Modeling non linear components in the frequency domain
Olivier Maurice

To cite this version:
Olivier Maurice. Modeling non linear components in the frequency domain. 2017. <hal-01648381>
MODELING NON LINEAR COMPONENTS IN THE FREQUENCY DOMAIN

OLIVIER MAURICE

CONTENTS

1 Introduction 2
2 Basic principles with a diode 2
  2.1 Spectrum transformation 2
  2.2 Impedance operator and current source 3
  2.3 Impedance operators on domains 3
  2.4 Dynamic connectivity 3
3 Programing the diode case 4
  3.1 Generator 5
  3.2 Functions of domains 5
  3.3 Loop including the generator and the diode 5
4 Conclusion 5

LIST OF FIGURES

Figure 1 Case of a simple diode 11
Figure 2 Added current source 11
Figure 3 Generator modeling 11
Figure 4 Function of domains 12
Figure 5 Answer for low level 12
Figure 6 Answer for high level 13

ABSTRACT

The frequency domain is the major domain used in electromagnetic compatibility (EMC). But it has an important default: it classically cannot be used when non linear components are involved. The purpose of this paper is to submit a technique, under the formalism of the tensorial analysis of networks (TAN), that may allow to model non linear components remaining in the frequency domain. The approach is based on current sources that creates the added harmonics coming from the non linear behavior of the components. An example is shown with a diode. The principle can be generalized to any circuits. We conclude giving the tracks of this generalization.

1 Department EMC and Laser, ARIANE Group, route de Verneuil, 78 Les Mureaux, France
1 INTRODUCTION

Electromagnetic compatibility works on the electromagnetic interactions that occur between electronics. Fields propagate in free space or using waveguides like lines or cavities. But all these interactions end on active devices including non linear components like diodes, transistors, etc. The response of these components depends on the level applied across them. While skin effect or radiations are easily modeled in the frequency domain, this kind of behavior is more difficult to model in the frequency domain. Previous works have shown how to define impedance on more than one law, depending on the values of some parameters like voltage, current, temperature, etc. By a similar technique we can define an impedance that changes with the amplitude of a voltage for example. This impedance can drive a current source which takes in charge the creation of harmonics coming from the non linear behavior of the component. Previous works were done on the same subject, using filters to reproduce the creation of harmonics from pure sinusoidal waveforms coming from non linear components. We try here to submit a technique as easier as possible under the TAN formalism. The way to reach our objective is different from the previous work, perhaps less general, but we hope simpler to use.

2 BASIC PRINCIPLES WITH A DIODE

Basic idea consists in creating a generator, seted with a given parameter. This generator will take in charge the creation of the harmonics coming from the non linear behavior. If we make a simple test, for example exciting a diode with a square, symmetric signal, when the amplitude goes over the diode threshold, i.e. around 0.6 V, the signal should rapidly becomes non symmetric and only following the diode polarization. Figure 1 illustrates this mechanism.

2.1 Spectrum transformation

When the voltage across the diode is under the threshold, it behaves like a resistance. We know the spectrum $\hat{S}$ of a rectangular, symmetric signal, it is given under Laplace’s formalism ($p$ being Laplace’s operator) by:

$$\hat{S}(p) = A \left( \frac{1}{p} - \frac{2}{p} e^{-\tau p} + \frac{1}{p} e^{-2\tau p} \right)$$

(1)

$\tau$ being the half period duration and $A$ the signal peak amplitude in the frequency domain. If the diode works as a perfect diode, the spectrum $\hat{S}'$ should become the one of a single rectangular signal:

$$\hat{S}'(p) = A \left( \frac{1}{p} - \frac{1}{p} e^{-\tau p} \right)$$

(2)

So some operations should transform the spectrum of current across the diode from the function $\hat{S}$ to the function $\hat{S}'$. How to do that?
2.2 Impedance operator and current source

When the level of voltage (the voltage being here our control parameter) across the diode is too low, the diode must be a simple resistance of high value. If the voltage is negative, this resistance must remain very high (we don’t consider the avalanche process for the moment). Now if the voltage goes over the threshold, the resistance must become low (that’s the dynamic resistance of the diode) and the spectrum must become the one of a single positive rectangular waveform. It means that the negative part must be suppressed by some added source. If the dynamic resistance of the diode is $R_{\text{dyn}}$, the nominal and linear current in the diode may be (the generator $E$ having the expression of $\hat{S}$):

$$i(p) = \frac{\hat{S}(p)}{R_0 + R_{\text{dyn}}}$$

(3)

$R_0$ being the generator self resistance. To suppress the negative part of this spectrum in current, we must add an external current source that creates an opposite spectrum for the negative part. If $J$ is this current source, it is defined by:

$$J(p) = \frac{A}{R_0 + R_{\text{dyn}}} \left( \frac{1}{p} e^{-\tau p} - \frac{1}{p} e^{-2\tau p} \right)$$

(4)

We can verify that:

$$i(p) + J(p) = \frac{\hat{S}'(p)}{R_0 + R_{\text{dyn}}}$$

(5)

But how can we change the impedance dynamically and add the current source?

2.3 Impedance operators on domains

Impedances can be defined, under the tensorial analysis of networks, on domains [MAURICE(2017)]. If $R_i$ is the resistance of the diode when it is polarized in inverse (voltage across the diode is negative following the description of $i$), $R_d$ when it is polarized in direct but with a level of voltage $v$ under the threshold value and $R_{\text{dyn}}$ when the diode becomes active, its global impedance whatever the voltage value can be described by the function:

$$Z_d = \mathcal{D}_1 R_1 + \mathcal{D}_2 R_2 + \mathcal{D}_3 R_{\text{dyn}}^3$$

(6)

With this function, the diode behavior in impedance is now completely defined once we know the voltage developed across it. The domain functions $\mathcal{D}_x$ are equal to 1 when the parameter (here $v$) belongs to the interval $x$. Their profiles can be more or less abrupt (logistic functions as arctangent ones or gaussian ones can be used) and it exists some intermediate states, when the parameter value belongs to both intervals where the diode can have both impedances and behavior. The speed of change depends on the time rise of the domain functions.

2.4 Dynamic connectivity

It remains to add the current source which depends also of the parameter $v$ value. Two techniques can be imagined. The first one uses domains to make
this current source available when the threshold is covered. The second one acts on the connectivity between the branch currents and the meshes one.

Figure 2 shows the circuit completed with the current source as a added mesh, having its own voltage value unknown \( U \) while the current value \( J \) is known.

We can construct the connectivity \( C \) between the branch currents and the mesh ones \( (K^1, K^2) \):

\[
C = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}
\]  

(7)

If \( i^q \) are the branch currents (one flows into \( R_0 \) and the other into \( Z_d \)), we have \( i^q = C^q m^m \).

The impedance operator in the branch space being defined by:

\[
Z = \begin{bmatrix} R_0 & 0 \\ 0 & Z_d \end{bmatrix}
\]  

(8)

Changing from branch to the nodes-pair space, we apply \( \zeta = C^T ZC \) in order to obtain:

\[
\zeta = \begin{bmatrix} Z_d + R_0 & Z_d \\ Z_d & Z_d \end{bmatrix}
\]  

(9)

This impedance operator leads us to the system of equations:

\[
\begin{cases}
E = (Z_d + R_0) K^1 + Z_d K^2 \\
U = Z_d K^1 + Z_d K^2
\end{cases}
\]  

(10)

The first equation should be sufficient to solve the unknown current \( K^1 \) with the known source \( E \) and current \( K^2 \):

\[
E - Z_d K^2 = (Z_d + R_0) K^1
\]  

(11)

But \( Z_d \) is never equal to zero. it means that \( K^2 \) intervenes even when the level of \( v \) is lower than the threshold. We understand that another function acting on \( K^2 \) must be added. The simplest way to modify the influence of \( K^2 \) is to make it existing only when the threshold is exceeded. It means that the connectivity can be defined using a domain like:

\[
C = \begin{bmatrix} 1 & 0 \\ 1 & v \end{bmatrix}
\]  

(12)

With this connectivity, which depends on \( v \), the relation between \( i^2 \) and \( K^2 \) exists only when the threshold is exceeded. For all the other values of \( v \), the residual equation is:

\[
E = (Z_d + R_0) K^1
\]  

(13)

More, the fact to use a dynamic connectivity is an accurate description of the physical process involved, because the non linear current source exists effectively only when the threshold is exceeded. For all the other values of \( v \), this source doesn’t exist.

3 **Programing the Diode Case**

Next step is to see if our model can be computed using classical scientific programing under Python for example. First step is to choose a function for the domains. Various choices are available, more or less easy to use. Another challenge is to estimate the amplitude of the parameter \( v \).
3.1 Generator

We can start by making the generator. Any numeric application being a compromise between performances and accuracy, a first work consists in programming the spectrum of a known waveform and to verify that the inverse Fourier’s transformation of this spectrum leads to the good time waveform. Starting from the spectrum definition given in \ref{eq:1}, we can compute this spectrum choosing the best frequency sampling as possible to limit the number of points and to represent with a good pertinence the spectrum. The time waveform considered is 200 ns width, symmetric with a half period of 100 ns. 1000 points and a window of study of 10 \( \mu s \) allow to obtain a good representation of the signal. Figure \ref{fig:3} shows the spectrum and its correspondence in time computed under Python language.

The listing of this first step is given in annex \ref{annex:1}.

Once the generator is modeled, we can connect it through an internal resistance to a diode, changing the amplitude of the signal that we can verify in the time domain, using the previous result. To do that, we need to realize the functions of domains.

3.2 Functions of domains

We can use logistic functions \( D \) which is defined in general by:

\[
D(a, c, v) = \frac{1}{1 + ae^{-c(v-p_0)}} - \frac{1}{1 + ae^{-c(v-(p_0+\omega_i))}}
\]

for values of \( v \) that belong to \([0,100]\) V. \( p_0 \) is the starting value of the window (for example \( p_0 = 20 \)) and \( \omega_i \) the window width (for example \( \omega_i = 30 \)). If the parameter value \( v \) is inferior to 100 V, we can choose \( a = 1 \) and \( c = 10 \). With this original values we understand that to make \( v \) around 1 V for example, we may increase \( c, p_0, \omega_i \) of a scale factor \( f_e \) with:

\[
D(a, c, v) = \frac{1}{1 + ae^{-\frac{c}{f_e}(v-p_0f_e)}} - \frac{1}{1 + ae^{-\frac{c}{f_e}(v-(p_0+\omega_i)f_e)}}
\]

The program to trace the logistic curve is given in annex 2, an example of logistic curve is shown figure \ref{fig:4} for \( f_e = 10^{-3} \).

3.3 Loop including the generator and the diode

We have at this step all the elements to compute the whole circuit with the generator in series with its own impedance and the diode. The various impedances controlled by the functions of domains and the functions of domains themselves must be defined before to compute the current in the diode. For the example we use simple function (it is another solution) that set the values of the variables depending on the parameter values given in arguments. Equation \ref{eq:11} is computed. Listing is given in annex, and the result for a low parameter value is shown figure \ref{fig:5} while the result obtained for a high value of the parameter is shown figure \ref{fig:6}.

4 CONCLUSION

In a general case, a circuit can be separated in two parts: linear parts that do not change the spectrums and non linear parts that change them. The
propagation of the new spectrum created by the non linear behavior is unchanged by the passive parts. So we need only a technique to model non linear component in general, making some exception with magnetic materials. The technique submitted in this article uses the assumption that a previous characterization of the component has been made, giving the frequency working of the component. We have shown its mechanism in the case of a simple diode, but the method can be applied on any non linear source. The added harmonics are taken in charge through a current source in the nodes-pair space. In the example given, we clearly see the change in the spectrum of the signal for the current in the diode. At low level, the diode behaves like a high resistor and the current waveform follows the electromotive force waveform. But when the level becomes high enough, a current source suppresses the negative part of the source waveform, giving the result obtained for a single positive pulse. In that case, an average value appears in the spectrum that doesn’t exist in the previous case where the waveform was symmetric, and so without any average value.

Kron’s method in the nodes-pair space is available to take into account non linear devices in the harmonic domain, coupled with the domain functions previously presented by the author[MAURICE(2017)].

REFERENCES

[MAURICE(2017)] Maurice, O. Impedances defined on domains. In Elements of theory for electromagnetic compatibility and systems; Bookelis; Publishing House: Aix en Provence, France, 2017; pp. 32-58, ISBN.

ANNEXES

Annex 1

# -*- coding: utf-8 -*-

Created on Mon Nov  6 17:37:37 2017

@author: oliviermaurice


import numpy as np
import pylab as plt

#
N=1000
E=np.zeros(N,dtype=complex)
ax=np.zeros(N,dtype=float)
at=np.zeros(N,dtype=float)
ut=np.zeros(N,dtype=float)
res=np.zeros(N,dtype=float)
T=10000E-9
fo=1./T#100E6/N
A=fo*N
dt=T/N
for f in range(1,N):
    s=f*fo
    w=2.*np.pi*s
    p=1]*w
    #
    E[f]=A*(1./p-2./p*np.exp(-100E-9*p)+1./p*
    np.exp(-200E-9*p))*np.exp(-100E-9*p)
    res[f]=abs(E[f])
    ax[f]=s
    #at[f]=f/(T)

## iLT computation
#
for ot in range(N):
    #ut[ot]=abs(spec[ot])
at[ot]=ot*dt*1E9
for iin in range(1,N):
    p=1]*2.*np.pi*iin/T
    ut[ot]=ut[ot]+np.real(E[iin]*np.exp(p*ot*dt))

plt.subplot(1,2,1)
plt.plot(ax,res)
plt.grid(True,which='both')
plt.xscale('log')
plt.xlabel('frequency [Hz]')
plt.ylabel('level [V]')

plt.subplot(1,2,2)
#timer=abs(np.fft.ifft(E))
plt.plot(at[0:100],ut[0:100])
plt.grid(True,which='both')
plt.xlabel('time [ns]')
plt.ylabel('level [V]')

plt.show()

Annex 2

# coding: utf-8

Created on Tue Nov 7 11:24:56 2017

@author: oliviermaurice

import numpy as np
import pylab as plt

#
N=1000
lo=np.zeros(N,dtype=float)
av=np.zeros(N,dtype=float)
References

#
fe=1E-2 # facteur d’echelle
a=1.
c=10./fe # raideur de pente
po=40.*fe # point initial
wi=40.*fe # duree
for iv in range(N):
    v=iv*1E-1*fe # v varie de (0 a 100)*fe
    lo[iv]=1./(1.+a*np.exp(-c*(v-po)))-1./(1.+a*np.exp(-c*(v-(po+wi))))
    av[iv]=v
plt.plot(av,lo)
plt.grid(True,which='both')
plt.xlabel('v')
plt.ylabel('logistic(v)')
plt.show()

Annex 3

# -*- coding: utf-8 -*-

Created on Mon Nov 6 17:37:37 2017

@author: oliviermaurice

import numpy as np
import pylab as plt
#
def Dri(v):
    if (v<0.6): return 1
    else: return 0
def Drdyn(v):
    if (v>=0.6): return 1
    else: return 0
#
N=1000
res=np.zeros(N,dtype=float)
T=10000E-9
fo=1./T#/100E6/N
dt=T/N
#
Ri=1E4
Rdyn=1.
Ro=10.

#----------------------------------------------
fe=1E-2 # scale factor
a=1.
c=10./fe
pori=-40.*fe
wiri=80.*fe
pordyn = 40. * fe
wirdyn = 40. * fe
#
Amp = 1E7
#
#
v = 6. * fe # - 0.1

print "v : " + str(v)
E = np.zeros(N, dtype=complex)
K1 = np.zeros(N, dtype=complex)
K2 = np.zeros(N, dtype=complex)
ax = np.zeros(N, dtype=float)
# v = np.zeros(N, dtype=float)
at = np.zeros(N, dtype=float)
ut = np.zeros(N, dtype=float)
vt = np.zeros(N, dtype=float)
et = np.zeros(N, dtype=float)
for f in range(1, N):
s = f * fo
w = 2. * np.pi * s
p = 1. / w
#
E[f] = Amp * (1. / p - 2. / p * np.exp(-100E-9*p) + 1. / p * np.exp(-200E-9*p)) * np.exp(-100E-9*p)
K2[f] = Amp / (Ro + Rdyn) * (1. / p * np.exp(-100E-9*p) - 1. / p * np.exp(-200E-9*p)) * np.exp(-100E-9*p)
Zd = Dri(v) * Ri + Drdyn(v) * Rdyn
K1[f] = (E[f] + Drdyn(v) * K2[f] * np.exp(p * ot * dt)) / (Ro + Zd)

##
# for ot in range(N):
#    ut[ot] = abs(spec[ot])
#    at[ot] = ot * dt * 1E9
#    for iin in range(1, N):
#        p = 1. / 2. * np.pi * iin / T
#        ut[ot] = ut[ot] + np.real(K1[iin] * np.exp(p * ot * dt))
#    #
# for ot in range(N):
#    # ut[ot] = abs(spec[ot])
#    at[ot] = ot * dt * 1E9
#    for iin in range(1, N):
#        p = 1. / 2. * np.pi * iin / T
#        vt[ot] = vt[ot] + np.real(K2[iin] * np.exp(p * ot * dt))
#    #
# for ot in range(N):
#    # ut[ot] = abs(spec[ot])
#    at[ot] = ot * dt * 1E9
#    for iin in range(1, N):
#        p = 1. / 2. * np.pi * iin / T
#        et[ot] = et[ot] + np.real(E[iin] * np.exp(p * ot * dt))
plt.subplot(1,2,1)
plt.plot(abs(K1))
plt.grid(True, which='both')
plt.xscale('log')
plt.title('K1(f)')
plt.xlabel('sample')
plt.ylabel('level')

plt.subplot(1,2,2)
plt.plot(at[0:100], ut[0:100])
plt.grid(True, which='both')
plt.xlabel('time [ns]')
plt.ylabel('level [V]')
plt.title('K1(t)')
plt.show()
Figure 1: Case of a simple diode.

Figure 2: Added current source.

Figure 3: Generator modeling.
Figure 4: Function of domains.

Figure 5: Answer for low level.
Figure 6: Answer for high level.