J.N. Hwang, J.J. Choi, S. Oh, R.J. Marks II, "Query learning based on boundary search and gradient computation of trained multilayer perceptrons", Proceedings of the International Joint Conference on Neural Networks, San Diego, June, 1990, 17-21 June 1990, vol. III, pp.III57-III62.

Frequency Selective Surface Design Based on Iterative Inversion of Neural Networks

Jenq-Neng Hwang, Chi-Hou Chan, Robert J. Marks II

Department of Electrical Engr., FT-10
University of Washington
Seattle, WA 98195

Abstract

This paper proposes a novel approach to solve a constrained inverse problem encountered in the design of frequency selective surfaces (FSS's). Due to the many-to-one nonlinear functional relationship between an FSS and its frequency response, there is no closed form solution directly from the given desired frequency response to the corresponding surface. Therefore, to design an FSS for a given response, one has to search in the knowledge base through a trial-and-error procedure. This procedure can be a very laborious and tedious process. Our approach adopts an iterative regularized inversion technique, which starts with an inversion algorithm for multilayer perceptrons to generate the corresponding 2-D surface for the given desired frequency response, a constraint satisfaction mechanism is then used to reshape the 2-D surface to satisfy the constraints, and the resulting surface is used as the initial point for the next inversion algorithm. This procedure is mathematically similar to the projection onto convex set algorithm for constrained optimization problems.

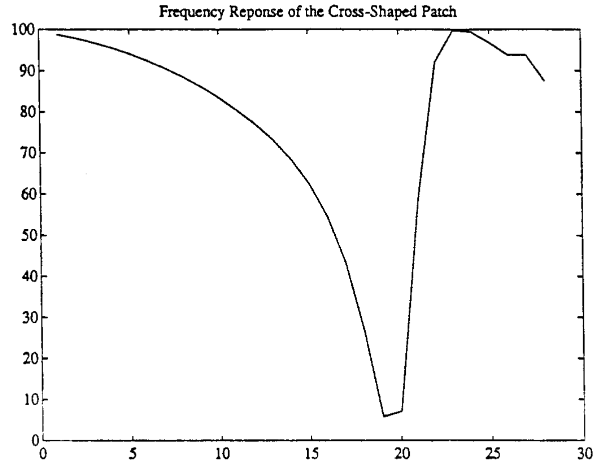
1 Frequency Selective Surface Design

Frequency selective surfaces (FSS's) have widespread applications over much of the electromagnetic spectrum [4]. In the microwave region, they are used as reflector antenna dichroic surfaces and antenna radomes. In the far-infrared region, they are used as polarizers, beam splitters, and mirrors for laser applications. Another application of the FSS in this frequency range is in infrared sensors. In the near-infrared and visible portions of the spectrum, they are used to aid in the collection of solar energy. FSS's usually comprise periodically arranged identical metallic patch or aperture elements supported by dielectric layers. They exhibit total reflection or transmission in the neighborhood of the element resonance. To model an arbitrarily-shaped patch or aperture elements, the unit cell is uniformly divided into an $N \times N$ array of subcells. The geometrical description of the unit-cell is given as follows. Subcells that correspond to the conductor region of the unit cell are represented by 1's whereas subcells outside the conductor region are represented by 0's (one example is shown in Figure 1(a)). The frequency response of the FSS's are then calculated for a given incident plane wave (see Figure 1(b)).

There is no closed form solution directly from the given desired frequency response to the corresponding surface. Therefore, to design an FSS for a given response, one has to search in the knowledge base to look for the surface that gives the closest response to the desired one. Unit-cell geometry of the chosen surface is then perturbed (through a trial-and-error procedure) until its response matches all the design criteria. The process is too laborious and human dependent. We propose an alternate design procedure that does not have these negative features.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)



(b)

Figure 1: (a) In an FSS, subcells that correspond to the conductor region of the unit cell are represented by 1's whereas subcells outside the conductor region are represented by 0's. (b) The corresponding frequency response of the FSS.

2 Learning and Inversion of Multilayer Perceptrons

Multilayer perceptrons are feed-forward neural networks which have one or more layers of *hidden neurons* between the input and output layers. When the net is trained, it is used to generate response given the input test data. The converse problem of generating input vectors corresponding to a given output vector is referred to as *inversion* (see Figure 2). The system dynamics in the retrieving phase of an L-layer neural net can be described by the following equations:

$$\begin{aligned} u_i(l+1) &= \sum_{j=1}^{N_l} w_{ij}(l+1)a_j(l) + \theta_i(l+1) \\ a_i(l+1) &= f(u_i(l+1)) \quad 1 \le i \le N_{l+1}, \quad 0 \le l \le L-1 \end{aligned} \tag{1}$$

where $a_j(l)$ denotes the activation value of the j^{th} neuron at the l^{th} layer and f is the nonlinear activation function.

Network Learning The learning phase of a multilayer perceptron uses the back propagation learning rule, an iterative gradient descent algorithm designed to minimize the mean squared error between the the desired target values and the actual output values [1]:

$$w_{ij}(l) \Longleftarrow w_{ij}(l) - \eta \, \frac{\partial E}{\partial w_{ij}(l)} \tag{2}$$

where $E = \frac{1}{2} \sum_{i=1}^{N_L} (t_i - a_i(L))^2$.

Inversion of a Network The inversion of a network will generate the input $\{a_j(0)\}$ (or inputs) that can produce a desired output vector. By taking advantage of the *duality* between the weights and the activation in minimizing the mean squared error between the the desired target values and the actual output values, the iterative gradient descent algorithm can also be applied to obtain the desired input.

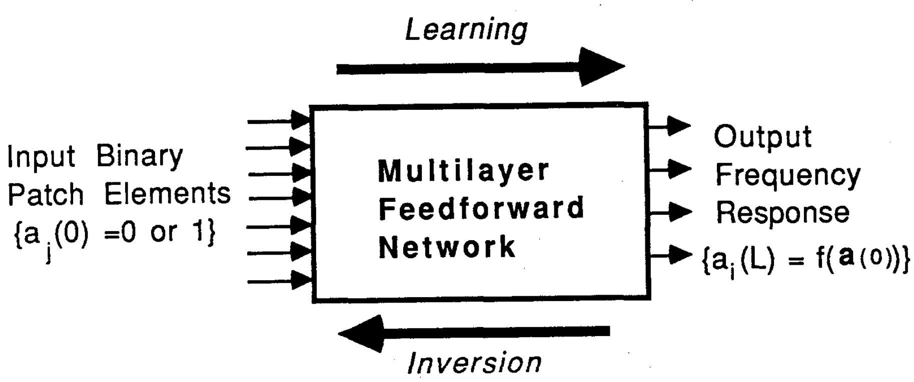


Figure 2: Learning and inversion of a multilayer perceptron.

$$a_j(0) \Longleftarrow a_j(0) - \eta \frac{\partial E}{\partial a_j(0)} \tag{3}$$

The idea is similar to the back-propagation algorithm, where the error signals are propagated back to tell the weights the manner in which to change in order to decrease the output error. The inversion algorithm back-propagates the error signals down to the input layer to update the activation values of input units so that the output error is decreased [2]. The inversion algorithm has been successfully used as an excellent tool for locating the region of ambiguity in a classifier training, and tremendously improve the classification accuracy through an oracle based query learning technique [3].

3 Iterative Network Inversion and Constraints Satisfaction

The FSS design problem can be treated as an *many-to-one* nonlinear mapping problem, which maps a 2-D surface of $N \times N$ binary subcells to a corresponding 1-D frequency response. In order to have the multilayer perceptron efficiently trained to realize the network mapping, the input to the network should be the 2-D surface and the output of the network should be the frequency response. Note that we cannot interchange the role of the I/O, because the non-invertible many-to-one functional relationship requires presentation of conflicting data to the neural network in the training stage.

After the network is trained, the inversion algorithm, with a random input initialization, is used to generate the corresponding 2-D surface for the given desired frequency response. We do not expect the resulting 2-D surface of the network inversion algorithm to be satisfactory. It's accuracy, however, will be improved iteratively through the imposition of constraints. We massage the inversion result to be a grid of zeros and ones that is, in some sense, close to the inversion result. This result will then be used as a new point of initialization of the inversion procedure, etc. Mathematically, this procedure is similar to the *projection onto convex set* algorithm [6, 7]. We choose the following two constraints for the surface patch elements.

1. The generated subcells should contain binary values.
2. The "1" subcell should cluster together to maximize the spatial continuity (many clusters are allowed).

In general, more constraints can be imposed, *e.g.*, the constraints on the dielectric constant and the thickness of the surface.

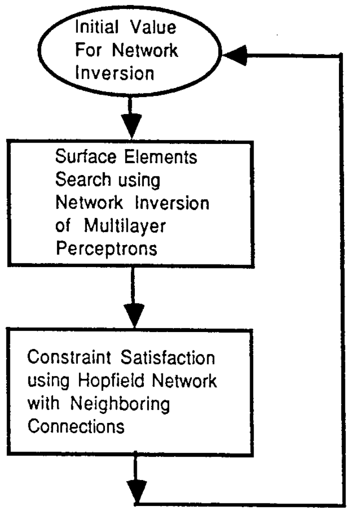


Figure 3: The schematic diagram of an complete FSS design based on iterative inversion of neural networks.

The continuous valued Hopfield model [5] with neighboring excitatory connections proves to be a good choice for the constraint satisfaction mechanism. There are some other alternatives considered, *e.g.*, the adaptive median filter. As shown in Figure 3, the complete FSS design procedure starts with a random guess of FSS subcells for the network inversion mechanism, and results in a surface with frequency response close to the desired one, while without satisfying the two imposed constraints. The inverted surface is then passed to the network constraint mechanism and results in a surface with constraints more satisfied, while with its frequency response more deviated from the desired one. The surface is again used as the initial state for the network inversion mechanism. The above procedure iterates several times until convergence is reached, which gives rises to a constraint satisfied surface with desired frequency response.

The preliminary simulation for small size FSS design is quite encouraging. A total of 75 FSS 2-D surfaces (16×16) with their 1-D frequency responses (of length 28) are used to train a 2-layer perceptron (256 input units, 50 hidden units, and 28 outputs). After the network is trained, a desired target frequency response (see Figure 4 (a)) is tested to obtain the corresponding 2-D surface (see Figure 4(b)). After the first inversion, the network creates the unconstrained 2-D surface as shown in Figure 4(c). This unconstrained surface is then reshaped by an locally excited Hopfield model and results in a new constraint satisfied 2-D surface (see Figure 4(d)) and its associated frequency response (see Figure 4(e)). This new surface is used as the initial point for the next inversion, the resulting inverted surface gives us the desired surface as shown in Figure 4(b). Therefore, convergence occurred in only two iterations.

4 Generalization and Illustration

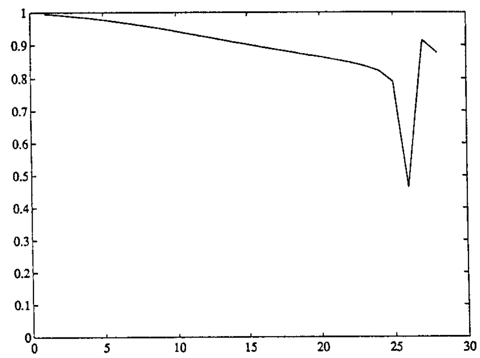
We can conceptually generalize the FSS design procedure to solution of more general constrained inverse problems. Let $\vec{a}(L)$ denote the response of the trained neural network to an input of $\vec{a}(0)$. We can abstract the operations in Equations 1 by $\vec{a}(L) = \mathcal{NN}\vec{a}(0)$ where $\mathcal{NN}$ is the neural net operator. For a given response, $\vec{a}(L) = \vec{\lambda}$, define the set

$$\Lambda = \{\vec{\alpha} \mid \mathcal{NN}\vec{\alpha} = \vec{\lambda}\}. \tag{4}$$

In other words, Λ is the set of all inputs that yield $\vec{\lambda}$ as the net's output. The neural network inversion algorithm finds an element of Λ that is close, in some sense, to the inversion's initialization. For two

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)



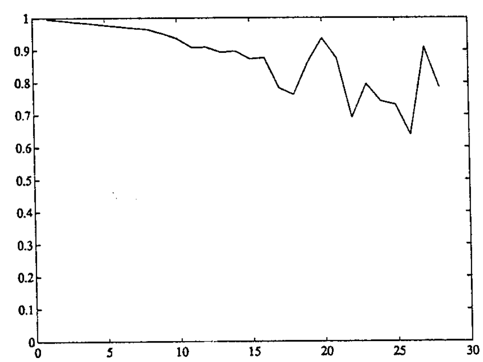
(b)

0.02	0.02	0.02	0.02	0.03	0.02	0.02	0.97	0.02	0.02	0.02	0.02	0.97	0.98	0.03	0.97
0.02	0.01	0.01	0.20	0.05	0.21	0.11	0.01	0.01	0.25	0.18	0.60	0.01	0.01	0.32	0.01
0.98	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.09	0.00	0.00	0.01	0.01	0.01	0.01	0.98
0.02	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.48	0.00	0.00	0.00	0.00	0.00	0.97
0.97	0.02	0.01	0.03	0.01	0.00	0.00	0.01	0.01	0.41	0.00	0.00	0.01	0.01	0.01	0.02
0.98	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.02	0.00	0.01	0.04	0.03	0.01	0.97
0.98	0.01	0.04	0.02	0.00	0.81	0.99	0.67	0.02	0.99	0.63	0.00	0.01	0.03	0.01	0.97
0.03	0.00	0.99	0.26	0.01	0.05	0.98	0.99	0.99	0.98	0.57	0.97	0.40	0.92	0.00	0.01
0.97	0.00	0.02	0.99	0.32	0.01	0.94	0.99	0.02	0.48	0.01	0.99	0.03	0.02	0.01	0.98
0.97	0.06	0.03	0.27	0.00	0.01	0.99	0.01	0.47	0.99	0.01	0.00	0.12	0.00	0.11	0.01
0.97	0.00	0.00	0.98	0.69	0.00	0.01	0.00	0.00	0.98	0.00	0.88	0.98	0.02	0.08	0.02
0.02	0.00	0.02	0.06	0.05	0.00	0.01	0.00	0.00	0.10	0.00	0.00	0.99	0.01	0.01	0.03
0.98	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.16	0.01	0.00	0.00	0.00	0.00	0.98
0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.08	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.98
0.02	0.01	0.23	0.86	0.77	0.01	0.15	0.21	0.01	0.01	0.61	0.01	0.01	0.76	0.46	0.02
0.01	0.97	0.02	0.02	0.02	0.02	0.02	0.02	0.97	0.98	0.02	0.02	0.03	0.02	0.02	0.02

(c)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(d)



(e)

Figure 4: (a) A desired frequency response of an FSS. (b) The corresponding 2-D surface. (c) The first inverted unconstrained surface. (d) The constraints satisfied surface of (c). (e) The frequency response of the surface in (d).

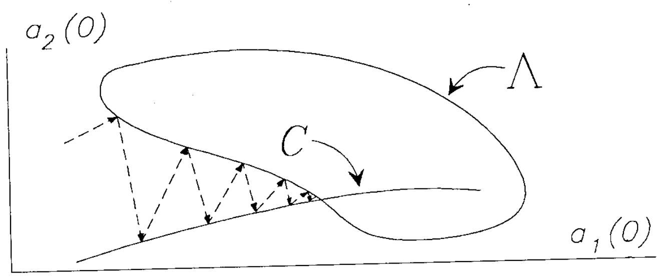


Figure 5: By iteratively inverting the net to find an element in Λ, followed by a projection onto the constraint set, C, we can approach the solution to the constrained inversion problem.

inputs and one output, an example of the set Λ is shown in Figure 5.

In addition, we have an input constraint set, C. For the FSS problem, this space is the set of clustered ones. Once the net is inverted, we project onto that space. For the FSS problem, this was done by the locally connected Hopfield neural net. This projection is then used as the initialization for the next inversion and the process is repeated. As is shown in Figure 5, repeated inversion and projection can result in a solution satisfying both the inversion criterion and the constraint set.

We note, of course, that convergence, in general, cannot be guaranteed. Convergence has also never been proven in general, for example, in the Gerchberg-Saxton iterative algorithm or, for that matter, iterative back propagation training of a layered perceptron. Nevertheless, the general procedure for solution of constrained inverse problem solution using inverted neural networks is one that deserves further investigation in general and for the FSS problem specifically.

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition (Vol. 1)*, MIT Press, 1986.

[2] A. Linden and J. Kindermann. Inversion of multilayer nets. In *Proc. Int'l Joint Conf. on Neural Networks*, II 425-430, Washington D.C., June 1989.

[3] J. N. Hwang, S. Oh, J. J. Choi, and R. J. Marks II Classification boundaries and gradients of trained multilayer perceptrons. To appear in *Proc. Int'l Symposium on Circuits and Systems*, New Orleans, Louisiana, 1990.

[4] R. Mittra, C. H. Chan, and T. Cwik. Techniques for analyzing frequency selective surfaces – a review. *Proceedings IEEE*, Vol. 72, No. 12, pp. 1593-1615, December 1988.

[5] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. SCi. USA*, 81:3088–3092, 1984.

[6] H. Stark, editor *Image Recovery: Theory and Application* Academic Press, Orlando, 1987

[7] M.H. Goldburg and R.J. Marks II Signal synthesis in the presence of an inconsistent set of constraints, *IEEE Transactions on Circuits and Systems*, vol. CAS-32 pp. 647-663 (1985).