Model Order Reduction for Lumped RC Transmission Lines

Farid N. Najm, *Fellow, IEEE*
f.najm@utoronto.ca
ECE Dept., University of Toronto
https://www.ece.utoronto.ca/people/najm-f-n

Abstract– We start with a detailed review of the PACT approach for model order reduction of RC networks. We then develop a method that uses PACT as a preprocessing step to transform a generic lumped RC transmission line of some nominal order, based on a nominal $(\tilde{r}, \tilde{c})$ setting, into a parameterized circuit captured in a `SPICE` sub-circuit description. Then, given any other lumped RC line of the same order, we pass its (r, c) setting as parameters to this sub-circuit so as to automatically transform and reduce the line into a reduced order model without having to rerun PACT. In this way, we effectively characterize lumped RC transmission lines in a way that allows them to be reduced on-the-fly without any expensive processing.

1 Introduction

Transmission lines are often modelled by a chain of lumped RLCG segments, as an approximation of the true distributed nature of the line parasitics. The larger the number of segments, the more accurate this approximation. We are interested in the case of an RC transmission line, where the series inductance and shunt conductance are set to zero. The `HSPICE` user guide [1, pg. 203] recommends as default a number of $N = 20$ segments for lumped element RC approximations of transmission lines. Beyond this, it says, one gets negligible improvement for the increased simulation time. It is interesting that this guidance is independent of the line length. The problem of modelling a transmission line with an efficiently-computable compact model is a model order reduction (MOR) problem. Given a high-accuracy lumped RC-chain model for a transmission line, one would like to reduce the order of the linear system that represents this model, from the given large value to a smaller value that is accurate enough for the frequency range of interest.

This problem is a special case of the general MOR problem for RC networks, which was pretty much solved in the 1990s. A very useful result from this line of research is the 1997 paper by Kerns and Yang [2], which describes the Pole Analysis via Congruence Transformations (PACT) approach, also described in [3] and [4]. This approach applies to networks with multiple ports. A *port* is a pair of nodes that belong to both the RC-network and the rest of the circuitry outside it. One of the two nodes is designated as the *reference node* for that port and may be shared, as reference, with other ports. The other node will be called the *terminal node* of that port. The voltage at the terminal node, relative to the reference node, is the *port voltage*, and the current going through the terminal node into the RC-network is the *port current.* A transmission line has two ports that share the same reference node as the rest of the circuit around it. The PACT method has many advantages, including structure-preservation, guaranteed passivity and stability in the reduced network, as well as having a simple way to synthesize an RC netlist for the reduced network.

The flow of the method includes, first the generation of a *transformed network* with the same number of nodes as the original, followed by a reduction of the transformed network into the *reduced network*, based on the magnitudes of its poles. Large poles, corresponding to small eigenvalues, are "dropped" by node removal. Because lumped RC transmission lines have the same r and c values in every segment, we have developed an approach in which we first use PACT to generate a *nominal/generic* transformed network, for a high segment count RC transmission line model, based on some nominal/generic $\tilde{r}$ and $\tilde{c}$ values. Then, we reuse this generic network, using a simple scaling and reduction process, to quickly generate the reduced network for any other given r and c, and frequency bandwidth, without requiring any further matrix operations.

2 Review of PACT

A general RC network can be modelled in the time domain using the *nodal equations*, $Gv(t) + C\dot{v}(t) = i(t)$, where G is the conductance matrix and C is the capacitance matrix; $v(t)$ is a vector of the voltages at all nodes of the RC network, while $i(t)$ is a vector of all the currents flowing into the network from the outside. This system can be so-ordered such that the first m rows correspond to all the port terminal nodes while the rest correspond to the remaining n nodes, called *internal nodes.* Thus, the first m entries of $v(t)$ are the port terminal voltages, the rest being the internal node voltages, and the first m entries of $i(t)$ are the port currents, while all the rest of its entries are zero. The corresponding Laplace domain formulation is

$$(G + sC)v(s) = i(s) \tag{1}$$

and it's useful to write this in the partitioned form,

$$\left(\begin{bmatrix} G_P & G_C^T \\ G_C & G_I \end{bmatrix} + s\begin{bmatrix} C_P & C_C^T \\ C_C & C_I \end{bmatrix}\right)\begin{bmatrix} v_P \\ v_I \end{bmatrix} = \begin{bmatrix} i_P \\ 0 \end{bmatrix}, \tag{2}$$

where the subscripts P and I stand for port nodes and internal nodes, respectively, and the matrices G_C and C_C represent the connections between the port and internal nodes. The problem to be solved is to reduce this system by removing some or all of the internal nodes, in such a way that the "view from the outside" via the ports remains approximately the same. One way to capture the "view from the outside" is via the admittance matrix $Y(s)$, defined as the matrix that solves the equation

$$Y(s)v_P(s) = i_P(s), \tag{3}$$

which can be found by a simple substitution step after writing the two sub-matrix equations in (2) separately,

$$\left(G_P + sC_P\right)v_P + \left(G_C^T + sC_C^T\right)v_I = i_P, \tag{4}$$

$$\left(G_C + sC_C\right)v_P + \left(G_I + sC_I\right)v_I = 0. \tag{5}$$

The second equation provides $v_I = -(G_I + sC_I)^{-1}(G_C + sC_C)v_P$ which, substituted in the first provides $(G_P + sC_P - (G_C + sC_C)^T (G_I + sC_I)^{-1} (G_C + sC_C))v_P = i_P$ so that [2],

$$Y(s) = G_P + sC_P - \left(G_C + sC_C\right)^T \left(G_I + sC_I\right)^{-1} \left(G_C + sC_C\right). \tag{6}$$

In the reduction process, these matrices will first be *transformed* in a couple of ways that preserve the original value of $Y(s)$, then some of them will be *reduced* in size by node elimination, which will result in a modified approximate $\hat{Y}(s) \approx Y(s)$, where the error in the approximation is under user control. Both transformations are called *congruence transformations* because they preserve the eigenvalues of the matrices in question, and they make use of two temporary variables,

$$A \triangleq G_I^{-1}G_C \quad \text{and} \quad B \triangleq C_C - C_I A. \tag{7}$$

An RC netlist can then be "reverse engineered" from this $\hat{Y}(s)$ by a process called *unstamping.* We will describe these steps below without fully justifying their correctness; the full justification is given in [2] and [3].

2.1 Transforming *G*

The first transformation focuses on the conductance matrix, with the goal of converting G_I into the identity matrix and eliminating the connection matrices G_C and G_C^T. It starts with a Cholesky factorization of G_I,

$$LL^T = G_I, \tag{8}$$

where L is a lower-triangular matrix, which is guaranteed to exist because G_I is positive definite. This is then used to construct the matrix

$$X = \begin{bmatrix} I & 0 \\ -A & L^{-T} \end{bmatrix} \tag{9}$$

that is used to perform a projection [5] of the system (1)(2), based on the change of variables $v(s) = Xv'(s)$ and the congruence transformations X^TGX and X^TCX, as follows. Replace $v(s)$ in (1) by $Xv'(s)$ and left-multiply both sides of the equation by X^T to get

$$(G' + sC')v'(s) = i'(s), \tag{10}$$

where $i'(s) = X^Ti(s)$,

$$G' = X^TGX = \begin{bmatrix} G_P - G_C^TA & 0 \\ 0 & L^{-1}G_IL^{-T} \end{bmatrix} \triangleq \begin{bmatrix} G'_P & 0 \\ 0 & I \end{bmatrix} \tag{11}$$

and

$$C' = X^TCX = \begin{bmatrix} C_P - B^TA - A^TC_C & B^TL^{-T} \\ L^{-1}B & L^{-1}C_IL^{-T} \end{bmatrix} \triangleq \begin{bmatrix} C'_P & C'^T_C \\ C'_C & C'_I \end{bmatrix}. \tag{12}$$

Notice that

$$v(s) = \begin{bmatrix} v_P \\ v_I \end{bmatrix} = X \begin{bmatrix} v'_P \\ v'_I \end{bmatrix} = \begin{bmatrix} v'_P \\ -Av'_P + L^{-T}v'_I \end{bmatrix} \quad \text{and} \quad i'(s) = \begin{bmatrix} i'_P \\ i'_I \end{bmatrix} = X^T \begin{bmatrix} i_P \\ 0 \end{bmatrix} = \begin{bmatrix} i_P \\ 0 \end{bmatrix}, \tag{13}$$

so that $v'_P = v_P$ and $i'_P = i_P$, i.e., the transformed system has the same port voltages and currents as the original system. The resulting, transformed system is

$$\left(\begin{bmatrix} G'_P & 0 \\ 0 & I \end{bmatrix} + s \begin{bmatrix} C'_P & C'^T_C \\ C'_C & C'_I \end{bmatrix} \right) \begin{bmatrix} v_P \\ v'_I \end{bmatrix} = \begin{bmatrix} i_P \\ 0 \end{bmatrix}. \tag{14}$$

The admittance matrix for this transformed system is

$$Y'(s) = G'_P + sC'_P - s^2C'^T_C \left(I + sC'_I\right)^{-1} C'_C, \tag{15}$$

and the authors [2, 3] assert that $Y'(s) = Y(s)$, consistent with the above observation that $v'_P = v_P$ and $i'_P = i_P$. This form of the system can be the end-point of the analysis, in the following sense. If one is interested in simply a port-to-port reduced circuit, without any internal nodes, then the the rightmost term in the above expression, i.e., the term multiplied by s^2 can be simply deleted, leaving an approximate admittance matrix based on only the first two moments, but this approximation is probably coarse in general. If having some internal nodes is of interest, one proceeds with the second transformation.

2.2 Transforming C

The second transformation focuses on the capacitance matrix, with the goal of diagonalizing C'_I so that its eigenvalues are explicitly available on the diagonal. Start with the eigenproblem $C'_IU = U\Lambda$, where Λ is a diagonal matrix whose diagonal entries are the eigenvalues λ_i of C'_I, which are guaranteed to be real and non-negative, and U is a square matrix whose columns are the corresponding eigenvectors, which are guaranteed to be real, with the additional property that U is orthonormal, i.e., $U^TU = I = UU^T$. For convenience, U and Λ are permuted so that the λ_i are sorted in decreasing order $\lambda_1 \geq \lambda_2 \geq \dots$ along the diagonal of Λ. Then, right-multiply $C'_IU = U\Lambda$ on both sides by U^T, leading to the symmetric eigendecomposition of C'_I,

$$U\Lambda U^T = C'_I. \tag{16}$$

With U in hand, we then construct the matrix

$$Z = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \tag{17}$$

that is used to perform a projection [5] of the system (10)(14), based on the change of variables $v'(s) = Zv''(s)$ and congruence transformations $Z^TG'Z$ and $Z^TC'Z$, as follows. Replace $v'(s)$ in (10) by $Zv''(s)$ and left-multiply both sides of the equation by Z^T to get

$$(G'' + sC'')v''(s) = i''(s), \tag{18}$$

where $i''(s) = Z^T i'(s)$,

$$G'' = Z^T G' Z = \begin{bmatrix} G'_P & 0 \\ 0 & U^T I U \end{bmatrix} = \begin{bmatrix} G'_P & 0 \\ 0 & I \end{bmatrix} = G' \tag{19}$$

and

$$C'' = Z^T C' Z = \begin{bmatrix} C'_P & C'^T_C U \\ U^T C'_C & U^T C'_I U \end{bmatrix} = \begin{bmatrix} C'_P & C'^T_C U \\ U^T C'_C & \Lambda \end{bmatrix} \triangleq \begin{bmatrix} C'_P & C''^T_C \\ C''_C & C''_I \end{bmatrix}, \tag{20}$$

so that $C''_I = \Lambda$ is the diagonal matrix containing the eigenvalues of C'_I. Notice that

$$v'(s) = \begin{bmatrix} v_P \\ v'_I \end{bmatrix} = Z \begin{bmatrix} v''_P \\ v''_I \end{bmatrix} = \begin{bmatrix} v''_P \\ U v''_I \end{bmatrix} \quad \text{and} \quad i''(s) = \begin{bmatrix} i''_P \\ i''_I \end{bmatrix} = Z^T \begin{bmatrix} i_P \\ 0 \end{bmatrix} = \begin{bmatrix} i_P \\ 0 \end{bmatrix}, \tag{21}$$

so that $v''_P = v'_P = v_P$ and $i''_P = i'_P = i_P$, i.e., the transformed system has the same port voltages and currents as the original system. The resulting, transformed system is in its final form

$$\left(\begin{bmatrix} G'_P & 0 \\ 0 & I \end{bmatrix} + s \begin{bmatrix} C'_P & C''^T_C \\ C''_C & \Lambda \end{bmatrix} \right) \begin{bmatrix} v_P \\ v''_I \end{bmatrix} = \begin{bmatrix} i_P \\ 0 \end{bmatrix}. \tag{22}$$

Note that v''_I is not the same as the original v_I, but represents the voltages at a new set of internal variables/nodes. The admittance matrix for this transformed system is

$$Y''(s) = G'_P + sC'_P - s^2 C''^T_C \left(I + s\Lambda\right)^{-1} C''_C, \tag{23}$$

and the authors [2, 3] assert that $Y''(s) = Y'(s) = Y(s)$, consistent with the above observation that $v''_P = v'_P = v_P$ and $i''_P = i'_P = i_P$. Furthermore, $Y(s)$ can be written in an alternate form [2] that illustrates the roles of the eigenvalues, as

$$Y(s) = G'_P + sC'_P - \sum_{i=1}^{n} \frac{s^2 r_i r_i^T}{1 + s\lambda_i}, \tag{24}$$

where r_i^T is the ith row of C''_C and λ_i is the eigenvalue that occupies the ith diagonal position of $C''_I = \Lambda$. At this point, one can start dropping eigenvalues with small magnitudes, relative to some threshold that's based on user specifications. Effectively, dropping an eigenvalue means to remove the corresponding term (which is a matrix) from the above summation, thereby eliminating one pole of the system. This is efficiently accomplished by removing the corresponding rows and columns from G'' and C'', as well as the corresponding variables from v''_I, in order to arrive at a reduced model with acceptable accuracy, i.e., with some new $\hat{Y}(s) \approx Y(s)$. The details of the error control mechanism are in [2, 3], and can be summarized as follows. The user specifies a frequency bandwidth of interest $\omega_c > 0$ and a relative (i.e., a fractional or percentage) error threshold ϵ_c, then finds the λ_c that solves the equation $\omega_c^3 \lambda_c^3 + \omega_c \lambda_c = \epsilon_c$. Then, if all the eigenvalues with $\lambda_i < \lambda_c$ are dropped, it is shown that the relative[1] error of each of the individual terms of the resulting admittance matrix is guaranteed to be less than ϵ_c, for all inputs with $|\omega| < \omega_c$. The cubic equation $x^3 + px - q = 0$, where p and q are real numbers, is called a *depressed cubic* and has a well-known solution, based on Cardano's method. One first computes the so-called *discriminant* $\Delta = 4p^3 + 27q^2$ and, if $\Delta > 0$, then the cubic has one real root and two complex roots, and the real root is

$$z = \sqrt[3]{\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} + \sqrt[3]{\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}. \tag{25}$$

In our case, $x = \omega_c \lambda_c$, $p = 1$ and $q = \epsilon_c > 0$, and we definitely have $\Delta = 4 + 27\epsilon_c^2 > 0$, so the real root is

$$z = \sqrt[3]{\frac{\epsilon_c}{2} + \sqrt{\left(\frac{\epsilon_c}{2}\right)^2 + \frac{1}{27}}} + \sqrt[3]{\frac{\epsilon_c}{2} - \sqrt{\left(\frac{\epsilon_c}{2}\right)^2 + \frac{1}{27}}}. \tag{26}$$

[1] The relative error is defined in relation to certain representative values on the diagonal. Details are in the paper and thesis.

Because $\epsilon_c/2 < \sqrt{(\epsilon_c/2)^2 + 1/27}$ then the term under the cubic root on the right is negative, so that

$$z = \sqrt[3]{\sqrt{\left(\frac{\epsilon_c}{2}\right)^2 + \frac{1}{27}} + \frac{\epsilon_c}{2}} - \sqrt[3]{\sqrt{\left(\frac{\epsilon_c}{2}\right)^2 + \frac{1}{27}} - \frac{\epsilon_c}{2}}, \tag{27}$$

from which, clearly $z > 0$ and we can find $\lambda_c = z/\omega_c > 0$. For example, if $\omega_c = 1$ MHz and $\epsilon_c = 10\%$, then $\lambda_c = 0.099025/1\text{E}6 \approx 99$ nF (recall, the λ_i are eigenvalues of C_I', and so have units of capacitance).

3 Transmission Line

It is common in the field to discretize a transmission line based on π-RC approximations of short segments of the original line. Let $\bar{r}$ and $\bar{c}$ be the per-unit-length resistance and capacitance of the line. For a short segment of the line of length δ, the corresponding π section would consist of a single resistor $r = \delta\bar{r}$ and a total capacitance of $c = \delta\bar{c}$ that is split into two capacitors of value $c/2$ to ground at each terminal of the resistor. As a result, the internal nodes of the line end up with a capacitor c each while the terminals get a capacitor $c/2$ each, as in the simple 3-segment example shown in Fig. 1, where nodes 1 and 2 are the port nodes while nodes 3 and 4 are the internal nodes. For our example, let $r = 1\text{k}\Omega$, so $g = 1\text{m}\mho$, and $c = 1\text{mF}$.

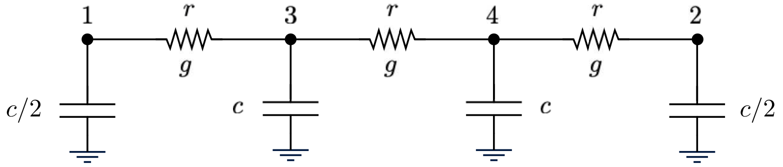


Figure 1: A simple 3-segment π-RC network for a transmission line.

The G and C circuit matrices are created automatically based on so-called *element stamps*, using a process called *stamping*, which is used in circuit simulators as part of *nodal analysis* [6]. A resistor with conductance $g \neq 0$ and resistance $r = 1/g$, that is connected between nodes i and j contributes (additively) the following "stamp" into the matrix G

$$\begin{array}{cccccc} & & i & & j & \\ & & \vdots & & \vdots & \\ i & \cdots & +g & \cdots & -g & \cdots \\ & & \vdots & & \vdots & \\ j & \cdots & -g & \cdots & +g & \cdots \\ & & \vdots & & \vdots & \end{array} \tag{28}$$

which is the contribution of this resistor to the equations resulting from the application of Kirchhoff's current law (KCL) at the two nodes i and j during nodal analysis. A similar stamping approach applies to the capacitors and produces the matrix C. These stamps are *additive* in the sense that they are added to the partially-built G and C matrices, as part of a sequential scan of all the circuit elements that starts with a zero matrix and leads to the full matrix. If an element is connected between node i and ground, then the above stamp is modified so that only the contribution at (i, i) needs to be included.

Based on this, the G and C matrices, in which the order of rows and columns follows the node numbers, are as follows,

$$G = \begin{bmatrix} g & 0 & -g & 0 \\ 0 & g & 0 & -g \\ -g & 0 & 2g & -g \\ 0 & -g & -g & 2g \end{bmatrix} = \begin{bmatrix} 1\text{m} & 0 & -1\text{m} & 0 \\ 0 & 1\text{m} & 0 & -1\text{m} \\ -1\text{m} & 0 & 2\text{m} & -1\text{m} \\ 0 & -1\text{m} & -1\text{m} & 2\text{m} \end{bmatrix} = \begin{bmatrix} G_P & G_C^T \\ G_C & G_I \end{bmatrix} \tag{29}$$

and

$$C = \begin{bmatrix} c/2 & 0 & 0 & 0 \\ 0 & c/2 & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & c \end{bmatrix} = \begin{bmatrix} 1\text{m}/2 & 0 & 0 & 0 \\ 0 & 1\text{m}/2 & 0 & 0 \\ 0 & 0 & 1\text{m} & 0 \\ 0 & 0 & 0 & 1\text{m} \end{bmatrix} = \begin{bmatrix} C_P & C_C^T \\ C_C & C_I \end{bmatrix}, \tag{30}$$

where all the sub-matrices are 2×2. Note that the ground node is *not* represented by a row & column in G and C, and is not part of the voltage vector in (1). Thus, the system equation in the time domain is

$$\begin{bmatrix} 1\text{m} & 0 & -1\text{m} & 0 \\ 0 & 1\text{m} & 0 & -1\text{m} \\ -1\text{m} & 0 & 2\text{m} & -1\text{m} \\ 0 & -1\text{m} & -1\text{m} & 2\text{m} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} + \begin{bmatrix} 1\text{m}/2 & 0 & 0 & 0 \\ 0 & 1\text{m}/2 & 0 & 0 \\ 0 & 0 & 1\text{m} & 0 \\ 0 & 0 & 0 & 1\text{m} \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{v}_3 \\ \dot{v}_4 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ 0 \\ 0 \end{bmatrix}. \tag{31}$$

With the matrices in hand, the workflow starts by finding A and B, based on

$$G_I = \begin{bmatrix} 2\text{m} & -1\text{m} \\ -1\text{m} & 2\text{m} \end{bmatrix} \implies G_I^{-1} = \begin{bmatrix} 2\text{k}/3 & 1\text{k}/3 \\ 1\text{k}/3 & 2\text{k}/3 \end{bmatrix}, \tag{32}$$

so that

$$A = G_I^{-1} G_C = \begin{bmatrix} 2\text{k}/3 & 1\text{k}/3 \\ 1\text{k}/3 & 2\text{k}/3 \end{bmatrix} \begin{bmatrix} -1\text{m} & 0 \\ 0 & -1\text{m} \end{bmatrix} = \begin{bmatrix} -2/3 & -1/3 \\ -1/3 & -2/3 \end{bmatrix} \tag{33}$$

and

$$B = C_C - C_I A = -C_I A = -\begin{bmatrix} 1\text{m} & 0 \\ 0 & 1\text{m} \end{bmatrix} \begin{bmatrix} -2/3 & -1/3 \\ -1/3 & -2/3 \end{bmatrix} = \begin{bmatrix} 2\text{m}/3 & 1\text{m}/3 \\ 1\text{m}/3 & 2\text{m}/3 \end{bmatrix}. \tag{34}$$

For the first transformation, we start with the Cholesky decomposition of G_I, which is

$$LL^T = G_I = \begin{bmatrix} 2\text{m} & -1\text{m} \\ -1\text{m} & 2\text{m} \end{bmatrix} \implies L = \begin{bmatrix} 1/(10\sqrt{5}) & 0 \\ -1/(20\sqrt{5}) & 1/(20\sqrt{5/3}) \end{bmatrix}, \tag{35}$$

from which,

$$L^{-1} = \begin{bmatrix} 10\sqrt{5} & 0 \\ 10\sqrt{5/3} & 20\sqrt{5/3} \end{bmatrix}. \tag{36}$$

We then find the transformed matrices, as

$$G'_P = G_P - G_C^T A = \begin{bmatrix} 1\text{m} & 0 \\ 0 & 1\text{m} \end{bmatrix} - \begin{bmatrix} -1\text{m} & 0 \\ 0 & -1\text{m} \end{bmatrix} \begin{bmatrix} -2/3 & -1/3 \\ -1/3 & -2/3 \end{bmatrix} = \begin{bmatrix} 1/3\text{k} & -1/3\text{k} \\ -1/3\text{k} & 1/3\text{k} \end{bmatrix}, \tag{37}$$

$$C'_P = C_P - B^T A - A^T C_C = \begin{bmatrix} 1\text{m}/2 & 0 \\ 0 & 1\text{m}/2 \end{bmatrix} - \begin{bmatrix} 2\text{m}/3 & 1\text{m}/3 \\ 1\text{m}/3 & 2\text{m}/3 \end{bmatrix} \begin{bmatrix} -2/3 & -1/3 \\ -1/3 & -2/3 \end{bmatrix} = \begin{bmatrix} 19\text{m}/18 & 4\text{m}/9 \\ 4\text{m}/9 & 19\text{m}/18 \end{bmatrix}, \tag{38}$$

$$C'_C = L^{-1} B = \begin{bmatrix} 10\sqrt{5} & 0 \\ 10\sqrt{5/3} & 20\sqrt{5/3} \end{bmatrix} \begin{bmatrix} 2\text{m}/3 & 1\text{m}/3 \\ 1\text{m}/3 & 2\text{m}/3 \end{bmatrix} = \begin{bmatrix} 1/(30\sqrt{5}) & 1/(60\sqrt{5}) \\ 1/(15\sqrt{15}) & 1/(12\sqrt{15}) \end{bmatrix} \tag{39}$$

and

$$C'_I = L^{-1} C_I L^{-T} = \begin{bmatrix} 10\sqrt{5} & 0 \\ 10\sqrt{5/3} & 20\sqrt{5/3} \end{bmatrix} \begin{bmatrix} 1\text{m} & 0 \\ 0 & 1\text{m} \end{bmatrix} \begin{bmatrix} 10\sqrt{5} & 10\sqrt{5/3} \\ 0 & 20\sqrt{5/3} \end{bmatrix} = \begin{bmatrix} 1/2 & 1/(2\sqrt{3}) \\ 1/(2\sqrt{3}) & 5/6 \end{bmatrix}. \tag{40}$$

For the second transformation, we start with the eigendecomposition of C'_I, which is

$$U \Lambda U^T = C'_I \implies \Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 1/3 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix}, \tag{41}$$

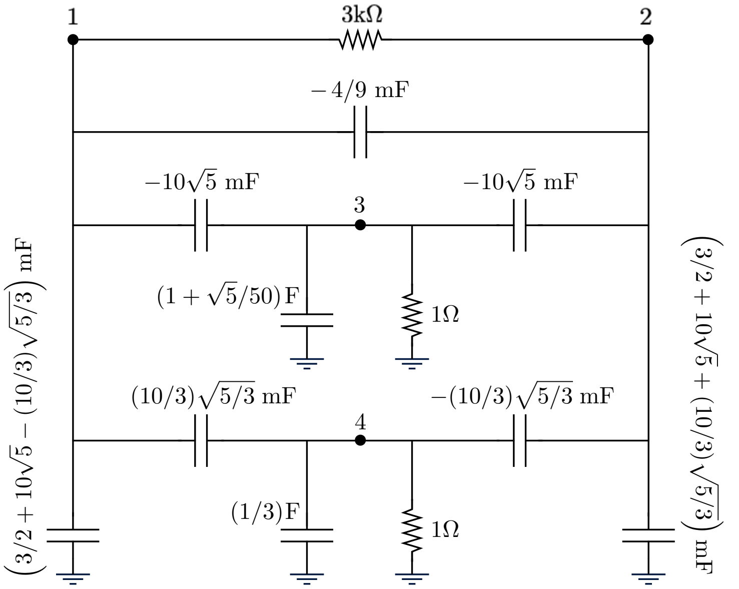


Figure 2: Transformed circuit for the 3-segment π-RC network.

from which,

$$C''_C = U^T C'_C = \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1/(30\sqrt{5}) & 1/(60\sqrt{5}) \\ 1/(15\sqrt{15}) & 1/(12\sqrt{15}) \end{bmatrix} = \begin{bmatrix} 1/(20\sqrt{5}) & 1/(20\sqrt{5}) \\ -1/(60\sqrt{15}) & 1/(60\sqrt{15}) \end{bmatrix}. \tag{42}$$

The final form of the transformed system equation is

$$\left(\begin{bmatrix} 1/3\text{k} & -1/3\text{k} & 0 & 0 \\ -1/3\text{k} & 1/3\text{k} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + s \begin{bmatrix} 19\text{m}/18 & 4\text{m}/9 & 1/(20\sqrt{5}) & -1/(60\sqrt{15}) \\ 4\text{m}/9 & 19\text{m}/18 & 1/(20\sqrt{5}) & 1/(60\sqrt{15}) \\ 1/(20\sqrt{5}) & 1/(20\sqrt{5}) & 1 & 0 \\ -1/(60\sqrt{15}) & 1/(60\sqrt{15}) & 0 & 1/3 \end{bmatrix} \right) \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ 0 \\ 0 \end{bmatrix}. \tag{43}$$

The resulting transformed circuit is shown in Fig. 2, and it's easy to tell by inspection (by a process of stamping) that this circuit produces the transformed system equation (43), benefiting from $1/(20\sqrt{5}) = 10\sqrt{5}\,\text{m}$ and $1/(60\sqrt{15}) = (10/3)\sqrt{5/3}\,\text{m}$. The circuit includes two internal nodes marked '3' and '4', but these are not the same original internal node 3 and 4. The circuit includes a $3\,\text{k}\Omega$ resistor between the two ports, so that the DC behaviour of the original circuit is maintained; this is due to the terms in G'_P. There are also two $1\,\Omega$ resistors to ground, at nodes 3 and 4 due to the 1 terms in G'', but they do not affect the DC behaviour because there is no DC path from the new nodes 3 and 4 to the ports. It may be surprising to see the very large $(1/3)\,\text{F}$ and $(1+\sqrt{5}/50) \approx 1\,\text{F}$ capacitors in parallel with the $1\,\Omega$ resistors, but the products of these parallel RC combinations are comparable to the $\approx 1\,\text{sec}$ RC time constant of the elements in the original circuit. It may also be surprising to see negative capacitors in the circuit, but this is fairly common to see in model order reduction, and it's not problematic for modern circuit simulators.

If one would like to drop the smaller eigenvalue, i.e., $\lambda_2 = 1/3$, the corresponding row 4 in the system equation and column 4 in G'' and C'' are removed, and the system becomes

$$\left(\begin{bmatrix} 1/3\text{k} & -1/3\text{k} & 0 \\ -1/3\text{k} & 1/3\text{k} & 0 \\ 0 & 0 & 1 \end{bmatrix} + s \begin{bmatrix} 19\text{m}/18 & 4\text{m}/9 & 10\sqrt{5}\,\text{m} \\ 4\text{m}/9 & 19\text{m}/18 & 10\sqrt{5}\,\text{m} \\ 10\sqrt{5}\,\text{m} & 10\sqrt{5}\,\text{m} & 1 \end{bmatrix} \right) \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ 0 \end{bmatrix}. \tag{44}$$

The resulting reduced circuit is shown in Fig. 3, and it's again easy to tell by inspection that this circuit corresponds to the reduced system equation (44).

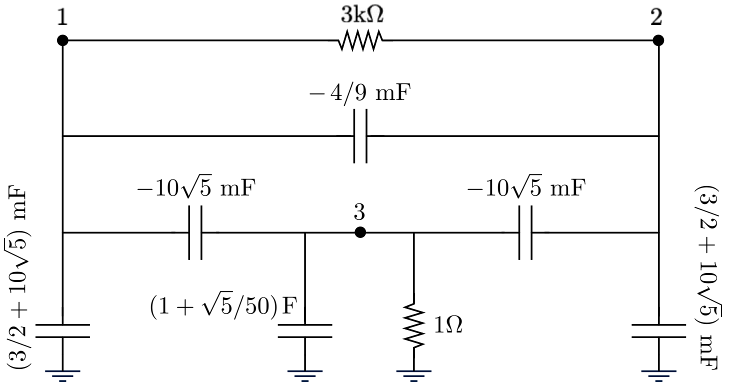


Figure 3: Reduced circuit for the transmission line RC network.

3.1 Unstamping

The reduced RC circuit in Fig. 3 was generated by a process of *unstamping* [7, 4, 5], which consists of a reversal of the steps taken during stamping, and which can be very simply described, for the general case of a reduced RC network for a lumped RC transmission line between port nodes 1 and 2, as follows. For the reduced conductance matrix G'', with entries g_{ij}, the value $-g_{1,2}$ is the conductance value to be added in the reduced network between nodes 1 and 2; so, a resistor with resistance $-1/g_{1,2}$ will do the job. Then, a 1Ω resistor should be added from every internal node to ground, i.e., from nodes $3, 4, \ldots$ to ground. As for the capacitance, let c_{ij} be the matrix entries of C'' and let C_{ij}^{ckt} be the capacitor to be connected between nodes i and j in the circuit, with C_{ii}^{ckt} denoting the value of the capacitor from node i to ground. First, the algebraic sum of the entries in every row i provides the value of a capacitor that should be added from the corresponding node i to ground. Thus, the product $C''\mathbb{1}$ gives all the node-to-ground capacitor values, where $\mathbb{1}$ is a vector of all-ones. Second, for every non-zero c_{ij} above the diagonal, i.e. with $i < j$, add a capacitor of value $-c_{ij}$ between nodes i and j in the circuit. In other words,

$$C_{ij}^{\text{ckt}} = -c_{ij} \qquad \text{and} \qquad C_{ii}^{\text{ckt}} = \sum_{j=1}^{n} c_{ij}. \tag{45}$$

3.2 Direct circuit reduction

We have seen how reducing the size of the transformed circuit can be achieved by deleting the corresponding row of the system equation and the corresponding columns of its matrices. But it would be useful to see how the reduction can be carried out directly in the circuit domain. How can the transformed circuit in Fig. 2 be directly converted to the reduced circuit in Fig. 3, without having to build and manipulate the system matrices? Here too, we are focused on the general case of a reduced RC network for a lumped RC transmission line between port nodes 1 and 2. Firstly, the resistance values are not affected, except that the resistors from internal nodes to ground will need to be removed for any internal nodes that correspond to eigenvalues to be dropped. Secondly, any capacitors connected to internal nodes will need to be removed, for any node that corresponds to an eigenvalue to be dropped. In addition, the values of the two capacitors from the ports to ground will be affected, as follows. Let $i \in \{1, 2\}$ be any of the two port nodes, and let k be an internal node that corresponds to an eigenvalue that is a candidate for being dropped, and notice that

$$C_{ii}^{\text{ckt}} = c_{i1} + c_{i2} + \sum_{j>2} c_{ij} = c_{i1} + c_{i2} - \sum_{j>2} C_{ij}^{\text{ckt}} = c_{i1} + c_{i2} - \sum_{j>2, j\neq k} C_{ij}^{\text{ckt}} - C_{ik}^{\text{ckt}}, \tag{46}$$

so that

$$c_{i1} + c_{i2} - \sum_{j>2, j\neq k} C_{ij}^{\text{ckt}} = C_{ii}^{\text{ckt}} + C_{ik}^{\text{ckt}}. \tag{47}$$

However, the new capacitor value $\hat{C}_{ii}^{\text{ckt}}$ once column k in the matrix has been removed would be

$$\hat{C}_{ii}^{\text{ckt}} = c_{i1} + c_{i2} + \sum_{j>2, j\neq k} c_{ij} = c_{i1} + c_{i2} - \sum_{j>2, j\neq k} C_{ij}^{\text{ckt}}, \tag{48}$$

so that

$$\hat{C}_{ii}^{\text{ckt}} = C_{ii}^{\text{ckt}} + C_{ik}^{\text{ckt}}. \tag{49}$$

For example, considering removal of node 4 in Fig. 2, the port to ground capacitor values would become

$$\hat{C}_{11}^{\text{ckt}} = \left(3/2 + 10\sqrt{5} - (10/3)\sqrt{5/3} + (10/3)\sqrt{5/3}\right)\text{mF} = (3/2 + 10\sqrt{5})\,\text{mF} \tag{50}$$

$$\hat{C}_{22}^{\text{ckt}} = \left(3/2 + 10\sqrt{5} + (10/3)\sqrt{5/3} - (10/3)\sqrt{5/3}\right)\text{mF} = (3/2 + 10\sqrt{5})\,\text{mF}, \tag{51}$$

hence the values in Fig. 3. All other capacitance values are not affected. *It is interesting and convenient that all the above reductions can be achieved by the single step of grounding any internal nodes that are designated for removal.* That would *effectively* achieve the removal the resistor and capacitor to ground at any node k that is to be removed, as well as automatically causing the coupling capacitors from these nodes to the port nodes to become in parallel with the original port-to-ground capacitors, consistent with (49).

4 Parameterized Reduced Circuit

We are interested in RC networks whose topology is fixed but whose values are subject to some scaling. Transmission lines are a special case of such networks. The scale factors will be useful as *parameters*, so as to adapt the reduced circuit directly for a certain context, without having to redo the factorizations and transformations. So, consider an RC network whose G and C matrices can be written as

$$G = \alpha_g \tilde{G} \quad \text{and} \quad C = \alpha_c \tilde{C}, \tag{52}$$

where $\alpha_g > 0$ and $\alpha_c \neq 0$ are unitless scale factors or parameters, while $\tilde{G}$ and $\tilde{C}$ are regular conductance and capacitance matrices for some generic or nominal RC network, in standard (SI) units. If we have previously generated the $\tilde{G}''$ and $\tilde{C}''$ corresponding to $\tilde{G}$ and $\tilde{C}$ using the PACT factorizations and transformations, how can we quickly generate G'' and C'' for any given α_g and α_c? The answer to this question is simple enough but tedious to arrive at, as follows. If we apply the PACT transformations to both the (G, C) and $(\tilde{G}, \tilde{C})$ networks, we would find for a start that

$$A = G_I^{-1} G_C = \left(\alpha_g^{-1} \tilde{G}_I^{-1}\right)\left(\alpha_g \tilde{G}_C\right) = \tilde{G}_I^{-1} \tilde{G}_C = \tilde{A}. \tag{53}$$

So, the scaling does not affect the unitless matrix $A = \tilde{A}$, but it does affect most other matrices, as

$$B = C_C - C_I A = \alpha_c \left(\tilde{C}_C - \tilde{C}_I \tilde{A}\right) = \alpha_c \tilde{B}, \tag{54}$$

$$G'_P = G_P - G_C^T A = \alpha_g \left(\tilde{G}_P - \tilde{G}_C^T \tilde{A}\right) = \alpha_g \tilde{G}'_P \tag{55}$$

and

$$C'_P = C_P - B^T A - A^T C_C = \alpha_c \left(\tilde{C}_P - \tilde{B}^T \tilde{A} - \tilde{A}^T \tilde{C}_C\right) = \alpha_c \tilde{C}'_P. \tag{56}$$

For the Cholesky factorization of G_I,

$$LL^T = G_I = \alpha_g \tilde{G}_I = \alpha_g \tilde{L}\tilde{L}^T, \tag{57}$$

so that

$$L = \sqrt{\alpha_g}\tilde{L}, \tag{58}$$

which allows us to write

$$C'_C = L^{-1}B = \frac{\alpha_c}{\sqrt{\alpha_g}}\tilde{L}^{-1}\tilde{B} = \frac{\alpha_c}{\sqrt{\alpha_g}}\tilde{C}'_C \tag{59}$$

and

$$C'_I = L^{-1}C_I L^{-T} = \frac{1}{\sqrt{\alpha_g}}\tilde{L}^{-1}\alpha_c\tilde{C}_I\frac{1}{\sqrt{\alpha_g}}\tilde{L}^{-T} = \frac{\alpha_c}{\alpha_g}\tilde{L}^{-1}\tilde{C}_I\tilde{L}^{-T} = \frac{\alpha_c}{\alpha_g}\tilde{C}'_I. \tag{60}$$

Considering the eigendecomposition of C'_I, we have

$$C'_I U = U\Lambda = \frac{\alpha_c}{\alpha_g}\tilde{C}'_I U \quad \Longrightarrow \quad \tilde{C}'_I U = \frac{\alpha_g}{\alpha_c}U\Lambda \tag{61}$$

so that

$$\Lambda = \frac{\alpha_c}{\alpha_g}\tilde{\Lambda} \quad \text{and} \quad U = \tilde{U}. \tag{62}$$

This allows us to complete the analysis for the remaining parts of C'', as

$$C''_C = U^T C'_C = \tilde{U}^T\frac{\alpha_c}{\sqrt{\alpha_g}}\tilde{C}'_C = \frac{\alpha_c}{\sqrt{\alpha_g}}\tilde{C}''_C \tag{63}$$

and

$$C''_I = U^T C'_I U = \tilde{U}^T\frac{\alpha_c}{\alpha_g}\tilde{C}'_I\tilde{U} = \frac{\alpha_c}{\alpha_g}\tilde{C}''_I. \tag{64}$$

To summarize,

$$G'' = \begin{bmatrix} \alpha_g\tilde{G}''_P & 0 \\ 0 & I \end{bmatrix} \tag{65}$$

$$C'' = \begin{bmatrix} \alpha_c\tilde{C}''_P & (\alpha_c/\sqrt{\alpha_g})\tilde{C}''^T_C \\ (\alpha_c/\sqrt{\alpha_g})\tilde{C}''_C & (\alpha_c/\alpha_g)\tilde{C}''_I \end{bmatrix}. \tag{66}$$

Given $\tilde{G}''$ and $\tilde{C}''$ from a prior transformation of a generic RC network, we can use the above (65) and (66) to directly compute G'' and C'' for any given α_g and α_c, then drop the insignificant eigenvalues and generate the reduced circuit by the simple process of unstamping that we saw earlier.

For example, consider again the same 3-segment RC transmission line as in Fig. 1, but now let us use $\tilde{r} = 1\Omega$ and $\tilde{c} = 1\text{F}$ as the generic/nominal design point. We can run the `MATLAB` script shown in Fig. 4 to arrive at the *transformed generic* system

$$\tilde{G}'' = \begin{bmatrix} 0.333333 & -0.333333 & 0 & 0 \\ -0.333333 & 0.333333 & 0 & 0 \\ 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 1.00 \end{bmatrix} \tag{67}$$

$$\tilde{C}'' = \begin{bmatrix} 1.055555 & 0.444444 & 0.707107 & -0.136083 \\ 0.444444 & 1.055555 & 0.707107 & 0.136083 \\ 0.707107 & 0.707107 & 1.00 & 0.00 \\ -0.136083 & 0.136083 & 0.00 & 0.333333 \end{bmatrix}. \tag{68}$$

These matrices can be stored as a template to be used whenever one encounters a 3-segment transmission line of this kind. For comparison with the earlier example, we can scale these matrices using (65) and (66) based on $\alpha_g = \alpha_c = 10^{-3}$, to get

$$G'' = \begin{bmatrix} 0.333333\times 10^{-3} & -0.333333\times 10^{-3} & 0 & 0 \\ -0.333333\times 10^{-3} & 0.333333\times 10^{-3} & 0 & 0 \\ 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 1.00 \end{bmatrix} \tag{69}$$

```
NS = 3;
g = 1;
c = 1;

N = NS + 1;
M = N - 2;

GP = g*eye(2);
GC = zeros(M,2);
GC(1,1)= -g;
GC(M,2) = -g;
GI = diag(2*g*ones(1,M)) ...
+ diag(-g*ones(1,M-1),1) + diag(-g*ones(1,M-1),-1);
G = [GP, GC'; GC, GI];

CP = (c/2)*eye(2);
CC = zeros(M,2);
CI = c*eye(M);
C = [CP, CC'; CC, CI];

A = GI\GC;
GP1 = GP - GC'*A;
B = CC - CI*A;
CP1 = CP - B'*A - A'*CC;
L = chol(GI,'lower');
CC1 = L\B;
CI1 = L\CI/L';
[U, Lambda] = eig(CI1);
[vec, perm] = sort(diag(Lambda),'descend');
Lambda = Lambda(perm, perm);
U = U(:, perm);
CC2 = U'*CC1;

G2 = [GP1, zeros(2,M); zeros(M,2), eye(M)];
C2 = [CP1, CC2'; CC2, Lambda];
```

Figure 4: `MATLAB` script for PACT for a generic 3-segment transmission line.

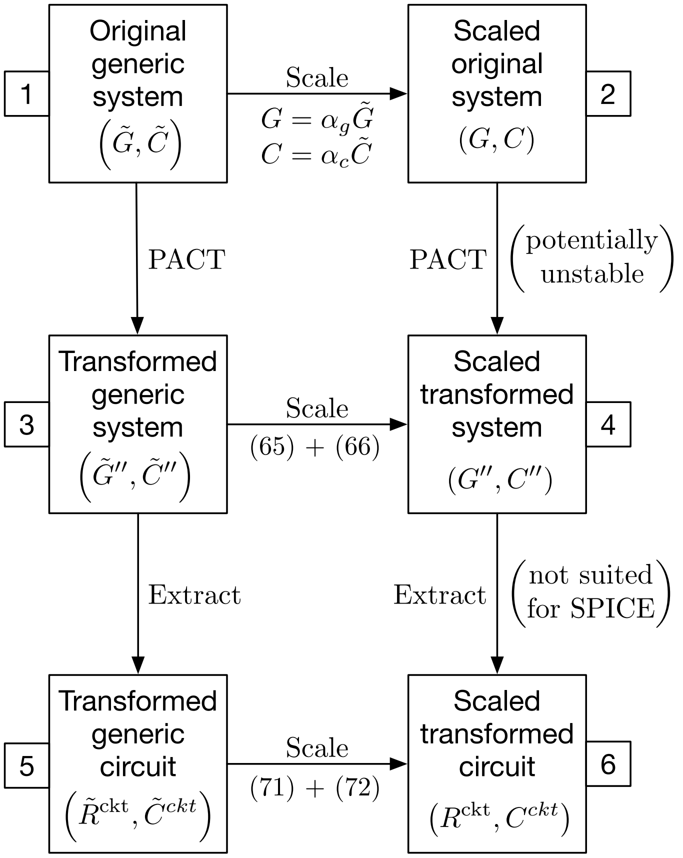
and

$$C'' = \begin{bmatrix} 1.055555 \times 10^{-3} & 0.444444 \times 10^{-3} & 0.022361 & -0.004303 \\ 0.444444 \times 10^{-3} & 1.055555 \times 10^{-3} & 0.022361 & 0.004303 \\ 0.022361 & 0.022361 & 1.00 & 0.00 \\ -0.004303 & 0.004303 & 0.00 & 0.333333 \end{bmatrix}, \tag{70}$$

which are the same as the matrices in (43). *Because α_g and α_c are unitless and we use $\tilde{r} = 1\Omega$ and $\tilde{c} = 1F$ as the generic/nominal design point, then α_g and α_c are equal to the (unitless) values of g and c.*

4.1 Direct circuit scaling

Once a transformed circuit is available, such as that in Fig. 2, it can be scaled for any given α_g and α_c based on (65) and (66). This can be done either directly on the matrix, as indicated, or on the circuit itself by suitably scaling the values of all resistors and capacitors, as will be briefly explained with the help of Fig. 5. The figure shows six different formulations of the system/circuit; those on the left are for the generic case and on the right for the scaled case. The best pathway through the diagram is 1 → 3 → 5 → 6. We have found that scaling after PACT gives much better numerical stability than, for example, setting g and c to more realistic values (that will typically be very small, such as 10^{-6} and 10^{-3}) and then running the PACT script in Fig. 4. And the step 5 → 6 is much easier to capture in a `SPICE` circuit description than 4 → 6. We have already described the steps 1 → 3 → 5 and it remains to explain 5 → 6, which scales a circuit directly, without reference to the matrices. Let $\tilde{R}_{ij}^{\text{ckt}}$ and $\tilde{C}_{ij}^{\text{ckt}}$ be resistors and capacitors in the transformed generic circuit (box 5 in Fig. 5) while R_{ij}^{ckt} and C_{ij}^{ckt} are resistors and capacitors in the scaled transformed circuit



1
Original generic system
$\left(\tilde{G}, \tilde{C}\right)$
Scale
$G = \alpha_g \tilde{G}$
$C = \alpha_c \tilde{C}$
Scaled original system
(G, C)
2
PACT
PACT
$\begin{pmatrix}\text{potentially}\\ \text{unstable}\end{pmatrix}$
3
Transformed generic system
$\left(\tilde{G}'', \tilde{C}''\right)$
Scale
(65) + (66)
Scaled transformed system
(G'', C'')
4
Extract
Extract
$\begin{pmatrix}\text{not suited}\\ \text{for SPICE}\end{pmatrix}$
5
Transformed generic circuit
$\left(\tilde{R}^{\text{ckt}}, \tilde{C}^{ckt}\right)$
Scale
(71) + (72)
Scaled transformed circuit
$\left(R^{\text{ckt}}, C^{ckt}\right)$
6

Figure 5: Multiple paths of transformation, scaling and extraction.

(box 6). The scaling of resistors is easy; it simply follows the scaling in (65), so that $R_{12}^{\text{ckt}} = \tilde{R}_{12}^{\text{ckt}}/\alpha_g$, while all other resistors remain the same at 1Ω. As for the capacitors, note that, for $i \neq j$, $\tilde{C}_{ij}^{\text{ckt}} = -\tilde{c}_{ij}$ in $\tilde{C}''$ and $C_{ij}^{\text{ckt}} = -c_{ij}$ in C'', so that, for both $i, j \leq 2$, $C_{ij}^{\text{ckt}} = \alpha_c \tilde{C}_{ij}^{\text{ckt}}$, while otherwise, $C_{ij}^{\text{ckt}} = (\alpha_c/\sqrt{\alpha_g})\tilde{C}_{ij}^{\text{ckt}}$. As for the case $i = j$, the node-to-ground capacitors can be scaled as follows. If $i \leq 2$, then

$$\begin{aligned} C_{ii}^{\text{ckt}} &= c_{i1} + c_{i2} + \sum_{j>2} c_{ij} = \alpha_c \left(\tilde{c}_{i1} + \tilde{c}_{i2}\right) + \sum_{j>2} \frac{\alpha_c}{\sqrt{\alpha_g}} \tilde{c}_{ij} \\ &= \alpha_c \left(\tilde{c}_{i1} + \tilde{c}_{i2} + \sum_{j>2} \tilde{c}_{ij}\right) - \alpha_c \sum_{j>2} \tilde{c}_{ij} + \sum_{j>2} \frac{\alpha_c}{\sqrt{\alpha_g}} \tilde{c}_{ij} = \alpha_c \tilde{C}_{ii}^{\text{ckt}} + \left(\alpha_c - \frac{\alpha_c}{\sqrt{\alpha_g}}\right) \sum_{j>2} \tilde{C}_{ij}^{\text{ckt}}, \end{aligned} \tag{71}$$

otherwise,

$$\begin{aligned} C_{ii}^{\text{ckt}} &= c_{ii} + \sum_{j \neq i} c_{ij} = \frac{\alpha_c}{\alpha_g} \tilde{c}_{ii} + \sum_{j \neq i} \frac{\alpha_c}{\sqrt{\alpha_g}} \tilde{c}_{ij} = \frac{\alpha_c}{\alpha_g} \left(\tilde{c}_{ii} + \sum_{j \neq i} \tilde{c}_{ij}\right) - \frac{\alpha_c}{\alpha_g} \sum_{j \neq i} \tilde{c}_{ij} + \sum_{j \neq i} \frac{\alpha_c}{\sqrt{\alpha_g}} \tilde{c}_{ij} \\ &= \frac{\alpha_c}{\alpha_g} \tilde{C}_{ii}^{\text{ckt}} + \left(\frac{\alpha_c}{\alpha_g} - \frac{\alpha_c}{\sqrt{\alpha_g}}\right) \sum_{j \neq i} \tilde{C}_{ij}^{\text{ckt}}. \end{aligned} \tag{72}$$

The resulting circuit can then be reduced by removing some internal nodes, along with any resistors or capacitors directly connected to them, starting from the bottom, i.e., corresponding to the smallest eigenvalues. As described earlier, this can be achieved by simply grounding any internal nodes that are to be removed.

5 Workflow

Given the guidance from the HSPICE documentation that 20 segments is the default and reasonable setting for discretizing a general transmission line, we will adopt a 20-segment lumped π-RC line as the standard model. For this case, we will now review and consolidate the workflow that allows one to arrive at a parameterized reduced transformed circuit. We will refer to this circuit/test case as **PIRC20**.

5.1 The starting point

For a *generic* RC transmission line with 20 π-RC segments based on $\tilde{g} = 1\mho$ and $\tilde{c} = 1\text{F}$, the PACT script seen earlier in Fig. 4 can be run with $\texttt{NS} = 20$ and $g = c = 1$, and the matrices G and C, as constructed in the first section of this script, would constitute the system corresponding to block 1 in Fig. 5.

5.2 The G2 and C2 matrices

The PACT script generates the matrices G2 and C2 which are the conductance and capacitance matrices of the transformed circuit, given by G'' and C'' in (19) and (20). In this case, we get

$$\texttt{G2} = \begin{bmatrix} \begin{bmatrix} 0.05 & -0.05 \\ -0.05 & 0.05 \end{bmatrix} & 0 \\ 0 & I \end{bmatrix}, \tag{73}$$

while the first two columns of the symmetric C2 are shown in the left-hand-side of (74) and its diagonal is given as the right-hand-side vector.

$$\texttt{C2(:,1:2)} = \begin{bmatrix} 6.6750000000000 & 3.3250000000000 \\ 3.3250000000000 & 6.6750000000000 \\ 12.8030239865711 & 12.8030239865711 \\ 3.1907667152466 & -3.1907667152466 \\ 1.4105907045333 & 1.4105907045333 \\ -0.7873749722376 & 0.7873749722376 \\ 0.4987421044063 & 0.4987421044063 \\ 0.3417648255157 & -0.3417648255157 \\ 0.2469082445098 & 0.2469082445098 \\ 0.1851229586822 & -0.1851229586822 \\ -0.1425268516803 & -0.1425268516802 \\ -0.1118033988750 & 0.1118033988750 \\ -0.0887959982498 & -0.0887959982498 \\ -0.0709975569564 & 0.0709975569564 \\ 0.0568189936237 & 0.0568189936237 \\ -0.0452090108906 & 0.0452090108906 \\ -0.0354445100453 & -0.0354445100453 \\ -0.0270090756738 & 0.0270090756738 \\ -0.0195192097317 & -0.0195192097317 \\ 0.0126774701283 & -0.0126774701282 \\ 0.0062411556326 & 0.0062411556327 \end{bmatrix} \qquad \texttt{diag(C2)} = \begin{bmatrix} 6.6750000000000 \\ 6.6750000000000 \\ 40.6119096993972 \\ 10.2158645472653 \\ 4.5874305436788 \\ 2.6180339887499 \\ 1.7071067811865 \\ 1.2129599990798 \\ 0.9157349821963 \\ 0.7236067977500 \\ 0.5927221766165 \\ 0.5000000000000 \\ 0.4323634543204 \\ 0.3819660112501 \\ 0.3438812014862 \\ 0.3149040459206 \\ 0.2928932188135 \\ 0.2763932022500 \\ 0.2644094526404 \\ 0.2562714077342 \\ 0.2515484896643 \end{bmatrix}. \tag{74}$$

These matrices G2 and C2 constitute the system corresponding to block 3 in Fig. 5 and will be the basis for the circuit constructions given below.

5.3 Transformed generic circuit

We will now construct the transformed version of the generic circuit. For PIRC20, the transformed generic circuit is shown in Fig. 8, and it corresponds to block 5 in Fig. 5. The structure of these circuits is always the same, arising from the structure of the G'' and C'' matrices, and the element values are found by unstamping as follows. The resistor values are easy to find. The resistor between nodes 1 and 2 has a resistance of $-1/g''_{1,2}$, where g''_{ij} are the elements of G''. In this case, this resistance is $1/0.05 = 20\Omega$, from (73). As for the resistors from internal nodes to ground, their values are each equal to the sum of the elements in the corresponding row of G'', which is 1Ω, from (73). The coupling capacitor between nodes 1 and 2 is found, using (45), as $-c''_{1,2} = -c''_{2,1}$, where c''_{ij} are the elements of the matrix C''. In this case, this capacitance is -3.3250 F, from (74). The coupling capacitors between node 1 and the various internal nodes $3, 4, \ldots, 21$ can each be found as the negative of the corresponding entry of the 1st column of C'', shown as the 1st column in (74). Likewise, the coupling capacitors between node 2 and the various internal nodes $3, 4, \ldots, 21$ can each be found as the negative of the corresponding entry of the 2nd column of C'', given in (74). Finally, the capacitances from every node to ground can be found as the sum of the elements in the corresponding row of C'', due to (45), which is the following vector that can be found as $C''\mathbb{1}$ using the MATLAB command

$$\texttt{C2*ones(N,1)} = \begin{bmatrix} 27.4239765745091 \\ 22.0481006646304 \\ 66.2179576725394 \\ 10.2158645472653 \\ 7.4086119527454 \\ 2.6180339887499 \\ 2.7045909899992 \\ 1.2129599990798 \\ 1.4095514712158 \\ 0.7236067977500 \\ 0.3076684732560 \\ 0.5000000000000 \\ 0.2547714578208 \\ 0.3819660112501 \\ 0.4575191887336 \\ 0.3149040459206 \\ 0.2220041987228 \\ 0.2763932022500 \\ 0.2253710331770 \\ 0.2562714077342 \\ 0.2640308009296 \end{bmatrix}. \tag{75}$$

5.4 Parameterized transformed circuit

With the transformed generic circuit in hand, it is then very useful to construct a SPICE sub-circuit for the corresponding scaled (i.e., parameterized) circuit. This would constitute the circuit in block 6 in Fig. 5. The SPICE sub-circuit is called TRC20 and is listed in section A of the Appendix. It takes as input the (unitless) values $\texttt{RX} = r/\tilde{r}$ and $\texttt{CX} = c/\tilde{c}$ for any 20-segment π-RC transmission line and includes built-in values for all the elements in the transformed generic sub-circuit in Fig. 8. As pointed out earlier, because we have chosen $\tilde{r} = 1\Omega$ and $\tilde{c} = 1$F as the generic/nominal design point, then RX and CX are effectively the values (without units) of r and c. These are used to compute $\alpha_g = 1/\texttt{RX}$ and $\alpha_c = \texttt{CX}$. The circuit listing is fairly long, because it first sets up about 40 parameters that capture the transformed generic circuit, then generates the scaled (parameterized) sub-circuit based on the provided r and c values, and also offers the option to eliminate any desired internal nodes by the very simple manual step of changing a 1 to a 0, as we will now describe.

The script starts by defining a few parameters that represent α_g, α_c and similar, which will be used to implement the scaling. Then, a number of parameters named TC0100, TC0200, ... are defined to represent the capacitors to ground in the transformed generic circuit of Fig. 8; the TC stands for tilde-C, as in the $\tilde{C}''$ notation introduced in section 4. Then, the *absolute value* (magnitude) of the coupling capacitance between the two port nodes is captured by the parameter TC0102. This is followed by the introduction of 19 parameters named TC03, TC04, ... that hold the *magnitudes* of the coupling capacitors from port nodes to internal nodes. There are in fact $2 \times 19 = 38$ such capacitors, as is clear from the circuit diagram, but every pair of them have the same magnitude. So, for example, TC03 will be used to compute the capacitance between node 1 and node 3, as well as the capacitance between node 2 and node 3. This part of the script then finishes up with the computation of the sum SUM01, which carries the sum of all the capacitors from port node 1 to internal nodes (in Fig. 8), while SUM02 does the same for port node 2. These will be used later in the script to compute the two port-to-ground capacitors in the scaled circuit.

The rest of the script provides the SPICE statements that create the scaled circuit, including performing the computation of various element values, First the port-to-port resistor and capacitor values are computed as R0102 and C0102. The expression for C0102 is the result of simply scaling TC0102 by α_c, with the appropriate sign from the circuit diagram, based on the top-left part of C'' in (66). The expression for R0102 should be similar, but instead we have used a shortcut that makes use of the NSEG parameter, as the resistor between nodes 1 and 2 in Fig. 8 is simply 1Ω multiplied by 20 (the number of segments, NSEG). The script then includes a section that creates the resistors to ground from all the internal nodes, all of which are set to 1Ω, consistent with the bottom-right part of G'' in (65). A key feature of this script is that it can be used to check (by trial and error) which internal nodes one needs to keep, while eliminating all others, in order to get reasonable accuracy. Any internal node that is to be eliminated should have the value of its resistor to ground manually set to 0, in *this* portion of the script, because this has the effect of grounding that node, consistent with the discussion at the end of section 3.2. The next section of the script creates the capacitors from every node to ground. For the first two capacitors, C0100 and C0200, which are from the two port nodes to ground, they are computed based on (71) using SUM01 and SUM02, while the rest (C0300, C0400, etc.) are found based on (72), using the parameters TC03, TC03, etc. defined earlier. Notice that the signs applied to TC03, TC04, etc. follow the signs of the corresponding coupling capacitors in Fig. 8. The final two sections of the script create and compute the coupling capacitors in the scaled circuit, first those from port node 1 to all internal nodes, then those from port node 2 to all internal nodes, based on the diagonal blocks of C'' in (66) and making use again of the TC03, etc. parameters with appropriate signs from the circuit diagram.

5.5 Reduced parameterized transformed circuit

The last step of the process is to reduce the scaled transformed sub-circuit by removing certain internal nodes. This requires an accuracy check by running various test cases, before one can settle on a choice for the reduced model. We are interested in transmission lines with a fairly slow response, so we have tested the case where $r = 880\text{k}\Omega$ and $c = 151\text{mF}$. We have found that removing all internal nodes gives poor results, keeping only the largest eigenvalue (node N03) does not help much at all, keeping N04 only seems to work just as well as keeping both N03 and N04, providing a very good approximation of the dynamic response. We refer to the case when only N04 is kept as the PACTN4 case. It only remains to generate the circuit diagram for the PACTN4 reduced circuit. The easiest way of doing this is to go back to the C'' and G'' matrices for the transformed generic circuit, i.e., the result of section 5.2, then remove all the rows and columns from both matrices that correspond to all internal nodes except N04, and then unstamp that circuit and apply scaling to it as was done in section 5.4. Applying this to our PIRC20 case, the resulting reduced C'' is

$$\begin{bmatrix} 6.6750000000000 & 3.3250000000000 & 3.1907667152466 \\ 3.3250000000000 & 6.6750000000000 & -3.1907667152466 \\ 3.1907667152466 & -3.1907667152466 & 10.2158645472653 \end{bmatrix} \tag{76}$$

which when scaled using (66) becomes

$$\begin{bmatrix} 6.6750000000000\alpha_c & 3.3250000000000\alpha_c & 3.1907667152466\alpha_c/\sqrt{\alpha_g} \\ 3.3250000000000\alpha_c & 6.6750000000000\alpha_c & -3.1907667152466\alpha_c/\sqrt{\alpha_g} \\ 3.1907667152466\alpha_c/\sqrt{\alpha_g} & -3.1907667152466\alpha_c/\sqrt{\alpha_g} & 10.2158645472653\alpha_c/\alpha_g \end{bmatrix}. \tag{77}$$

Similar treatment for G'' and making use of (65) gives the scaled reduced matrix

$$\begin{bmatrix} \alpha_g/20 & -\alpha_g/20 & 0 \\ -\alpha_g/20 & \alpha_g/20 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{78}$$

Unstamping these matrices gives the reduced circuit shown in Fig. 6, where the capacitors to ground are found based on the product of the matrix in (77) by the vector $\mathbb{1}$, which gives the vector

$$\begin{bmatrix} 10\alpha_c + 3.1907667152466\alpha_c/\sqrt{\alpha_g} \\ 10\alpha_c - 3.1907667152466\alpha_c/\sqrt{\alpha_g} \\ 10.2158645472653\alpha_c/\alpha_g \end{bmatrix} . \tag{79}$$

The resulting reduced sub-circuit can be expressed with the SPICE snippet shown in Fig. 7.

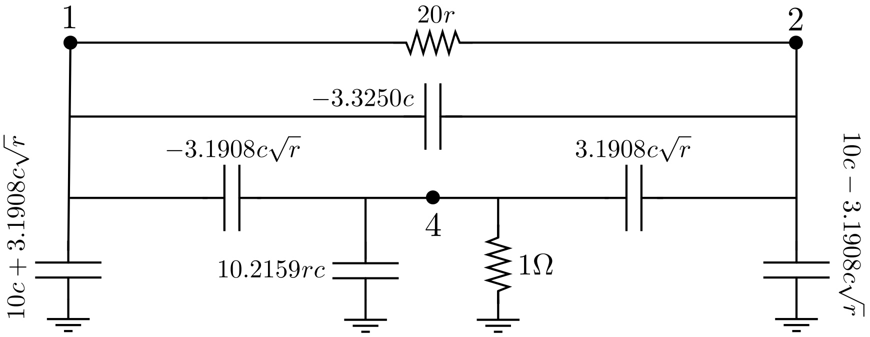


Figure 6: The reduced circuit for the 20-segment π-RC line when only N04 is kept.

```
C10 N01 GND '10*CX + 3.1907667152466*CX*SQRT(RX)'
C20 N02 GND '10*CX - 3.1907667152466*CX*SQRT(RX)'
R12 N01 N02 '20*RX'
C12 N01 N02 '-3.3250*CX'
C14 N01 N04 '-3.1907667152466*CX*SQRT(RX)'
C24 N02 N04 ' 3.1907667152466*CX*SQRT(RX)'
C40 N04 GND '10.2158645472653*RX*CX'
R40 N04 GND 1
```

Figure 7: SPICE snippet capturing the parameterized reduced circuit PACTN4.

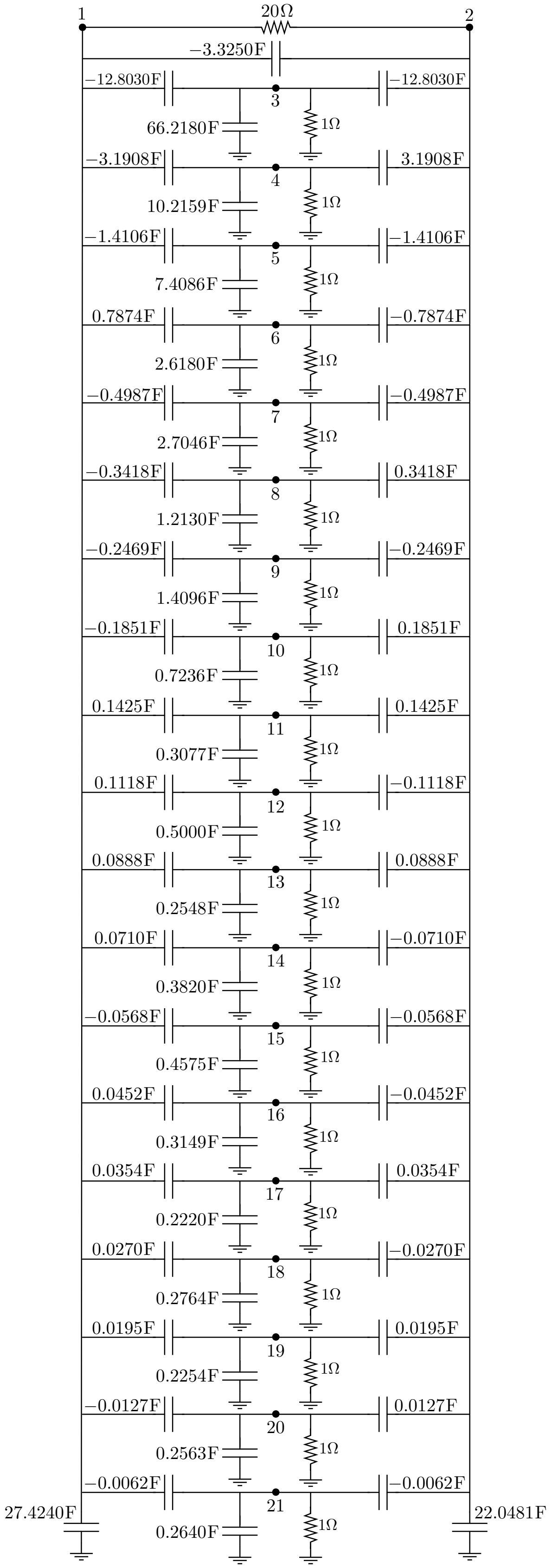


Figure 8: Transformed generic circuit for a 20-segment π-RC transmission line.

Appendix

A The TRC20 SPICE sub-circuit for PIRC20

```
****************************************************************************
** Sub-circuit for a PACT-transformed 20-segment PI-RC transmission line ***

.SUBCKT TRC20 N01 N02 RX=880k CX=151m

* We start by defining the transformed generic RC sub-circuit for this line.
* First, some key coefficients:
*
.PARAM NRES = 20        * The number of RC segments in this generic circuit.
.PARAM ALPHA_G = '1/RX'
.PARAM ALPHA_C = CX
.PARAM K1 = 'ALPHA_C/ALPHA_G'
.PARAM K2 = 'ALPHA_C/SQRT(ALPHA_G)'
.PARAM K3 = 'K1 - K2'

* Then capacitors to ground in the transformed generic circuit:
*
.PARAM TC0100 = 27.423976574509147
.PARAM TC0200 = 22.048100664630397
.PARAM TC0300 = 66.217957672539427
.PARAM TC0400 = 10.215864547265337
.PARAM TC0500 =  7.408611952745449
.PARAM TC0600 =  2.618033988749895
.PARAM TC0700 =  2.704590989999189
.PARAM TC0800 =  1.212959999079799
.PARAM TC0900 =  1.409551471215791
.PARAM TC1000 =  0.723606797749982
.PARAM TC1100 =  0.307668473255971
.PARAM TC1200 =  0.499999999999999
.PARAM TC1300 =  0.254771457820824
.PARAM TC1400 =  0.381966011250105
.PARAM TC1500 =  0.457519188733557
.PARAM TC1600 =  0.314904045920622
.PARAM TC1700 =  0.222004198722772
.PARAM TC1800 =  0.276393202250029
.PARAM TC1900 =  0.225371033177039
.PARAM TC2000 =  0.256271407734233
.PARAM TC2100 =  0.264030800929579

* MAGNITUDE of coupling capacitor between the port nodes in the generic ckt:
*
.PARAM TC0102 = 3.325000000000013

* MAGNITUDE of coupling capacitors connected to internal nodes in generic ckt:
*
.PARAM TC03 = 12.803023986571095
.PARAM TC04 =  3.190766715246610
```

```
.PARAM TC05 =  1.410590704533310
.PARAM TC06 =  0.787374972237641
.PARAM TC07 =  0.498742104406320
.PARAM TC08 =  0.341764825515694
.PARAM TC09 =  0.246908244509763
.PARAM TC10 =  0.185122958682192
.PARAM TC11 =  0.142526851680255
.PARAM TC12 =  0.111803398874989
.PARAM TC13 =  0.088795998249806
.PARAM TC14 =  0.070997556956411
.PARAM TC15 =  0.056818993623664
.PARAM TC16 =  0.045209010890557
.PARAM TC17 =  0.035444510045341
.PARAM TC18 =  0.027009075673769
.PARAM TC19 =  0.019519209731663
.PARAM TC20 =  0.012677470128251
.PARAM TC21 =  0.006241155632650

* Two key sums of coupling capacitors from each port node to internal node in
* the generic ckt, used to find the two capacitors from port nodes to ground
* in the scaled circuit
*
.PARAM SUM01 = '- TC03 - TC04 - TC05 + TC06 - TC07 - TC08 - TC09 - TC10 \
+ TC11 + TC12 + TC13 + TC14 - TC15 + TC16 + TC17 + TC18 + TC19 - TC20 - TC21'
.PARAM SUM02 = '- TC03 + TC04 - TC05 - TC06 - TC07 + TC08 - TC09 + TC10 \
+ TC11 - TC12 + TC13 - TC14 - TC15 - TC16 + TC17 - TC18 + TC19 + TC20 - TC21'

***************************************
* We now define the new (scaled) sub-circuit based on the given RX and CX
***************************************
* The two R and C elements connected between the two port nodes in scaled ckt
*
R0102 N01 N02 'NRES/ALPHA_G'
C0102 N01 N02 '- ALPHA_C * TC0102'

* The 1 Ohm resistors from each internal node to ground in the scaled circuit.
* Change the 1 to 0 for any nodes to be removed in reduced ckt.
*
R0300 N03 GND 1
R0400 N04 GND 1
R0500 N05 GND 1
R0600 N06 GND 1
R0700 N07 GND 1
R0800 N08 GND 1
R0900 N09 GND 1
R1000 N10 GND 1
R1100 N11 GND 1
R1200 N12 GND 1
R1300 N13 GND 1
R1400 N14 GND 1
R1500 N15 GND 1
```

```
R1600 N16 GND 1
R1700 N17 GND 1
R1800 N18 GND 1
R1900 N19 GND 1
R2000 N20 GND 1
R2100 N21 GND 1

* The capacitors from each node to ground in the new scaled circuit
*
C0100 N01 GND ’ALPHA_C * TC0100   + (ALPHA_C - K2) * SUM01’
C0200 N02 GND ’ALPHA_C * TC0200   + (ALPHA_C - K2) * SUM02’
C0300 N03 GND ’K1 * TC0300 + K3 * (- TC03 - TC03)’
C0400 N04 GND ’K1 * TC0400 + K3 * (- TC04 + TC04)’
C0500 N05 GND ’K1 * TC0500 + K3 * (- TC05 - TC05)’
C0600 N06 GND ’K1 * TC0600 + K3 * (  TC06 - TC06)’
C0700 N07 GND ’K1 * TC0700 + K3 * (- TC07 - TC07)’
C0800 N08 GND ’K1 * TC0800 + K3 * (- TC08 + TC08)’
C0900 N09 GND ’K1 * TC0900 + K3 * (- TC09 - TC09)’
C1000 N10 GND ’K1 * TC1000 + K3 * (- TC10 + TC10)’
C1100 N11 GND ’K1 * TC1100 + K3 * (  TC11 + TC11)’
C1200 N12 GND ’K1 * TC1200 + K3 * (  TC12 - TC12)’
C1300 N13 GND ’K1 * TC1300 + K3 * (  TC13 + TC13)’
C1400 N14 GND ’K1 * TC1400 + K3 * (  TC14 - TC14)’
C1500 N15 GND ’K1 * TC1500 + K3 * (- TC15 - TC15)’
C1600 N16 GND ’K1 * TC1600 + K3 * (  TC16 - TC16)’
C1700 N17 GND ’K1 * TC1700 + K3 * (  TC17 + TC17)’
C1800 N18 GND ’K1 * TC1800 + K3 * (  TC18 - TC18)’
C1900 N19 GND ’K1 * TC1900 + K3 * (  TC19 + TC19)’
C2000 N20 GND ’K1 * TC2000 + K3 * (- TC20 + TC20)’
C2100 N21 GND ’K1 * TC2100 + K3 * (- TC21 - TC21)’

* The coupling capacitors connected from the port node N01 to internal nodes
* in the new scaled circuit
*
C0103 N01 N03 ’- K2 * TC03’
C0104 N01 N04 ’- K2 * TC04’
C0105 N01 N05 ’- K2 * TC05’
C0106 N01 N06 ’  K2 * TC06’
C0107 N01 N07 ’- K2 * TC07’
C0108 N01 N08 ’- K2 * TC08’
C0109 N01 N09 ’- K2 * TC09’
C0110 N01 N10 ’- K2 * TC10’
C0111 N01 N11 ’  K2 * TC11’
C0112 N01 N12 ’  K2 * TC12’
C0113 N01 N13 ’  K2 * TC13’
C0114 N01 N14 ’  K2 * TC14’
C0115 N01 N15 ’- K2 * TC15’
C0116 N01 N16 ’  K2 * TC16’
C0117 N01 N17 ’  K2 * TC17’
C0118 N01 N18 ’  K2 * TC18’
C0119 N01 N19 ’  K2 * TC19’
```

```
C0120 N01 N20 ’- K2 * TC20’
C0121 N01 N21 ’- K2 * TC21’

* The coupling capacitors connected from the port node N02 to internal nodes
* in the new scaled circuit
*
C0203 N02 N03 ’- K2 * TC03’
C0204 N02 N04 ’  K2 * TC04’
C0205 N02 N05 ’- K2 * TC05’
C0206 N02 N06 ’- K2 * TC06’
C0207 N02 N07 ’- K2 * TC07’
C0208 N02 N08 ’  K2 * TC08’
C0209 N02 N09 ’- K2 * TC09’
C0210 N02 N10 ’  K2 * TC10’
C0211 N02 N11 ’  K2 * TC11’
C0212 N02 N12 ’- K2 * TC12’
C0213 N02 N13 ’  K2 * TC13’
C0214 N02 N14 ’- K2 * TC14’
C0215 N02 N15 ’- K2 * TC15’
C0216 N02 N16 ’- K2 * TC16’
C0217 N02 N17 ’  K2 * TC17’
C0218 N02 N18 ’- K2 * TC18’
C0219 N02 N19 ’  K2 * TC19’
C0220 N02 N20 ’  K2 * TC20’
C0221 N02 N21 ’- K2 * TC21’

.IC N01=0 N02=0 N03=0 N04=0 N05=0 N06=0 N07=0 N08=0 N09=0 N10=0
+ N11=0 N12=0 N13=0 N14=0 N15=0 N16=0 N17=0 N18=0 N19=0 N20=0 N21=0

.ENDS
******************* End of Sub-circuit for TRC20 ************************
```

References

[1] *HSPICE® Signal Integrity User Guide, X-2005-09.* Mountain View, CA, USA: Synopsys, Inc., 2005.

[2] K. J. Kerns and A. T. Yang, “Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 734–744, Jul. 1997.

[3] K. J. Kerns, “Accurate and Stable Reduction of RLC Networks Using Split Congruence Transformations,” PhD Thesis, University of Washington, Seattle, Washington, 1996.

[4] R. Ionutiu, “Model Order Reduction for Multi-terminals Systems : with Applications to Circuit Simulation.” PhD Thesis, Technische Universiteit Eindhoven, Jan. 2011, iSBN: 978-90-386-2580-5.

[5] S. Grivet-Talocia and B. Gustavsen, *Passive Macromodeling*, ser. Wiley Series in Microwave and Optical Engineering. Hoboken, New Jersey: John Wiley & Sons, 2016.

[6] F. N. Najm, *Circuit Simulation.* Hoboken, NJ: John Wiley & Sons, Inc., 2010.

[7] F. Yang, X. Zeng, Y. Su, and D. Zhou, “RLCSYN: RLC equivalent circuit synthesis for structure-preserved reduced-order model of interconnect,” in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 2710–2713.