CIRCUIT CAD AND MODELING THROUGH SPACE MAPPING

J.W. Bandler and J.E. Rayas-Sánchez

SOS-99-7-V

March 1999

CIRCUIT CAD AND MODELING THROUGH SPACE MAPPING

J.W. Bandler and J.E. Rayas-Sánchez

Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1

bandler@mcmaster.ca
www.sos.mcmaster.ca

presented at

WORKSHOP ON NOVEL METHODOLOGIES FOR DEVICE MODELING AND CIRCUIT CAD

1999 IEEE MTT-S Int. Microwave Symposium, Anaheim, CA, June 13, 1999

CIRCUIT CAD AND MODELING THROUGH SPACE MAPPING

J.W. Bandler and J.E. Rayas-Sánchez

Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4K1

bandler@mcmaster.ca
www.sos.mcmaster.ca

Abstract

We review the Space Mapping (SM) approach to circuit design and discuss modeling of microwave circuits using Artificial Neural Networks (ANN). We show that SM and ANN methodologies can be combined into a powerful design framework. SM based neuromodels decrease the cost of training, improve generalization ability and reduce the complexity of the ANN topology with respect to the classical neuromodeling approach. We present and illustrate a variety of possible SM based neuromodels, including SMN, FDSMN, FSMN, FMN and FPSM. We contrast SM based neuromodeling with the classical neuromodeling approach as well as with other state-of-the-art neuromodeling techniques. The SM based neuromodeling techniques are illustrated by a microstrip line, a microstrip right angle bend and an HTS filter.

Space Mapping Optimization
(Bandler et al., 1994-)

Aggressive Space Mapping (ASM) has been applied to design examples exploiting the EM simulators

Sonnet's ***em***

Ansoft HFSS

HP HFSS

coarse models exploit coarse grid EM models or circuit-theoretic/analytical models

coarse models, decomposed into subnetworks, can even consist of a mixture of EM based subnetworks and empirical elements connected through circuit theory

new ASM algorithms TRASM *(Bakr et al., 1998)*, HASM *(Bakr et al., 1999)* have been proposed

Space Mapping Based Artificial Neural Network (ANN) Modeling
(Bandler, Ismail, Rayas-Sánchez and Zhang, 1999)

Artificial Neural Networks can model high-dimensional and highly nonlinear problems *(White et al., 1992)*

ANN models are computationally efficient and can be more accurate than empirical models

ANNs are suitable models for microwave circuit optimization and statistical design *(Zaabab, Zhang and Nakhla, 1995, Gupta et al., 1996, Burrascano and Mongiardo, 1998, 1999)*

ANN modeling of microwave circuits based on Space Mapping technology are exploited for the first time *(Bandler et al., 1999)*

this takes advantage of the vast set of empirical models already available

Novel Applications of Space Mapping Technology
(Bandler et al., 1999)

we illustrate several new techniques to generate SM based neuromodels

Space Mapped Neuromodeling (SMN)

Frequency-Dependent Space Mapped Neuromodeling (FDSMN)

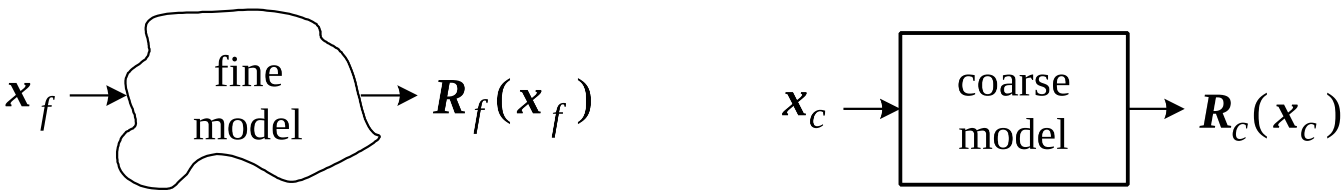
Frequency Space Mapped Neuromodeling (FSMN)

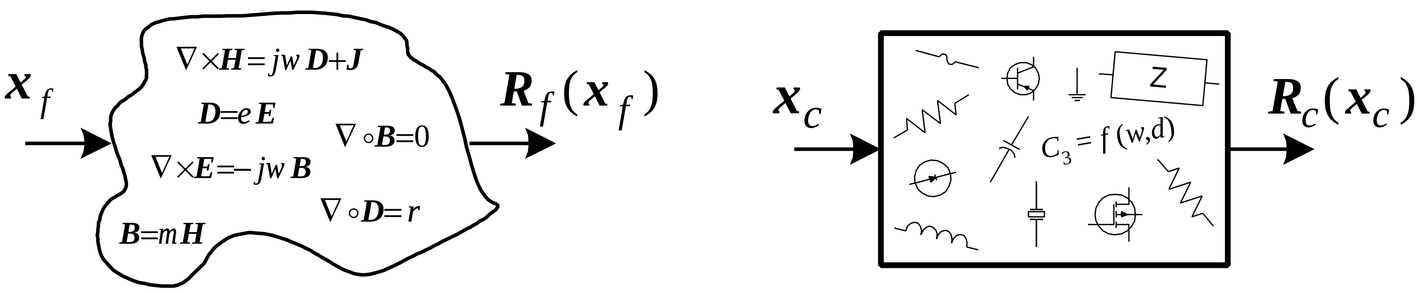
Frequency Mapped Neuromodeling (FMN)

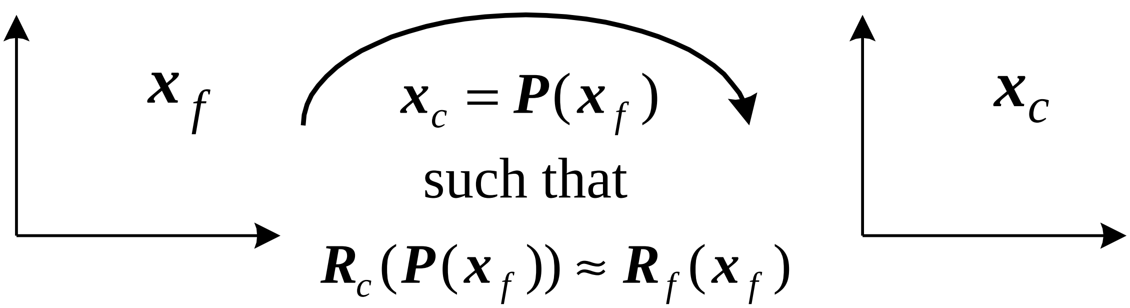
Frequency Partial-Space Mapped Neuromodeling (FPSMN)

these techniques
- exploit the vast set of empirical models already available
- decrease the fine model evaluations needed for training
- improve generalization ability
- reduce complexity of the ANN topology
  - w.r.t. the classical neuromodeling approach

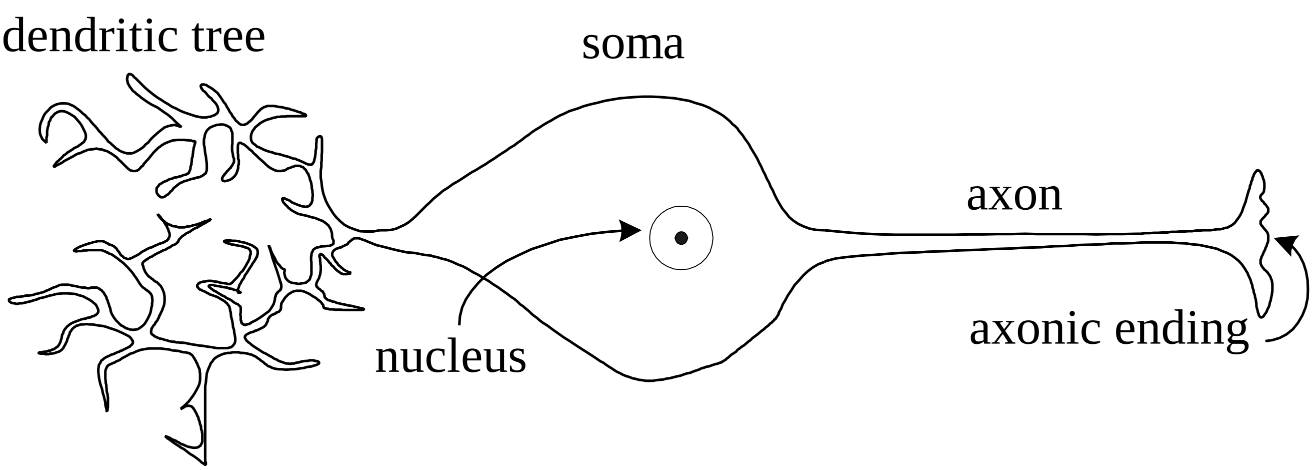
The Aim of Space Mapping



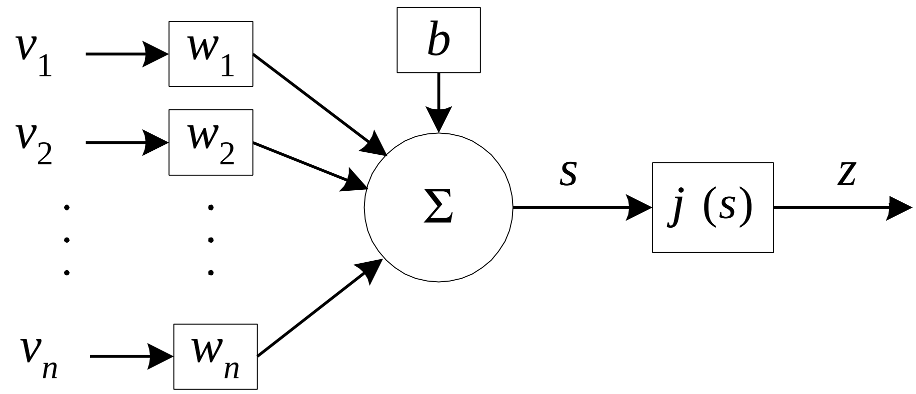




Biological Neuron
(Kartalopoulos, 1996)



Basic Model of a Neuron



$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$ vector of inputs: signals from other neurons

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$ vector of weights: represent corresponding synapse strength

b is the bias or offset term

$s = b + \boldsymbol{v}^T\ \boldsymbol{w}$ is the activation signal

$z = j\ (s)$ is the output signal

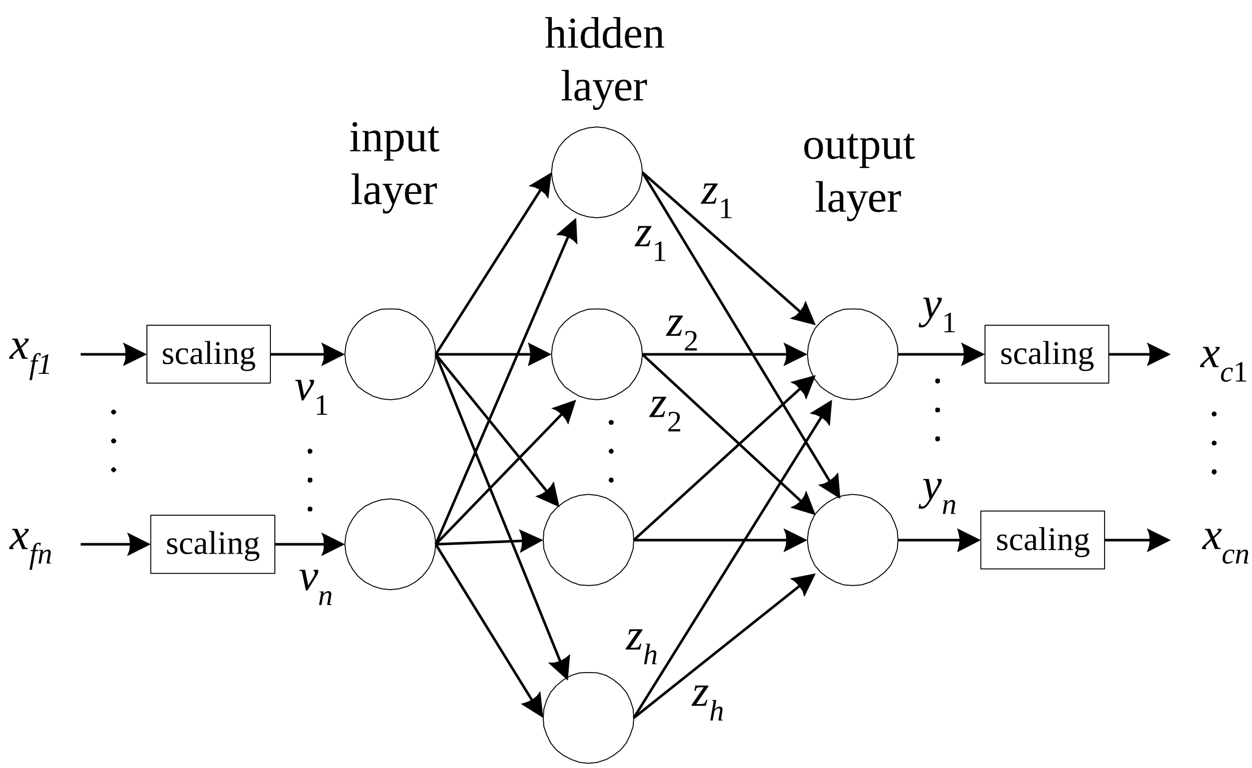
if a sigmoid activation function is used

$$z = j\ (s) = \frac{1}{1 + e^{-s}}$$

Neural Space Mapping

$$\boldsymbol{x}_f \longrightarrow \boxed{\boldsymbol{P}(\boldsymbol{x}_f)} \longrightarrow \boldsymbol{x}_c \qquad\qquad \boldsymbol{x}_f \longrightarrow \boxed{\text{ANN}} \longrightarrow \boldsymbol{x}_c$$

using a three layer perceptron (3LP)



Three Layer Perceptron (3LP)

$\boldsymbol{x}_f = [x_{f1} \quad x_{f2} \quad \cdots \quad x_{fn}]^T$ are n input physical parameters

$\boldsymbol{v} = [v_1 \quad v_2 \quad \cdots \quad v_n]^T$ are input signals after scaling

$\boldsymbol{z} = [z_1 \quad z_2 \quad \cdots \quad z_h]^T$ are signals from the h hidden neurons

$\boldsymbol{y} = [y_1 \quad y_2 \quad \cdots \quad y_n]^T$ are n output signals before scaling

$\boldsymbol{x}_c = [x_{c1} \quad x_{c2} \quad \cdots \quad x_{cn}]^T$ are the neuromapping outputs

to control the relative importance of the input parameters and define a suitable dynamic range, scaling can be used

$$v_i = -1 + \frac{2(x_{fi} - x_{fi\,\min})}{(x_{fi\,\max} - x_{fi\,\min})}, \quad i = 1, 2, \cdots, n$$

Three Layer Perceptron (continued)

the hidden layer signals are calculated by

$$z_i = \varphi\,(b_i^h + \boldsymbol{v}^T \boldsymbol{w}_i^h), \;\; i = 1, 2, \cdots, h$$

$\boldsymbol{w}_i^h$ are the vectors of synaptic weights of the hidden neurons

$$\boldsymbol{w}_i^h = \begin{bmatrix} w_{i1}^h & w_{i2}^h & \cdots & w_{in}^h \end{bmatrix}^T, \;\; i = 1, 2, \cdots, h$$

$\boldsymbol{b}^h$ is the vector of bias elements for the hidden neurons

$$\boldsymbol{b}^h = \begin{bmatrix} b_1^h & b_2^h & \cdots & b_h^h \end{bmatrix}^T$$

the output layer signals are given by

$$y_i = b_i^o + \boldsymbol{z}^T \boldsymbol{w}_i^o, \;\; i = 1, 2, \cdots, n$$

$\boldsymbol{w}_i^o$ are the vectors of synaptic weights of the output neurons

$$\boldsymbol{w}_i^o = \begin{bmatrix} w_{i1}^o & w_{i2}^o & \cdots & w_{ih}^o \end{bmatrix}^T, \;\; i = 1, 2, \cdots, n$$

$\boldsymbol{b}^o$ is the vector of bias elements for the output neurons

$$\boldsymbol{b}^o = \begin{bmatrix} b_1^o & b_2^o & \cdots & b_n^o \end{bmatrix}^T$$

Three Layer Perceptron (continued)

to provide a scaling for the output signals equivalent to the one used in the input

$$x_{ci} = x_{fi\,\mathrm{min}} + \frac{1}{2}(y_i + 1)\,(x_{fi\,\mathrm{max}} - x_{fi\,\mathrm{min}}),\quad i = 1, 2, \cdots, n$$

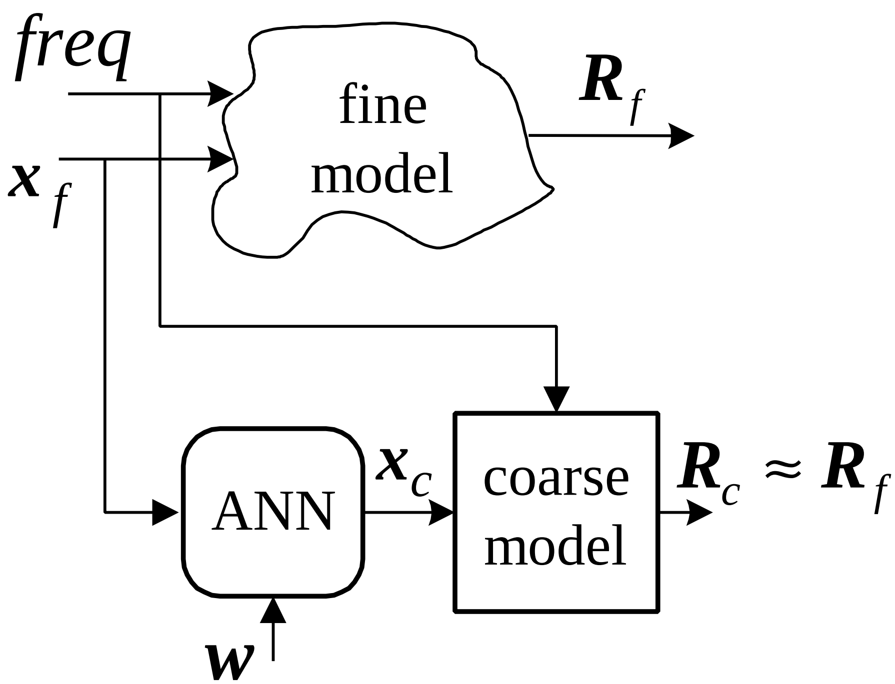
all internal parameters of the ANN can be grouped as

$$\boldsymbol{w} = [(\boldsymbol{b}^h)^T \quad (\boldsymbol{w}_1^h)^T \quad \ldots \quad (\boldsymbol{w}_h^h)^T \quad (\boldsymbol{b}^o)^T \quad (\boldsymbol{w}_1^o)^T \quad \ldots \quad (\boldsymbol{w}_n^o)^T\,]^T$$

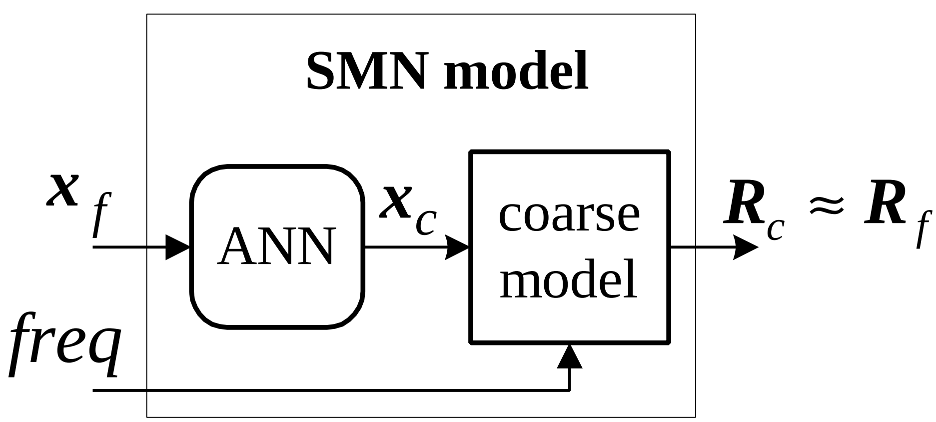
the number of optimization variables for a three-layer perceptron with n inputs, n outputs and h hidden neurons is

$$n(2h+1)+h$$

Space Mapped Neuromodeling (SMN) Concept



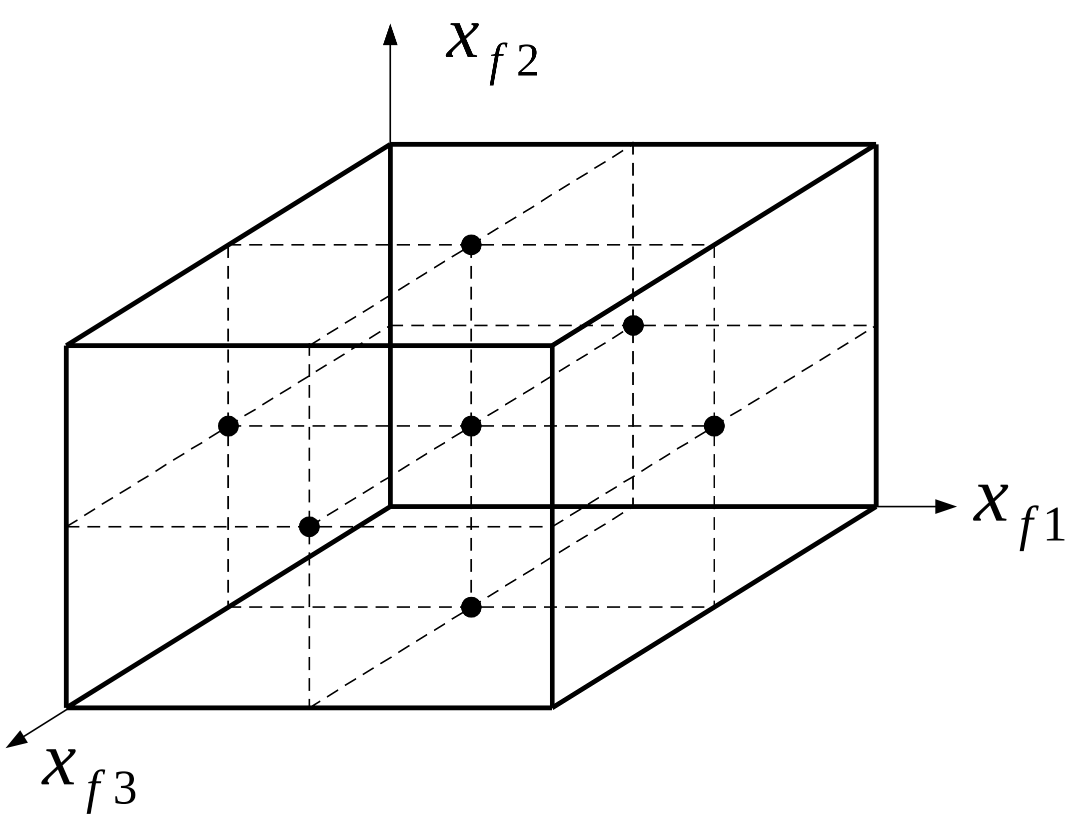
once the ANN is trained



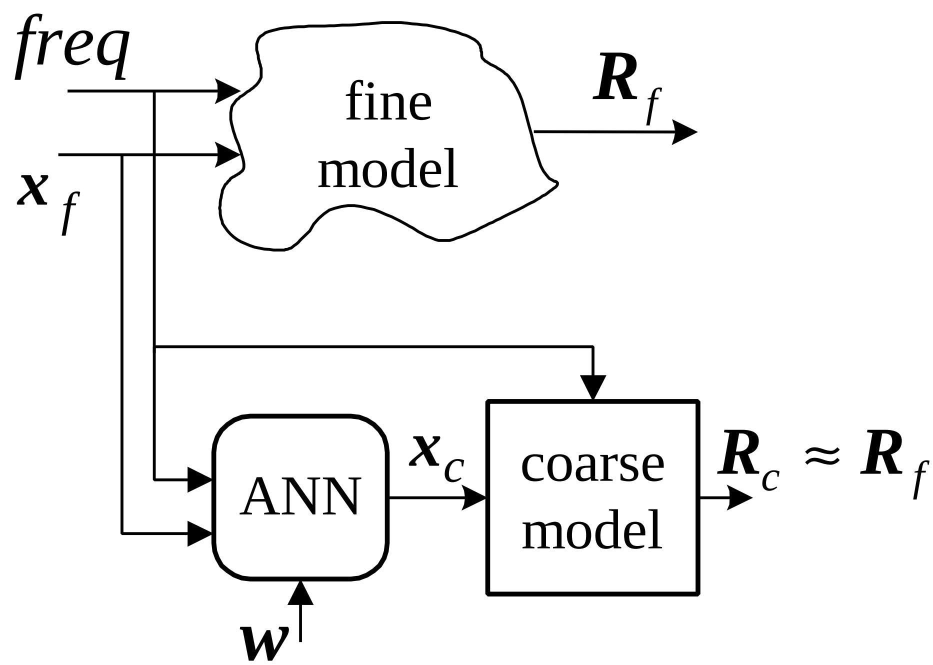
Three-dimensional Star Distribution for the Learning Base Points
(Bandler et al., 1989)

to keep a reduced set of learning data samples, we consider an *n*-dimensional star distribution for the base learning points

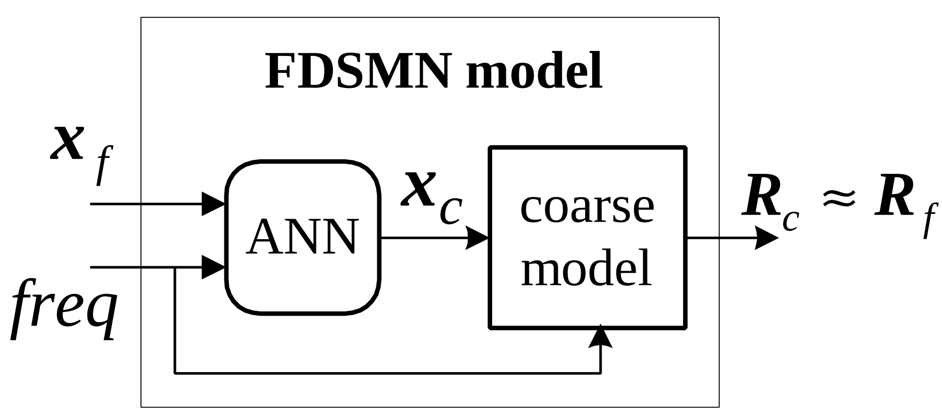
the number of learning base points for a microwave circuit with *n* design parameters is $B_p = 2n + 1$



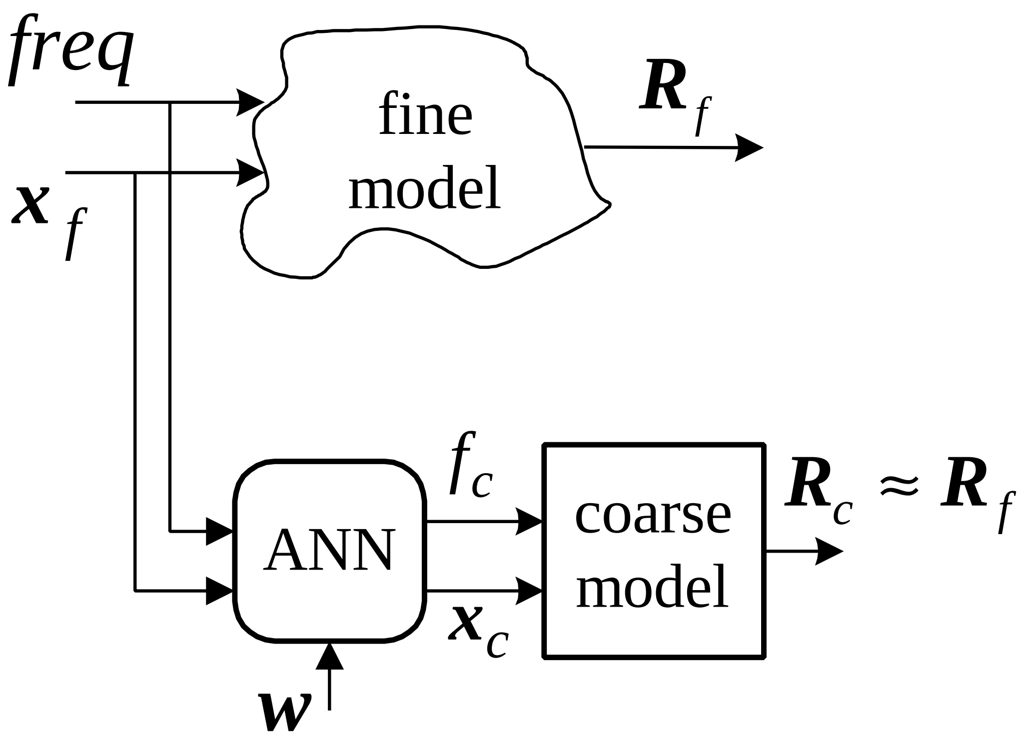
Frequency-Dependent Space Mapped Neuromodeling (FDSMN) Concept



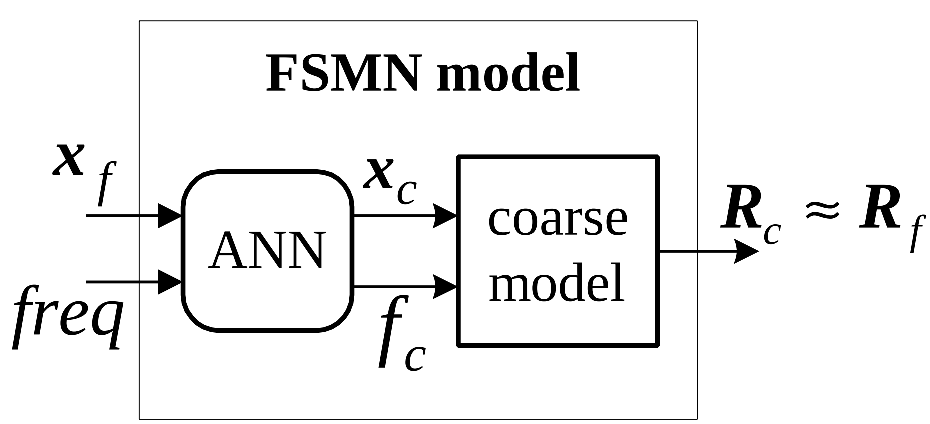
once the ANN is trained



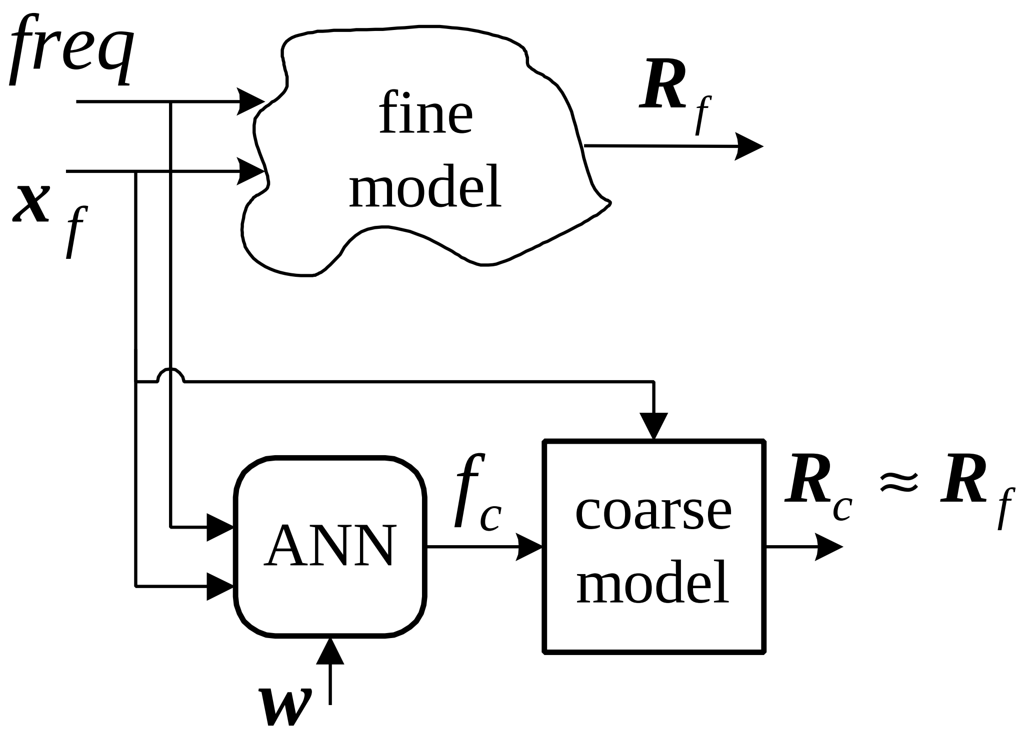
Frequency Space Mapped Neuromodeling (FSMN) Concept



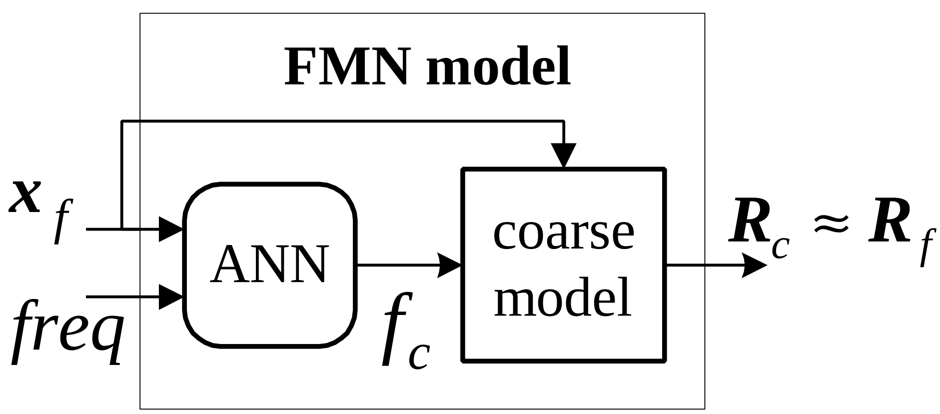
once the ANN is trained



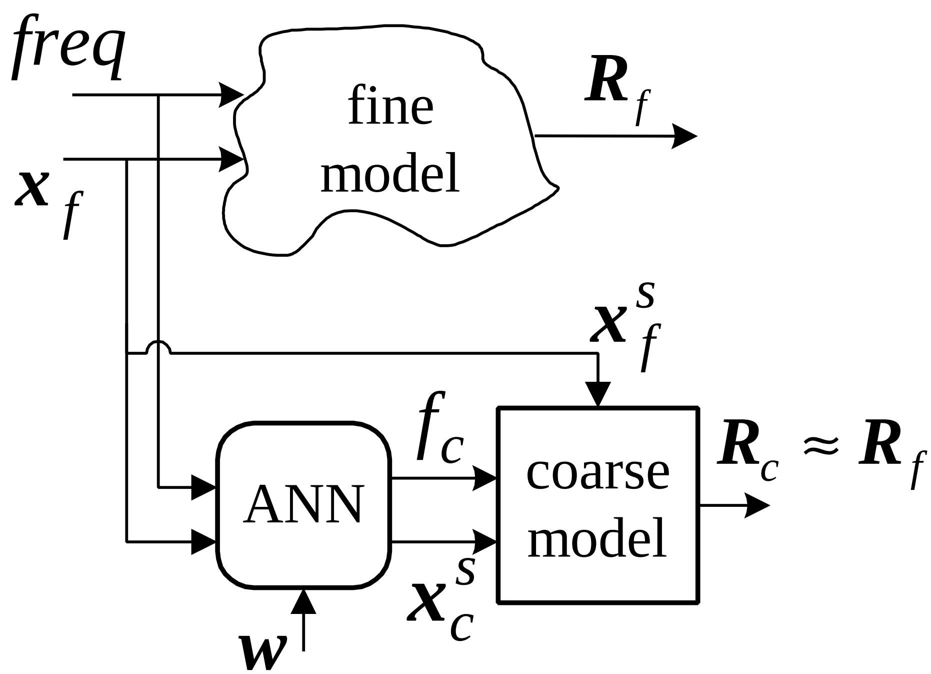
Frequency Mapped Neuromodeling (FMN) Concept



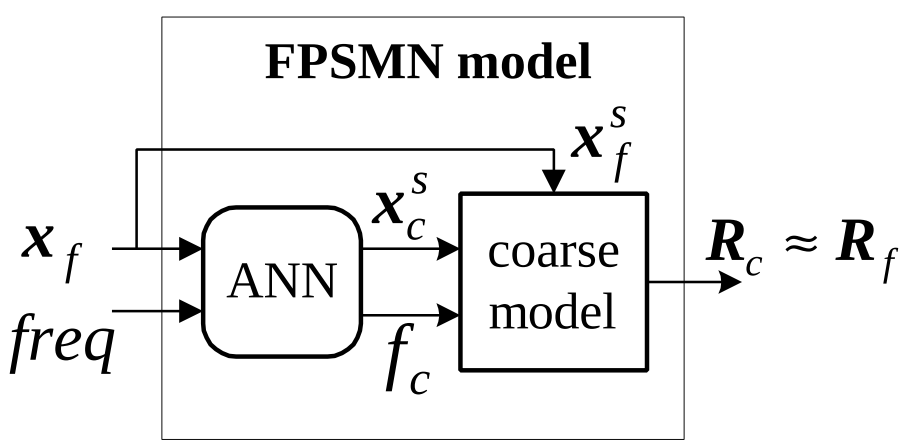
once the ANN is trained



Frequency Partial-Space Mapped Neuromodeling (FPSMN) Concept



once the ANN is trained



Training the ANN

the neuromapping can be found by solving the optimization problem

$$\min_{\boldsymbol{w}} \left\| [\boldsymbol{e}_1^T \quad \boldsymbol{e}_2^T \quad \cdots \quad \boldsymbol{e}_l^T]^T \right\|$$

w contains the internal parameters of the ANN (weights, bias, etc.) selected as optimization variables

l is the total number of learning samples

$\boldsymbol{e}_k$ is the error vector given by

for SMN

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \ freq_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \ freq_j)$$

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_{f_i})$$

for FDSMN

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \ freq_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \ freq_j)$$

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}_{f_i}, \ freq_j)$$

for FSMN

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i}, \ freq_j) - \boldsymbol{R}_c(\boldsymbol{x}_c, \ f_c)$$

Training the ANN (continued)

$$\begin{bmatrix} \boldsymbol{x}_c \\ f_c \end{bmatrix} = \boldsymbol{P}\,(\boldsymbol{x}_{f_i},\ freq_j)$$

for FMN

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i},\ freq_j) - \boldsymbol{R}_c(\boldsymbol{x}_{f_i},\ f_c)$$

$$f_c = P\,(\boldsymbol{x}_{f_i},\ freq_j)$$

for FPSMN

$$\boldsymbol{e}_k = \boldsymbol{R}_f(\boldsymbol{x}_{f_i},\ freq_j) - \boldsymbol{R}_c(\boldsymbol{x}^s_{f_i},\ \boldsymbol{x}^s_c,\ f_c)$$

$$\begin{bmatrix} \boldsymbol{x}^s_c \\ f_c \end{bmatrix} = \boldsymbol{P}\,(\boldsymbol{x}_{f_i},\ freq_j)$$

with

$$i = 1, \ldots, B_p$$

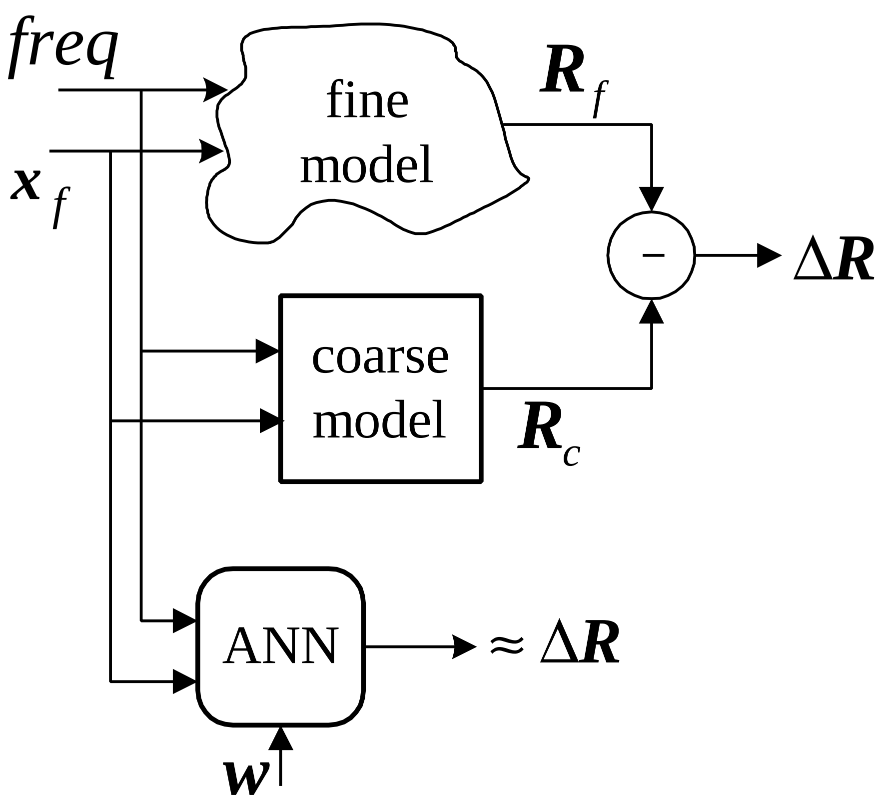
$$j = 1, \ldots, F_p$$

$$k = j + F_p(i-1)$$

EM-ANN Neuromodeling Concept

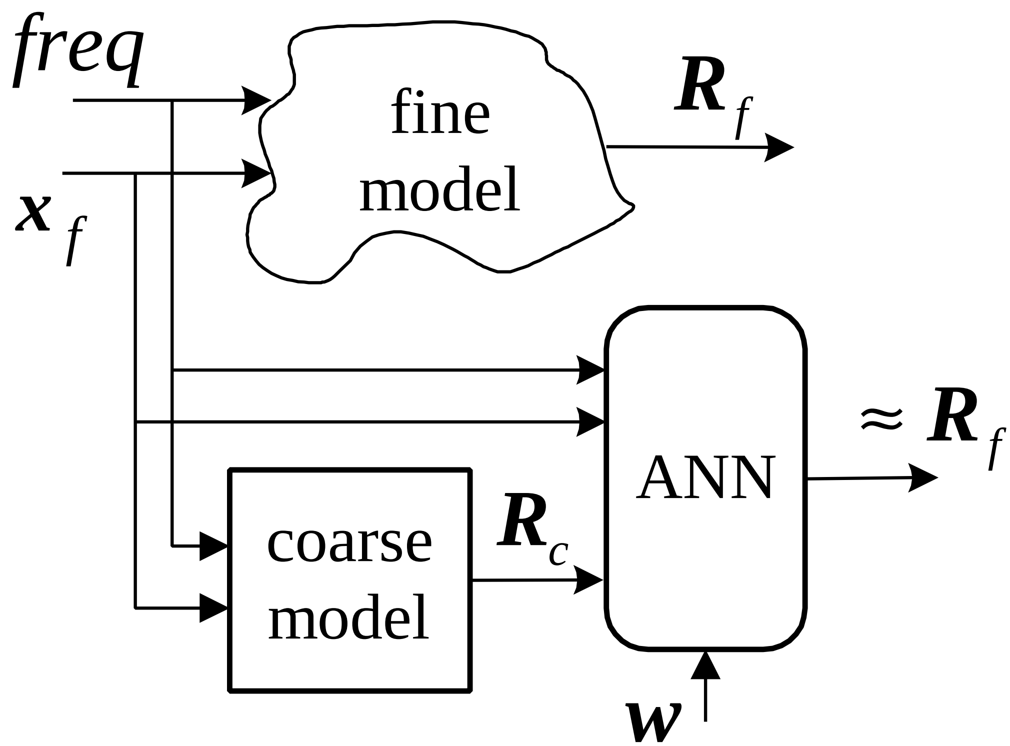
(Gupta et al., 1996)

an interpretation using our notation



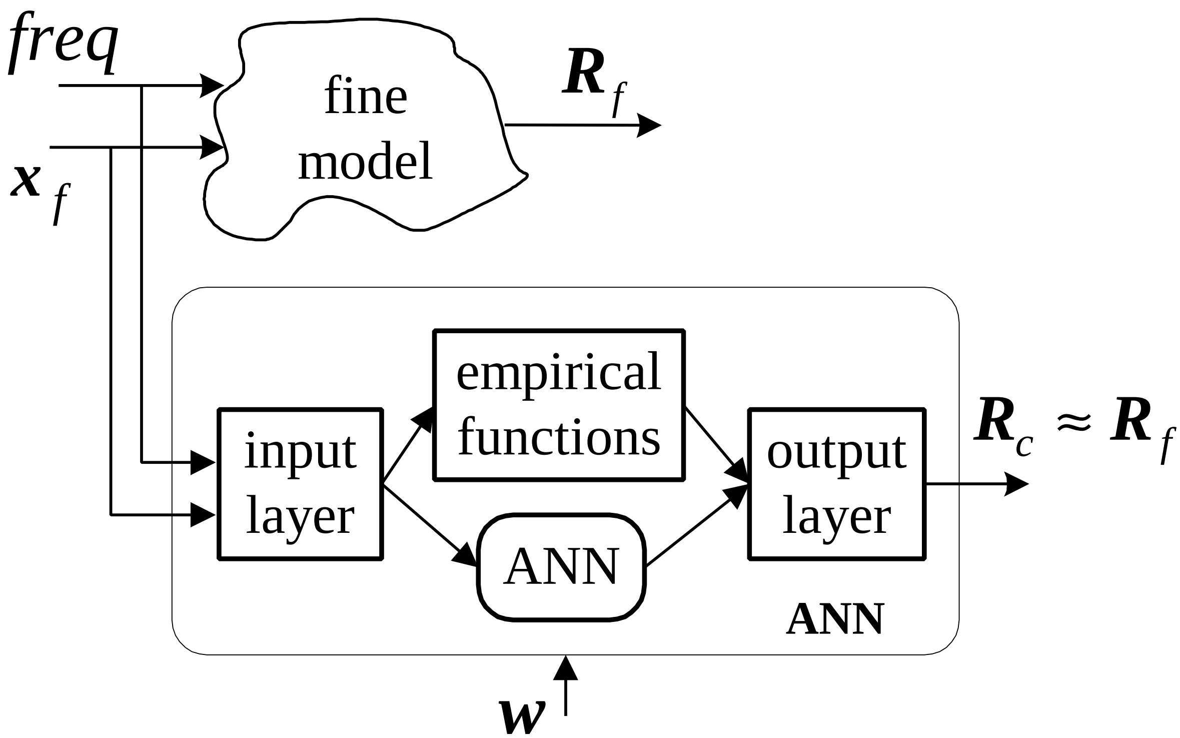
PKI Neuromodeling Concept
(Gupta et al., 1996)

an interpretation using our notation

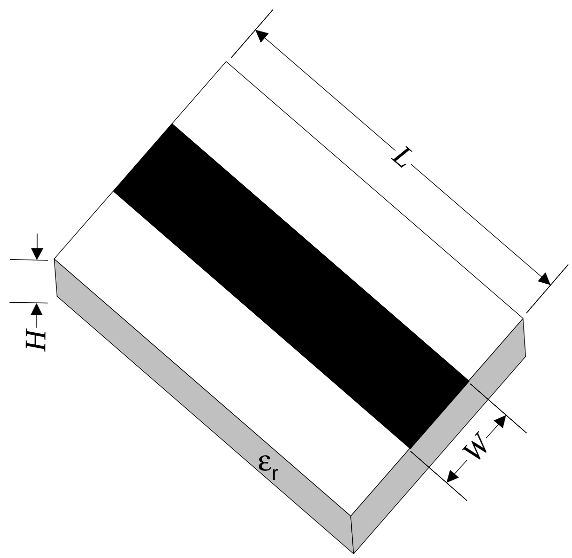


KBNN Neuromodeling Concept
(Zhang et al., 1997)

an interpretation using our notation



Microstrip Line with High Dielectric Constant



region of interest

$$5\text{mil} \le W \le 9\text{mil}$$
$$15\text{mil} \le H \le 25\text{mil}$$
$$40\text{mil} \le L \le 60\text{mil}$$
$$20 \le \boldsymbol{e}_r \le 25$$
$$27\text{GHz} \le freq \le 30\text{GHz}.$$

“coarse” model: Pozar’s formulas *(Pozar, 1998)*

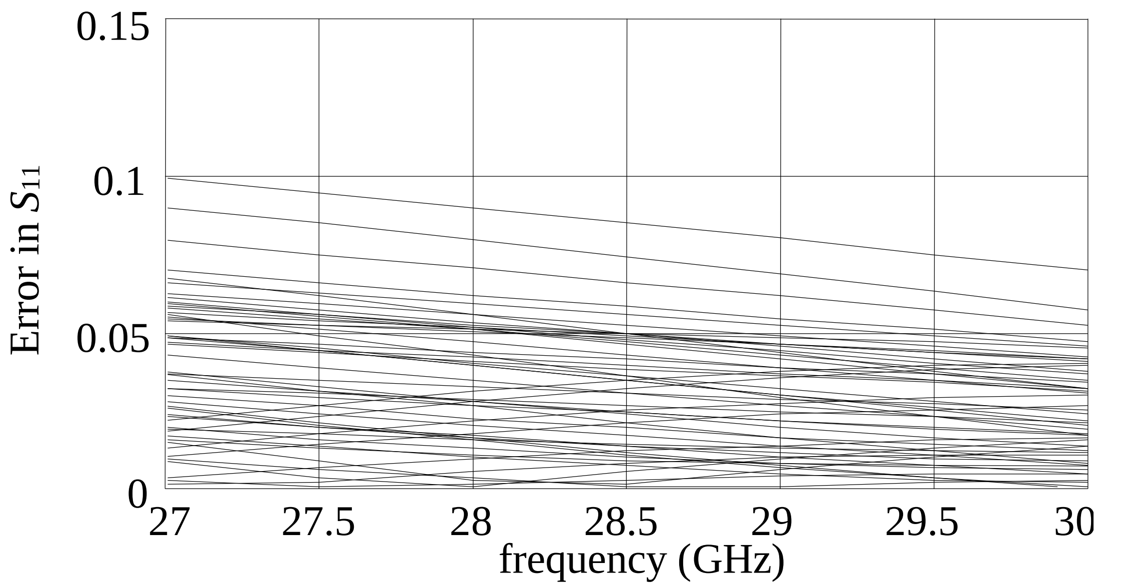
“fine” model: Sonnet’s ***em***™

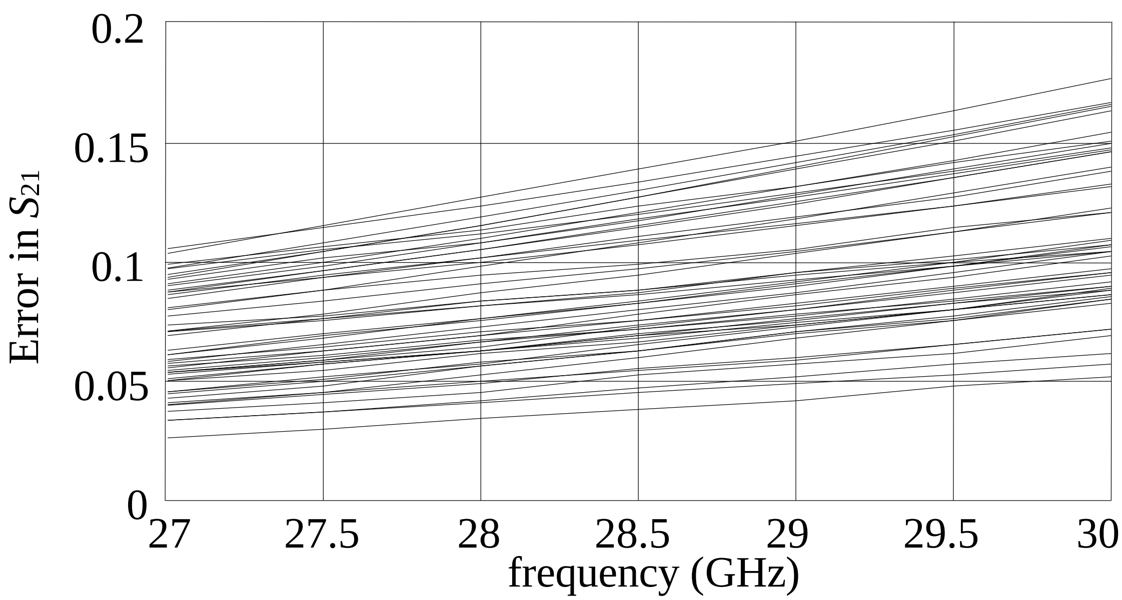
learning set: 9 base points with “star” distribution

testing set: 50 random base points in the region of interest

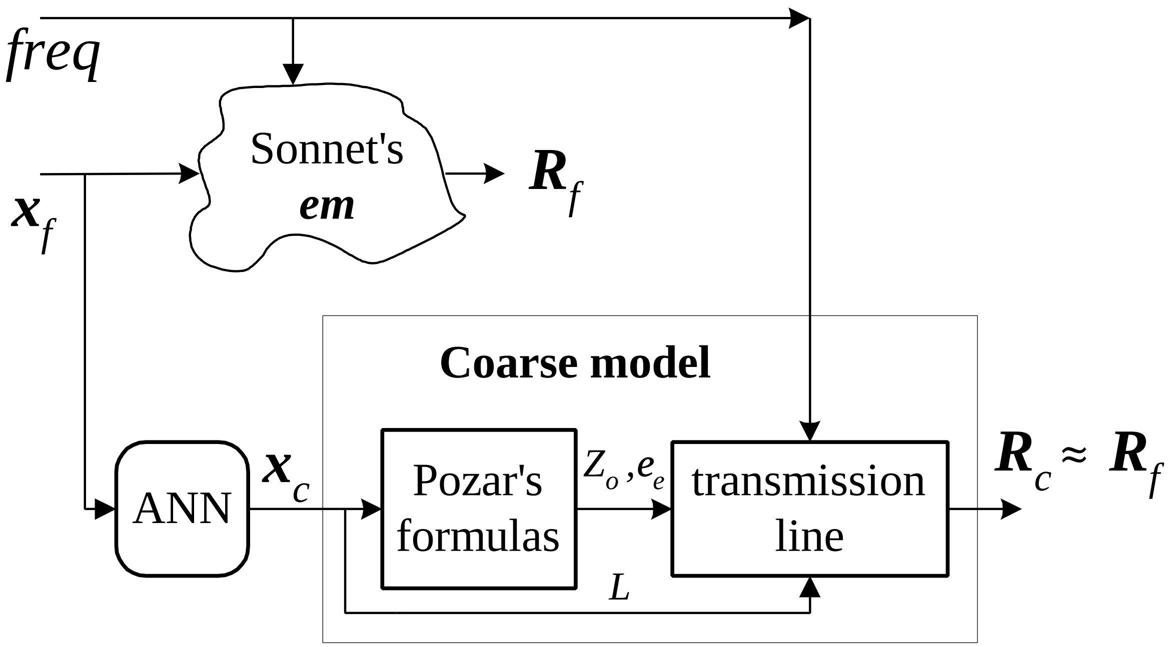
Microstrip Line Response Errors

comparison before neuromodeling between ***em***™ and Pozar's model at 50 random test points



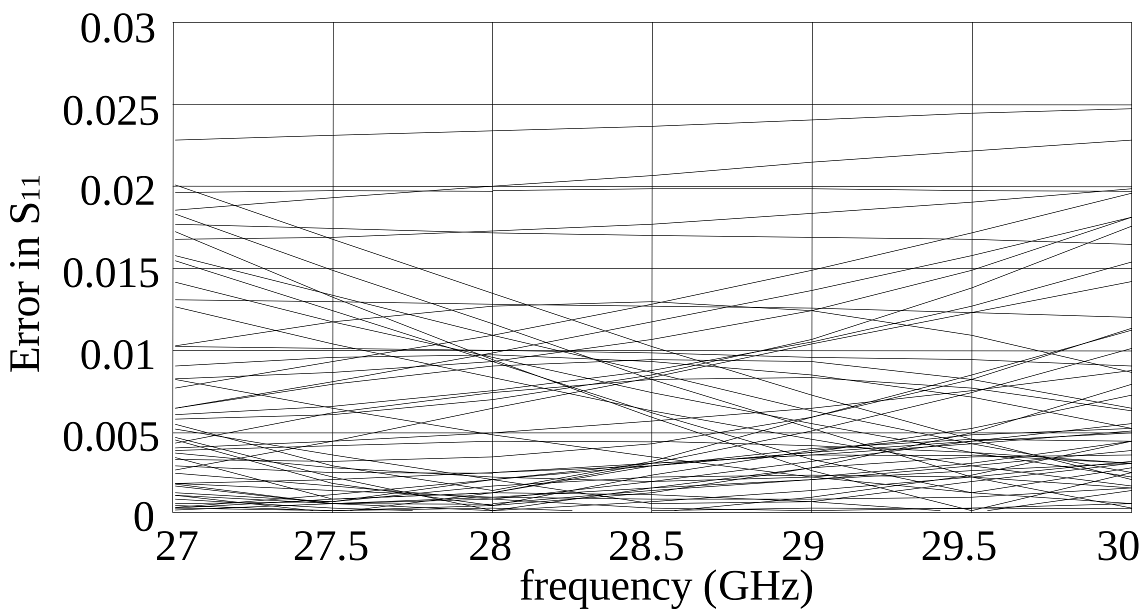


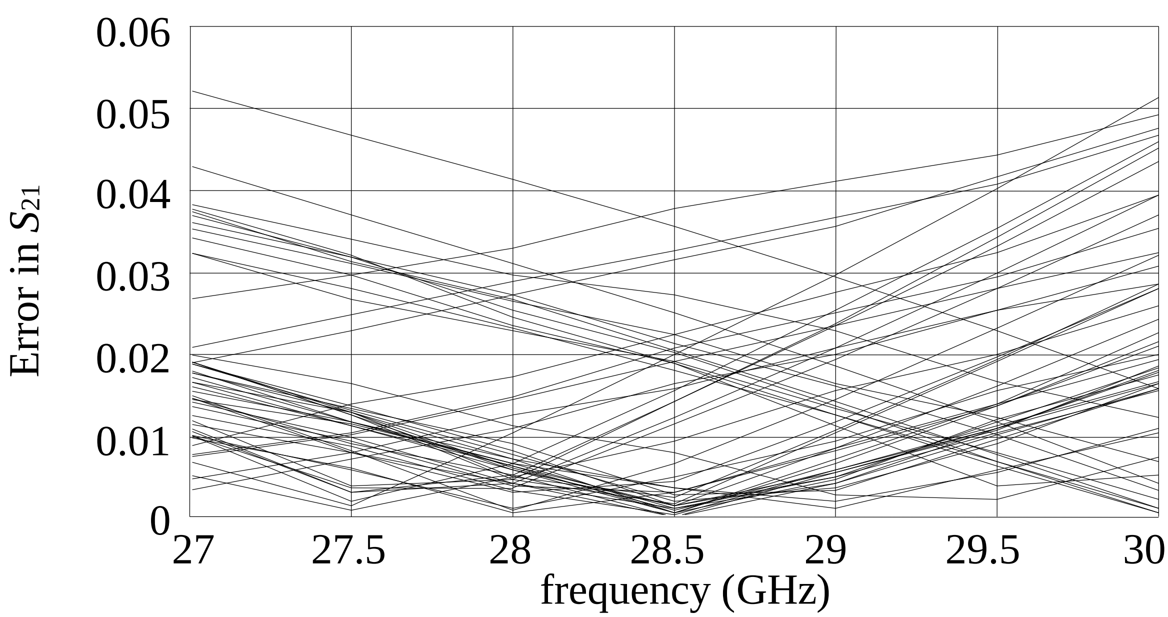
SMN Model for the Microstrip Line (3LP:4-3-4)



SMN Model Results for the Microstrip Line

comparison between ***em***™ and the SMN model





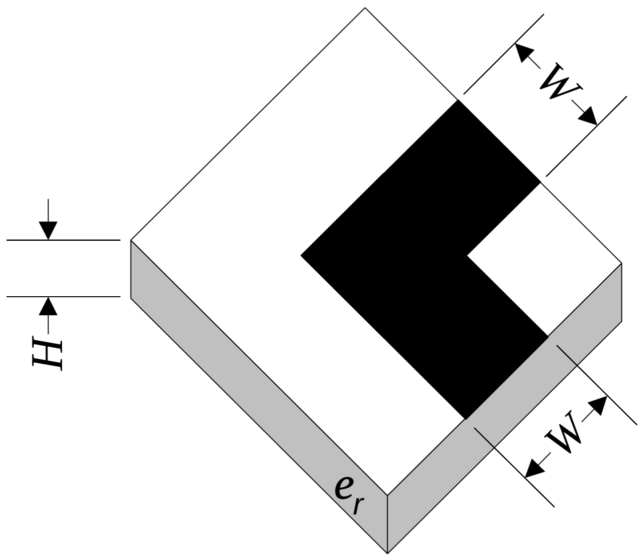
SMN Model for the Microstrip Line Implemented in OSA90

```
Expression
! w : Width of the flat conductor in the PCB (in mils)
! h : Thickness of the PWB laminate (in mils)
! l : Length of the flat conductors (in mils)
! epsr : Dielectric constant of the PWB laminate
! Xf[i]= [w(i) h(i) l(i) epsr(i)]
i: 1;! Index for the training/test points
end

Model

#include "mcsl_hepsr.inc";
! SONNET'S MODEL:
mcsl_hepsr  @f1 @f2 0
l=(Xf[i,3]*1mil) w=(Xf[i,1]*1mil)
h=(Xf[i,2]*1mil) epsr=(Xf[i,4]);
ports @f1 0 @f2 0 ;! Ports 1-2 for Sonnet's model

! Neuromapping  (3LP: 4-3-4)
! ..........................

! input scaling
v1=-1+2*(Xf[i,1]-Xf1_min) / (Xf1_max - Xf1_min);
v2=-1+2*(Xf[i,2]-Xf2_min) / (Xf2_max - Xf2_min);
v3=-1+2*(Xf[i,3]-Xf3_min) / (Xf3_max - Xf3_min);
v4=-1+2*(Xf[i,4]-Xf4_min) / (Xf4_max - Xf4_min);

! vectors of synaptic weights of the hidden neurons : wh
wh1[4]:  [?0.0997064? ?0.00926408? ?-0.0010517? ?0.00555616?];
wh2[4]:  [?-0.0254024? ?0.100381? ?0.00506993? ?-0.0277744?];
wh3[4]:  [?-0.00263021? ?0.00403475? ?0.152244? ?0.0449023?];

! vector of bias elements for the hidden neurons : bh
bh[3]: [?0.0379664? ?-0.0373888? ?0.016498?];

! hidden layer
z1 = tanh(bh[1]+v1*wh1[1]+v2*wh1[2]+v3*wh1[3]+v4*wh1[4]);
z2 = tanh(bh[2]+v1*wh2[1]+v2*wh2[2]+v3*wh2[3]+v4*wh2[4]);
z3 = tanh(bh[3]+v1*wh3[1]+v2*wh3[2]+v3*wh3[3]+v4*wh3[4]);

! vectors of synaptic weights of the output neurons : wo
```

```
wo1[3]: [?9.97323? ?8.37909e-005? ?6.37812e-006?];
wo2[3]: [?-0.000228351? ?10.0461? ?-7.25383e-005?];
wo3[3]: [?0.0112826? ?-0.0014412? ?6.85617?];
wo4[3]: [?0.00258169? ?0.000209437? ?-0.00683818?];
! vector of bias elements for the output neurons : bo
bo[4]: [?0.00394671? ?-0.00393648? ?0.0121073? ?-0.01383?];
! output layer
y1 = bo[1]+z1*wo1[1]+z2*wo1[2]+z3*wo1[3];
y2 = bo[2]+z1*wo2[1]+z2*wo2[2]+z3*wo2[3];
y3 = bo[3]+z1*wo3[1]+z2*wo3[2]+z3*wo3[3];
y4 = bo[4]+z1*wo4[1]+z2*wo4[2]+z3*wo4[3];

! output scaling
Xc1 = Xf1_min + 0.5*(y1+1)*(Xf1_max - Xf1_min);
Xc2 = Xf2_min + 0.5*(y2+1)*(Xf2_max - Xf2_min);
Xc3 = Xf3_min + 0.5*(y3+1)*(Xf3_max - Xf3_min);
Xc4 = Xf4_min + 0.5*(y4+1)*(Xf4_max - Xf4_min);

! POZAR'S MODEL (TRANSMISSION LINE)

epse=(Xc4+1)/2+(Xc4-1)/(2 *sqrt(1+12*Xc2/Xc1));
Zo= if((Xc1/Xc2)<1)
  (60/sqrt(epse) * log(8*Xc2/Xc1+Xc1/(4*Xc2)))
    else
  (120*pi/(sqrt(epse)*(Xc1/Xc2+1.393+0.667*log(Xc1/Xc2+1.444))));
TRL @c1  @c2  Z=Zo L=(Xc3*1mil) K=epse F=FREQ;
ports @c1 0 @c2 0 ;! Ports 3-4 for Pozar's model
CIRCUIT;
end

Sweep
AC: i: from 1 to N step 1
    FREQ: from Freq_min to Freq_max step=Freq_step
"rS11 (Sonnet)","iS11 (Sonnet)","rS21 (Sonnet)","iS21 (Sonnet)"
"rS11 (SMN)","iS11 (SMN)","rS21 (SMN)","iS21 (SMN)"
end

Specification
   AC:  i: from 1 to NL step 1
          FREQ: from Freq_min to Freq_max step=Freq_step
                  "rS11 (SMN)" = "rS11 (Sonnet)"
                  "iS11 (SMN)" = "iS11 (Sonnet)"
                  "rS21 (SMN)" = "rS21 (Sonnet)"
                  "iS21 (SMN)" = "iS21 (Sonnet)"
```

```
end

Control
Perturbation_Scale=1.0e-4;
Disable_Adjoint;
Allow_Neg_Parameters;
Optimizer=Huber;
N_iterations=100;
Display_N_digits=6;
Accuracy=1.0e-5;
Huber_threshold=0.15;
end
```

Microstrip Right Angle Bend



region of interest

$$20\text{mil} \le W \le 30\text{mil}$$
$$8\text{mil} \le H \le 16\text{mil}$$
$$8 \le e_r \le 10$$
$$1\text{GHz} \le freq \le 41\text{GHz}$$

"coarse" model: Gupta model

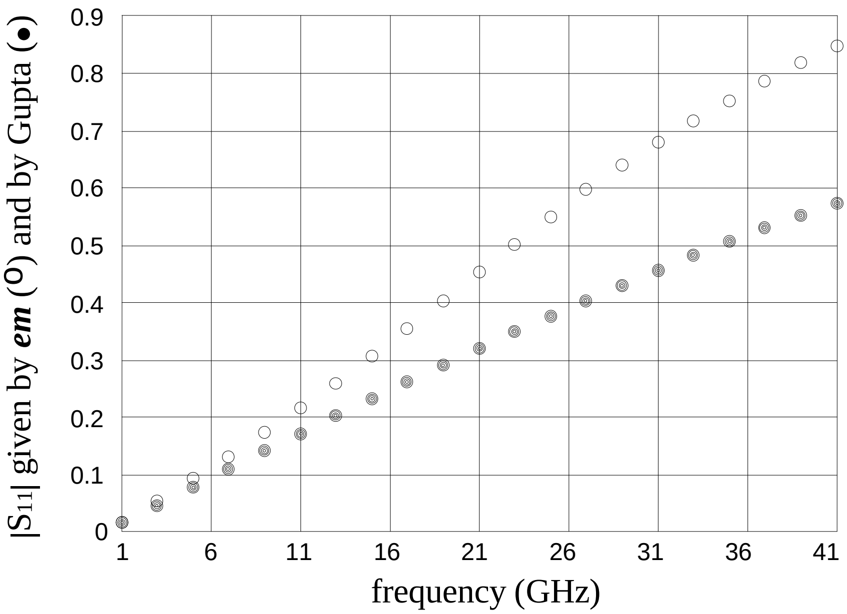
"fine" model: Sonnet's ***em***™

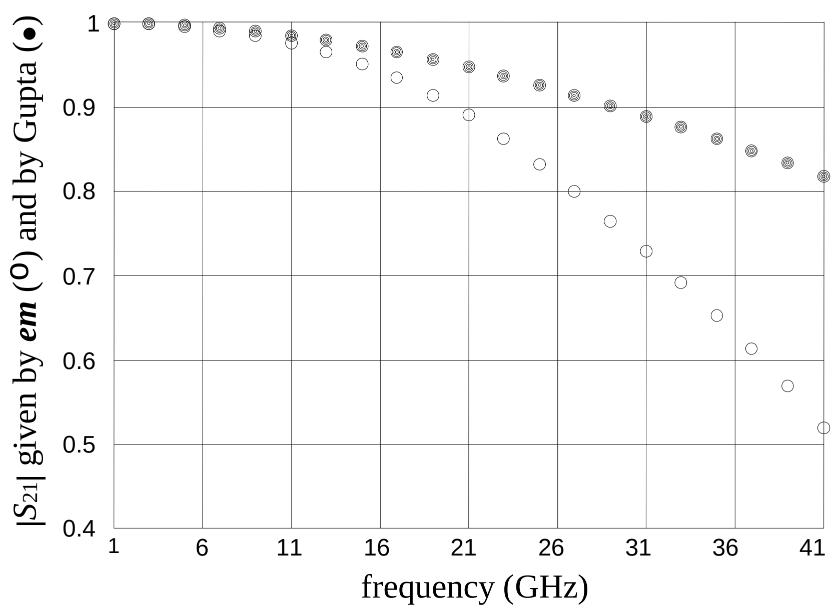
learning set: 7 base points with "star" distribution

testing set: 50 random base points in the region of interest

Microstrip Right Angle Bend Responses

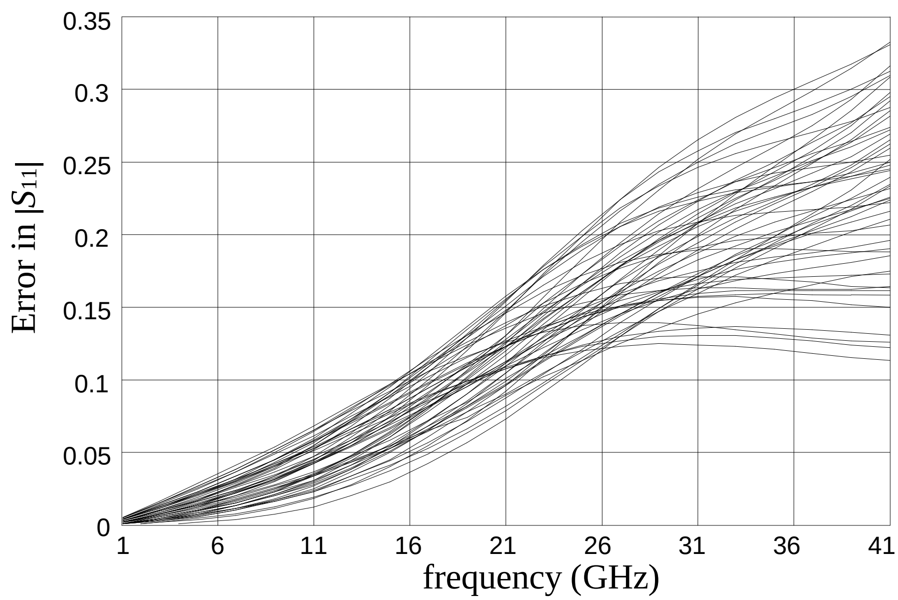
typical responses before neuromodeling
em™ (o), Gupta model (•)

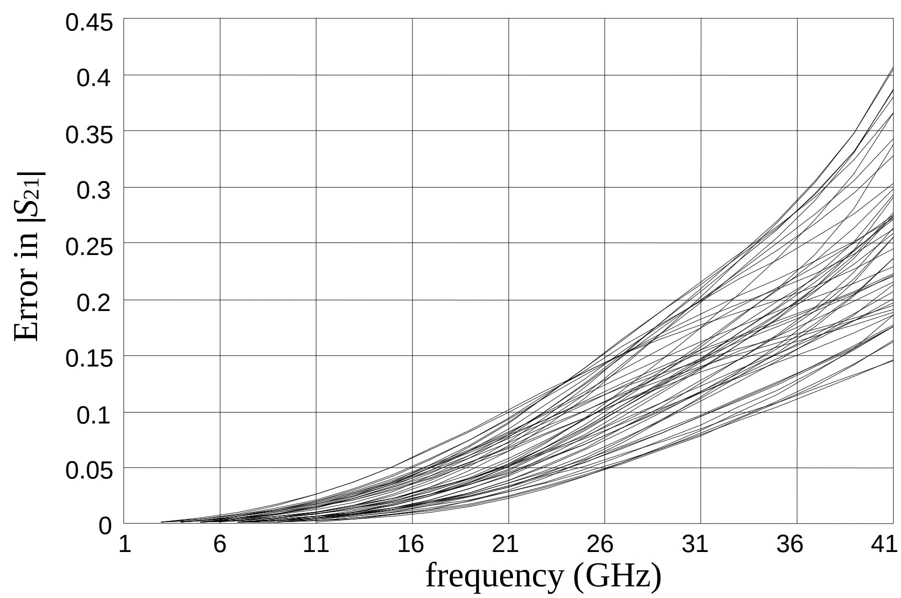




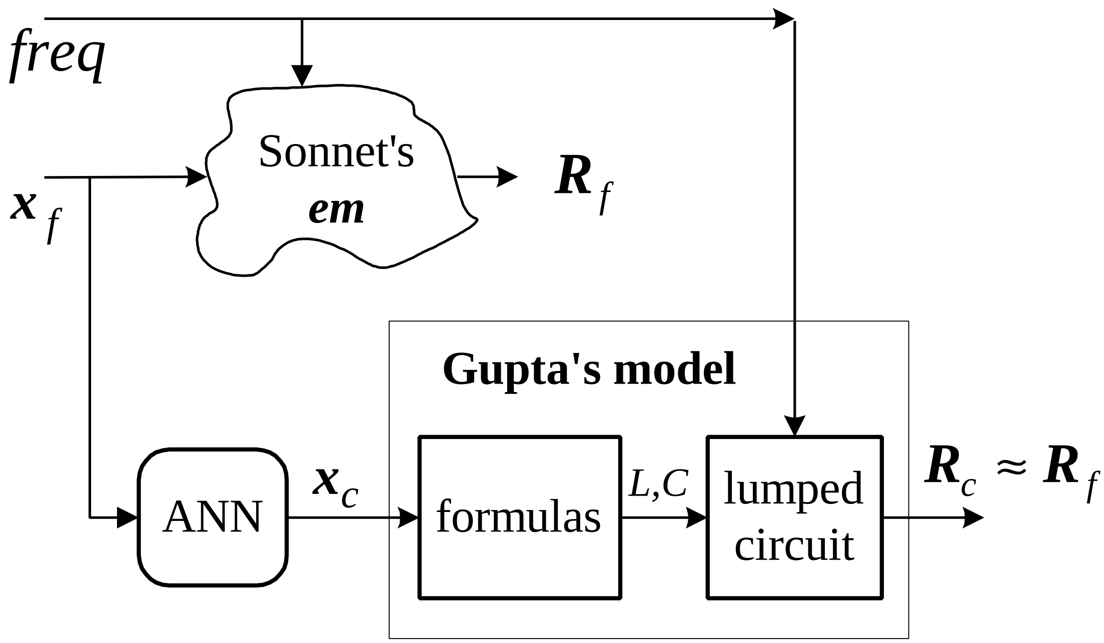
Microstrip Right Angle Bend Response Errors

comparison before neuromodeling between ***em***™ and Gupta model at 50 random test points



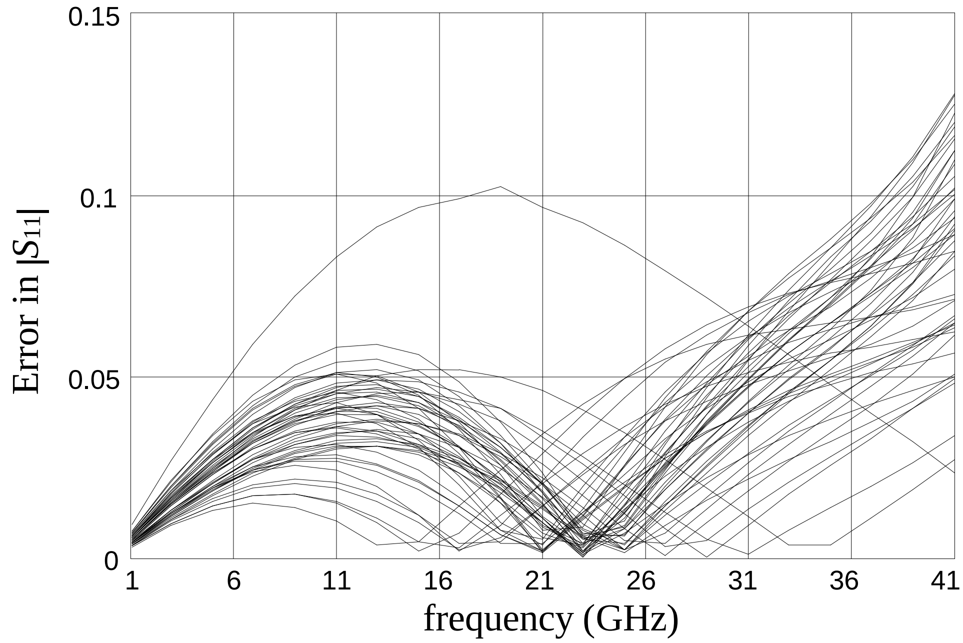


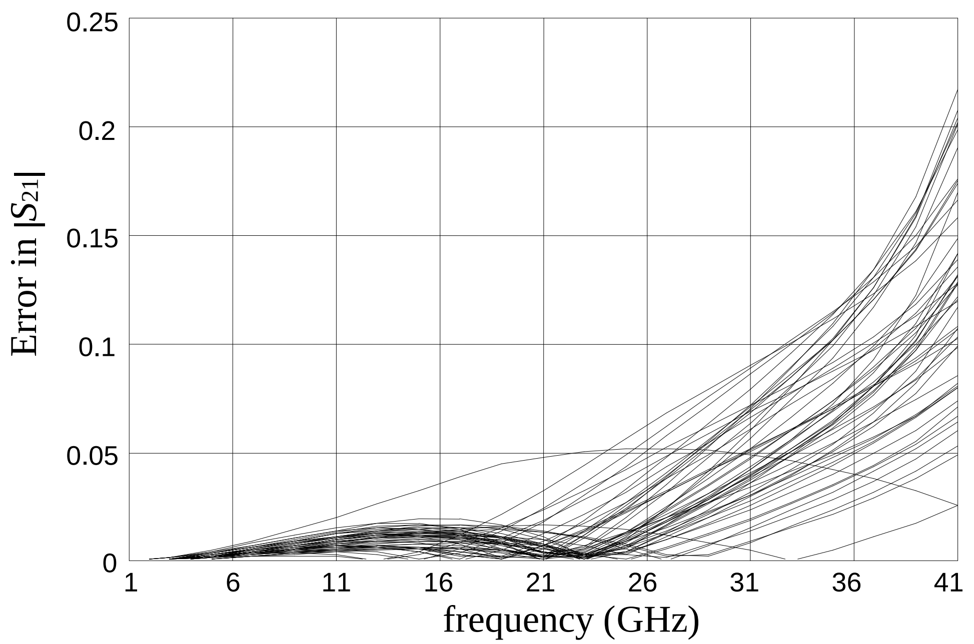
SMN Model for the Right Angle Bend (3LP:3-6-3)



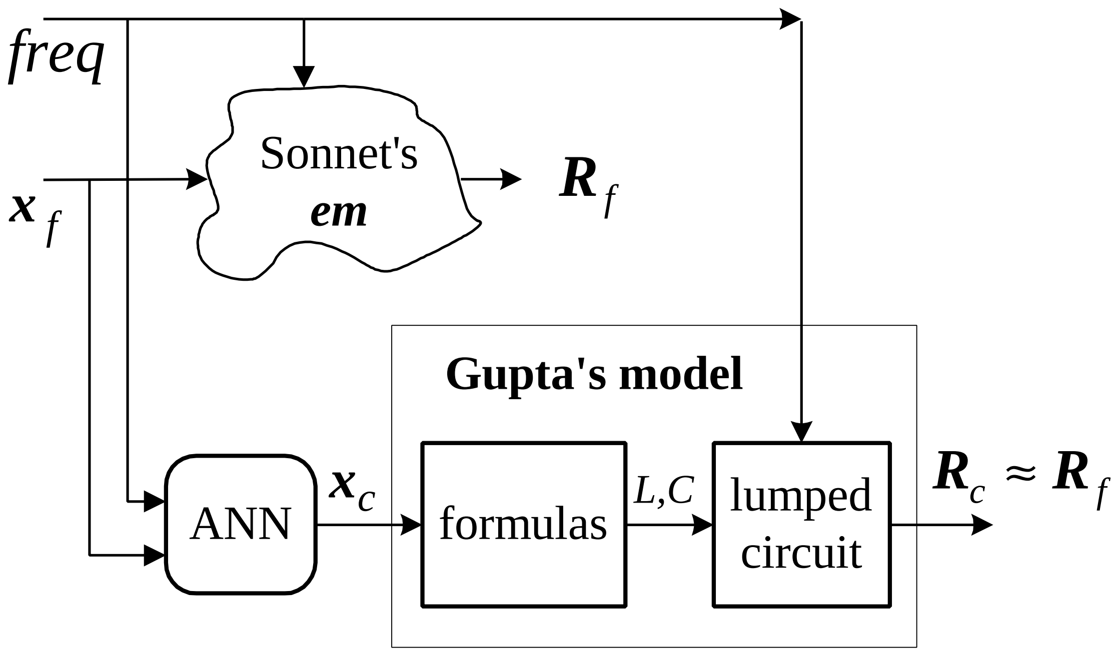
SMN Model Results for the Right Angle Bend

comparison between ***em***™ and the SMN model



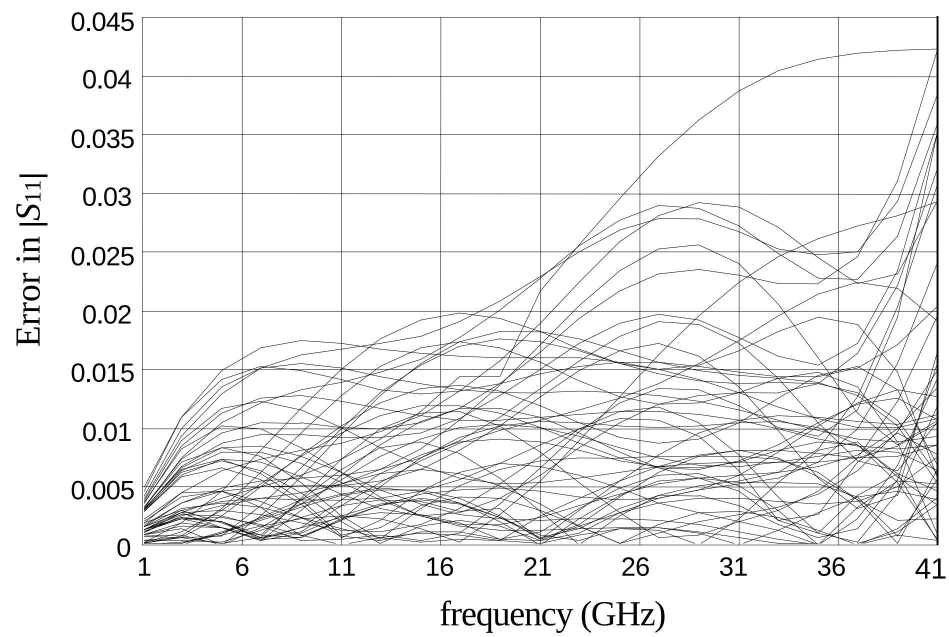


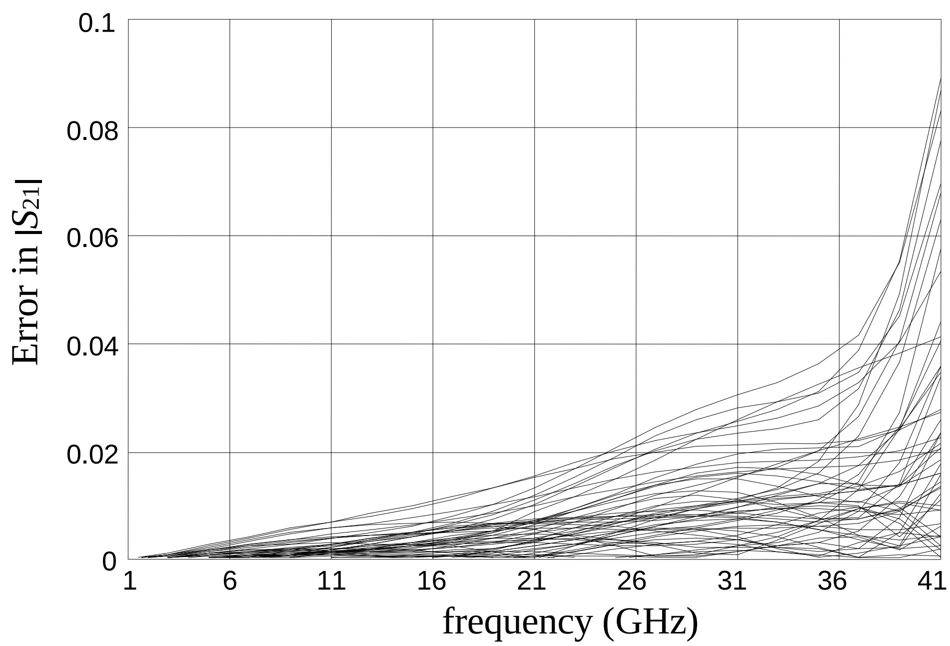
FDSMN Model for the Right Angle Bend (3LP:4-7-3)



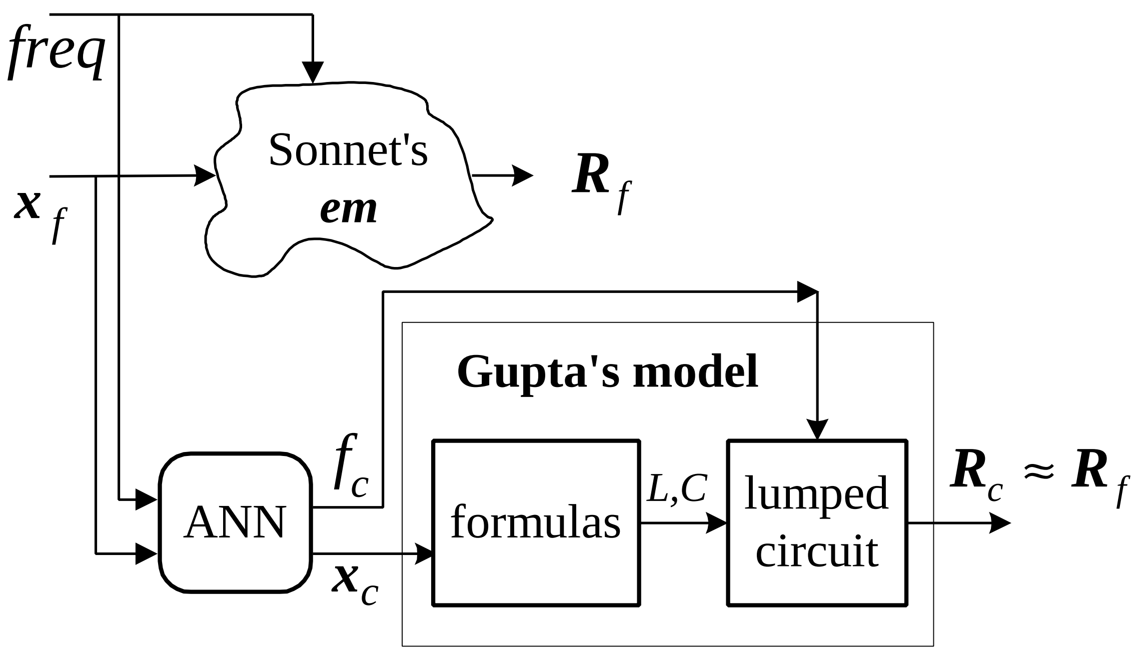
FDSMN Model Results for the Right Angle Bend

comparison between ***em***™ and the FDSMN model





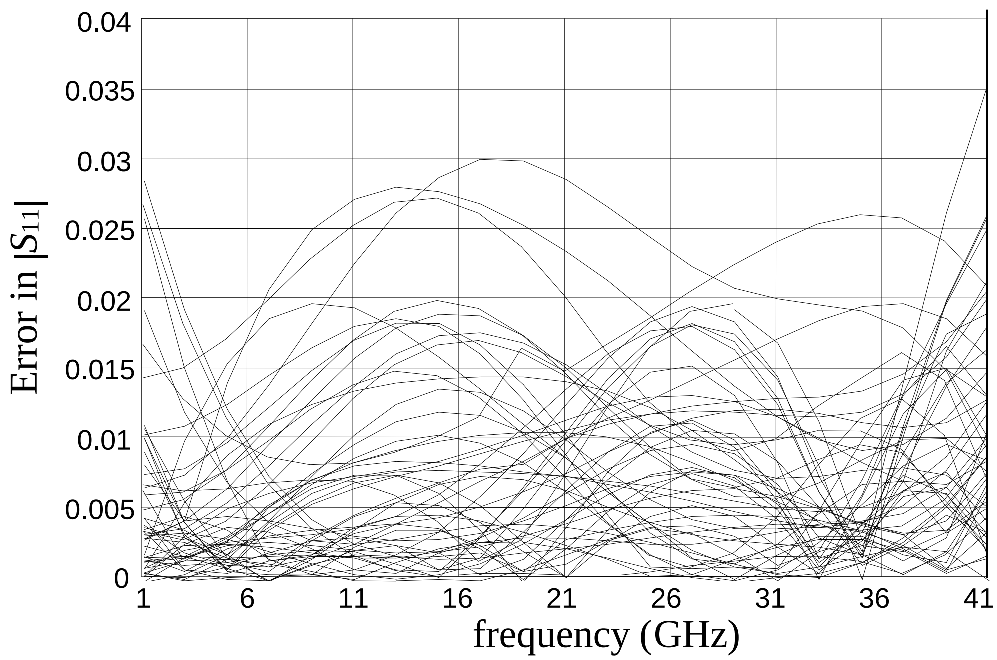
FSMN Model for the Right Angle Bend (3LP:4-8-4)

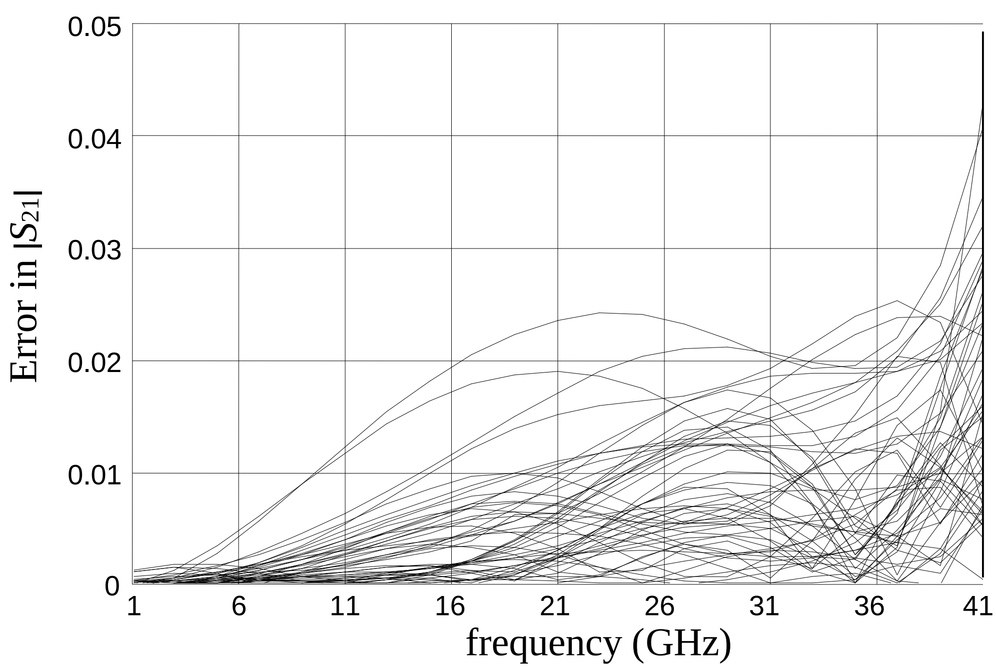


to implement the FSMN approach, an OSA90/hope™ child program is employed to simulate the coarse model with a different frequency variable using Datapipe

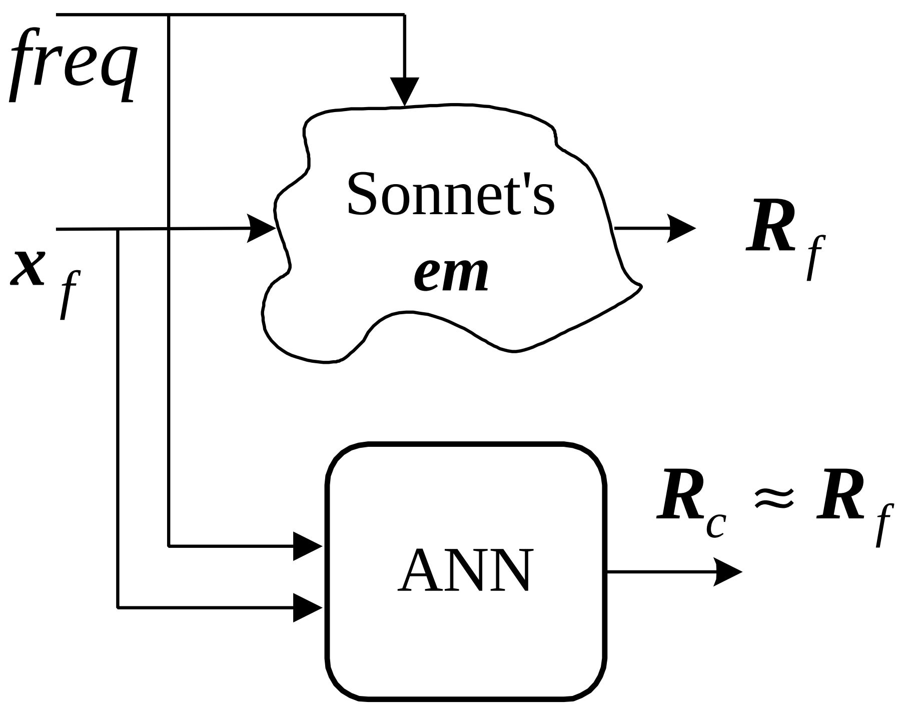
FSMN Model Results for the Right Angle Bend

comparison between ***em***™ and the FSMN model





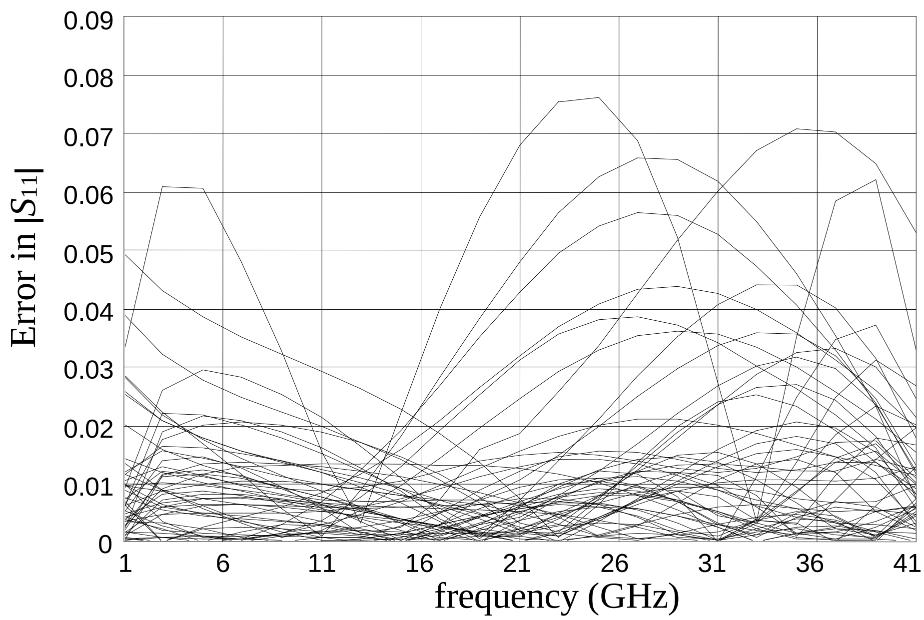
Classical Neuromodel for the Right Angle Bend (3LP:4-15-4)

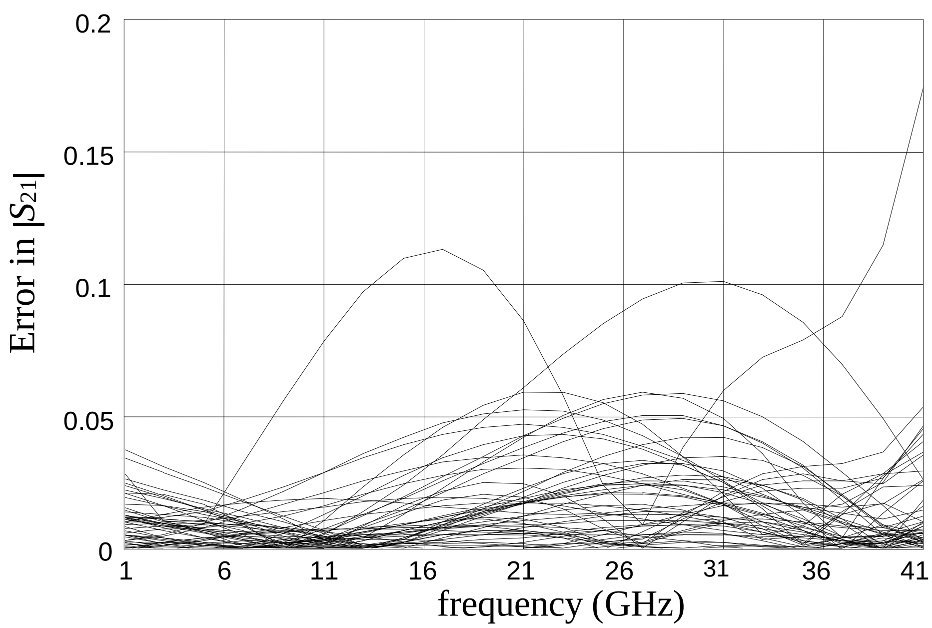


Classical Neuromodel Results for the Right Angle Bend

(Neuromodeler, 1998)

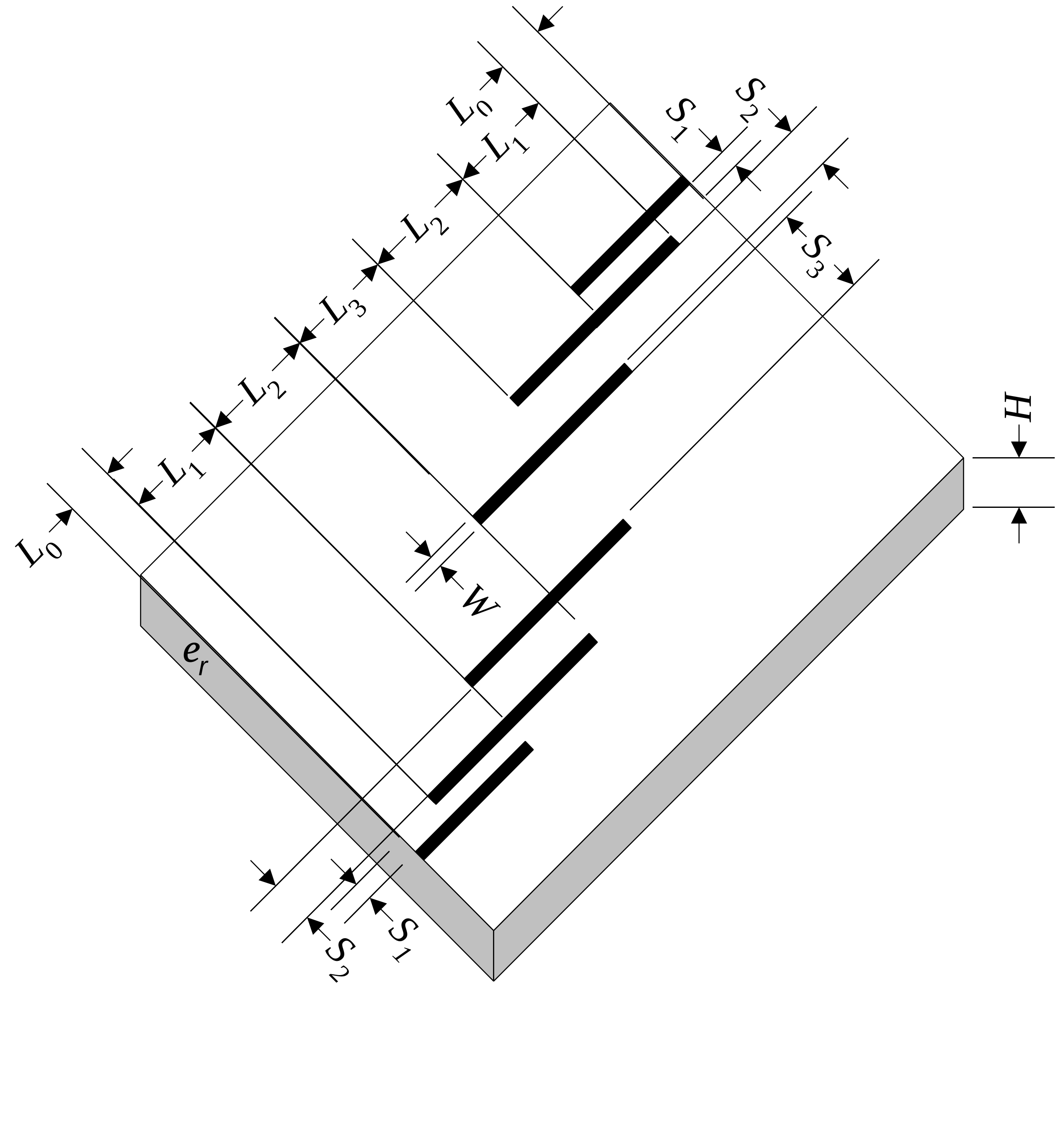
comparison between ***em***™ and classical neuromodel





HTS Quarter-Wave Parallel Coupled-Line Microstrip Filter

(Westinghouse, 1993)



SM Based Neuromodeling of the HTS Filter

region of interest

$$175\text{mil} \le L_1 \le 185\text{mil}$$
$$190\text{mil} \le L_2 \le 210\text{mil}$$
$$175\text{mil} \le L_3 \le 185\text{mil}$$
$$18\text{mil} \le S_1 \le 22\text{mil}$$
$$75\text{mil} \le S_2 \le 85\text{mil}$$
$$70\text{mil} \le S_3 \le 90\text{mil}$$
$$3.901\text{GHz} \le freq \le 4.161\text{GHz}$$

$$L_0 = 50\text{mil}$$
$$H = 20\text{mil}$$
$$W = 7\text{mil}$$
$$\boldsymbol{e}_r = 23.425$$
$$\text{loss tangent} = 3\times10^{-5}$$

“coarse” model: OSA90/hope™ empirical models

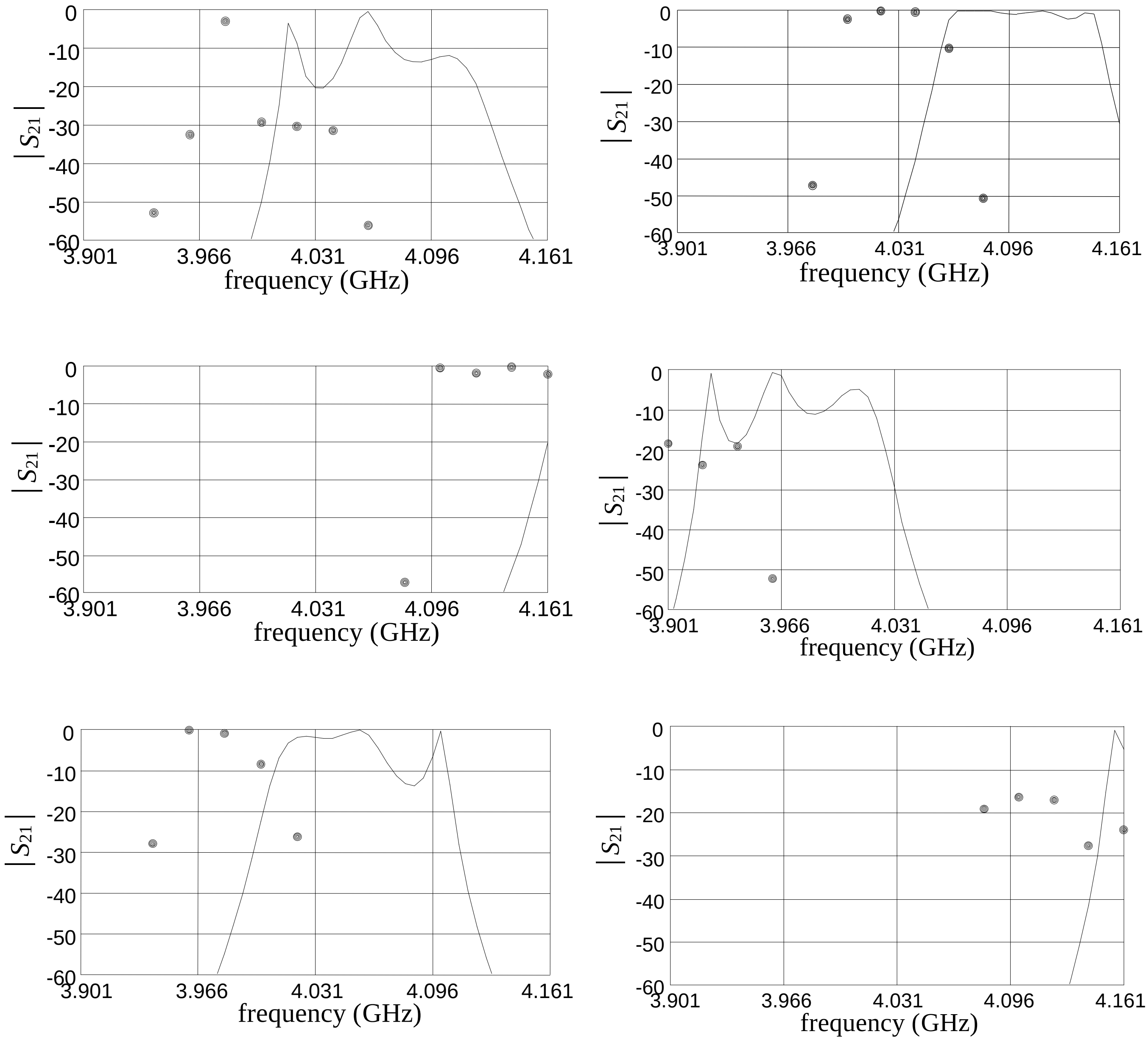
“fine” model: Sonnet’s ***em***™ with high resolution grid

learning set: 13 base points with “star” distribution

testing set: 7 random base points in the region of interest (not seen in the learning set)

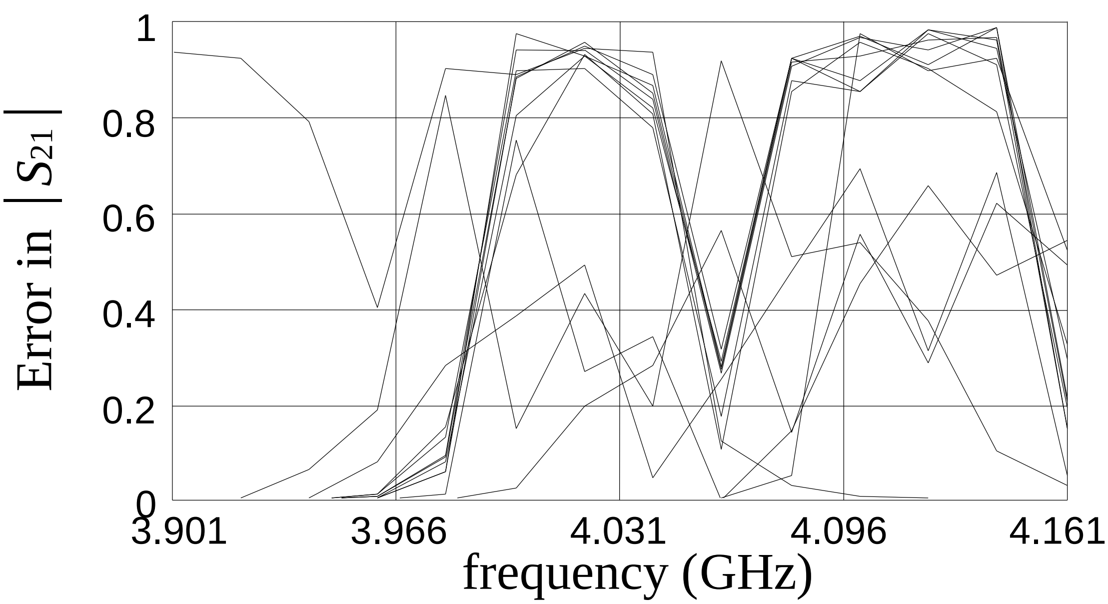
HTS Filter Responses Before Neuromodeling

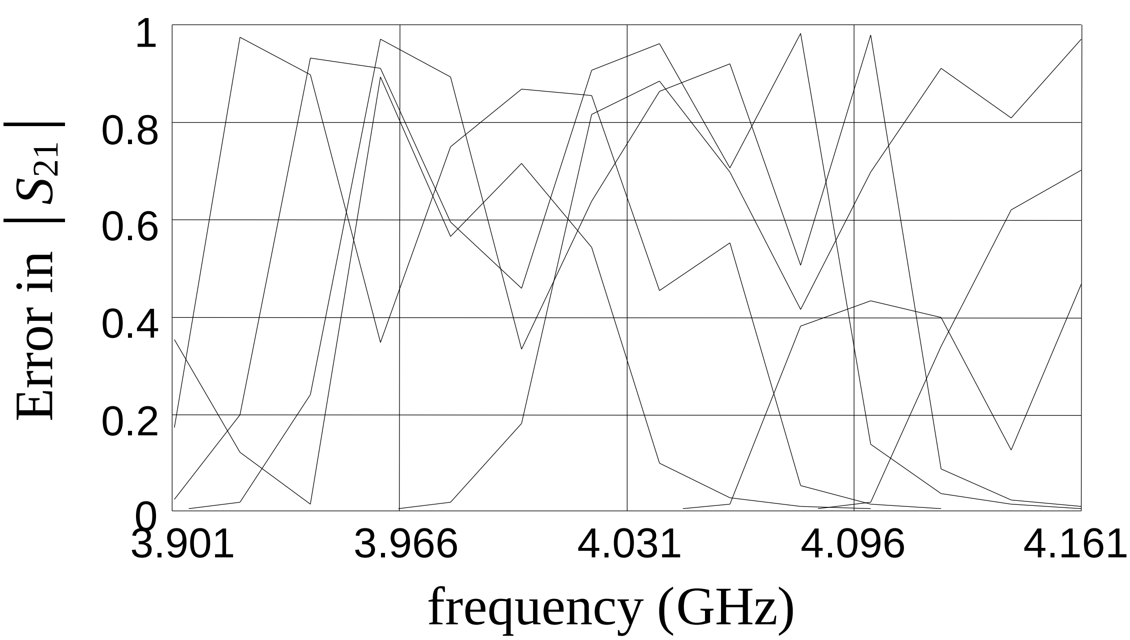
responses using ***em***™ (•) and OSA90/hope™ (–) at three learning and three test points



HTS Filter Response Errors Before Neuromodeling

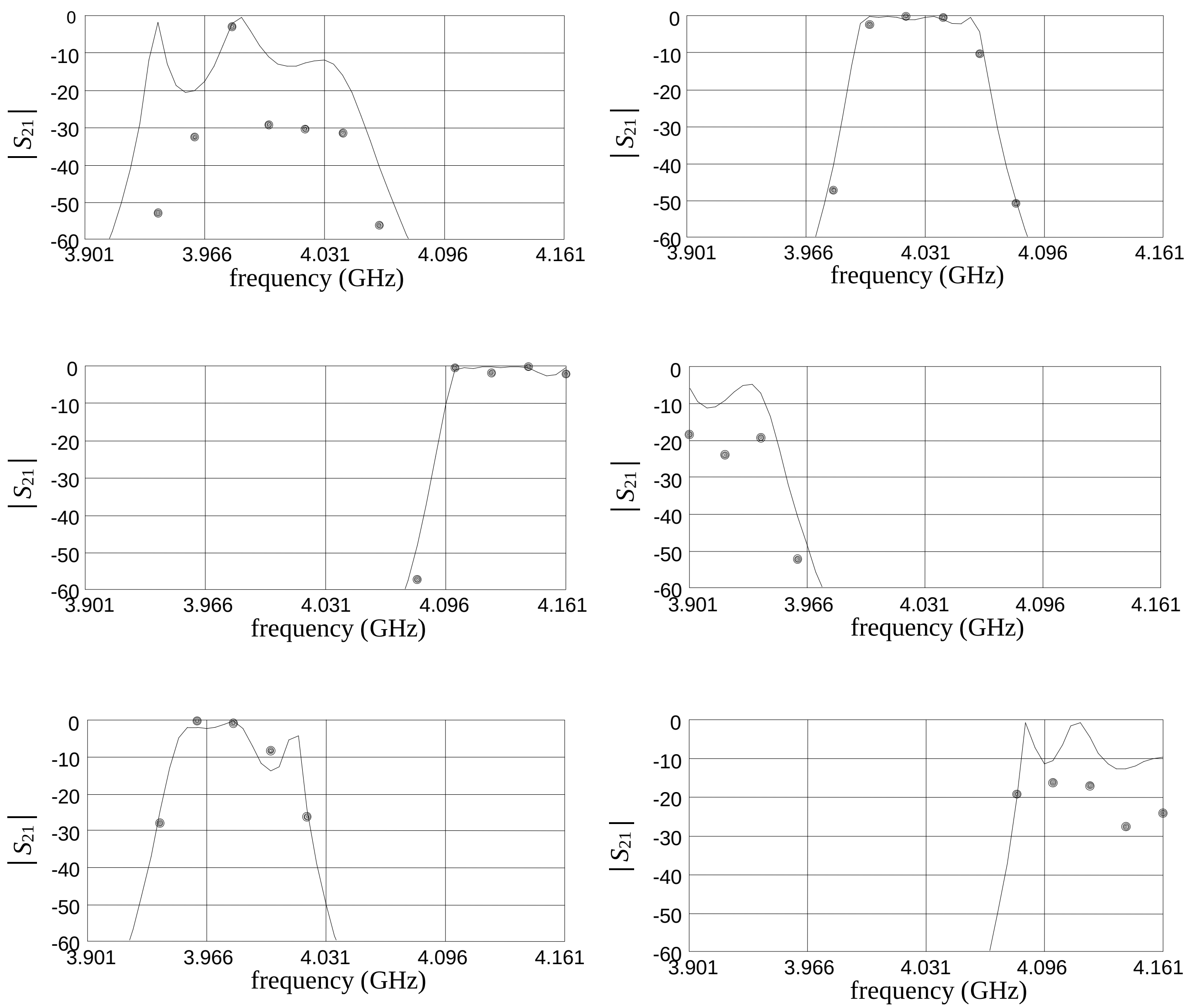
coarse model error w.r.t. ***em***™ at the learning and testing sets





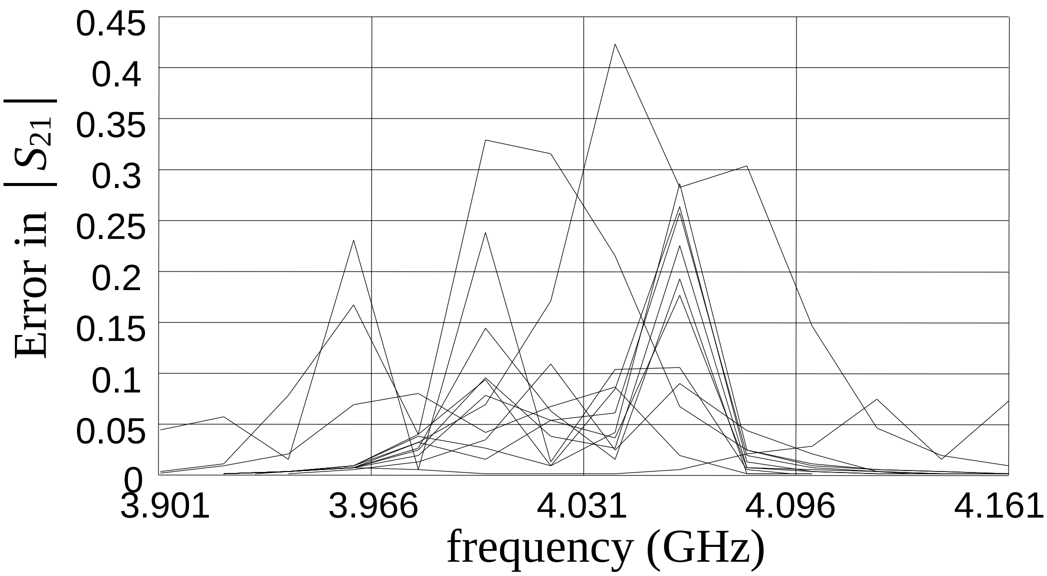
FMN Model for the HTS Filter (3LP:7-5-1)

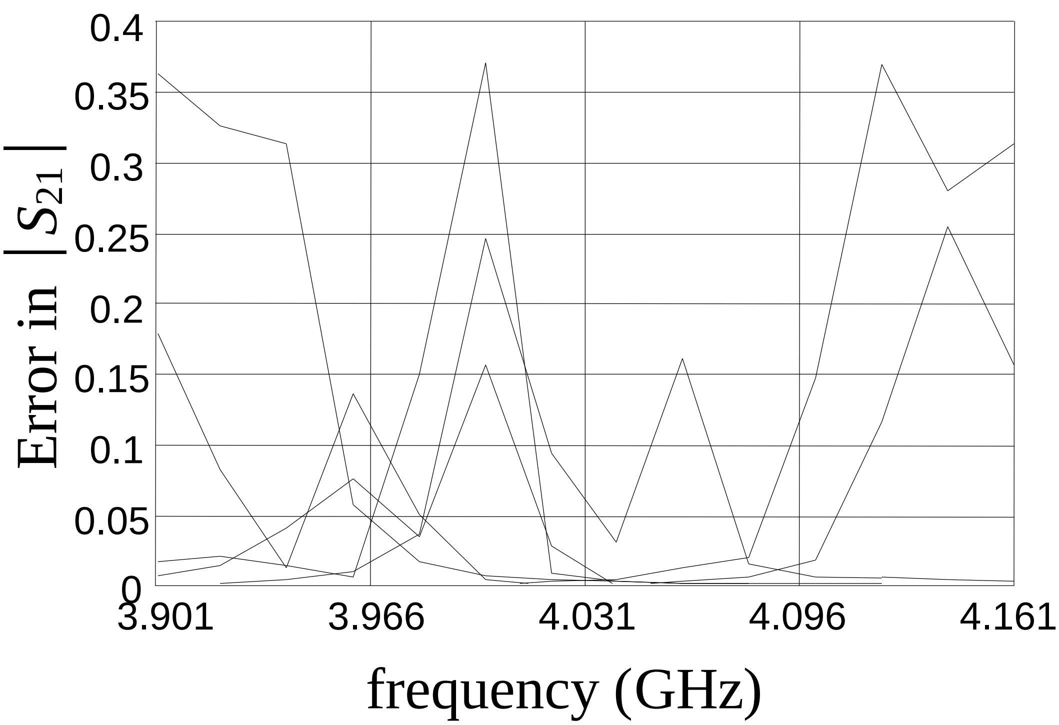
responses using ***em***™ (•) and FMN model (–) at the three learning and three testing points



FMN Model Response Errors for the HTS Filter

FMN model error w.r.t. ***em***™ at the learning and testing sets

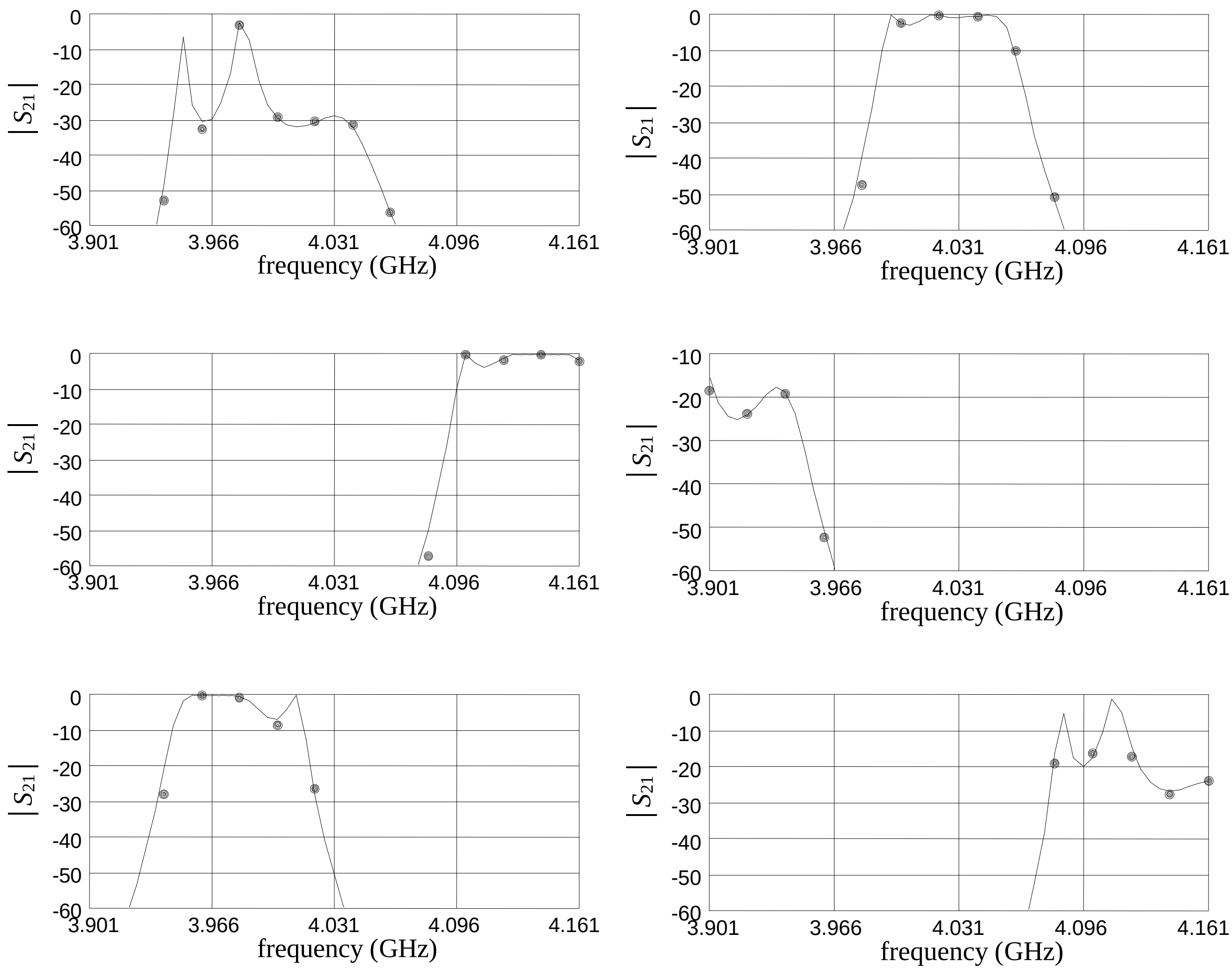




FPSMN Model Responses for the HTS Filter (3LP:7-7-3)

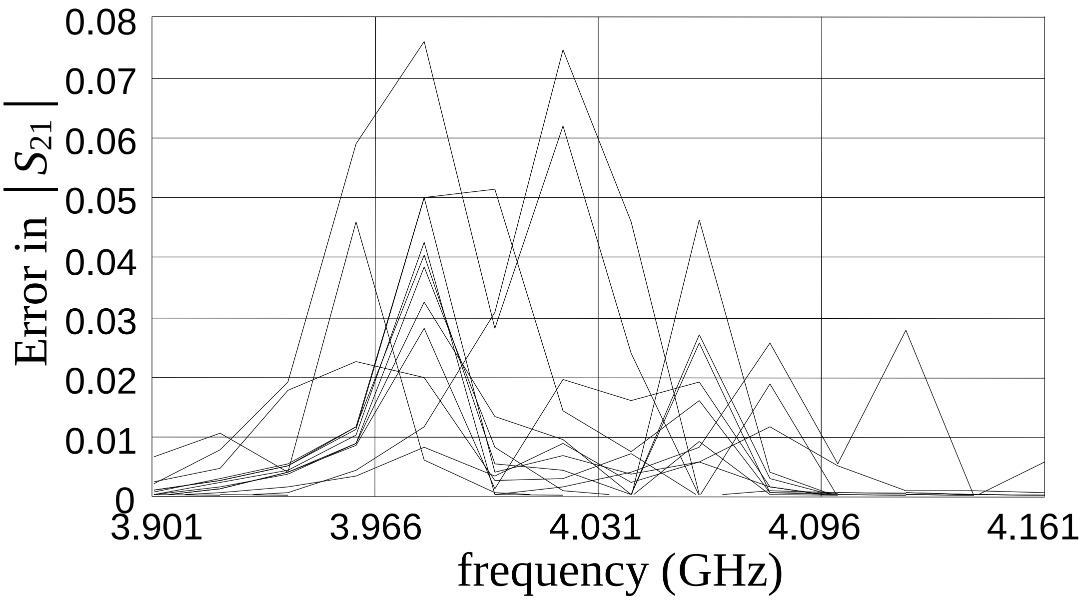
taking $x^s_c = [L_{1c}\ S_{1c}]^T$ and $x^s_f = [L_2\ L_3\ S_2\ S_3]^T$

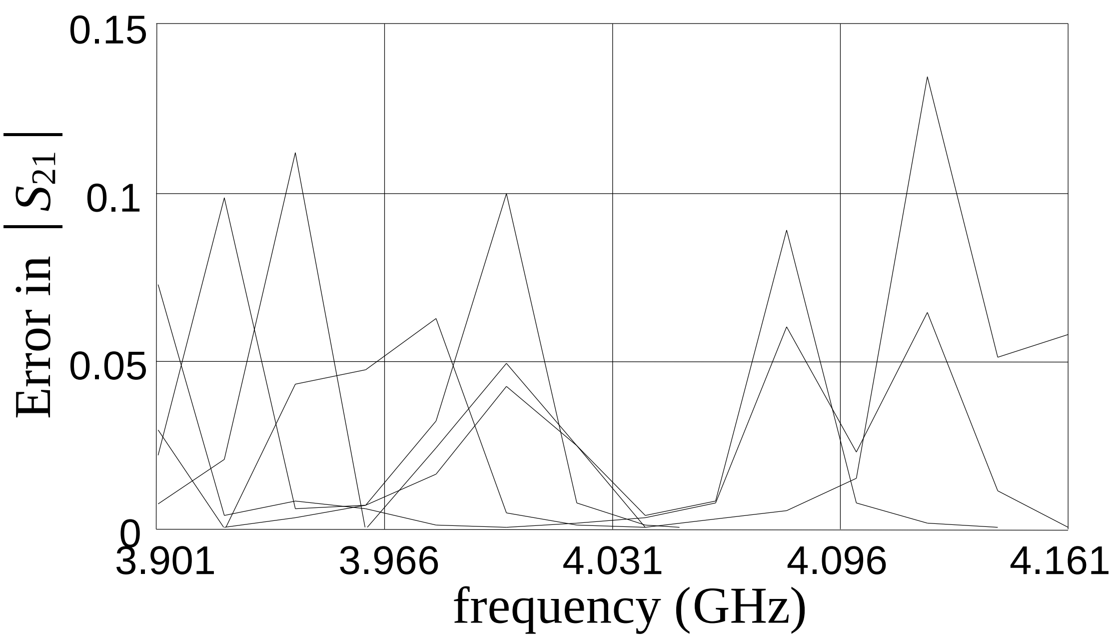
responses using *em*™ (•) and FPSMN model (–) at the three learning and three testing points



FPSMN Model Response Errors for the HTS Filter

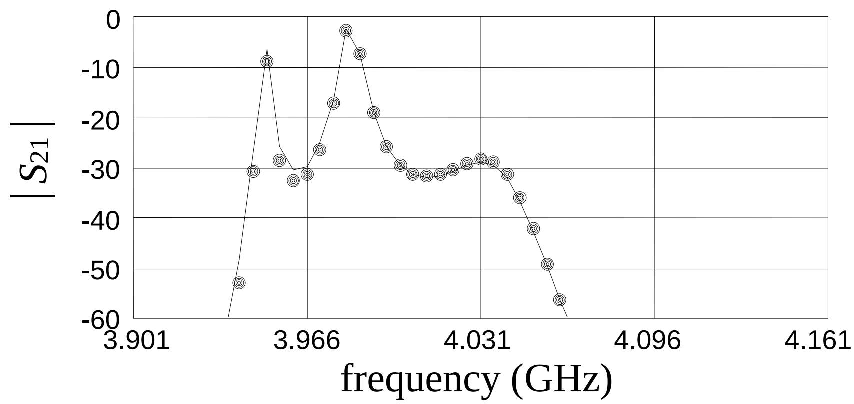
FPSMN model error w.r.t. ***em***™ at the learning and testing sets

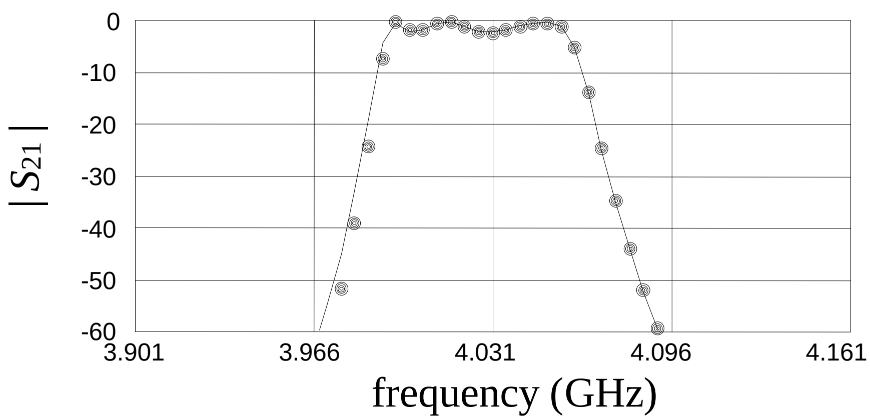


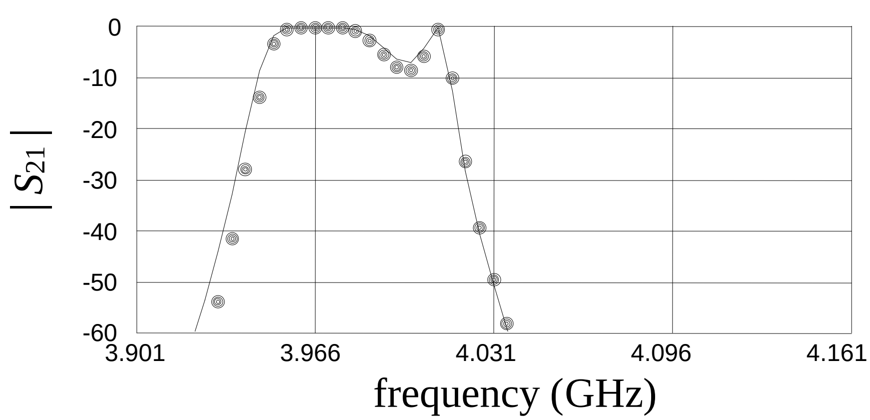


FPSMN Model for the HTS Filter: Fine Frequency Sweep

comparison between ***em***™ (•) and FPSMN model (–) at two learning and one testing points







Conclusions

we present novel applications of Space Mapping technology to the neuromodeling of microwave circuits

five powerful techniques to generate SM based neuromodels are illustrated: SMN, FDSMN, FSMN, FMN and FPSM

OSA90/hope™ implementations are illustrated

frequency-sensitive neuromappings expand the usefulness of quasi-static empirical models

FMN effectively aligns frequency-shifted responses

Huber optimization efficiently trains the neuromappings, exploiting its robust characteristics for data fitting

References

J.W. Bandler, M.A. Ismail, J.E. Rayas-Sánchez and Q.J. Zhang, "Neuromodeling of microwave circuits exploiting space mapping technology," *IEEE MTT-S Int. Microwave Symp*. (Anaheim, CA), June 1999.

P. Burrascano and M. Mongiardo, "A review of artificial neural networks applications in microwave CAD," *Int. J. RF and Microwave CAE*, Special Issue on Applications of ANN to RF and Microwave Design, vol. 9, 1999.

J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny and R.H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Trans. Microwave Theory Tech.*, vol. 42, 1994, pp. 2536-2544.

H. White, A.R. Gallant, K. Hornik, M. Stinchcombe and J. Wooldrige, *Artificial Neural Networks: Approximation and Learning Theory*. Oxford, UK: Blackwell, 1992.

A.H. Zaabab, Q.J. Zhang and M.S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Trans. Microwave Theory Tech.*, vol. 43, 1995, pp. 1349-1358.

P. Burrascano, M. Dionigi, C. Fancelli and M. Mongiardo, "A neural network model for CAD and optimization of microwave filters," *IEEE MTT-S Int. Microwave Symp. Dig.* (Baltimore, MD), 1998, pp. 13-16.

P. Watson and K.C. Gupta, "EM-ANN models for microstrip vias and interconnects in multilayer circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 44, 1996, pp. 2495-2503.

F. Wang and Q.J. Zhang, "Knowledge based neuromodels for microwave design," *IEEE Trans. Microwave Theory Tech.*, vol. 45, 1997, pp. 2333-2343.

M. Pozar, *Microwave Engineering*. Amherst, MA: John Wiley and Sons, 1998.

K.C. Gupta, R. Garg and I.J. Bahl, *Microstrip Lines and Slotlines*. Dedham, MA: Artech House, 1979.

R.M. Biernacki, J.W. Bandler, J. Song and Q.J. Zhang, "Efficient quadratic approximation for statistical design," *IEEE Trans. Circuit Syst.*, vol. 36, 1989, pp. 1449-1454.

OSA90/hope™ Version 4.0, formerly Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7, 1997, now HP EEsof Division, Hewlett-Packard Company, 1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799.

References (continued)

em™ Version 4.0b, Sonnet Software, Inc., 1020 Seventh North Street, Suite 210, Liverpool, NY 13088, 1997.

Empipe™ Version 4.0, formerly Optimization Systems Associates Inc., P.O. Box 8083, Dundas, Ontario, Canada L9H 5E7, 1997, now HP EEsof Division, Hewlett-Packard Company, 1400 Fountaingrove Parkway Santa Rosa, CA 95403-1799.

J.W. Bandler, S.H. Chen, R.M. Biernacki, L. Gao, K. Madsen and H. Yu, "Huber optimization of circuits: a robust approach," *IEEE Trans. Microwave Theory Tech.*, vol. 41, 1993, pp. 2279-2287.

NeuroModeler Version 1.0, Prof. Q.J. Zhang, Dept. of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6, 1998.

J.W. Bandler, R.M. Biernacki, S.H. Chen, W.J. Getsinger, P.A. Grobelny, C. Moskowitz and S.H. Talisa, "Electromagnetic design of high-temperature superconducting microwave filters," *Int. J. Microwave and Millimeter-Wave CAE*, vol. 5, 1995, pp. 331-343.