ECE 546
Lecture -14
Macromodeling

Spring 2022

Jose E. Schutt-Aine
Electrical & Computer Engineering
University of Illinois
jesa@illinois.edu
Blackbox Macromodeling

Objective: Perform time-domain simulation of composite network to determine timing waveforms, noise response or eye diagrams.
Macromodel Implementation

Flowchart:

1. Frequency-Domain Data
   - IFFT
   - MOR
   - Discrete Convolution
   - Recursive Convolution
2. STAMP
3. Circuit Simulator
4. Output
Blackbox Synthesis

Motivations
• Only measurement data is available
• Actual circuit model is too complex

Methods
• Inverse-Transform & Convolution
  • IFFT from frequency domain data
  • Convolution in time domain
• Macromodel Approach
  • Curve fitting
  • Recursive convolution
Terminations are described by a source vector $G(\omega)$ and an impedance matrix $Z$.

Blackbox is described by its scattering parameter matrix $S$. 
Blackbox - Method 1

Scattering Parameters

\[ B(\omega) = S(\omega)A(\omega) \]  \hspace{1cm} (1)

Terminal conditions

\[ A(\omega) = \Gamma B(\omega) + TG(\omega) \]  \hspace{1cm} (2)

where

\[ \Gamma = -\left[U + ZZ_o^{-1}\right]^{-1}\left[U - ZZ_o^{-1}\right] \]

and

\[ T = \left[U + ZZ_o^{-1}\right]^{-1} \]

\(U\) is the unit matrix, \(Z\) is the termination impedance matrix and \(Z_o\) is the reference impedance matrix
Blackbox - Method 1

Combining (1) and (2) \[ A(\omega) = \left[ U - \Gamma S(\omega) \right]^{-1} TG(\omega) \]

and \[ B(\omega) = S(\omega)A(\omega) = S(\omega) \left[ U - \Gamma S(\omega) \right]^{-1} TG(\omega) \]

\[ V(\omega) = A(\omega) + B(\omega) = \left[ U + S(\omega) \right] \left[ U - \Gamma S(\omega) \right]^{-1} TG(\omega) \]

\[ I(\omega) = Z_o^{-1} \left[ A(\omega) - B(\omega) \right] = Z_o^{-1} \left[ U - S(\omega) \right] \left[ U - \Gamma S(\omega) \right]^{-1} TG(\omega) \]

\[ v(t) = IFFT \{ V(\omega) \} \]

\[ i(t) = IFFT \{ I(\omega) \} \]
Method 1 - Limitations

• **No Frequency Dependence for Terminations**
  ➢ Reactive terminations cannot be simulated

• **Only Linear Terminations**
  ➢ Transistors and active nonlinear terminations cannot be described

• **Standalone**
  ➢ This approach cannot be implemented in a simulator
Blackbox - Method 2

In frequency domain \( B = SA \)

In time domain \( b(t) = s(t) * a(t) \)

Convolution: \( s(t) * a(t) = \int_{-\infty}^{\infty} s(t - \tau)a(\tau)d\tau \)
Discrete Convolution

When time is discretized the convolution becomes

\[ s(t) \ast a(t) = \sum_{\tau=1}^{t} s(t-\tau)a(\tau)\Delta\tau \]

Isolating \(a(t)\)

\[ s(t) \ast a(t) = s(0) a(t) \Delta\tau + \sum_{\tau=1}^{t-1} s(t-\tau)a(\tau)\Delta\tau \]

Since \(a(t)\) is known for \(t < t\), we have:

\[ H(t) = \sum_{\tau=1}^{t-1} s(t-\tau)a(\tau)\Delta\tau : \text{ History} \]
Terminal Conditions

Defining \( s'(0) = s(0) \Delta \tau \), we finally obtain

\[
b(t) = s'(0)a(t) + H(t)
\]

\[
a(t) = \Gamma(t)b(t) + T(t)g(t)
\]

By combining these equations, the stamp can be derived.
The solutions for the incident and reflected wave vectors are given by:

\[ a(t) = \left( 1 - \Gamma(t)s'(0) \right)^{-1} \left[ T(t)g(t) + \Gamma(t)H(t) \right] \]

\[ b(t) = s'(0)a(t) + H(t) \]

The voltage wave vectors can be related to the voltage and current vectors at the terminals

\[ a(t) = \frac{1}{2} \left[ v(t) + Z_0i(t) \right] \]

\[ b(t) = \frac{1}{2} \left[ v(t) - Z_0i(t) \right] \]
Stamp Equation Derivation

From which we get

$$\frac{1}{2} [v(t) - Z_o i(t)] = \frac{s'(0)}{2} [v(t) + Z_o i(t)] + H(t)$$

or

$$Z_o i(t) + s'(0) Z_o i(t) + 2H(t) = [1 - s'(0)] v(t)$$

or

$$[1 + s'(0)] Z_o i(t) = [1 - s'(0)] v(t) - 2H(t)$$

which leads to

$$i(t) = Z_o^{-1} [1 + s'(0)]^{-1} [1 - s'(0)] v(t) - 2Z_o^{-1} [1 + s'(0)]^{-1} H(t)$$
Stamp Equation Derivation

$i(t)$ can be written to take the form

$$i(t) = Y_{\text{stamp}} v(t) - I_{\text{stamp}}$$

in which

$$Y_{\text{stamp}} = Z_o^{-1} \left[ 1 + s'(0) \right]^{-1} \left[ 1 - s'(0) \right]$$

and

$$I_{\text{stamp}} = 2Z_o^{-1} \left[ 1 + s'(0) \right]^{-1} H(t)$$
**Stamp Equations**

\[ i(t) = Y_{\text{stamp}} v(t) - I_{\text{stamp}} \]

\[ Y_{\text{stamp}} = Z_o^{-1} \left[ 1 + s'(0) \right]^{-1} \left[ 1 - s'(0) \right] \]

\[ I_{\text{stamp}} = 2Z_o^{-1} \left[ 1 + s'(0) \right]^{-1} H(t) \]

\[
(Y_g + Y_{\text{stamp}}) v(t) = I_g + I_{\text{stamp}}
\]
Effects of DC Data

No DC Data Point

With DC Data Point

If low-frequency data points are not available, extrapolation must be performed down to DC.
Effect of Low-Frequency Data

Port 1

Port 2

Volts

Time (ns)

Volts

Time (ns)
Effect of Low-Frequency Data

Calculating inverse Fourier Transform of: \[ V(f) = \frac{2\sin(2\pi ft)}{2\pi ft} \]

Left: IFFT of a sinc pulse sampled from 10 MHz to 10 GHz. Right: IFFT of the same sinc pulse with frequency data ranging from 0-10 GHz. In both cases 1000 points are used.
Convolution Limitations

Frequency-Domain Formulation

\[ Y(\omega) = H(\omega)X(\omega) \]

Time-Domain Formulation

\[ y(t) = h(t) * x(t) \]

Convolution

\[ y(t) = h(t) * y(t) = \int_{0}^{t} h(t - \tau)y(\tau)d\tau \]

Discrete Convolution

\[ h(t) * x(t) = \sum_{\tau=1}^{t} h(t - \tau)x(\tau)\Delta\tau \]

\[ H(t) = \sum_{\tau=1}^{t-1} h(t - \tau)x(\tau)\Delta\tau : \text{History} \]

Computing History is computationally expensive \( \Rightarrow \) Use FD rational approximation and TD recursive convolution
Frequency and Time Domains

1. For negative frequencies use conjugate relation $V(-\omega) = V^*(\omega)$

2. DC value: use lower frequency measurement

3. Rise time is determined by frequency range or bandwidth

4. Time step is determined by frequency range

5. Duration of simulation is determined by frequency step
Problems and Issues

- **Discretization:** (not a continuous spectrum)
- **Truncation:** frequency range is band limited

F: frequency range  
N: number of points  
$\Delta f = F/N$: frequency step  
$\Delta t =$ time step
# Problems and Issues

<table>
<thead>
<tr>
<th>Problems &amp; Limitations (in frequency domain)</th>
<th>Consequences (in time domain)</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discretization</strong></td>
<td>Time-domain response will repeat itself periodically (Fourier series) Aliasing effects</td>
<td>Take small frequency steps. Minimum sampling rate must be the Nyquist rate</td>
</tr>
<tr>
<td><strong>Truncation in Frequency</strong></td>
<td>Time-domain response will have finite time resolution (Gibbs effect)</td>
<td>Take maximum frequency as high as possible</td>
</tr>
<tr>
<td><strong>No negative frequency values</strong></td>
<td>Time-domain response will be complex</td>
<td>Define negative-frequency values and use $V(-f)=V^*(f)$ which forces $v(t)$ to be real</td>
</tr>
<tr>
<td><strong>No DC value</strong></td>
<td>Offset in time-domain response, ringing in base line</td>
<td>Use measurement at the lowest frequency as the DC value</td>
</tr>
</tbody>
</table>
Complex Plane

– An arbitrary network’s transfer function can be described in terms of its s-domain representation

– $s$ is a complex number $s = \sigma + j\omega$

– The impedance (or admittance) or transfer function of networks can be described in the s domain as

$$T(s) = \frac{\sum_{m} a_{m} s^{m} + \sum_{m-1} a_{m-1} s^{m-1} + \ldots + \sum_{1} a_{1} s + \sum_{0} a_{0}}{\sum_{n} b_{n} s^{n} + \sum_{n-1} b_{n-1} s^{n-1} + \ldots + \sum_{1} b_{1} s + \sum_{0} b_{0}}$$
Transfer Functions

\[ T(s) = \frac{a_m s^m + a_{m-1} s^{m-1} + \ldots + a_1 s + a_0}{s^n + b_{n-1} s^{n-1} + \ldots + b_1 s + b_0} \]

The coefficients \( a \) and \( b \) are real and the order \( m \) of the numerator is smaller than or equal to the order \( n \) of the denominator

A stable system is one that does not generate signal on its own.

For a stable network, the roots of the denominator should have negative real parts.
Transfer Functions

The transfer function can also be written in the form

\[ T(s) = a_m \frac{(s - Z_1)(s - Z_2)\ldots(s - Z_m)}{(s - P_1)(s - P_2)\ldots(s - P_m)} \]

\( Z_1, Z_2, \ldots Z_m \) are the *zeros* of the transfer function

\( P_1, P_2, \ldots P_m \) are the *poles* of the transfer function

For a stable network, the poles should lie on the left half of the complex plane
Model Order Reduction

Large Network (>1,000 nodes) \rightarrow \text{Reduced Order Model} (< 30 poles)

SPICE
\[ Y(t) v(t) = i(t) \]

\[ \tilde{Y}(\omega) = \left[ A_l + \sum_{i=1}^{L} \frac{a_{ii}}{1 + j\omega / \omega_{ci}} \right] \]

\[ Y(\omega) V(\omega) = I(\omega) \]

Order Reduction

\[ Y(\omega) \approx \tilde{Y}(\omega) \]

Recursive Convolution

\[ \tilde{Y}(t) v(t) = i(t) \]
Model Order Reduction

Objective: Approximate frequency-domain transfer function to take the form:

\[ H(\omega) = \left[ A_1 + \sum_{i=1}^{L} \frac{a_{1i}}{1 + j\omega / \omega_{ci}} \right] \]

Methods

- AWE – Pade
- Pade via Lanczos (Krylov methods)
- Rational Function
- Chebyshev-Rational function
- Vector Fitting Method
Model Order Reduction (MOR)

Question: Why use a rational function approximation?

Answer: because the frequency-domain relation

\[ Y(\omega) = H(\omega)X(\omega) = \left[ d + \sum_{k=1}^{L} \frac{c_k}{1 + j\omega / \omega_{ck}} \right] X(\omega) \]

will lead to a time-domain recursive convolution:

\[ y(t) = d x(t - T) + \sum_{k=1}^{L} y_{pk}(t) \]

where

\[ y_{pk}(t) = a_k x(t - T) \left( 1 - e^{-\omega_{ck}T} \right) + e^{-\omega_{ck}T} y_{pk}(t - T) \]

which is very fast!
Model Order Reduction

Transfer function is approximated as

\[ H(\omega) = d + \sum_{k=1}^{L} \frac{c_k}{1 + j\omega / \omega_{ck}} \]

In order to convert data into rational function form, we need a curve fitting scheme \( \Rightarrow \) Use Vector Fitting
History of Vector Fitting (VF)

- 1998 - Original VF formulated by Bjorn Gustavsen and Adam Semlyen*

- 2003 - Time-domain VF (TDVF) by S. Grivet-Talocia.


- 2006 - Relaxed VF by Bjorn Gustavsen.

- 2006 - VF re-formulated as Sanathanan-Koerner (SK) iteration by W. Hendrickx, Dirk Deschrijver and Tom Dhaene, et al.

Vector Fitting (VF)

Vector fitting algorithm

\[
\begin{bmatrix}
\sigma(s)f(s) \\
\sigma(s)
\end{bmatrix}
\approx
\begin{bmatrix}
\sum_{n=1}^{N} \frac{c_n}{s - \tilde{a}_n} + d + sh \\
\sum_{n=1}^{N} \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1
\end{bmatrix}
\]

Avoid ill-conditioned matrix

Guarantee stability

Converge, accurate

With Good Initial Poles

Can show* that the zeros of \( \sigma(s) \) are the poles of \( f(s) \) for the next iteration

Examples

1.- DISC: Transmission line with discontinuities

Length = 7 inches

2.- COUP: Coupled transmission line2

\[ d_x = 5^{1/2} \text{ inches} \]

Frequency sweep: 300 KHz – 6 GHz
DISC: Approximation Results

DISC: Approximation order 90
DISC: Simulations

Microstrip line with discontinuities
Data from 300 KHz to 6 GHz

Observation: Good agreement
COUP: Approximation Results

COUP: Approximation order 75 – Before Passivity Enforcement
COUP: Simulations

Port 1: a – Port 2: d
Data from 300 KHz to 6 GHz

Observation: Good agreement
Orders of Approximation

- **Low order**
  - Approximation function
  - Measurement Data
  - \( L_T \)

- **Medium order**
  - Approximation function
  - Measurement Data
  - \( L_T \) and \( C_T \)

- **Higher order**
  - Approximation function
  - Measurement Data
  - \( L_T \) and \( C_T \)
MOR Attributes

• **Accurate**: over wide frequency range.

• **Stable**: All poles must be in the left-hand side in s-plane or inside in the unit-circle in z-plane.

• **Causal**: Hilbert transform needs to be satisfied.

• **Passive**: H(s) is analytic

\[ h[n] = h_e[n] + h_o[n] \iff H(j\omega) = H_R(j\omega) + jH_I(j\omega) \]

\[ H^*(s) = H(s^*), \]

\[ z^T [H^T(s^*) + H(s)]z \geq 0, \quad \Re[s] > 0 \]

, for Y or Z-parameters.

\[ I - H^T(s^*)H(s) \geq 0, \quad \Re[s] > 0 \]

, for S-parameters.
MOR Problems

• **Bandwidth**
  - Low-frequency data must be added

• **Passivity**
  - Passivity enforcement

• **High Order of Approximation**
  - Orders > 800 for some serial links
  - Delay need to be extracted
Causality Violations

\[ Z(f) = R_o \sqrt{f} + jL\omega \]

Near (blue) and Far (red) end responses of lossy TL

\[ Z(f) = R_o \sqrt{f} + jR_o \sqrt{f} + jL\omega \]
Fourier Transform Pairs

\( a_{re}(t) \): real part of even time-domain function
\( a_{ie}(t) \): imaginary part of even time-domain function
\( a_{ro}(t) \): real part of odd time-domain function
\( a_{io}(t) \): imaginary part of odd time-domain function

\[
a(t) = a_{re}(t) + j a_{ie}(t) + a_{ro}(t) + j a_{io}(t)
\]

In the frequency domain accounting for all the components, we can write:

\( A_{RE}(\omega) \): real part of even function in the frequency domain
\( A_{IE}(\omega) \): imaginary part of even function in the frequency domain
\( A_{RO}(\omega) \): real part of odd function in the frequency domain
\( A_{IO}(\omega) \): imaginary part of odd function in the frequency domain

\[
A(\omega) = A_{RE}(\omega) + j A_{IE}(\omega) + A_{RO}(\omega) + j A_{IO}(\omega)
\]
Fourier Transform Pairs

We also have the Fourier-transform-pair relationships:

**Time Domain**:\[ a(t) = a_{re}(t) + ja_{ie}(t) + a_{ro}(t) + ja_{io}(t) \]

\[ \uparrow \quad \uparrow \quad \uparrow \quad \downarrow \]

**Freq Domain**:\[ A(\omega) = A_{RE}(\omega) + jA_{IE}(\omega) + A_{RO}(\omega) + jA_{IO}(\omega) \]

\[ B(\omega) = S(\omega) \left[ A_{RE}(\omega) + jA_{IE}(\omega) + A_{RO}(\omega) + jA_{IO}(\omega) \right] \]

In the time domain, this corresponds to:

\[ b(t) = s(t) * \left[ (a_{re}(t) + a_{ro}(t)) + j(a_{ie}(t) + a_{io}(t)) \right] \]
Fourier Transform Pairs

We now impose the restriction that in the time domain, the function must be real. As a result,

\[ a_{ie}(t) = a_{io}(t) = 0 \quad \text{which implies that:} \quad A_{IE}(\omega) = A_{RO}(\omega) = 0 \]

The Fourier-transform pair relationship then becomes:

- **Time Domain:** \( a(t) = a_{re}(t) + a_{ro}(t) \)
  \[ \uparrow \quad \uparrow \quad \uparrow \]
  \[ \downarrow \quad \downarrow \quad \downarrow \]

- **Freq Domain:** \( A(\omega) = A_{RE}(\omega) + jA_{IO}(\omega) \)

The frequency-domain relations reduce to:

\[ B(\omega) = S(\omega) \left[ A_{RE}(\omega) + jA_{IO}(\omega) \right] \]
Fourier Transform Pairs

In summary, the general relationship is:

Time Domain : \( b(t) = b_{re}(t) + jb_{ie}(t) + b_{ro}(t) + jb_{io}(t) \)

Freq Domain : \( B(\omega) = B_{RE}(\omega) + jB_{IE}(\omega) + B_{RO}(\omega) + jB_{IO}(\omega) \)

But for a real system:

Time Domain : \( b(t) = b_{re}(t) + jb_{ie}(t) + b_{ro}(t) + jb_{io}(t) \)

Freq Domain : \( B(\omega) = B_{RE}(\omega) + jB_{IE}(\omega) + B_{RO}(\omega) + jB_{IO}(\omega) \)
Fourier Transform Pairs

So, in summary

Time Domain: \( b(t) = b_e(t) + b_o(t) \)

Freq Domain: \( B(\omega) = B_R(\omega) + jB_I(\omega) \)

The real part of the frequency-domain transfer function is associated with the even part of the time-domain response.

The imaginary part of the frequency-domain transfer function is associated with the odd part of the time-domain response.
Causality Principle

Consider a function $h(t)$

$$h(t) = 0, \quad t < 0$$

Every function can be considered as the sum of an even function and an odd function

$$h(t) = h_e(t) + h_o(t)$$

$$h_e(t) = \frac{1}{2} [h(t) + h(-t)] \quad \text{Even function}$$

$$h_o(t) = \frac{1}{2} [h(t) - h(-t)] \quad \text{Odd function}$$

$$h_o(t) = \begin{cases} h_e(t), & t > 0 \\ -h_e(t), & t < 0 \end{cases}$$

$$h_o(t) = \text{sgn}(t) h_e(t)$$
Hilbert Transform

\[ h(t) = h_e(t) + \text{sgn}(t)h_e(t) \]

In frequency domain this becomes

\[ H(f) = H_e(f) + \frac{1}{j\pi f} \ast H_e(f) \]

\[ H(f) = H_e(f) - j\hat{H}_e(f) \]

\( \hat{H}_e(f) \) is the Hilbert transform of \( H_e(f) \)

\[ \hat{x}(t) = x(t) \ast \frac{1}{\pi t} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \]

⇒ Imaginary part of transfer function is related to the real part through the Hilbert transform
Discrete Hilbert Transform

- Imaginary part of transfer function can be recovered from the real part through the Hilbert transform.

- If frequency-domain data is discrete, use discrete Hilbert Transform (DHT)*

\[
H(f_n) = \hat{f}_k = \begin{cases} 
\frac{2}{\pi} \sum_{n \text{ odd}} \frac{f_n}{k - n}, & k \text{ even} \\
\frac{2}{\pi} \sum_{n \text{ even}} \frac{f_n}{k - n}, & k \text{ odd} 
\end{cases}
\]

HT for Via: 1 MHz – 20 GHz

**Observation:** Poor agreement (because frequency range is limited)
Example: 300 KHz – 6 GHz

Observation: Good agreement

Actual is red, HT is blue
Microstrip Line S11
Microstrip Line S21
Discontinuity S11
Discontinuity S21
Backplane S21
HT of Minimum Phase System

\[ |H_{ij}(s)| = |M_{ij}(s)| \| P_{ij}(s) \| e^{-s\tau_{ij}} \]

\[ |P_{ij}(j\omega)| = e^{-j\omega\tau_{ij}} = 1 \]

\[ s = j\omega \quad |H_{ij}(j\omega)| = |M_{ij}(j\omega)| \]

The phase of a minimum phase system can be completely determined by its magnitude via the Hilbert transform

\[ \arg[M_{ij}(\omega)] = \frac{2\omega}{\pi} \int_{0}^{\infty} \frac{U(\xi) - U(\omega)}{(\xi + \omega)(\xi - \omega)} d\xi \]

\[ U(\omega) = \ln |M_{ij}(\omega)| = \ln |H_{ij}(\omega)| \]
Enforcing Causality in TL

The complex phase shift of a lossy transmission line

\[ X = e^{-\gamma l} = e^{-\sqrt{(R+j\omega L)(G+j\omega C)}l} \]

is non causal

We assume that

\[ e^{-j\phi(\omega)} e^{-\alpha(\omega)} = e^{j\omega \sqrt{LC}l} e^{-\sqrt{(R+j\omega L)(G+j\omega C)}l} \]

is minimum phase non causal

\[ HT \left\{ \ln \left| e^{-j\phi(\omega)} e^{-\alpha(\omega)} \right| \right\} = HT \left\{ \ln \left| e^{-\gamma l} \right| \right\} = -\phi'(\omega) \]

\[ e^{-j\phi'(\omega)} e^{-\alpha(\omega)} \]

is minimum phase and causal

\[ X' = e^{-j\phi'(\omega)} e^{-\alpha(\omega)} e^{-j\omega \sqrt{LC}l} \]

is the causal phase shift of the TL

In essence, we keep the magnitude of the propagation function of the TL but we calculate/correct for the phase via the Hilbert transform.
Passivity Assessment

Can be done using $S$ parameter Matrix

$$D = \left( 1 - S^* S \right) = \text{Dissipation Matrix}$$

All the eigenvalues of the dissipation matrix must be greater than 0 at each sampled frequency points.

This assessment method is not very robust since it may miss local nonpassive frequency points between sampled points.

➤ Use Hamiltonian from State Space Representation
MOR and Passivity

\[ H(\omega) = \left[ A_1 + \sum_{i=1}^{L} \frac{a_{1i}}{1 + j\omega / \omega_{ci}} \right] \]
State-Space Representation

The State space representation of the transfer function is given by

\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) + Du(t) \]

The transfer function is given by

\[ S(s) = C \left( sI - A \right)^{-1} B + D \]
Procedure

• Approximate all $N^2$ scattering parameters using Vector Fitting

• Form Matrices $A$, $B$, $C$ and $D$ for each approximated scattering parameter

• Form $A$, $B$, $C$ and $D$ matrices for complete N-port

• Form Hamiltonian Matrix $H$
Constructing $A_{ii}$

Matrix $A_{ii}$ is formed by using the poles of $S_{ii}$. The poles are arranged in the diagonal.

$$A_{ii} = \begin{pmatrix}
a_1^{(ii)} & b_1^{(ii)} & 0 & 0 \\
-b_1^{(ii)} & a_1^{(ii)} & 0 & 0 \\
0 & 0 & \bullet & \bullet \\
0 & 0 & \bullet & a_L^{(ii)}
\end{pmatrix}$$

Complex poles are arranged with their complex conjugates with the imaginary part placed as shown.

$A_{ii}$ is an $L \times L$ matrix
Constructing $C_{ij}$

Vector $C_{ij}$ is formed by using the residues of $S_{ij}$.

$$C_{ij} = \left( c_{1}^{(ij)} \ c_{2}^{(ij)} \ \cdots \ c_{N}^{(ij)} \right)$$

where $c_{k}^{(ij)}$ is the $k$th residue resulting from the $L$th order approximation of $S_{ij}$

$C_{ij}$ is a vector of length $L$
Constructing $B_{ii}$

For each real pole, we have an entry with a 1

For each complex conjugate pole, pair we have two entries as:

$$B_{ii} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 0 \\ \cdot \\ 1 \end{pmatrix}$$

$B_{ii}$ is a vector of length $L$
Constructing $D_{ij}$

$D_{ij}$ is a scalar which is the constant term from the Vector fitting approximation:

$$S_{ij} = d_{ij} + \sum_{k=1}^{L} \frac{c_{k}^{(ij)}}{s - a_{k}^{(ii)}}$$

$$D_{ij} = d_{ij}$$
Constructing $\mathbf{A}$

Matrix $\mathbf{A}$ for the complete N-port is formed by combining the $A_{ii}$'s in the diagonal.

$$\mathbf{A} = \begin{pmatrix}
A_{11} & & \\
& A_{22} & \\
& & \ddots \\
& & & A_{NN}
\end{pmatrix}$$

$\mathbf{A}$ is a $NL \times NL$ matrix
Constructing $C$

Matrix $C$ for the complete N-port is formed by combining the $C_{ij}$'s.

$$
C = \begin{pmatrix}
C_{11} & C_{12} & \cdots \\
C_{21} & C_{22} & \\
\vdots & \ddots & \\
C_{NN} & & \\
\end{pmatrix}
$$

$C$ is a $N \times NL$ matrix
Constructing $B$

Matrix $B$ for the complete two-port is formed by combining the $B_{ii}$'s.

\[
B = \begin{pmatrix}
B_{11} & 0 & 0 \\
0 & B_{22} & 0 \\
\cdot & \cdot & \cdot \\
0 & 0 & B_{NN}
\end{pmatrix}
\]

$B$ is a $NL \times N$ matrix
Hamiltonian

Construct Hamiltonian Matrix $M$

$$M = \begin{bmatrix}
A - BR^{-1} D^T C & -BR^{-1} B^T \\
C^T S^{-1} C & -A^T + C^T DR^{-1} B^T
\end{bmatrix}$$

$$R = \left( D^T D - I \right) \text{ and } S = \left( DD^T - I \right)$$

The system is passive if $M$ has no purely imaginary eigenvalues.

If imaginary eigenvalues are found, they define the crossover frequencies ($j\omega$) at which the system switches from passive to non-passive (or vice versa).

→ gives frequency bands where passivity is violated
Perturb Hamiltonian

Perturb the Hamiltonian Matrix $M$ by perturbing the pole matrix $A$

$$A \rightarrow A' = A + \Delta A$$

$$M + \Delta M = \begin{bmatrix}
A + \Delta A - B \left(D + D^T\right)^{-1} C & B \left(D + D^T\right)^{-1} B^T \\
-C^T \left(D + D^T\right)^{-1} C & -\left(A + \Delta A\right)^T + C^T \left(D + D^T\right)^{-1} B^T
\end{bmatrix}$$

$$\Delta M = \begin{bmatrix}
\Delta A & 0 \\
0 & -\left(\Delta A\right)^T
\end{bmatrix}$$

This will lead to a change of the state matrix:

$$A \rightarrow A' = A + \Delta A$$
State-Space Representation

The State space representation of the transfer function in the time domain is given by

\[ \dot{x}(t) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) + Du(t) \]

The solution in discrete time is given by

\[ x[k+1] = A_d x[k] + B_d u[k] \]
\[ y[k] = C_d x[k] + D_d u[k] \]
State-Space Representation

where

\[ A_d = e^{A^T} \quad B_d = \left( \int_0^T e^{A\tau} d\tau \right) B \]

\[ C_d = C \quad D_d = D \]

which can be calculated in a straightforward manner

When \( y(t) \rightarrow b(t) \) is combined with the terminal conditions, the complete blackbox problem is solved.
State-Space Passive Solution

If $M'$ is passive, then the state-space solution using $A'$ will be passive.

$$A \rightarrow A' = A + AA$$

The *passive* solution in discrete time is given by

$$x[k + 1] = A'_d x[k] + B_d u[k]$$

$$y[k] = C_d x[k] + D_d u[k]$$
Size of Hamiltonian

\[ M = \begin{bmatrix} 
A - BR^{-1}D^T C & -BR^{-1}B^T \\
C^T S^{-1} C & -A^T + C^T DR^{-1} B^T 
\end{bmatrix} \]

M has dimension 2NL

For a 20-port circuit with VF order of 40, M will be of dimension \(2 \times 40 \times 20 = 1600\)

The matrix M has dimensions 1600 \(\times\) 1600

Too Large!

➔ Eigen-analysis of this matrix is prohibitive
Example

\[
A = \begin{bmatrix}
-10 & 0 & 0 \\
0 & -1 & 100 \\
0 & -100 & -1
\end{bmatrix}
\quad \quad \quad
B = \begin{bmatrix}
1 \\
2 \\
0
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
1 & 1 & 0.1
\end{bmatrix}
\quad \quad \quad
D = \begin{bmatrix}
10^{-5}
\end{bmatrix}
\]

This macromodel is nonpassive between 99.923 and 100.11 radians
Example

\[ A' = \begin{bmatrix} -10 & 0 & 0 \\ 0 & -1 - 0.005 & 100 \\ 0 & -100 & -1 - 0.005 \end{bmatrix} \]

The Hamiltonian \( M' \) associated with \( A' \) has no pure imaginary \( \Rightarrow \) System is passive
Passivity Enforcement Techniques

- Hamiltonian Perturbation Method (1)
- Residue Perturbation Method (2)


### Benchmarks*

<table>
<thead>
<tr>
<th>Data file</th>
<th>No. of points</th>
<th>MOR with Vector Fitting</th>
<th>Fast Convolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Order</td>
<td>VFIT(^\d)</td>
<td>Passivity Enforcement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time (s)</td>
<td></td>
</tr>
<tr>
<td>Blackbox 1</td>
<td>501</td>
<td>10*</td>
<td>0.14</td>
</tr>
<tr>
<td>Blackbox 2</td>
<td>802</td>
<td>20*</td>
<td>0.41</td>
</tr>
<tr>
<td>Blackbox 3</td>
<td>802</td>
<td>40*</td>
<td>1.08</td>
</tr>
<tr>
<td>Blackbox 4</td>
<td>802</td>
<td>60*</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>3.17</td>
<td>5.34</td>
</tr>
<tr>
<td>Blackbox 5</td>
<td>2002</td>
<td>50*</td>
<td>4.97</td>
</tr>
<tr>
<td>Blackbox 6</td>
<td>802</td>
<td>100*</td>
<td>3.17</td>
</tr>
<tr>
<td>Blackbox 7</td>
<td>1601</td>
<td>100*</td>
<td>24.59</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>31.16</td>
<td>27.64</td>
</tr>
<tr>
<td>Blackbox 8</td>
<td>5096</td>
<td>220</td>
<td>250.08</td>
</tr>
<tr>
<td>Blackbox 9</td>
<td>1601</td>
<td>200*</td>
<td>58.47</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>80.64</td>
<td>122.83</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>106.53</td>
<td>61.58(^{\dagger})</td>
</tr>
</tbody>
</table>

Passive VF Simulation Code

- PerformsVF with commonpoles
- Assessment via Hamiltonian
- Enforcement: Residue Perturbation Method
- Simulation: Recursive convolution

<table>
<thead>
<tr>
<th>Number of Ports</th>
<th>Order</th>
<th>CPU-Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Port</td>
<td>20</td>
<td>1.7 secs</td>
</tr>
<tr>
<td>6-port</td>
<td>32</td>
<td>3.69 secs</td>
</tr>
<tr>
<td>10-port</td>
<td>34</td>
<td>8.84 secs</td>
</tr>
<tr>
<td>20-port</td>
<td>34</td>
<td>33 secs</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>142 secs</td>
</tr>
<tr>
<td>80</td>
<td>12</td>
<td>255 secs</td>
</tr>
</tbody>
</table>
Passive VF Code - Examples

Example 1
4 ports
order = 60

Example 2
40 ports
order = 50
Passivity Enforced VF

4 ports, 2039 data points - VFIT order = 60 (4 iterations ~6-7mins), Passivity enforcement: 58 Iterations (~1hour)
Passive Time-Domain Simulation

![Graph showing voltage and time for port 1 (original) and port 2 (passive).]
40-Port Passivity Enforced VF

Magnitude of $S_{1-21}$

-0.252
-0.505
-0.757
-1.010
-1.262
-1.515
-1.767
-2.020

Approx
Actual

Time [ns]
40-Port Passivity Enforced VF

Phase of $S_{1-21}$
40-Port Passivity Enforced VF

Phase of $S_{21}$
40-Port Passivity Enforced VF

Magnitude of $S_{21}$
40-Port Passivity Enforced VF

Magnitude of $S_{11}$
40-Port Passivity Enforced VF

Phase of $S_{11}$
40-Port Time-Domain Simulation
40-Port Time-Domain Simulation

Voltage [V]

Time [ns]

Port 1
Port 1
Port 2
Port 2