An Analytic/Numeric Approach to ARGESIM Comparisons C3 Class-E Amplifier with Mathematica

M. Wibmer, F. Breitenecker;
Vienna University of Technology
mwibmer@osiris.tuwien.ac.at

Simulator: Mathematica is a tool for scientific computing which offers a wide variety of symbolical and numerical operations.

Model: We make use of Mathematica for a simulation study of a nonautonomous linear system of ordinary differential equations. Depending on each task we choose between the capabilities for symbolical and numerical computations within the Mathematica framework.

The model it self isn't very complicated to implement by the help of the Mathematica programming language. As it provides no special "piecewise" function as e.g. Maple V, the time dependent resistor function R(t) has to be modelled via the Unitstep Mathematica function. So the function constructed by

```
S[t_]:=Unitstep[t-a]-Unitstep[t-b]
```

is one for a<=t<=b and zero otherwise. With help of this concept the resistor function R(t) is implemented as follows

```
R[t_]:=Module[{ttr},ttr=t-
              IntegerPart[t/t2]t2;
((R2-R1)/TRF*ttr+R1)*
(UnitStep[ttr] - UnitStep[ttr - TRF])
+R2*(UnitStep[ttr-TRF]-UnitStep[ttr - t1])+
((R1-R2)/TRF*(ttr-t1)+R2)*(UnitStep[ttr-t1]
- UnitStep[ttr-t1-TRF])+
R1(UnitStep[ttr-t1-TRF]-UnitStep[ttr-t2])+
R1 UnitStep[t2]]
```

The further implementation is straight forward.

Task a: Calculation of Eigenvalues. We choose now specific values for R(t). The on-period has R(t)=0.05Ohm and the off period R(t)=5E+6Ohm. These values are constant, so the right-hand side of the governing equations is nonautonomous and the eigenvalues can be determined in as follows (not: all parameters except Res are predefined):

```
A={{0,-1/L1,0,0},{1/C2,-1/(C2*Res),-
1/C2,0}, {1/L3,0,-RL/L3,-
1/L3},{0,0,1/C4,0}};
Eigenvalues[A /.{Res->R_on}]
Eigenvalues[A /.{Res->Roff}]
```

Results are given in the following table.

Eigenvalues ON-Period	Eigenvalues OFF-Period
-1.11732E+9	-3941.59 + 833296i
-625.786	-3941.59 -833296i
-112931+658369i	-108995-661370i
-112931-658369i	-108995+661370i

Task b: Simulation of the stiff system. To simulate the system Mathematica provides the NDSolve command. It chooses automatically an integration algorithm whether the model is stiff or not. The following commands produce results as given in figure 1:

```
sol = NDSolve[dglwithinit /. TRF -> 10^(-
15), {x1, x2, x3, x4}, {t, 0, 100 *10^(-6)},
MaxSteps -> 5000, AccuracyGoal -> 30];
Plot[Evaluate[{{x2[t]/R[t]} /. TRF -> 10^(-
15), x3[t] RL} /. sol] , {t, 0, 0.0001},
PlotPoints -> 5000] ;
```

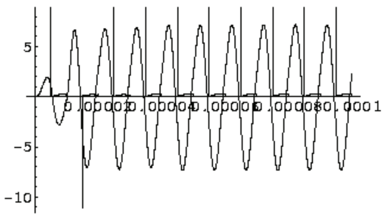

Figure 1 : IR and VL as a function of time

Task c: Parameter Study. With help of Tables this task can be solved as in Task b. The phase portrait is plotted with help of the ParametricPlot command.

```
newinitvals=Evaluate[{x1[t2], x2[t2],
x3[t2],x4[t2]}/.sol];
taskcdgl = dgl \[Union] {newinitconds};
TRFlist={10^(-15),10^(-11),10^(-9),10^(-7)};
taskcsol=Table[ NDSolve[ taskcdgl/.TRF->
  TRFlist[[i]],{x1,x2,x3,x4},{t,0,9*10^(-6)},
  MaxSteps->5000,AccuracyGoal->5],{i,1,4}];
ParametricPlot[Evaluate[Table[{{x3[t],
              1/L3(x2[t]-RL*x3[t]-x4[t])}/.
              taskcsol[[i, 1]]},{i, 1, 4}],
              {t, 0, 9 10^(-6)}]]
```

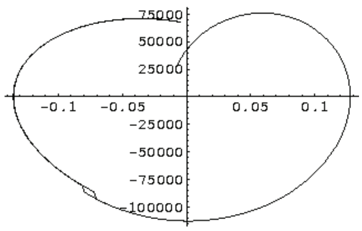

Figure 2 : Phase portraits for different TRF times

C3 Classification: Numerical / Symbolical Approach

Simulator: Mathematica Rec. Release 2003