [Vs R1 R2];

% Calculate Responses of Interest at Nominal Design
[psi,R] = VoltDiv_SPICE1(Ynom);
OPresp = R{1};
Vout_nom = OPresp(:,3);

% Define Outcomes and Tolerances
N = 1000; % Number of outcomes.
tauVs = 0.03; % Tolerance for Vs.
tauR1 = 0.05; % Tolerance for R1.
tauR2 = 0.1;  % Tolerance for R2.
tau = [tauVs tauR1 tauR2]; % Vector of tolerances.

% Generate Random Outcomes with Uniform PDF
Y = zeros(N,length(Ynom)); % Matrix to store outcomes.
for j = 1:N
    Y(j,1) = Ynom(1)*(1+tau(1)*(2*rand-1));
    Y(j,2) = Ynom(2)*(1+tau(2)*(2*rand-1));
    Y(j,3) = Ynom(3)*(1+tau(3)*(2*rand-1));
end

% Calculate Responses of Interest at Random Outcomes
Vout_rand = zeros(1,N);
for j = 1:N
    [psi,R] = VoltDiv_SPICE1(Y(j,:));
    OPresp = R{1};
    Vout_rand(j) = OPresp(:,3);
end
```

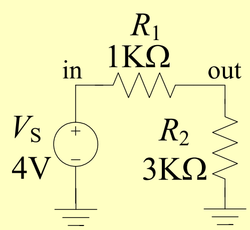

Yield Estimation – Example (cont.)



Define a minimax objective function using design specs

```
% Minimax Objective Function

function u = MinMax_VoltDiv(Vo)

% Design Specs
Vo_lb = 2.9;
Vo_ub = 3.1;

% Evaluate Vector of Error Functions
eps = 1e-12;
e1 = Vo/(Vo_ub+eps) - 1;
e2 = 1 - Vo/(Vo_lb+eps);
e = [e1 e2];

% Calculate Minimax Objective Function Value
u = max(e);
```

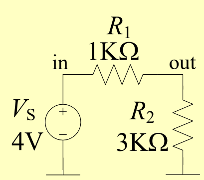
Yield Estimation – Example (cont.)



Estimate yield using Monte-Carlo responses and minimax objective function

```
% Calculate Index of Acceptability
u = zeros(1,N); % Initialize minimax objective function
                % values.
Ia = ones(1,N); % Initialize vector of acceptability
                % index (100% yield).
for j = 1:N
    u = MinMax_VoltDiv(Vout_rand(j));
    if u > 0
        Ia(j) = 0;
    end
end

% Estimate Yield
yield = sum(Ia)/N;

% Display Yield Results
disp('Yield Estimation of a Simple Voltage Divider')
Yield_label = ['yield = ' num2str(yield*100) ' %'];
disp(Yield_label);
disp(['outcomes = ' num2str(N)]);
```


Yield Estimation – Example (cont.)



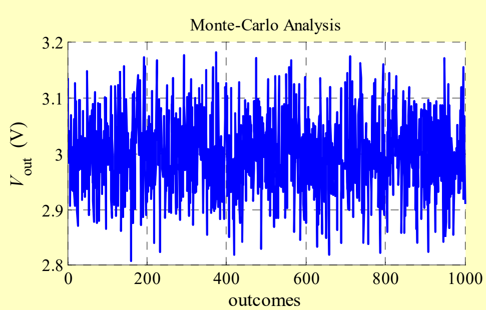
$$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$$

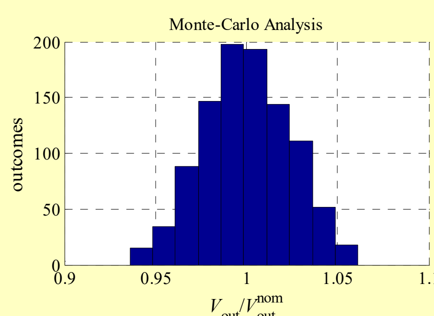
$$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$$

Specs:

$$2.9\text{V} \le V_{out} \le 3.1\text{V}$$

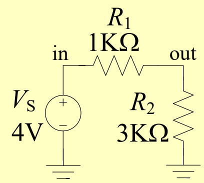
Yield estimation from a Monte-Carlo analysis with 1000 outcomes and uniform PDFs





Yield = 83.3%

Yield Estimation – Example (cont.)



$$\boldsymbol{y} = [V_S \quad R_1 \quad R_2]^T$$

$$\boldsymbol{\tau} = [0.03 \quad 0.05 \quad 0.1]^T$$

Specs:

$$2.9\text{V} \le V_{out} \le 3.1\text{V}$$

Yield estimations from a Monte-Carlo analysis with *N* outcomes and uniform PDFs

N	yield (%)				
	run 1	run 2	run 3	μ_{yield}	σ_{yield}
50	82.0	64.0	92.0	79.3	11.59
100	75.0	80.0	85.0	80.0	4.08
250	81.6	86.0	81.2	82.9	2.17
500	85.2	83.8	83.4	84.1	0.77
1000	82.1	83.6	83.3	83.0	0.65
1500	83.5	83.8	82.9	83.4	0.37

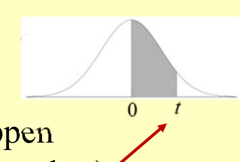
Calculating the Number of Outcomes, *N*

- We assume that all the statistical circuit parameters follow a Gaussian probability distribution function
- The number of outcomes *N* needed to have a certainty *c* when calculating the yield can be obtained from

$$N = \text{round}\left\{ \frac{[t(c)]^2}{\varepsilon^2} (Y)(1-Y) \right\}$$

where

- Y : estimated (expected) yield, $(0 < Y < 1)$
- ε : error in the yield estimation
- t : statistical value with a probability c to happen (c is the area under the bell curve between $-t$ and $+t$)



(Meehan and Purviance, 1993)

Calculating the Number of Outcomes, N (cont.)

Example:

- If the expected yield for a circuit during Monte Carlo analysis, using normal probability distributions for the fluctuating circuit parameters, is 83% ± 1%, within a 95% certainty, calculate the number of outcomes needed