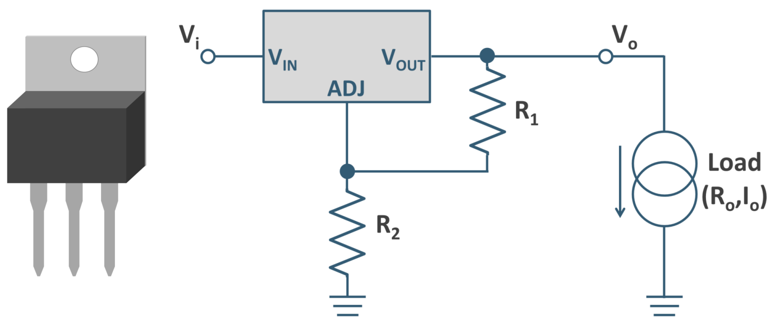
Use Optimization for Worst Case Analysis

Introduction

In this application, the worst case analysis for a circuit with Three-pin regulator is performed. Usually, Extreme value analysis, Roor Sum Square, and Monte Carlo analysis are used for the worst case analysis. As the other approach, the optimization technique can be applied for it, especially to obtain maximum and minimum cases to replace the extreme value analysis. In terms of finding the global optimal solution, the Global Optimization toolbox (add-on) is used effectively in this calculation.

Note : The Global Optimization toolbox (Add-on) is required to run this application.

Equation of Output voltage



The output voltage is calculated with the following equation.

$$> \ Eq_{vout} := V_o = \left(V_{ref} - I_o \cdot R_o\right) \cdot \left(1 + \frac{R_2}{R_1}\right) + I_{adj} \cdot R_2 :$$

Parameters

In this section, the setting of parameter are defined.

Parameters

$$> \ Par := \left[R_1, R_2, V_{ref}, I_{adj}, R_o, I_o\right] :$$

Nominal value

> $Nom := [243, 715, 1.25, 50 \cdot 10^{-6}, 0.007, 0.6]:$

Tolerance [%]

> $Tol := [1.24, 1.24, 4.00, 25, 100, 100]:$

By using Data Frame, easier to check the parameters.

> $DataFrame(\langle Nom[\ ]; Tol[\ ]\rangle, columns = Par, rows = [Nominal, Tolerance]);$

	R_1	R_2	V_{ref}	I_{adj}	R_o	I_o
Nominal	243	715	1.25	$\frac{1}{20000}$	0.007	0.6
Tolerance	1.24	1.24	4.00	25	100	100

(3.1)

Extreme value analysis

Generate a list of parameter for Extreme value analysis.

> $PList := [seq(Nom + \sim Nom \cdot \sim Tol \cdot \sim (Bits{:-}Split(i, bits = nops(Nom)) \cdot 2 - \sim 1) \cdot 0.01, i = 1 .. nops(Nom)^2)]:$

Calculate the output voltage with the above patterns of parameter values.

> $Res_{EVA} := [seq(eval(rhs(Eq_{vout}), [seq(Par[i] = PList[j][i], i = 1 .. nops(Par))]), j = 1 .. nops(PList))]:$

Search max/min value and the position of paremeter values in the list.

> $maxvalue_{EVA}, maxpos_{EVA} := ListTools{:-}FindMaximalElement(Res_{EVA}, position):$

> $minvalue_{EVA}, minpos_{EVA} := ListTools{:-}FindMinimalElement(Res_{EVA}, position):$

> $nomvalue_{EVA} := eval(rhs(Eq_{vout}), [seq(Par[i] = Nom[i], i = 1 .. nops(Par))]):$

Thus, the result of extreme value analysis is shown in the following table

> $DataFrame(\langle minvalue_{EVA}, PList[minpos_{EVA}][\ ]; nomvalue_{EVA}, Nom[\ ]; maxvalue_{EVA}, PList[maxpos_{EVA}][\ ]\rangle, columns = [value, Par[\ ]], rows = [Minimum, Nominal, Maximum])$

$$[[, value, R_1, R_2, V_{ref}, I_{adj}, R_o, I_o], \tag{3.1.1}$$

$$[Minimum, 4.670851303, 246.0132, 706.1340, 1.200000, 0.00003750000000, 0., 0.],$$

$$[Nominal, 4.947175514, 243, 715, 1.25, \frac{1}{20000}, 0.007, 0.6],$$

$$[Maximum, 5.266398121, 239.9868, 723.8660, 1.300000, 0.00006250000000, 0., 0.]]$$

Use Optimization

In order to calculate Maximum/Minimum value, the optimization technique can be applied. And, the Global optimization package help to avoid the local minimum(maximum), so that you can obtain the true minimum/maximum value.

Instead of generating all patterns of parameter values for Extreme value analysis, create maximum and minimum value list to limit the range of each parameter in Optimization.

> $MaxList := Nom + Nom *\sim Tol * 0.01 :$

> $MinList := Nom - Nom *\sim Tol * 0.01 :$

Calculate the maximum/minimum value with Global optimization package. The default settings of algorithm is Differential Evolution. See Global Optimization.

> $Res_{OPT\_max} := GlobalOptimization\text{:-}GlobalSolve\left(rhs\left(Eq_{vout}\right), seq\left(Par[i] = MinList[i] .. MaxList[i], i = 1 .. nops(Par)\right), maximize\right) :$

> $Res_{OPT\_min} := GlobalOptimization\text{:-}GlobalSolve\left(rhs\left(Eq_{vout}\right), seq\left(Par[i] = MinList[i] .. MaxList[i], i = 1 .. nops(Par)\right)\right) :$

Thus, the result of extreme value analysis with Optimization is shown in the following table

> $DataFrame\left(\left\langle Res_{OPT\_min}[1], eval\left(Par, Res_{OPT\_min}[2]\right)[\ ]; nomvalue_{EVA}, Nom[\ ]; Res_{OPT\_max}[1], eval\left(Par, Res_{OPT\_max}[2]\right)[\ ]\right\rangle, columns = [value, Par[\ ]], rows = [Minimum, Nominal, Maximum]\right)$

$$\left[\left[\ , value, R_1, R_2, V_{ref}, I_{adj}, R_o, I_o\right], \left[Minimum, 4.60583010475181798, 246.013200000000, 706.134000000000, 1.20000000000000, 0.0000375000000000000, 0.0140000000000000, 1.20000000000000\right], \left[Nominal, 4.947175514, 243, 715, 1.25, \frac{1}{20000}, 0.007, 0.6\right], \left[Maximum, 5.26639812194066526, 239.986800000000, 723.866000000000, 1.30000000000000, 0.0000625000000000000, 0., 0.\right]\right] \qquad (3.2.1)$$

In this case, the same maximum pattern can be obtained with both approaches. On the other hand, the optimization technique shows the better result than the usual Extreme value analysis. In addition to the accuracy of the result, the calculation cost can be reduced by using the optimization technique because of the efficient algorithm compared with the round-robin approach, especially if the number of parameter will be bigger.