ELSEVIER

Thin Solid Films 370 (2000) 122–127

thin solid films

www.elsevier.com/locate/tsf

# Use of artificial neural networks to predict thickness and optical constants of thin films from reflectance data

Milad F. Tabet*, William A. McGahan

*Nanometrics Inc., 310 De Guigne Drive, Sunnyvale, CA 94086-3906, USA*

Received 29 September 1999; received in revised form 14 March 2000; accepted 21 March 2000

**Abstract**

Artificial neural networks and the Levenberg–Marquardt algorithm are combined to calculate the thickness and refractive index of thin films from spectroscopic reflectometry data. Two examples will be discussed, the first is a measurement of thickness and index of transparent films on silicon, and the second is a measurement of three thicknesses and index of poly-silicon in a rough poly-silicon on oxide stack. A neural network is a set of simple, highly interconnected processing elements imitating the activity of the brain, which are capable of learning information presented to them. Reflectometry has been used by the semiconductor industry to measure thin film thickness for decades. Modeling the optical constants of a film in the visible region with a Cauchy dispersion model allows the determination of both thickness and refractive index of most transparent thin films from reflectance data. The use of an alloy interpolation model for the optical constants of poly-silicon allows the determination of thicknesses and poly optical constants. In this work artificial neural networks are used to obtain good initial estimates for thickness and dispersion model parameters, these estimates are then used as the starting point for the Levenberg–Marquardt algorithm which converges to the final solution in a few iterations. These measurement programs were implemented on a Nanometrics NanoSpec 8000XSE. 

*Keywords*: Thin films; Reflection spectroscopy; Neural networks; Optical properties

## 1. Introduction

Spectroscopic reflectometry is widely used in the semiconductor, flat panel, and thin film magnetic head industries to measure film thickness. Absolute reflectometry measures the ratio of the intensity of the reflected light to the intensity of the incident light. This measurement is usually done at normal incidence at a wavelength range over the visible and/or ultraviolet regions. The technique is nondestructive which makes it ideal for production wafer monitoring. In this work the absolute reflectance data is measured using a patented method [1] based on the use of a bare crystalline silicon reference wafer. Fig. 1 shows a diagram of the NanoSpec 8000XSE optical system, which is used in this work. Thin film thickness is calculated by minimizing the differences between measured and model-generated reflectance data, if optical constants of the film and substrate are known. The Levenberg–Marquardt algorithm is used to minimize the mean square error (MSE) by smoothly interpolating between the gradient and inverse Hessian methods [2]. The MSE is a measure of the quality of the match between measured and model calculated data.

Most dielectric films are transparent in the visible region so that a simple Cauchy dispersion model using the standard equation can describe the index of refraction of these materials

$$n(\lambda) = An + \frac{Bn}{\lambda^2} + \frac{Cn}{\lambda^4} \quad (1)$$

$\lambda$ is the wavelength of the light, *An*, *Bn* and *Cn* are the Cauchy parameters. Modeling the optical constants of a film with a Cauchy dispersion model allows the determination of both thickness and refractive index of most transparent thin films from reflectance data. In this work we are using artificial neural networks (ANN) to obtain a good estimate of thickness and parameters *An* and *Bn*. These estimates then become the starting point of the Levenberg–Marquardt regression algorithm which does a few iterations to reach the final solution. Measurement of thickness and index are normally performed with a material specific program, so that a silicon dioxide film is measured with one program and a nitride film with another. In this work all transparent films with refractive index ranging

* Corresponding author. Tel.: +1-408-746-1600; fax: +1-408-720-0916.
*E-mail address:* mtabet@nanometrics.com (M.F. Tabet).

0040-6090/00/$ - see front matter 
PII: S0040-6090(00)00952-4

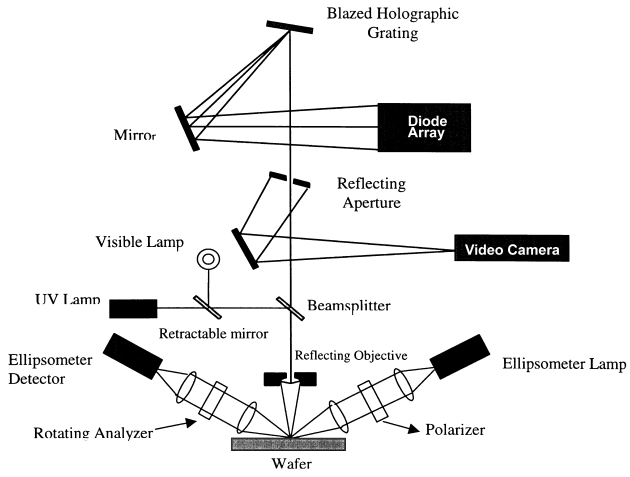


Fig. 1. Diagram of the NanoSpec 8000XSE optical system.

from around 1.4 to 2.35 will be measured with one measurement program.

The second example is a three-layer stack of rough poly-silicon on oxide on silicon. An alloy interpolation model was used for the poly-silicon optical constants, and neural networks were employed to obtain a good estimate of roughness, poly-, and oxide thickness and poly-silicon crystalline fraction. These estimates then became the starting point of the Levenberg–Marquardt regression algorithm, which does a few iterations to reach the final solution. The alloy interpolation model describes the optical constants of a material as it varies with a controllable physical parameter, for instance, the annealing temperature of poly-silicon, the Ge concentration of SiGe [3]. This model has one parameter called alloy fraction, and in this case it is the crystalline fraction of the poly-silicon. An alloy fraction of zero results in the optical constants of amorphous silicon and a fraction of one gives the optical constants of crystalline silicon. Adjusting this one parameter can produce the optical constants of poly-Si with any crystalline fraction. This measurement on the NanoSpec 8000XSE can fit for this physical parameter providing means of monitoring the crystallinity of poly-silicon thin films.

## 2. Artificial neural networks

Neural networks are used mainly in areas involving pattern mapping and pattern classification, like visual images and speech recognition and other problems that are too complicated for traditional methods. This architecture, inspired by the structure of the human brain, is different from the computers used today. A neural network is a highly interconnected array of elementary processors, and as such is a massively parallel system. These elementary processors, called neurons, may do only one or, at most a few calculations, and are much simpler then CPUs. The ANN has more connections than processors and it is this tremendous number of interconnections that give it its power [4]. While there are a variety of different artificial neural network architectures in use, the one used here is the multilayer perceptron (MLP) type [5]. Typically, this type of network consists of three or more layers. Fig. 2 shows an example of an ANN, which has four input neurons, eight neurons in one hidden layer, and three output neurons. The connections between the neurons act as a link and pass a numeric value from one neuron to another. These values are weighted by a connection strength, which is adjusted during training to produce the final network. Each neuron, except for the input layer, receives a linearly weighted sum of all the outputs from the neurons of the previous layer, and propagates its activation to the neurons in the next layer.

When $h$ hidden layers exist, layer 0 is the input layer, and layer $(h + 1)$ is the output layer. The activation of a neuron $j$ in the layer $k$ is defined by

$$u_j^{(k)} = h_j\left(\sum_i w_{ij} u_i^{(k-1)}\right) \tag{2}$$

$i$ covers all the neurons in the layer $(k - 1)$, $u$ is the output of a neuron, and $w_{ij}$ are the weights between layer $i$ and $j$, where

$$h_j(x) = \frac{1}{(1 + \exp(-x))} \tag{3}$$

$h_j(x) = x$ if $j$ is an input neuron. The nonlinearity $h_j(x)$ does not necessarily have to be the sigmoid function $1/(1 + \exp(-x))$. Other monotonic functions that are differentiable in the domain of $x$ can be used.

The back-propagation learning algorithm is employed for training the network. This algorithm is classified as a supervised learning algorithm since it requires a target value for a given set of input parameters. The network is presented with a series of pattern pairs, an input pattern and a target output pattern. Upon each presentation, weights are adjusted to reduce the difference between the network output and the target output. The gradient descent algorithm is used to find the best estimates of the interconnected weights [4]. The back-propagation learning algorithm consists of a forward propagating step followed by a backward propagating step. The output of the network is compared to the target output and an error is calculated. Then the weights of the connections going into the output layer are adjusted. Next an error value for the output of the neurons in the hidden layer is calculated and the weights going into the hidden neurons are adjusted. The process ends when the last layer of weights is adjusted. The adjustment to the connecting weights is done as follows [5]

$$\Delta w_{ji} = \eta \delta_j a_i \tag{4}$$

The adjustment is proportional to $\delta_j$, which is the error value of the target neuron. A larger error out of a neuron means a larger adjustment to its incoming weights. The weight adjustment is also proportional to $a_i$, the output of the origi-

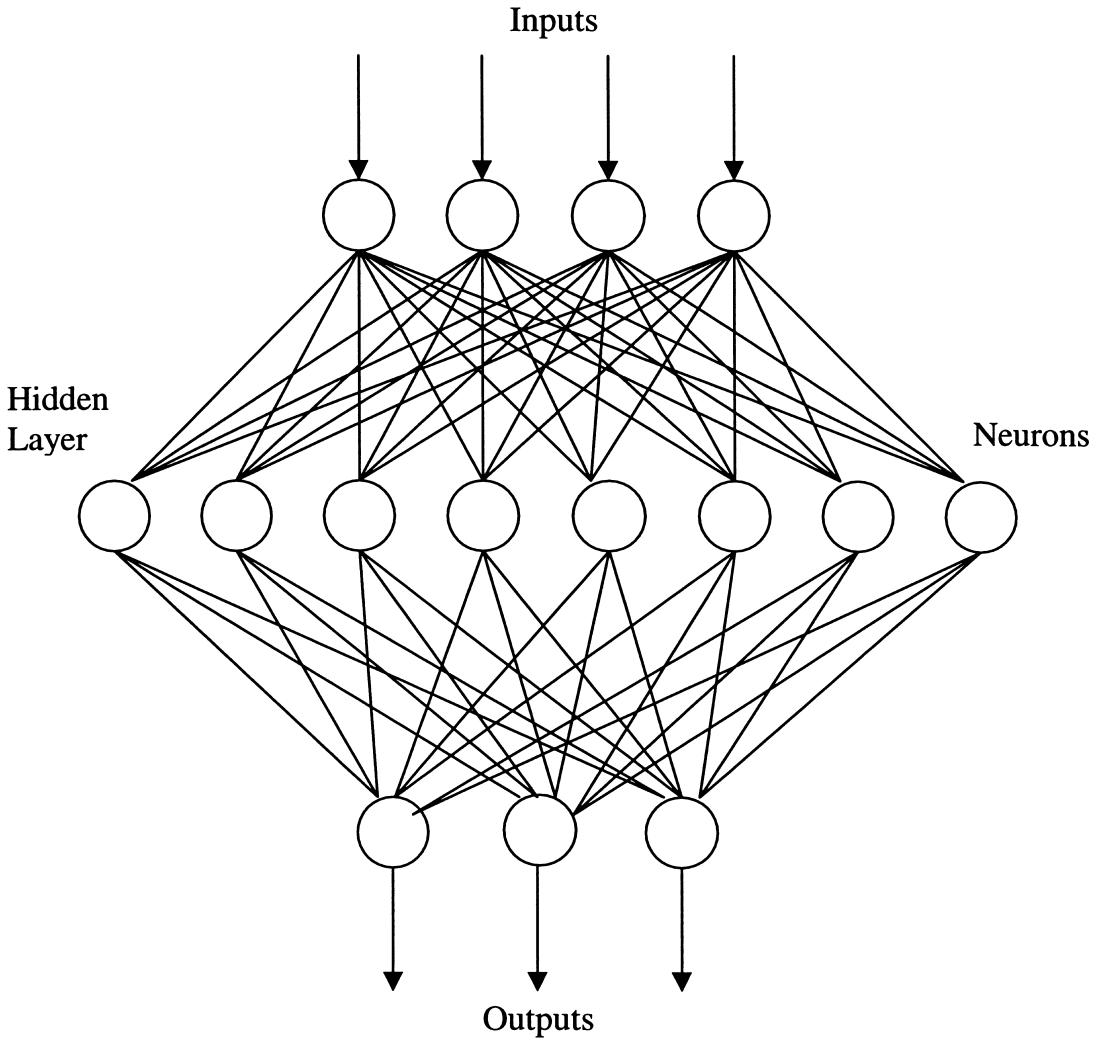


Fig. 2. MLP type ANN, with an architecture of four input, eight hidden and three output neurons.

nating neuron. The learning rate $\eta$ is chosen by the neural network user, and indicates the rate of learning of the network. Large values can cause instability in the network, and smaller values lead to slow learning; a variable learning rate is often used.

In order to apply the MLP type ANN to our problem we will use the reflectance equations to generate our training data set [6]. For the first example of transparent films on silicon the output ($O$) and input ($I$) patterns will be as follows

$$(O) = [R(\lambda)]$$

$$(I) = [t, An, Bn] \tag{5}$$

The process starts by creating $N$ data sets by applying different inputs ($t$,$An$,$Bn$) to the reflectance equations, and generating ($O_i$,$I_i$), for $i = 1, ..., N$. Then training the ANN with $O_i$ as input and $I_i$ as desired output. For our second example the output and input were as follows

$$(O) = [R(\lambda)]$$

$$(I) = [t_1, t_2, t_3, \text{alloy}] \tag{6}$$

$t_1$, $t_2$ and $t_3$ are the thickness of the oxide, poly- and roughness layers,, and alloy is the interpolation model parameter which is the poly-Si crystalline fraction. The process as before uses the reflectance equations with many different inputs to generate the training data. The neural network after a few hours or days of training is ready to predict thickness and model parameters based on the training data. The behavior and accuracy of the ANN may be improved by using additional training data, increasing the number of hidden layer neurons, or longer training. The trained ANN provides thickness, and model parameters in less than 1 ms. Commercially available neural network software was used for training and recall [7]. The training, which can take up to several days depending on training parameters, was performed on a 200-MHz Pentium computer.

## 3. Transparent films on silicon

The advantages of both the Levenberg–Marquardt and the ANN algorithms are obtained by combining them [8]. Raw reflectance data are processed first by the ANN which provides an output within a milliseconds, with a few percent error depending upon training. The ANN outputs become the initial estimates of the Levenberg–Marquardt algorithm, which performs a few iterations and further refines the accuracy of the solution.

The accuracy of the neural network improves when we limit the range of the output parameters [9]. For this application five networks with a 300-nm thickness range each were used. The total thickness range was 100–1600 nm. Cauchy parameter *An* was varied from 1.35 to 2.15, and *Bn* from 0.003 to 0.08. Five networks were trained each covering a specific thickness range and all *An* and *Bn* ranges. The wavelength range of the input reflectance data was from 400 to 800 nm with data every 10 nm, requiring a network with 41 input neurons. Experiments with many different numbers of hidden layer neurons were performed, ranging from 10 to 50 hidden neurons. Thirty hidden neurons gave the best results so this was used for the final networks.

When performing this measurement the collected reflectance data is scaled to 41 points (the number of neural network inputs) and then presented to the ANN. The Levenberg–Marquardt algorithm does five iterations using the output of each of the five networks. The number of iterations was limited to five at this point in order to keep the measurement time at a minimum. The network with the lowest MSE after five iterations is chosen and a regression fit is performed using the output of that network. The result of the last fit becomes the final output of the measurement.

Five hundred randomly generated data sets consisting of thickness, *An*, *Bn* and matching reflectance data were used to test the accuracy of the neural networks. Table 1 contains the results of the comparison between network output and actual output, actual output being the thickness, *An* and *Bn* that were used to generate the reflectance data, and would be a perfect result if achieved by the ANN. The average difference between network and actual output for *An* was 0.0352 with a 0.0344 standard deviation (SD), for *Bn* it was 0.0026 (SD 0.0026), and the average difference in thickness was 12.59 (SD 14.75) nm. Fig. 3 shows the actual versus the ANN outputs for *An*, for all 500 test points. The diagonal line in this graph shows a perfect match between actual and ANN output. Fig. 4 shows actual versus ANN output for thickness for all 500 test points covering the thickness range of 100–1600 nm.

Table 1
Results of the transparent films on silicon neural network testing[a]

| | *An* | *Bn* | Thickness (nm) |
|---|---|---|---|
| Average difference between actual and network output | 0.0352 | 0.0026 | 12.594 |
| Standard deviation | 0.0344 | 0.0026 | 14.749 |
| Minimum difference | 0.0001 | 0.0000 | 0.03530 |
| Maximum difference | 0.2799 | 0.0355 | 126.082 |

[a] Five hundred random test data were used.

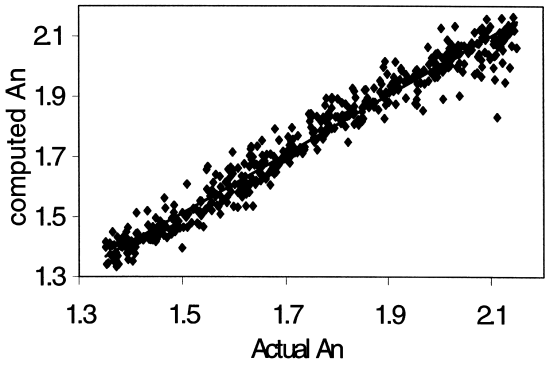


Fig. 3. Plot of actual versus ANN outputs of *An*, for 500 randomly generated reflectance data in the thickness, *An* and *Bn* ranges that were used to train the network.

Since the Levenberg–Marquardt algorithm requires a starting point near the solution in order to converge, one more test of the combined program was performed to determine if the ANN output is a good starting point and convergence is possible. The full measurement program was tested with 500 randomly generated reflectance data. Ninety-five percent of the cases reached the correct solution, meaning that one of the five networks gave an output that was close to the solution allowing the final regression to reach the correct result. In the cases where this neural network strategy fails, this measurement program performs a grid search covering all possible thicknesses and will output the solution with the lowest MSE.

Fig. 5 shows experimental reflectance data from a 600-nm silicon dioxide film on a silicon wafer, and reflectance data generated from a model using the neural network outputs for thickness, *An* and *Bn*. Fig. 6 shows the same data as in Fig. 5, after the final Levenberg–Marquardt fit.

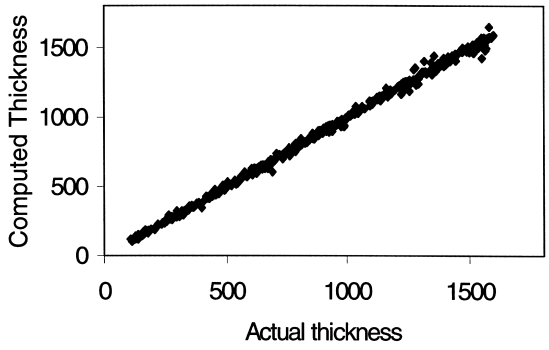


Fig. 4. Plot of actual versus ANN outputs of thickness, for 500 randomly generated reflectance data in the thickness, *An* and *Bn* ranges that were used to train the ANN (thickness in nm).

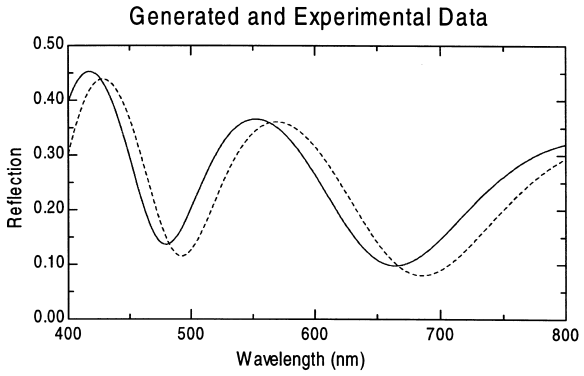


Fig. 5. Experimental and ANN model-generated reflectance data from a 6000-Å $SiO_2$ film on a Si wafer.

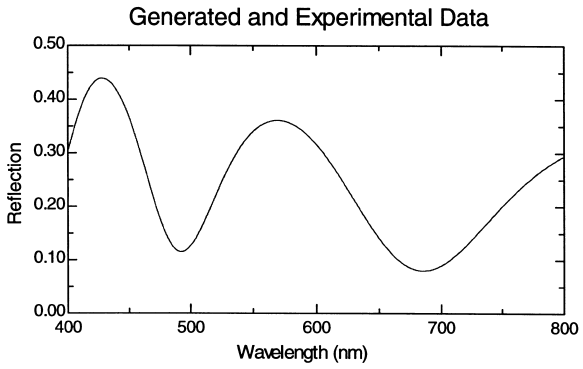


Fig. 6. Experimental and model-generated reflectance data from a 6000-Å $SiO_2$ film on a Si wafer after the final fit.

## 4. Rough poly-silicon on oxide on silicon

For this application again the raw reflectance data are processed first by the ANN which provides an output within a millisecond, with a few percent error depending upon training. The ANN outputs, which are three thicknesses and alloy fraction, then become the initial estimates of the Levenberg–Marquardt algorithm, which further refines the accuracy of the solution within a few iterations.

In order to improve the accuracy of the neural network we limited the range of the output parameters [9]. Four networks with an approximately 300 nm thickness range each for the poly thickness were used. The total poly thickness range was 50–1200 nm. The oxide thickness varied from 2.5 to 150 nm, the roughness from 0 to 15 nm, and alloy fraction from 0.6 to 1. The four networks were trained, each covering a specific poly thickness range and all other ranges. The wavelength range of the input reflectance data was from 400 to 800 nm with data every 10 nm, requiring a network with 41 input neurons. Experiments were conducted with many networks with a different number of hidden layer neurons; 40 was optimal and it was used for the final networks.

When performing this measurement the collected reflectance data are scaled to 41 points and then presented to the ANN. The Levenberg–Marquardt algorithm, fitting the poly-Si and roughness thicknesses only, does five iterations using the output of each of the four networks. The network with the lowest MSE after five iterations is chosen and a regression fitting all parameters is performed using the output of that network. The result of the last fit becomes the final output of the measurement.

Five hundred randomly generated data sets consisting of the three thicknesses, alloy fraction and matching reflectance data were used to test the accuracy of the neural networks. Table 2 contains the results of the comparison between network output and actual output. The average difference between network and actual output for the oxide thickness was 6.12 (SD 5.44) nm, for poly thickness it was 13.39 (SD 18.61) nm, and the average difference for the roughness thickness was 0.65 (SD 0.78) nm. The alloy fraction average difference was 0.0103 (SD 0.0122). An additional test of the combined program was performed to determine if the ANN output is a good starting point for the Levenberg–Marquardt algorithm and convergence is possible. The full measurement program was tested with 500 reflectance data generated from random thicknesses and alloy fraction in the same range as the ANN training data. Ninety-one percent of the cases reached the correct solution. In the cases where the neural network strategy fails, this measurement program performs a grid search covering the thickness range of all three layers and alloy fraction, and will output the solution with the lowest MSE.

Fig. 7 shows experimental reflectance data from a 500-nm poly-Si film on 10-nm $SiO_2$ film on a silicon wafer with 3 nm of surface roughness on top, and reflectance data generated from a model using the neural network outputs for thicknesses, and alloy fraction. Fig. 8 shows the same data as in Fig. 7, after the final Levenberg–Marquardt fit.

Table 2
Results of the rough poly on oxide on silicon neural network testing[a]

| | Oxide (nm) | Poly-Si (nm) | Roughness (nm) | Alloy fraction |
|---|---|---|---|---|
| Average difference between actual and network output | 6.120845 | 13.38546 | 0.654854 | 0.0103 |
| Standard deviation | 5.438105 | 18.61172 | 0.78207 | 0.012181 |
| Minimum difference | 0.017563 | 0.010071 | 0.00124 | 7E-06 |
| Maximum difference | 47.27914 | 117.6812 | 6.105217 | 0.126597 |

[a] Five hundred random test data were used.

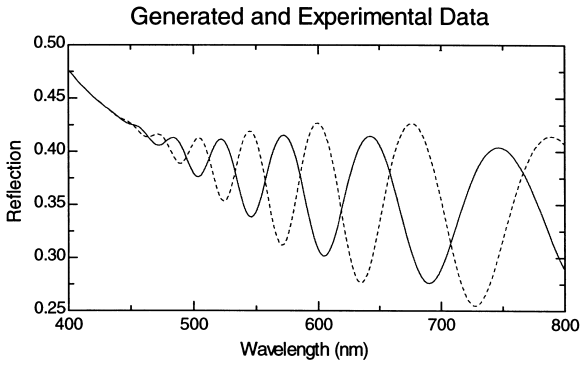


Fig. 7. Experimental and ANN model-generated reflectance data from a 500-nm poly-Si film on 10-nm $SiO_2$ on a Si wafer with 3 nm of top surface roughness.

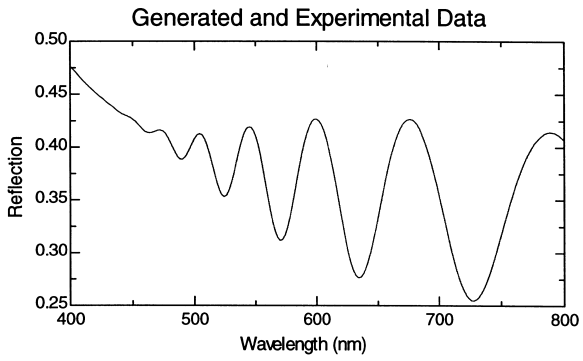


Fig. 8. Experimental and model-generated reflectance data from a 500 nm poly-Si film on 10-nm $SiO_2$ film on a Si wafer with 3 nm of surface roughness after the final fit.

## 5. Conclusion

The thickness and refractive index of most transparent thin films, with index ranging from around 1.4 to 2.35 at 633 nm, and thickness ranging from 100 to 1600 nm, were measured with a single measurement program. All three thicknesses of a rough poly-Si on oxide stack along with the poly-Si crystalline fraction were measured without any prior knowledge. These measurement programs were created on a Nanometrics NanoSpec 8000XSE thin film thickness analyzer and will acquire visible reflectance data, use a trained neural network to obtain initial estimates for thicknesses and model parameters, and refine the solution with a final Levenberg–Marquardt fit. The output of the programs is film thickness and refractive index at any desired wavelength in the visible region.

## References

[1] V.J. Coates, US Patent 5,045,704 (1991).
[2] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge University Press, Cambridge, 1986.
[3] W.A. McGahan, B.R. Spady, B.D. Johs, O. Laparra, Proc. SPIE 2725 (1996) 450.
[4] J.E. Dayhoff, Neural Network Architectures: An Introduction, Van Nostrand Reinhold, New York, 1990.
[5] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Parrallel Distributed Processing, Vol. 2, MIT Press, Cambridge, MA, 1987.
[6] F.K. Urban, M.F. Tabet, J. Vac. Sci. Technol. 12 (4) (1994) 1952.
[7] Qnet, Qnet 97 Neural Network Modeling for Windows, Qnet, 1997.
[8] F.K. Urban, D.C. Park, M.F. Tabet, Thin Solid Films 220 (1992) 247.
[9] F.K. Urban, D. Barton, N.I. Boudani, Thin Solid Films 332 (1998) 50.