11/12/06

SUPPLEMENT TO *Tolerance Analysis of Electronic Circuits Using MATLAB*, CRC Press, R. Boyd, 1999.

INTRODUCTION

These Adobe PDF files and MATLAB M-files illustrate additional original concepts and methods for tolerance analysis of electronic circuits. Added material includes:

- The most accurate method of RSS analysis of circuits with monotonic components and assymetric tolerances. Provides the same answers as Monte Carlo Analysis (MCA) using Normal distribution inputs, with large number of samples Nk. (diffamp.m and fig1819.m)
- Demonstrates how to detect circuits with non-monotonic components without plotting output vs. component value. (Applying EVA, RSS, or FMCA methods to circuits with non-monotonic components will yield erroneous results; with the degree of error directly proportional to the severity of the non-monotonicity. Then MCA is the only tolerance analysis method that can be used.) (bpfinflpts.m & compmon.pdf)
- Gives examples of tolerance analyis methods for non-linear circuits such as clock oscillators and A-to-D converters. (temptest.m & comp100.m)
- Shows how to handle circuits with a large number of nodes. One example is a uA733 VHF Video Amplifier with 18 nodes. (uA733.m)
- Provides one method of transient tolerance analysis using MATLAB's ODE functions. (dftvivo6.m)
- Shows the results of a DC MCA analysis with a skewed (assymetric) distribution input.
- Illustrates one method of ratiometric tolerancing using a uA733 VHF Video Amplifier IC as an example. (uA733.m)

The M-files are included in the zip file; some are also included in the Adobe PDF in the files. See the *.pdf files for contents.

In addition, the following improvements have been made:

- Faster method of forming {Nf Nc} binary array for FMCA, using `dec2bin`. Eliminates six statements.
- Faster MCA and FMCA execution times by reserving space for large output arrays, e.g.,

  ```
  Tn=zeros(Nc,Nk)
  ```

- A vector of nominal components is used for circuit function entry. The vector is unpacked in the circuit function. This allows cleaner, shorter and faster circuit function calls. Example

  Old method:
  `Vo(i,k)=G3(R1*Tn(1,k),R2*Tn(2,k),R3*Tn(3,k),C1*Tn(4,k),C2*Tn(5,k),s)`
  New method: `Nom=[R1 R2 R3 C1 C2];Vo(i,k)=G3(Nom.*Tn(:,k),s);`

- Sensitivity analyses no longer use an {Nc Nc} Q array (Nc = number of components) with 1+dpf on the diagonal. Q = 1+dpf (and the new B = 1-dpf; see 5. below) is now sequentially permuted in an Nc vector of 1's.

- Centered difference approximations are used for sensitivities which are more accurate. Components are perturbated in both directions with Q = 1+dpf and B = 1-dpf to get an average slope. That is, instead of the usual derivative approximation:

$$\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

The more accurate method is

$$\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x-\Delta x)}{2 \cdot \Delta x}$$

- For-end loops have been replaced with faster vectorization whenever possible.

Old method: `for q=1:nb;bin(q)=VL + intv*(q-1);end;`
New method: `q=1:nb;bin(q)=VL + intv*(q-1);`

- More efficient RSS calculation using `norm.`

Old method:

```
sum1(i)=0;sum2(i)=0;
for p=1:Nc
   sum1(i)=sum1(i)+(Sen(i,p)*(L(i,p)-1))^2;
   sum2(i)=sum2(i)+(Sen(i,p)*(H(i,p)-1))^2;
end
Vrss1(i)=Vo(i)*(1-sqrt(sum1(i)));
Vrss2(i)=Vo(i)*(1-sqrt(sum2(i)));
```

New method:

```
Mr=1+(T(2,:)+T(1,:))/2; % Also for asymmetric tolerances in EVA
Tv=(T(2,:)-T(1,:))./(2*Mr);
STn=norm(Sen(i,:).*Tv);
Vrss1(i)=Va(i)*(1-STn);Vrss2(i)=Va(i)*(1+STn);
```

- Unhampered by publisher restrictions, plots use color for easier interpretation.

- Circuit examples are in Adobe PDF format which can be easily read. (Some Word files with Mathtype and grahics corrupt when zipped and uploaded to a website.)

- New circuits have been added that are not in the book. For example, a time-domain analysis of a three-opamp fullwave rectifier. (hwrsine2.m)

- Figure numbers correspond to figure numbers in the book.

The principle drawback of the methods used here and in the book is that a circuit transfer function or nodal A(s) and B(s) arrays must be obtained for the circuit analysis. For large circuits, this is time-consuming, error-prone, and impractical. The reference below by the author provides methods for SPICE-like node lists describing small or large circuits (maximum of 89 nodes, maximum of 10 inputs, no restrictions on output nodes) with no loop or nodal analysis math required. The Mathcad programs used there can be easily transcribed into MATLAB M-files.

Node List Tolerance Analysis – Enhancing SPICE Capabilities With Mathcad, CRC Press, 2006.

The author can be contacted via Email at bobboyd@ieee.org