Analysis of Impulse Response for a linear system

Processes the Impulse response to obtain a frequency-domain "transfer function," which is then used to solve for the response to an arbitrary driving function.

Last modified by Edward Eyler, Dec 4, 2005. Tested with Mathcad 11, saved for MathCad 8.

First, read a data file containing the measured impulse response, obtained from an A/D converter on a computer. (Note: for this example spreadsheet, evaluation of the first four formulas has been disabled, so that a data file is not needed to display representative results.)

$Data :=$ A:\Data.tx

Use the data file input component to read the data

$Data_0 := Data_1$

Throw away the first data point (often not valid)

$t := 0 .. \mathrm{last}(Data)$

$Data := Data - Data_{\mathrm{last}(Data)}$

Subtract the offset of the equilibrium position, here assumed to be represented accurately by the last data point.

For example purposes, substitute a perfect damped sine wave, which is the impulse response of an ideal under-damped harmonic oscillator.

$t := 0 .. 7000$

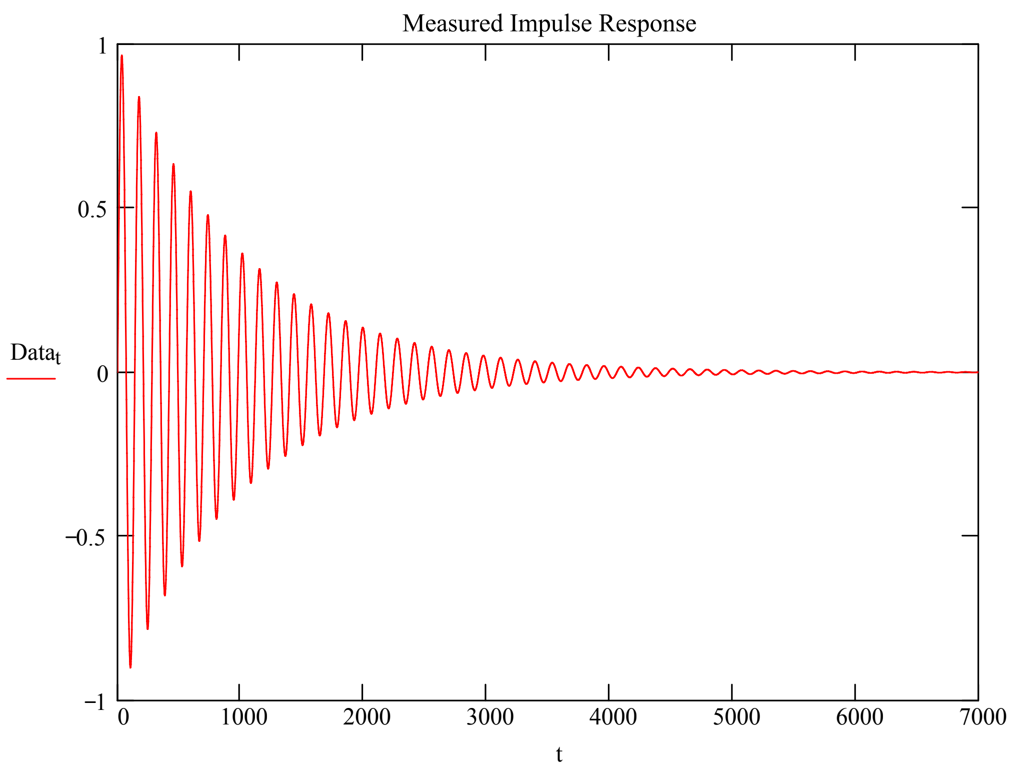
(Length would normally be supplied by the formula above.)

$\phi := 0.0$

This variable is provided so that you can experiment with phase offsets.

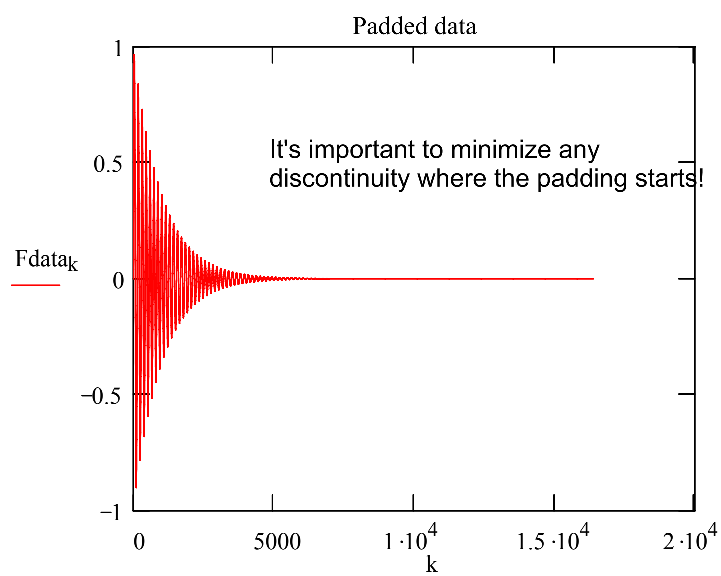
$$Data_t := \sin\left[\left(2\cdot\pi\cdot\frac{t}{140}\right) + \phi\right]\cdot e^{\frac{-t}{1000}}$$

Plot the data to make sure things look reasonable. The plot can also be used to directly determine the period of oscillation and the exponential decay time constant (the former, by measuring the zero-crossings the latter, by reading off the maxima for each oscillation, then fitting to an exponential). Note that the "zoo and "trace" features of MathCad, available by right-clicking on the graph, are extremely useful for direct quantitative measurements of graphical data.



Define a "padded" data array for the Fourier transform (length must be a power of 2)

$k := 0..2^{14}-1$ $\qquad$ $Fdata_k := 0$ $\qquad$ $Fdata_t := Data_t$



Now take the Fourier transform. The FFT algorithm returns essentially the same result as the familiar continuous Fourier integral. However, a few notes may be helpful:

1. If the input array Fdata has time steps of size Δt, the frequency-domain output has frequency steps of size Δf = (number of elements in input array)/Δt. See the plot below for clarification.
2. The output is an array of complex numbers, specifying both the magnitude and the phase of the response as a function of frequency.
3. If an inverse transform is to be performed, use the function IFFT.

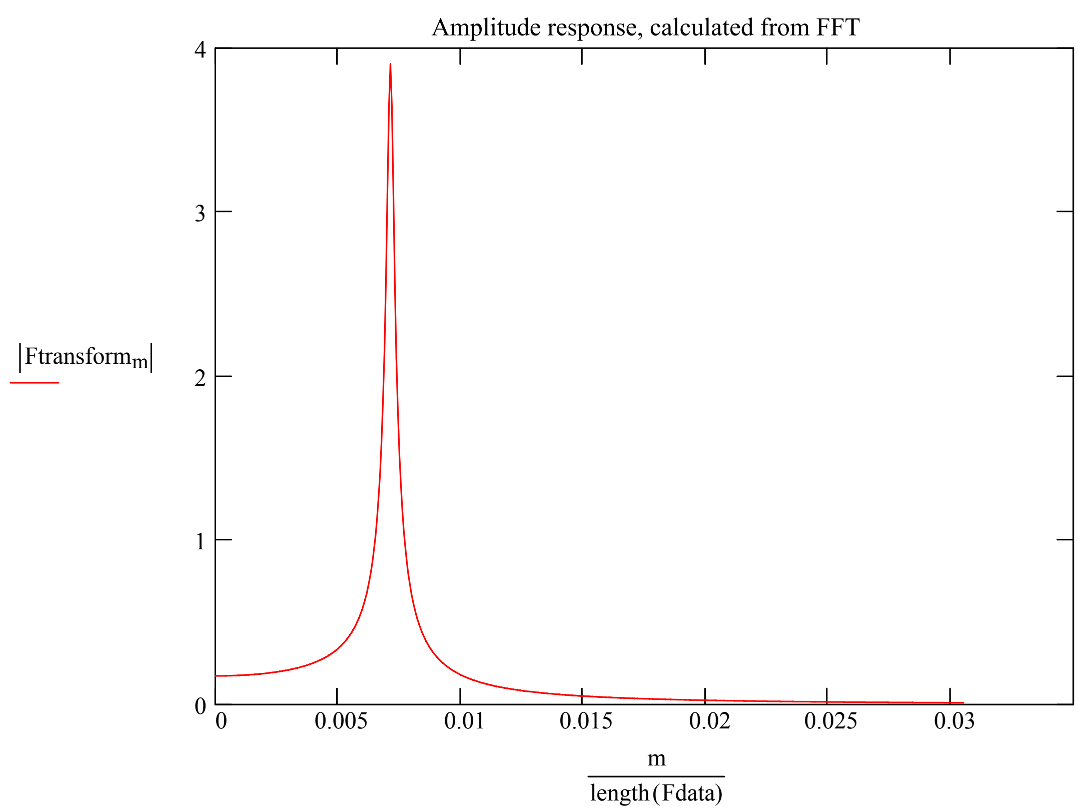
The plots below are EXACTLY THE SAME as the conventional plots shown in every freshman physics textbook showing the amplitude response and phase as a function of driving frequency. However, by measuring the impulse response and using linear response theory, we have obtained the information with a single measurement of the response function, completely avoiding the need to painstakingly measure the steady-state response at several hundred different frequencies. Wow!

$\mathrm{Ftransform} := \mathrm{fft}(\mathrm{Fdata})$

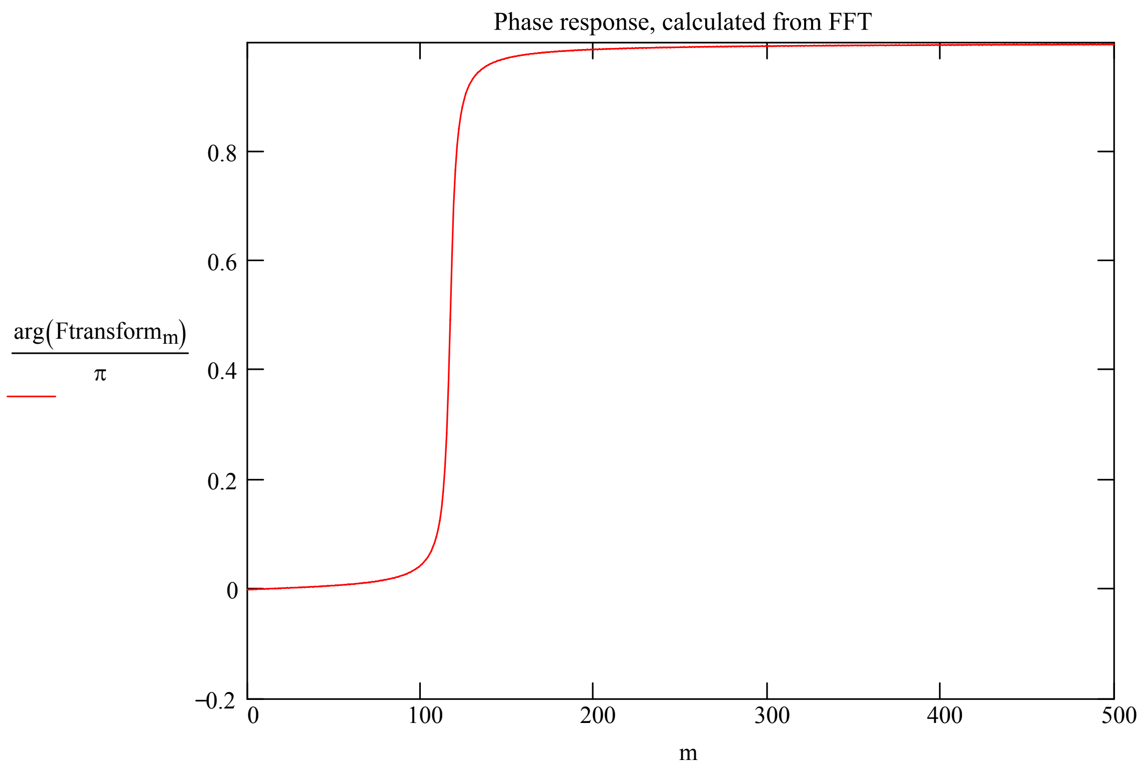
$\mathrm{NFFT} := \mathrm{last}(\mathrm{Ftransform})$

$n := 0 .. \mathrm{NFFT}$ Define index variable for frequency-domain data

$m := 0 .. 500$ Define a more limited frequency region for plotting



The phase of the Fourier transform evolves by π as the frequency sweeps through a resonance. Phase data calculated from experimental measurements may be much more complex, since nonlinearities and other imperfections can introduce additional resonances, each with its own phase excursions, as well as noise. In particular, for real experimental data the phase becomes ill-defined at high frequencies, where the signals are small and the noise is not.



Perhaps the most impressive thing about this approach is that given the frequency-domain transfer function, we can calculate the response to ANY time-dependent driving force. In time domain, we would take the convolution integral of the impulse response with the driving function. In frequency domain, we just multiply tl transfer function by the (complex) Fourier transform of the driving function. An inverse FFT then returns us tc the time domain, where we can plot and examine the results.

In this example, determine the response to two very different driving functions. The first is a sine-wave "burst starting at t=500 and continuing to t=5000. The second is a short rectangular pulse. Because this is essentia an impulse, the calculated response should be very similar to the original impulse response entered at the beginning of this spreadsheet.

The results are plotted on the following page.

$Drive_k := 0$

$start := 500 .. 5000$

$Drive_{start} := \sin\left(2 \cdot \pi \cdot \frac{start}{140}\right)$

$ImpulseDrive_k := 0$

$impulse\_on := 500 .. 510$

$ImpulseDrive_{impulse\_on} := 10$

$FDrive := fft(Drive)$

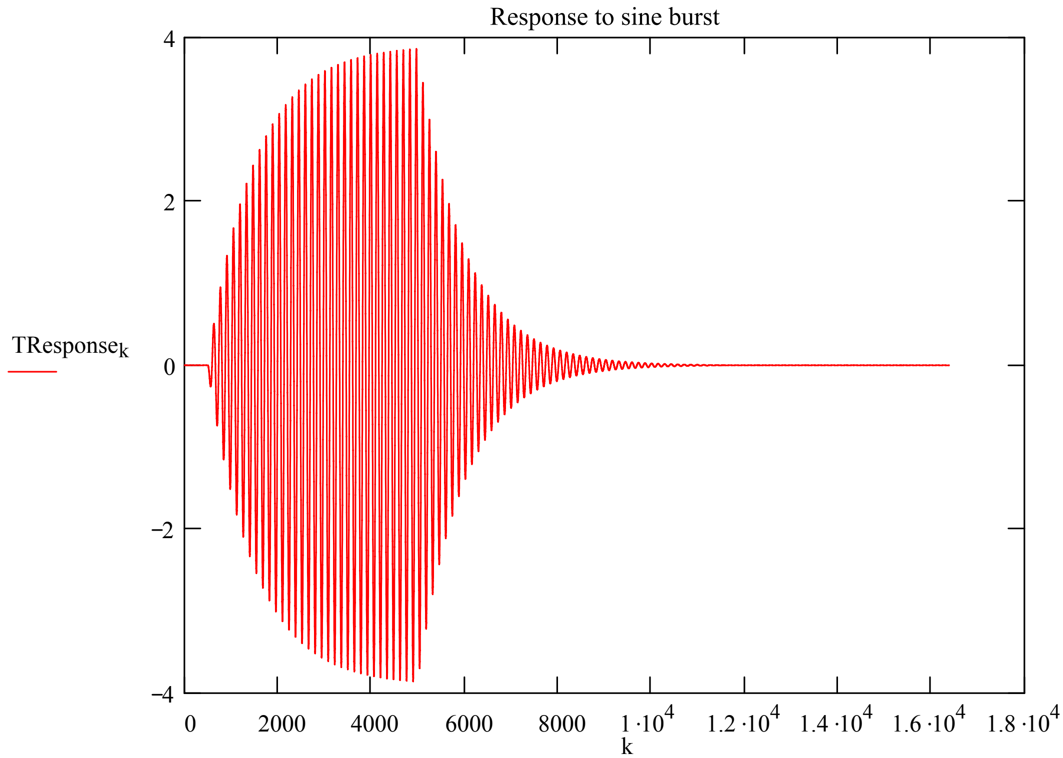
$Response_n := Ftransform_n \cdot FDrive_n$

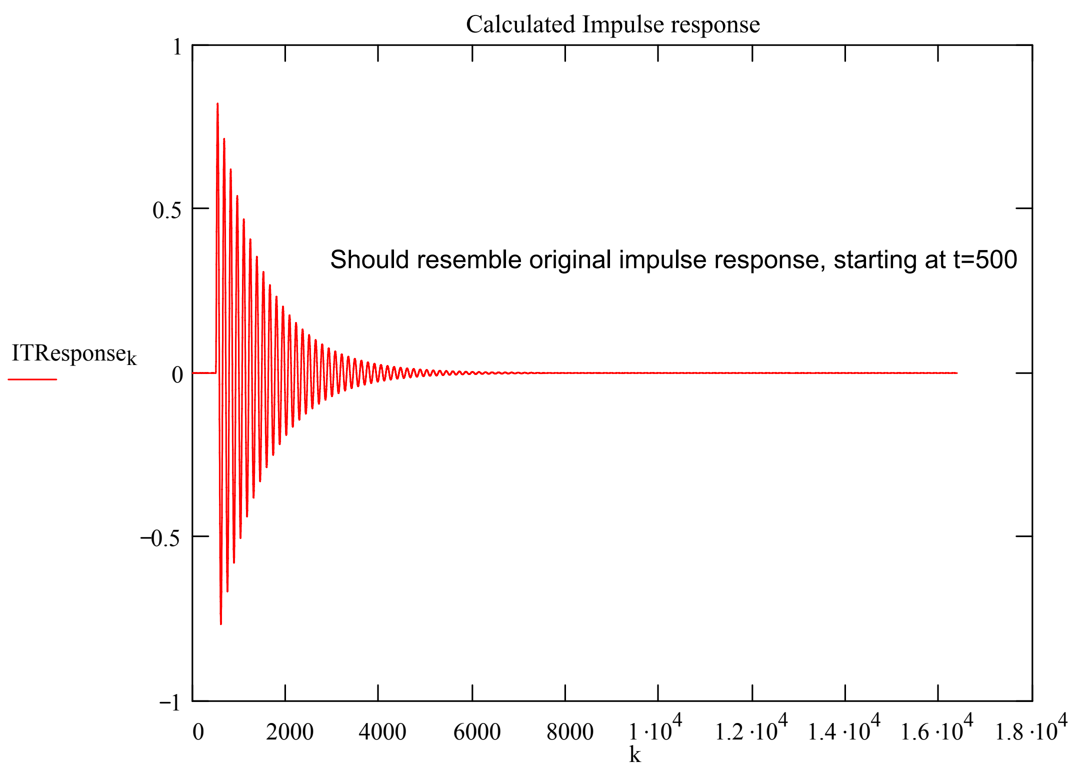
$TResponse := ifft(Response)$

$FIDrive := fft(ImpulseDrive)$

$IResponse_n := Ftransform_n \cdot FIDrive_n$

$ITResponse := ifft(IResponse)$


Response to sine burst
TResponse_k
4
2
0
−2
−4
0
2000
4000
6000
8000
1 ·10^4
1.2 ·10^4
1.4 ·10^4
1.6 ·10^4
1.8 ·10^4
k


Calculated Impulse response
Should resemble original impulse response, starting at t=500
ITResponse_k
1
0.5
0
−0.5
−1
0
2000
4000
6000
8000
1 ·10^4
1.2 ·10^4
1.4 ·10^4
1.6 ·10^4
1.8 ·10^4
k