AND8214/D

General Thermal Transient RC Networks

Packaging Technology Development

Prepared by: Roger Paul Stout, PE
ON Semiconductor

ON Semiconductor®

http://onsemi.com

APPLICATION NOTE

INTRODUCTION

This monograph develops a generalized thermal RC network methodology suitable for generating transient response numerical simulations of thermal systems. Specific classes of boundary conditions are considered, namely, steps and ramps of applied power, and steps and ramps in temperature degree−of−freedom constraints. It also discusses the implications of the form of the general solutions to such networks with respect to the existence of mathematically equivalent “non−grounded−capacitor” thermal RC network models.

Glossary of Symbols

A_{ij}	cofactor or minor of a determinant
c	constant of integration
C_i	thermal capacitance of a variable temperature node
i, i^{th}, i = j,j, j^{th}	node index
k_j	constant representing generalized factor in a Laplace transform
$\underset{\sim}{G}$	the thermal conductance matrix, and its determinant det $\underset{\sim}{G}$
$\underset{\sim}{G}^i$	Cramerized thermal conductance matrix, and its determinant det $\underset{\sim}{G}^i$
$\underset{\sim}{G}^{-1}$	thermal impedance matrix
m	“order” of power vector
n	n + 1 (2n + 3) expressions relating to number of variable nodes
$\overline{Q(s)}$	power vector incorporating only initial temperatures, steps and ramps
q_i	power input at a node
$\overline{\mathbf{Q}}$	Laplace transform of general power vector
R_{ij}, R_{ji}	thermal resistance between two nodes (assumed bidirectional)
s	Laplace transform complex variable, and its square s^2
T_i, T_j	nodal temperatures
$\overline{\mathbf{T}}$	Laplace transform of temperature vector
$\mathbf{T_i}$	Laplace transform of one nodal temperature
$\Delta\overline{\mathbf{T}}_i$	Laplace transform of one nodal temperature rise due to step heating

Basic Node Model

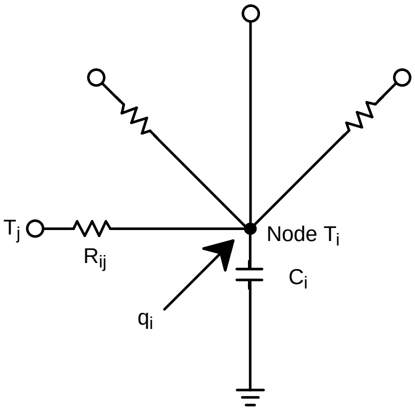

Figure 1. T_i

Let a typical variable temperature node in a thermal network be represented as shown in Figure 1, T_i. It consists of a thermal capacitance C_i (tied to thermal “ground”), any number of resistance links to other nodes R_{ij} (whether those nodes be at fixed temperatures or variable in their own right), and a power input q_i at the node. For illustration, we have simply labeled one of the other nodes as T_j, but the idea is that when we use a summation over resistance links, the index j will denote each of the other nodes connected to the node of interest (and not the node of interest itself). Then the equation which describes the energy balance at node T_i is:

$$q_i + \sum \frac{T_j - T_i}{R_{ij}} = C_i \frac{dT_i}{d_t} \qquad \text{(eq. 1)}$$

Transformation to *s*−plane

This differential equation can be converted to an algebraic equation through the use of the Laplace transform. Bolded symbols will represent the **Laplace** transformed quantities corresponding to the original variables. Note also that one aspect of the Laplace transform is that the initial value of the temperature appears explicitly in the transformed equation.

$$\mathbf{q}_i + \sum \frac{\mathbf{T}_j - \mathbf{T}_i}{R_{ij}} = sC_i\mathbf{T}_i - C_iT_{io} \qquad \text{(eq. 2)}$$

This can now be rearranged as follows:

$$\left(C_i s + \sum_{var} \frac{i}{R_{ij}}\right)\mathbf{T}_i - \sum_{var} \frac{\mathbf{T}_i}{R_{ij}} = \mathbf{q}_i + \sum_{fixed} \frac{\mathbf{T}_j}{R_{ij}} + C_iT_{io} \qquad \text{(eq. 3)}$$

where we have made a further distinction between variable temperature nodes and fixed temperature nodes. In this case, by fixed we do not mean so much that the temperature is necessarily constant, but that it is specified (a constraint or boundary condition), as opposed to being an unknown variable in the model. Each temperature node in a particular model will thus appear on either the left or right of the equation − one or the other, depending on its function in the particular model. For instance, in a thermal coldplate model which recognizes heat loss to the environment as well as into the coldplate itself, we will need both “chuck” and “ambient” fixed−temperature nodes. Their temperatures will be specified during the analysis, conceivably even changing over time. It is again important to recognize that the summations are always local to each variable temperature node of the network, and represent whatever set of links connect to that node. The “var” summations on the left include only the links between the node of interest and its adjacent variable nodes; the “fixed” summation on the right includes only the links between the (variable) node of interest and its adjacent fixed temperature nodes (if any). Note that no summation ever includes an $i = j$ term (i.e., there is no physical sense in having a resistor short itself out at a node!). Also, we assume that links are completely bidirectional, so there is no distinction between R_{ij} and R_{ji}.

Matrix Form

We can express the entire thermal network in one matrix equation, where each row and column of the matrix (and each element of the vectors) represent the quantities associated with a different variable temperature node.

$$\begin{pmatrix} C_1 s + \sum\limits_{var} \frac{1}{R_{1j}} & \cdots & -\frac{1}{R_{1i}} & \cdots \\ \vdots & \ddots & & \\ -\frac{1}{R_{1i}} & & C_i s + \sum\limits_{var} \frac{1}{R_{ij}} & \\ \vdots & & & \ddots \end{pmatrix} \begin{Bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_i \\ \vdots \end{Bmatrix} = \begin{Bmatrix} q_1 + \sum\limits_{fixed} \frac{\mathbf{T}_j}{R_{1j}} + C_1T_{1o} \\ \vdots \\ q_i + \sum\limits_{fixed} \frac{\mathbf{T}_j}{R_{ij}} + C_iT_{io} \\ \vdots \end{Bmatrix} \qquad \text{(eq. 4)}$$

If we define the following quantities:

$$\underset{\sim}{G} = \begin{pmatrix} C_1 s + \sum\limits_{var} \frac{1}{R_{1j}} & \cdots & -\frac{1}{R_{1i}} & \cdots \\ \vdots & \ddots & & \\ -\frac{1}{R_{1i}} & & C_i s + \sum\limits_{var} \frac{1}{R_{ij}} & \\ \vdots & & & \ddots \end{pmatrix} \qquad \text{(eq. 5)}$$

(which matrix we note, in passing, is symmetric)

$$\overline{\mathbf{T}} = \begin{Bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_i \\ \vdots \end{Bmatrix} \qquad \text{(eq. 6)}$$

and

$$\overline{\mathbf{Q}} = \begin{Bmatrix} q_1 + \sum\limits_{fixed} \frac{\mathbf{T}_j}{R_{1j}} + C_1T_{1o} \\ \vdots \\ q_i + \sum\limits_{fixed} \frac{\mathbf{T}_j}{R_{ij}} + C_iT_{io} \\ \vdots \end{Bmatrix} \qquad \text{(eq. 7)}$$

Then the entire network can be represented simply as:

$$\underset{\sim}{G} \cdot \overline{\mathbf{T}} = \overline{\mathbf{Q}} \qquad \text{(eq. 8)}$$

In ordinary application of this network model to either experimental data fitting, or otherwise in prediction of thermal performance of a system, we will generally specify all the quantities of $\underset{\sim}{G}$ and $\overline{\mathbf{Q}}$, and will seek solutions for $\overline{\mathbf{T}}$.

Thus $\overline{\mathbf{T}} = \underset{\sim}{G}^{-1} \cdot \overline{\mathbf{Q}}$ (eq. 9) which is easy enough to write symbolically, but provides certain challenges in practice.

Boundary Conditions

Before we go any further into specific applications, however, let us consider what sort of values are of interest for $\overline{\mathbf{Q}}$. Obviously, constant power inputs and truly fixed “reference” temperature nodes will be useful. If we limited ourselves to these boundary conditions, we could solve problems of any desired complexity in a piecewise fashion, simulating time−varying power and constraint conditions with square−edged constant waveforms (i.e. constant “steps”) approximating the shapes of the actual conditions. In so doing, we see that even though we might begin our first step with all nodes at some common temperature (thus momentarily eliminating the initial temperature constant terms), once we had started the solution, the nodes would move to different temperatures. The ending temperatures from each completed step would then become the initial temperature terms of the following step − so in general we must retain these initial temperature terms, and in general they will all be different from each other. We may also find that square−edged power and temperature constraints are too unrealistic for some situations, so let us permit them to include a “ramp” characteristic (i.e., constant non−zero slopes as needed) from the outset, and see if the extra manipulations become too burdensome. Utilizing the Laplace transforms of steps and ramps, we can write the ith element of $\overline{\mathbf{Q}}$ as follows:

$$\mathbf{Q}_i = \frac{q_i}{s} + \frac{\dot{q}_i}{s^2} + \frac{\sum \frac{T_j}{R_{ij}}}{s} + \frac{\sum \frac{\dot{T}_j}{R_{ij}}}{s^2} + C_i T_{io} \qquad \text{(eq. 10)}$$

Where all quantities (except s) are now true constants, and the “dotted” values are the slopes of the associated quantities. Finally, collecting powers of s yields:

$$\mathbf{Q}_i = \left(\dot{q}_i + \sum \frac{\dot{T}_j}{R_{ij}}\right)\frac{1}{s^2} + \left(q_i + \sum \frac{T_j}{R_{ij}}\right)\frac{1}{s} + C_i T_{io} \qquad \text{(eq. 11)}$$

It should now be clear that if we need to handle power “ramps,” it costs us nothing in complexity to throw in constrained−temperature node “ramps” as well. So with the step and ramp restriction, our system can be expressed entirely as polynomials in s, like this (where all the new subscripted quantities are simply constants):

$$\begin{pmatrix} C_1 s + g_{11} \cdots & -g_{1i} & \cdots \\ \vdots & \ddots & \\ -g_{1i} & C_i s + g_{ii} & \\ \vdots & & \ddots \end{pmatrix} \begin{Bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_i \\ \vdots \end{Bmatrix} = \frac{1}{s^2} \begin{Bmatrix} C_1 T_{1o} s^2 + a_1 s + b_1 \\ \vdots \\ C_i T_{io} s^2 + a_i s + b_i \\ \vdots \end{Bmatrix} \qquad \text{(eq. 12)}$$

Or, with an obvious definition of $\overline{Q(s)}$;

$$\underset{\sim}{G} \cdot \overline{\mathbf{T}} = \frac{1}{s^2}\overline{Q(s)} \qquad \text{(eq. 13)}$$

Solution Method

A brute−force solution of this system (using Cramer’s Rule) now can be carried out to the point that Laplace transforms of the nodal temperatures are expressed as:

$$\mathbf{T}_i = \frac{1}{s^2}\,\frac{\det \underset{\sim}{G}^i}{\det \underset{\sim}{G}} \qquad \text{(eq. 14)}$$

Where $\underset{\sim}{G}^i$ is the matrix obtained by substituting $\overline{Q(s)}$ for the ith column of $\underset{\sim}{G}$. It should be evident that (det $\underset{\sim}{G}$) ends up being a polynomial in s of order n, where n is the number of variable nodes in the network, and that (det $\underset{\sim}{G}^i$) ends up being a polynomial of order $n + 1$. With the additional s^2 term in the denominator, standard techniques will yield a fairly straightforward inverse transform of $\mathbf{T}_i$ into the time domain, resulting in a solution T_i for each node consisting of possibly a constant plus a linear term in time, and then mainly exponentials (in negative powers of time, with time constants derived from the roots of det $\underset{\sim}{G}$).

Should there be repeated roots, the result will get somewhat more complicated, nevertheless, the methodology is standard and direct. We can code this solution method into a computer program, and delay conversion from symbolic to numerical analysis until the last possible moment. A tool such as Theorist® or Mathematica® can be used to generate the determinants in full symbolic form, though if there are very many nodes in the model, there may be dubious value in so doing (especially since the equations will be unique for each network topology analyzed). Other tools, such as LabVIEW™, can be used to compute the determinants in a quasi−symbolic form (i.e. identifying which elements of the various matrices and vectors must be multiplied together and added with cofactor multiplicands), and resorting to numerical computations only whenever new polynomial coefficients are needed. It is not clear which approach, if either, has an advantage, since once the model consists of more than five nodes, there will never be a general, closed−form solution for the roots of the denominator. In either case, the determinants must be recomputed (symbolically or quasi−symbolically) whenever the topology changes; and in neither case do determinant structures have to be recomputed due to changes in specific network parameters. The question revolves more around the format of what gets saved in between parameter changes (and the associated costs in time and memory) − symbolic expressions for the total coefficients of the final numerator and denominator polynomials (each of which is a complex combination of the entire set of R's and C's, all of which appear multiple times in various permutations), or quasi−symbolic representations of the determinants, with potentially hundreds of individual equations in terms of row−element references. The former likely trades off a much lengthier derivation time of the symbolic coefficients (for a possibly somewhat faster recomputation of their numerical values), whereas the latter may be faster in generating the quasi−symbolic determinants, at the expense of having more numerical substitutions to be made into the actual computation of the polynomial coefficients.

There is at least one special case of further interest here, however. That is the “step heating” problem, where the entire network is at a uniform−temperature thermal equilibrium, and one or possibly more nodes are then powered up, each with a constant power (unpowered nodes being considered as constant zero power). We can refer all temperatures to that uniform initial temperature, eliminate the resulting zero terms (including the power and temperature “ramp” values), and obtain:

$$\begin{pmatrix} C_1 s + g_{11} \cdots & -g_{1i} & \cdots \\ \vdots & \ddots & \\ -g_{1i} & C_i s + g_{ii} & \\ \vdots & & \ddots \end{pmatrix} \begin{Bmatrix} \Delta \mathbf{T}_1 \\ \vdots \\ \Delta \mathbf{T}_i \\ \vdots \end{Bmatrix} = \frac{1}{s} \begin{Bmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \end{Bmatrix} \qquad \text{(eq. 15)}$$

Considering but one of the nodal solutions, it may now be seen that the result for $\Delta \mathbf{T}_i$ will be of the form:

$$\Delta \mathbf{T}_i = \frac{1}{s} \, \frac{\sum (q_j \cdot A_{ij})}{\det \underset{\sim}{G}} \qquad \text{(eq. 16)}$$

Rearranging slightly, we can also write this as:

$$\Delta \mathbf{T}_i = \sum q_j \cdot \frac{1}{s} \frac{A_{ij}}{\det \underset{\sim}{G}} \qquad \text{(eq. 17)}$$

Thus the Laplace transform of the solution for node i will be a series of n polynomial fractions, one for each node in the system. Each term has the same denominator, but each has a different numerator according to which node it represents, and each is multiplied by the power dissipation at that node. One term will be present for each powered node, and the solution for every node in the system (whether itself powered or not) will have the same number of independent terms. Powered nodes thus will have a so−called “self heating” term, plus “interaction heating” terms for each of the other powered nodes.

Implications for “Non−Grounded−Capacitor” Models

The main point in this derivation has been to show that an arbitrary thermal RC−network model can have any of its individual nodes' solutions expressed as a combination of terms individually proportional to the power inputs of each associated heated node. Obviously a computer program can be written to generate these solutions systematically. There may, however, be an unanticipated secondary conclusion to be drawn from this work. We have commonly referred to physically−significant, grounded−capacitor models, versus physically meaningless (but mathematically convenient) non−grounded−capacitor models. (See AND8221/D and AND8215/D for reference.) For simple single−input ladder networks, it is often not evident that the equivalent non−grounded models are nothing except mathematical convenience; nor has it been clear to us that for multiple−input grounded−capacitor models, there is even a corresponding non−grounded network at all. This derivation (of Equation 17) demonstrates that for the special case of constant power heating from a uniform equilibrium, there is at least a mathematically convenient equivalent to a non−grounded network for each node, though it is still not clear how one would construct such a network diagram. (The implication is that there would have to be some node whose temperature was simply the sum of the temperatures of several otherwise independent nodes. Perhaps a SPICE model would permit a variable to be declared that was just such a sum, but the non−grounded capacitor circuit being modeled could not be physically realized.) But because the A_{ij} are different for each node, each node in the non−grounded model must be represented by a different combination network. Further, it is not at all obvious that a circuit so derived (and the node whose temperature was the sum of the several components) would respond correctly to other combinations of inputs (especially from non−uniform starting temperatures, and certainly not different combinations of power and temperature ramps). This can be seen a little more clearly if we return to Equation 12 and dissect the right−hand side:

$$\underset{\sim}{G}\cdot\mathbf{T} = \left\{\begin{matrix} C_1T_{1o} \\ \vdots \\ C_iT_{io} \\ \vdots \end{matrix}\right\} + \frac{1}{s}\left\{\begin{matrix} q_1 + \sum\frac{T_j}{R_{1j}} \\ \vdots \\ q_i + \sum\frac{T_j}{R_{ij}} \\ \vdots \end{matrix}\right\} + \frac{1}{s^2}\left\{\begin{matrix} \dot{q}_1 + \sum\frac{\dot{T}_j}{R_{1j}} \\ \vdots \\ \dot{q}_i + \sum\frac{\dot{T}_j}{R_{ij}} \\ \vdots \end{matrix}\right\} \qquad \text{(eq. 18)}$$

If we choose to separate the temperature into three obvious components:

$$\mathbf{T} = \mathbf{T}_o + \mathbf{T}_{step} + \mathbf{T}_{ramp} \qquad \text{(eq. 19)}$$

Where:

$$\mathbf{T}_o = \underset{\sim}{G}^{-1}\left\{\begin{matrix} C_1T_{1o} \\ \vdots \\ C_iT_{io} \\ \vdots \end{matrix}\right\} \qquad \text{(eq. 20)}$$

and

$$\mathbf{T}_{step} = \frac{1}{s}\,\underset{\sim}{G}^{-1}\cdot\left\{\begin{matrix} q_1 + \sum\frac{T_j}{R_{1j}} \\ \vdots \\ q_i + \sum\frac{T_j}{R_{ij}} \\ \vdots \end{matrix}\right\} \qquad \text{(eq. 21)}$$

and

$$\mathbf{T}_{ramp} = \frac{1}{s^2}\,\underset{\sim}{G}^{-1}\cdot\left\{\begin{matrix} \dot{q}_1 + \sum\frac{\dot{T}_j}{R_{1j}} \\ \vdots \\ \dot{q}_i + \sum\frac{\dot{T}_j}{R_{ij}} \\ \vdots \end{matrix}\right\} \qquad \text{(eq. 22)}$$

We see that the influences of initial conditions, constant power (and constant temperature constraints), and ramps of power (with ramps of constraint temperature) each add their own independent contributions to the overall response of the network. It can now be seen precisely how the different contributions relate to each other − and it is not in simple "proportion." It is true (and perhaps a valuable insight), that fixed−temperature nodes act just as if they represented fixed power inputs. Likewise, power ramps and temperature constraint ramps are completely equivalent in effect. However, going back into the time domain, the relationship between initial conditions and constant power would be that the "power" contribution is related to the integral of the "initial condition" contribution. To see this, first consider a straightforward modification of Equation 17, which now incorporates the fixed−temperature contribution to the solution for any node i:

$$\Delta\mathbf{T}_{i\text{−}step} = \sum\left(\left(q_j + \sum\frac{T_k}{R_{jk}}\right)\cdot\frac{1}{s}\,\frac{A_{ij}}{\det\underset{\sim}{G}}\right) \qquad \text{(eq. 23)}$$

And by similarity, we can simply write the other two components of $\mathbf{T}_i$:

$$\Delta\mathbf{T}_{i\text{−}ramp} = \sum\left(\left(\dot{q}_j + \sum\frac{\dot{T}_k}{R_{jk}}\right)\cdot\frac{1}{s^2}\,\frac{A_{ij}}{\det\underset{\sim}{G}}\right) \qquad \text{(eq. 24)}$$

$$\Delta\mathbf{T}_{io} = \sum\left(C_jT_{jo}\cdot\frac{A_{ij}}{\det\underset{\sim}{G}}\right) \qquad \text{(eq. 25)}$$

Because the cofactors A_{ij} depend only on $\underset{\sim}{G}$ (and upon the node for which we're solving), what we have in all three cases is the inner product of some constant vector (which depends on whether we're talking about the ramp, the step, or the initial conditions), and a vector which depends only on $\underset{\sim}{G}$ (and is therefore fixed for whichever node is of interest). To put it another way, all three contributions to the overall solution can be written in the form:

$$\Delta\mathbf{T}_i\cdot m = \frac{1}{s^m}\sum\left(k_j\cdot\frac{A_{ij}}{\det\underset{\sim}{G}}\right) \qquad \text{(eq. 26)}$$

For any specific node i, therefore, the time−domain solution will be of the form:

$$\Delta T_{i(m)} = \sum\left(K_{j(m)}e^{-\frac{t}{\tau_j}}\right) \qquad \text{(eq. 27)}$$

As we increase m, the solution will change like this:

$$\Delta T_{i(m\,+\,1)} = c - \sum\left(\tau_jK_{j(m)}e^{-\frac{t}{\tau_j}}\right) \qquad \text{(eq. 28)}$$

Where c is a constant of integration. However, because the k_j are different for the different "orders" of the solution contributions, one cannot actually derive the values of the constants for one order, from a solution to another order. The most important point here is that there is not a simple relationship between the various contributions to the overall solution, and the non−grounded networks which describe one portion of the solution will therefore not work for the others.

LabVIEW is a trademark of National Instruments Corporation.
Mathematica is a registered trademark of Mathematica Policy Research, Inc.
Theorist is a registered trademark of Theorist Interactive, LLC.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 61312, Phoenix, Arizona 85082−1312 USA
Phone: 480−829−7710 or 800−344−3860 Toll Free USA/Canada
Fax: 480−829−7709 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada

Japan: ON Semiconductor, Japan Customer Focus Center
2−9−1 Kamimeguro, Meguro−ku, Tokyo, Japan 153−0051
Phone: 81−3−5773−3850

ON Semiconductor Website: http://onsemi.com

Order Literature: http://www.onsemi.com/litorder

For additional information, please contact your local Sales Representative.