



Simulation and Analysis of ARGESIM Benchmark C3 ‘Class-E Amplifier’ based on a Simulink Model with MATLAB functions

Johannes Asamer, arsenal research, Austria

Günther Zauner, Vienna University of Technology, Austria

Simulator: MatLab is platform independent software, developed by MathWorks. The software was originally designed for matrix calculations (Matrix Laboratory). Programs can be written as executable script, which allows a generation of (commercial) toolboxes for general use. Code can also be exported to other programming languages (e.g. C or Java).

Simulink also has been developed by MathWorks and needs MatLab for execution. By using predefined blocks different tasks can be modelled graphically. Moreover it is possible to create new blocks by implementing MatLab-Code. Additional blocks with complex functions from many areas of science are available. With Simulink, dynamic systems can be modelled and it is used in control theory and signal processing. Because it is based on MatLab the solutions are also calculated numerical, wherefore several ODE-solver can be used.

Model: In following section the modelling of a class E amplifier, introduced by N.O. Sokal and A.D. Sokal is described. By applying Kirchhoff’s circuit laws the amplifier can be characterized by following formulas:

$$dx1 / dt = (-x2 + VDC) / L1 \quad (1)$$

$$dx2 / dt = (x1 - x2 / R(t) - x3) / C2 \quad (2)$$

$$dx3 / dt = (x2 - RL * x3 - x4) / L3 \quad (3)$$

$$dx4 / dt = x3 / C4 \quad (4)$$

The aim was to implement these characteristic formulas into MatLab/Simulink. Therefore four blocks have been created, which contain the formulas. The properties of one block can be defined by referencing to a MatLab-script, which contains the associated formula. The input of each block is a vector with four elements (\vec{x}) and the output is a single value, representing the first derivation of one element of \vec{x} . Although inside the formulas just some elements of the vector are used, the whole vector is transferred. A single element is accessed just by declaring the according index of the vector-element.

The definition of formula 1 in the MatLab script is described in the following way:

```
1 function dx1 = equn1(x)
2 global VDC RL L1 C2 L3 C4
3 dx1 = (-x(2) + VDC) / L1;
```

By applying an integrator to the output of each block, the original (neither integrated nor differentiated) values are obtained. Afterwards these values are put together to one vector by using a multiplexer, provided by Simulink. Actually the multiplexer is used to put different signals on one bus, but is interpreted in this project as a vector with four elements. Moreover for the input of formula 2 a fifth element has to be added to the vector (or bus) namely the controlling resistance $R(t)$. Because Simulink is a common environment for modelling signal flow, there are different kinds of signal sources. One of them is called ‘repeating sequence’ and is here used for modelling $R(t)$. For the definition of this block a set of paired values is necessary. The first value of a pair indicates a point of time and the second value the output value for this point. Values between two points are interpolated linearly. Because the signal form of $R(t)$ is a rectangle with a rise and fall time (TRF) five points have to be defined.

An alternative would be to build up the whole system with blocks and use one line for each signal. The advantage of the chosen way to model the Class-E amplifier is the compact structure. By using the blocks which are able to execute a predefined MatLab code, the big part of the model is reduced to a set of four blocks. Moreover the usage of signal busses leads to more clearness in representation of the model.

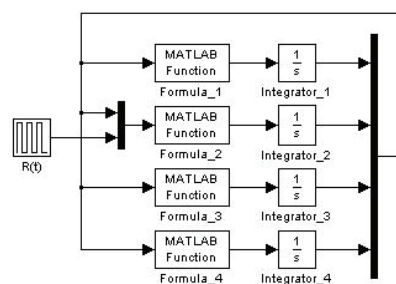


Figure 1. Simulink block diagram



A-Task: The first task is to calculate the eigenvalues of the system in the ON- and OFF period of $R(t)$. To do so first the Jacobi-matrix has to be calculated, which is the derivation of the model formulas with respect to \vec{x} . As mentioned above \vec{x} contains four elements, which represent the currents and voltages in the circuit.

Because the element in the second row and second column of the matrix depends on $R(t)$, the Jacobi-matrix changes over time. MatLab contains a function called $eig()$, which is suitable for calculating the eigenvalues of a matrix and therefore has been applied for this task.

The result of $eig(J_{on})$ and $eig(J_{off})$:

$$J_{OFF} = \begin{bmatrix} -54708 + i(1.04e+6) \\ -54708 - i(1.04e+6) \\ -58228 + i(5.33e+5) \\ -58228 - i(5.33e+5) \end{bmatrix} \quad J_{ON} = \begin{bmatrix} -1.12e+9 \\ -625.78 \\ -113040 + i(6.59e+5) \\ -113040 - i(6.59e+5) \end{bmatrix}$$

Out of the eigenvalues different conclusions can be drawn concerning the system behaviour. All eigenvalues have negative real parts which mean that the system is stable. For the OFF-Period the system can be observed as not-stiff because the absolute values of the eigenvalues are approximately in the same order of magnitude, in contrast to the eigenvalues of the ON-period, where this is not the case. This means the system behaviour changes between stiff (ON) and not stiff (OFF).

B-Task: The circuit has been build up in MatLab/Simulink as described above. Before executing the simulation an appropriate ODE-solver has to be chosen. Basically it has to be decided between solver with variable and fixed step size. Because in parts the system is stiff a solver with variable step size has to be used. It is also possible to use this solver for none-stiff systems but it would be problematic in terms of calculation time to use a solver with fixed step-size for stiff systems.

MatLab/Simulink offers seven different ODE-solvers with variable step size for continuous simulation. All of them resulted in (approximately) the same output.

Beside the runtime, differences could be observed in the form of the current and voltage on the output resistance. Basically the current on the load resistance should be zero except the periodically repeating peaks. But for three ODE-solvers (ODE45, ODE23 and ODE113) a noise with quite big amplitude ($\sim 2V$)

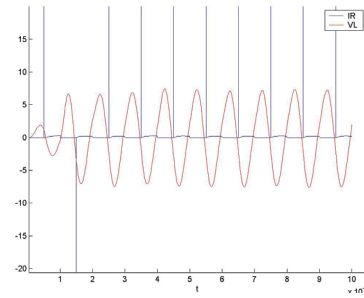


Figure 2. Desired output calculated with ODE23s and ODE23tb

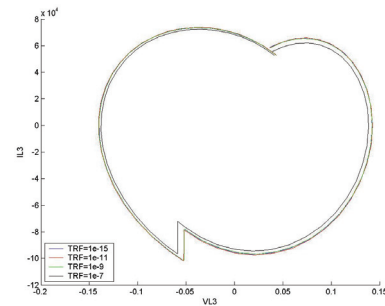


Figure 3. Phase diagram of $VL3$ versus $IL3$

was overlaid to the current. For two other solvers (ODE15s and ODE23t) the voltage on the output showed a blurred form. Only two of the seven solvers (ODE23s and ODE23tb) provide the results which were expected from the simulation.

C-Task: In the third task the influence of the rise and fall time of $R(t)$ (TRF) was investigated. Therefore a phase diagram was visualized containing the current and the voltage on coil $L3$. For the different simulation runs the TRF has been varied between 100ns and 1fs. Again the results are very similar. Only at a TRF of 100ns the phase diagram looks slightly different.

Resumé: MATLAB/Simulink offers a convenient tool for modelling and analysis of stiff ODE – systems. As discussed above, the choice of the best suited ODE – solver for a defined problem is one of the main points to be focused on before starting a simulation run.

Corresponding author: Johannes Asamer, arsenal research, Transport technologies, Giefinggasse 3, 1210 Vienna, Austria
Johannes.asamer@arsenal.ac.at

Received: June 20, 2007

Revised: October 3, 2007

Accepted: October 21, 2007