Fast Parametric Models for EM Design Using Neural Networks and Space Mapping

José E. Rayas-Sánchez

Research Group on Computer-Aided Engineering of Circuits and Systems (CAECAS)
Department of Electronics, Systems and Informatics
Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO)
Guadalajara, Mexico, 45090

http://www.desi.iteso.mx/caecas/

Q.J. Zhang

Department of Electronics, Carleton University
Ottawa, Canada K1S 5B6

http://www.doe.carleton.ca/~qjz

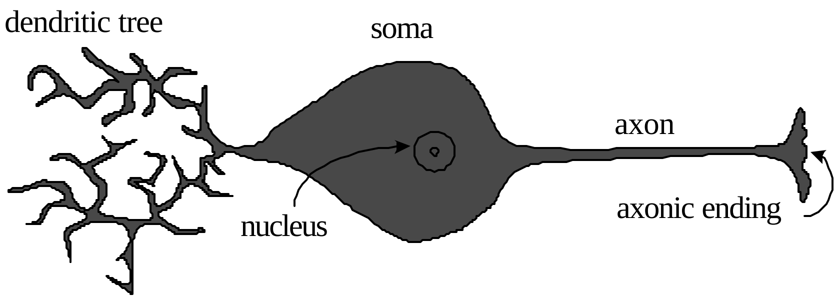
Tutorial on Advances in CAD Techniques for EM Modeling and Design (TSC)
2008 IEEE MTT-S International Microwave Symposium, Atlanta, GA, June 15, 2008

Outline

- Brief introduction to ANNs
- EM-based statistical analysis
- Input Space Mapping
- Linear-Input Neural-Output Space Mapping (LINO-SM)
- LINO-SM approach to yield estimation
- Constrained Broyden-Based Space Mapping
- Training the Output Neuromapping
- Example
- Conclusions

2

Biological Neuron



(Kartalopoulos, 1996)

Artificial Neural Networks (ANNs)

- An Artificial Neural Network (ANN) is a massively parallel distributed processor made up of simple processing units, that is able of acquiring knowledge from its environment though a learning process
- ANNs are also information processing systems that emulate biological neural networks: they are inspired in the ability of human brain to learn from observation and generalize by abstraction

Artificial Neurons

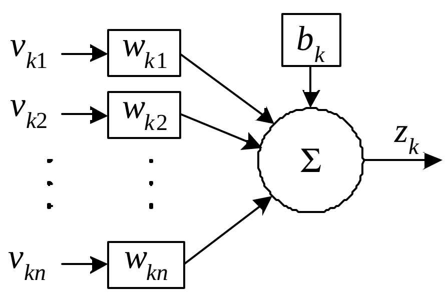
- An artificial neuron is a simple processing unit that receives and combines signals from many other neurons
- Common types of artificial neurons are:

  Linear Neurons

  Inner-Product Nonlinear Neuron

  Euclidean Distance Neuron

Linear Neuron



$$z_k = b_k + \boldsymbol{v}_k^T \boldsymbol{w}_k$$

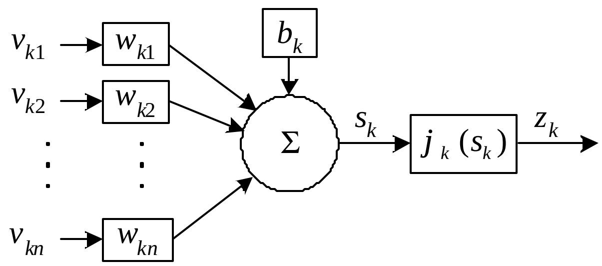
$\boldsymbol{v}_k = [v_{k1} \ldots v_{kn}]^T$ vector of inputs

$\boldsymbol{w}_k = [w_{k1} \ldots w_{kn}]^T$ vector of weighting factors

b_k bias or offset term

z_k activation potential or induced local field, output signal

Inner Product Nonlinear Neuron



$$s_k = b_k + \boldsymbol{v}_k^T \boldsymbol{w}_k$$

$$z_k = j_k(s_k)$$

$\boldsymbol{v}_k = [v_{k1} \ldots v_{kn}]^T$ inputs $\boldsymbol{w}_k = [w_{k1} \ldots w_{kn}]^T$ weighting factors

b_k bias or offset term s_k activation potential

$j_k(s_k)$ activation function or squashing function

z_k output signal

Typical Activation Functions

Sigmoid or logistic function

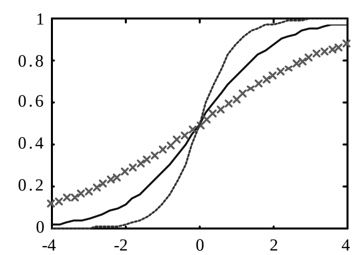
$$z_k = j_k(s_k) = \frac{1}{1+e^{-s_k}}$$

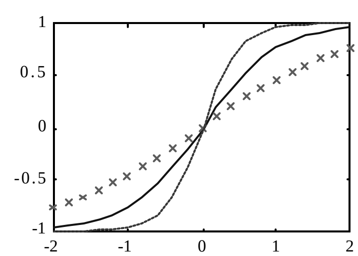
With a slope parameter

$$z_k = j_k(s_k) = \frac{1}{1+e^{-a s_k}}$$

Hyperbolic tangent function

$$z_k = j_k(s_k) = \tanh(s_k) = \frac{e^{s_k} - e^{-s_k}}{e^{s_k} + e^{-s_k}}$$



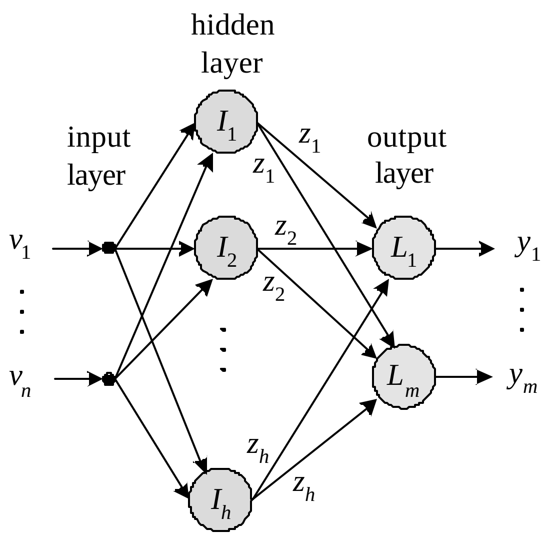


a = 1 (¾), a = 0.5 (´) and a = 2 (¾)

Some ANN Paradigms

- Multilayer Perceptrons
- Radial Basis Functions
- Recurrent Neural Networks

3-Layer Perceptrons (3LP)



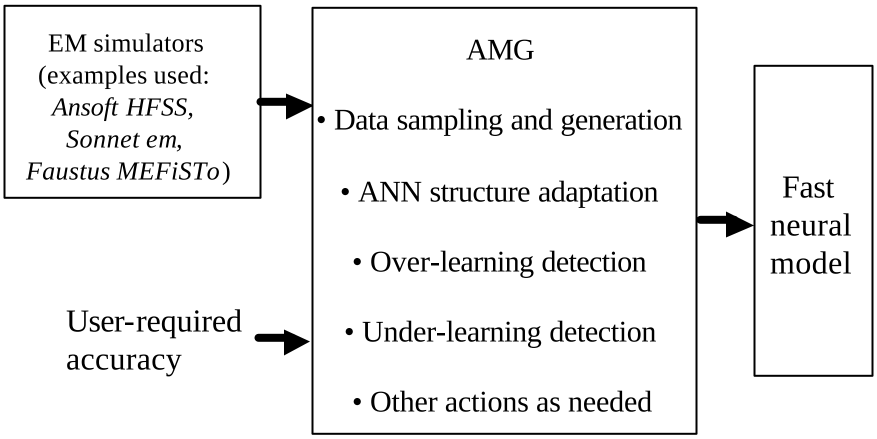
$$\boldsymbol{y} = \boldsymbol{b}^o + \boldsymbol{W}^o \boldsymbol{F}(\boldsymbol{s})$$
$$\boldsymbol{s} = \boldsymbol{b}^h + \boldsymbol{W}^h \boldsymbol{v}$$

where

$$\boldsymbol{W}^o = [\boldsymbol{w}_1^{o^T} \quad \cdots \quad \boldsymbol{w}_m^{o^T}]^T$$
$$\boldsymbol{W}^h = [\boldsymbol{w}_1^{h^T} \quad \cdots \quad \boldsymbol{w}_h^{h^T}]^T$$
$$\boldsymbol{F}(\boldsymbol{s}) = [j(s_1) \quad \cdots \quad j(s_h)]^T$$
$$\boldsymbol{b}^h = [b_1^h \quad b_2^h \quad \cdots \quad b_h^h]^T$$
$$\boldsymbol{b}^o = [b_1^o \quad b_2^o \quad \cdots \quad b_m^o]^T$$

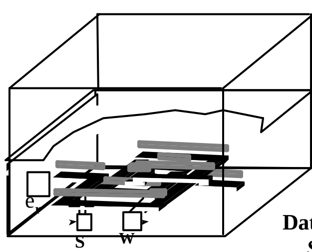
$\boldsymbol{y}, \boldsymbol{b}^o \in \Re^m$ $\quad \boldsymbol{s}, \boldsymbol{b}^h, \boldsymbol{F} \in \Re^h$ $\quad \boldsymbol{W}^o \in \Re^{m \times h}$ $\quad \boldsymbol{W}^h \in \Re^{n \times h}$

Automated Model Generation (AMG)

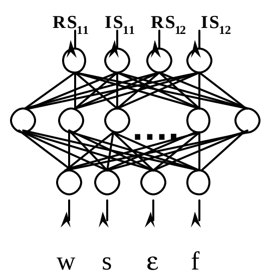


11

AMG for Spiral Inductor



Data Generators:
Sonnet em
Ansoft-HFSS



AMG uses data sampling algorithm to sample data at critical locations. With the same amount of training data, AMG obtains better model accuracy than conventional training techniques.

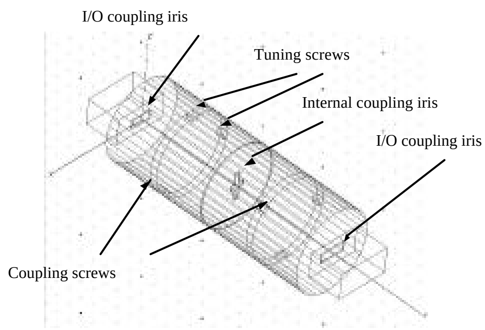
Model accuracy under limited training data

ANN Training Technique	Testing Error
Conventional training	6.25%
AMG	2.34%
Advanced AMG (KAMG-PKI)	0.85%

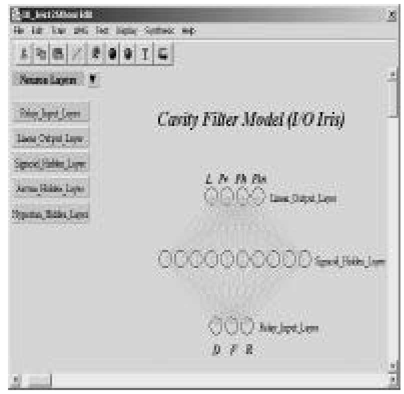
12

Inverse Modeling by Neural Network

(Waveguide Filter Example, H. Kabir, Y. Wang, M. Yu and Q. Zhang, 2006)



Data Generator: ***Ansoft-HFSS (3D EM)***



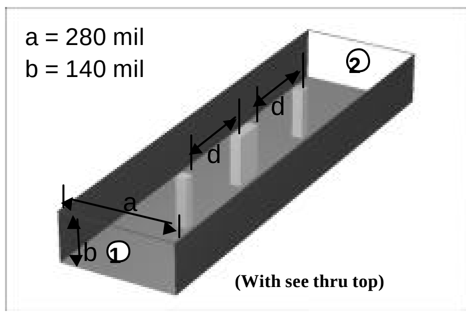
Input neurons: electrical parameters
Output neurons: geometrical parameters

Filter Dimensions By Neural Net vs Measurement

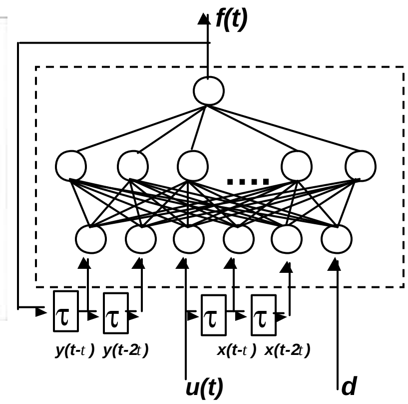
	Neural Model (inch)	Measurement (inch)	Difference (inch)
I/O irises	0.405	0.405	0
M_{23} iris	0.299	0.297	-0.002
M_{14} iris	0.212	0.216	0.004
M_{11}/M_{44} tuning screws	0.045	0.005	-0.040
M_{22}/M_{33} tuning screws	0.133	0.135	0.002
M_{12}/M_{34} coupling screws	0.111	0.115	0.004
Cavity length	1.865	1.864	-0.001

Time-Domain Modeling by Neural Networks

WR-28 Rectangular Waveguide Ka-band (26.5 to 40Ghz)



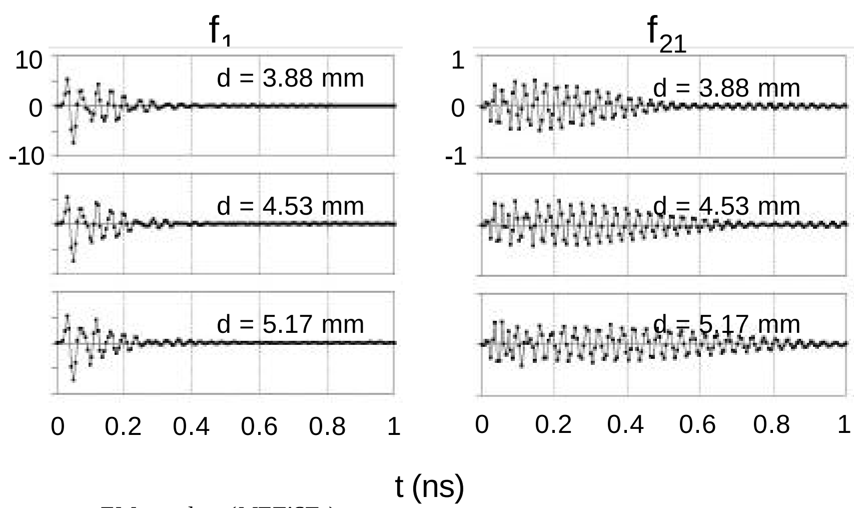
Location of conducting posts (*d*) controls pass band behavior



Data Generator: *MEFiSTo 3D Professional*

Recurrent neural network (RNN)

EM and RNN Transient Responses



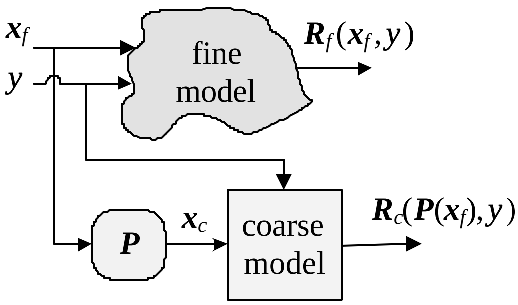
• • • EM test data (*MEFiSTo*)
——— RNN

EM-based Interpolating Surrogates for Yield Estimation using Neural Space Mapping Methods

EM-based Statistical Analysis

- Statistical analysis and yield prediction are crucial for manufacturability
- Reliable yield prediction typically requires massive amount of high-fidelity simulations (full-wave EM simulations)
- Performing Monte Carlo yield analysis by directly using EM simulations is not feasible for most practical problems
- Using an interpolating surrogate based on linear-input neural-output space mapping can be a solution

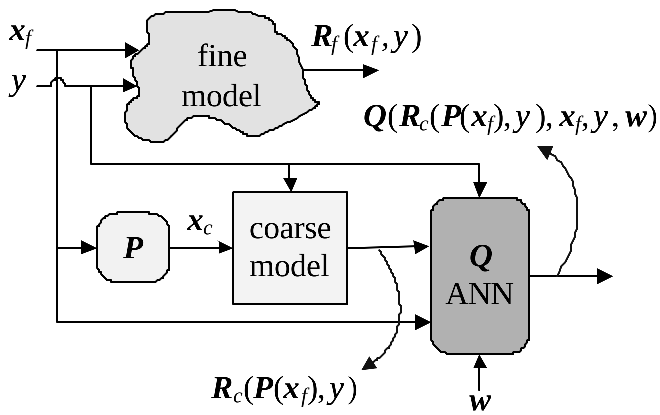
Input Space Mapping



☺ $R_f(x_f^{SM}, y) \approx R_c(x_c^*, y)$

☹ $R_c(P(x_f^{SM}))$ can not accurately estimate the fine model yield around x_f^{SM}

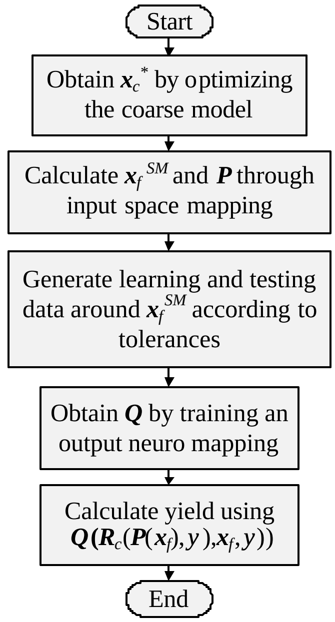
Linear-Input Neural Output Space Mapping



$$Q(R_c(Bx_f + c, ?), x_f, ?, w^*) = R_f(x_f, ?)$$

for all x_f and y in the training region

LINO-SM approach to Yield Estimation



21

Constrained Broyden-Based SM

Begin

find $\boldsymbol{x}_c^*$ solving (1)

$i = 0$, $\boldsymbol{x}_f^{(i)} = \boldsymbol{x}_c^*$, $\boldsymbol{B}^{(i)} = \boldsymbol{I}$, $d = 0.3$

$\boldsymbol{f}^{(i)} = \boldsymbol{P}(\boldsymbol{x}_f^{(i)}) - \boldsymbol{x}_c^*$ using (2)

repeat until *stopping_criterion*

solve $\boldsymbol{B}^{(i)}\boldsymbol{h}^{(i)} = -\boldsymbol{f}^{(i)}$ for $\boldsymbol{h}^{(i)}$

$\boldsymbol{x}_f^{(test)} = \boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}$

while $\boldsymbol{x}_f^{(test)} \prec \boldsymbol{x}_f^{\min} \vee \boldsymbol{x}_f^{(test)} \succ \boldsymbol{x}_f^{\max}$

$\boldsymbol{h}^{(i)} = d\,\boldsymbol{h}^{(i)}$

$\boldsymbol{x}_f^{(test)} = \boldsymbol{x}_f^{(i)} + \boldsymbol{h}^{(i)}$

end

$\boldsymbol{x}_f^{(i+1)} = \boldsymbol{x}_f^{(test)}$

$\boldsymbol{f}^{(i+1)} = \boldsymbol{P}(\boldsymbol{x}_f^{(i+1)}) - \boldsymbol{x}_c^*$ using (2)

$$\boldsymbol{B}^{(i+1)} = \boldsymbol{B}^{(i)} + \frac{\boldsymbol{f}^{(i+1)}\boldsymbol{h}^{(i)T}}{\boldsymbol{h}^{(i)T}\boldsymbol{h}^{(i)}},\quad i = i + 1$$

end

(1)

$$\boldsymbol{x}_c^* = \arg\min_{\boldsymbol{x}_c} U(\boldsymbol{R}_c(\boldsymbol{x}_c, ?))$$

(2)

$$\boldsymbol{P}(\boldsymbol{x}_f) = \arg\min_{\boldsymbol{x}_c} \left\| \boldsymbol{e}_1^T \quad \ldots \quad \boldsymbol{e}_p^T \right\|_2^2$$

$$\boldsymbol{e}_j(\boldsymbol{x}_f) = \boldsymbol{R}_{fs}(\boldsymbol{x}_f, ?_j) - \boldsymbol{R}_{cs}(\boldsymbol{x}_c, ?_j)$$

$$\boldsymbol{x}_f^{SM} = \boldsymbol{x}_f^{(i)}$$

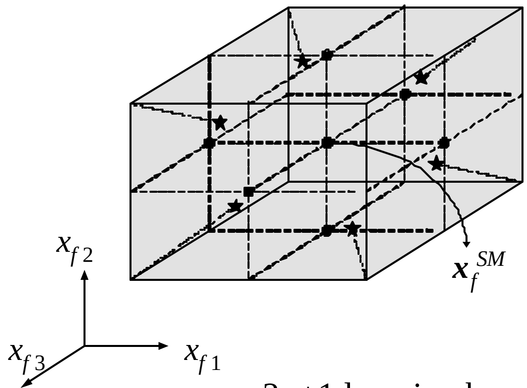
$$\boldsymbol{P}(\boldsymbol{x}_f) = \boldsymbol{B}\boldsymbol{x}_f + \boldsymbol{c}$$

where $\boldsymbol{B} = \boldsymbol{B}^{(i)}$ and $\boldsymbol{c} = \boldsymbol{x}_c^* - \boldsymbol{B}\boldsymbol{x}_f^{SM}$

22

Generating Learning and Testing Points

- ■ learning base point
- ★ testing base point



2n+1 learning base points in a star distribution
2n testing base points in a rotated star distribution

Training the Output Neuro Mapping

Begin

Generate $\boldsymbol{R}_{CL}$, $\boldsymbol{R}_{CT}$, $\boldsymbol{R}_{FL}$ and $\boldsymbol{R}_{FT}$

$e_L^{old} = \| \boldsymbol{R}_{CL} - \boldsymbol{R}_{FL} \|_F$, $e_T^{old} = \| \boldsymbol{R}_{CT} - \boldsymbol{R}_{FT} \|_F$

$h = m$, $i = 1$

$\boldsymbol{w}^{(i)} = \arg \min_{\boldsymbol{w}} \| \boldsymbol{E}_L(\boldsymbol{w}) \|_F$

$e_L = \| \boldsymbol{Q}_L(\boldsymbol{w}^{(i)}) - \boldsymbol{R}_{FL} \|_F$

$e_T = \| \boldsymbol{Q}_T(\boldsymbol{w}^{(i)}) - \boldsymbol{R}_{FT} \|_F$

while $e_T^{old} \geq e_T \quad \vee \quad e_L \geq e_T$

$e_T^{old} = e_T$, $e_L^{old} = e_L$, $i = i + 1$, $h = h + 1$

$\boldsymbol{w}^{(i)} = \arg \min_{\boldsymbol{w}} \| \boldsymbol{E}_L(\boldsymbol{w}) \|_F$

$e_L = \| \boldsymbol{Q}_L(\boldsymbol{w}^{(i)}) - \boldsymbol{R}_{FL} \|_F$

$e_T = \| \boldsymbol{Q}_T(\boldsymbol{w}^{(i)}) - \boldsymbol{R}_{FT} \|_F$

end

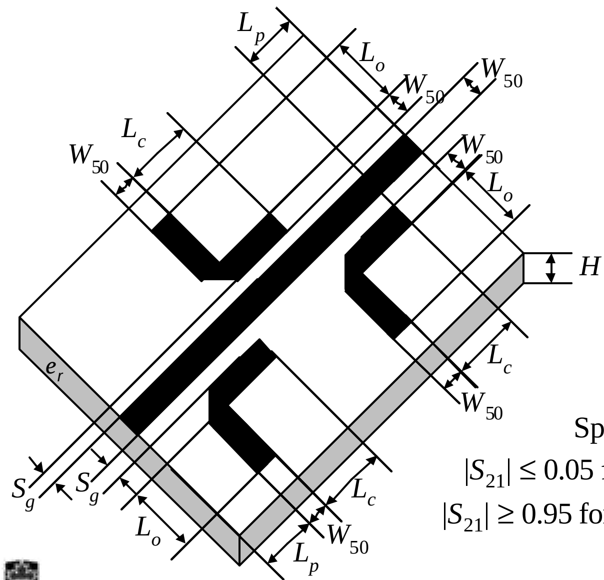
$\boldsymbol{w}^* = \boldsymbol{w}^{(i-1)}$

end

$$\boldsymbol{E}_L(\boldsymbol{w}) = \boldsymbol{R}_{FL} - \boldsymbol{Q}_L(\boldsymbol{w})$$

$\boldsymbol{Q}(\boldsymbol{R}_c(\boldsymbol{B}\boldsymbol{x}_f + \boldsymbol{c}, ?), \boldsymbol{x}_f, ?, \boldsymbol{w}^*) = \boldsymbol{R}_f(\boldsymbol{x}_f, ?)$
for all $\boldsymbol{x}_f$ and $\boldsymbol{y}$ in the training region

Microstrip Notch Filter



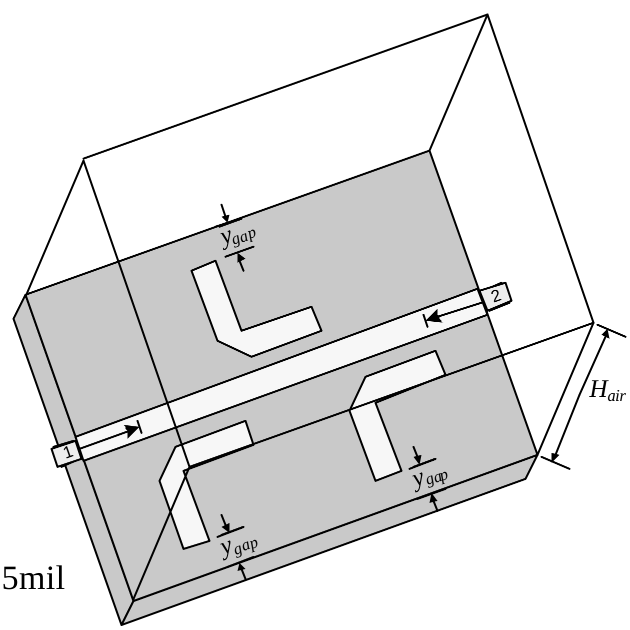
H = 10mil
W_{50} = 31mil
e_r = 2.2
loss tan = 0.0009
(RT Duroid 5880)

$$x_f = [L_c \;\; L_o \;\; S_g]^T$$

Specifications:
$|S_{21}| \leq 0.05$ for 13.19GHz $\leq f \leq$ 13.21GHz
$|S_{21}| \geq 0.95$ for $f \leq$ 13GHz and $f \geq$ 13.4GHz

25

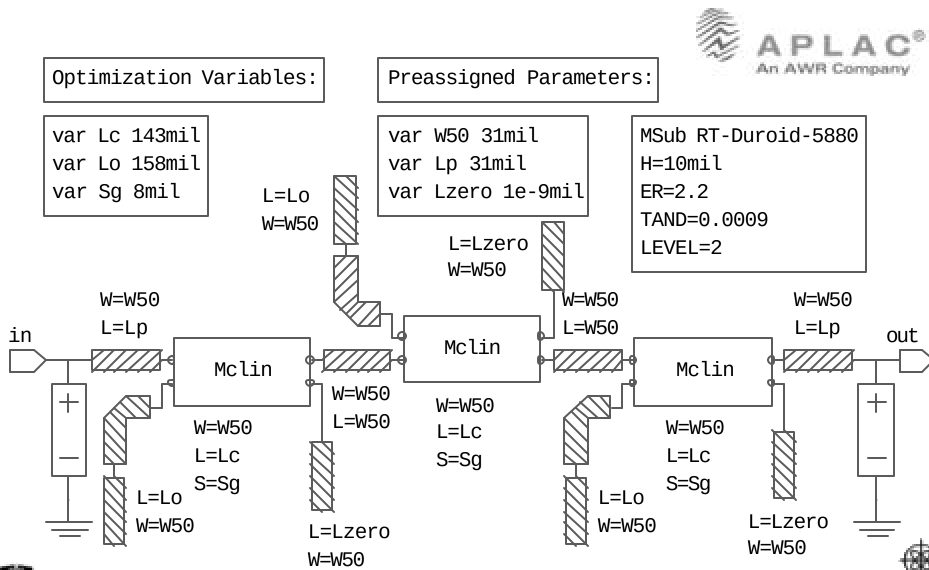
Microstrip Notch Filter – Fine Model



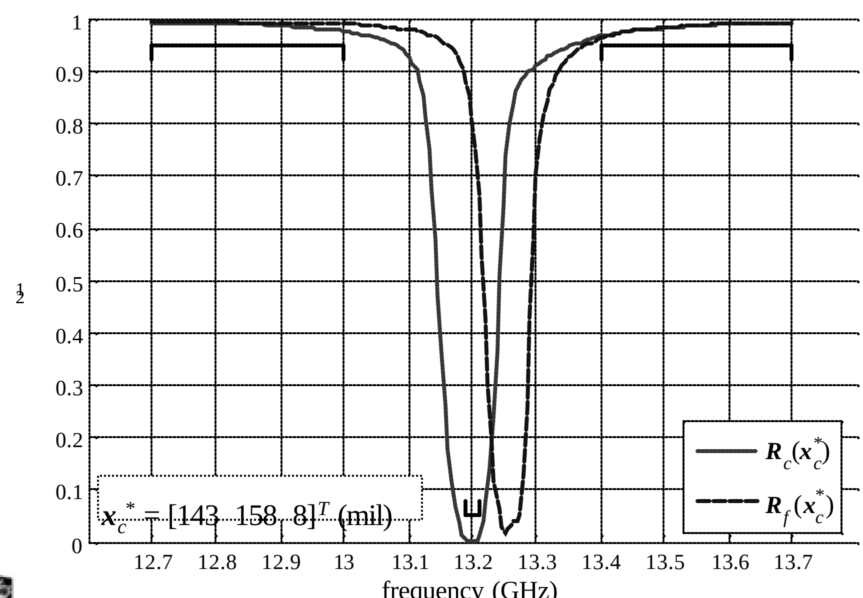
H_{air} = 60 mil
$L_p = ½(L_o+L_c)$
$Y_{gap} = L_o$
grid = 0.5mil×0.5mil

26

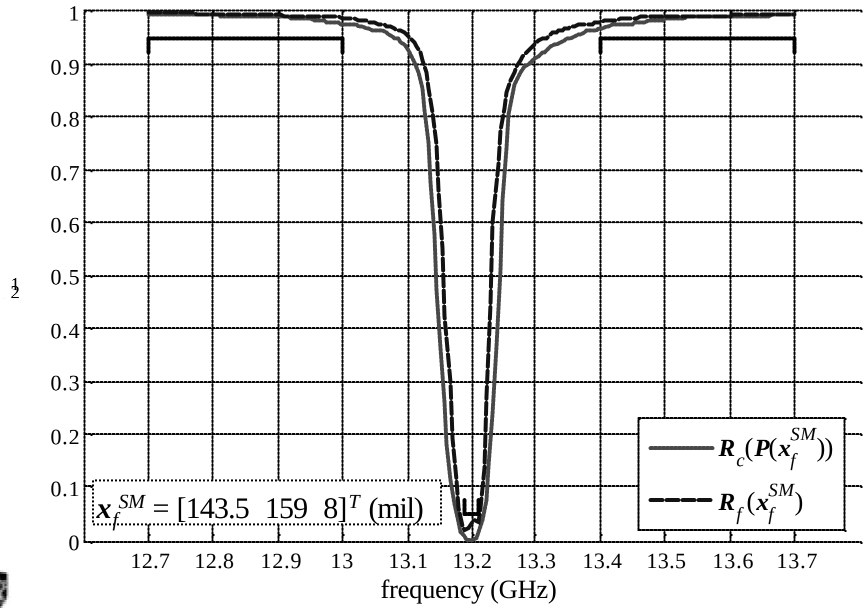
Microstrip Notch Filter – Coarse Model



Microstrip Notch Filter – Starting Point

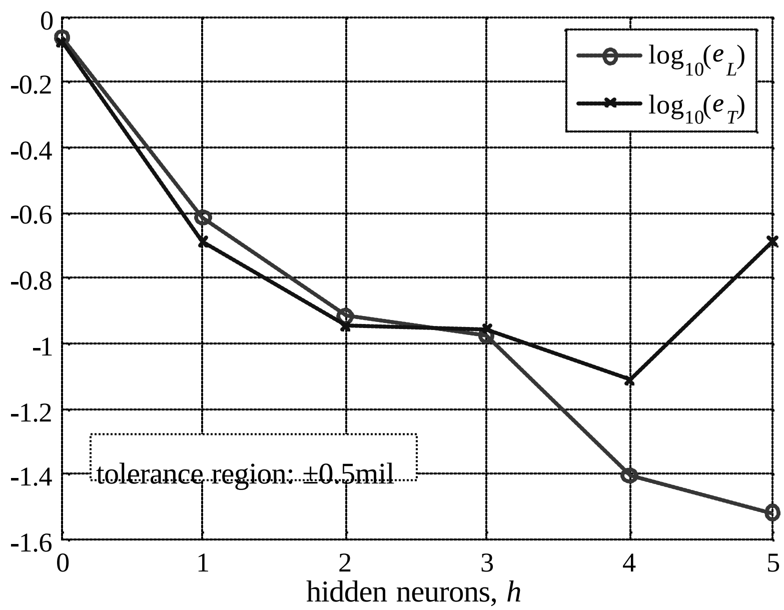


Microstrip Notch Filter – SM Solution



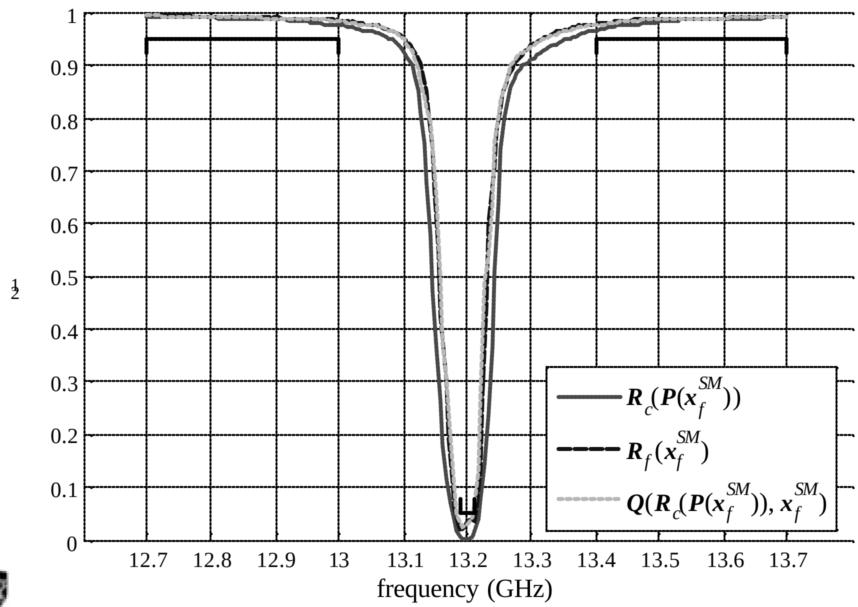
29

Microstrip Notch Filter – Training *Q*



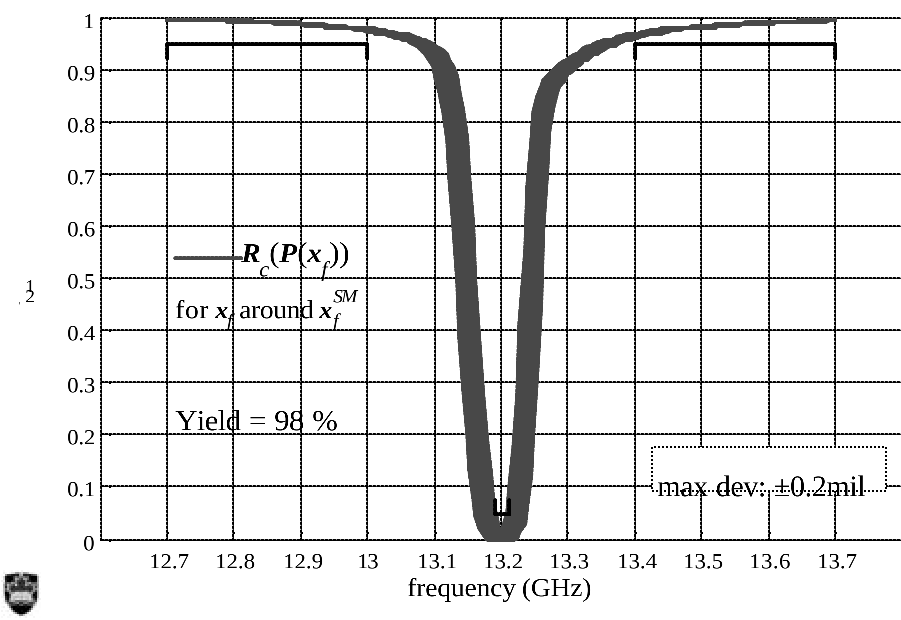
30

Microstrip Notch Filter – LINOSM Solution



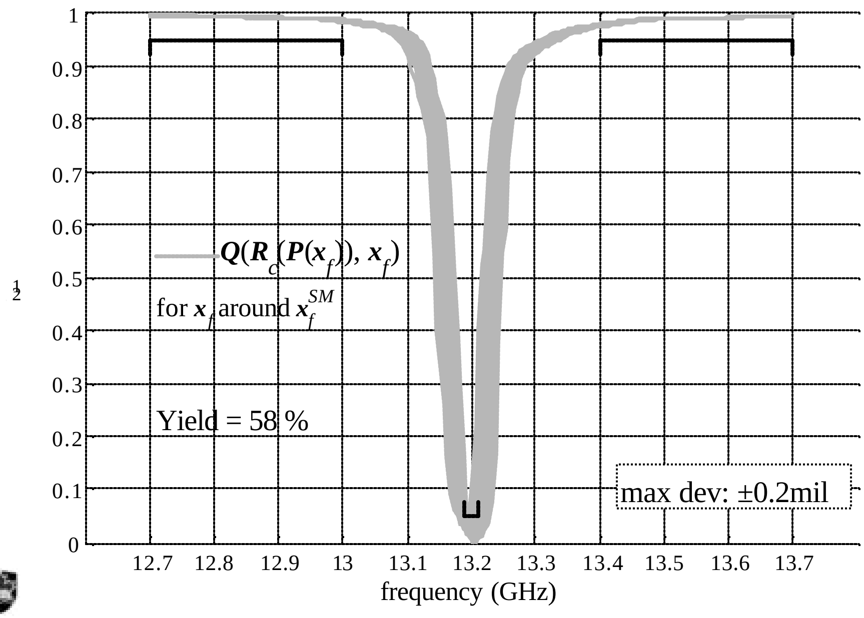
31

Microstrip Notch Filter – LISM Yield



32

Microstrip Notch Filter – LINOSM Yield



Conclusions

- We described a method for highly accurate EM-based statistical analysis and yield estimation of RF and microwave circuits
- It consists of applying a constrained Broyden-based linear-input space mapping, followed by a neural-output space mapping, in which the responses, the design parameters and independent variable are mapped
- The output neuromodel is trained using reduced sets of learning and testing samples
- The resultant interpolating surrogate model is used as a very efficient vehicle for accurate statistical analysis and yield prediction