NOCKIT - Theory and Simulation doc

Samuel & Guy

September 29, 2020

1 Writing an ansatz for voltage and current in each segment

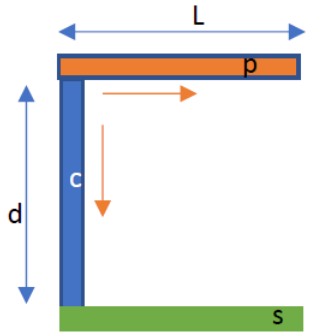

Figure 1: a unit cell of a ladder network, composed of three segments p, s and c. p and s denote the primary and secondary lines. c denotes the coupling line. the blue arrows indicate the dimensions of the unit-cell and the orange arrows indicate the origin and positive direction of the coordinates defined for each segment.

Each unit cell is made out of three segments p,s and c Where P denotes the primary line, S denotes the secondary line, and C denotes the coupling line. We can write the voltage and the current in each segment as a guess of the form (eq. 2.14a and 2.14b in Pozar):

$$V(x) = te^{ikx} + re^{-ikx} \tag{1}$$

$$I(x) = \frac{1}{Z_0}\left[te^{ikx} - re^{-ikx}\right] \tag{2}$$

several notes regarding these expressions:

1. note that our $+i$ equals Pozar's $-j$.
2. If we assume a given frequency, then k and Z_0 are known properties of the lines, so they can only take one of two values: one value for the main lines and a different (and known) value for the coupling lines.

3. The variables t and r are not reflectance and transmission coefficients. They are unknown variables, and in principle we have $3N$ r's and $3N$ t's , one for each segment in the network. We will denote them as: $r_p^n, t_p^n, r_s^n, t_s^n, r_c^n, t_c^n$.

4. We assume that the voltage is normalized to the input signal amplitude, so r and t are dimensionless.

5. Each segment has it's own coordinate x, which goes from 0 to L in the case of the main lines (p,s) and from 0 to d in the case of the coupling lines (c).

In the light of these remarks, let's rewrite (1) and (2) for each segment in the nth unit cell:

$$V_p^n(x) = t_p^n e^{ikx} + r_p^n e^{-ikx} \tag{3}$$
$$V_s^n(x) = t_s^n e^{ikx} + r_s^n e^{-ikx} \tag{4}$$
$$V_c^n(x) = t_c^n e^{ik_c x} + r_c^n e^{-ik_c x} \tag{5}$$

$$I_p^n(x) = \frac{1}{Z_0}\left[t_p^n e^{ikx} - r_p^n e^{-ikx}\right] \tag{6}$$
$$I_s^n(x) = \frac{1}{Z_0}\left[t_s^n e^{ikx} - r_s^n e^{-ikx}\right] \tag{7}$$
$$I_c^n(x) = \frac{1}{Z_c}\left[t_c^n e^{ik_c x} - r_c^n e^{-ik_c x}\right] \tag{8}$$

where Z_c is the characteristic impedance of the coupling lines, and k_c is their wave number.

2 writing equations for the r's and the t's :

Now we have to require continuity of the voltage and conservation of current in the two nodes of the cell. Note that each voltage continuity requirement is actually two (independent) equations. For the upper node :

$$V_p^n(0) = V_p^{n-1}(L) \tag{9}$$
$$V_p^n(0) = V_c^n(0) \tag{10}$$

for the lower node :

$$V_s^n(0) = V_s^{n-1}(L) \tag{11}$$
$$V_s^n(0) = V_c^n(d) \tag{12}$$

The current equations:

$$0 = I_p^{n-1}(L) - I_p^n(0) - I_c^n(0) \tag{13}$$
$$0 = I_s^{n-1}(L) - I_s^n(0) + I_c^n(d) \tag{14}$$

plugging (3) - (8) into (9) - (14) , we get:

$$t_p^n + r_p^n = t_p^{n-1} e^{ikL} + r_p^{n-1} e^{-ikL} \tag{15}$$
$$t_p^n + r_p^n = t_c^n + r_c^n \tag{16}$$
$$t_s^n + r_s^n = t_s^{n-1} e^{ikL} + r_s^{n-1} e^{-ikL} \tag{17}$$
$$t_s^n + r_s^n = t_c^n e^{ik_c d} + r_c^n e^{-ik_c d} \tag{18}$$
$$0 = \frac{1}{Z_0}\left[t_p^{n-1} e^{ikL} - r_p^{n-1} e^{-ikL}\right] - \frac{1}{Z_0}\left[t_p^n - r_p^n\right] - \frac{1}{Z_c}\left[t_c^n - r_c^n\right] \tag{19}$$
$$0 = \frac{1}{Z_0}\left[t_s^{n-1} e^{ikL} - r_s^{n-1} e^{-ikL}\right] - \frac{1}{Z_0}\left[t_s^n - r_s^n\right] + \frac{1}{Z_c}\left[t_c^n e^{ik_c d} - r_c^n e^{-ik_c d}\right] \tag{20}$$

This is true for $2 \leq n \leq N$, as long as we use the boundary conditions :

$$t_p^1 = 1, \;\; t_s^1 = 0 \tag{21}$$

$$r_p^N = r_s^N = 0 \tag{22}$$

These assume an incident voltage signal with amplitude 1 in the primary line and no input signals to the secondary line. The first cell has no coupling segment, so the variables t_c^1 and r_c^1 do not exist. the primary lines are assumed to continue to infinity after the Nth cell and before the 1st. This means that we have $6\,(N-1)$ linear equations for $6\,(N-1)$ variables.

3 Solving the equations

3.1 method I: numerically solve $6\,(N-1)$ equations for $6\,(N-1)$ variables

in order to solve $6\,(N-1)$ linear equations for $6\,(N-1)$ variables, we want to encode the system of equation in matrix form:

$$M\vec{x} = \vec{v} \tag{23}$$

where M is a matrix, $\vec{v}$ is a column vector and $\vec{x}$ is the column vector of solutions. We'll start by encoding the 6 equations describing one intersection between two unit cell. rewriting (15)-(20) such that the left side is zero:

$$0 = t_p^n + r_p^n - t_p^{n-1} e^{ikL} - r_p^{n-1} e^{-ikL} \tag{24}$$
$$0 = t_p^n + r_p^n - t_c^n - r_c^n \tag{25}$$
$$0 = t_s^n + r_s^n - t_s^{n-1} e^{ikL} - r_s^{n-1} e^{-ikL} \tag{26}$$
$$0 = t_s^n + r_s^n - t_c^n e^{ik_c d} - r_c^n e^{-ik_c d} \tag{27}$$
$$0 = \frac{1}{Z_0}\left[t_p^{n-1} e^{ikL} - r_p^{n-1} e^{-ikL}\right] - \frac{1}{Z_0}\left[t_p^n - r_p^n\right] - \frac{1}{Z_c}\left[t_c^n - r_c^n\right] \tag{28}$$
$$0 = \frac{1}{Z_0}\left[t_s^{n-1} e^{ikL} - r_s^{n-1} e^{-ikL}\right] - \frac{1}{Z_0}\left[t_s^n - r_s^n\right] + \frac{1}{Z_c}\left[t_c^n e^{ik_c d} - r_c^n e^{-ik_c d}\right] \tag{29}$$

note that for these six equations there are 10 variables, so in order to write in in the form (23) we need a 6×10 matrix M. We will use the following order for the variables:

$$\vec{x} = \left(t_p^{n-1}, r_p^{n-1}, t_s^{n-1}, r_s^{n-1}, t_c^n, r_c^n, t_p^n, r_p^n, t_s^n, r_s^n\right) \tag{30}$$

such that :

$$M = \begin{pmatrix} -e^{ikL} & -e^{-ikL} & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -e^{ikL} & -e^{-ikL} & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -e^{ik_c d} & -e^{-ik_c d} & 0 & 0 & 1 & 1 \\ \frac{1}{Z_0}e^{ikK} & -\frac{1}{Z_0}e^{-ikL} & 0 & 0 & -\frac{1}{Z_c} & \frac{1}{Z_c} & -\frac{1}{Z_0} & \frac{1}{Z_0} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_0}e^{ikL} & -\frac{1}{Z_0}e^{-ikL} & \frac{1}{Z_c}e^{ik_c d} & -\frac{1}{Z_c}e^{-ik_c d} & 0 & 0 & -\frac{1}{Z_0} & \frac{1}{Z_0} \end{pmatrix} \tag{31}$$

choosing the following order for **all** the variables:

$$\vec{x} = \left(t_p^1, r_p^1 t_s^1, r_s^1, t_c^2, r_c^2, t_p^2, r_p^2 t_s^2, r_s^2, ..., t_p^{n-1}, r_p^{n-1}, t_s^{n-1}, r_s^{n-1}, t_c^n, r_c^n, t_p^n, r_p^n t_s^n, r_s^n\right)$$

note that there are $6N-2$ variables (since there are no r_c^1 and t_c^1) of which 4 are known by the boundary conditions. We can build the matrix M for the entire problem by pasting copies of the single intersection matrix (31) with the correct shifts into a $(6N-6) \times (6N-2)$ matrix. Then we add 4 more trivial rows (rows that are already solved) to encode the boundary conditions (the 4 known variables). So our matrix will look something like figure2 .

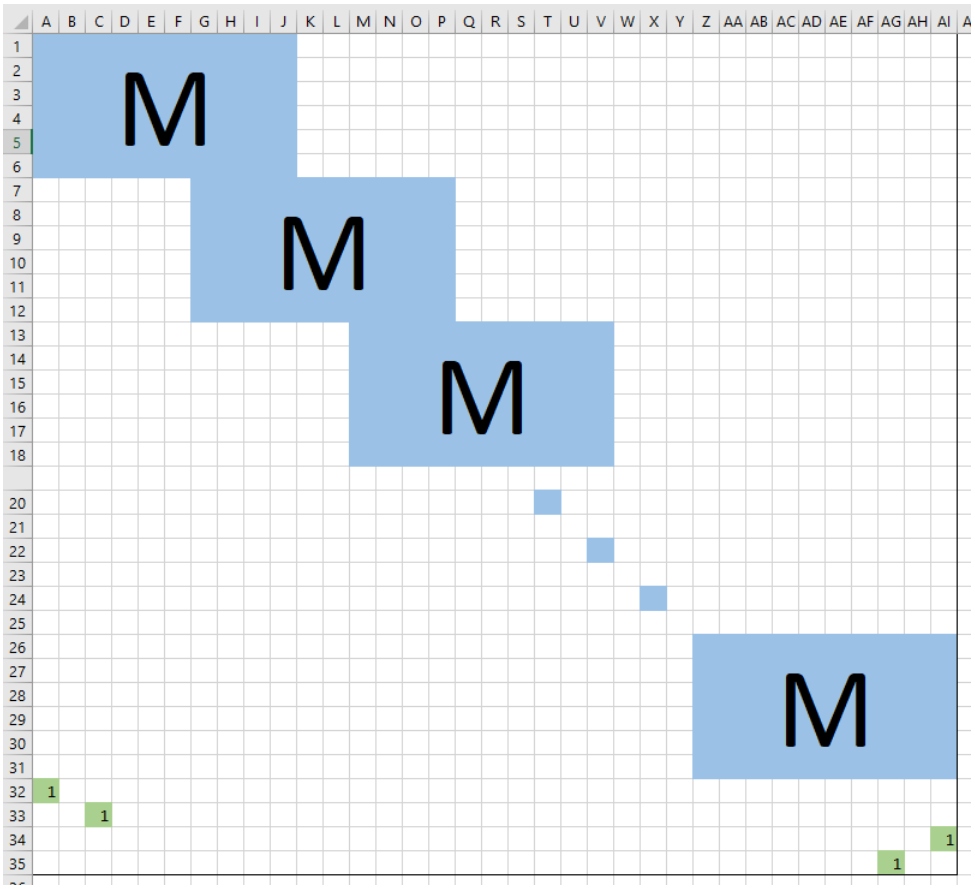

Figure 2: a schematic of the total $(6N-6)\times(6N-2)$ matrix encoding our system of linear equations, constructed by pasting copies of the single-intersection 6×10 matrix M (in blue) with the appropriate shifts, and adding 4 trivial rows at the bottom (in green).

the vector $\vec{v}$ is all zeros except of the component corresponding to the boundary condition $t_p^1 = 1$.

3.2 method II : try to express t_i^n, r_i^n in terms of t_i^{n-1},r_i^{n-1}:

note: the bottom line is: we used method I. Method II looked promising but we didn't find a way to make it work, a more detailed discussion below:

We look at eqns (24)-(29) as if every t_i^{n-1} and r_i^{n-1} is known. We will want to write these 6 equations in the form $M\vec{x} = \vec{v}$ where

$$\vec{x} = \left(r_p^n, t_p^n, r_s^n, t_s^n, r_c^n, t_c^n\right)$$

is the (column) vector of solutions. re-arranging:

$$r_p^n + t_p^n = t_p^{n-1} e^{ikL} + r_p^{n-1} e^{-ikL} \tag{32}$$

$$r_p^n + t_p^n - r_c^n - t_c^n = 0 \tag{33}$$

$$r_s^n + t_s^n = t_s^{n-1} e^{ikL} + r_s^{n-1} e^{-ikL} \tag{34}$$

$$r_s^n + t_s^n - r_c^n e^{-ik_c d} - t_c^n e^{ik_c d} = 0 \tag{35}$$

$$\left[-r_p^n + t_p^n\right] + \frac{Z_0}{Z_c}\left[-r_c^n + t_c^n\right] = \left[t_p^{n-1} e^{ikL} - r_p^{n-1} e^{-ikL}\right] \tag{36}$$

$$\left[-r_s^n + t_s^n\right] - \frac{Z_0}{Z_c}\left[-r_c^n e^{-ik_c d} + t_c^n e^{ik_c d}\right] = \left[t_s^{n-1} e^{ikL} - r_s^{n-1} e^{-ikL}\right] \tag{37}$$

or in matrix form:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -e^{-ik_c d} & -e^{ik_c d} \\ -1 & 1 & 0 & 0 & -\frac{Z_0}{Z_c} & \frac{Z_0}{Z_c} \\ 0 & 0 & -1 & 1 & \frac{Z_0}{Z_c} e^{-ik_c d} & -\frac{Z_0}{Z_c} e^{+ik_c d} \end{pmatrix} \begin{pmatrix} r_p^n \\ t_p^n \\ r_s^n \\ t_s^n \\ r_c^n \\ t_c^n \end{pmatrix} = \begin{pmatrix} t_p^{n-1} e^{ikL} + r_p^{n-1} e^{-ikL} \\ 0 \\ t_s^{n-1} e^{ikL} + r_s^{n-1} e^{-ikL} \\ 0 \\ t_p^{n-1} e^{ikL} - r_p^{n-1} e^{-ikL} \\ t_s^{n-1} e^{ikL} - r_s^{n-1} e^{-ikL} \end{pmatrix}$$

now we can use it to construct an iterative numerical solution in Matlab, or we can solve it analytically in Mathematica:

$$\begin{pmatrix} r_p^n \\ t_p^n \\ r_s^n \\ t_s^n \\ r_c^n \\ t_c^n \end{pmatrix} = \begin{pmatrix} \frac{e^{-ikL}\left(Z_0\left(-\left(e^{2ikL}\left(t_p\left(1+e^{2idk_c}\right)-2t_s e^{idk_c}\right)-2r_s e^{idk_c}\right)\right)-r_p\left(Z_0\left(1+e^{2idk_c}\right)-2Z_c\left(-1+e^{2idk_c}\right)\right)\right)}{2Z_c\left(-1+e^{2idk_c}\right)} \\ \frac{e^{-ikL}\left(e^{2ikL}\left(t_p\left(Z_0\left(1+e^{2idk_c}\right)+2Z_c\left(-1+e^{2idk_c}\right)\right)-2Z_0 t_s e^{idk_c}\right)+Z_0 r_p\left(1+e^{2idk_c}\right)-2Z_0 r_s e^{idk_c}\right)}{2Z_c\left(-1+e^{2idk_c}\right)} \\ \frac{e^{-ikL}\left(-Z_0 e^{2ikL}\left(t_s\left(1+e^{2idk_c}\right)-2t_p e^{idk_c}\right)+2Z_0 r_p e^{idk_c}-r_s\left(Z_0\left(1+e^{2idk_c}\right)-2Z_c\left(-1+e^{2idk_c}\right)\right)\right)}{2Z_c\left(-1+e^{2idk_c}\right)} \\ \frac{e^{-ikL}\left(e^{2ikL}\left(t_s\left(Z_0\left(1+e^{2idk_c}\right)+2Z_c\left(-1+e^{2idk_c}\right)\right)-2Z_0 t_p e^{idk_c}\right)-2Z_0 r_p e^{idk_c}+Z_0 r_s\left(1+e^{2idk_c}\right)\right)}{2Z_c\left(-1+e^{2idk_c}\right)} \\ \frac{e^{i(dk_c-kL)}\left(e^{2ikL}\left(-t_s+t_p e^{idk_c}\right)+r_p e^{idk_c}-r_s\right)}{-1+e^{2idk_c}} \\ \frac{e^{-ikL}\left(e^{2ikL}\left(-t_p+t_s e^{idk_c}\right)+r_s e^{idk_c}-r_p\right)}{-1+e^{2idk_c}} \end{pmatrix}$$

*in this solution r_p means r_p^{n-1} etc. otherwise Mathematica will think it's r_p to the $(n-1)$th power and it will be a mess...

in second thought - working iteratively is not that easy - for example, in order to calculate any unit cell 2 variable, we need r_p^1 and r_s^1 which we don't know).

4 Simulation results and discussion

After solving for all the r's nad the t's , we reconstruct the voltage and current along the main lines from equations (3)-(8). an example of such a reconstruction is shown in figure 3, with parameters that we'll call default configurations throughout this section: the frequency is $f = 6$GHz, the coupling segment width is $w_c = 200$nm, the coupling length $d = 20\mu m$, the unit cell length is $L = 100\mu m$ and the number of unit-cells $N = 51$.

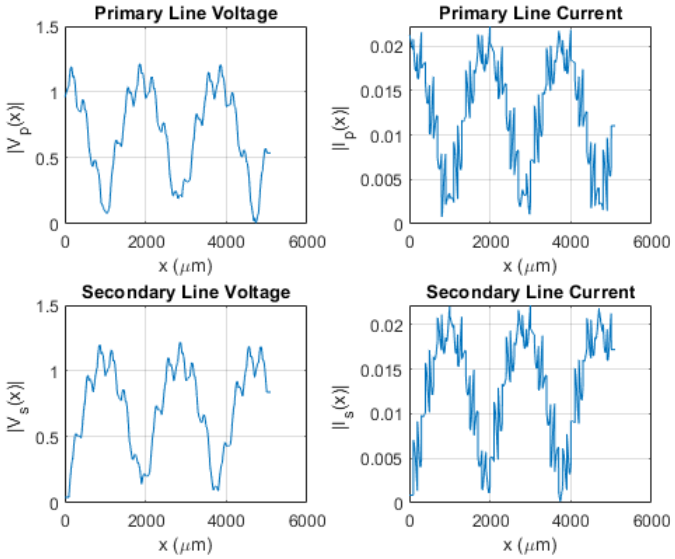

Figure 3: Simulated voltage and current magnitudes along the primary and secondary lines.
at default configurations: the frequency is $f = 6$GHz, the coupling segment width is $w_c = 200$nm, the coupling length $d = 20\mu m$, the unit cell length is $L = 100\mu m$ and the number of unit-cells $N = 51$.

We also calculated the power (see figure 4) along the lines, using :

$$P = \frac{1}{2}\mathrm{Re}\left[V \cdot I^*\right] \tag{38}$$

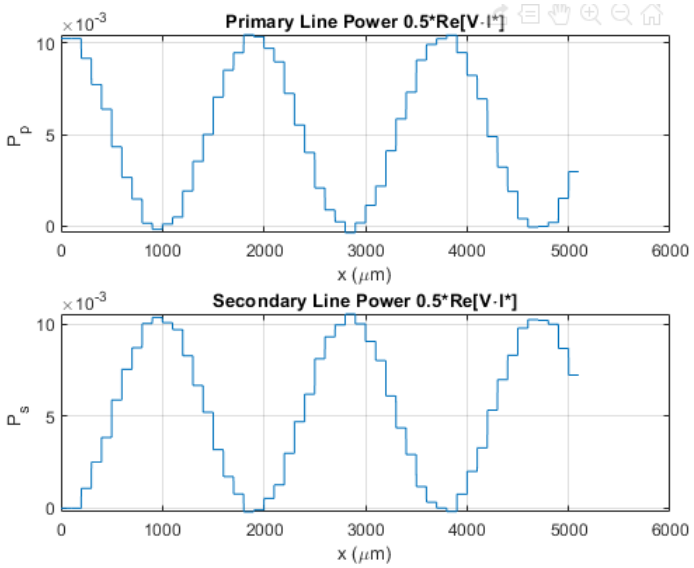

Figure 4: Simulated power along the primary and secondary line. default configuration:the frequency is $f = 6$GHz, the coupling segment width is $w_c = 200$nm, the couling length $d = 20\mu m$, the unit cell length is $L = 100\mu m$ and the number of unit-cells $N = 51$. one can see the energy oscillates between the two lines.

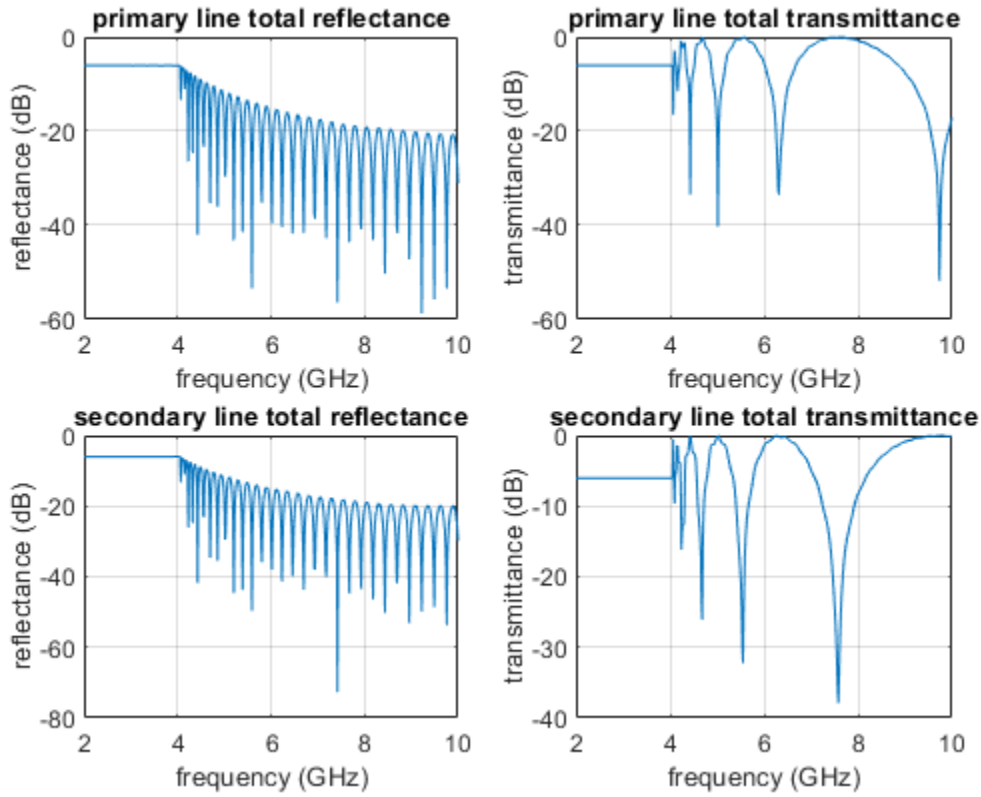

Figure 5: Simulated network frequency response for default configurations: the coupling segment width is $w_c = 200$nm, the coupling length $d = 20\mu m$, the unit cell length is $L = 100\mu m$ and the number of unit-cells $N = 51$.

4.1 problems for small d / low frequencies

We found that when either d or L are too small relative to the relevant wavelengths (for example around $d = 10\mu$m for 6GHz, or below 4GHz for the default configurations as can be seen in figure 5), there is no oscillation of the energy between the primary and the secondary lines, and instead the energy seems to divide equally between the 4 ports :

$$|r_p(1)|^2 = |r_s(1)|^2 = |t_p(N)|^2 = |t_s(N)|^2 \approx 0.25$$

This can be explained by considering the extreme limit $d = 0$. In this case each coupling is actually a short-circuit and other than the first intersection there are no reflections at all (it behaves like a single transmission line).

5 Generalization for a lattice of transmission lines

We will try to keep as much of the notation as possible the same. instead of two main lines p and s , we now have M lines, denoted with the index j instead of the letters p and s. The notation is presented in figure 6.

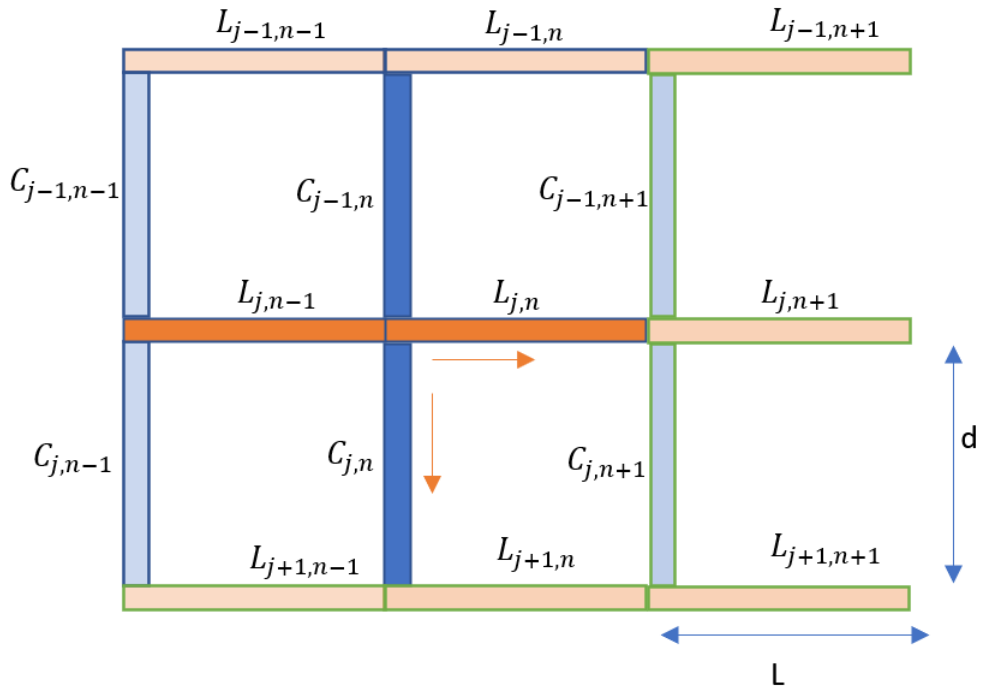

Figure 6: a lattice of transmission lines with coupling segments.
the line segments are denoted $L_{j,n}$ and the coupling segments $C_{j,n}$

using the same ansatz as before - equations (1), (2) - we write the equations for the highlighted node in figure 6:

$$\begin{aligned} V_{j,n}(0) &= V^c_{j,n}(0) \\ V_{j,n}(0) &= V_{j,n-1}(L) \\ V_{j,n}(0) &= V^c_{j-1,n}(d) \\ 0 &= I_{j,n}(0) + I^c_{j,n}(0) - I_{j,n-1}(L) - I^c_{j-1,n}(d) \end{aligned} \tag{39}$$

substituting the ansatz in:

$$\begin{aligned} 0 &= t^c_{j,n} + r^c_{j,n} - t_{j,n} - r_{j,n} \\ 0 &= t_{j,n-1}e^{ikL} + r_{j,n-1}e^{-ikL} - t_{j,n} - r_{j,n} \\ 0 &= t^c_{j-1,n}e^{ik_c d} + r^c_{j-1,n}e^{-ik_c d} - t_{j,n} - r_{j,n} \\ 0 &= \frac{1}{Z_0}(t_{j,n} - r_{j,n}) + \frac{1}{Z_c}\left(t^c_{j,n} - r^c_{j,n}\right) - \frac{1}{Z_0}\left(t_{j,n-1}e^{ikL} - r_{j,n-1}e^{-ikL}\right) - \frac{1}{Z_c}\left(t^c_{j-1,n}e^{ik_c d} - r^c_{j-1,n}e^{-ik_c d}\right) \end{aligned} \tag{40}$$

for $2 \le n \le N$ and $2 \le j \le M-1$.

5.1 dealing with boundary conditions:

the $n = 1$ boundary is dealt with in a similar way to the simpler case: we assume that the first unit cell has no coupling and thus $t^c_{j,1}, r^c_{j,1}$ do not exist for all j. Similarly $r^c_{M,n}, t^c_{M,n}$ do not exist for all n. $t_{j,1}$ are known for all j (that is - we know into which line(s) we send a signal), and $r_{j,N} = 0$ for all j.
for the $j = 1$ boundary we need a different set of equations - specifically the third equation in (39) is irrelevant, and the last equation has less terms:

$$\begin{aligned}
0 &= t^c_{1,n} + r^c_{1,n} - t_{1,n} - r_{1,n} \\
0 &= t_{1,n-1}e^{ikL} + r_{1,n-1}e^{-ikL} - t_{1,n} - r_{1,n} \\
0 &= \frac{1}{Z_0}\left(t_{1,n} - r_{1,n}\right) + \frac{1}{Z_c}\left(t^c_{1,n} - r^c_{1,n}\right) - \frac{1}{Z_0}\left(t_{1,n-1}e^{ikL} - r_{1,n-1}e^{-ikL}\right)
\end{aligned} \tag{41}$$

for $2 \le n \le N$. Similarly for $j = M$, the first equation is irrelevant and the last one has to be modified :

$$\begin{aligned}
0 &= t_{M,n-1}e^{ikL} + r_{M,n-1}e^{-ikL} - t_{M,n} - r_{M,n} \\
0 &= t^c_{M-1,n}e^{ik_c d} + r^c_{M-1,n}e^{-ik_c d} - t_{M,n} - r_{M,n} \\
0 &= \frac{1}{Z_0}\left(t_{M,n} - r_{M,n}\right) - \frac{1}{Z_0}\left(t_{M,n-1}e^{ikL} - r_{M,n-1}e^{-ikL}\right) - \frac{1}{Z_c}\left(t^c_{M-1,n}e^{ik_c d} - r^c_{M-1,n}e^{-ik_c d}\right)
\end{aligned} \tag{42}$$

for $2 \le n \le N$

5.2 counting variables and equations

we have 4 variables for each n and j, that is $4MN$ of which and $2M + 2N - 2$ do not exist. this gives a total of $4MN - 2M - 2N + 2$. we have $4\left(M-2\right)\left(N-1\right)$ equation from (40) , another $6\left(N-1\right)$ equation from (41) and (42), and yet another $2M$ from the boundary conditions (the known variables are treated here as trivial equations). this gives a total of $4\left(M-2\right)\left(N-1\right)+6\left(N-1\right)+2M = 4MN-8N-4M+8+6N-6+2M = 4MN-2M-2N+2$ equations - the same as the number of variables.

5.3 Encoding the equations

first, look at the bulk equations for a given n, j: Think of the variables as stored in a matrix:

$$\begin{pmatrix} M_{j-1,n-1} & M_{j-1,n} \\ M_{j,n-1} & M_{j,n} \end{pmatrix}$$

where

$$M_{j,n} := \begin{pmatrix} t_{j,n} & r_{j,n} \\ t^c_{j,n} & r^c_{j,n} \end{pmatrix}$$

or, explicitly:

$$\begin{pmatrix} t_{j-1,n-1} & r_{j-1,n-1} & t_{j-1,n} & r_{j-1,n} \\ t^c_{j-1,n-1} & r^c_{j-1,n-1} & t^c_{j-1,n} & r^c_{j-1,n} \\ t_{j,n-1} & r_{j,n-1} & t_{j,n} & r_{j,n} \\ t^c_{j,n-1} & r^c_{j,n-1} & t^c_{j,n} & r^c_{j,n} \end{pmatrix} \tag{43}$$

the same way that a matrix can be used to encode a large number of equations for a vector of variables, we can use a 3D tensor to encode a large number of equations for a matrix of variables. In the previous case, each column

corresponds to a variable, and each line to an equation. now, each matrix-slice will correspond to an equation, encoding the coefficients in it's rows and columns. s.t the system of equations will be encoded by

$$T * X = V$$

where T is a 3D tensor of coefficients, X is a 2D matrix of variables, and V is a 1D vector . for this to work we need to define the product $*$ of a tensor and a matrix like this

$$(T * X)_i = \sum_{k,m} T_{i,k,m} X_{k,m}$$

so, we see that the first index i of T is the index of the equations, and for a specific i T_{km} are the coefficients of the equation.

for example, using this encoding, the matrix for the first equation in (40) will be encoded by:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

and the second eqn by:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ e^{ikL} & e^{-ikL} & -1 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

the third:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & e^{ik_c d} & e^{-ik_c d} \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and the fourth:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -Y_c e^{ik_c d} & Y_c e^{-ik_c d} \\ -Y_0 e^{ikL} & Y_0 e^{-ikL} & Y_0 & -Y_0 \\ 0 & 0 & Y_c & -Y_c \end{pmatrix}$$

This way we can construct the "unit-tensor" - that is the tensor that encodes the 4 equations of a single unit cell. We can than construct the big 3D tensor for all the bulk equations by pasting shifted copies of the "unit-tensor". We can do a similar thing with the boundary equations. Note that since $t^c_{j,1}, r^c_{j,1}$ do not exist for all j, and $r^c_{M,n}, t^c_{M,n}$ do not exist for all n, in the big matrix of variables (built from the unit-cell matrix of variables (43)) there are redundant elements. In the case of $r^c_{M,n}, t^c_{M,n}$ this is the last row of the big variables matrix, so we can just remove it. It is not so simple for $t^c_{j,1}, r^c_{j,1}$ because in our encoding these do not correspond to a single row or column. On the other hand, these variables are not participating in any equation, so we can add trivial equations that fix them to some value (say, zero) and not worry about them any more.

The actual solving of the equations is done by "unfolding" the tensor into a big matrix using matlab's reshape().

5.4 Simulation Results

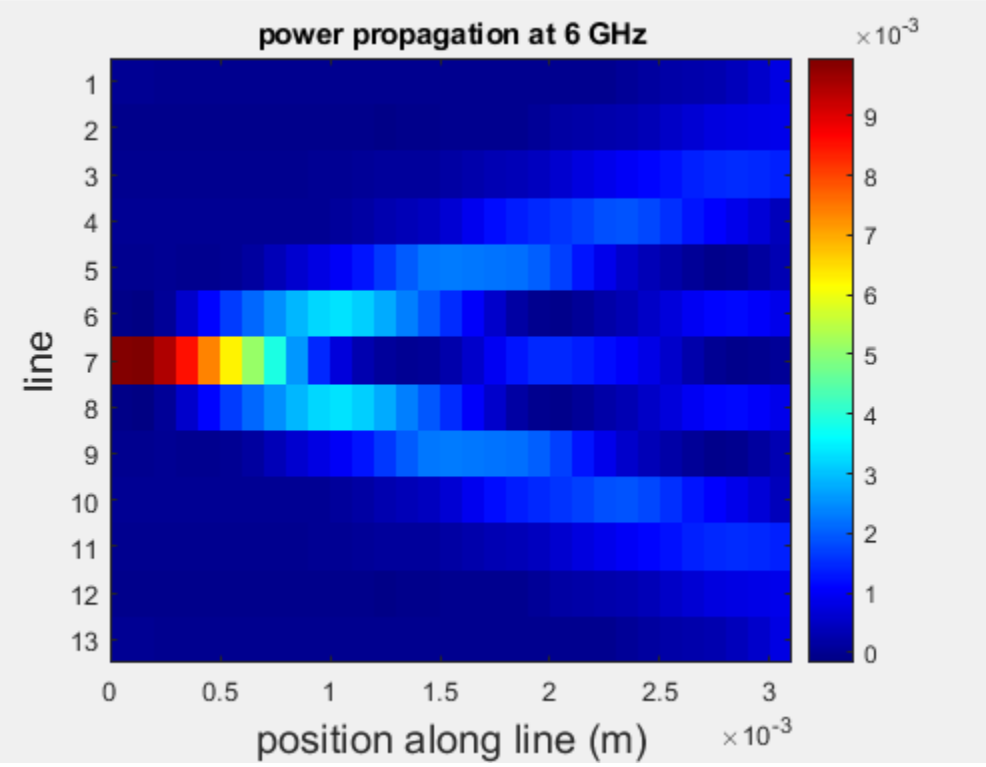

Figure 7: power propagation through 13 lines at 6GHz. configurations: $w = 2.3\mu$m , $t = 8$nm (thickness of WSi), $H = 16$nm (thickness of dielectric), $w_c = 300$nm$L = 100\mu$m,$d = 20\mu$m, $N = 31$, $M = 13$, input from line 7.

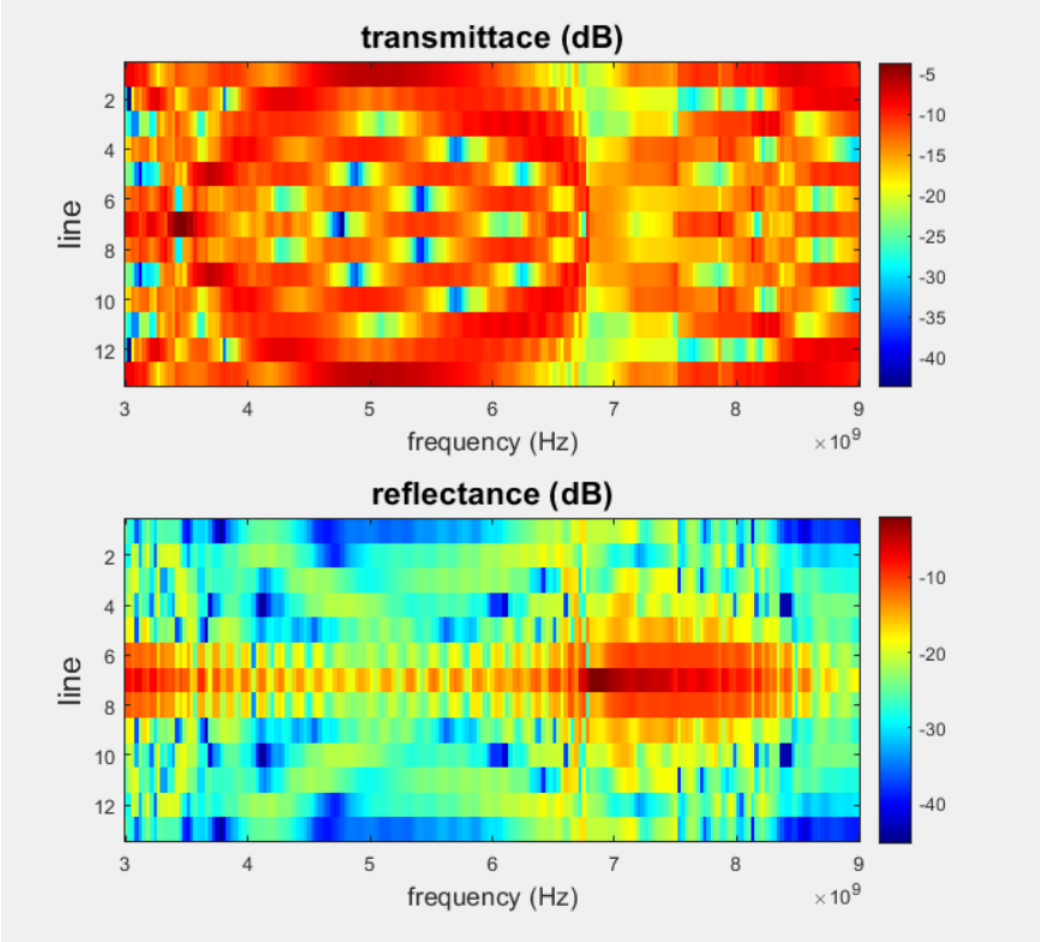

Figure 8: Frequancy response for 13 lines. configurations: $w = 2.3\mu$m , $t = 8$nm (thickness of WSi), $H = 16$nm (thickness of dielectric), $w_c = 300$nm$L = 100\mu$m,$d = 20\mu$m, $N = 31$, $M = 13$, input from line 7.

6 Introducing reflections

up until now we assumed that the traces go on to infinity on either side. This is a valid assumption if everything is impedance-matched, but it is technically challanging to make sure numerous chip outputs (around $2M$) are terminated by a 50Ω cap. Therefore, we would like to introduce reflections to the simulation. As a first step, let's assume that the lines are ended with an open circuit after the last or first segment, and thus we have full reflections. Previously we had $t_{j,1} = 0$ for all j (other than the special ones through which we send the signal), and $r_{j,N} = 0$ for all j, now we have instead : $t_{j,1} = r_{j,1}$ for all j other than the special ones, and $r_{j,N} = t_{j,N}$. Solving for such a scenario shows that we don't get the behavior we wanted, so an alternative solution is to introduce attenuators in

the way, such that we have:

$$\begin{aligned} t_{j,1} &= a^2 e^{2ik_{cx}L_{cx}} r_{j,1} \quad \text{(for all } j \text{ other than the special ones)} \\ r_{j,N} &= a^2 e^{2ik_{cx}L_{cx}} t_{j,1} \quad \text{(for all } j \text{)} \end{aligned}$$

where a is a voltage attenuation factor, $k_{cx} \approx \frac{3}{2}\frac{\omega}{c}$ is the wavenumber for the readout coaxial lines, and L_{cx} is the readout line total length ($\sim$ 30cm).

7 Introducing loss

Based on Pozar chapter 2.1, we can introduce loss by considering the transmission line finite series resistance R per unit length, and its finite shunt conductance G per unit length. Solving the telegrapher's equations we get a general solution of the form (eqn 2.6 and 2.8 from Pozar)

$$\begin{aligned} V(z) &= V_0^+ e^{-\gamma z} + V_0^- e^{\gamma z} \\ I(z) &= \frac{1}{Z_0}\left(V_0^+ e^{-\gamma z} - V_0^- e^{\gamma z}\right) \end{aligned} \tag{44}$$

where

$$\begin{aligned} \gamma &:= \sqrt{(R + j\omega L)(G + j\omega C)} \\ Z_0 &:= \sqrt{\frac{R + j\omega L}{G + j\omega C}} \end{aligned}$$

two steps are required in order to translate Pozar's expressions to our notation:

1. Pozar's j becomes $-i$. That's because his convention assumes everything goes with $e^{+j\omega t}$, and we use $e^{-i\omega t}$.
2. we want to speak in the terms of a complex $\tilde{k}$ and $\tilde{Z}_0$ s.t the general solution is

$$\begin{aligned} V(x) &= V_0^+ e^{+i\tilde{k}x} + V_0^- e^{-i\tilde{k}x} \\ I(z) &= \frac{1}{\tilde{Z}_0}\left(V_0^+ e^{+i\tilde{k}x} - V_0^- e^{-i\tilde{k}x}\right) \end{aligned} \tag{45}$$

   comparing (45) to (44) this means that $k = -\frac{\gamma}{i} = +i\gamma$. Note that the sign has flipped since in our notation e^{+ikx} is a traveling wave in the positive direction, and Pozar uses the opposite convention.

Thus our ansatz becomes of the form of eqns (1) (2) , with

$$\begin{aligned} \tilde{k} &= i\sqrt{(R - i\omega L)(G - i\omega C)} \\ \tilde{Z}_0 &= \sqrt{\frac{R - i\omega L}{G - i\omega C}} \end{aligned} \tag{46}$$

It will be more convenient to express these in terms of the lossless phase-velocity v_{ph} and lossless characteristic impedance Z_0. note that

$$C = \frac{1}{Z_0 v_{ph}} \quad L = \frac{Z_0}{v_{ph}}$$

plugging into (46):

$$\tilde{k} = i\sqrt{\left(R - i\omega\frac{Z_0}{v_{ph}}\right)\left(G - i\omega\frac{1}{Z_0 v_{ph}}\right)} \tag{47}$$

$$\tilde{Z}_0 = \sqrt{\frac{R - i\omega\frac{Z_0}{v_{ph}}}{G - i\omega\frac{1}{Z_0 v_{ph}}}}$$

It is important to take the square roots in such a way that $\mathrm{Re}\left(\tilde{k}\right) > 0$, $\mathrm{Im}\left(\tilde{k}\right) > 0$ and $\mathrm{Re}\left(\tilde{Z}_0\right) > 0$. Matlab takes the square-root by halving the angle in the complex plane, when the angle is defined on $(-\pi, \pi]$. For $\tilde{Z}_0$ the expression inside the square root has a positive real part, so $\tilde{Z}_0$ also has a positive real part. for $\tilde{k}$ there is a subtle point - if the $\mathrm{Im}\left\{(R - i\omega L)(G - i\omega C)\right\} < 0$, we get $\mathrm{Re}\left(\tilde{k}\right) > 0$ and $\mathrm{Im}\left(\tilde{k}\right) > 0$, but if $\mathrm{Im}\left\{(R - i\omega L)(G - i\omega C)\right\} = 0$, we get a real and negative $\tilde{k}$, which is not what we want. One way to get around that is to use only nonzero values for R or G. another way (which we used) is to multiply $\tilde{k}$ by the sign of it's real part.

8 Non-linearity

The kinetic inductance has the following non-linear current dependence:

$$L_{\mathrm{kin}} = L_0\left(1 + \left(\frac{I}{I_*}\right)^2\right)$$

where L_0 is the zero current kinetic inductance and , and I_* is comparable to the critical current. We can try to take that into account by working in iterations: solving the linear problem, get the current distribution along the lattice, calculate the correction to the kinetic inductance and then solve again with corrected effective wavenumber $\tilde{k}$ and characteristic impedance $\tilde{Z}_0$. For each line segment (coupler or not) we need two values for L_{kin} : one for $x = 0$ and one for $x = L$ (or $x = d$).
to be continued..

9 Measurement results

Below are some of the results from the measurement of NOCKIT5:

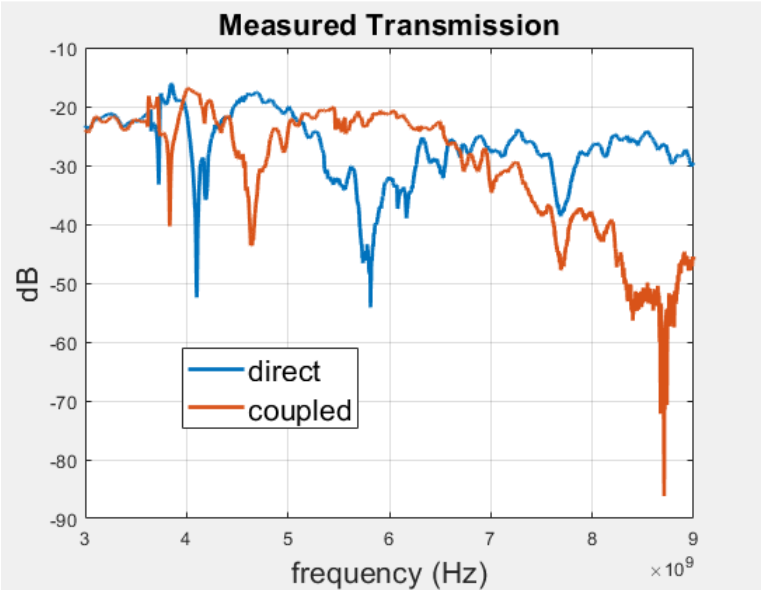

Figure 9: measured transmission of the two lines ("ladder") device. The transmission through the primary line (through which a signal is sent) is plotted in blue, the transmission through the coupled line is plotted in red.

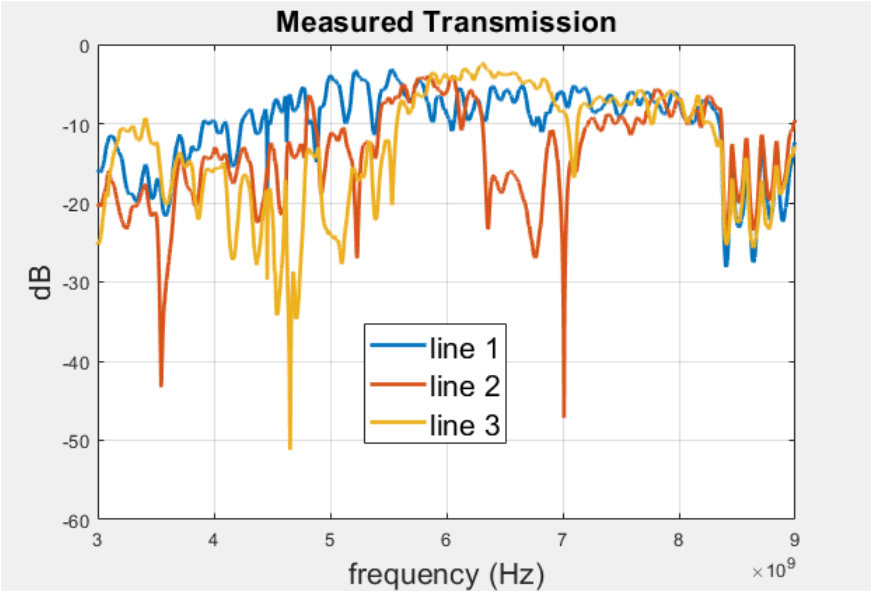

Figure 10: measured transmission of the three lines device. the input signal is through line 1.

9.1 Fitting

There are essentially three parameters in the simulation that we can tune independently:

1. the phase velocity of the lines v_{ph}
2. the phase velocity of the couplers v_{ph_c}
3. the ratio of characteristic impedances Z_c/Z_0.

(we can of course tune the impedance independently, but it is only the ratio that effects the result)

We performed the fitting by having an initial educated guess for all the characteristic-impedances and phase-velocities based on the geometry and dimensions of the the fabricated device. Than we multiply v_{ph}, $v_{ph_c}$and Z_c by three different factors (x_1,x_2,x_3) and minimize the deviation of simulation from measurement w.r.t $\vec{x}$.

For the 2-lines case we got some promising results for

$$\vec{x} = (2.0000, 1.0141, 0.9227) \tag{48}$$

as can be seen in figure 11.

It is a encouraging that x_2 and x_3 are close to 1, but having x_1 at 2.000 is suspicious. It seems like we have a factor of 2 mistake somewhere, but we haven't found where yet. It is particularly weird that the error is in v_{ph} but not in v_{ph_c} , as the two are calculated basically the same way.

For the 3 lines device, the results are less definitive: We suspect the middle line got disconnected somehow. The simulation does seem to bear some qualitative resemblance to the measurement, particularly if we allow reflections at the input: see figure 12.

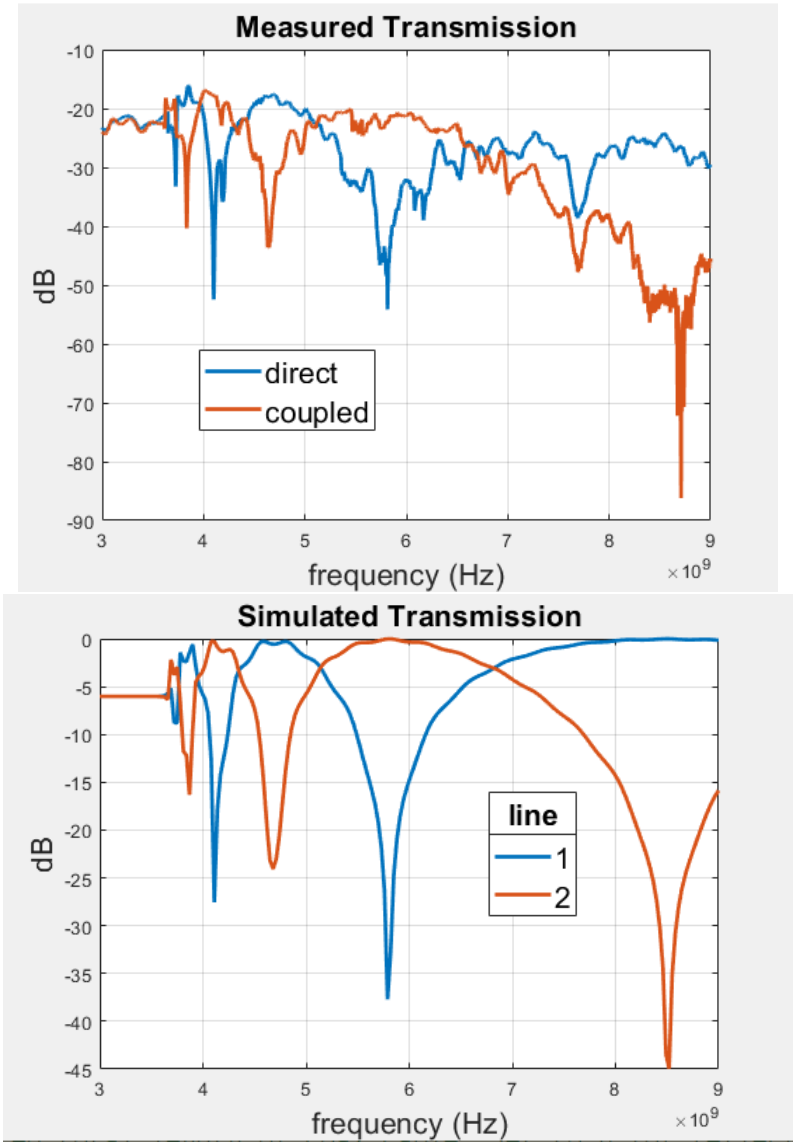

Figure 11: Measurement (top) and simulation with parameters from fit procedure (bottom) of a two lines device.

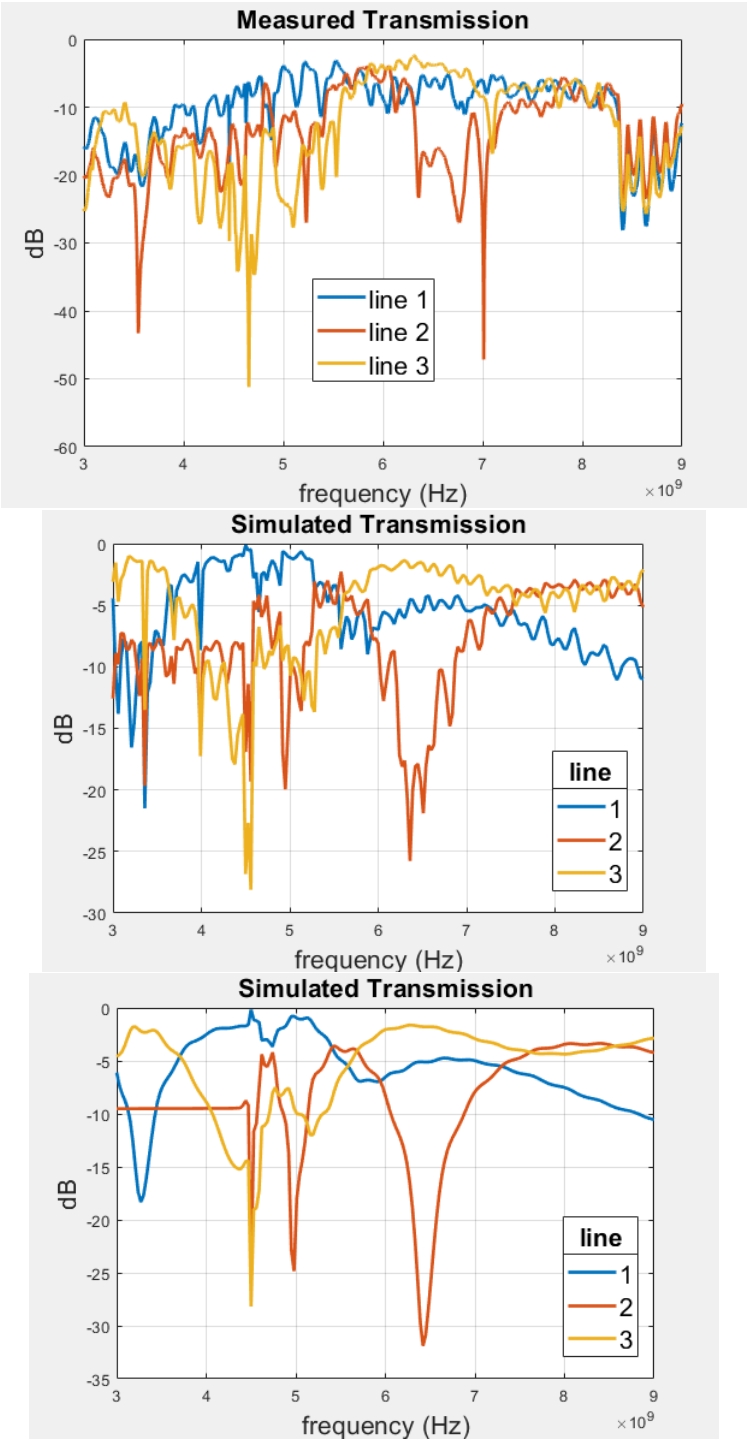

Figure 12: (top) measured transmission of the three lines device. the input signal is through line 1, (center) simulated transmission of three lines with fit parameters $\vec{x}$ from 2 lines measurement, with full reflections at the input, (bottom) simulated transmission of three lines with fit parameters $\vec{x}$ from 2 lines measurement, no reflections.