

Technical Report

**One-Dimensional Inverse Scattering:  
Reconstruction of Permittivity Profiles for  
Stratified Dielectric Media**

Jason Agron and S. Prasad Gogineni

ITTC-FY2004-TR-27640-04

August 2003

Project Sponsors:  
The National Science Foundation  
under grant OPP-0122520; and  
NASA under grants NAG5-12659 and NAG5-12980

***Abstract:***

The goal of this research is to find a method for the reconstruction of the one-dimensional dielectric permittivity profile for homogenous layers of dielectric media. This method uses a layer-stripping technique to successively solve for the reflection coefficients between dielectric interfaces. As the reflection coefficient for each interface is found, the permittivity values for the dielectric layers that comprise the interface can be solved for iteratively. This method is quite fast, and it provides an excellent approximate reconstruction of the permittivity profile without the use of any *a priori* information concerning the layered media.

## Introduction

Forward modeling of electromagnetic phenomena, in which physical properties are transformed into the electromagnetic realm, has been done for over 100 years. In fact, most equations in electromagnetics serve to derive the electric and magnetic fields that result from measured physical data.

Only recently has some of the focus turned to the idea of inverse modeling of electromagnetic phenomena, a process in which electromagnetic data are translated back to the physical realm. Inverse modeling of data, or data inversion, proves to be very useful for the non-destructive characterization of layered media, especially subsurface layers, because in this way the physical characteristics of the subsurface layers can be measured without having to actually see, touch, or examine the layers themselves.

The difficulty with inverting electromagnetic data is due to the inherent complexity of the laws of electromagnetics themselves. These laws, when put into mathematical equations, are in the form of partial differential equations. Partial differential equations, or PDEs, reside in a field of mathematics that has not yet been fully mastered or understood. In other words, solving PDEs can be extremely difficult in almost all but the most simplified cases.

Many of the methods for solving inverse scattering problems revolve around the non-linear Ricatti PDE (1) found below:

$$\frac{dr(k_i, x)}{dx} = 2jk_i \sqrt{\epsilon(x)} r(k_i, x) + \frac{1 - r^2(k_i, x)}{4\epsilon(x)} \frac{d\epsilon(x)}{dx}. \quad (1)$$

Where:

$r(k_i, x)$  = Reflection Coefficient as a function of depth (x) and wave number ( $k_i$ ).  
 $\epsilon(x)$  = Permittivity Profile as a function of depth (x).

The main problem of interest in inverse scattering is to calculate the permittivity profile,  $\epsilon(x)$ , from the total reflection coefficient,  $r(k_i, 0)$ .

Due to the non-linearity of the Ricatti equation (1), exact and unique inverse solutions are almost impossible to calculate. However, many approximate inverse solutions, formed from iterative non-linear optimization schemes, have been shown to be quite successful (such as Born, Rytov, and Newton solutions) (Hopcraft and Smith, 1992; Mikhnev, 2003; Mikhnev and Vainikainen, 2000). The optimization methods often require multiple iterations to develop an adequate approximation, and the number of iterations often grows with the complexity of the permittivity profile (Mikhnev and Vainikainen, 2000).

The goal of this research is to develop an approximation method that does not require optimization techniques or multiple iterations to develop an adequate approximation of the actual permittivity profile of the layered media.

Instead of optimization techniques, the method uses an approximate layer-stripping technique that isolates the reflections between dielectric layers, in order to simplify the

inversion process to a simple mathematical equation that relates the permittivity of two dielectric layers to the reflection coefficient found at the dielectric interface.

## Theory

### Scattering Matrix Theory:

The equations that govern reflectivity from a layered model are quite easy to solve when only two to three layers are present. However, for a multi-layer model, these equations become far too complex, so numerical optimization methods are often used to solve the problem. A remedy for this problem is to represent the multi-layer media as a stack of homogenous, isotropic layers with varying thickness and permittivity. This stack of layers is surrounded by two semi-infinite media: ambient, the air above the stack, and the substrate, the final layer below the stack of layers (Fig. 1).

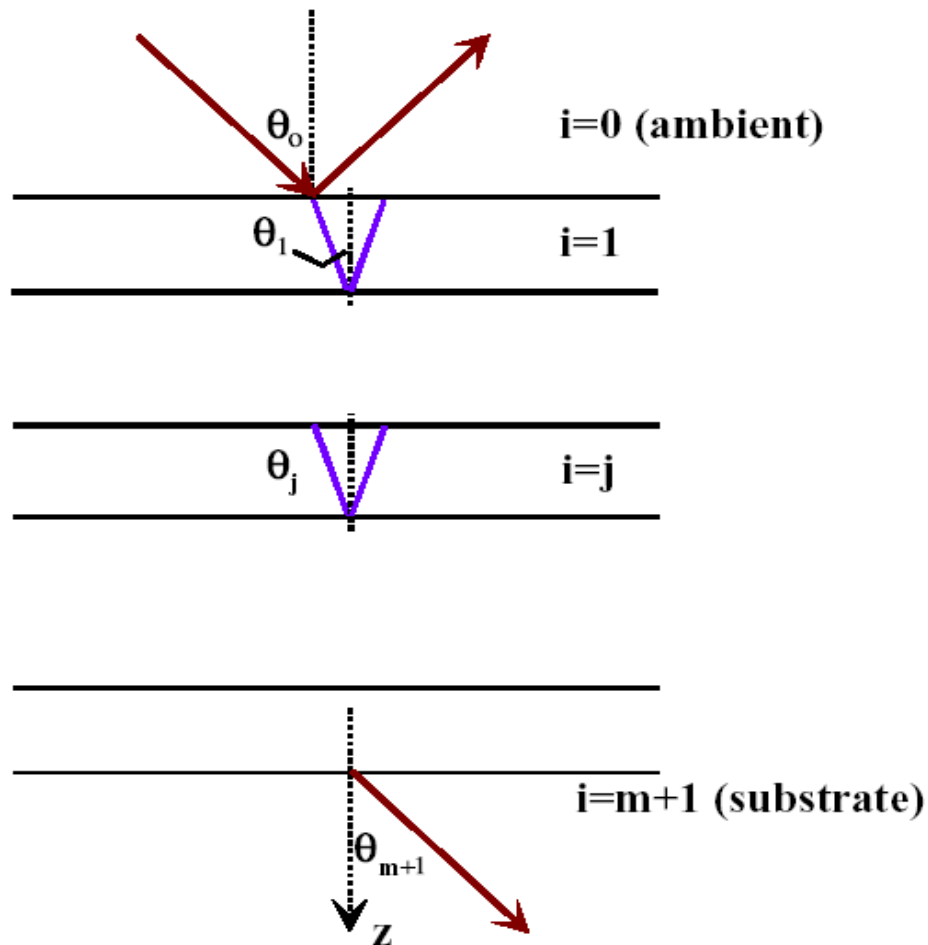


Figure 1: Layer model for  $m$  layers with ambient and substrate

With the assumptions presented above about the layered media, a scattering matrix can be used to represent the multi-layer model. The scattering matrix,  $S$ , relates the

electromagnetic waves incident on a multi-layer stack to the electromagnetic waves reflected from the stack (“Reflection and Transmission,” 1999).

The scattering matrix,  $S$ , for an  $m$ -layer stack, where layer 0 is the ambient, and layer  $m$  is the substrate, is described by the following sets of matrix equations (Mikhnev and Vainikainen, 2000):

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = I_{01} L_1 I_{12} L_2 I_{23} \dots L_{(m-1)} I_{(m-1)m} . \quad (2)$$

The scattering matrix,  $S$ , describes the propagation of electromagnetic waves through each individual interface and layer of the medium.

The interface matrix,  $I_{i(i+1)}$ ,

$$I_{i(i+1)} = \frac{1}{1 + r_{i(i+1)}} \begin{pmatrix} 1 & r_{i(i+1)} \\ r_{i(i+1)} & 1 \end{pmatrix}. \quad (3)$$

is the matrix that represents the interface between two adjacent layers of number  $i$  and  $i+1$ . The interface matrix,  $I_{i(i+1)}$ , describes the boundary conditions when an electromagnetic wave passes from layer number  $i$  to layer number  $i+1$ .

Next, the layer matrix,  $L_i$ , is the matrix that represents the homogenous layer  $i$ .

$$L_i = \begin{pmatrix} e^{j\beta_i} & 0 \\ 0 & e^{j\beta_i} \end{pmatrix}. \quad (4)$$

The layer matrix,  $L_i$ , describes the propagation of an electromagnetic wave as it travels through isotropic layer number  $i$ .

Also, the interface reflection coefficient,  $r_{i(i+1)}$ ,

$$\begin{aligned} r_{i(i+1)} &= \frac{\sqrt{q_i} - \sqrt{q_{i+1}}}{\sqrt{q_i} + \sqrt{q_{i+1}}}; \\ \beta_i &= kd_i \sqrt{q_i}; \\ q_i &= \varepsilon_i - j\eta_0 \sigma_i / k \end{aligned} \quad (5)$$

is the reflection coefficient at the interface between adjacent layers with numbers  $i$  and  $i+1$  where:

- $\varepsilon_i$  : permittivity of layer number  $i$ .
- $\sigma_i$  : conductivity of layer number  $i$ .
- $d_i$  : thickness of layer number  $i$ .

Also, the ambient medium is assumed to be air, with  $q_0 = 1$ , and the substrate is assumed to be the last layer in the multi-layer stack.

The scattering matrix (2),  $S$ , describes the relationship between waves in layer number  $0$  (ambient), and waves in layer number  $m$  (substrate) by the following equation:

$$\begin{bmatrix} V_0^+ \\ V_0^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_m^+ \\ V_m^- \end{bmatrix}. \quad (6)$$

However, the substrate (layer  $m$ ), is semi-infinite, therefore  $V_m^- = 0$ . This leads to the following equations by matrix multiplication:

$$\begin{aligned} V_0^+ &= S_{11}V_m^+ \\ V_0^- &= S_{21}V_m^+. \end{aligned} \quad (7)$$

Using the equations found above, the total reflection coefficient, or ratio between reflected wave and incident wave, is calculated from (7) as:

$$R = V_0^- / V_0^+ = S_{21} / S_{11}. \quad (8)$$

One can see that if the interface reflection coefficients,  $r_{i(i+1)}$ , could be found from the total reflection coefficient,  $R$ , then the permittivity and depth of each layer could be found quite easily from the equations describing the interface reflection coefficient (5).

### Layer Stripping Theory:

The total reflection coefficient,  $R$ , can also be thought of as the frequency response of the multi-layer stack. If the Inverse Fourier Transform of the total reflection coefficient is taken, then the result is the impulse response of the multi-layer stack. By causality, reflections will not occur until a dielectric interface has been encountered by an electromagnetic wave; therefore, the locations of the reflection peaks in the impulse response of the multi-layer stack correspond to the times where an electromagnetic wave encounters a dielectric interface in the stack.

If the effects of each reflection could be isolated and transformed back into the frequency domain, then the interface reflection coefficients,  $r_{i(i+1)}$ , would be known for each dielectric interface. However, because isolation of a reflection requires truncation and estimation of the actual signal, ripples will be introduced into the interface reflection coefficients (frequency domain) by means of the Gibbs Phenomenon. In order to reduce the effects of the ripples introduced by the Gibbs Phenomenon, the interface reflection coefficients (frequency response) for a single interface are averaged between the frequencies of radar interest; then a new relation between reflection coefficient and permittivity values is apparent (9):

$$\pm \text{mean}(|r_{i(i+1)}|) \approx \frac{\sqrt{\varepsilon_i} - \sqrt{\varepsilon_{i+1}}}{\sqrt{\varepsilon_i} + \sqrt{\varepsilon_{i+1}}}. \quad (9)$$

\*\* Note:  $\pm$  corresponds to the sign of the interface reflection peak in the impulse response form. \*\*

Where:

$r_{i(i+1)}$  : reflection coefficients of isolated interface.  
 $\varepsilon_x$  : relative permittivity of layer number x.

One can see that if the reflection peaks can be found in the impulse response of the multi-layer stack, then the layers can be successively stripped away from the total reflection coefficients, and the permittivity of each layer can be solved for by successive solutions of (9).

Once the permittivity of each layer is found, the speed of an electromagnetic wave through each isotropic layer can be found by (10).

$$c_x = \frac{1}{\sqrt{\varepsilon_x \varepsilon_0 \mu_0}}. \quad (10)$$

Where:

$c_x$  : velocity of electromagnetic wave in layer number x [m/s]  
 $\varepsilon_x$  : relative permittivity of layer number x  
 $\varepsilon_0$  : permittivity of free space [F/m]  
 $\mu_0$  : permeability of free space [H/m]

Next, the depth of each layer can be calculated by associating the speed of an electromagnetic wave in each layer with the time of interface reflection peaks in the impulse response.

By combining the layer-stripping technique with a modified scattering matrix equation (9), the permittivity of each layer can easily be approximated. Furthermore, the depth of each layer can be found by using the speed of an electromagnetic wave in each layer (10) along with the time of interface reflection peaks in the impulse response of the multi-layer stack. This data inversion process does not require any optimization processes or multiple iterations to complete. It works by successive stripping of the layers, while iteratively solving for the permittivity and depth of each layer.

### Data Inversion Process

The data inversion technique described above is implemented in conjunction with the RSL MATLAB Radar Simulation Package written by Vijaya Ramasami (2003). This simulation package computes the forward modeling of radar returns from layered surfaces, while the data inversion functions invert the forward modeled data to calculate the permittivity and depth of each isotropic layer.



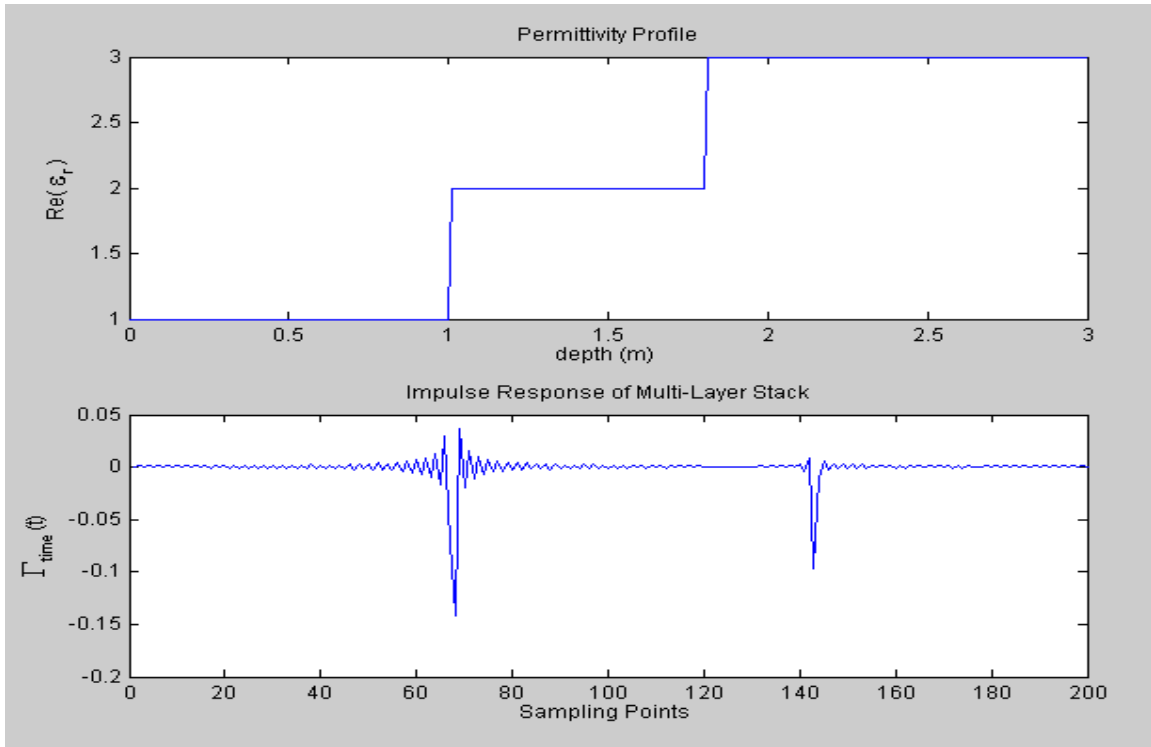


Figure 3: Negative Polarity Interfaces.

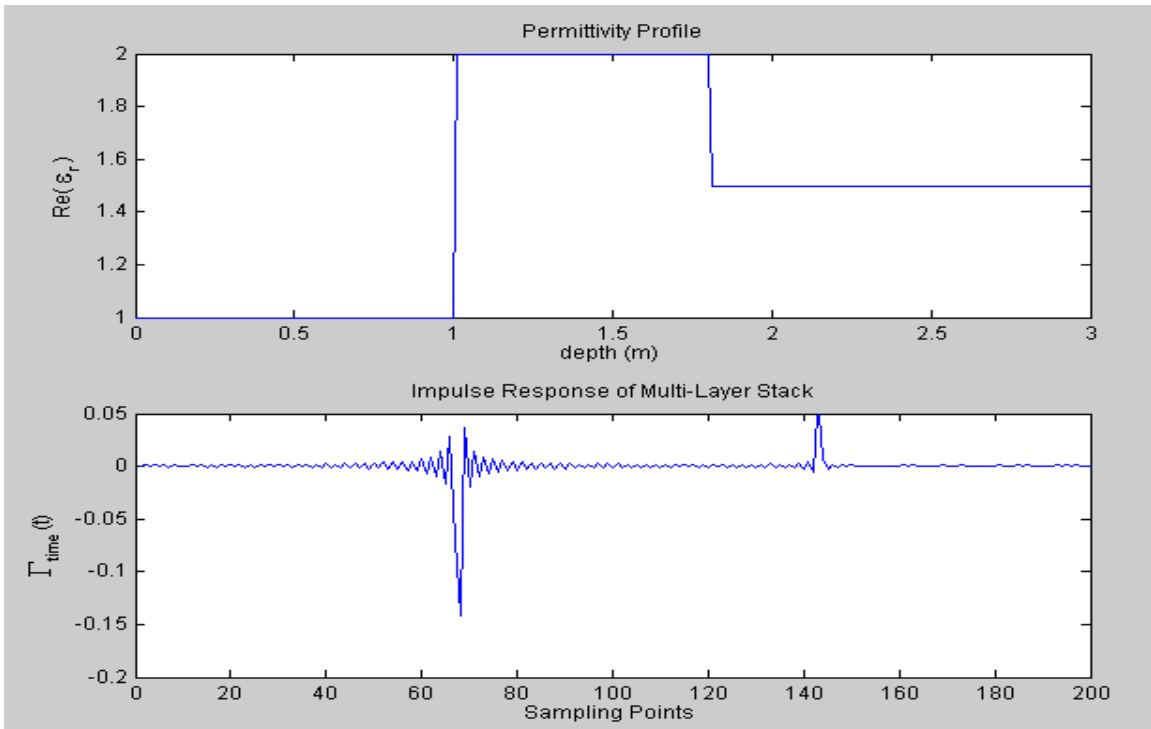


Figure 4: Positive and Negative Polarity Interfaces.

Once the total reflection coefficients are found, the locations of the interface reflections are found using a function called *findInterfaceIndeces.m*. This function finds the maxima

in  $\Gamma_{\text{time}}(t)$  by taking the derivative with respect to time, and locating points at which the slope goes from positive to negative.

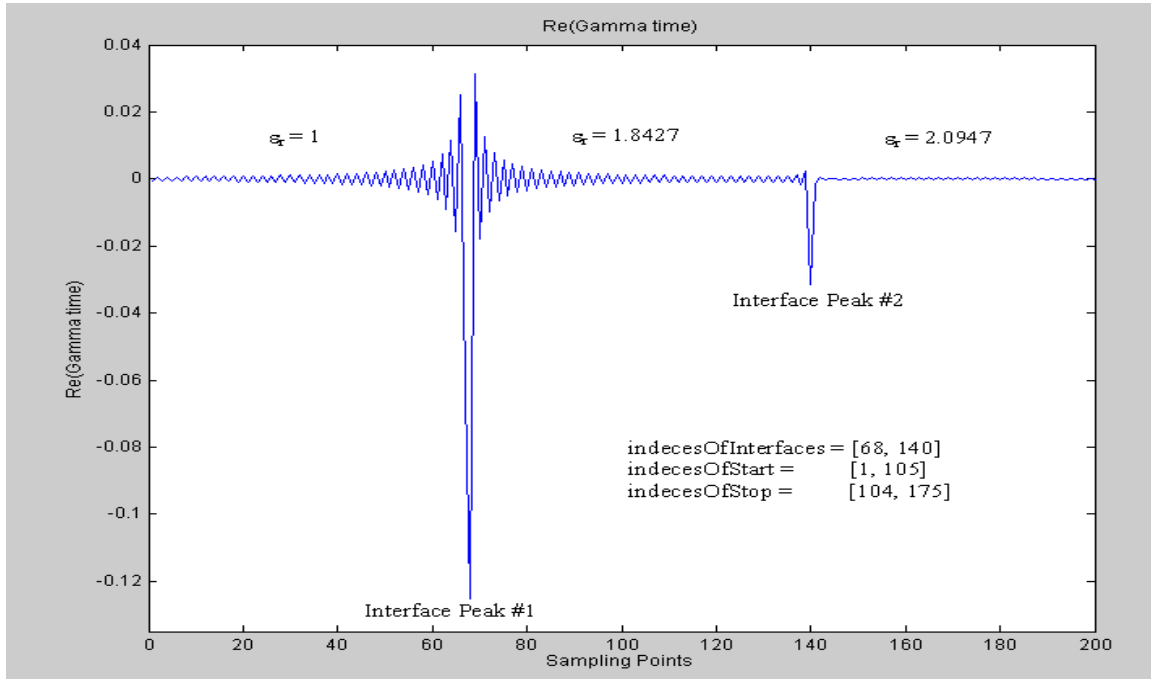


Figure 5: Example  $\Gamma_{\text{time}}(t)$  of 3-layer medium with results from *findIndecesOfInterfaces.m*.

Next, the permittivity values of each layer are calculated by a function called *erCalcFromLayerSubtraction.m*. This function successively strips reflections from subsequent layers away from the total reflection coefficient, and calculates the permittivity using a modified interface reflection equation (9).

Each reflection is “stripped” by separating the reflection in the time domain by using the indices from *indicesOfStart* and *indicesOfStop* provided by *findInterfaceIndeces.m*. Next, the reflection is padded with zeros to make the reflection the same length as  $\Gamma_{\text{time}}(t)$ . Then the FFT of the reflection is taken and it is averaged between the frequencies of radar interest found in the *radarParams* vector. This mean value is the average reflection coefficient for the reflection, and is used in the modified interface reflection equation (9). Finally the reflection is subtracted from the total reflection coefficient in the time domain, and the process is repeated until all average reflection coefficients have been calculated. The final step in this function is to calculate the permittivity of each layer by using the average reflection coefficients and the modified interface reflection equation (9).

Once the permittivity values of all detected layers are calculated, a distance profile can be calculated by a function called *calcDistanceProfile.m*. This function calculates the speed of an electromagnetic wave in each layer using (10), and then generates a distance profile based on these calculated speeds and the times of interface reflections found in the impulse response of the multi-layer stack. The distance profile reveals the thickness and depth of each layer in the multi-layer stack.

Next, continuous permittivity profiles can be created for plotting purposes with a function called *calcPermittivityProfiles.m*. This function takes the actual (forward modeled) and calculated (inverted) permittivity values, and creates a continuous, step-like profile that can be plotted. This function allows the permittivity profiles of the actual and calculated data to be compared to each other in graphical form.

The data inversion process is described in the following diagram:

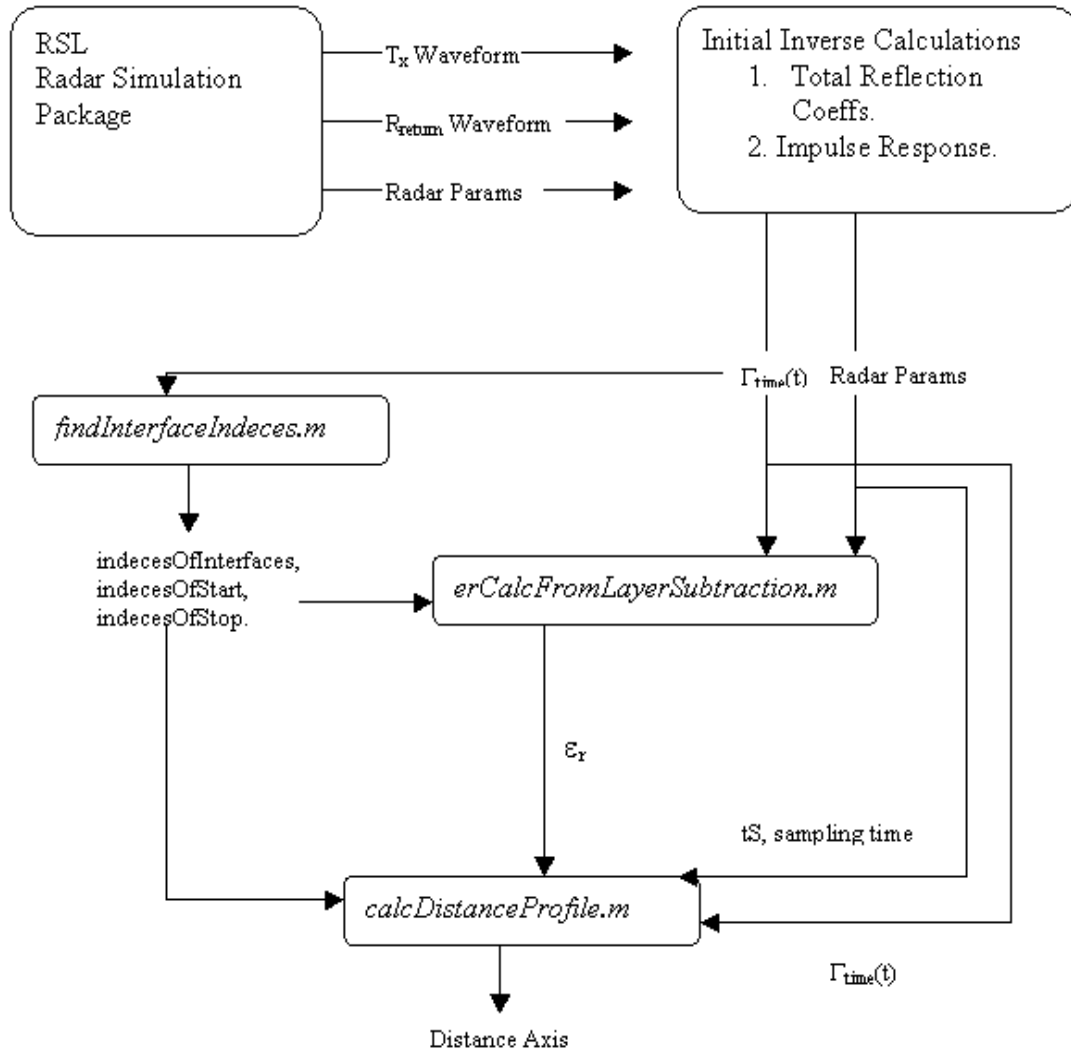


Figure 6: Block Diagram of Data Inversion.

## Data Inversion Example

The following example is contained in *fullTestInvert.m*. It contains the following multi-layer model:

Layer Number	Composition	Thickness (m)	Temperature (°C)	Density (kg/m <sup>3</sup> )	$\epsilon_r$
1 (Ambient)	Air	1.00	-1.0	n/a	1.0000
2	Ice	0.80	-10.0	600.0	1.8427
3	Ice	0.75	-12.0	550.0	1.5637
4 (Substrate)	Ice	0.70	-15.0	680.0	2.2345

Table 1: Geophysical Model For *fullTestInvert.m*.

$N$ (Number of Samples)	$T_s$ (Sampling Time)	Frequency Range
2000	1.00E-10	600-900 MHz

Table 2: Radar Parameters For *fullTestInvert.m*.

This profile is first forward modeled, and then the data inversion process begins with a calculation of the total reflection coefficient, which is then transformed into time domain by IFFT. The resulting impulse response,  $\Gamma_{\text{time}}(t)$ , is shown in the figure below.

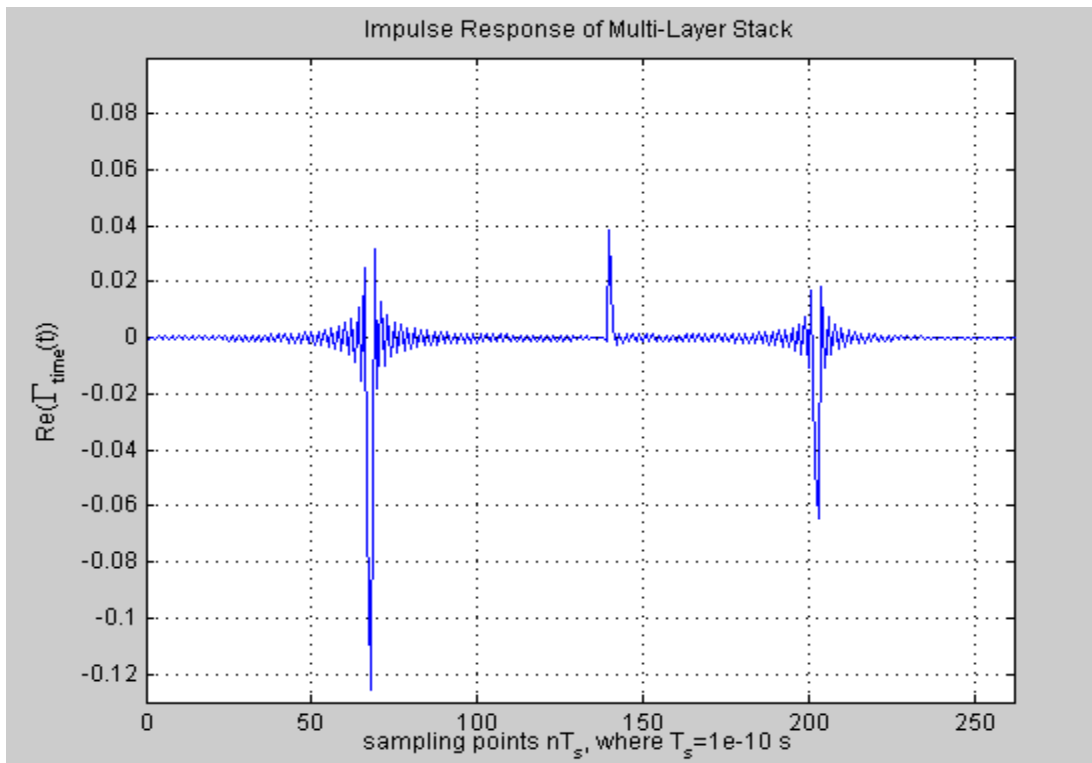


Figure 7: Impulse response of multi-layer stack

The large peaks in Figure 7 correspond to the dielectric interfaces in the multi-layer model. The model for *fullTestInvert.m* has four layers, therefore three interface

reflections will occur, and thus, there are only three major peaks in the impulse response of the multi-layer stack.

Next, the locations of dielectric interfaces are found by *findInterfaceIndeces.m*. The output of this function is the following:

```
indecesOfInterfaces = [ 68    140    203]
indecesOfStart      = [  1    105    173]
indecesOfStop       = [ 104    172    1999]
```

The next step in the data inversion process is to calculate the permittivity values for each layer in the model. The output from *erCalcFromLayerSubtraction.m* is the following:

```
erCalc = [ 1    1.8429    1.5699    2.2221]
```

Each separated reflection is plotted in both the frequency and time domains below:

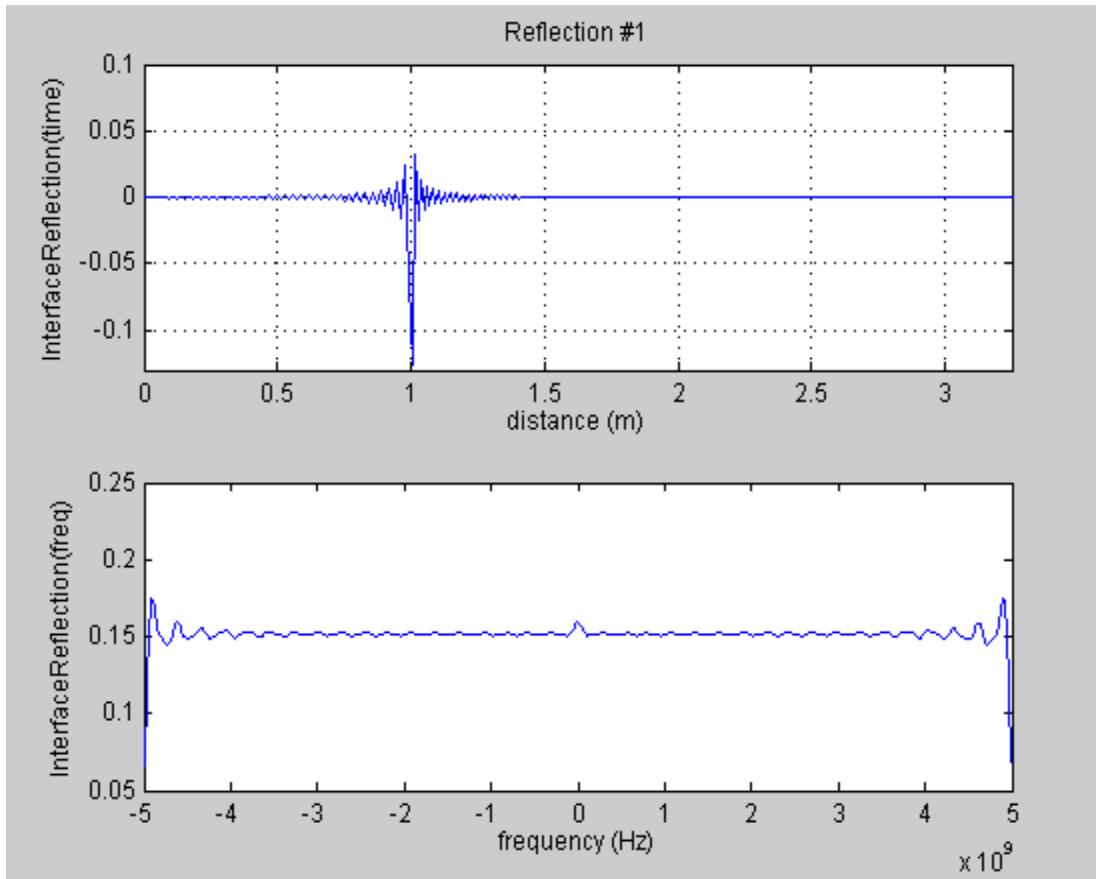


Figure 8: First Interface Reflection for *fullTestInvert.m*.

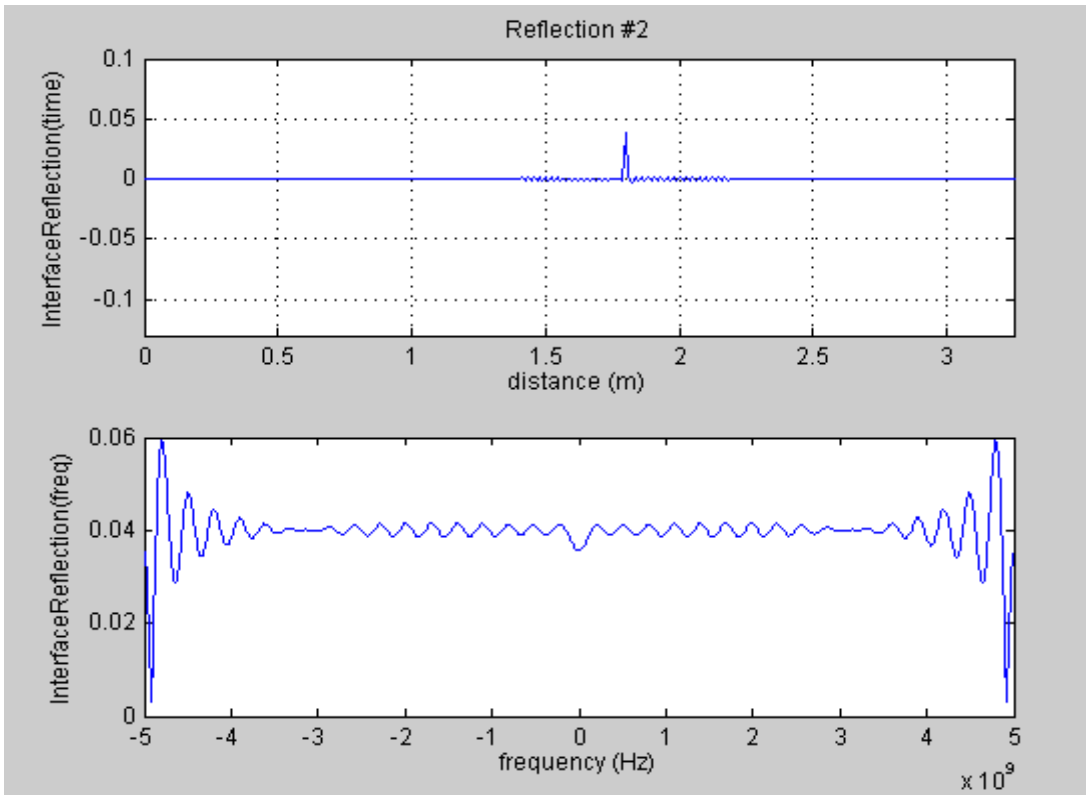


Figure 9: Second Interface Reflection for *fullTestInvert.m*.

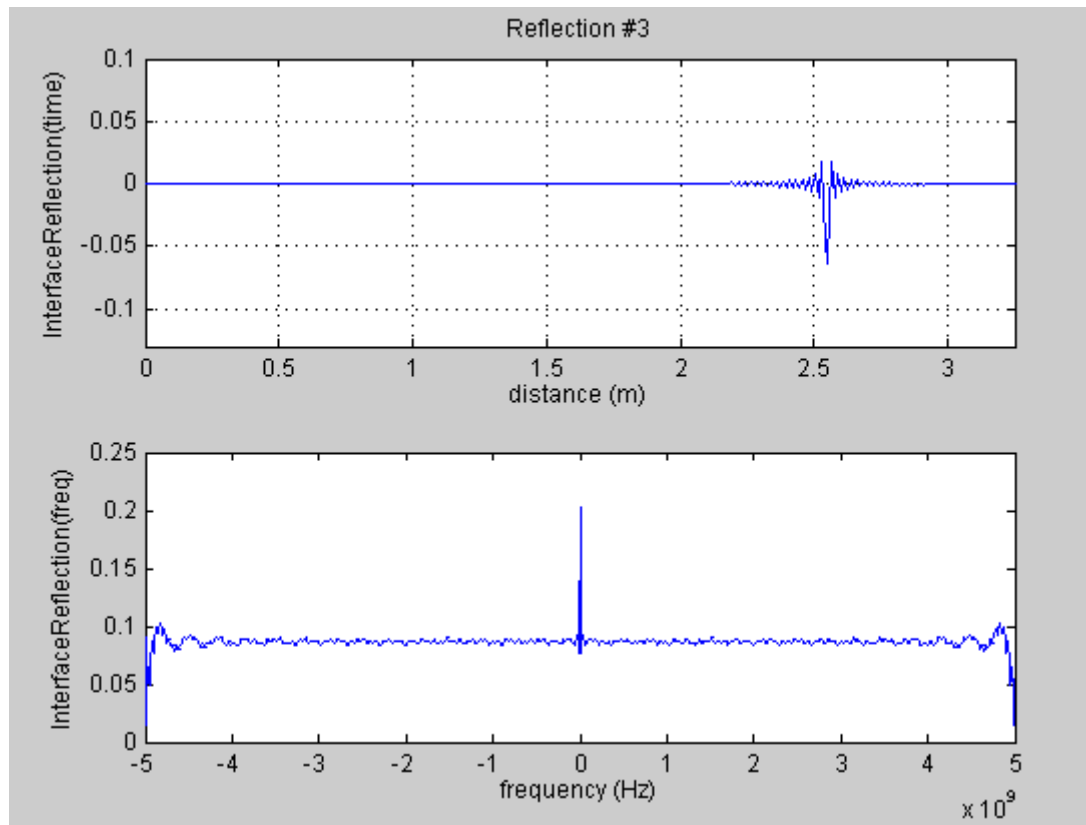


Figure 10: Third Interface Reflection for *fullTestInvert.m*.

Next, the distance profile and continuous permittivity profiles are calculated using *calcDistanceProfile.m* and *calcPermittivityProfiles.m* respectively. The continuous permittivity profiles can be seen in the figure below.

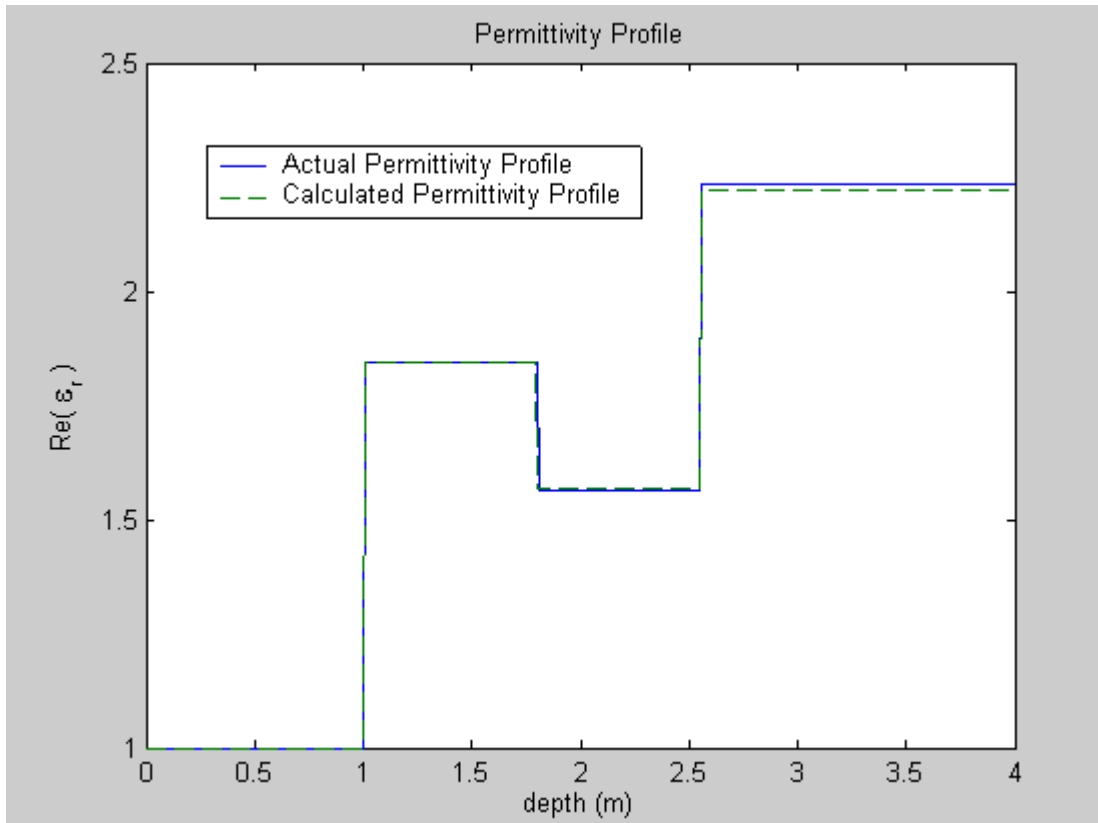


Figure 11: Continuous Permittivity Profiles for *fullTestInvert.m*.

From Figure 11, one can see that reconstructed permittivity profile from the data inversion process is an excellent approximation of the actual permittivity profile. It is also important to note that this reconstruction uses no *a priori* information. The only information required for the data inversion process is the reflectivity data, sampling parameters, and the radar parameters of the system.

## Discussion

The data inversion process described in this paper depends on dielectric interface detection in the impulse response of the multi-layer stack ( $\Gamma_{\text{time}}(t)$ ). If interfaces are not detected correctly -- i.e., false interfaces are detected -- then error is introduced into the permittivity reconstruction. The error is magnified because every permittivity calculation is based on the calculated permittivity of the previous layer. Therefore, an improvement in the automated interface detection scheme would make the data inversion process more accurate and reliable.

Possible ways of accomplishing better interface detection would involve the de-noising of  $\Gamma_{\text{time}}(t)$ , and then a piece-wise linear approximation of the waveform. Thresholding could also be used to only allow reflections of a certain amplitude to be used in the permittivity reconstruction process. Another solution would allow the program user to select dielectric interface points from a graph, or from a finite set of points provided by the program itself.

More work will be devoted to advanced dielectric interface detection in the future. Signal processing techniques to improve resolution of interface peaks in  $\Gamma_{\text{time}}(t)$  will be investigated, along with methods of de-noising, and waveform approximation.

## Conclusion

The permittivity reconstruction process described in this paper transforms simple radar parameters and reflectivity data into a meaningful characterization of layers in a multi-layer stack. In the past, taking samples of layered structures and examining each layer separately was the only way to find the permittivity data provided by the data inversion process. Examining layered structures often involves some type of destructive measures to extract samples of the layers. With new and emerging ground-penetrating radar (GPR) technology, radar reflectivity data can be collected over a layered structure in a totally non-destructive manner. Data inversion algorithms can process this radar reflectivity data and produce characterizations of the layered structure without any *a priori* information about the layered media. Now, layered structures can be characterized without ever having to actually examine the layered structure itself.

The data inversion process presented here allows for this characterization without the use of any *a priori* information. The process allows for a fast and accurate approximation of the permittivity profile of the stratified, or layered, structure. The combination of layer stripping with scattering matrix theory allows for a permittivity reconstruction process that is quite easy to understand and implement.

The permittivity profiles for layered structures can be reconstructed with extremely high precision and accuracy. The process works very well for low- to medium-contrast permittivity profiles. In the case of extremely high-contrast permittivity profiles (huge jumps in permittivity), the permittivity reconstruction provided by this process will not be as accurate, due to false interface detection. However, the permittivity profile generated

provides an excellent starting point for use in optimization algorithms involving the Riccati Equation (1).

## References

- Hopcraft, K.I., and P.R. Smith, *An Introduction to Electromagnetic Inverse Scattering*, Kluwer Academic Publishers, Netherlands, pp. 1-201, 1992
- Mikhnev, V. A., "Microwave Reconstruction Approach for Stepped-Frequency Radar," <<http://www.ndt.net/article/wcndt00/papers/idn354/idn354.htm>>, 4/9/2003.
- Mikhnev, V. A., and P. Vainikainen, "Two-Step Inverse Scattering Method For One-Dimensional Permittivity Profiles," *IEEE Trans. Antennas and Propagation*, vol. AP-48, pp. 293-298, 2000.
- Ramasami, V. C., "A Simulation Package for Radar Systems," Radar and Remote Sensing Laboratory, Information and Telecommunication Technology Center, University of Kansas, 2003.
- "Reflection and Transmission by Planar Stratified Structures," <<http://mpi.leeds.ac.uk/teaching/SIA/lect5.pdf>>, 10/10/1999.

# MATLAB Function Documentation

## **findInterfaceIndeces.m:**

The `findInterfaceIndeces` function operates on the reflection coefficients in the time domain (AKA `gamma_time`). The large peaks, or spikes, in the `gamma_time` waveform represent dielectric interfaces in the stratified medium, as shown in Figure 12.

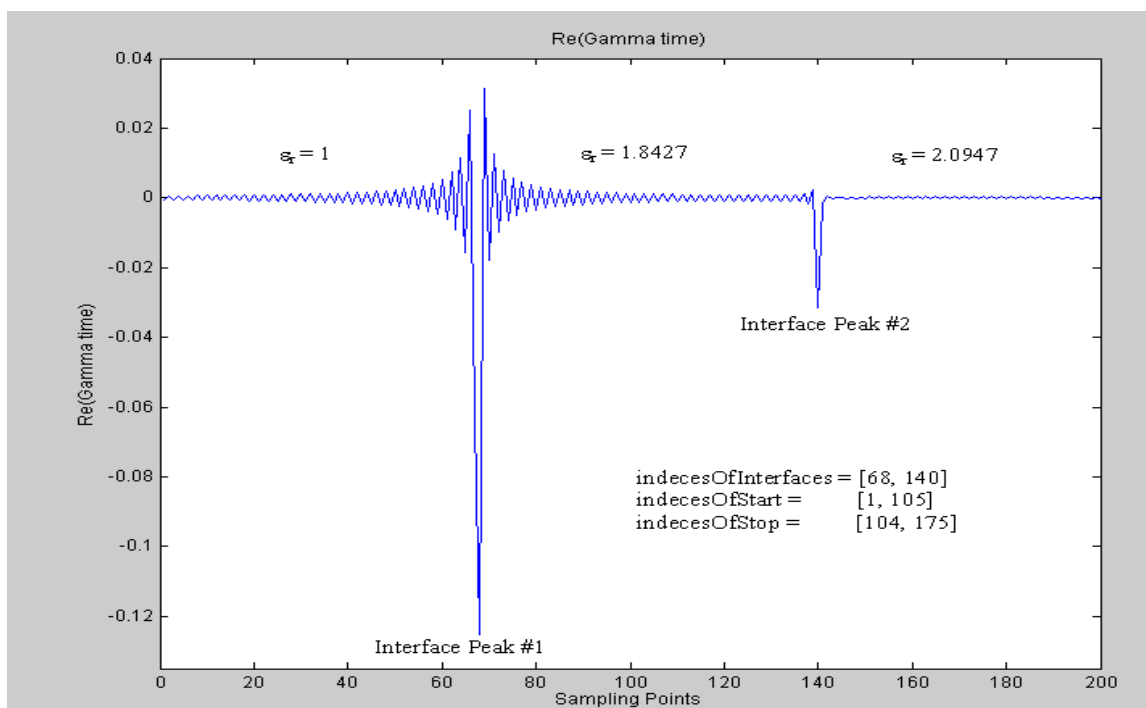


Figure 12: Three-layer medium.

The function returns three different vectors after it has completed:

- **indicesOfInterfaces:** indices in `gamma_time` of the center of each interface peak
- **indicesOfStart:** indices of “beginning” of each interface peak
- **indicesOfStop:** indices of “end” of each interface peak

The index values returned can later be converted into time values or distance values by converting with the formulas based on sampling time and electromagnetic wave speed.

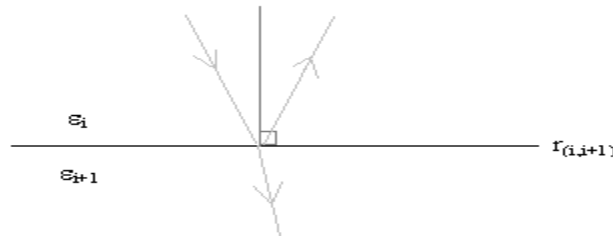
The `findInterfaceIndeces` function works by taking the derivative of the `gamma_time` waveform and finding the local maxima that correspond to interface peaks. This process is implemented by taking the derivative of the absolute value of `gamma_time` using the built-in `diff` and `abs` functions. Once the derivative has been calculated, a small positive offset on the order of  $10^{-3}$  is added to the derivative in order to get rid of the extremely small maxima that occur due to the high-frequency oscillations in `gamma_time`. Next, maxima are detected by recording the indices in which the slope goes from positive to

negative. After the `indecesOfInterfaces` have been found, the `indecesOfStart` and `indecesOfStop` are found by calculating midpoints between the interfaces and rounding to the nearest integer for the use of that number as an array index.

**erCalcFromLayerSubtraction.m:**

The `erCalcFromLayerSubtraction` function operates on the reflection coefficients in both the time and frequency domains, in order to calculate the electric permittivity of each layer.

Using a looping structure, the function iteratively separates each reflection and subtracts it from the total reflection coefficient. As each reflection is separated, its FFT is calculated and then averaged between the frequencies of radar interest (Figure 13). This mean value corresponds to the reflection coefficient,  $\pm r_{(i,i+1)}$ , between dielectric layer (i) and dielectric layer (i+1). Once the reflection coefficient has been calculated for all of the dielectric interfaces, another iterative loop calculates the electric permittivity of each layer using the following simple formula:



$$\pm r_{(i,i+1)} = \frac{\sqrt{\epsilon_i} - \sqrt{\epsilon_{i+1}}}{\sqrt{\epsilon_i} + \sqrt{\epsilon_{i+1}}}$$

Note #1:  $\pm$  sign corresponds to the sign of `gamma_time` at the interface

Note #2:  $\epsilon_0 = 1$ , because the radar antenna begins in air

The `erCalcFromLayerSubtraction` function returns two parameters: a matrix of reflection coefficients for each separated interface (the R matrix, `R(:,x)` are the reflection coefficients for the  $x^{\text{th}}$  interface), and a row vector of permittivity values corresponding to each dielectric layer (the `newErCalc` vector).

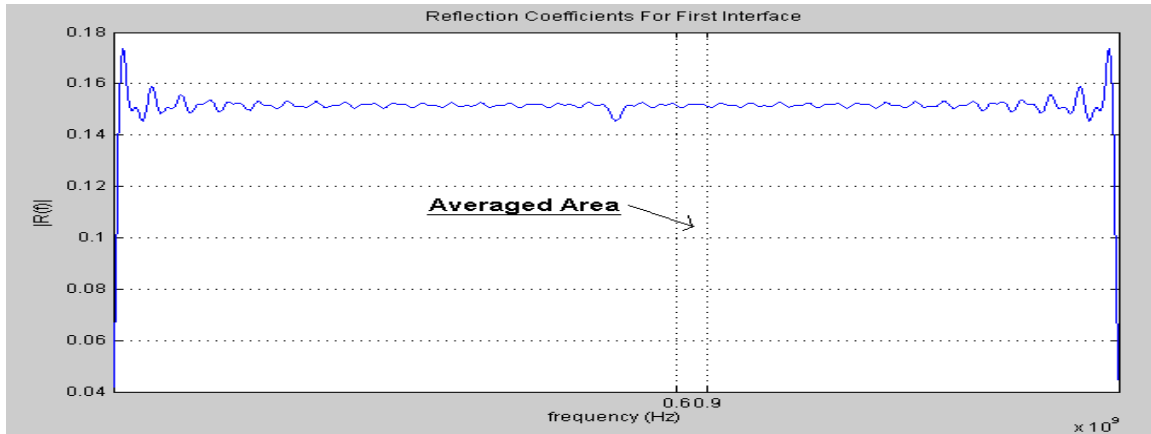


Figure 13: Isolated Reflection Coefficient, Radar Range: 600-900 MHz

### calcDistanceProfile.m:

The calcDistanceProfile function creates a distance axis that relates sampling points to distance by electromagnetic wave speed in a dielectric medium. The distance information returned by this function can be used to find the depth of each of the subsurface dielectric layers.

The speed of an electromagnetic wave in each dielectric layer is calculated using the following formula:

$$c = \frac{1}{\sqrt{\epsilon_r \epsilon_0 \mu_0}}$$

Where:

$$\begin{aligned} c &= \text{wave speed [m/s]} \\ \epsilon_r &= \text{relative permittivity} \\ \epsilon_0 &= \text{permittivity of free-space [F/m]} \\ \mu_0 &= \text{permeability of free-space [H/m]} \end{aligned}$$

Next, the distance traveled per unit time (tS) is calculated in each layer and a distance axis is created by the following formula relating wave travel time to distance:

$$t = \frac{2d}{c}$$

Where:

$$\begin{aligned} t &= \text{time [s]} \\ c &= \text{wave speed [m/s]} \\ d &= \text{distance to target [m]} \end{aligned}$$

The distance axis that is returned can be used to plot gamma\_time against actual depth instead of by time or sampling points.

The depth, relative to transmitter, of each dielectric layer can be found by the following piece of MATLAB code:

```
>> [speeds,distanceSteps,distanceAxis]=calcDistanceProfile(...)
>> distanceAxis(indecesOfInterfaces)
```

### **calcPermittivityProfiles.m:**

The calcPermittivityProfiles function creates a continuous permittivity profile of both the actual and calculated permittivity values. This function assumes each dielectric layer is homogenous; therefore, the resulting permittivity profiles will be piecewise linear functions, in which permittivity can be seen versus depth.

Here is an example permittivity profile calculated using “testDeepInvert.m”:

### **Program Output:**

```
erCalc = 1.0000 1.8426 2.0878 2.2220 1.7357 2.9509 [Permittivity profile from data inversion]
erActual = 1.0000 1.8427 2.0947 2.2345 1.7346 3.0000 [Actual permittivity profile]
```

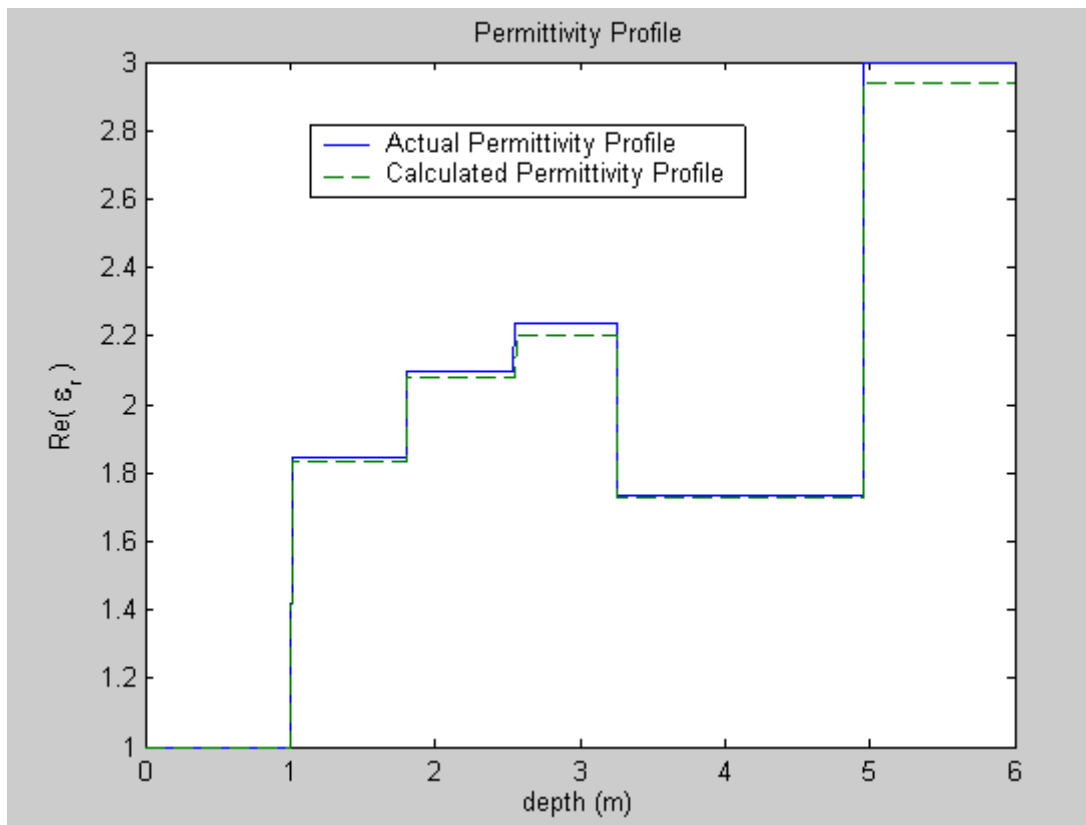


Figure 14: Continuous Permittivity Profiles For “testDeepInvert.m”