Group Delay

CHRISTOPHER J. STRUCK

Group Delay is defined as the negative derivative (or slope) of the phase response vs. frequency. It is a measure of the relative delay at different frequencies from the input to the output in a system. Group Delay, however, generally represents the relative delay vs. frequency, with any processing, propagation, or other “all-pass” delay (sometimes referred to as “excess phase”) removed from the calculation -- i.e., the frequency dependent delay(s) corresponding to magnitude and phase variations in the frequency response. If this “all-pass” delay is not removed from the calculation, the Group Delay will indicate the propagation delay through the system plus the frequency response dependent phase derivative. Alternatively, the magnitude of the impulse response also reveals the delay in a system directly in the time domain. In a bandpass, or band-limited system, the peak in the impulse response magnitude generally represents the average group delay in the pass-band.

MatLab can be used for manipulating complex frequency domain data. A practical example would be creating a filter and then using this to filter a user-defined (e.g. measured) input response or spectrum. If the data are complex in the frequency domain, e.g., magnitude and phase versus frequency, the real and imaginary parts (also vs. frequency in Hertz) can also easily be calculated. This is similar to a polar to rectangular coordinate conversion. It is often also interesting, useful, and/or necessary to calculate the Group Delay from this complex vector.

Phase is somewhat difficult to handle because of the uncertainty due to the "wrapping" nature of the phase (modulo 360°). Before trying to find the delay (both the phase delay and the group delay) it is important that the phase is "unwrapped". "Unwrapping" the phase is generally enough to find the group delay, but even that may not always be enough to find a phase delay which makes sense.

Phase unwrapping can be done as follows:

```
While
 (Phase(n) - Phase(n-1))< -180)
     Do
       Phase(n) = Phase(n) + 360;
While
 (Phase(n) - Phase(n-1)) > 180)
     Do
       Phase(n)= Phase(n) - 360;                     [1]
```

where `Phase(n)` is the phase in degrees at the nth point in the data set and n >1.

This works for most cases, but in the case of a zero on the frequency axis, the phase jumps 180° and it is not trivial to decide in which direction. Notice that this never changes the first point. If the data set extends to a rather high frequency (and does not start down at DC), this last high frequency point might already be "wrapped" itself. For the Group Delay, however, it does not matter if there is an unknown offset, because the slope (or derivative) will be the same.

If the phase has to be unwrapped, there is also a problem if not all of the values down to DC are known. Then it is often necessary to try with a number of offsets (multiples of 360°) and then just see which one makes sense. That may not be convenient in many cases, so this is where some "measurement application experience" is required. For physical systems, such as loudspeakers, the phase must asymptote to either 0° or 180° at DC.

As previously mentioned, Group Delay is defined as the negative derivative of the phase

$$\tau_{Group\ Delay} = -\frac{d\varphi}{d\omega} = -\frac{d\theta}{df \cdot 360°} \qquad [2]$$

where ϕ and θ are the phase in radians and degrees, respectively and ω and f are the angular frequency [in radians/s] and the frequency [in Hz], respectively.

This Group Delay derivative can be calculated as a simple difference. In order to make this approximation "symmetric", the value is compared with both the next frequency upwards and downwards:

```
GroupDelay(n) = -1*[((Phase(n)-Phase(n-1))/(f(n)-f(n-1))
      +((Phase(n+1)-Phase(n))/(f(n+1)-f(n))]/(2*360)
```

[3]

where `Phase` is the phase in degrees and `f` is the frequency [in Hz].

In most cases this expression is almost the same as the simpler:

```
GroupDelay(n)=-1*[((ph(n+1)-ph(n-1))/(f(n+1)-f(n-1))]/(360)
```

[4]

This is true if

```
     (f(n)-f(n-1)) = (f(n+1)-f(n))
```

[5]

i.e. for a linear frequency format.

For a logarithmic frequency format (e.g., ISO R40), it is nearly the same, but it is more accurate to use the general expression in Eq. 3.

There are two exceptions: the lowest and highest frequency points. Those are handled just accepting the "asymmetry" at the ends:

```
GroupDelay(1)= -1*[((Phase(2)-Phase(1))/(f(2)-f(1)) ]/(360)
```

[6]

and

```
GroupDelay(N)=-1*[((Phase(N)-Phase(N-1))/(f(N)-f(N-1))]/(360)
```

[7]

Phase Delay is defined as:

$$\tau_{\textit{Phase Delay}} = -\frac{\varphi}{\omega} = -\frac{\theta}{f \cdot 360^{\circ}} \qquad [8]$$

This seems simpler because there is no derivative, but again it is necessary that the phase be unwrapped for this to make sense, i.e. the Phase Delay is "uncertain" modulo 1/f, something which depends on the frequency, while the phase is just modulo 360 degrees.

The Phase Delay is useful in some cases (e.g. when analyzing directional microphones), but it needs care when interpreting the results. The Group Delay also requires care, but in general it describes the delay of energy transport through the system at that particular frequency.

The following examples show what happens for a pure delay (Fig. 1), a pure delay with a 180° phase shift (inversion) (Fig. 2), a pure delay and a 1st order Low Pass system (Fig. 3), and finally a pure delay and a 1st order High Pass system (Fig. 4). The graphs also show the problems with unwrapping.

The examples also reveal another issue: Undersampling of the frequency response. If the delay in the system is so large that the phase shift is more than 180° between adjacent points, it will not be possible to unwrap easily. This happens above 5 kHz in the examples. Generally, for such non-minimum phase systems, it is suggested to first remove (or compensate) the constant all-pass delay in the system. The classic example of this is the simulated free field measurement of a loudspeaker using TDS where the propagation delay to the microphone (ca. 2.92ms / meter) must be accounted for. Therefore, most measurement systems have a “Delay” parameter. The “true” Group Delay can then be calculated from the unwrapped “residual” phase (Lipshitz & Vanderkooy).

Appendix

MATLAB function for calculating Group Delay

```
function t = GroupDelay(f,phase)
% GroupDelay Calculates the Group Delay from frequency data (in Hz)
%     and phase data (in degrees).
%        Christopher J. Struck - 3 August 2000

elements = size(phase);
k = elements(2);

% Unwrap phase data
for n = 2:k
   while phase(n) - phase(n - 1) <= -180
      phase(n) = phase(n) + 360;
   end
   while phase(n) - phase(n - 1) >= 180
      phase(n)= phase(n) - 360;
   end
end

% Calculate Group Delay
for n = 2:k-1
   t(n) = (-1/720) * (((phase(n) - phase(n - 1)) / (f(n) - f(n - 1)))...
      + ((phase(n + 1) - phase(n)) / (f(n + 1) - f(n))));
end
t(1) = (-1/360) * (((phase(2) - phase(1))/(f(2) - f(1))));
t(k) = (-1/360) * (((phase(k) - phase(k - 1))/(f(k) - f(k - 1))));
```

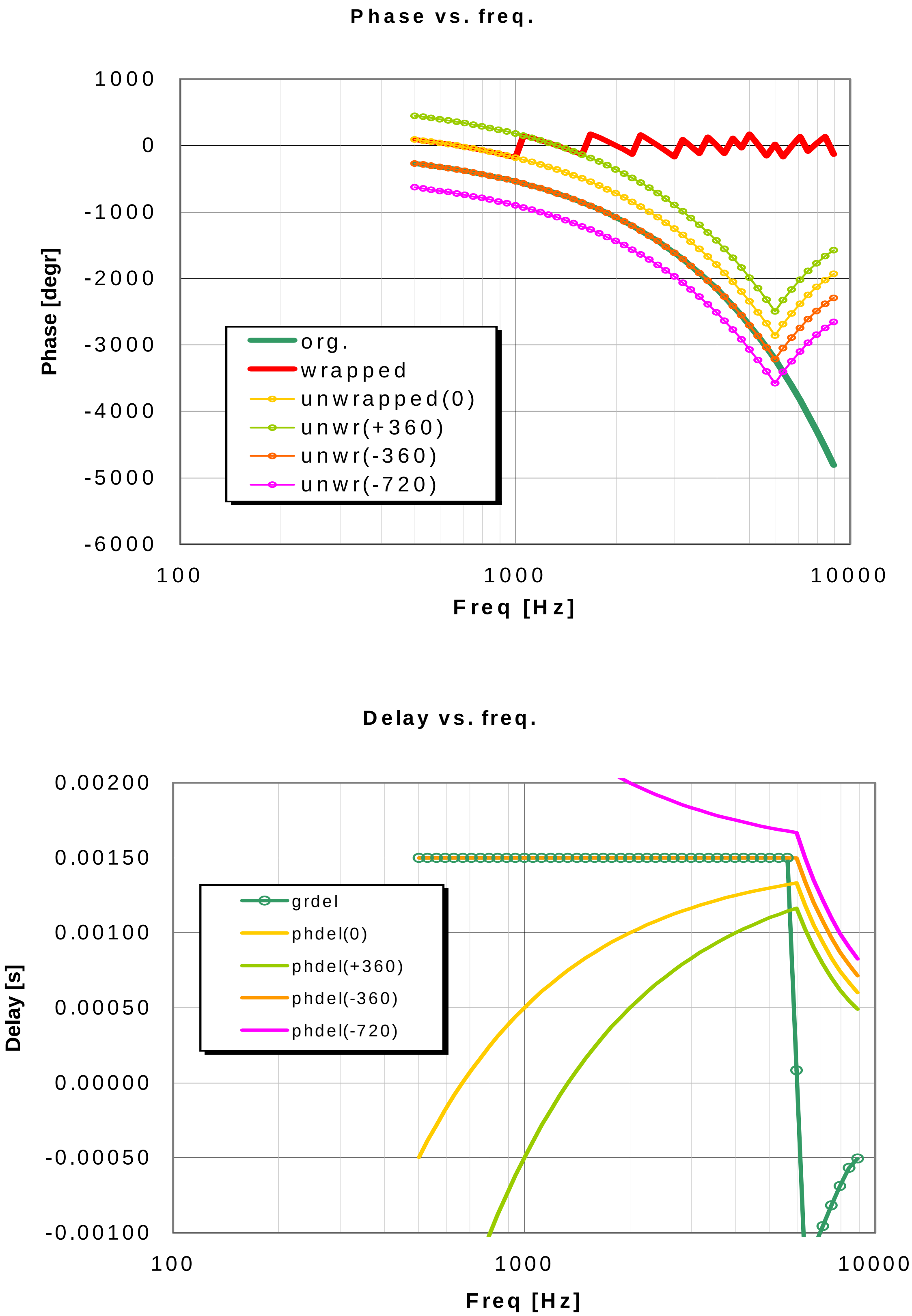


Figure 1 Constant Delay. a) Phase. b) Group Delay

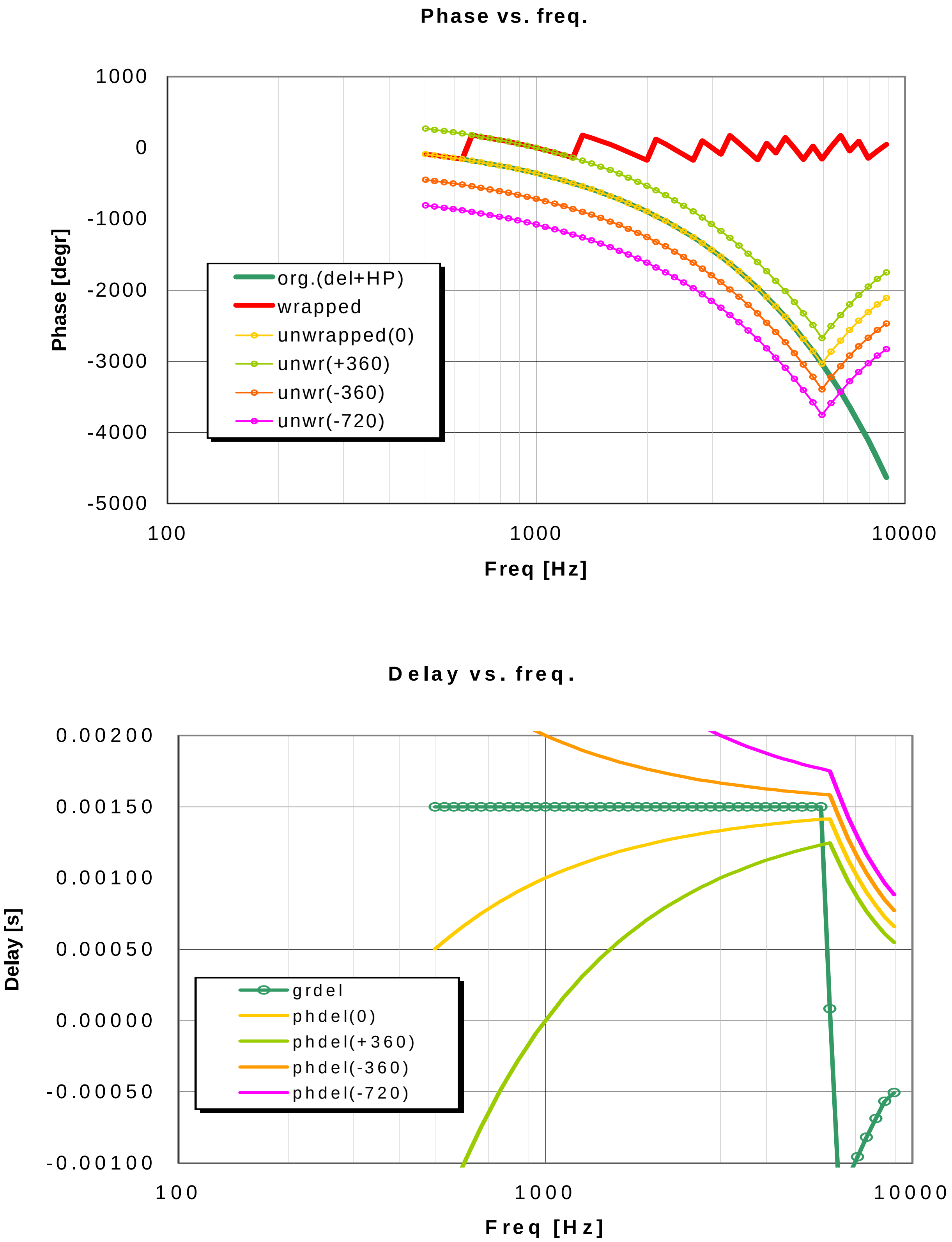
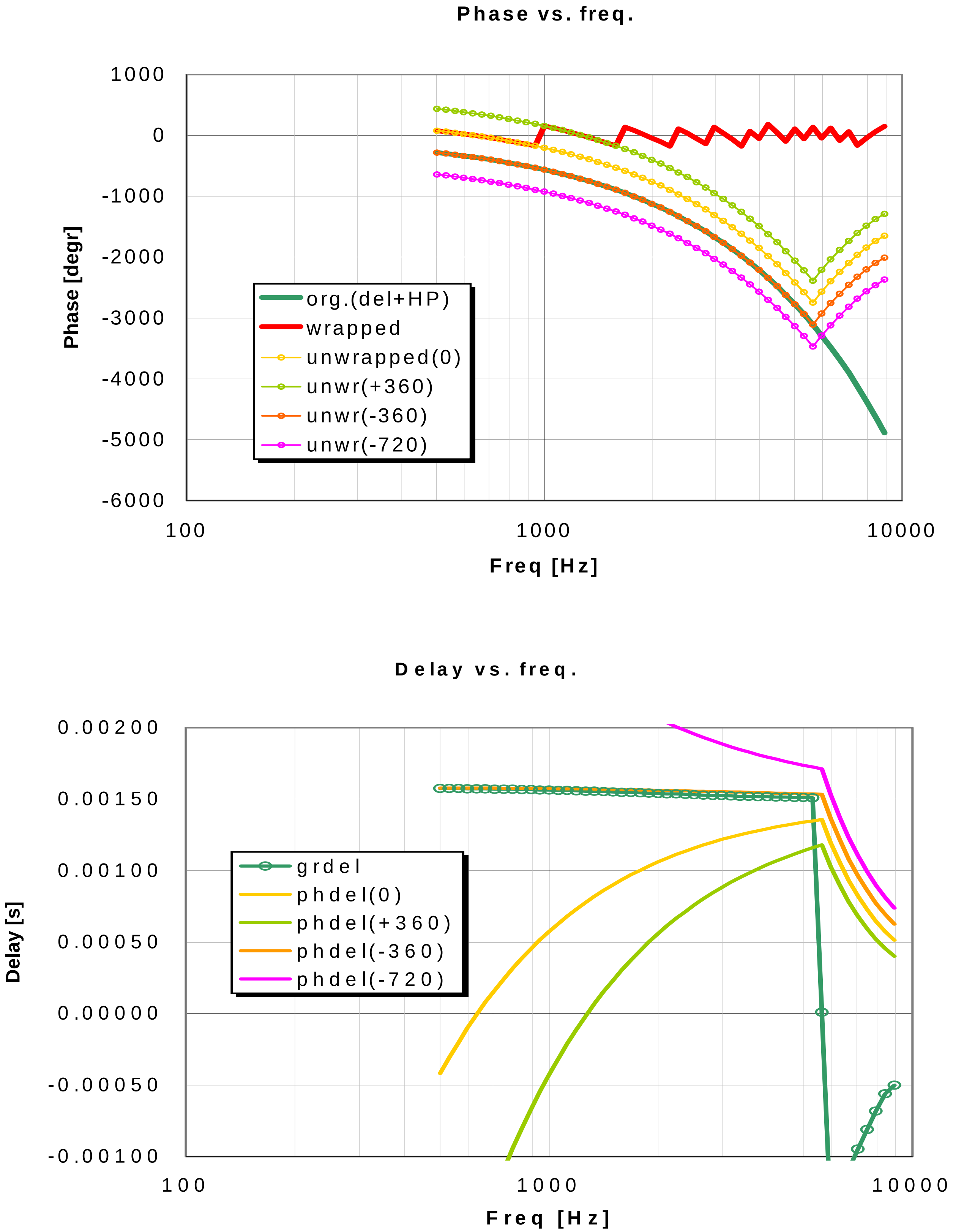


Figure 2. Constant Delay with inversion. a) Phase. b) Group Delay.



Phase vs. freq.

Delay vs. freq.

Figure 3. Constant delay with low pass filtering. a) Phase. b) Group Delay.

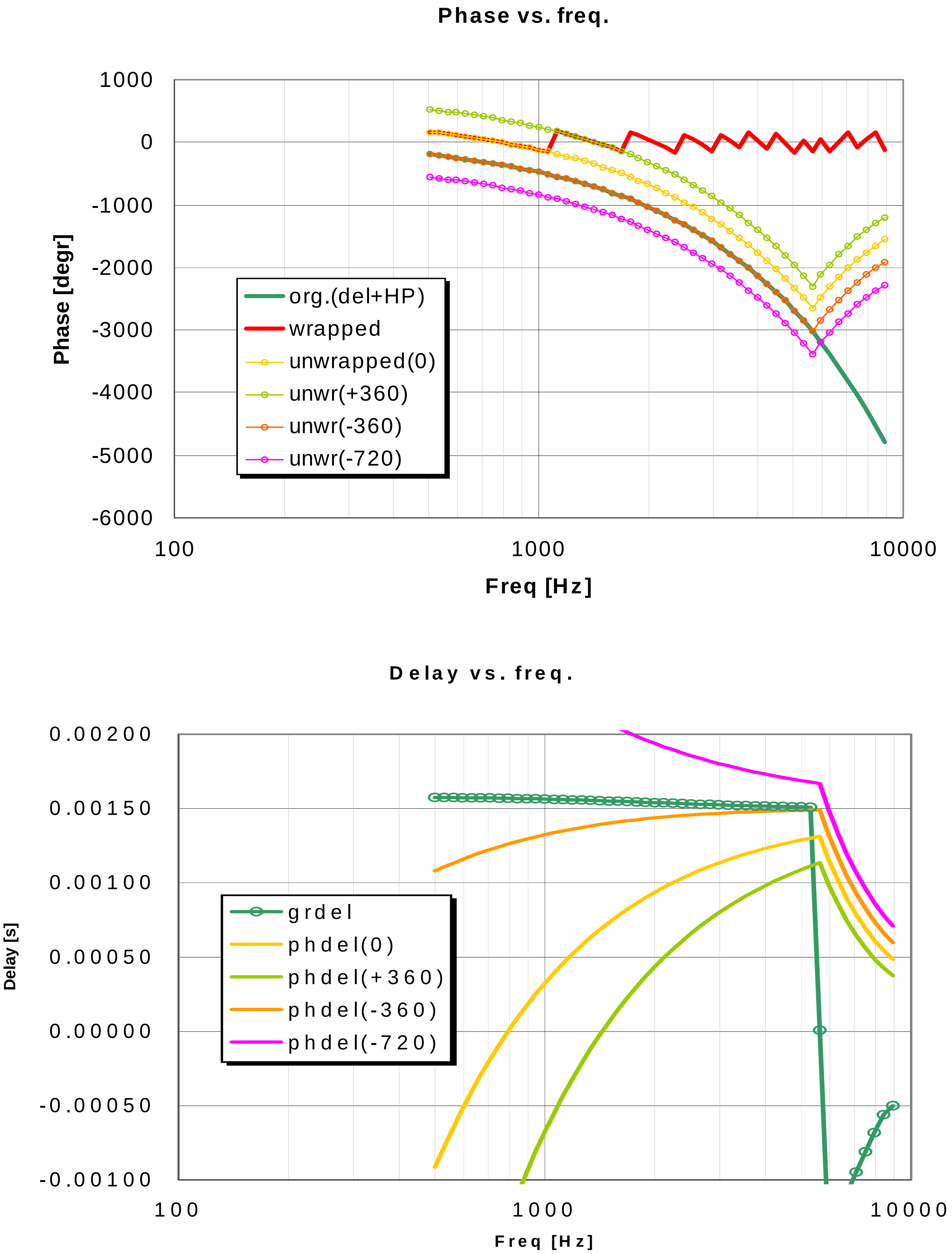


Figure 4. Constant delay with high pass filtering. a) Phase. b) Group Delay.