CHAPTER 6. COMPARISONS OF NEURAL NETWORK FILTERS WITH OTHER METHODS

6.1 Comparison of the model training method with least-square and simple stacking methods

6.1.1 MTM and least-square power rule fitting

The general form of the power rule for a layered earth is given as:

$$A(t_i) = \sum_{j=1}^{m} a_j t_i^{\lambda_j} , \tag{6.1}$$

where $A(t_i)$ is the TEM response at the delay time t_i, a_j are amplitude coefficients, and λ_j are time constants. The first (m=1) and second order (m=2) power rules were used for comparison between an MTM and least-square fitting. From Equation 6.1, the first and second order forms of Equation 6.1 are given by:

$$A(t_i) = a_1 t_i^{\lambda_1} \tag{6.2a}$$

$$A(t_i) = a_1 t_i^{\lambda_1} + a_2 t_i^{\lambda_2} . \tag{6.2b}$$

By taking the logarithm of both sides of Equation 6.2a, the parameters, i.e., the coefficient and time constant, of the first order power rule can be solved by linear least-square. One way to find the parameters of Equation 6.2b is to solve the minimisation problem of nonlinear least-squares.

Both an MTM and least-square method were applied to the same noisy transient generated from the model TS3 and EM noise data recorded in Darwin. The noisy transient and the S/N ratio are shown in Figure 6.1. The S/N ratio at early delay times is worse than the average S/N ratio (Figure 3.4) calculated from 1000 different noisy transients while the S/N ratio at late delay times is better than the average S/N ratio. For example, the S/N ratio of this noisy transient at a delay time of 0.487 ms is

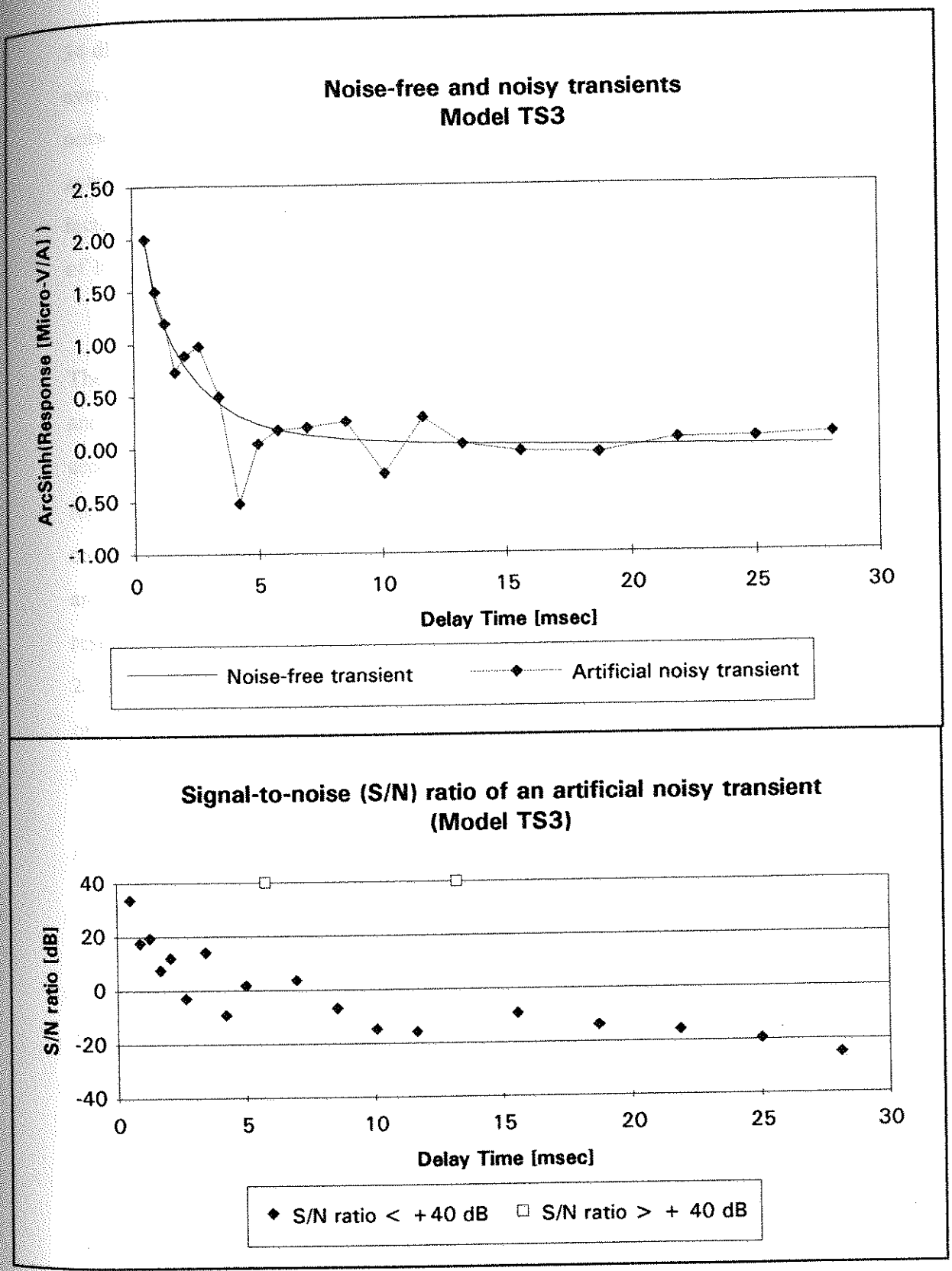

Figure 6.1 Noise-free and noisy transient responses of the testing model TS3 and S/N ratios over a range of delay times.

34 dB (i.e., the signal amplitude is higher than the noise by a factor of 50), while the average S/N ratio is 44 dB (i.e., the signal amplitude is 158 times higher than the noise). At a delay time of 24.99 ms, the S/N ratio of this noisy transient and the average S/N ratio are -19 dB (i.e., the signal amplitude is smaller than the noise by a factor of 9) and -25 dB (i.e., the signal amplitude is smaller than the noise by a factor of 18), respectively.

The neural network results and least-square fit curves are shown in Figure 6.2. The coefficient and time constant of the first order power rule are obtained as 0.012 and -1.948, respectively, by linear least-square fitting. For the parameters of the second order power rule, the values of 0.001, 0.011, -2.668, and -1.842 are obtained for a_1, a_2, λ_1, and λ_2, respectively, by non-linear least-square fitting method. The comparison between an MTM and least-square fittings shows that the MTM achieves an *NFE* value (i.e., as defined in Equation 3.1, a normalised root-mean-square of the residual value between the true and filtered responses) of 12 % at a delay time of 28.12 ms. The *NFE* values of 32 and 68 % are obtained using the first and second order power rules, respectively (Figure 6.2), and therefore the neural network appears to reduce more effectively noise produced by sferics.

6.1.2 MTM and a simple stacking method

Figure 6.3 shows a graphical example of how time windowing used in most TEM systems can produce low-frequency noise on TEM responses at late times. The effect is illustrated with a time window of 3 ms derived from summing 60 consecutive samples taken at a rate of 20 kHz. The top diagram of Figure 6.3 shows 600 samples of Darwin 1983 sferics data after removing the DC offset introduced by the tape recorder used to record the data. As described in Section 3.2 of Chapter 3, in order to remove this DC offset, two noise samples are chosen with an interval of 1160 sample points equivalent to 25 SIROTEM standard delay times, the second is subtracted from the first, and the difference is divided by 2. The middle diagram of Figure 6.3 shows

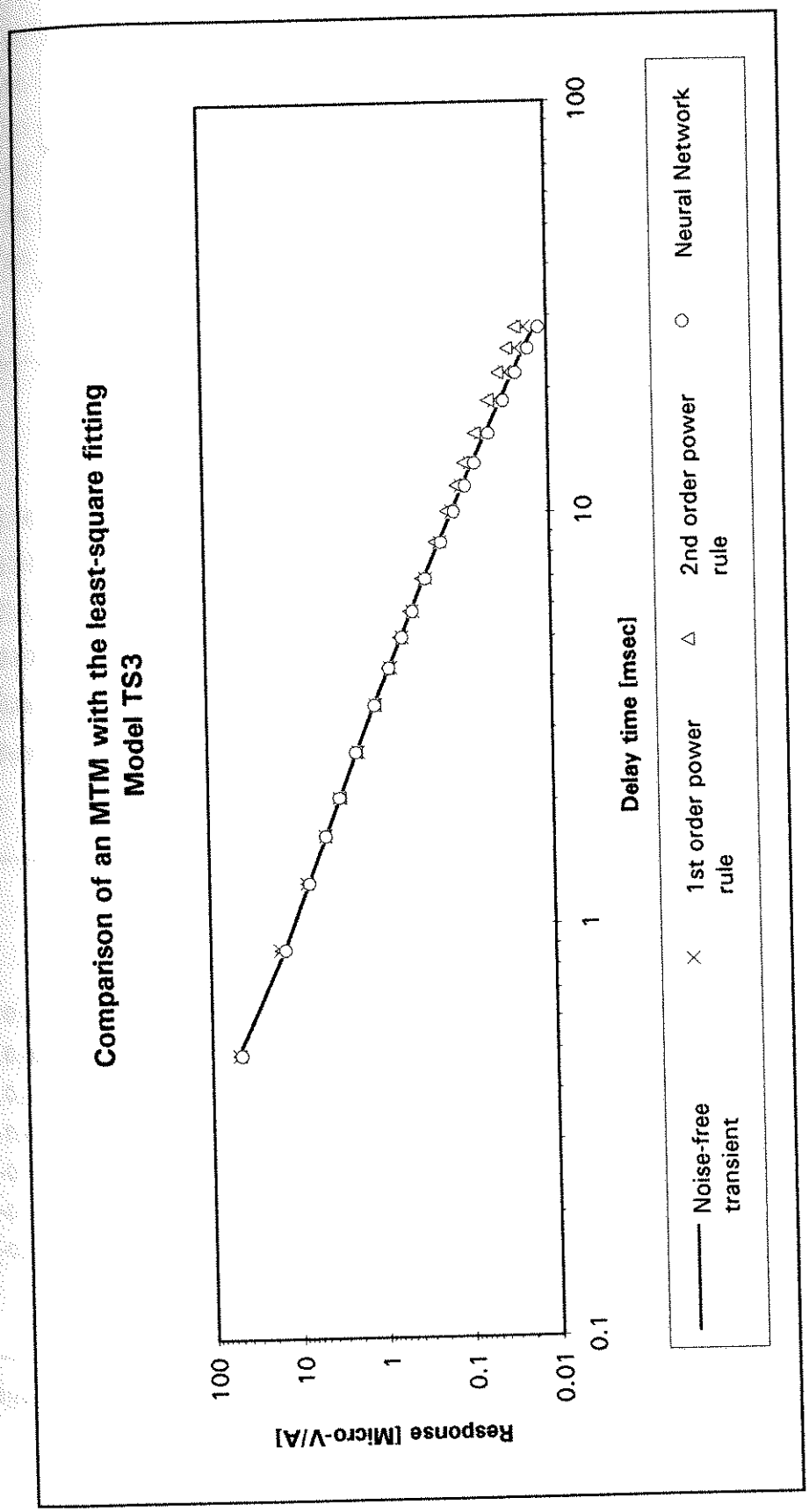

Figure 6.2 Comparison of an MTM with a least-square fit method.

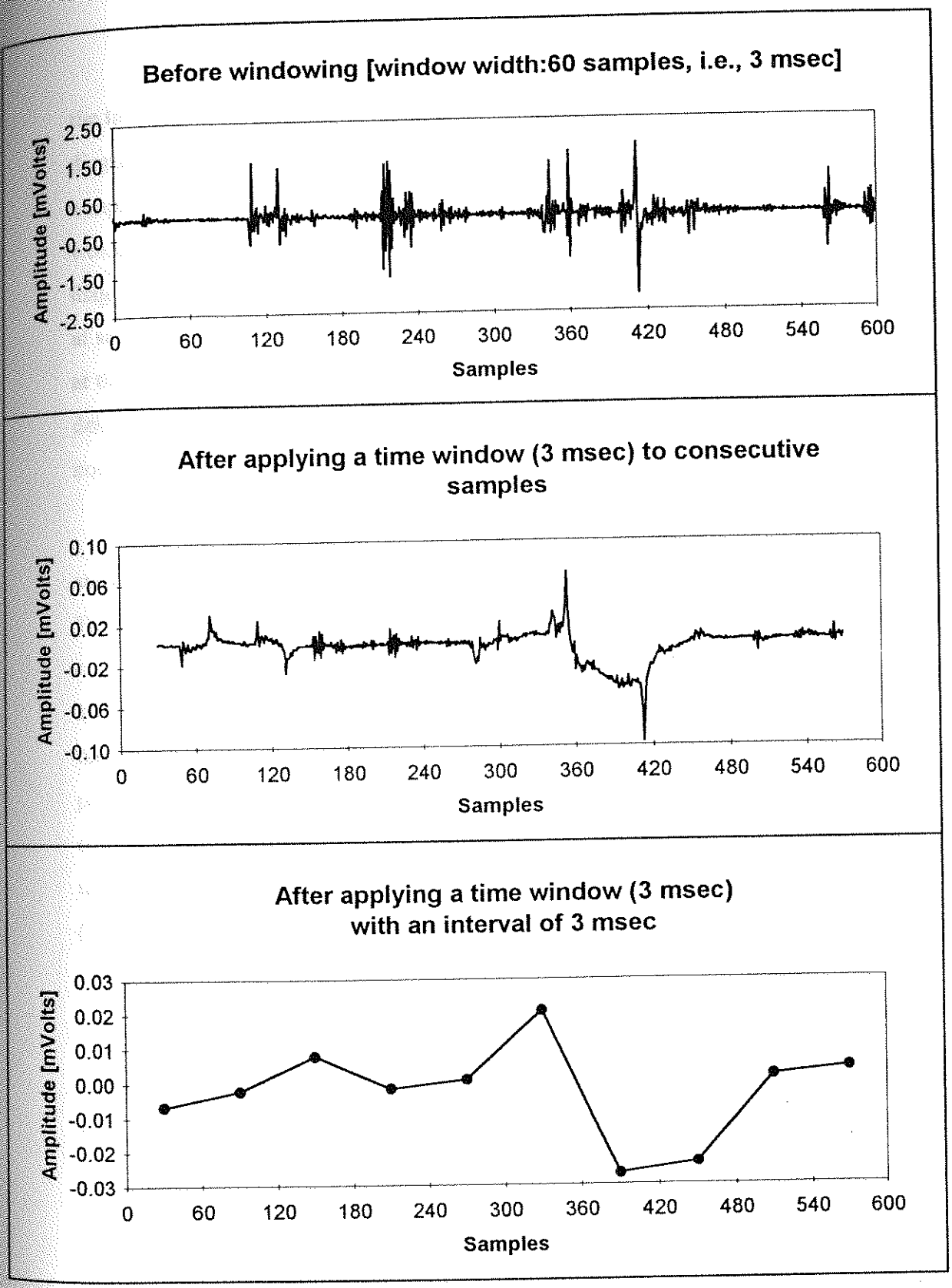

Figure 6.3 Example of effect of the time windowing scheme when the time window (a width of 60 samples) was applied to a 600-sample block of the vertical component time series shown in Figure 3.2.

the time series when a moving time window is applied to the time series of the top diagram. The time window produces low-frequency (~111 Hz) noise in the range from sample points 300 to 480. The bottom diagram of Figure 6.3 shows the effect of time windowing when contiguous time windows are obtained from the time series in the top diagram of Figure 6.3 in the same manner as applied by SIROTEM. If the noise values at sample values of 330 to 570 were added to the noise-free responses of Model TS3 at the delay times of 15.58, 18.72, 21.85, 24.99, and 28.12 ms, the peak at the late delay times shown in the noisy transient would be considered to be low-frequency EM noise.

Figure 6.4 shows a comparison of noise reduction of the MTM with a simple stacking method. The noise-free transient of model TS3 is represented as a solid line. The transient obtained by windowing and stacking 256 noisy transients is denoted as crosses. The open circles represents the filtered transient obtained from the MTM. As shown in Figure 6.4, the 256 stacked transient still shows spikes at the late delay times. It is noted that the noise peak at the delay time of 24.99 ms in Figure 6.4 is produced by windowing high-frequency sferics spikes rather than by low-frequency components of the sferics pulse itself (see bottom diagram of Figure 6.3). Also, the transient obtained with 1000 stack still shows spikes at the late delay times (Figure 6.5). Even though it is generally known that the worse S/N ratio at late delay times is affected by low-frequency EM noise, the high-frequency EM spike-like noise is an important contributor to the worse S/N ratio at the late delay times.

To compare the results of inversion of the two transients obtained from the MTM and simple stacking methods, 2-layer models are used with program GRENDL. The true values of the first and second layer resistivities and the first layer thickness in the model TS3 are 600, 60 Ω-m, and 350 m, respectively. Table 6.1 shows the values of the model parameters and the inversion results obtained from the filtered transient produced by the MTM and the transient obtained from 256 stacks. As shown in Table 6.1, the MTM produces better inversion results for all parameters. The initial

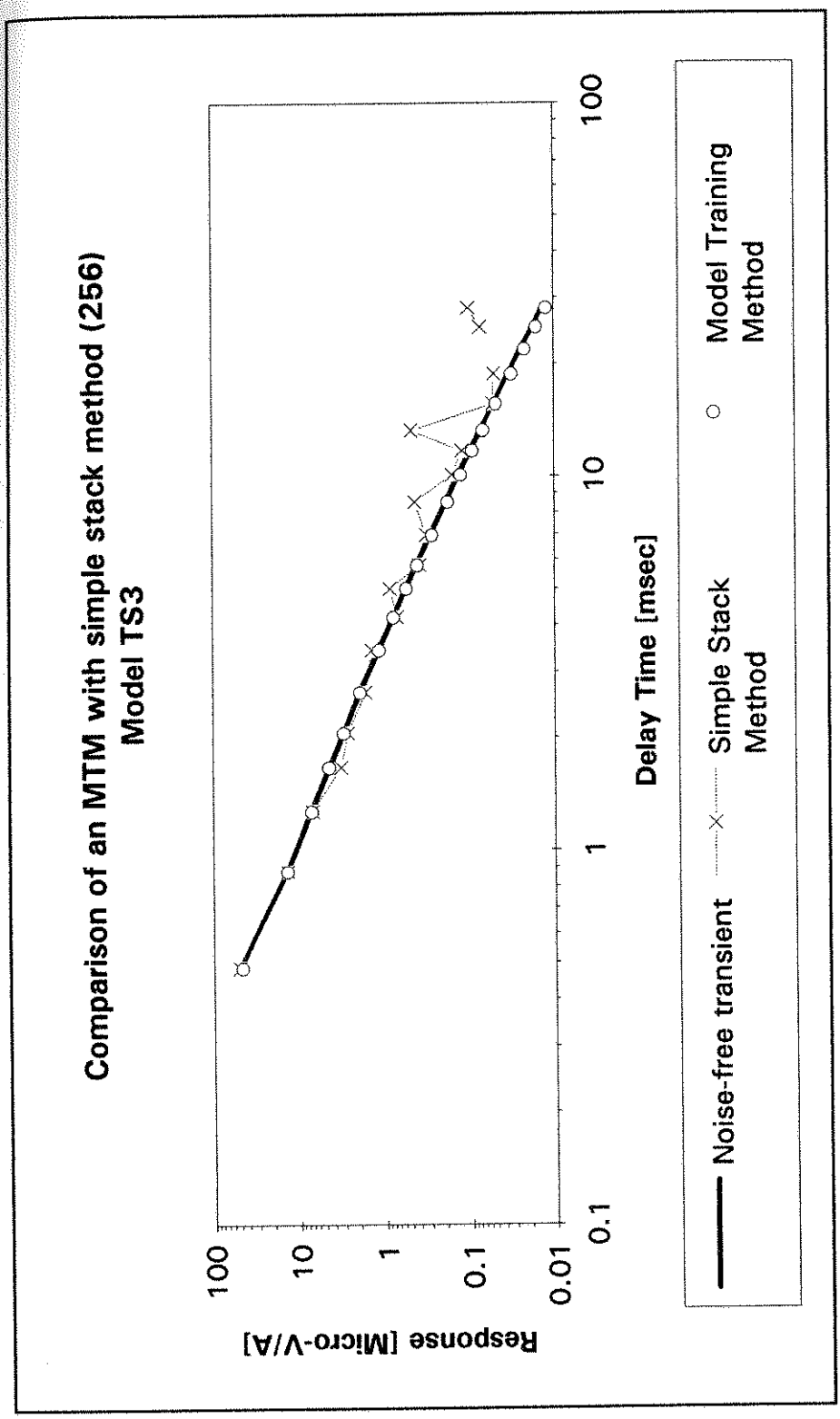

Figure 6.4 Comparison of an MTM with simple stacking (256 stacks).

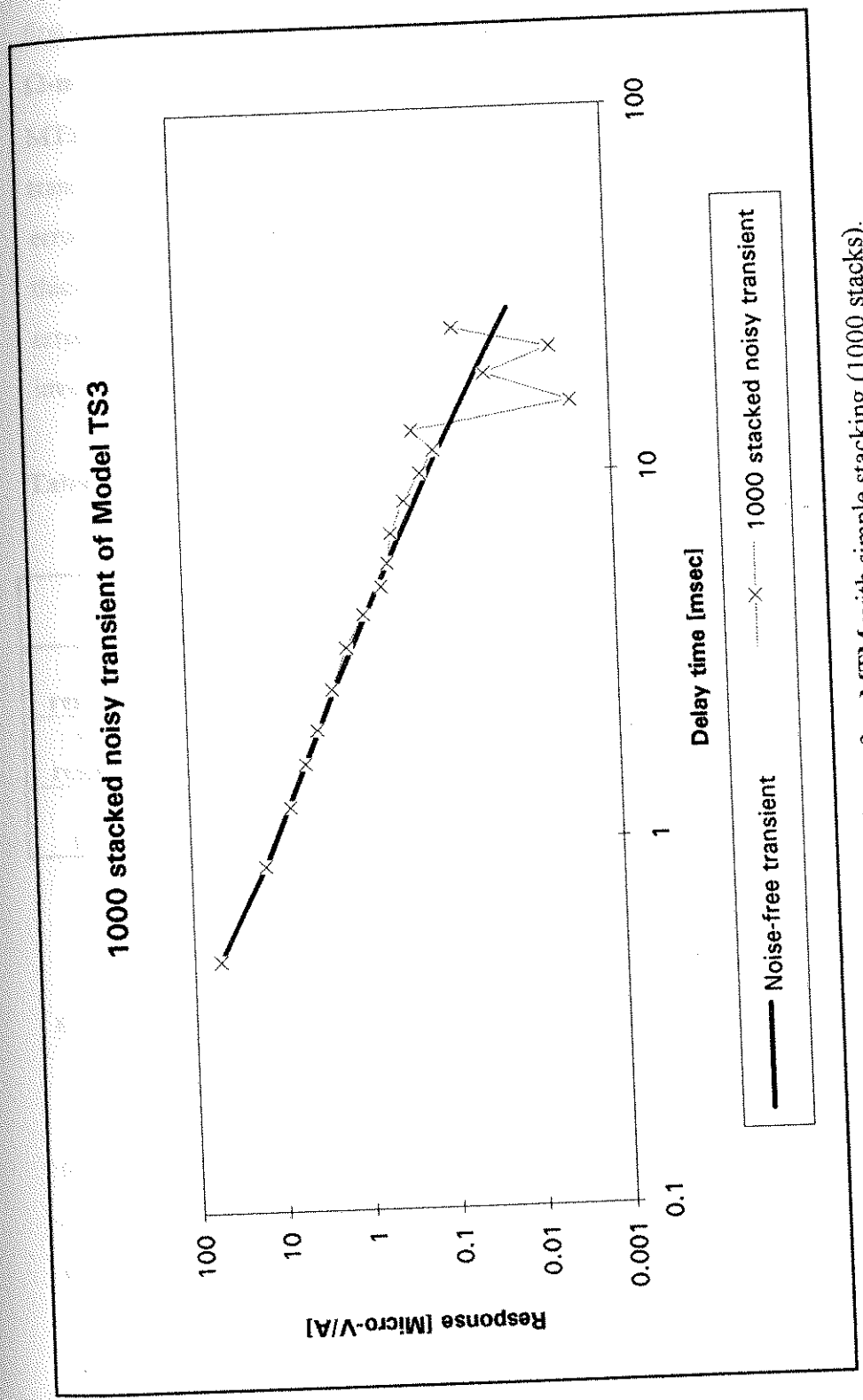

Figure 6.5 Comparison of an MTM with simple stacking (1000 stacks).

2-layer model parameters used in the inversion are 1000, 500 Ω-m, and 50 m as the first and second layer resistivities and the first layer thickness. The values of 679, 62 Ω-m and 344 m are obtained from inversion of the filtered transient produced by the MTM, while the values of 414, 14 Ω-m and 475 m are obtained from the 256 stacked transient (Table 6.1 and Figure 6.6). In model parameter space, the MTM shows errors of 13, 2, and 3 % in the first and second layer resistivities and the first layer thickness, respectively, while the simple stacking method shows the corresponding errors of 31, 35, and 76 %. In Table 6.1, the values within parentheses represent inversion errors in parameters.

Table 6.1 Results of inversion of two transient responses derived from an MTM and simple stacking method.

	True model	Model training method	Simple stacking method
First layer resistivity [Ω-m]	600	679 (13 %)	414 (31 %)
Second layer resistivity [Ω-m]	60	62 (2 %)	14 (35 %)
First layer thickness [m]	350	344 (3 %)	475 (76 %)

6.2 Comparison of the local noise prediction filter with the tipper method

In an LNPF based on the tipper method (called Spies LNPF in this thesis), the decomposition of Equation 4.2 into two coupled subsystems was applied to solve the overdetermined system of linear equations. Rearranging Equation 4.2 gives the two equations:

$$[X(t)*h_x(t)] = h_z(t) - [Y(t)*h_y(t)] \tag{6.3a}$$

$$[Y(t)*h_y(t)] = h_z(t) - [X(t)*h_x(t)] \; . \tag{6.3b}$$

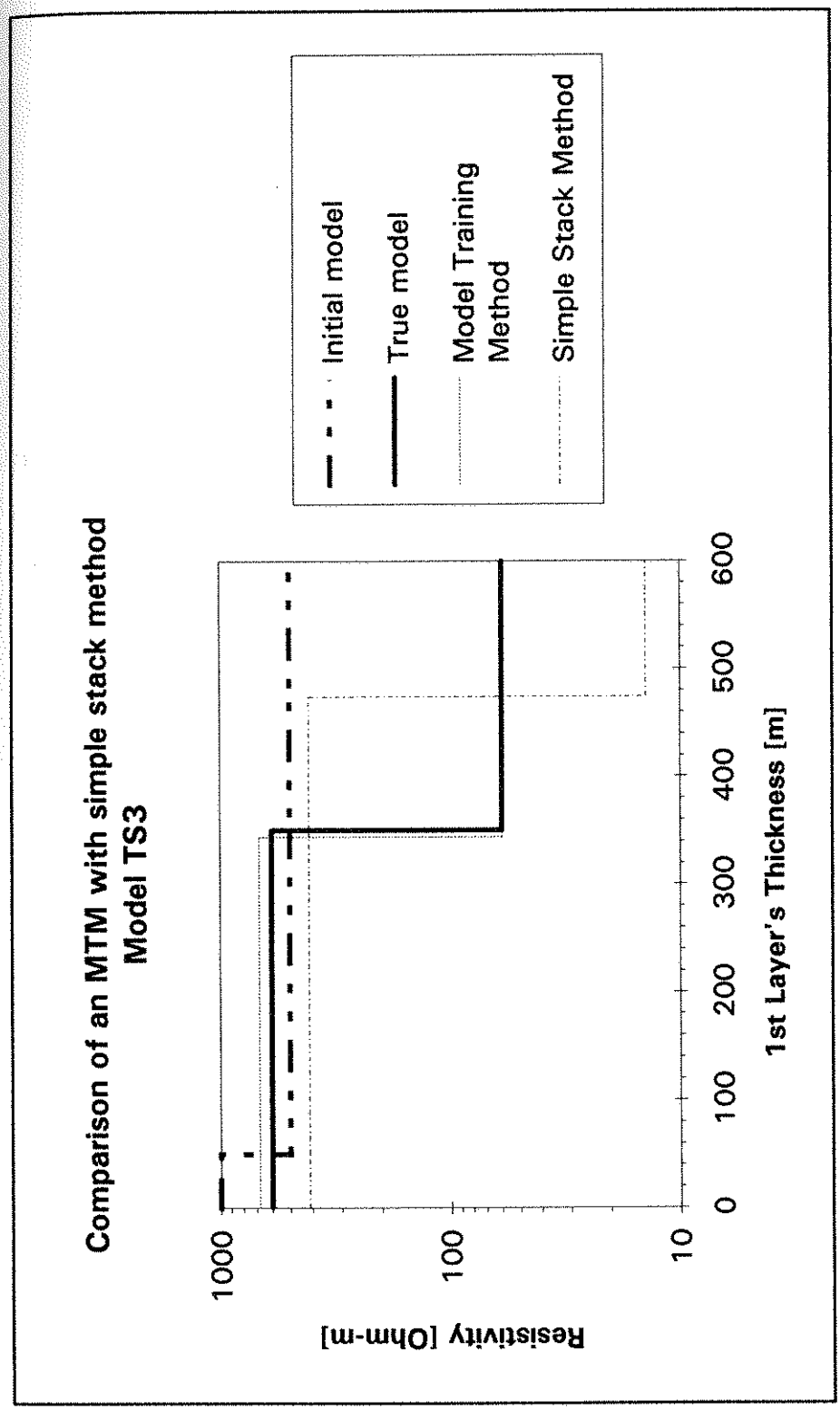

Figure 6.6 Comparison of inversion results of the filtered transient from an MTM and the stacked transient obtained from 256 stacks.

An iterative solution of the two coupled equations 6.3 is used and proceeds as follows:

- A starting estimate of $Y(t)*h_y(t)$ is inserted into Equation 6.3a, which is solved for $X(t)$ by least-square;
- With this $X(t)$, the right side of Equation 6.3b is computed and Equation 6.3b is solved for $Y(t)$ by least-square;
- This $Y(t)$ is put into Equation 6.3a and so on until the values of $X(t)$ and $Y(t)$ become stationary to the desired number of significant figures.

In the comparison between a neural network LNPF and the Spies LNPF, the iterative procedure of the Spies LNPF to update the filter coefficients stops when it meets one of three criteria as follows:

- The noise-to-signal ratio (*NSR*) (Spies, 1988) is less than or equal to 15 %;
- The absolute difference between the average of *NSR* values at the previous three iterations and an *NSR* value at the present iteration is less than or equal to 1.0E-10;
- The number of iterations is equal to 60.

Spies (1988) has demonstrated that the LNPF improves the S/N ratio by a factor of five using a three-point prediction filter (i.e., the length of the impulse responses of the magnetic transfer functions prediction filter is three). However, for comparison of the neural network LNPF with the Spies LNPF applied to the reduction of low-frequency (below 1 Hz) geomagnetic field variations, 3-point and 48-point prediction filters were used for the Spies LNPF. Actually, the filter size for the optimal performance of the Spies LNPF is highly related to the number of samples to be used in iterative procedures to update the impulse responses of the magnetic transfer functions of the Spies LNPF. For example, when few samples are used, then the size

of the filter is small but this filter gives rise to a non-stationary performance (i.e., the filter performance is not invariant with time). The data set used in the training of the neural network LNPF (see Figure 4.30) was used in the determination of the filter coefficients of the Spies LNPF.

The iterative procedure of the Spies LNPF with both the 3-point and 48-point prediction filters stopped because it had satisfied Criterion 2. It took about 2 and 54 minutes to train the 3-point and 48-point prediction filters, respectively. As mentioned in Section 4.3.3 of Chapter 4, the neural network was trained for 60,000 iterations, which are equivalent to less than 10 minutes of elapsed computation time on an Intel 486DX2-66 Mhz processor based PC.

The comparison between the three-point prediction Spies LNPF and the neural network LNPF shows that the neural network filter achieves an *NRF* value of 6.8 while the Spies LNPF obtains an *NRF* of 2.6 (Figure 6.7). Compared to the factor of 5 quoted by Spies, this illustrated the non-stationary performance of the 3-point filter. The result from the 48-point prediction Spies filter is shown in Figure 6.8. An *NRF* of 6.5 is obtained by this filter. The neural network LNPF is superior to the Spies LNPF with the three-point prediction filter. The Spies LNPF with the 48-point prediction filter is only marginally worse.

For high-frequency (above 5 kHz) sferics reduction, a 24-point prediction filter was chosen as the optimal prediction filter length of the Spies LNPF. An example of the Spies LNPF results obtained with Class 2 sferics (i.e., the amplitude of the X component of sferics is almost equal to that of the Y component, and the polarities of the two components are opposite to each other) is displayed in Figure 6.9. The learning procedure of the Spies LNPF stops when Criterion 2 is satisfied. An *NRF* of

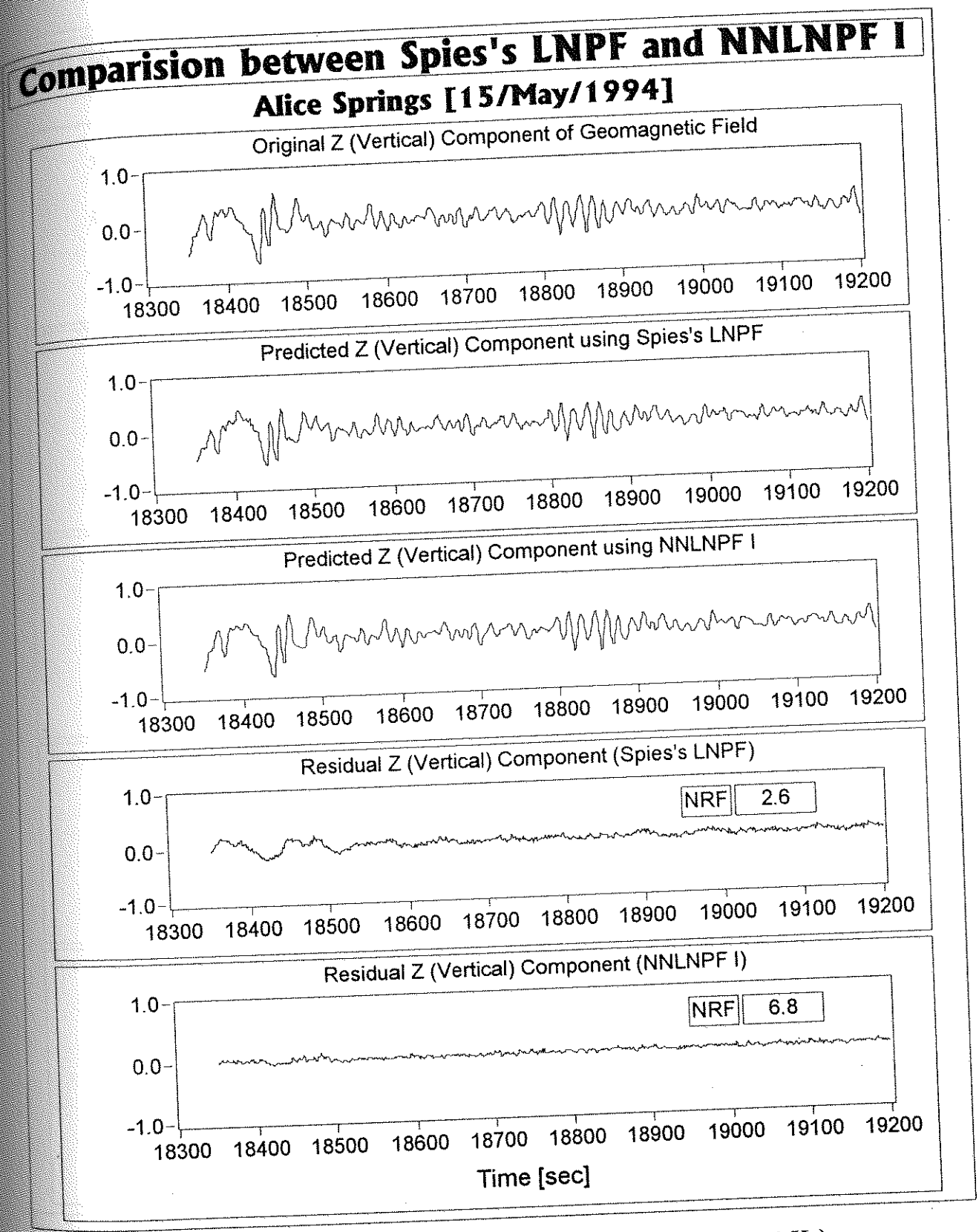

Figure 6.7 Comparison of the reduction of low-frequency (below 1 Hz) geomagnetic variations between the three-point prediction Spies LNPF and the neural network-based LNPF.

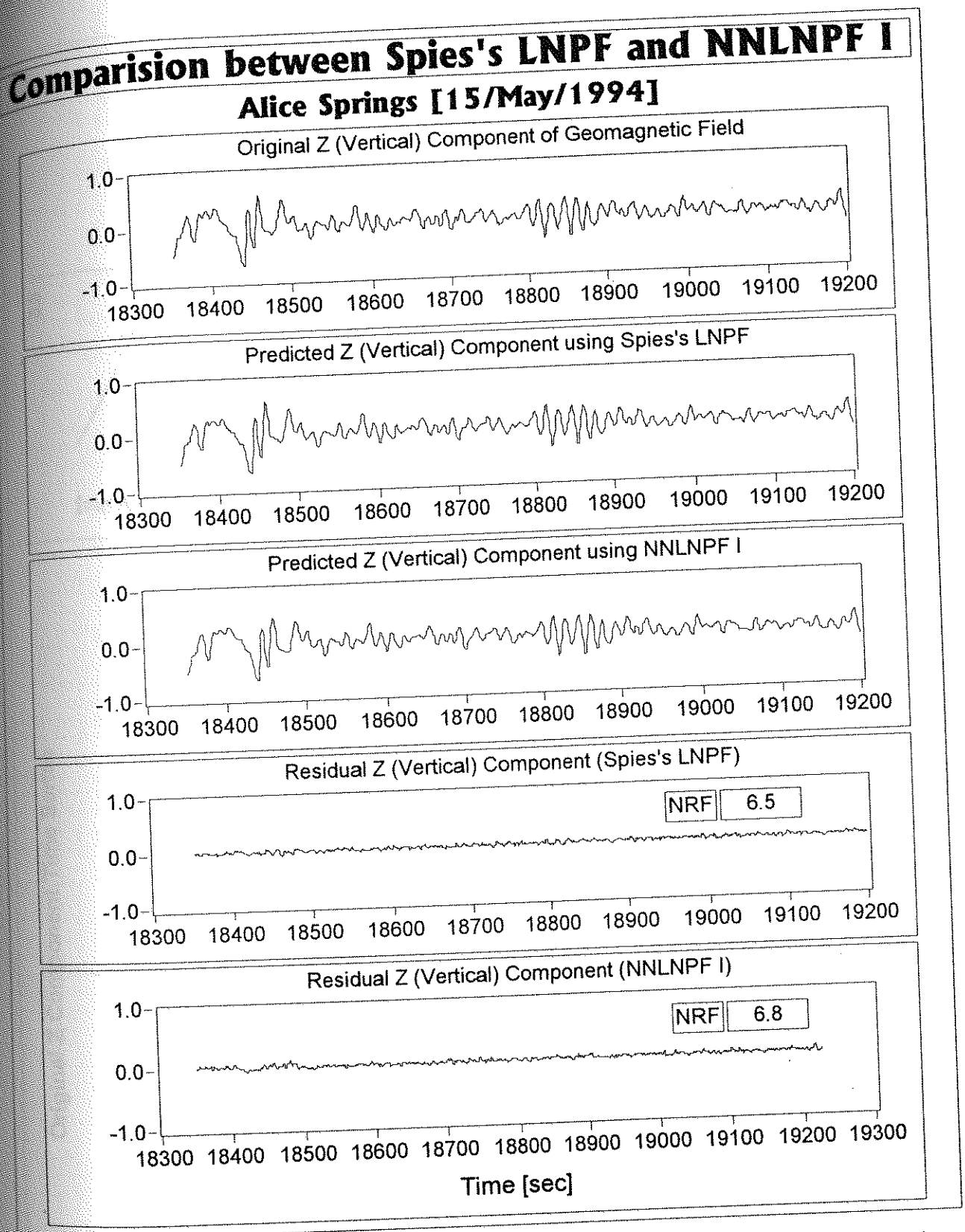

Figure 6.8 Comparison of the reduction of low-frequency (below 1 Hz) geomagnetic variations between the 48-point prediction Spies LNPF and the neural network-based LNPF.

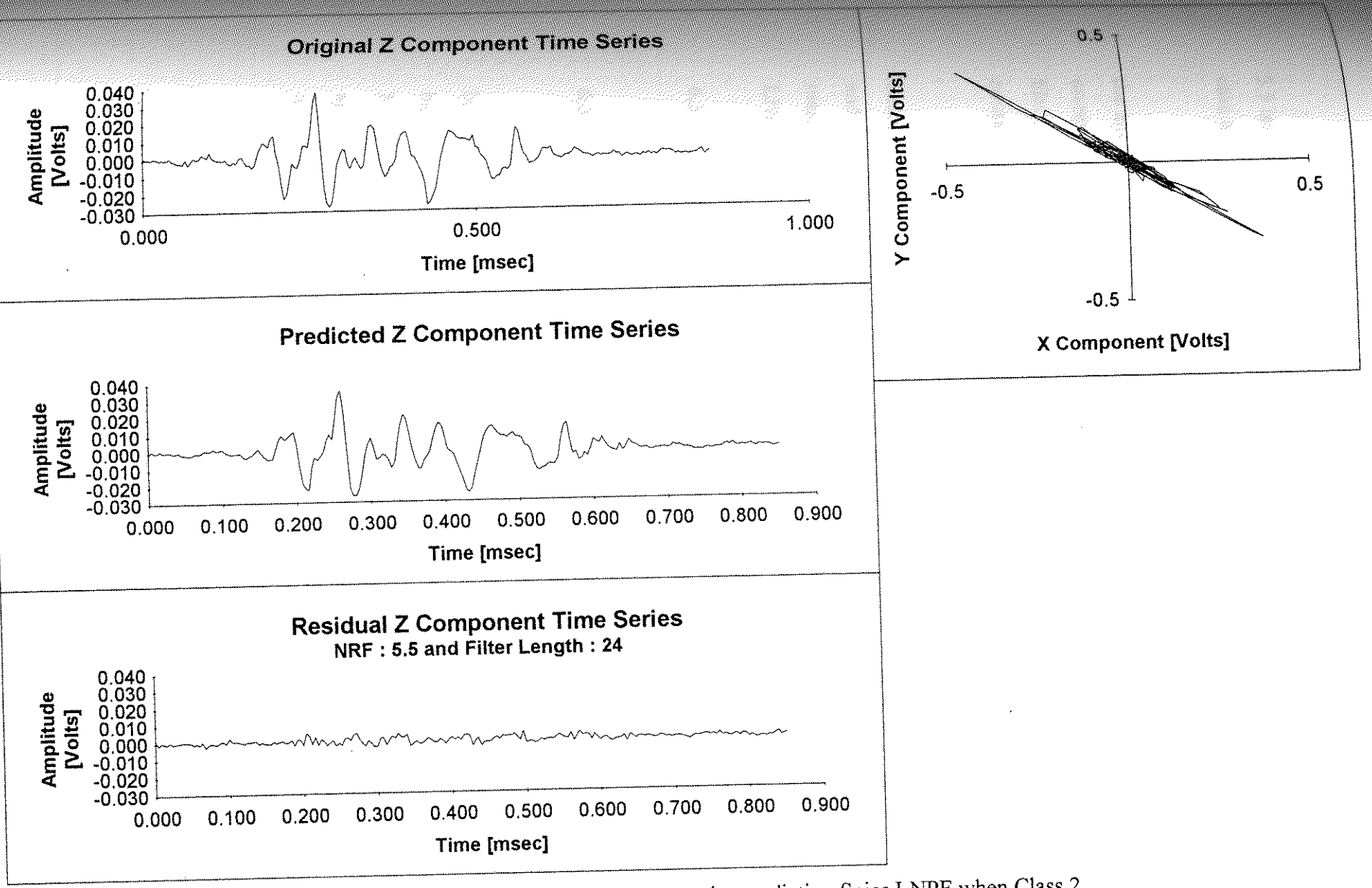

Figure 6.9 Performance of the 24-point prediction Spies LNPF when Class 2 sferics was presented to the Spies LNPF.

5.5 is obtained with this filter, and thus the neural network filter performs better in this case as it yielded an *NRF* of 7.8 (see Figure 4.23).

Figure 6.10 shows the Spies LNPF results obtained with Class 1 sferics (i.e., the amplitude of the X component of sferics is much smaller than that of the Y component, and the polarities of the two components are opposite to each other). This filter achieves an *NRF* value of 5.4. In this case, an *NRF* of 5.9 was obtained by the neural network LNPF.

It is concluded that the Spies method performs worse than the neural network based LNPF when the sferics pulse has approximately equal amplitudes in both the X and Y components (e.g., Figure 6.9). In cases where the sferics pulse is polarised mainly along the X or Y direction (e.g., Figure 6.10), the performance of the Spies LNPF is only marginally worse than the performance of the neural network filter.

6.3 Comparison of the remote noise prediction filter with simple subtraction, autoregressive, and interpolation methods

6.3.1 RNPF and a simple subtraction of two time series

Figure 6.11 shows the X-component time series and their power spectra when both the neural network RNPF and simple subtraction method are applied to background EM noise (mainly, ~20 kHz). The two residual time series show that the RNPF performs marginally better than the simple subtraction method. On the power spectra, both methods attenuate the ~20 kHz VLF by 20 dB (i.e., by a factor of 10 in amplitude).

Figure 6.12 shows the performance of the RNPF and subtraction method when they are applied to the X-component time series of sferics noise. The residual time shift between the local and remote time series is 1.86 μs. An *NRF* value of 2.9 is obtained by simple subtraction, while the RNPF achieves 4.9 as an *NRF* value. In the

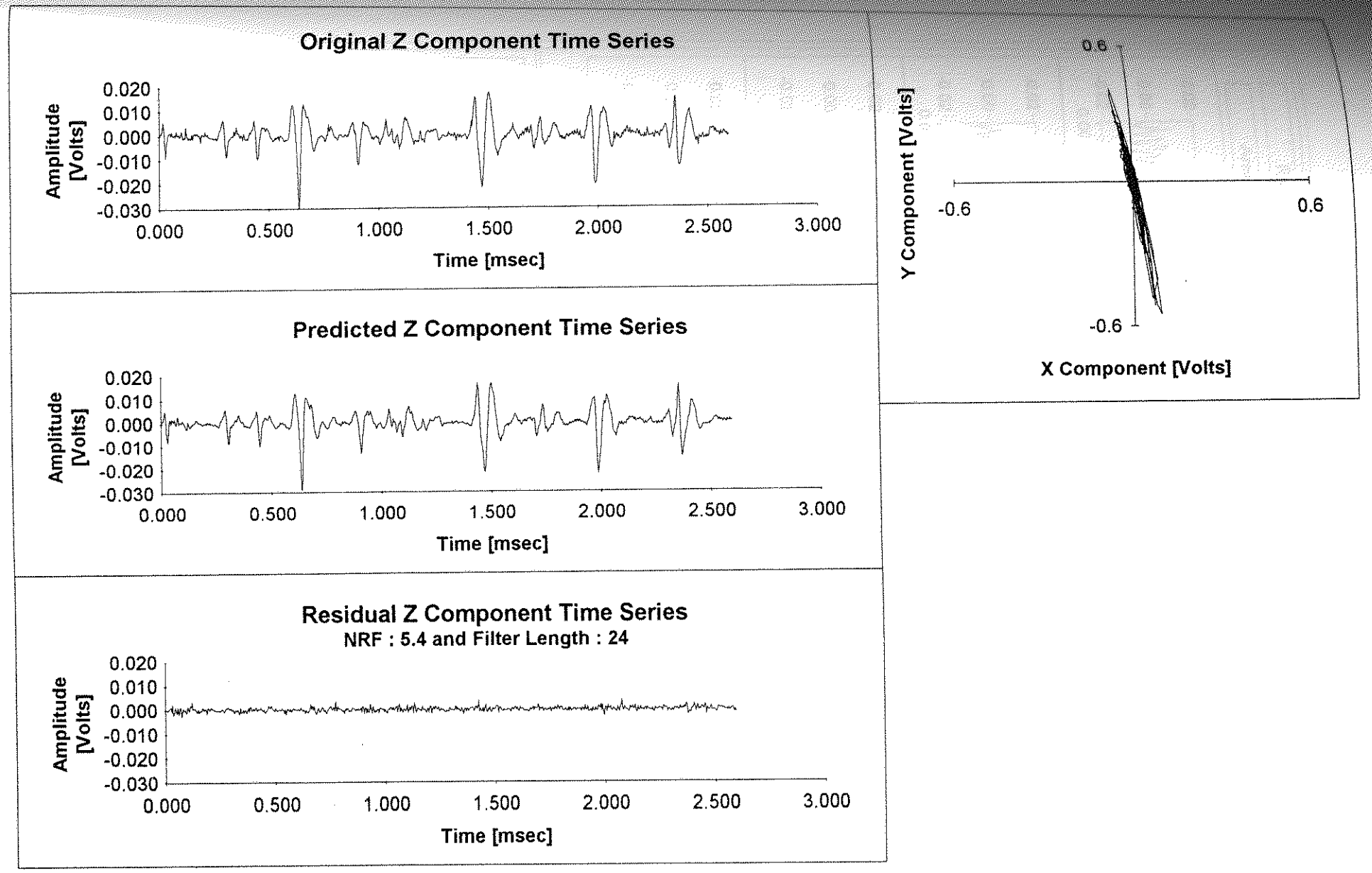

Figure 6.10 Performance of the 24-point prediction Spies LNPF when Class 1 sferics was presented to the Spies LNPF.

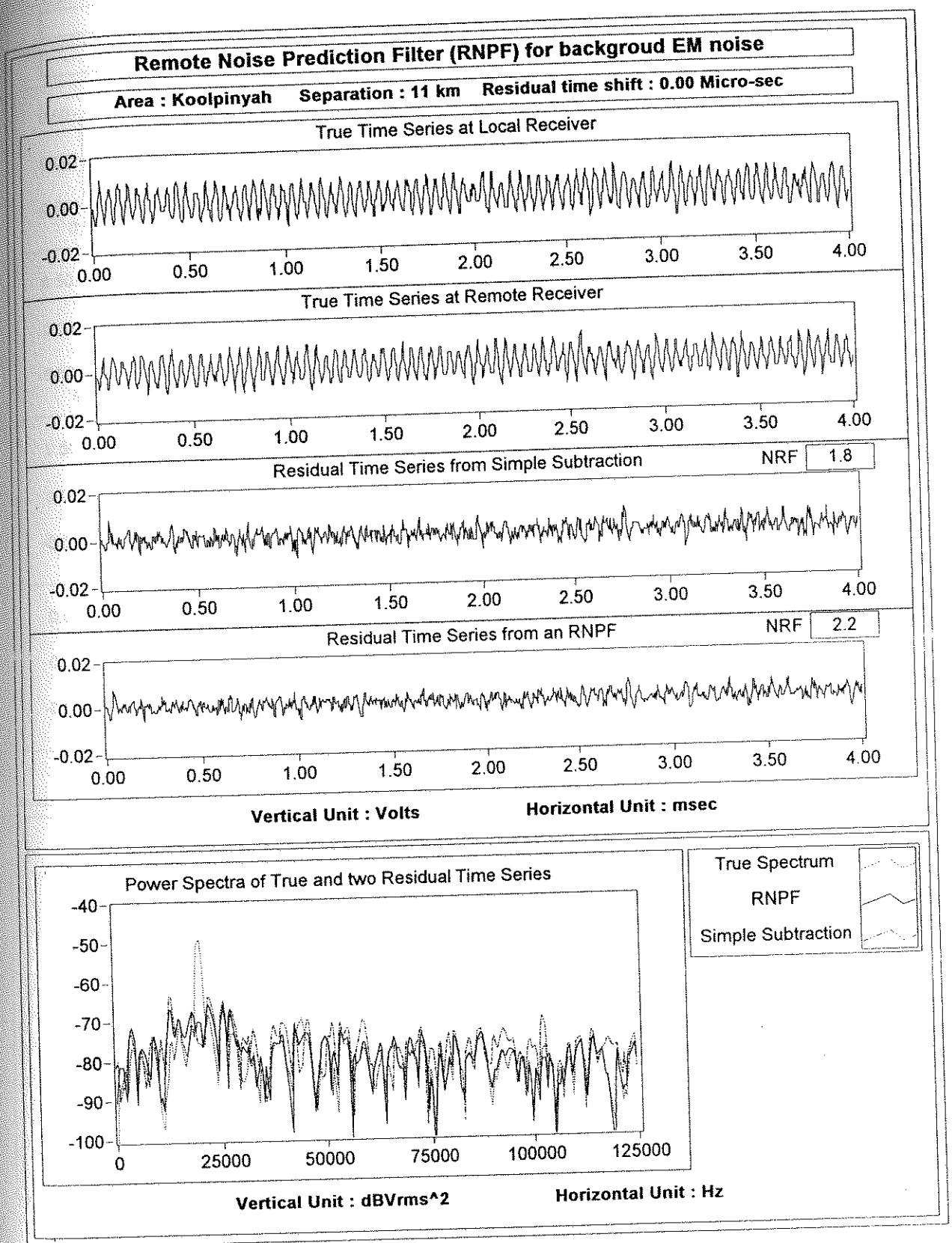

Figure 6.11 Example of performance of an RNPF and simple subtraction to reduce the horizontal component of background noise.

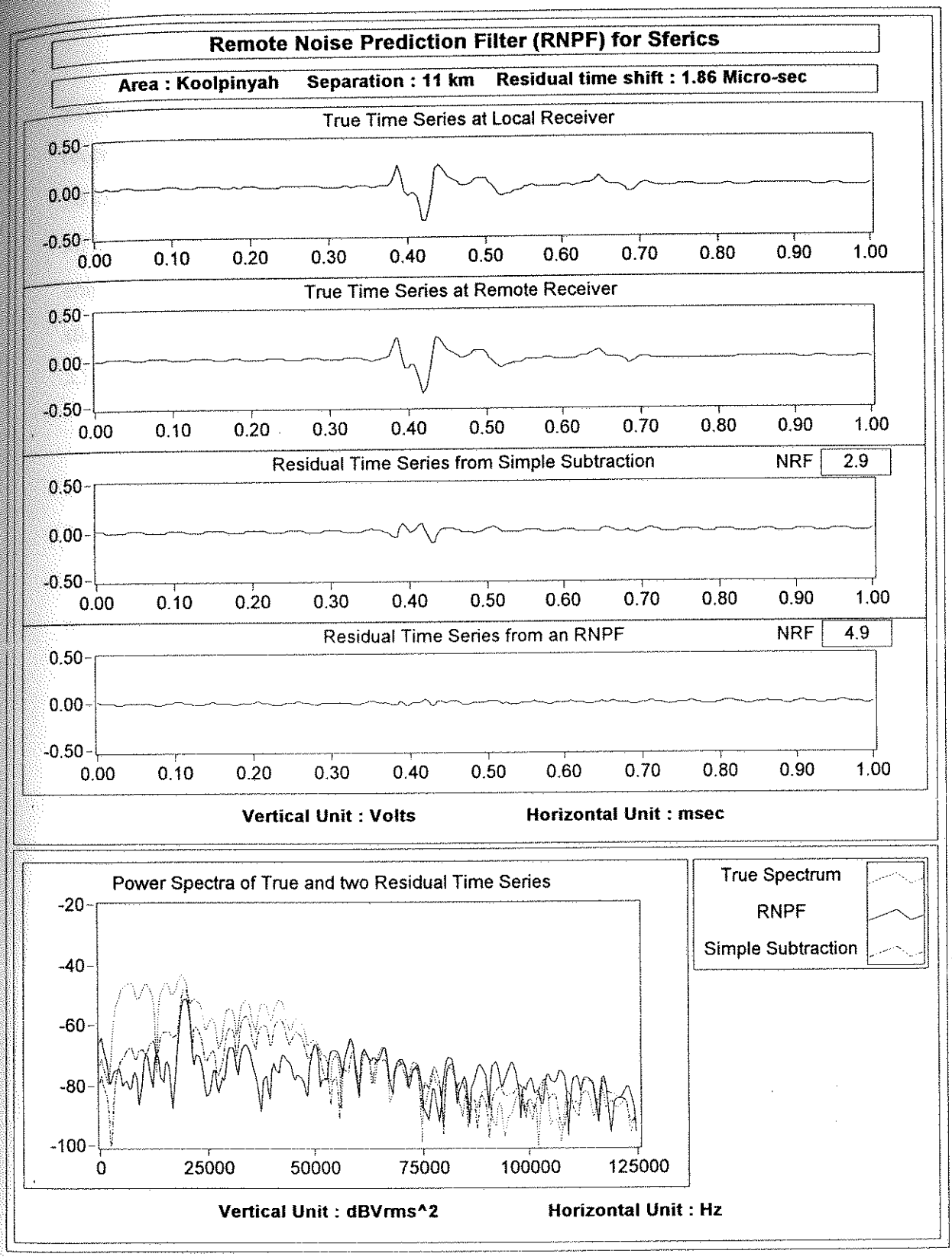

Figure 6.12 Example of performance of an RNPF and simple subtraction to reduce the X component of high-frequency sferics noise. Two sferics pulses were measured simultaneously at the local and remote receivers with a separation of 11 km and the residual time shift between the two sferics pulses is 1.86 μs.

frequency-domain, the power spectrum of the time series measured at the local receiver shows that the sferics pulse contributes power in the frequency range of ~5 to 50 kHz. At a frequency of ~9 kHz, the RNPF and simple subtraction reduce sferics by 30 (i.e., by a factor of ~32 in amplitude) and 20 dB (i.e., by a factor of 10 in amplitude), respectively. At a frequency of ~37 kHz, the neural network filter reduces sferics by 20 dB, while simple subtraction reduces sferics by 7 dB (i.e., by a factor of 2.2 in amplitude).

Figure 6.13 shows an example of attenuation with the neural network RNPF and simple subtraction methods, using the Y-component time series of a sferics pulse. The maximum amplitude of the sferics pulse is about ±3 V. The duration of the sferics pulse is about 1 ms. The residual time shift between the local and remote sferics pulses is -0.65 μs. *NRF* values of 9.7 and 20 are obtained from simple subtraction and the neural network RNPF, respectively. Effect of the residual time shift on simple subtraction is discussed in Section 6.3.3. At some frequencies, the neural network filter gives a much higher *NRF* than a factor of 20 in amplitude. For example, sferics power at 25 kHz is suppressed by 47 dB, i.e., an attenuation by a factor of 224 in amplitude. Probably, this noise is more coherent than average.

6.3.2 RNPF and an autoregressive moving average model

Suppose that pairs of observations (X_t, Y_t) are available at equispaced intervals of time, of an input X and an output Y from some dynamic system, which can be represented by a linear filter of the form

$$\begin{aligned} Y_t &= v_0 X_t + v_1 X_{t-1} + v_2 X_{t-2} + \cdots \\ &= \left(v_0 + v_1 B + v_2 B + \cdots\right) X_t \\ &= v(B) X_t \end{aligned} \tag{6.4}$$

in which the output at some time t is represented as a linear aggregate of the input at times t, t-1, ... The operator $v(B)$ is called the transfer function of the filter. The

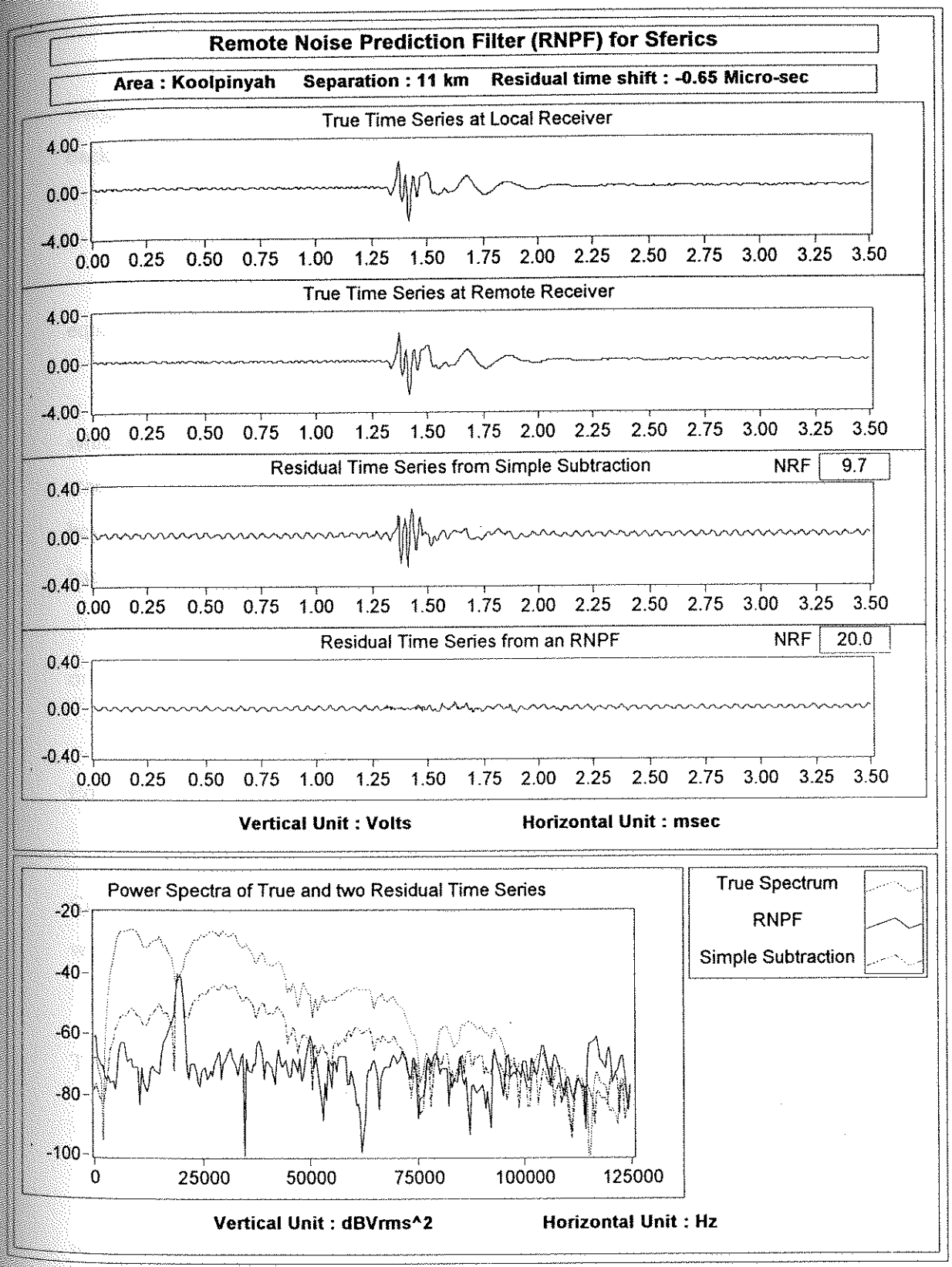

Figure 6.13 Example of performance of an RNPF and simple subtraction to reduce the Y component of high-frequency sferics noise. Two sferics pulses were measured simultaneously at the local and remote receivers with a separation of 11 km and the residual time shift between the two sferics pulses is -0.65 μs.

weights v_0, v_1, v_2,... are called the impulse response function of the system. B is the backward shift operator as follows:

$$BX_t = X_{t-1} \text{ and } B^j X_t = X_{t-j} .$$

Discrete dynamic systems can be represented by the general difference equation (Box, 1976; Bennett, 1979)

$$(1+\xi_1\nabla+\cdots\xi_r\nabla^r)Y_t = g(1+\eta_1\nabla+\cdots+\eta_s\nabla^s)X_{t-b} \tag{6.5}$$

which we refer to as a transfer function model of order (r,s), where g is called the steady state gain and b is pure delay or dead time before the response to a given input begins to take effect. The difference Equation 6.5 may also be written in terms of the backward shift operator B=1-∇ as

$$(1-\delta_1 B-\cdots\cdots-\delta_r B^r)Y_t = (\omega_0-\omega_1 B-\cdots\cdots-\omega_s B^s)X_{t-b} \tag{6.6a}$$

or

$$\delta(B)Y_t = \omega(B)X_{t-b} . \tag{6.6b}$$

Equivalently, writing $\Omega(B)=\omega(B)B^b$, the model (Equation 6.6b) becomes

$$\delta(B)Y_t = \Omega(B)X_t . \tag{6.7}$$

Comparing Equation 6.7 with Equation 6.4 the transfer function of this model is written by

$$v(B) = \delta^{-1}(B)\Omega(B) . \tag{6.8}$$

Thus, the transfer function is represented by the ratio of two polynomials in B.

An autoregressive integrated moving average (ARIMA) model

$$\varphi(B)z_t = \theta(B)a_t \tag{6.9}$$

used for the representation of a time series $\{z_t\}$ relates z_t and a_t by the linear filtering operation

$$z_t = \varphi^{-1}(B)\theta(B)a_t , \tag{6.10}$$

where a_t is white noise. Therefore, the ARIMA model postulates that a time series can be usefully represented as an output from a dynamic system to which the input is white noise and for which the transfer function can be parsimoniously expressed as the ratio of two polynomials in B.

In practice, the system will be infected by disturbances, or noise, whose net effect is to corrupt the output predicted by the transfer function model by an amount ε_t. The combined transfer function-noise model for predicting a local time series from a remote time series may then be written as

$$Y_t = \delta^{-1}(B)\omega(B)X_{t-b} + \varepsilon_t \,. \tag{6.11}$$

For testing whether or not the statistical model can be trained to accommodate any residual time shift smaller than the sample interval (4 μs), the transfer function model (Equation 6.6b) is used in this comparison. In order to apply a transfer function model for the prediction of a local time series from a remote time series, the input and output of Equation 6.6b can be replaced with remote and local sferics pulses, respectively, and b represents the time shift between the local and remote time series. The methods for identifying, fitting, and checking transfer function models are referred to in Chapter 11 of Box's book (1976).

Figure 6.14 shows an example of results obtained with the transfer function model when this model is applied to the remote X-component time series. Two X-component time series were simultaneously measured at stations separated by 11 km near Darwin. These time series are time shifted before doing identification (i.e., procedure to obtain some idea of the values of r and s in Equation 6.6 and to obtain initial guesses for the parameters) of the transfer function models (i.e., b=0). A residual time shift of -0.79 μs remains between the two series. The identification suggests the transfer function model of order (0,2) as follows:

$$Y_t = (\omega_0 - \omega_1 B - \omega_2 B^2) X_t$$

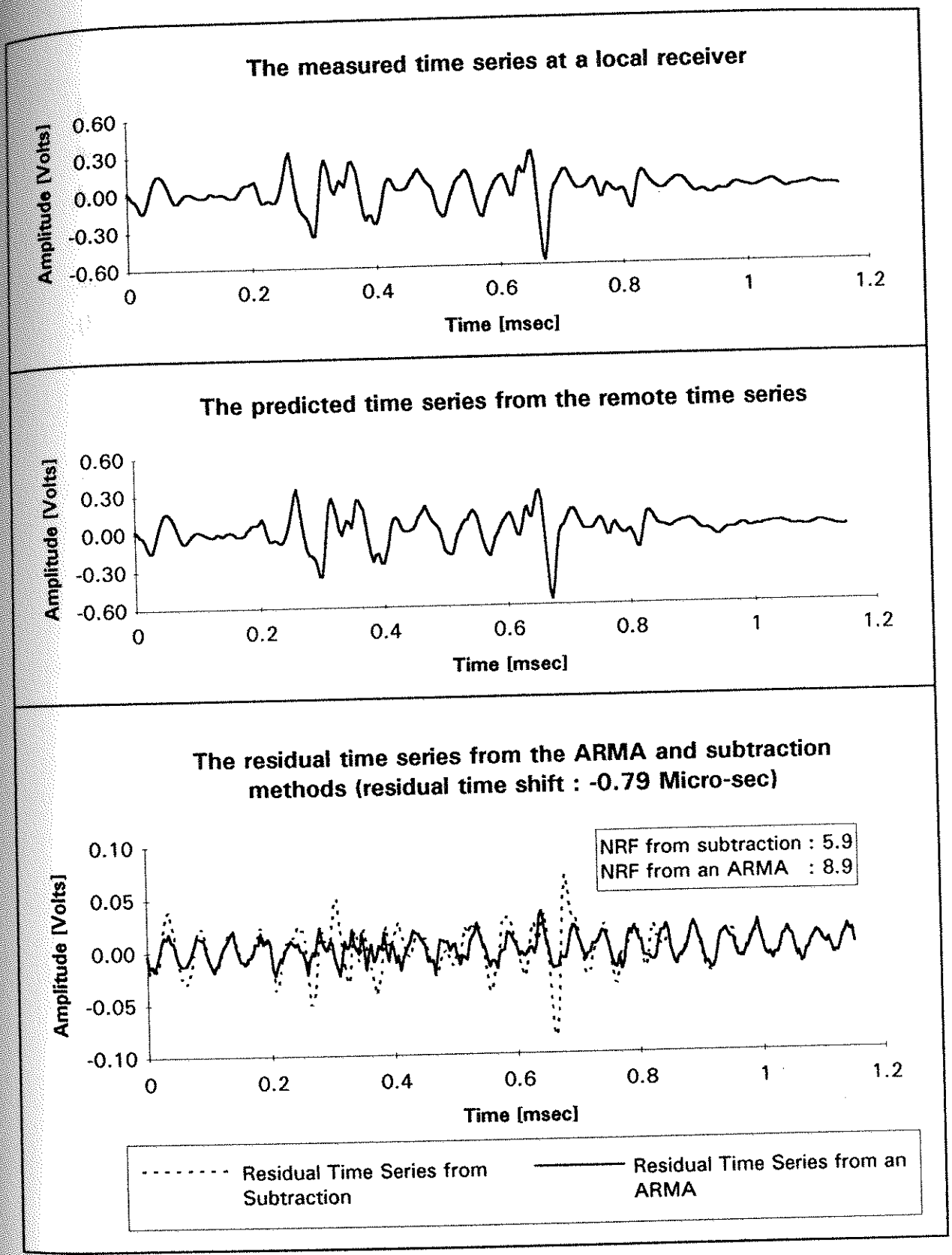

Figure 6.14 Example of performance of the transfer function model and simple subtraction to reduce the X component of high-frequency sferics noise. The residual time shift between the two sferics pulses is -0.79 μs.

for the prediction of a local time series from a remote time series. Furthermore, the initial estimates $\hat{\omega}_0$=0.96, $\hat{\omega}_1$=-0.46, and $\hat{\omega}_2$=0.23 can be used as rough starting values for the estimation procedures. The estimates of 1.26, 0.35, and -0.03 are obtained as ω_0, ω_1, and ω_2, respectively, by least-square minimisation. The transfer function model and simple subtraction method give *NRF* values of 8.9 and 5.9, respectively, (Figure 6.14) while a neural network RNPF achieves an *NRF* value of 9.6 (Figure 6.15).

Figure 6.16 shows the result of the transfer function model obtained above when it is applied to another time series that has the same residual time shift of -0.79 μs as the time series shown in Figure 6.14. In this case, the transfer function model shows consistent performance (an *NRF* value of 9.6). *NRF* values of 11.0 and 6.4 are obtained by the neural network RNPF and simple subtraction, respectively (Figure 6.17).

To test whether the transfer function model can accommodate any residual time shift between two sferics pulses measured simultaneously at local and remote stations, the trained transfer function model with a residual time shift of -0.79 μs was applied to the remote time series with a residual time shift of 1.71 μs with respect to the corresponding local time series. Figure 6.18 shows that the performance of the transfer function model is worse than that of simple subtraction. The transfer function model achieves an *NRF* of 1.6 while the simple subtraction method gives an *NRF* value of 2.3. It is therefore concluded that the transfer function model cannot be generally applied to predict the local sferics pulse with various residual time shift with respect to the corresponding remote sferics pulse. On the other hand, the neural network can be trained to accommodate any residual time shift between the local and remote sferics pulses (see Section 5.5).

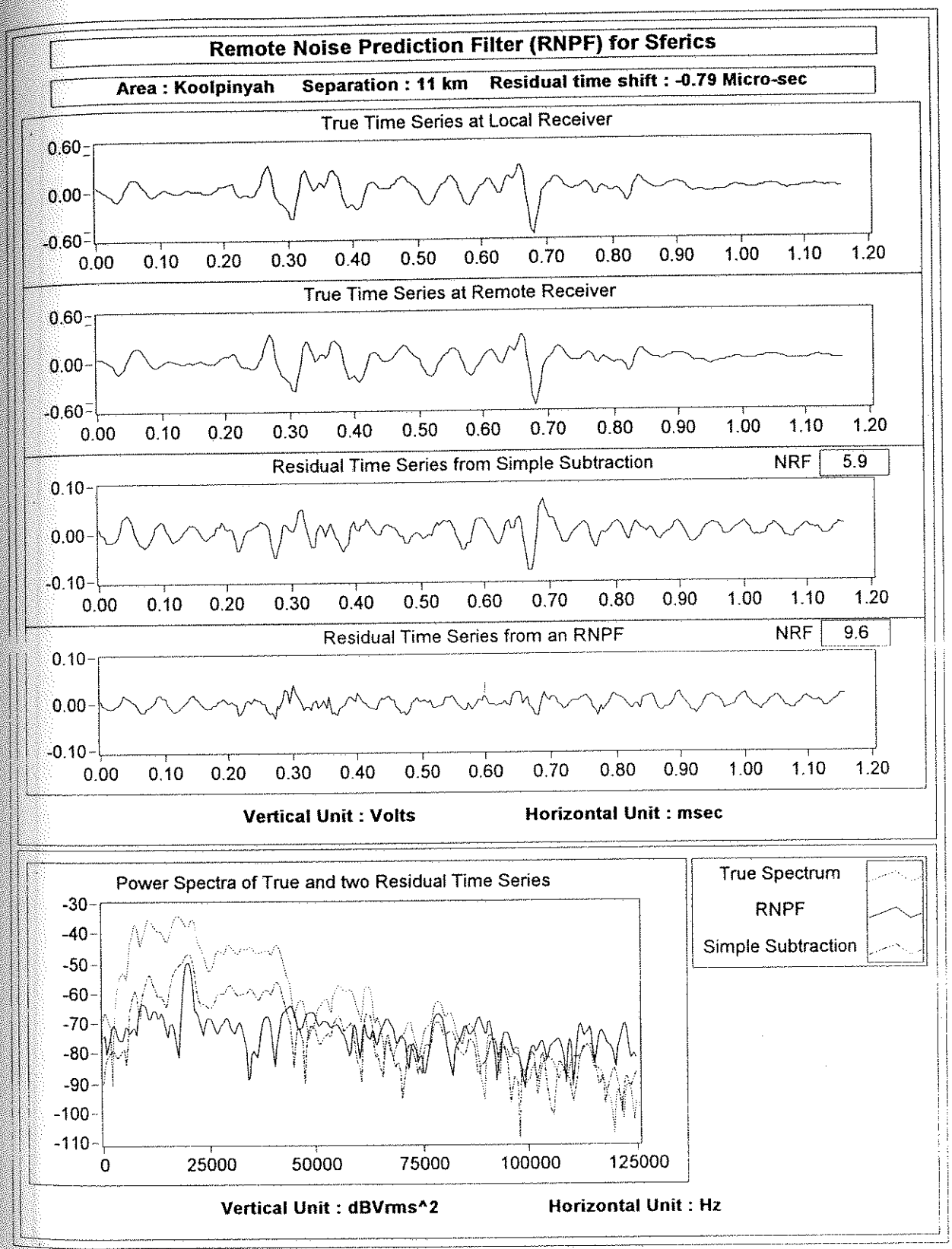

Figure 6.15 Example of performance of an RNPF and simple subtraction to reduce the X component of high-frequency sferics noise shown in Figure 6.14.

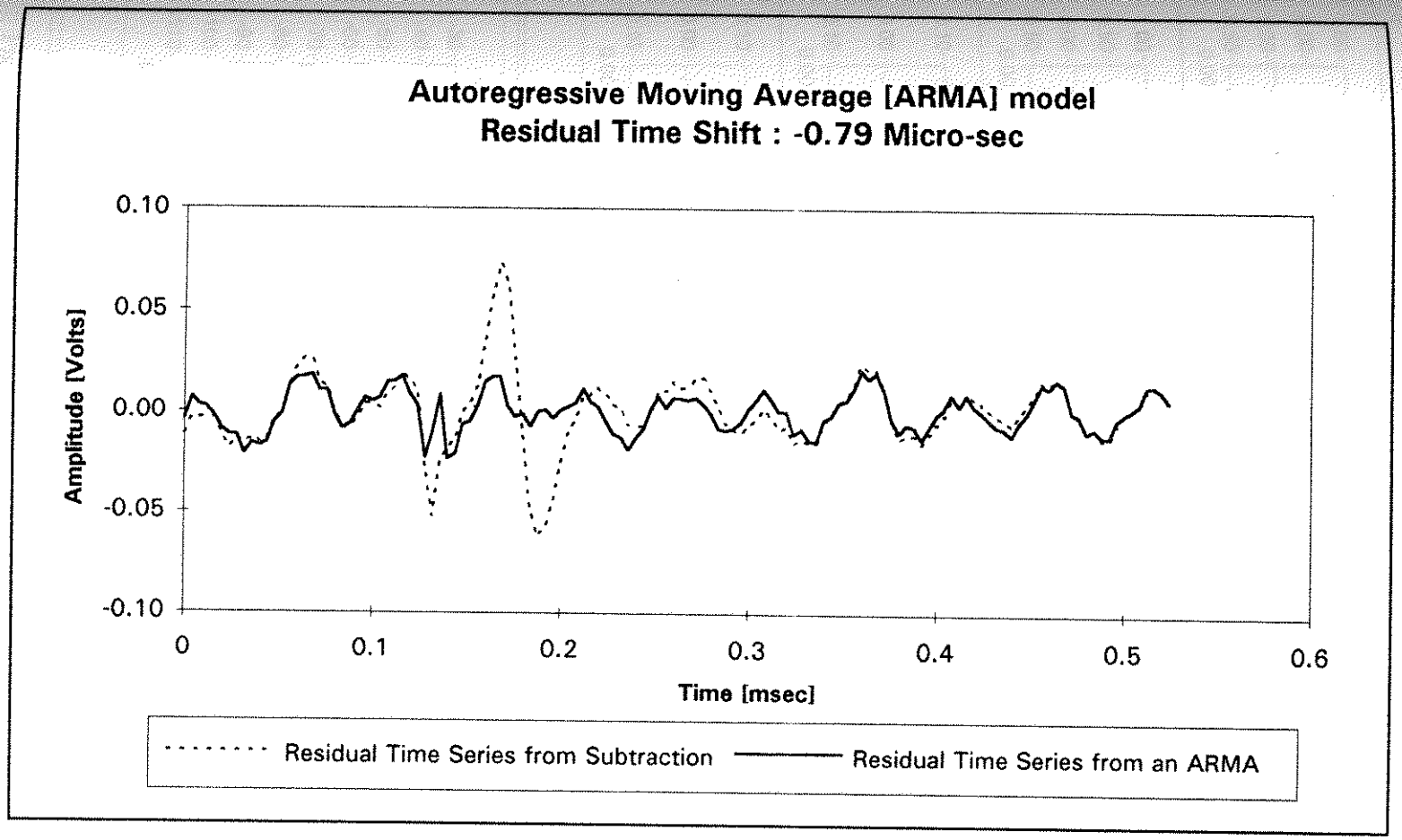

Figure 6.16 Example of performance of the transfer function model and simple subtraction when the trained transfer function model was applied to the remote sferics pulses with a residual time shift of -0.79 μs with respect to the corresponding local sferics pulse.

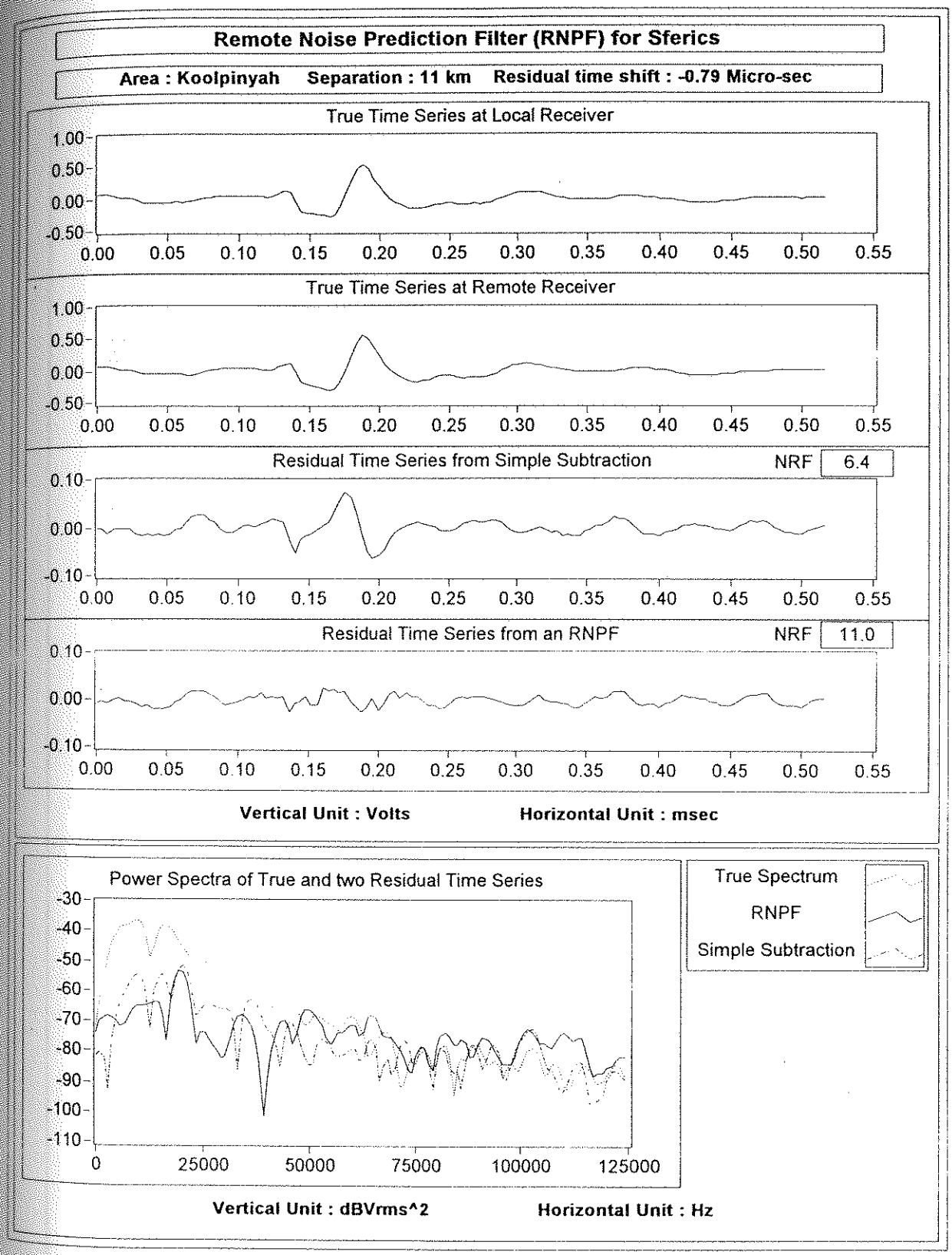

Figure 6.17 Example of performance of an RNPF and simple subtraction to reduce the X component of high-frequency sferics noise shown in Figure 6.16.

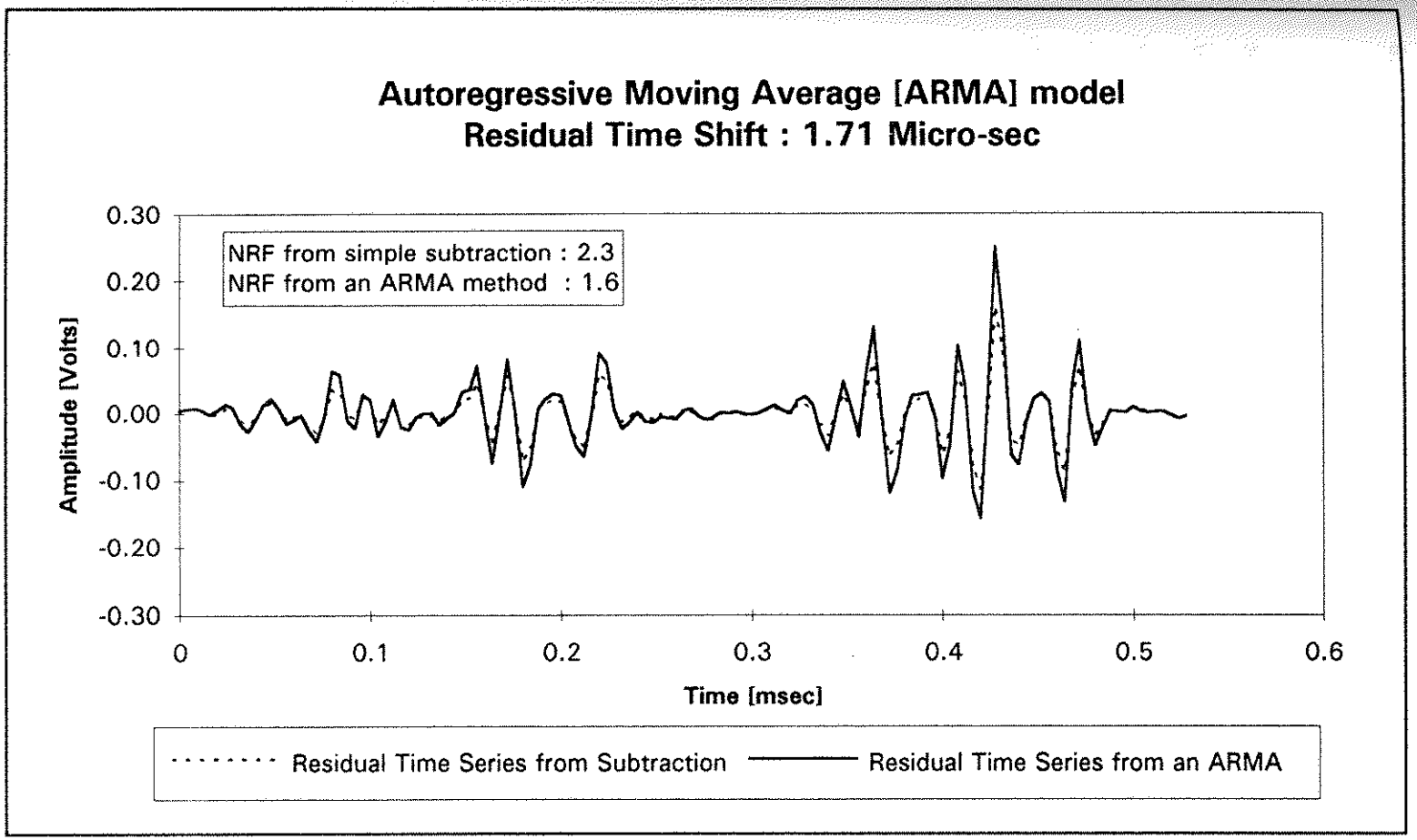

Figure 6.18 Example of performance of the transfer function model and simple subtraction when the trained transfer function model was applied to the remote sferics pulses with a residual time shift of 1.71 μs with respect to the corresponding local sferics pulse.

6.3.3 RNPF and the interpolation method

To investigate whether or not the performance difference between simple subtraction and the neural network LNPF is caused mainly by the residual time shift between two X-component time series at the local and remote stations, the two time series shown in Figure 6.14 (residual time shift of 0.79 μs in this case) were interpolated in time intervals of 0.4 μs and were compensated for the residual time shift of 0.8 μs (i.e., two times the interpolation interval) before subtraction of one interpolated time series from the other.

Figure 6.19 shows that the interpolation method achieves an *NRF* value of 7.8 without compensating for any difference between the characteristics of the sensors at the two stations. An *NRF* value of 6.3 is obtained from simple subtraction with compensation for both the residual time shift and sensor difference while the neural network RNPF achieves an *NRF* value of 9.6 (Table 6.2). Since in this case the total time shift between the local and remote sferics pulses has been essentially removed (apart from a residual time shift of 0.01 μs), the performance difference of 3.3 in the *NRF* value is considered to arise from an amplitude difference caused by different ground conductivity at the two stations.

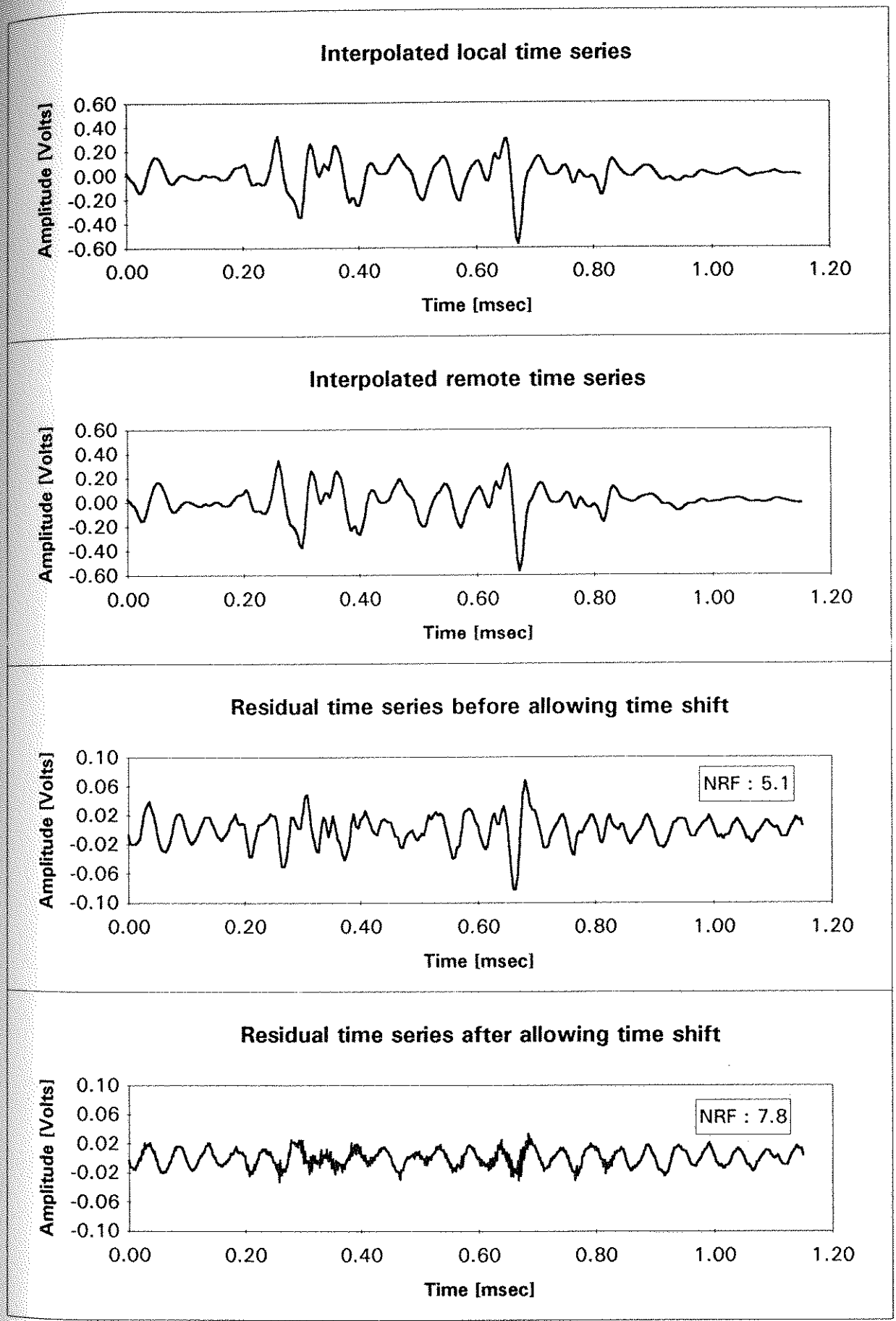

Figure 6.19 Example of performance of an RNPF and interpolation method. The local and remote sferics pulses with a residual time shift of -0.79 μs shown in Figure 6.14 were interpolated at time intervals of 0.4 μs.

Table 6.2 Summary of a remote-reference technique for the X component of sferics pulse shown in Figure 6.14 using simple subtraction, autoregressive moving average, neural network, and interpolation methods.

		Without interpolation		With interpolation	
		Before compensating for sensor difference	**After compensating for sensor difference**	**Before compensating for sensor difference**	**After compensating for sensor difference**
Simple subtraction	Before compensating for the residual time shift	5.9	5.0	5.1	4.6
	After compensating for the residual time shift	N/A	N/A	7.8	6.3
ARMA		8.9	N/A	N/A	N/A
Neural network		9.6	N/A	N/A	N/A

6.4 Conclusion

In the reduction of EM noise on a noisy single transient, the model training method (MTM) has been compared with the simple stacking and least-square methods. The first and second order power rules are used. The performance of the MTM at early delay times is equal to the performance of the least-square fitting while at late delay times, the MTM performs better than the least-square fitting method by a factor of 2.5.

In the comparison between the Spies and neural network local noise prediction filters (LNPF) for reduction of low-frequency variations of the geomagnetic field, the neural network LNPF is superior to the Spies LNPF based on the tipper method with three-point prediction filter. The Spies LNPF with 48-point prediction filter is only marginally worse, but takes 54 minutes to train, whereas the neural network based LNPF takes only less than 10 minutes to train.

For reduction of high-frequency (above 5 kHz) sferics, the Spies LNPF with 24-point prediction filter performs worse when the sferics pulse has approximately equal amplitudes in both the X and Y components. In cases where the sferics pulse is polarised mainly along the X or Y direction, the performance of the Spies filter is only marginally worse than the performance of the neural network LNPF.

In the use of a remote reference technique for reducing EM noise (VLF and sferics), simple subtraction, a transfer function model (e.g., autoregressive model), and interpolation method have been compared with a neural network based remote noise prediction filter (RNPF).

For reduction of background EM (VLF) noise, both the neural network RNPF and simple subtraction reduce VLF (~20 kHz) noise by 20 dB (i.e., by a factor of 10 in amplitude). In all horizontal (X and Y) component reductions of high-frequency

sferics, the performance of the neural network RNPF is superior to the performance of simple subtraction. The performance difference between simple subtraction and the neural network RNPF is caused by both the residual time shift between the local and remote time series, and amplitude differences between the two time series caused by a lateral conductivity change between the local and remote stations.

The performance of a transfer function model (e.g., an autoregressive integrated moving average model) is the worst when the residual time shift is different from the residual time shift between the local and remote time series used in the training of the transfer function model. Otherwise, the performance of the transfer function model is only marginally worse than the performance of the neural network RNPF.