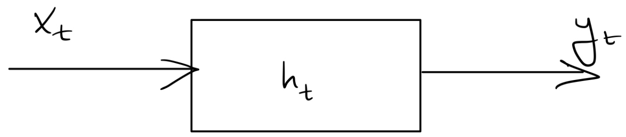
LUND UNIVERSITY

Group and phase delays

Andreas Jakobsson

Group delay and phase delay

When filtering a signal through a linear time-invariant (LTI) system, the resulting signal experience a frequency dependent delay, the so-called *group delay*, that cause the signal to be delayed in time as compared to the input signal.



Consider a signal, x_t, filtered through the LTI filter h_t, such that

$$y_t = \int_{-\infty}^{\infty} h_\tau x_{t-\tau} d\tau$$

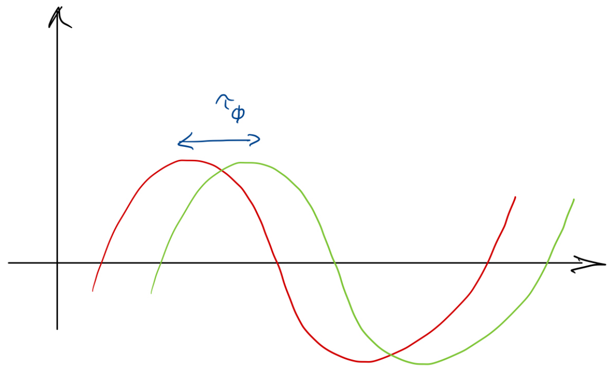
Then, if the input signal is a sinusoidal, i.e.,

$$x_t = a_t \cos(\omega t + \theta)$$

with some slowly time-varying envelope, a_t, the output signal may be well approximated as

$$y_t \approx |H(\omega)| a_{t-\tau_g} \cos(\omega(t - \tau_\phi) + \theta)$$

where $H(\omega)$ is the Fourier-transform of the filter, and τ_g and τ_ϕ are the group delay and the phase delay, respectively.

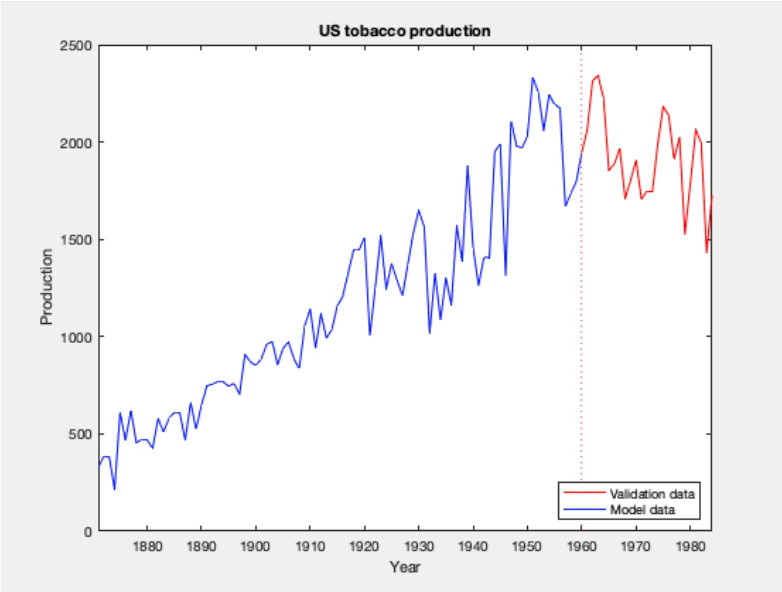


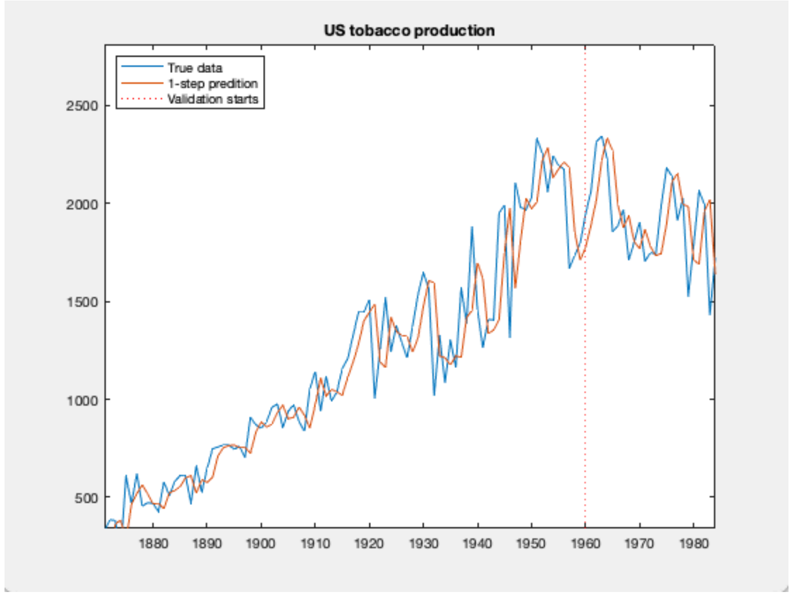
This means that the phase (of the sinusoid) is delayed by τ_ϕ, whereas the envelope of the siganl is delayed by τ_g. In general, these delays will be *frequency dependent*, meaning that different frequencies will experience different delays.

Group delay and phase delay

When predicting a signal, one typically implements the prediction using a linear filter, initiating the filter from the start of the modeling data, and then compare the result to the validation data, in order to form a k-step prediction for the entire validation data.

Asa result, the group and phase delays have the effect that the predicted signal (may) appear to be ”shifted” as compared to your validation data.





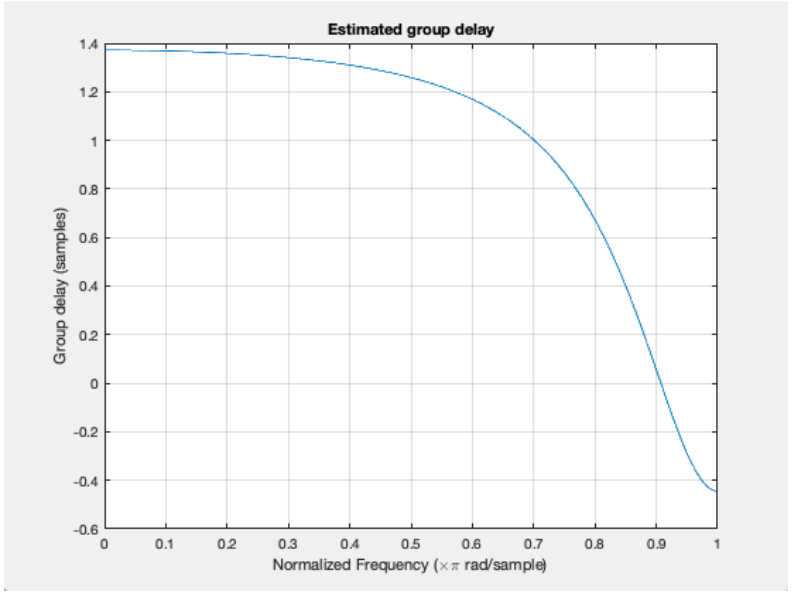
(see the code example *”Predicting the tobacco production”*)

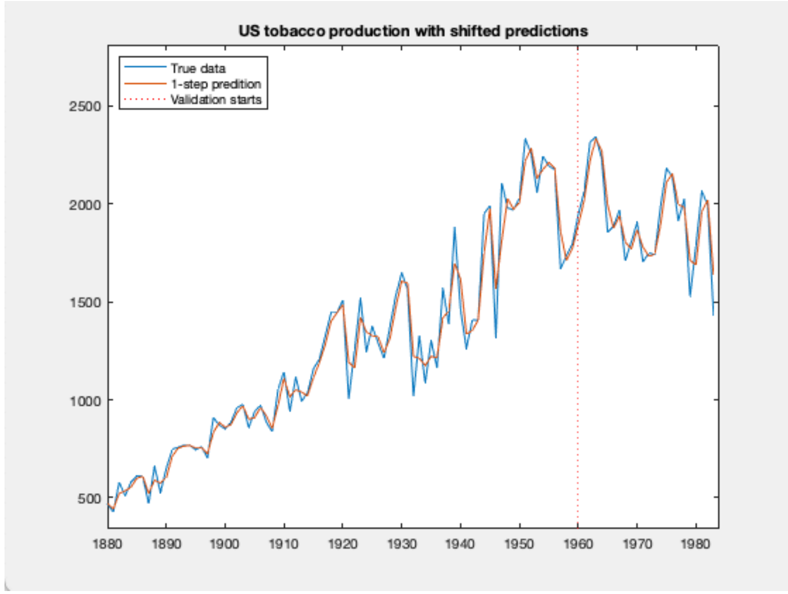
Group delay and phase delay

As the delays will be frequency dependent, these will generally vary over the signal. In Matlab, one can compute the resulting delay using the command *grpdelay*, showing how the delay varies over frequency.

If the phase is well behaved (this is not always the case), the signal will appear shifted with approximately the average delay (in this example this is 1).

When forming a *linear* prediction, this is the best we can achieve, and the resulting prediction should not be shifted to compensate for this. However, it may be helpful *visually* to do so when comparing the prediction with the validation data.





(see the code example *"Predicting the tobacco production"*)