

Deconvolution of System Impulse Responses and Time Domain Waveforms

James R. Andrews, Ph.D., IEEE Fellow
PSPL Founder & former President (retired)

INTRODUCTION — CONVOLUTION A classic deconvolution measurement situation is shown in Figure 1. Some excitation, say $v_{in}(t)$, is applied to a 'black box', $h(t)$, and then some output response occurs, $v_{out}(t)$. One situation might be the system identification of the black box's transfer function from measurements of both $v_{in}(t)$ and $v_{out}(t)$. The result is the black box's impulse response, $h(t)$. $h(t)$ is the output from the black box when the input excitation is a unit impulse. The second situation is the determination of an unknown input signal, $v_{in}(t)$, when one is only able to observe the signal, $v_{out}(t)$. If one knows $h(t)$, then $v_{in}(t)$ can be determined.

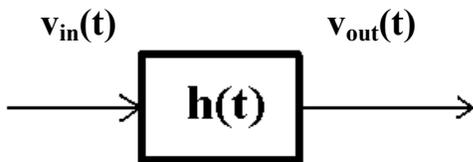


Fig. 1 Typical system problem
What is $v_{in}(t)$ or $h(t)$?

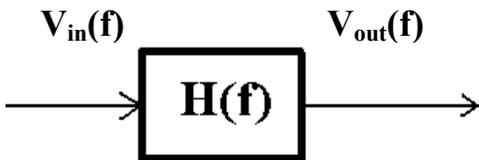


Fig. 2 Frequency domain system

This process is much clearer for most engineers when using sine waves in the frequency domain. Figure 2 shows the same situation for the frequency domain, except that upper case letters are used to denote the input, output, and transfer function. A simple equation describes the situation in Figure 2.

$$V_{out}(\omega) = V_{in}(\omega) \times H(\omega) \quad (1)$$

or
$$H(\omega) = V_{out}(\omega) / V_{in}(\omega) \quad (2)$$

where $H(\omega)$ is the transfer function. In matched impedance situations, $H(\omega)$ is also called the "S" parameter, $S_{21}(\omega)$. In the frequency domain, all of the parameters can be complex, but equations (1) or (2) are simple multiplication or division.

Returning to the time domain, Figure 1, the 'Convolution' integral equation describes the relationship between $v_{in}(t)$, $v_{out}(t)$, and $h(t)$.

$$v_{out}(t) = \int_0^t v_{in}(\tau) h(t-\tau) d\tau \quad (3)$$

Conceptually this convolution equation is harder to visualize and also to compute. The process of determining either $v_{in}(t)$ or $h(t)$ is called 'Deconvolution'.

Fortunately, both the time domain, Figure 1, and the frequency domain, Figure 2, situations are related. Depending upon the type of signals, either LaPlace transforms, Fourier transforms, or Fourier series provide the conversions between the two domains. When dealing with finite, discrete data sets, the Discrete Fourier Transform (DFT) is used. Today, a special, fast, computer algorithm, called the Fast Fourier Transform (FFT) is used to compute the DFT. Thus, the various above parameters are all related by

$$V_{in}(\omega) = \text{FFT}[v_{in}(t)] \quad (4)$$

$$V_{out}(\omega) = \text{FFT}[v_{out}(t)] \quad (5)$$

$$H(\omega) = \text{FFT}[h(t)] \quad (6)$$

We can make our measurements in the time domain but find our calculations easier to perform in the frequency domain.

DECONVOLUTION For the system identification deconvolution problem, we thus have from equations 2, 4, and 5

$$H(\omega) = \text{FFT}[v_{out}(t)] / \text{FFT}[v_{in}(t)] \quad (7)$$

$$h(t) = \text{invFFT}[H(\omega)] \quad (8)$$

where 'invFFT' denotes the Inverse FFT operation. It should be noted that equation 7 is the basis for the Time Domain Vector Network Analyzer (TDVNA) described in PSPL's application note, AN-16a [1].

For the input signal determination deconvolution problem, we thus have

$$V_{in}(\omega) = \text{FFT}[v_{out}(t)] / \text{FFT}[h(t)] \quad (9)$$

$$v_{in}(t) = \text{invFFT}[V_{in}(\omega)] \quad (10)$$

MATH PROBLEMS Conceptually, equations 7-10 look to be extremely simple and straight-forward to implement. In actual practice, however, their solutions are far from simple. Why? The problem lies in the frequency domain division operation in equation 7 or 9. When the denominator becomes very small, non-existent, or simply noise, then the division operation blows up and goes to infinity. Likewise, if both the numerator and denominator both become very small, non-existent, or simply noise, then the result is an indeterminate number. The presence of these very large, noise based numbers totally destroy the calculations in the next step (equation 8 or 10) when the inverse FFT is taken. The end result is an extremely noisy, incorrect waveform. Programs to do deconvolution typically require the introduction of an additional filter, $F(\omega)$, to keep under control the division problems. The result, $H_e(\omega)$ is an estimate of the true $H(\omega)$.

$$H_e(\omega) = F(\omega) \times \{\text{FFT}[v_{out}(t)] / \text{FFT}[v_{in}(t)]\} \quad (11)$$

Likewise, for the deconvolution of $v_{in}(t)$ signals, we also need to use a filter.

$$V_{in}(\omega) = F(\omega) \times \{\text{FFT}[v_{out}(t)] / \text{FFT}[h(t)]\} \quad (12)$$

All real world signals are ultimately corrupted by noise. They thus have finite signal to noise ratios. Figure 3 shows a typical impulsive time domain signal along with its frequency domain spectrum. The measurement system noise level is seen to be about -40dB μ V/MHz. Thus, for frequencies above about 65GHz, the spectral content of this impulsive signal is pure noise when used in equation 7.

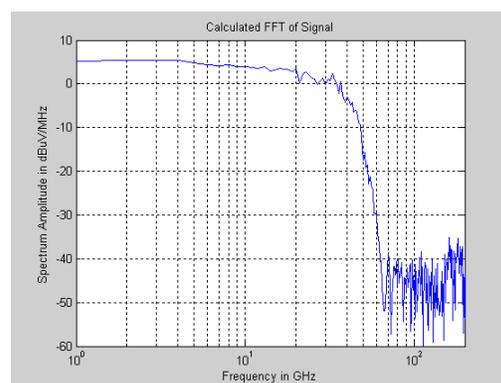
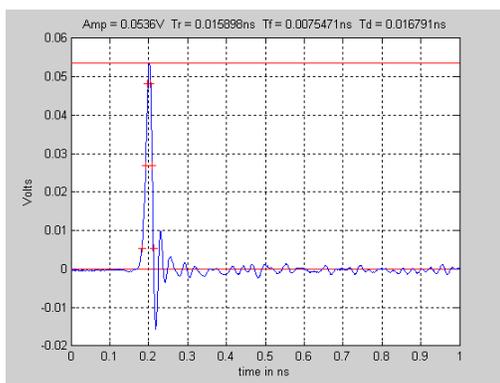


Fig. 3 A 54mV, 17ps impulse and its spectrum. Pulse generator was a PSPL 4020 TDR pulse head with a 5206 IFN. Measured by HP-54750A, 50 GHz oscilloscope.

NBS and PSPL Researchers have wrestled with this noisy deconvolution problem for many years. Many papers have been written on the subject. Prof. Dr. N.S. Nahman and his graduate students have been key figures in this research. His early work was at NBS, Boulder, Colorado (now NIST). His NBS group was responsible for the national standards for picosecond time domain pulse measurements. To accomplish this, his goals were to characterize the impulse response of NBS's ultra-wideband oscilloscope and then in turn use deconvolution to estimate the true waveforms of the pulse generators submitted to NBS for calibration [2]. His most important deconvolution papers are listed in the reference section [3-5]. Nahman's deconvolution techniques are implemented in the PSPL deconvolution programs. Nahman's NBS goals of yesterday are the same as PSPL's goals today. As PSPL continues to make faster risetime pulsers and samplers, we need math software tools, such as deconvolution, to enable us to perform more accurate time domain measurements of our products.

MATLAB DECONVOLUTION PROGRAMS

Two programs have been written in MatLab [6] to perform deconvolution. They are enhanced versions of earlier PSPL programs written by Martin VanPelt in 1989. *Hdecon.m* determines the impulse response, $h(t)$ from measured input and output waveforms. *VinDecon.m* determines the input waveform when given the output waveform and the system impulse response. Before proceeding further, the reader should read at this time PSPL's application note, AN-16a [1]. The deconvolution programs build upon first the spectrum analysis program *SpecAnalysis.m* and next the time domain vector network analyzer program, *TDVNA.m* discussed in AN-16a. The *SpecAnalysis* program performs the FFT operation, equation 4, while the *TDVNA* program performs the H or S_{21} division operation, equation 7.

All of these programs include a lot of internal 'comments' and references to explain the reasons for performing various operations. They can be read easily with a simple .txt reader.

These programs interact with the operator via the monitor screen and the keyboard. They provide lots of plots of intermediate steps to allow the operator to monitor the progress and gain insight into the program's functioning. The operator is allowed to alter the manner in which data is treated. Data can be input/output via either drive A: or C: Subtraction of baselines is optional. Input data is immediately plotted on the monitor to verify that the correct data was input. Next, the frequency spectrum of the input data is plotted. The next step is to apply a low pass filter, F , (equation 11) to the data to control the noise/noise division problem. The program gives the operator a selection of various types of filters, and he inputs a cut-off frequency. Next, the filter's frequency and time domain properties are plotted. The program then applies the filter to $H(\omega)$ or $V_{in}(\omega)$ and performs the inverse FFT to return the desired $h(t)$ or $v_{in}(t)$, which are then plotted on the monitor.

Both deconvolution programs are designed to work with either impulse-like or step-like waveforms. For either to function properly, the time window of observation needs to be wide enough for all transients to die out. For an impulse signal, it must return back to its initial baseline by the end of the time window, T_w . When a step-like waveform is used, it requires special treatment to avoid 'windowing' problems in the FFT. These programs use Riad's composite treatment [7]. This entails creating two new waveforms. One is a square wave, using the Andrews-Gans-Nahman technique [8,9]. The other is a subtracted ramp waveform, using the Nicholson technique [10]. The FFTs of both these new waveforms are then merged to create the composite step spectrum. Details of this operation are discussed in AN-16a [1].

Several filter options are included. One option is no filtering. This is included only so the operator can experience for himself the noise/noise disaster. The next filter is a Rectangular filter. This is a brick-wall filter with

$$F_R(\omega) = \begin{cases} 1 & \text{for } f < f_{co} \\ 0 & \text{for } f > f_{co} \end{cases} \quad (13)$$

The third filter is a Gaussian filter of variable order, m

$$F_G(\omega) = \exp[-0.5x(f/f_{co})^m] \quad (14)$$

The fourth filter is the Nahman-Guillaume Optimum Filter [3,4]. This is an adaptive filter that takes into account the spectral variations in the actual data, $v_{in}(t)$ or $h(t)$. The filter's equations are

$$c2 = 16 [\sin(\pi (0:N-1)/N)]^4 \quad (15)$$

$$X2 = \text{abs}[V_{in}(\omega)]^2 \quad (16)$$

$$F_{NG} = X2 / (X2 + (\gamma c2)) \quad (17)$$

The weighting parameter, γ , is user selectable. The programs also include an optional automatic γ optimization routine that minimizes the error in either $h(t)$ or $v_{in}(t)$. Error minimization is determined by minimizing the residual energy in the imaginary part of $h(t)$ (or $v_{in}(t)$) that is returned by the final inverse FFT operation. Ideally there should be no energy in the imaginary part. Any energy there is to be considered an error.

The fifth filter is the Bennis-Nahman minimum phase filter [5]. This is also an adaptive filter and is an extension of the Nahman-Guillaume adaptive filter. It uses a Hilbert transform technique [11] to convert the N-G filter into a causal, minimum phase filter. The weighting parameter, γ , is either user selectable or auto optimized.

EXAMPLES The following figures will demonstrate the use of these deconvolution programs. The programs output these various plots. The network to be evaluated was a PSPL model 5933-8GHz Low Pass Filter. The input test signal, Figure 4, was a 192mV step pulse from a PSPL 4020 TDR pulse head. The measurement oscilloscope was a 50GHz, HP-54750A oscilloscope with an 8.8ps risetime. The measured risetime of the pulser was 12.9ps. Its actual risetime was estimated to be 9.4ps.

The first example is the use of the program *Hdecon.m* to determine the impulse response, $h(t)$, of a network. For the measurement on the low pass filter, a time window, T_w , of 1ns was used. 1024 data points were taken. 64 digital averages were taken. The sampling time increment, Δt , was 0.98ps. The Nyquist folding frequency was 512GHz. Figure 5 shows the input and output waveforms. 0V baseline signals were also measured to enhance the quality of the measurement. These baselines were subtracted from the in/out waveforms. Figure 6 shows the spectrums of these various time domain waveforms.

Next, the deconvolution program calculates the simple transfer function, $H(f)$, using the division of the FFTs of V_{in} and V_{out} , equation 7. Figure 7 vividly demonstrates how this division of noise/noise results in an incorrect answer for $H(f)$ for high frequencies where the spectral energy of the input signal has dropped into the baseline noise level, Figure 6.

The program next allows the operator to apply an optional low pass filter to the noisy data of Figure 7. For this example, the Bennis-Nahman, optimal, minimum phase filter was selected. The best results were obtained using a γ weighting factor of -112dB. The resultant filter is shown in Figures 8 and 9. Its -3dB bandwidth was 50GHz.

Finally, the Bennis-Nahman filter, Figure 8, was applied to the raw, unfiltered spectrum of Figure 7, using equation 11, giving the corrected 5933 transfer function, $H_e(f)$, shown in Figure 10. The 5933's -3dB bandwidth was found to be 7.62GHz. Figures 11-13 show the calculated impulse response, $h(t)$, and the step response of the 5933 Low Pass Filter. The impulse response duration (FWHM) was 42.5ps. The step response 10%-90% risetime was 44.4ps. Thus, the risetime-bandwidth product for this filter was 0.338.

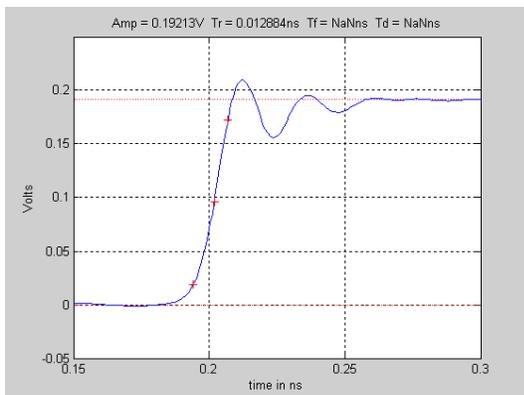


Fig. 4 Pulse from PSPL model 4020 TDR pulse head as measured by an HP 50GHz sampling oscilloscope. 192mV, 9.4 ps risetime.

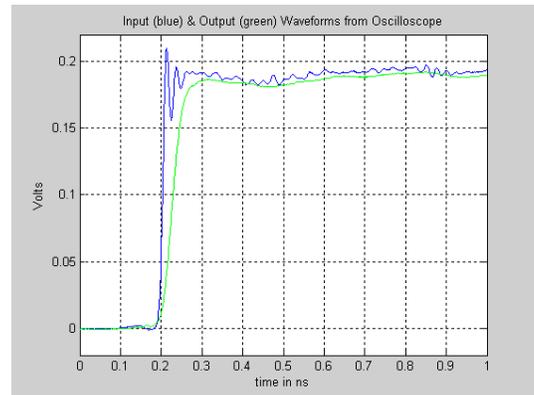


Fig. 5 Test of PSPL model 5933-8GHz Low Pass Filter. Input and output waveforms.

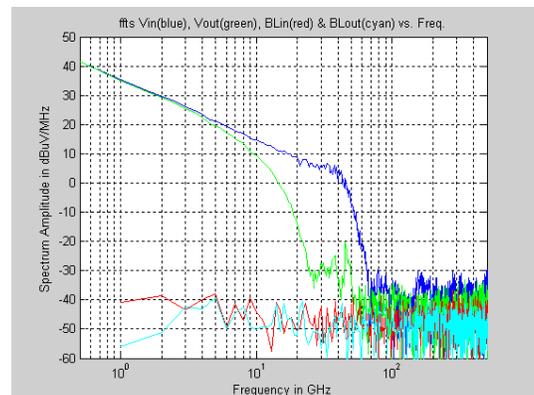


Fig. 6 Spectrums of input and output waveforms plus the baselines.

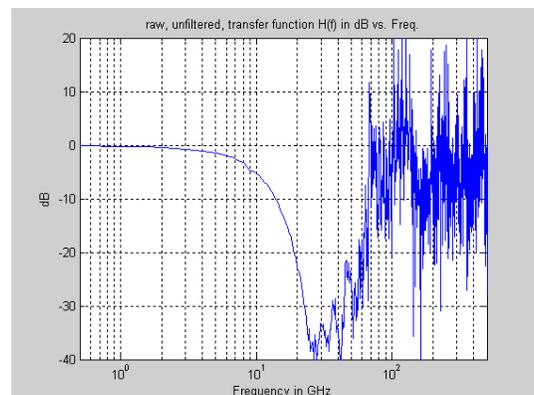


Fig. 7 $H(f)$ transfer function of 5933, using simple $\text{fft}(V_{out}(t)) / \text{fft}(V_{in}(t))$

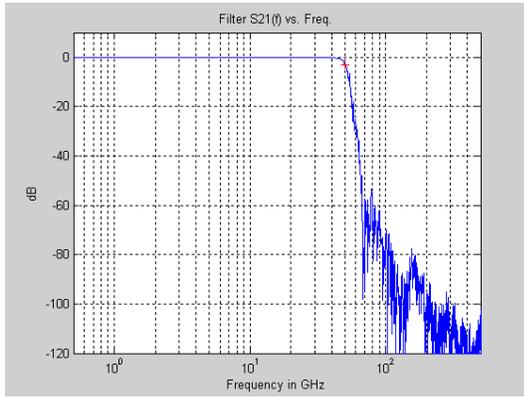


Fig. 8 $F(f)$ transfer function of Bennis - Nahman optimal, minimum phase filter. $\gamma = -112\text{dB}$, $\text{BW} = 50\text{GHz}$.

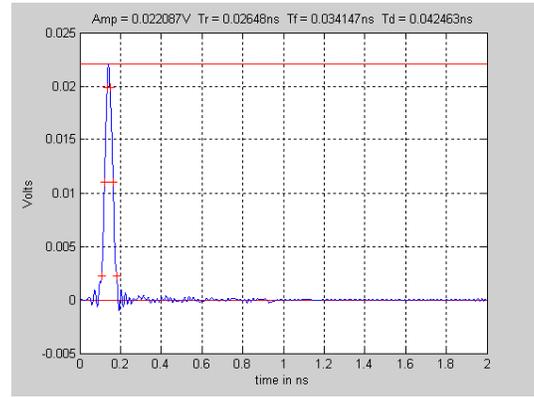


Fig. 11 Impulse Response, $h(t)$, of 5933 Low Pass Filter. 42.5ps duration.

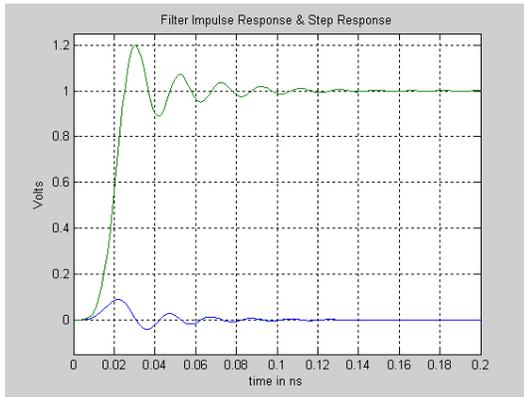


Fig. 9 Impulse and Step response of Bennis - Nahman optimal, minimum phase filter. $\gamma = -112\text{dB}$.

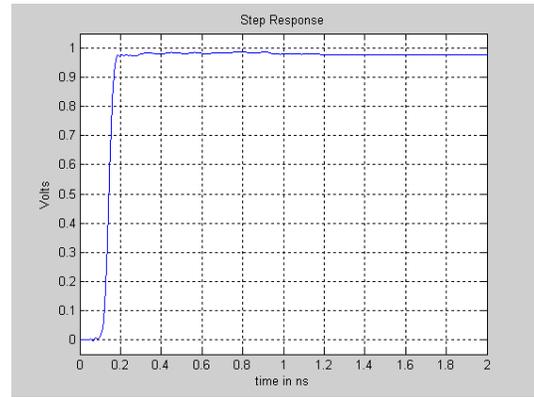


Fig. 12 Step Response of 5933 Low Pass Filter.

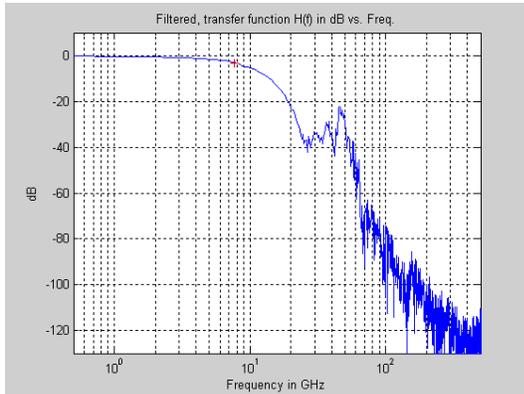


Fig. 10 Corrected transfer function, $H_e(f)$, of 5933 Low Pass Filter. $\text{BW} = 7.62\text{GHz}$.

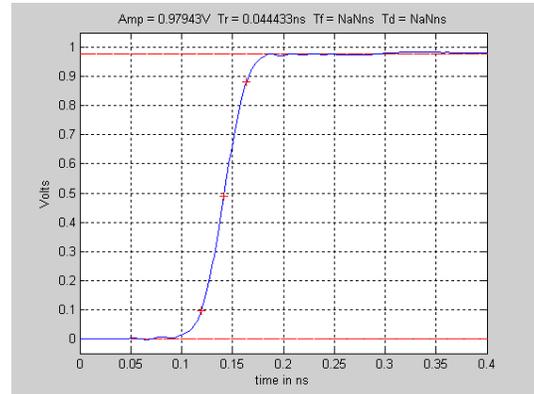


Fig. 13 Step Response of 5933 Low Pass Filter on expanded scale with measurements. 44.4ps risetime.

The second example uses the program *VinDecon.m* to determine an unknown input signal. For this example, we again use the 5933 Low Pass Filter. For the impulse response, we use the $h(t)$ determined previously. See Figure 11. For the output signal, we use $v_{out}(t)$ shown in Figure 5. To obtain a stable solution for $v_{in}(t)$ also requires the use of a low pass filter in equation 12. For this example, the Bennis-Nahman filter was used with a weighting factor of -26dB. The bandwidth of this filter was 50GHz. Figures 14 and 15 show the computed input signal, $v_{inc}(t)$. In Figure 15, $v_{inc}(t)$ is compared with the original $v_{in}(t)$. If these programs were perfect, these two waveforms would have been identical. They are seen to be very similar, but not identical. The risetime of the original $v_{in}(t)$ was 12.9ps, while the computed $v_{inc}(t)$ was 14.6ps. Also, $v_{inc}(t)$ is seen to exhibit a bit more pulse topline ringing than the original pulse. Still, if one has some knowledge of the impulse response of a network, the process of deconvolution allows one to greatly enhance the effective measurement bandwidth in determining an unknown input signal.

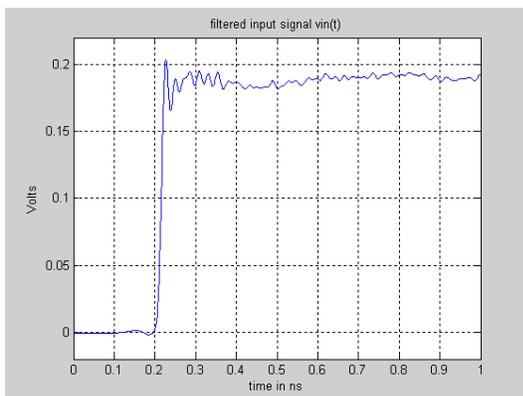


Fig. 14 Reconstructed input test signal after applying the program, *VinDecon.m*.

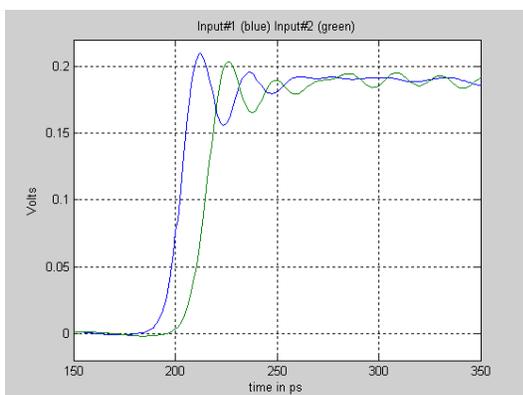


Fig. 15 Comparison of original input signal $v_{in}(t)$ (blue) with the reconstructed input signal $v_{inc}(t)$ (green).

MATLAB PROGRAM APPENDIX The most current versions of the actual *Hdecon.m*, *VinDecon.m*, and *PulseMeas.m* MATLAB programs are available for download from the PSPL website www.picosecond.com. They are written as simple MatLab *.m (i.e., .txt) files. They can also be obtained directly from the author at: jrandrews@picosecond.com.

REFERENCES

- [1] J.R. Andrews, "Time Domain Spectrum Analyzer and 'S' Parameter Vector Network Analyzer", Application Note AN-16a, Picosecond Pulse Labs, Boulder, CO, Nov. 2004, www.picosecond.com.
- [2] W. Gans, N.S. Nahman, J.R. Andrews & E.E. Baldwin, *Pulse Transition Duration Measurements and Standards at NBS --- 1975 to 1988*, NISTIR - 6550, NIST, Gaithersburg, MD, Aug. 2000, 127 pages.
- [3] N.S. Nahman & M. Guillaume, *Deconvolution of Time Domain Waveforms in the Presence of Noise*, NBS Tech.Note 1047, NBS, Boulder, Colo Oct. 1981.
- [4] N.S. Nahman, "Software Correction of Measured Pulse Data" *Fast Electrical & Optical Measurements*, Vol.I, pp.351-417, 1986 NATO ASI Series, Martin Nijhoff Publishers.
- [5] A. Bennis & N.S. Nahman, "Deconvolution of Causal Pulse & Transient Data", *IEEE Trans. I&M*, vol. 39, no.4, Dec. 1990, pp.933-939.
- [6] MATLAB, Math Works, Natick, MA, www.mathworks.com.
- [7] A. Shaarawi & S. Riad, "Computing the Complete FFT of a Step-Like Waveform", *IEEE Trans. Inst.& Meas.*, vol IM-35, no.1, pp.91-92, March 1986.
- [8] W. Gans & J.R. Andrews, *Time Domain Automatic Network Analyzer for Measurement of RF & Microwave Components*, NBS Technical Note 672, NBS, Boulder, CO, Sept. 1975, 176 pages.
- [9] W.L. Gans & N.S. Nahman, "Continuous and discrete Fourier transform of step-like waveforms", *IEEE Trans. Inst. & Meas.*, vol.IM-31, pp.97-101, June 1982.
- [10] A.M. Nicolson, "Forming the fast Fourier transform of a step response in time-domain metrology", *Electron.Lett.*, vol.9, pp.317-318, July 1973.
- [11] A. Oppenheim & R. Schaffer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [12] plus many of the references in AN-16a [1].