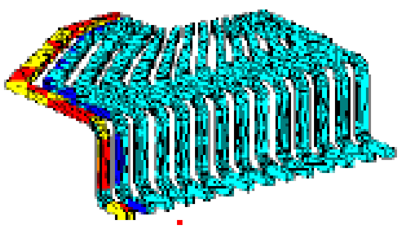
Circuit Synthesis of Multiport Networks from Passive Poles and Residues

Chiu-Chih (George) Chou
National Central University, Taoyuan, Taiwan
José E. Schutt-Ainé,
University of Illinois, Urbana, IL

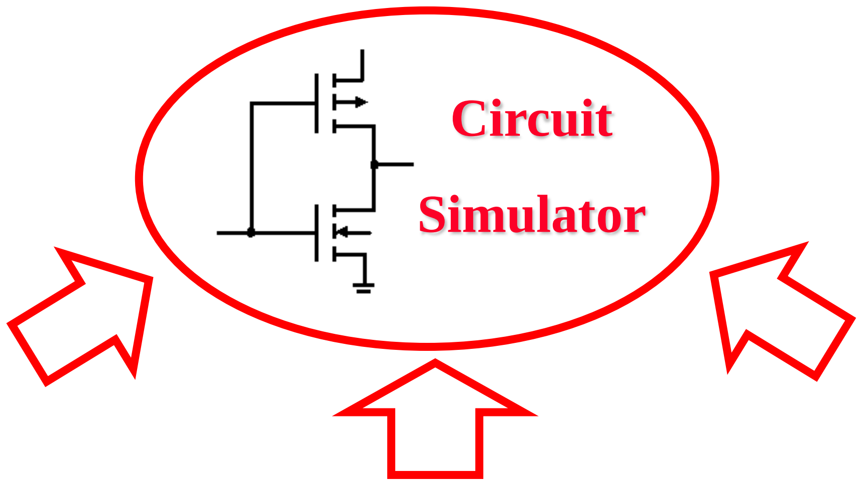
IBIS Summit at DesignCon 2022
Santa Clara, CA
Virtual
April 8, 2022

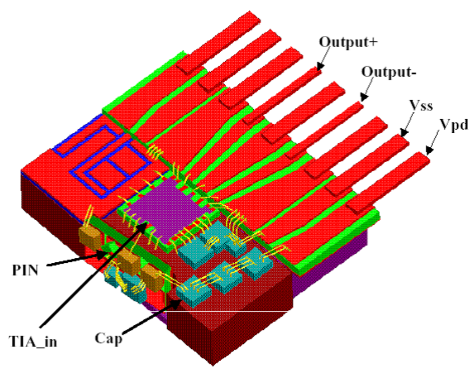
Interconnect Structures

Crosstalk
Couplings
Reflections
Losses
Dispersion

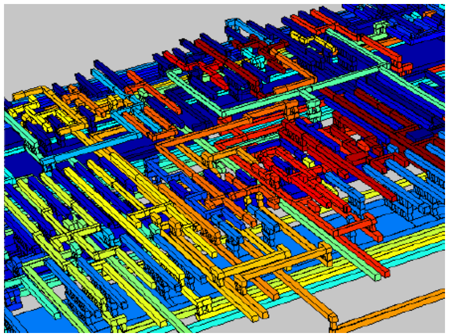


Packages



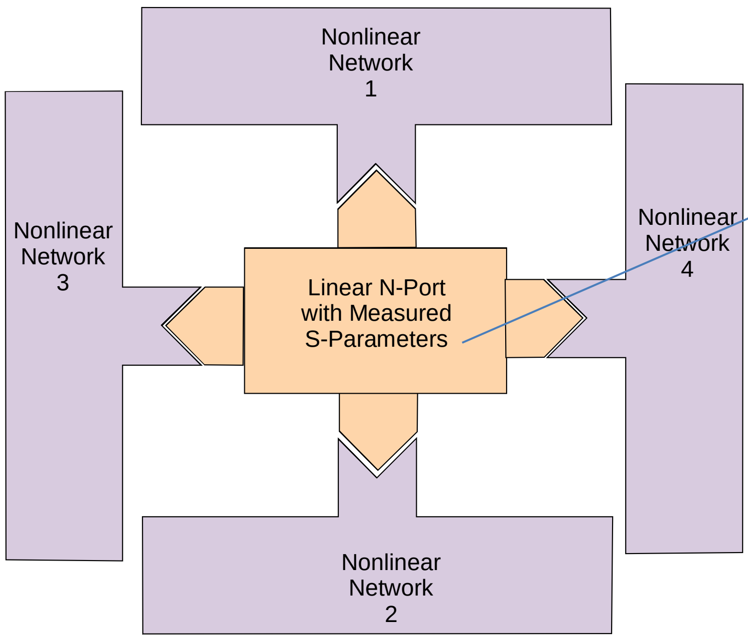


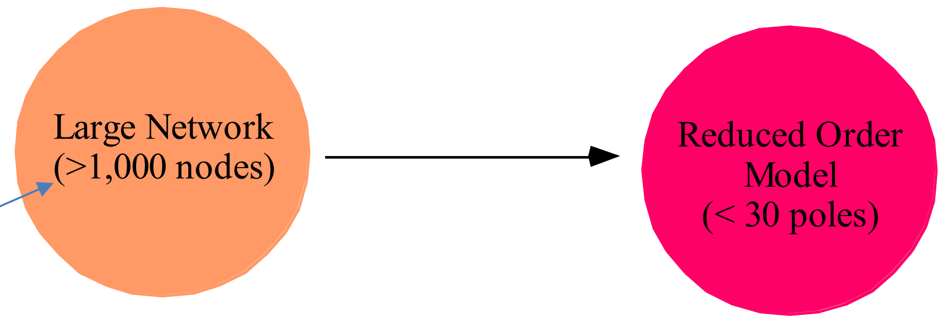
Ground Noise
Nonlinear effects
Radiation, EMI



Interconnects

Model-Order Reduction





SPICE

$Y(t)\, v(t) = i(t)$

$$\tilde{Y}(\omega) = \left[A_1 + \sum_{i=1}^{L} \frac{a_{1i}}{1 + j\omega / \omega_{c1i}} \right]$$

$Y(\omega)\, V(\omega) = I(\omega)$

↓

Order Reduction

↓

$Y(\omega) \simeq \tilde{Y}(\omega)$

↓

Recursive Convolution

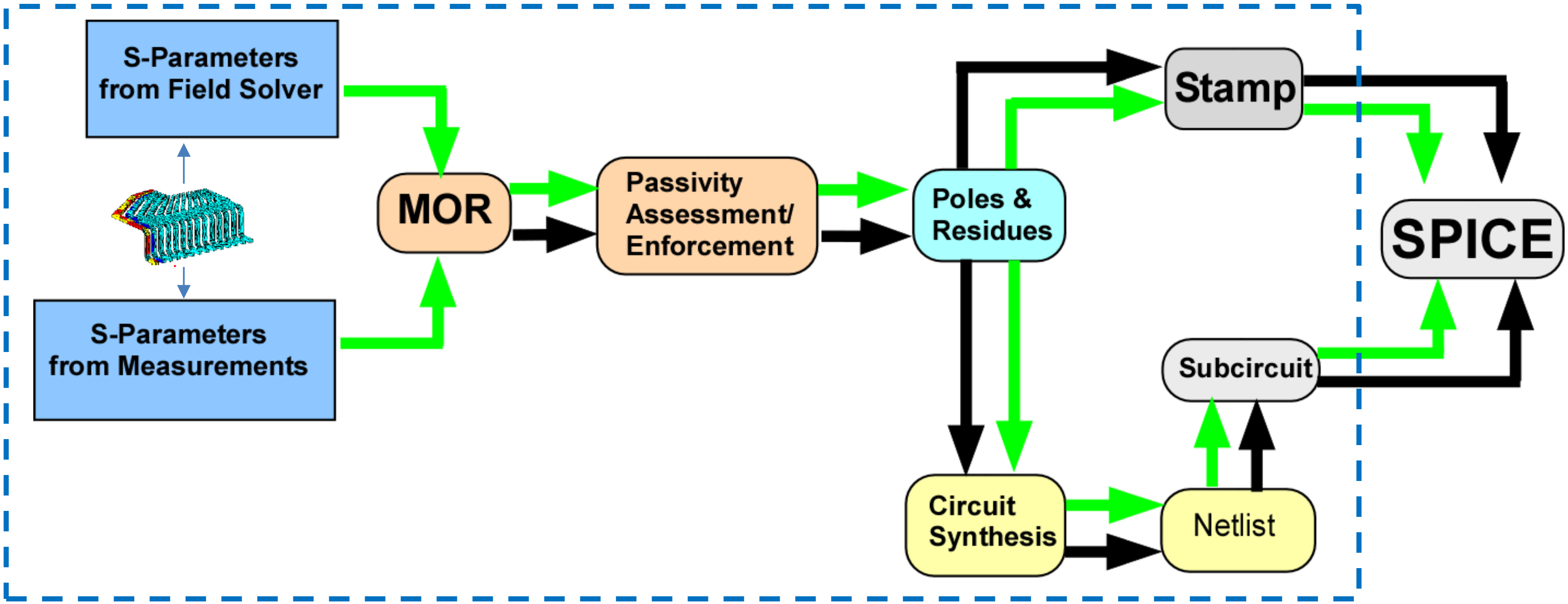
↓

$\tilde{Y}(t)\, v(t) = i(t)$

- **Strategy**: Use reduced order model to minimize computation time.

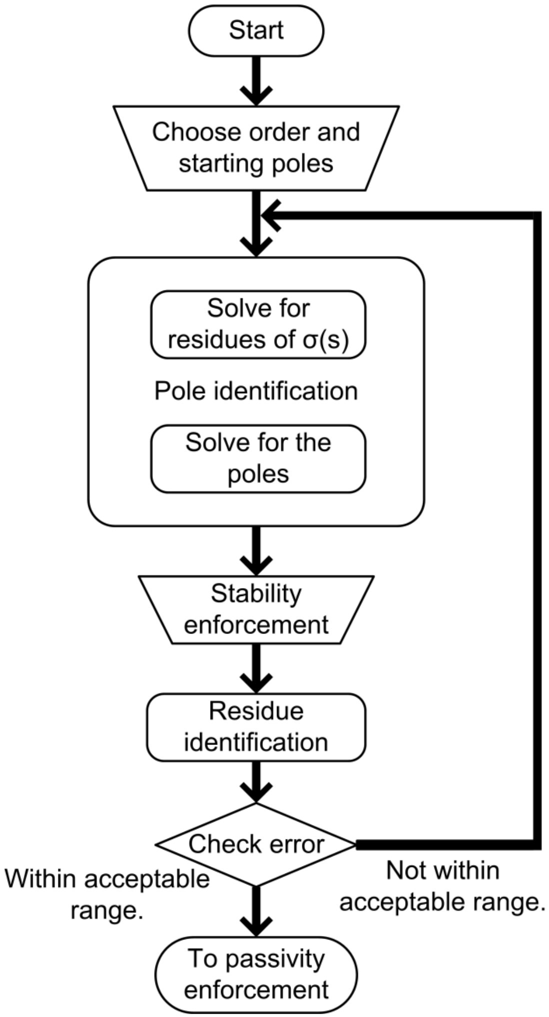
$$\tilde{Y}(\omega) \approx Y(\omega)$$

Model-Order Reduction



- **Objective**: Incorporate frequency dependence into time-domain simulator
- **Approaches**: 1) Direct integration of code into SPICE – 2) Generation of SPICE-compatible netlist

MOR via Vector Fitting



- Rational function approximation:

$$f(s) \approx \sum_{n=1}^{N} \frac{c_n}{s-a_n} + d + sh$$

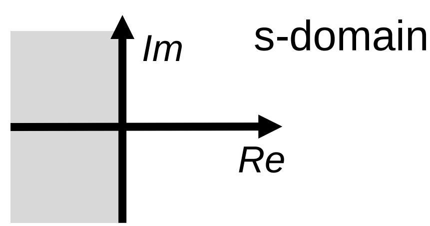
- Introduce an unknown function *σ(s)* that satisfies:

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

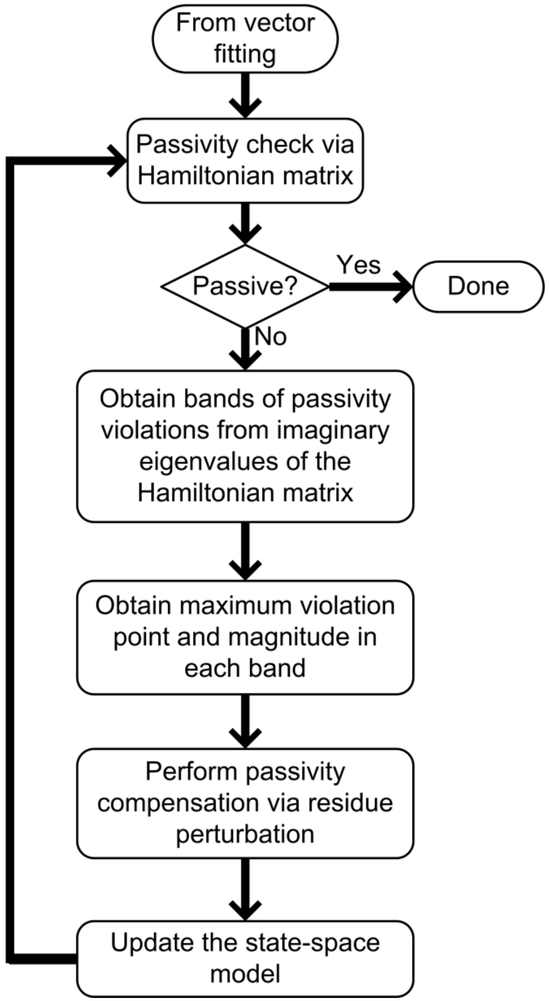
- Poles of *f(s)* = zeros of *σ(s)*:

$$f(s) \approx \frac{\displaystyle\sum_{n=1}^{N} \frac{c_n}{s-\tilde{a}_n} + d + sh}{\displaystyle\sum_{n=1}^{N} \frac{\tilde{c}_n}{s-\tilde{a}_n} + 1} = \frac{\displaystyle\prod_{n=1}^{N+1} (s - z_n)}{\displaystyle\prod_{n=1}^{N} (s - \tilde{z}_n)}$$

- Flip unstable poles into the left half plane.



Passivity Enforcement



- State-space form:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

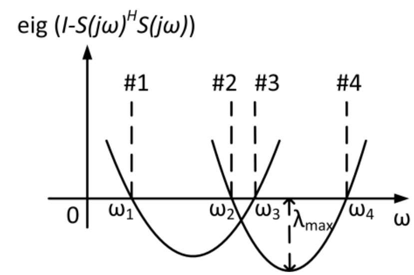
- Hamiltonian matrix:

$$M = \begin{bmatrix} A + BKD^T C & BKB^T \\ -C^T LC & -A^T - C^T DKB^T \end{bmatrix}$$

$$K = \left(I - D^T D\right)^{-1} \quad L = \left(I - DD^T\right)^{-1}$$

- Passive if *M* has no imaginary eigenvalues.

- Sweep: $eig\left(I - S(j\omega)^H S(j\omega)\right)$



- Quadratic programming:
  - Minimize *(change in response)* subject to *(passivity compensation).*

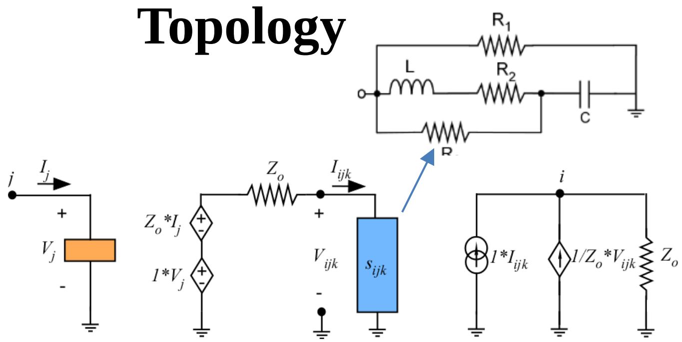
$$\min\left(vec(\Delta \mathbf{C})^{\mathbf{T}}\mathbf{H}\, vec(\Delta \mathbf{C})\right) \quad \textbf{subject to} \quad \Delta\lambda = G \cdot vec(\Delta \mathbf{C}).$$

SPICE Netlist Synthesis

- Goal is to generate (using pole/residue information) a circuit netlist that will exhibit the same (frequency-dependent) behavior as that of the S-parameters of connector under study

Passive Poles/Residues from MOR

Circuit Cell Topology

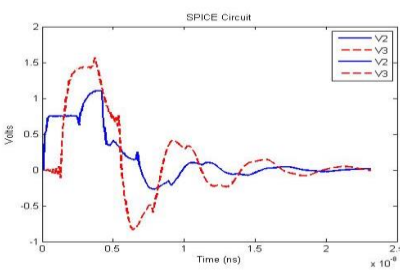


Synthesis Algorithm

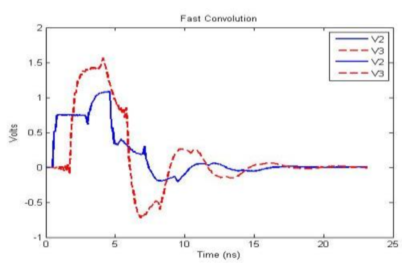
Netlist

Simulation

SPICE simulation from MOR generated Netlist



Direct Convolution



Equivalent-Circuit Extraction

Macromodel is curve-fit to take the form

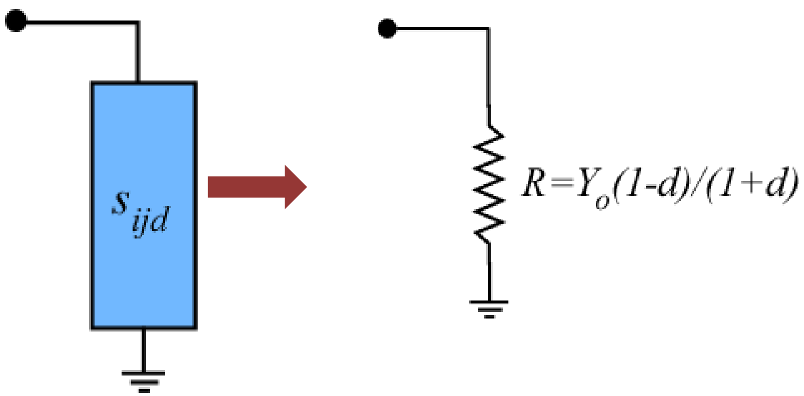
$$S(s) = d + \sum_{k=1}^{L} \frac{r_k}{s - p_k}$$

Need to find equivalent circuit associated with

- Constant term d
- Real Poles
- Complex Poles

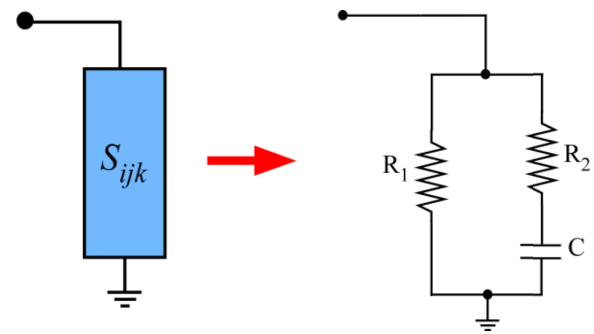
Equivalent-Circuit Extraction

Constant Term



$$R = Y_o\left(\frac{1-d}{1+d}\right)$$

Real Poles



$$R_2 = \frac{-1}{bC}$$

$$R_1 = -R_2 - \frac{1}{aC}$$

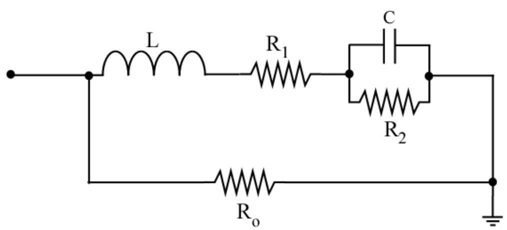
$$C = -\frac{(b-a)}{b^2 Z_o}$$

$$a = p_k + r_k, \quad \text{and} \quad b = p_k - r_k$$

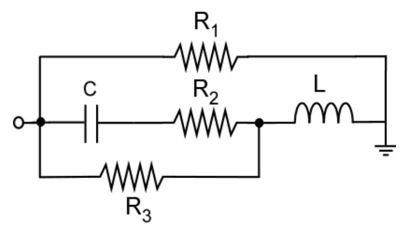
Realization – Complex Poles

There are several circuit topologies that will work

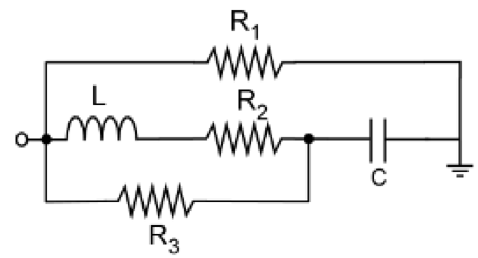
Model 1



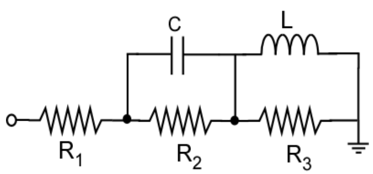
Model 8



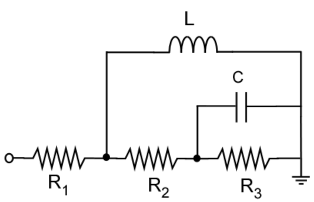
Model 9



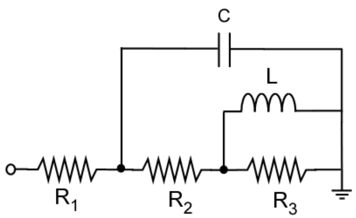
Model 11



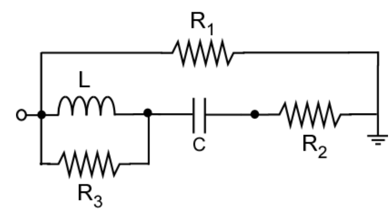
Model 12



Model 13



Model 10

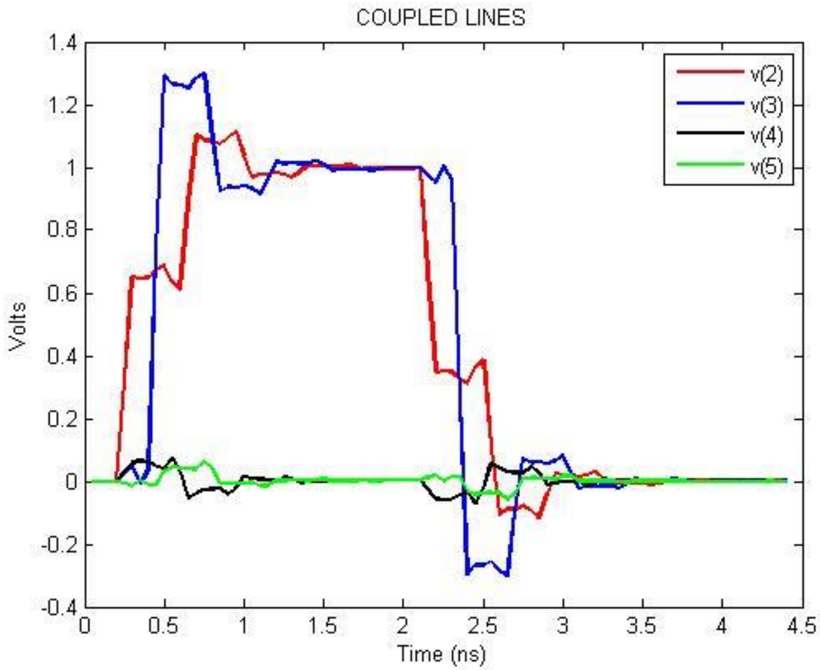


Netlist from Poles & residues

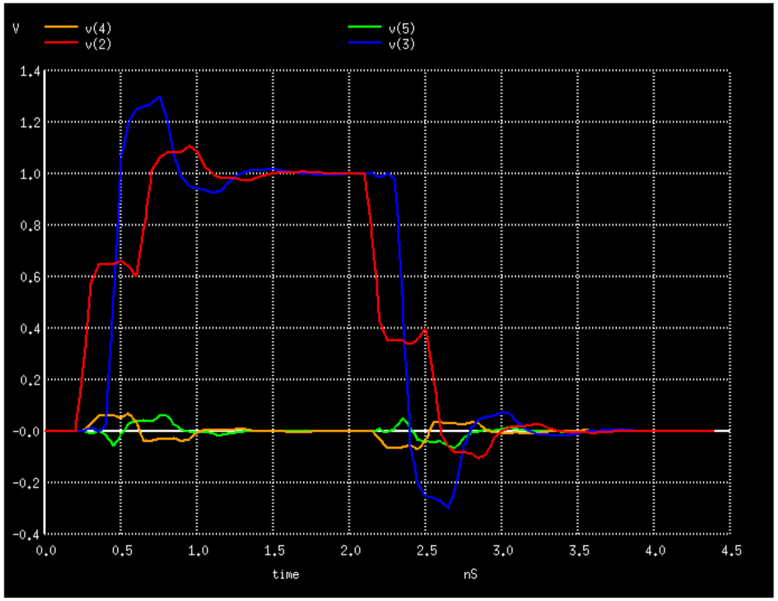
```
*Poll   2-port S-parameter circuit model
* 14 -pole approximation

.subckt Poll  42000  56000
vsens42001 42000 42001 0.0
vsens56001 56000 56001 0.0

*subcircuit for s[1][1]
*complex residue-pole pairs for S[1][1] at k=  1 -> 1st pole: -4.8961e+00 3.6506e+01 residue: 2.1006e-01 -2.8971e-01
*                                              -> 2nd pole: -4.8961e+00 -3.6506e+01 residue: 2.1006e-01 2.8971e-01
*circuit type =   9
elc1  1 0 42001 0 1.0
hc2  2  1 vsens42001 50.0
rtersc3  2  3 50.0
vp4  3  4 0.0
r1cd5  4  0 5.17406e+01
l1cd5  4  5 -1.25500e-08
r2cd6  5  6 -1.30103e+03
c1cd6  6  0 -7.19920e-15
r3cd6  4  6 1.48633e+03
*complex residue-pole pairs for S[1][1] at k=  2 -> 1st pole: -1.3039e+00 2.7679e+01 residue: -4.3856e-01 -1.9087e+00
*                                              -> 2nd pole: -1.3039e+00 -2.7679e+01 residue: -4.3856e-01 1.9087e+00
rtersc9  8  9 50.0
:
:
gs196    0 56001 196 0 0.020
rnort42001 42001 0 5.00000e+01
rnort56001 56001 0 5.00000e+01
.ends Poll
*main circuit
rgen 1 2 50.0
x1  2 3 Poll
vin 1 0  pulse (0 1  0.20000ns  0.10000ns  0.10000ns  2.00000ns  6.00000ns)
rport2 3 0 50000.0000000
.tran   0.00039ns  7.00000ns
.end
```

4-Port Network

Direct

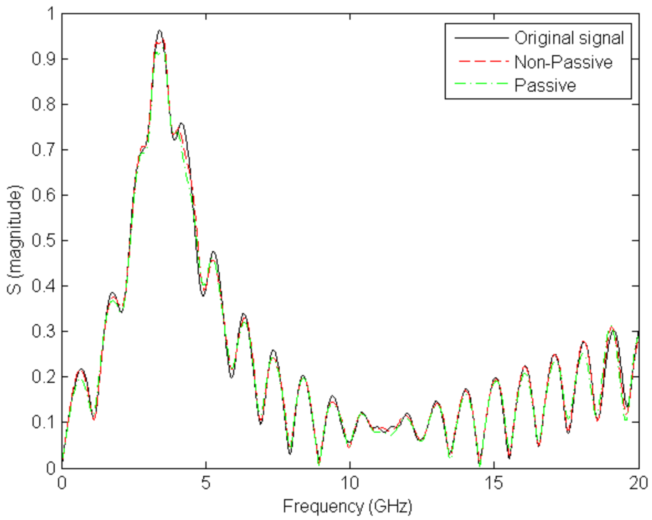


SPICE simulation Using generated netlist (Method 2)



Model-Order Reduction

- ➔ Start with S parameters from field solver
- ➔ Use vector fitting to get poles & residues
- ➔ Perform assessment via Hamiltonian
- ➔ Enforcement: Residue Perturbation Method
- ➔ Simulation: Recursive convolution➔Fast



Number of Ports	Order	CPU-Time
4	20	1.7 secs
6	32	3.69 secs
10	34	8.84 secs
20	34	33 secs
40	50	142 secs
80	12	255 secs

Review of some classic synthesis approaches for S matrix in pole-residue form*

1. (PI network for Y matrix)
2. PI network for S by Y + VCVS + CCVS
3. State-space S
4. State-space S-to-Y then PI
5. Pole-residue S-as-Y
6. Direct pole-residue specification

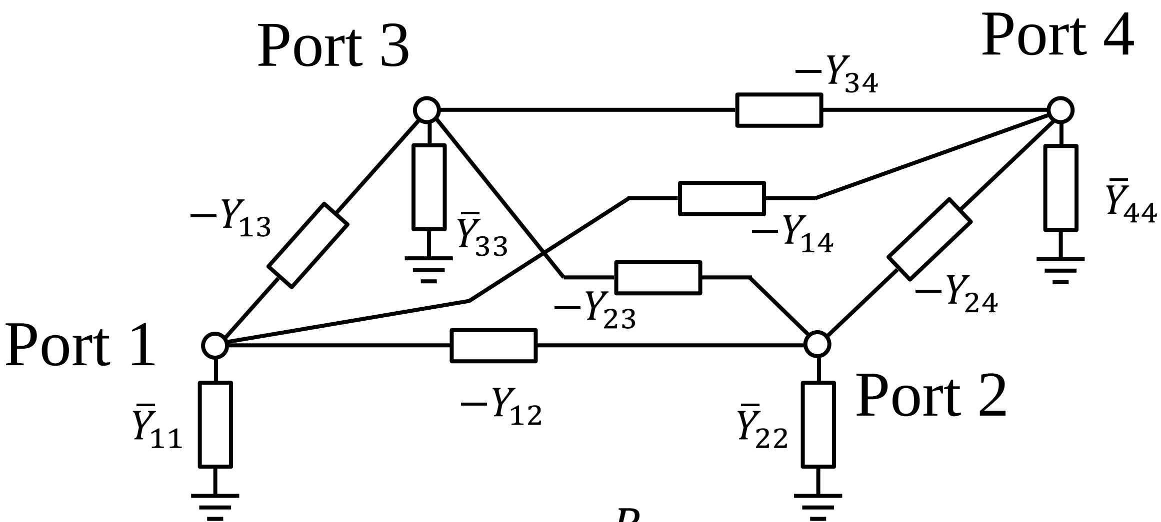
$$S_{ij} = d^{(ij)} + \sum_{k=1}^{N} \frac{r_k^{(ij)}}{s - p_k^{(ij)}} = d^{(ij)} + \sum_{k=1}^{N} S_{ij,k}$$

* Chiu-Chih Chou, José E. Schutt-Ainé, "Equivalent Circuit Synthesis of Multiport S Parameters in Pole–Residue Form", *IEEE Transactions on Components, Packaging and Manufacturing Technology,* Volume 11, Issue: 11, pp. 1971-1979, 2021, November 2021

Model 1. PI network for Y matrix (1/2)

$$Y_{ij} = d^{(ij)} + \sum_{k=1}^{N} \frac{r_k^{(ij)}}{s - p_k^{(ij)}}$$

Pole-residue Y matrix ➔ PI model (direct correspondence)

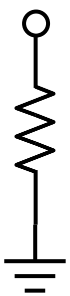


$$\bar{Y}_{ii} = \sum_{j=1}^{P} Y_{ij}$$

Model 1. PI network for Y matrix (2/2) (no controlled sources, but have negative elements)

Constant

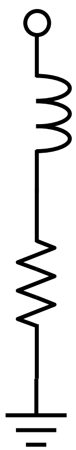
$$Y = d$$



$$R = \frac{1}{d}$$

A real pole

$$Y = \frac{r}{s-p}$$

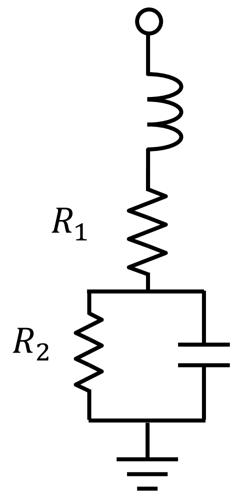


$$L = \frac{1}{r}$$

$$R = \frac{-p}{r}$$

A pair of complex conjugate poles (RLCR)

$$Y = \frac{r}{s-p} + \frac{r^*}{s-p^*}$$



$$L = \frac{1}{2r_r}$$

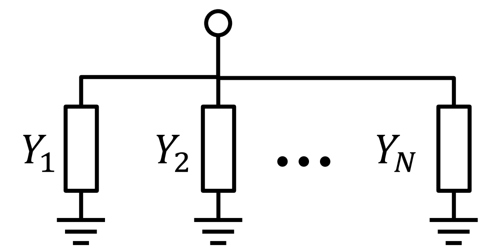
$$R_1 = \frac{-p_r r_r + p_i r_i}{2r_r^2}$$

$$C = \frac{-r_r}{(p_r r_r + p_i r_i)} \cdot \frac{1}{R_2}$$

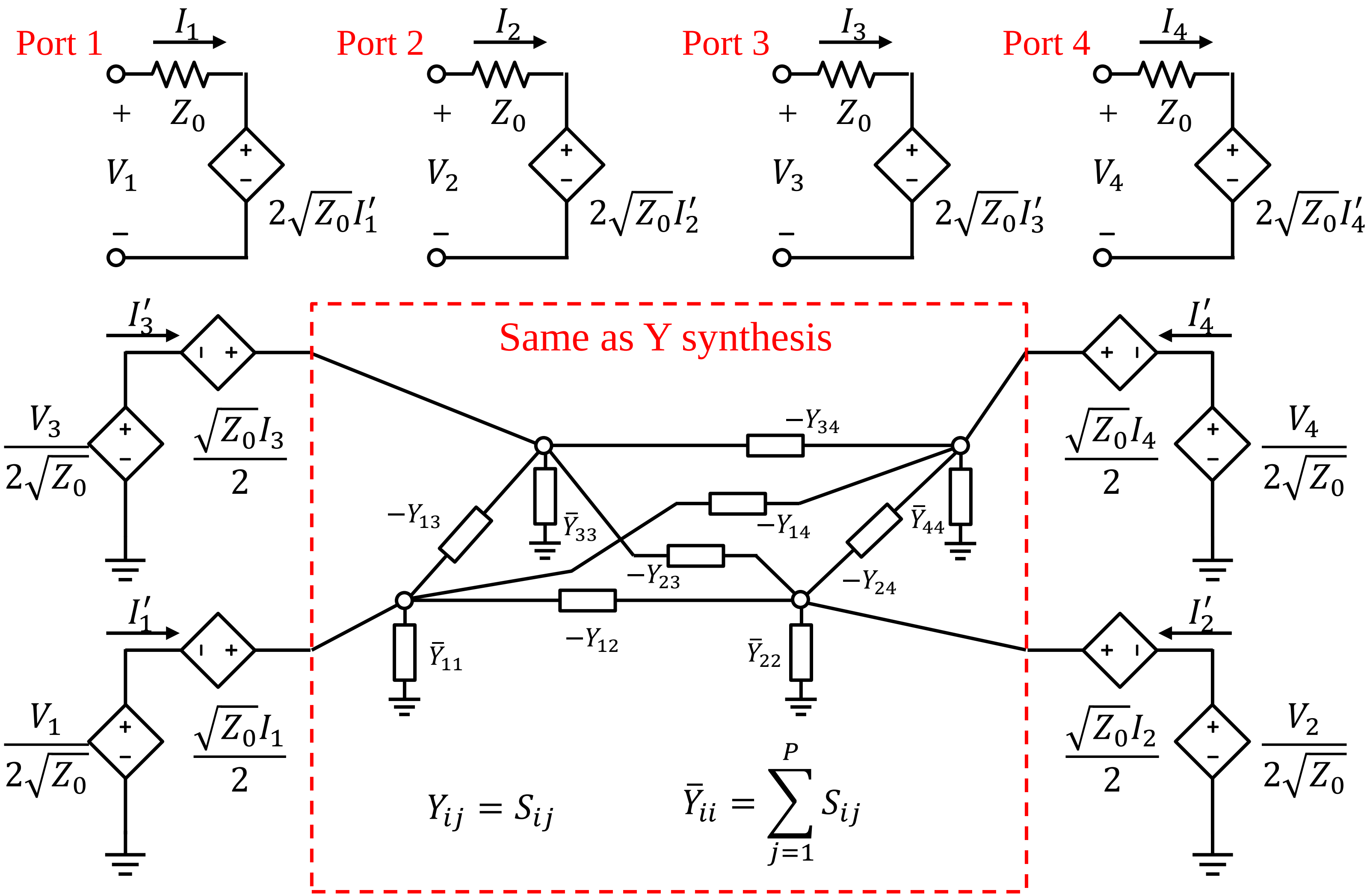
$$R_2 = -\frac{p_r^2 + p_i^2}{2(p_r r_r + p_i r_i)} - R_1$$

N pole-residue pairs

$$Y = \sum_{k=1}^{N} Y_k$$



Model 2. PI network for S by Y + VCVS + CCVS

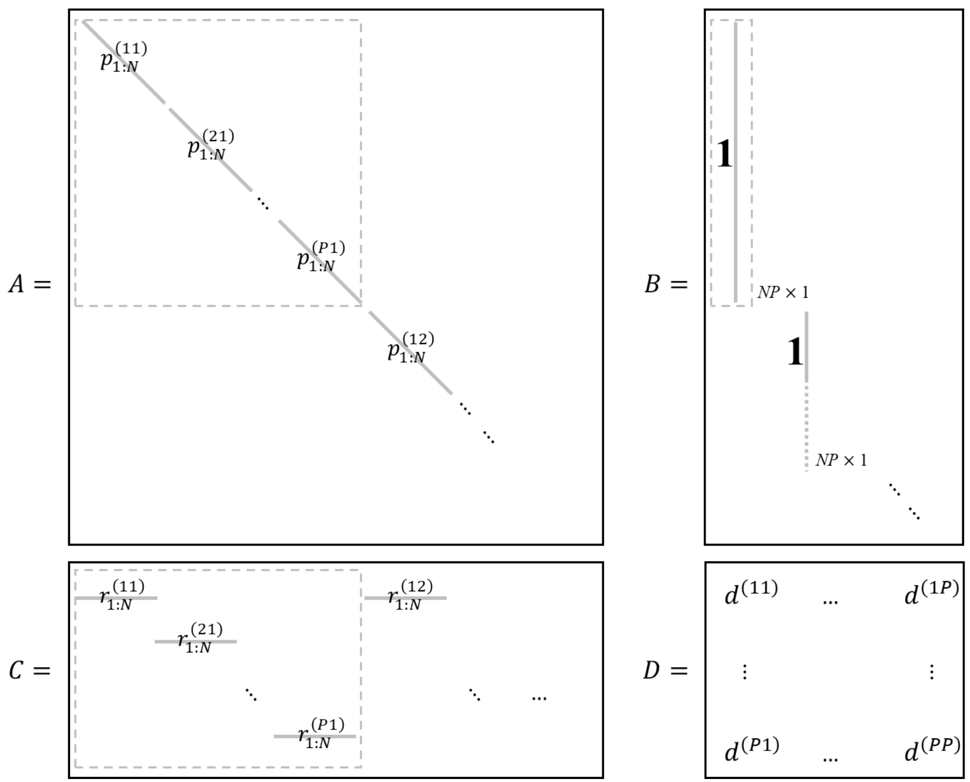


Model 3. State-space S (1/2) (a common cross-platform topology)

$$S_{ij} = d^{(ij)} + \sum_{k=1}^{N} \frac{r_k^{(ij)}}{s - p_k^{(ij)}}$$

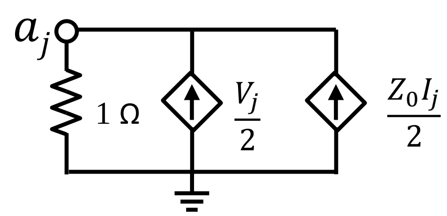
Pole-residue
to state-space

$$\begin{cases} A = \operatorname*{diag}_{\substack{j=1\ldots P \\ i=1\ldots P \\ k=1\ldots.N}} p_k^{(ij)} \\ B_{nj} = 1_{\{(j-1)NP<n\le jNP\}} \\ C_{in} = r_k^{(ij)} \cdot 1_{\{n\,=\,(j-1)NP+(i-1)N+k\}} \\ D_{ij} = d^{(ij)} \end{cases}$$



Model 3. State-space S (2/2)

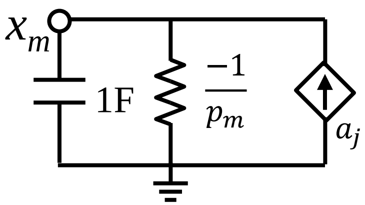
Input equations (incident wave):



State equations:

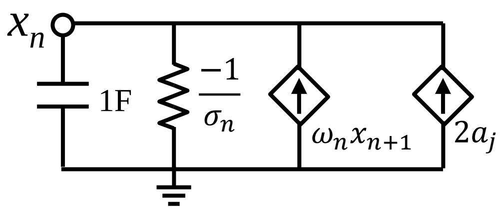
real pole p_m

$\dot{x}_m = p_m x_m + a_j$

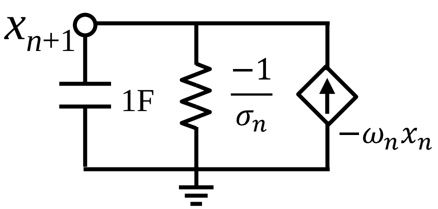


complex-conjugate poles $p_n = \sigma_n + j\omega_n, \quad p_{n+1} = \sigma_n - j\omega_n$

$\dot{x}_n = \sigma_n x_n + \omega_n x_{n+1} + 2a_j$

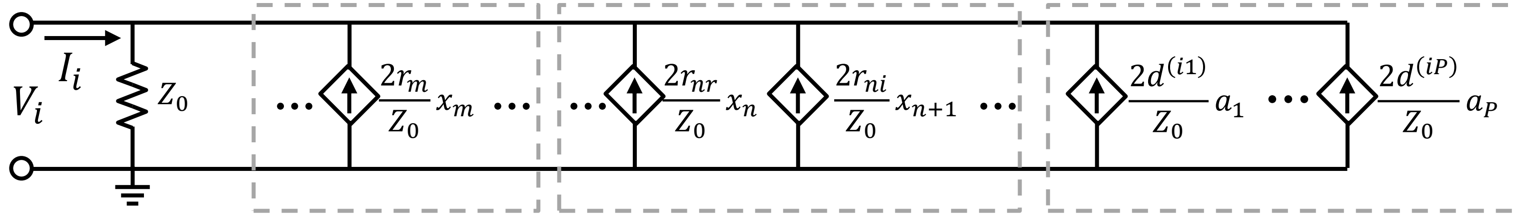


$\dot{x}_{n+1} = \sigma_n x_{n+1} - \omega_n x_n$



Output equations:

Port *i* | real poles | complex-conjugate poles | direct coupling terms



Model 4. State-space S to Y then PI (no controlled sources needed)

State-space for S matrix

$$\begin{cases} A = \operatorname{diag}\limits_{\substack{j=1\dots P \\ i=1\dots P \\ k=1\dots N}} p_k^{(ij)} \\ B_{nj} = 1_{\{(j-1)NP<n\le jNP\}} \\ C_{in} = r_k^{(ij)} \cdot 1_{\{n\ =\ (j-1)NP+(i-1)N+k\}} \\ D_{ij} = d^{(ij)} \end{cases}$$

State-space for Y matrix

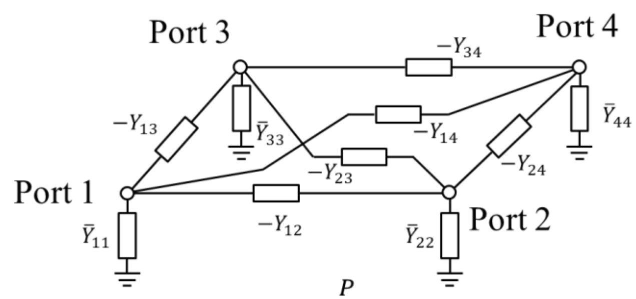
$$\begin{cases} A' = A - B(I+D)^{-1}C \\ B' = \dfrac{1}{\sqrt{Z_0}} B(I+D)^{-1} \\ C' = \dfrac{-2}{\sqrt{Z_0}} (I+D)^{-1}C \\ D' = \dfrac{1}{Z_0}(I-D)(I+D)^{-1} \end{cases}$$

Diagonalization

Pole-residue for Y matrix

$$Y_{ij} = d^{(ij)} + \sum_{k=1}^{N} \frac{r_k^{(ij)}}{s - p_k^{(ij)}}$$

PI model

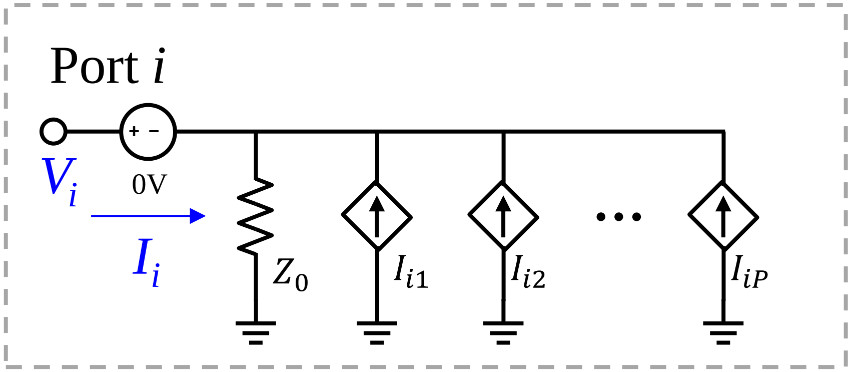


$$\bar{Y}_{ii} = \sum_{j=1}^{P} Y_{ij}$$

Problem: $(I + D)$ may be singular!

Model 5. Pole-residue S-as-Y (minimized for SISO pole)

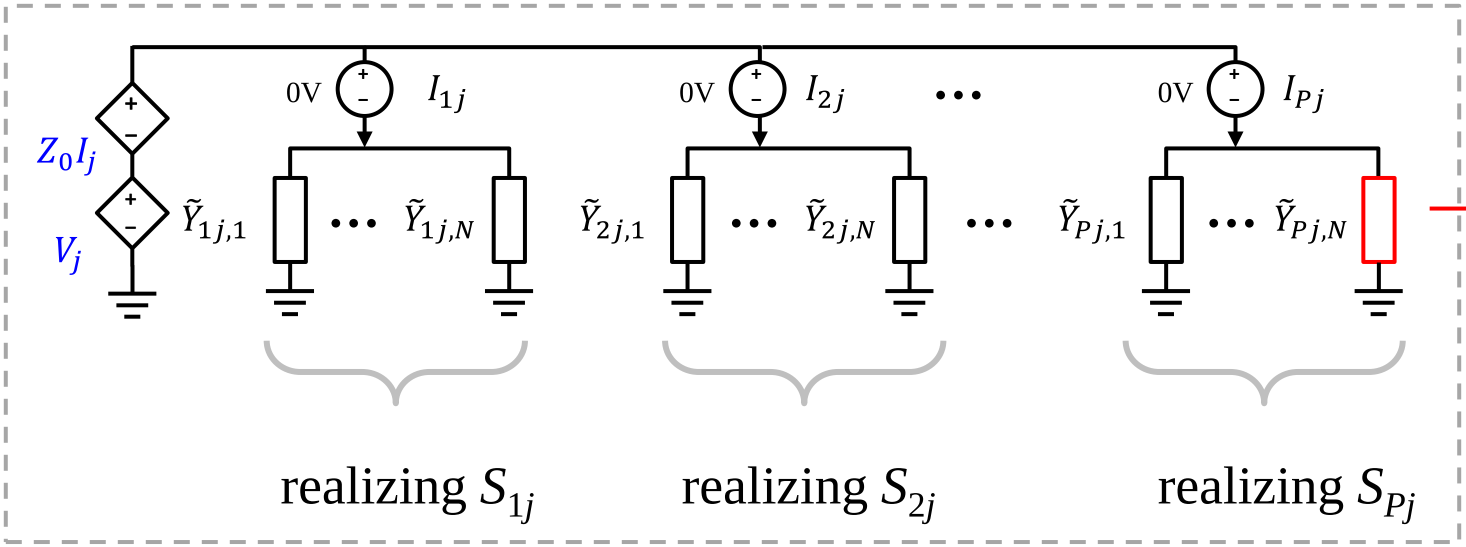
Port sections: $i = 1, 2, \ldots, P$



Use the same approach as model 1 (Y branch)

$$\tilde{Y}_{ij,k} \triangleq \frac{1}{Z_0} S_{ij,k}$$

Intermediate stages: $j = 1, 2, \ldots, P$

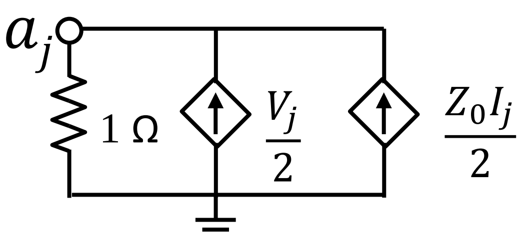


Model 6. Direct pole-residue specification (permit recursive convolution)

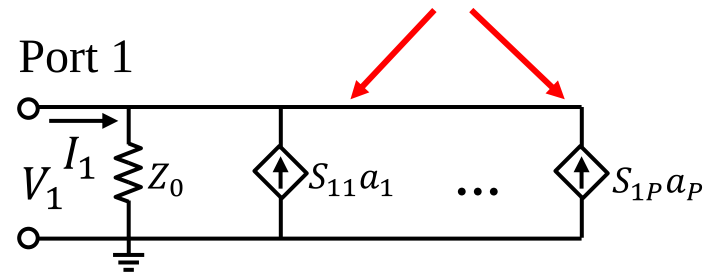
$$S_{ij} = d^{(ij)} + \sum_{k=1}^{N} \frac{r_k^{(ij)}}{s - p_k^{(ij)}}$$

Not all simulator support this format!

Incident wave calculator for each port



Foster G element




```
Gs_1_1  gnd_0  p_1  FOSTER  inc_1  0  -5.69243475541412 6e-03  0
+ ( -2.43915043185622 2e+04, 0 ) / ( -4.96047604161921 0e+07, 0 )
+ (  6.16073177854365 6e+07, 0 ) / ( -1.71417039466460 3e+09, 0 )
+ ( -1.89478842252053 8e+07,  5.66229376645860 2e+07 ) / ( -1.67218
+ (  8.71396634829680 2e+07, -8.13962212929225 4e+07 ) / ( -1.84733
+ (  2.61717341198152 7e+08,  1.35846443493299 4e+08 ) / ( -2.17708
+ (  1.63354379245143 9e+07,  6.27157348818774 0e+08 ) / ( -2.57889
+ ( -7.92545322499179 8e+08, -3.73145747944773 3e+07 ) / ( -2.81115
+ ( -1.39341694575600 4e+08, -3.07839808832204 8e+08 ) / ( -2.26115
+ (  3.61228306947034 6e+06, -4.99390841438674 2e+06 ) / ( -1.35763
+ ( -1.35975093314697 7e+07, -2.87575264814606 5e+08 ) / ( -2.39584
```

Comparison

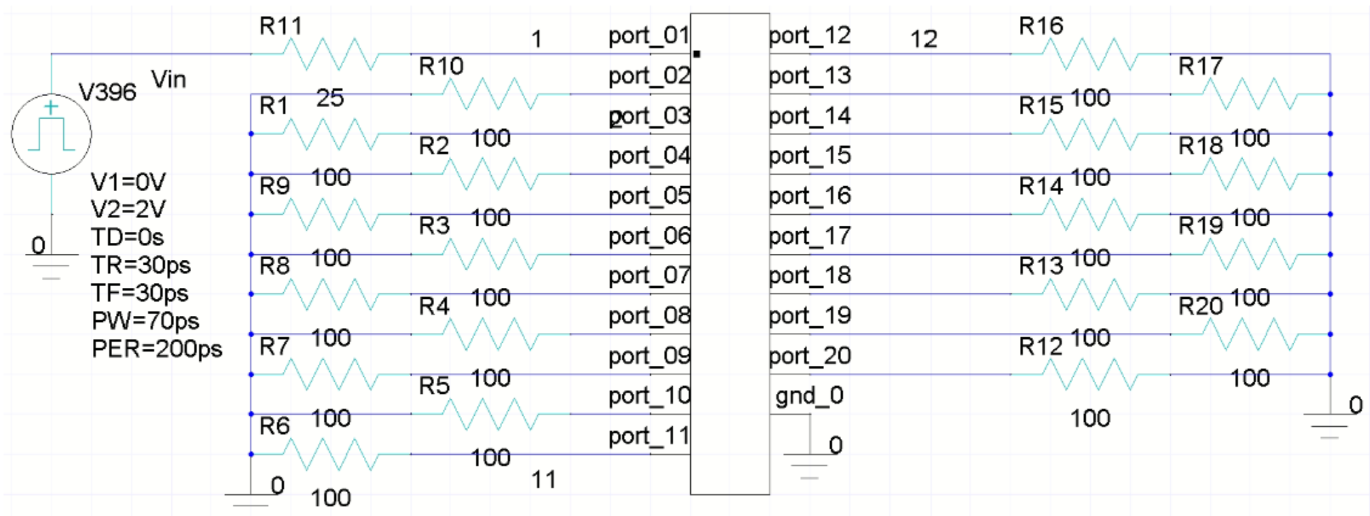
	Model 2	Model 3	Model 4	Model 5	Model 6
	PI network for S by Y + VCVS+CCVS	State-space S	State-space S to Y	Pole-residue S-as-Y	Direct pole-residue specification
Controlled sources	Yes	Yes	No	Yes	Yes
Negative RLC	Yes	No	Yes	Yes	No
Recursive convolution	No	No	No	No	Yes
Cross platform	Yes*	Yes	Yes*	Yes*	No

* if negative RLC permitted

Offered by many commercial EDA tools.

Example: macromodel of 20 ports and 110 MIMO poles

(10 coupled microstrips)



TRAN
Tstep = 1ps
Tstop = 100ns

Total simulation time (s)

	State-space S (model 3)	PI network (model 2)	Foster (model 6)
EDA Tool A	338	815	190
EDA Tool B	295	383	94
Ngspice	200	788	

Conclusion

- Many different ways to synthesize equivalent circuits for S-parameters in pole-residue form
- Considerations for choosing circuit topology
  - Want recursive convolution?
  - Want cross platform exchangeability?
  - Controlled source and negative RLC acceptable?
  - SISO or MIMO pole-residue model?