# Envelope Calculation from the Hilbert Transform

**Article** · March 2006



**1 author:**

Timothy J. Ulrich
Los Alamos National Laboratory

# Envelope calculation from the Hilbert transform

TJ Ulrich

March 17, 2006

Given a modulated waveform $g(t)$,

$$g(t) = \sin(\omega t)\sin(\Omega t) \tag{1}$$

where $\omega > \Omega$, the envelope can be constructed from the absolute value of the analytic signal $\aleph(g(t))$. This analytic signal is composed of the original waveform $g(t)$ and its Hilbert transform $\tilde{g}(t)$ in the following manner:

$$\aleph(g(t)) = g(t) + i\tilde{g}(t). \tag{2}$$

1. Construct the Hilbert transform[1] $\tilde{g}(t)$ from the $g(t)$, given:

$$\tilde{g}(t) = -\int_0^{\infty}[a(f)\sin(ft) - b(f)\cos(ft)]df, \tag{3}$$

$$a(f) = \frac{1}{\pi}\int_{-\infty}^{\infty} g(t)\cos(ft)dt, \tag{4}$$

and

$$b(f) = \frac{1}{\pi}\int_{-\infty}^{\infty} g(t)\sin(ft)dt. \tag{5}$$

Start with constructing $a(f)$:

$$\begin{aligned} a(f) &= \frac{1}{\pi}\int_{-\infty}^{\infty}\sin(\omega t)\sin(\Omega t)\cos(ft)dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}[\cos{(\omega-\Omega)t} - \cos{(\omega+\Omega)t}]\cos(ft)dt \\ &= \frac{1}{2\pi}[\delta(f-\omega+\Omega) - \delta(f-\omega-\Omega)]. \end{aligned}$$

*The end result for $\tilde{g}(t)$ is off by a factor of $-\pi$ ... is the factor of $1/\pi$ correct here?*

Similarly,

$$b(f) = \frac{1}{\pi}\int_{-\infty}^{\infty}\sin(\omega t)\sin(\Omega t)\sin(ft)dt.$$

[1] Details about the analytic signal and Hilbert transform can be found in *Standard Mathematical Tables and Formulae*, CRC Press, ed. Daniel Zwillinger, 30th edition, pgs 547-550, (1996).

Using the trigonometric identities

$$\begin{aligned}
\sin(\alpha)\sin(\beta) &= \frac{1}{2}[\cos(\alpha-\beta)-\cos(\alpha+\beta)],\\
\cos(\alpha)\sin(\beta) &= \frac{1}{2}[\sin(\alpha-\beta)+\sin(\alpha+\beta)],
\end{aligned}$$

then

$$b(f) = 0.$$

Now substitute $a(f)$ and $b(f)$ into (3).

$$\begin{aligned}
\tilde{g}(t) &= -\frac{1}{2\pi}\int_0^\infty [\delta(f-\omega+\Omega)-\delta(f-\omega-\Omega)]\sin(ft)df\\
&= -\frac{1}{2\pi}[\sin{(\omega-\Omega)t}-\sin{(\omega+\Omega)t}]\\
&= \frac{1}{2\pi}[\sin{(\Omega-\omega)t}-\sin{(\Omega+\omega)t}]\\
&= \frac{1}{\pi}\sin(\Omega t)\cos(\omega t)
\end{aligned}$$

*As noted above, the correct form of the hilbert transform for the given $g(t)$ should be*

$$\tilde{g}(t) = -\sin(\Omega t)\cos(\omega t). \tag{6}$$

*For the remainder of this derivation the form of $\tilde{g}(t)$ will be taken from (6).*

2. Now that $\tilde{g}(t)$ has been constructed, the analytic signal can be written explicitly as

$$\aleph = \sin(\omega t)\sin(\Omega t) - i\sin(\Omega t)\cos(\omega t). \tag{7}$$

3. To obtain the envelope of the original signal $g(t)$ it is necessary to take the absolute value of the analytic signal $\aleph$.

$$\begin{aligned}
|\aleph| &= \sqrt{\aleph\aleph^*}\\
&= [\sin^2(\omega t)\sin^2(\Omega t)+\sin^2(\Omega t)\cos^2(\omega t)]^{1/2}\\
&= |\sin(\Omega t)|[\sin^2(\omega t)+\cos^2(\omega t)]^{1/2}\\
&= |\sin(\Omega t)|. \qquad (8)
\end{aligned}$$

While this has now been shown analytically (*all except that factor of* $-\pi$), the result can now be used quite easily in MATLAB from the syntax:

```
>> t = (1:131072)*(1e-4);
>> gt = sin(t).*sin(10*t);
>> envelope = abs(hilbert(gt));
>> figure; plot(t,gt,'b-',t,imag(hilbert(gt)),'k--',t,envelope,'r:')
```

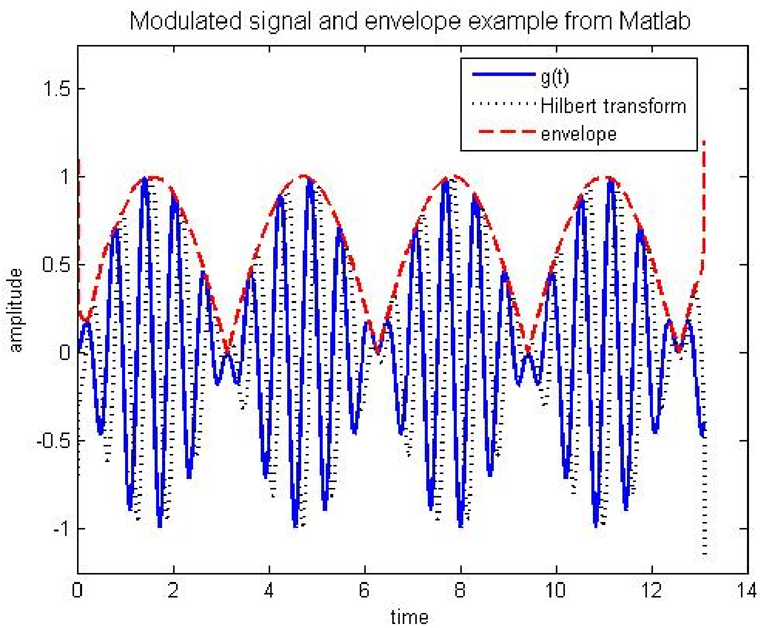


Figure 1: Results of the MATLAB example of calculating the envelope using the Hilbert transform. Note the end effects of the numerical computation.

Note, that in MATLAB, the command `hilbert(gt)` produces the analytic signal $\aleph$, NOT simply the Hilbert transform. To extract the Hilbert transform from the analytic signal it is necessary to use the MATLAB command `imag(hilbert(gt))` and extract only the imaginary part of the analytic signal. This is clear from (2). Figure 1 shows the MATLAB generated plot from the example above.

The above MATLAB example will only work properly if the user is using a version of MATLAB that contains the signal processing toolbox. A MATLAB function called `envelope` has been written that will produce both the envelope of a user defined waveform $g(t)$ and the Hilbert transform $\tilde{g}(t)$. This function will calculate the desired quantities (i.e., envelope and $\tilde{g}(t)$) without the need to use the signal processing toolbox of MATLAB. The envelope function (`envelope.m`) follows.

```
envelope.m


function [env, ht] = envelope(wf,varargin)

% Author: TJ Ulrich 5/23/05, updated 3/20/06 with code from RAG, see **
%         below.
% Purpose: create the envelope function of a given waveform, also provide
%          the hilbert transform of the original waveform
```

```
% Parameters: Input:  wf = waveform to be enveloped (preferably has a
%                          length that is a power of 2)
%                     sp = flag to indicate which method to use, i.e. is
%                          the signal processing toolbox present.
%                             0 = no (DO NOT use the signal proc. toolbox)
%                             1 = yes (use Sig. Proc. toolbox)
%             Output: env = envelop to be returned
%                     ht = Hilbert transform of wf
%
% ** the code from Robert Guyer was used/altered to make this function not
% dependent upon the availability of the signal processing toolbox.  The
% functionality using the sig. proc. toolbox is still available using a 1
% as the second input parameter.  a 0 or no second parameter will neither
% require nor use the signal processing toolbox.
%
% Syntax: env = envelope(wf) will produce the envelope of the waveform wf
%               without using the signal processing toolbox.
%         [env, HT] = envelope(wf) same as above but will also output the
%               hilbert transform in the variable HT.
%         env = hilbert(wf,1) same as env = hilbert(wf) but WILL use the
%               signal processing toolbox (i.e., the hilbert() function)
%         [env, HT] = envelope(wf,1) same as above but will also output the
%               hilbert transform in the variable HT.

% check for aditional arguments, i.e. whether or not to use signal
% processing toolbox functions
if nargin > 1
    sp = varargin{1};
else
    sp = 0;     % default value
end
% check for valid values of sp
switch sp
    case 0  % do nothing (valid value)
    case 1  % do nothing (valid value)
    otherwise
        sp = 0; % change to default valid value
end

% ----- create the envelope and Hilbert transform -----
if sp   % use the signal processing toolbox
    env = abs(hilbert(wf));
    ht = imag(hilbert(wf));
else
    %%%%%%
    Nfft=length(wf);
```

```
    sz = size(wf);           % get size of wf ... i.e. row or column vector
    hlfsz1 = ceil(sz/2);     % get half the size (use ceil() for odd # of points)
    hlfsz2 = floor(sz/2);    % get half the size (use floor() for odd # of points)
    hlfsz1(hlfsz1<1) = 1;    % fix fractional dimensions
    hlfsz2(hlfsz2<1) = 1;    % fix fractional dimensions
    %%%%%% FFT
    FFTf=fft(wf,Nfft);
    %%%%%% Hilbert transform
    nplus=(1:ceil(Nfft/2));
    nminus=(1+ceil(Nfft/2):Nfft);
    Hfactor=zeros(sz);
    Hfactor(nplus)=ones(hlfsz1);
    Hfactor(nminus)=-ones(hlfsz2);
    %%%%%%
    Hf=Hfactor.*FFTf;
    %%%%%% in time domain
    f1=ifft(FFTf,Nfft);
    h1=ifft(Hf,Nfft);
    ht = imag(h1);
    %%%%%% calculate envelope
    e1=f1+h1;
    env=sqrt(e1.*conj(e1));
end
```