Filter Theory in Electronic Engineering

Marya Troutman

First Edition, 2012

ISBN 978-81-323-3455-2

Published by:
Research World
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter-1

Filter Design

Filter design is the process of designing a filter (in the sense in which the term is used in signal processing, statistics, and applied mathematics), often a linear shift-invariant filter, that satisfies a set of requirements, some of which are contradictory. The purpose is to find a realization of the filter that meets each of the requirements to a sufficient degree to make it useful.

The filter design process can be described as an optimization problem where each requirement contributes with a term to an error function which should be minimized. Certain parts of the design process can be automated, but normally an experienced electrical engineer is needed to get a good result.

Typical design requirements

Typical requirements which are considered in the design process are:

- The filter should have a specific frequency response
- The filter should have a specific impulse response
- The filter should be causal
- The filter should be stable
- The filter should be localized
- The computational complexity of the filter should be low
- The filter should be implemented in particular hardware or software

The frequency function

Typical examples of frequency function are:

- A low-pass filter is used to cut unwanted high-frequency signals.
- A high-pass filter passes high frequencies fairly well; it is helpful as a filter to cut any unwanted low frequency components.
- A band-pass filter passes a limited range of frequencies.
- A band-stop filter passes frequencies above and below a certain range. A very narrow band-stop filter is known as a notch filter.

- A low-shelf filter passes all frequencies, but increases or reduces frequencies below the shelf frequency by specified amount.
- A high-shelf filter passes all frequencies, but increases or reduces frequencies above the shelf frequency by specified amount.
- A peak EQ filter makes a peak or a dip in the frequency response, commonly used in parametric equalizers.
- An all-pass filter passes through all frequencies unchanged, but changes the phase of the signal. This is a filter commonly used in phaser effects.

An important parameter is the required frequency response. In particular, the steepness and complexity of the response curve is a deciding factor for the filter order and feasibility.

A first order recursive filter will only have a single frequency-dependent component. This means that the slope of the frequency response is limited to 6 dB per octave. For many purposes, this is not sufficient. To achieve steeper slopes, higher order filters are required.

In relation to the desired frequency function, there may also be an accompanying *weighting* function which describes, for each frequency, how important it is that the resulting frequency function approximates the desired one. The larger weight, the more important is a close approximation.

The impulse response

There is a direct correspondence between the filter's frequency function and its impulse response: the former is the Fourier transform of the latter. That means that any requirement on the frequency function is a requirement on the impulse response, and vice versa.

However, in certain applications it may be the filter's impulse response that is explicit and the design process then aims at producing as close an approximation as possible to the requested impulse response given all other requirements.

In some cases it may even be relevant to consider a frequency function and impulse response of the filter which are chosen independently from each other. For example, we may want both a specific frequency function of the filter *and* that the resulting filter have a small effective width in the signal domain as possible. The latter condition can be realized by considering a very narrow function as the wanted impulse response of the filter even though this function has no relation to the desired frequency function. The goal of the design process is then to realize a filter which tries to meet both these contradicting design goals as much as possible.

Causality

In order to be implementable, any time-dependent filter must be causal: the filter response only depends on the current and past inputs. A standard approach is to leave this requirement until the final step. If the resulting filter is not causal, it can be made causal by introducing an appropriate time-shift (or delay). If the filter is a part of a larger system (which it normally is) these types of delays have to be introduced with care since they affect the operation of the entire system.

Stability

A stable filter assures that every limited input signal produces a limited filter response. A filter which does not meet this requirement may in some situations prove useless or even harmful. Certain design approaches can guarantee stability, for example by using only feed-forward circuits such as an FIR filter. On the other hand, filter based on feedback circuits have other advantages and may therefore be preferred, even if this class of filters include unstable filters. In this case, the filters must be carefully designed in order to avoid instability.

Locality

In certain applications we have to deal with signals which contain components which can be described as local phenomena, for example pulses or steps, which have certain time duration. A consequence of applying a filter to a signal is, in intuitive terms, that the duration of the local phenomena is extended by the width of the filter. This implies that it is sometimes important to keep the width of the filter's impulse response function as short as possible.

According to the uncertainty relation of the Fourier transform, the product of the width of the filter's impulse response function and the width of its frequency function must exceed a certain constant. This means that any requirement on the filter's locality also implies a bound on its frequency function's width. Consequently, it may not be possible to simultaneously meet requirements on the locality of the filter's impulse response function as well as on its frequency function. This is a typical example of contradicting requirements.

Computational complexity

A general desire in any design is that the number of operations (additions and multiplications) needed to compute the filter response is as low as possible. In certain applications, this desire is a strict requirement, for example due to limited computational resources, limited power resources, or limited time. The last limitation is typical in real-time applications.

There are several ways in which a filter can have different computational complexity. For example, the order of a filter is more or less proportional to the number of operations. This means that by choosing a low order filter, the computation time can be reduced.

For discrete filters the computational complexity is more or less proportional to the number of filter coefficients. If the filter has many coefficients, for example in the case of multidimensional signals such as tomography data, it may be relevant to reduce the number of coefficients by removing those which are sufficiently close to zero.

Another issue related to computational complexity is separability, that is, if and how a filter can be written as a convolution of two or more simpler filters. In particular, this issue is of importance for multidimensional filters, e.g., 2D filter which are used in image processing. In this case, a significant reduction in computational complexity can be obtained if the filter can be separated as the convolution of one 1D filter in the horizontal direction and one 1D filter in the vertical direction. A result of the filter design process may, e.g., be to approximate some desired filter as a separable filter or as a sum of separable filters.

Other considerations

It must also be decided how the filter is going to be implemented:

- Analog filter
- Analog sampled filter
- Digital filter
- Mechanical filter

Analog filters

The design of linear analog filters is for the most part covered in the linear filter section.

Digital filters

Digital filters are classified into one of two basic forms, according to how they respond to an unit impulse:

- Finite impulse response, or **FIR**, filters express each output sample as a weighted sum of the last N inputs, where N is the order of the filter. Since they do not use feedback, they are inherently stable. If the coefficients are symmetrical (the usual case), then such a filter is linear phase, so it delays signals of all frequencies equally. This is important in many applications. It is also straightforward to avoid overflow in an FIR filter. The main disadvantage is that they may require significantly more processing and memory resources than cleverly designed IIR variants. FIR filters are generally easier to design than IIR filters - the Remez exchange algorithm is one suitable method for designing quite good filters semi-automatically.

- Infinite impulse response, or **IIR**, filters are the digital counterpart to analog filters. Such a filter contains internal state, and the output and the next internal state are determined by a linear combination of the previous inputs and outputs (in other words, they use feedback, which FIR filters normally do not). In theory, the impulse response of such a filter never dies out completely, hence the name IIR, though in practice, this is not true given the finite resolution of computer arithmetic. IIR filters normally require less computing resources than an FIR filter of similar performance. However, due to the feedback, high order IIR filters may have problems with instability, arithmetic overflow, and limit cycles, and require careful design to avoid such pitfalls. Additionally, since the phase shift is inherently a non-linear function of frequency, the time delay through such a filter is frequency-dependent, which can be a problem in many situations. 2nd order IIR filters are often called 'biquads' and a common implementation of higher order filters is to cascade biquads. A useful reference for computing biquad coefficients is the RBJ Audio EQ Cookbook.

Sample rate

Unless the sample rate is fixed by some outside constraint, selecting a suitable sample rate is an important design decision. A high rate will require more in terms of computational resources, but less in terms of anti-aliasing filters. Interference and beating with other signals in the system may also be an issue.

Anti-aliasing

For any digital filter design, it is crucial to analyze and avoid aliasing effects. Often, this is done by adding analog anti-aliasing filters at the input and output, thus avoiding any frequency component above the Nyquist frequency. The complexity (i.e., steepness) of such filters depends on the required signal to noise ratio and the ratio between the sampling rate and the highest frequency of the signal.

Theoretical basis

Parts of the design problem relate to the fact that certain requirements are described in the frequency domain while others are expressed in the signal domain and that these may contradict. For example, it is not possible to obtain a filter which has both an arbitrary impulse response and arbitrary frequency function. Other effects which refer to relations between the signal and frequency domain are

- The uncertainty principle between the signal and frequency domains
- The variance extension theorem
- The asymptotic behaviour of one domain versus discontinuities in the other

The uncertainty principle

As stated in the uncertainty principle, the product of the width of the frequency function and the width of the impulse response cannot be smaller than a specific constant. This implies that if a specific frequency function is requested, corresponding to a specific frequency width, the minimum width of the filter in the signal domain is set. Vice versa, if the maximum width of the response is given, this determines the smallest possible width in the frequency. This is a typical example of contradicting requirements where the filter design process may try to find a useful compromise.

The variance extension theorem

Let σ_s^2 be the variance of the input signal and let σ_f^2 be the variance of the filter. The variance of the filter response, σ_r^2, is then given by

$$\sigma_r^2 = \sigma_s^2 + \sigma_f^2$$

This means that $\sigma_r > \sigma_f$ and implies that the localization of various features such as pulses or steps in the filter response is limited by the filter width in the signal domain. If a precise localization is requested, we need a filter of small width in the signal domain and, via the uncertainty principle, its width in the frequency domain cannot be arbitrary small.

Discontinuities versus asymptotic behaviour

Let *f(t)* be a function and let $F(\omega)$ be its Fourier transform. There is a theorem which states that if the first derivative of F which is discontinuous has order $n \geq 0$, then f has an asymptotic decay like t^{-n-1}.

A consequence of this theorem is that the frequency function of a filter should be as smooth as possible to allow its impulse response to have a fast decay, and thereby a short width.

Methodology

One common method for designing FIR filters is the Remez exchange algorithm. Here the user specifies a desired frequency response, a weighting function for errors from this response, and a filter order *N*. The algorithm then finds the set of *N* coefficients that minimize the maximum deviation from the ideal. Intuitively, this finds the filter that is as close as you can get to the desired response given that you can use only *N* coefficients. This method is particularly easy in practice and at least one text includes a program that takes the desired filter and *N* and returns the optimum coefficients. One possible drawback to filters designed this way is that they contain many small ripples in the passband(s), since such a filter minimizes the peak error.

Another method to finding a discrete FIR filter is *filter optimization* described in Knutsson et al., which minimizes the integral of the square of the error, instead of its maximum value. In its basic form this approach requires that an ideal frequency function of the filter $F_I(\omega)$ is specified together with a frequency weighting function $W(\omega)$ and set of coordinates x_k in the signal domain where the filter coefficients are located.

An error function ε is defined as

$$\varepsilon = \|W \cdot (F_I - \mathcal{F}\{f\})\|^2$$

where $f(x)$ is the discrete filter and $\mathcal{F}$ is the discrete-time Fourier transform defined on the specified set of coordinates. The norm used here is, formally, the usual norm on L^2 spaces. This means that ε measures the deviation between the requested frequency function of the filter, F_I, and the actual frequency function of the realized filter, $\mathcal{F}\{f\}$. However, the deviation is also subject to the weighting function W before the error function is computed.

Once the error function is established, the optimal filter is given by the coefficients $f(x)$ which minimize ε. This can be done by solving the corresponding least squares problem. In practice, the L^2 norm has to be approximated by means of a suitable sum over discrete points in the frequency domain. In general, however, these points should be significantly more than the number of coefficients in the signal domain to obtain a useful approximation.

Simultaneous optimization in both domains

The previous method can be extended to include an additional error term related to a desired filter impulse response in the signal domain, with a corresponding weighting function. The ideal impulse response can be chosen independently of the ideal frequency function and is in practice used to limit the effective width and to remove ringing effects of the resulting filter in the signal domain. This is done by choosing a narrow ideal filter impulse response function, e.g., an impulse, and a weighting function which grows fast with the distance from the origin, e.g., the distance squared. The optimal filter can still be calculated by solving a simple least squares problem and the resulting filter is then a "compromise" which has a total optimal fit to the ideal functions in both domains. An important parameter is the relative strength of the two weighting functions which determines in which domain it is more important to have a good fit relative to the ideal function.

Chapter-2

Comb Filter

In signal processing, a **comb filter** adds a delayed version of a signal to itself, causing constructive and destructive interference. The frequency response of a comb filter consists of a series of regularly spaced spikes, giving the appearance of a comb.

Applications

Comb filters are used in a variety of signal processing applications. These include:

- Cascaded Integrator-Comb (CIC) filters, commonly used for anti-aliasing during interpolation and decimation operations that change the sample rate of a discrete-time system.
- 2D and 3D comb filters implemented in hardware (and occasionally software) for PAL and NTSC television decoders. The filters work to reduce artifacts such as dot crawl.
- Audio effects, including echo, flanging, and digital waveguide synthesis. For instance, if the delay is set to a few milliseconds, a comb filter can be used to model the effect of acoustic standing waves in a cylindrical cavity or in a vibrating string.

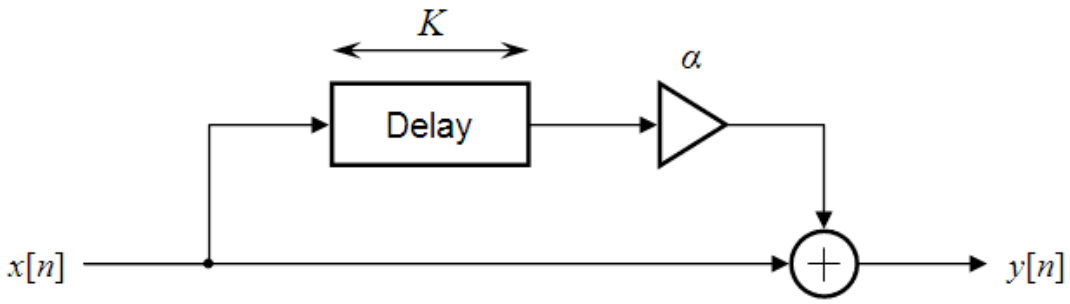
In acoustics, comb filtering can arise in some unwanted ways. For instance, when two loudspeakers are playing the same signal at different distances from the listener, there is a comb filtering effect on the signal. In any enclosed space, listeners hear a mixture of direct sound and reflected sound. Because the reflected sound takes a longer path, it constitutes a delayed version of the direct sound and a comb filter is created where the two combine at the listener.

Technical discussion

Comb filters exist in two different forms, *feed-forward* and *feedback*; the names refer to the direction in which signals are delayed before they are added to the input.

Comb filters may be implemented in discrete time or continuous time; here we will focus on discrete-time implementations; the properties of the continuous-time comb filter are very similar.

Feedforward form



Feedforward comb filter structure

The general structure of a feedforward comb filter is shown on the right. It may be described by the following difference equation:

$$y[n] = x[n] + \alpha x[n - K]$$

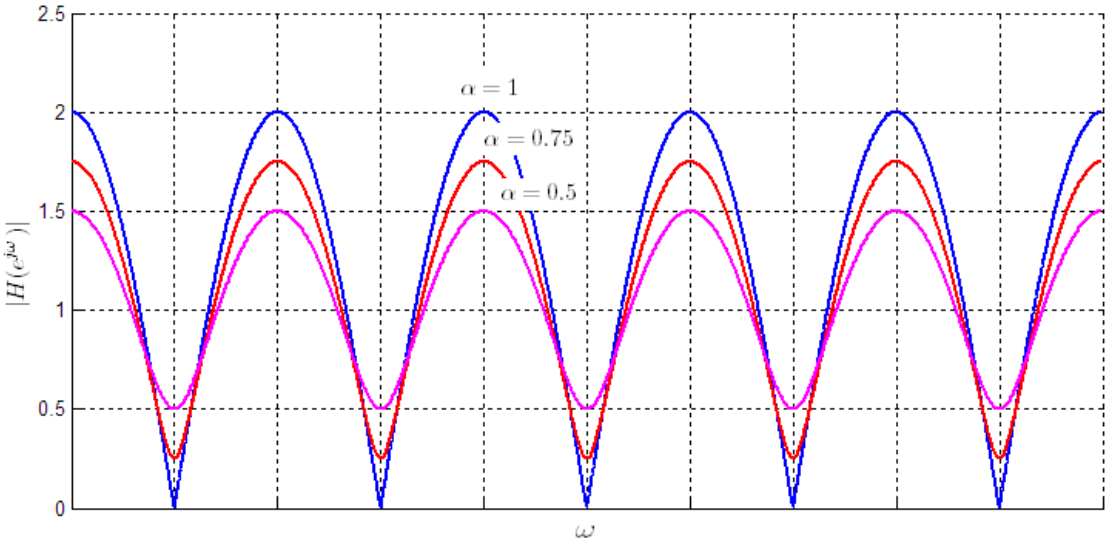
where K is the delay length (measured in samples), and α is a scaling factor applied to the delayed signal. If we take the Z transform of both sides of the equation, we obtain:

$$Y(z) = (1 + \alpha z^{-K})X(z)$$

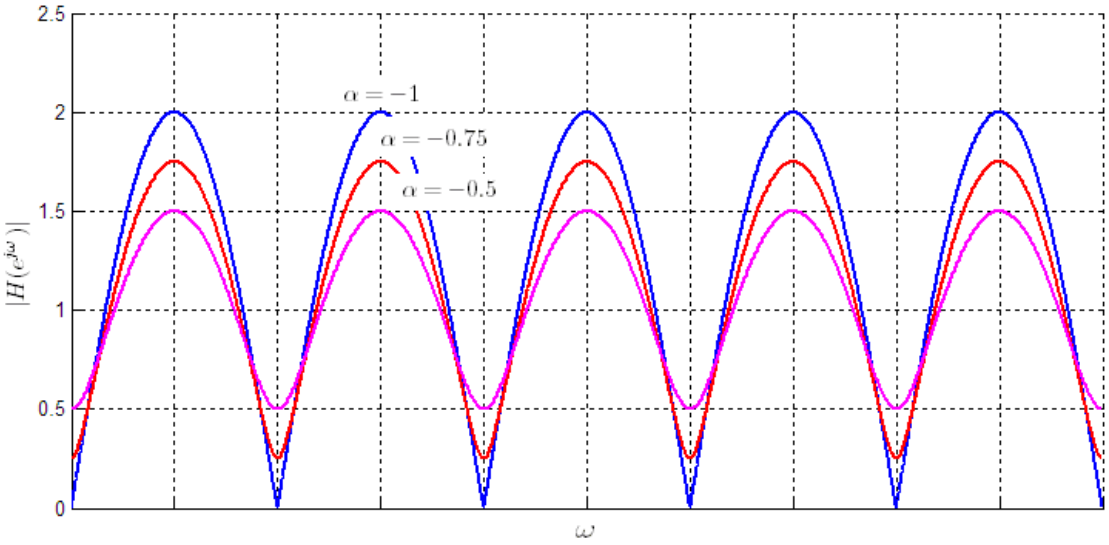
We define the transfer function as:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \alpha z^{-K} = \frac{z^K + \alpha}{z^K}$$

Frequency response



Feedforward magnitude response for various positive values of α



Feedforward magnitude response for various negative values of α

To obtain the frequency response of a discrete-time system expressed in the Z domain, we make the substitution $z = e^{j\omega}$. Therefore, for our feedforward comb filter, we get:

$$H(e^{j\omega}) = 1 + \alpha e^{-j\omega K}$$

Using Euler's formula, we find that the frequency response is also given by

$$H(e^{j\omega}) = [1 + \alpha \cos(\omega K)] - j\alpha \sin(\omega K)$$

Often of interest is the *magnitude* response, which ignores phase. This is defined as:

$$|H(e^{j\omega})| = \sqrt{\Re\{H(e^{j\omega})\}^2 + \Im\{H(e^{j\omega})\}^2}$$

In the case of the feedforward comb filter, this is:

$$|H(e^{j\omega})| = \sqrt{(1 + \alpha^2) + 2\alpha \cos(\omega K)}$$

Notice that the $(1 + \alpha^2)$ term is constant, whereas the $2\alpha\cos(\omega K)$ term varies periodically. Hence the magnitude response of the comb filter is periodic.

The graphs to the right show the magnitude response for various values of α, demonstrating this periodicity. Some important properties:

- The response periodically drops to a local minimum (sometimes known as a *notch*), and periodically rises to a local maximum (sometimes known as a *peak*).
- The levels of the maxima and minima are always equidistant from 1.
- When $\alpha = \pm 1$, the minima have zero amplitude. In this case, the minima are sometimes known as *nulls*.
- The maxima for positive values of α coincide with the minima for negative values of α, and vice versa.

Impulse response

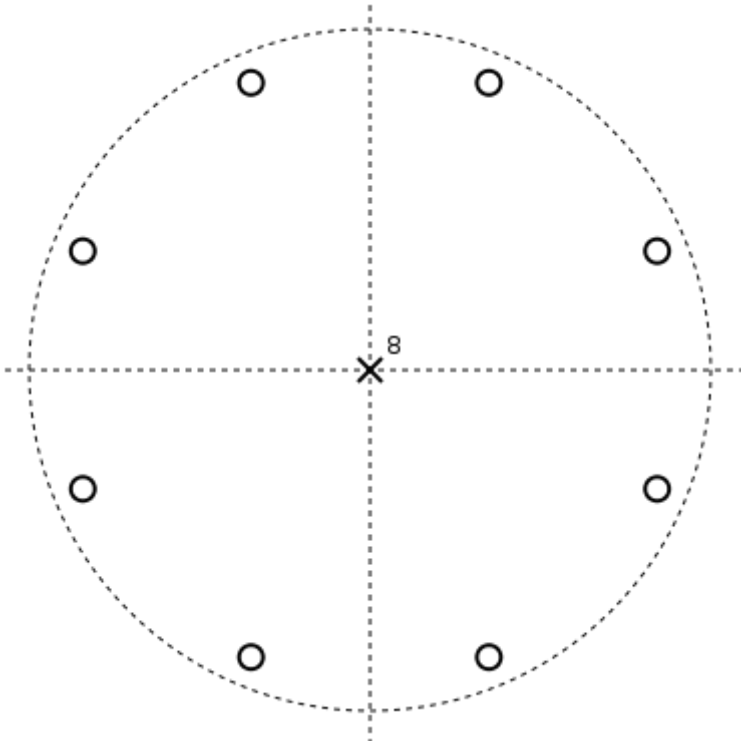
The feedforward comb filter is one of the simplest finite impulse response filters. Its response is simply the initial impulse with a second impulse after the delay.

Pole-zero interpretation

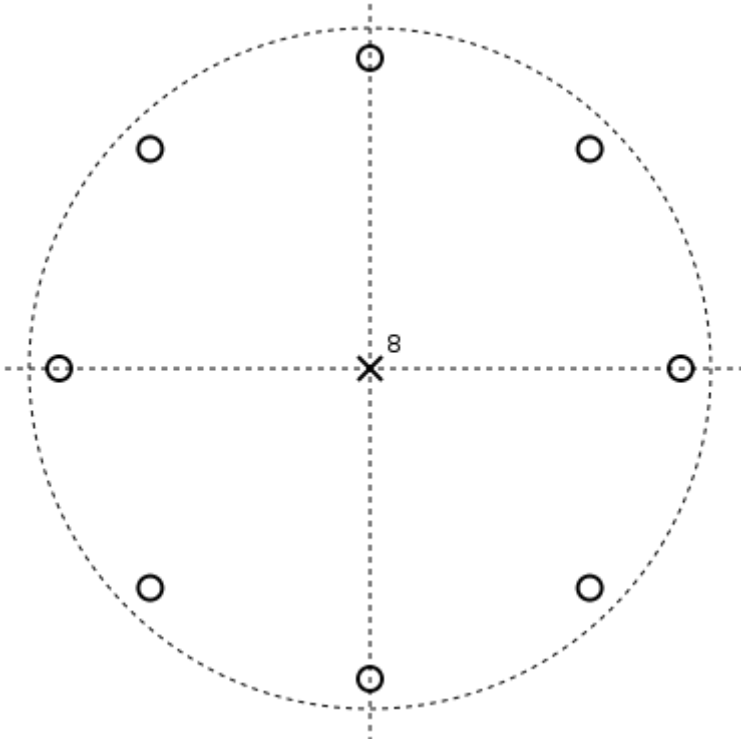
Looking again at the Z-domain transfer function of the feedforward comb filter:

$$H(z) = \frac{z^K + \alpha}{z^K}$$

we see that the numerator is equal to zero whenever $z^K = -\alpha$. This has K solutions, equally spaced around a circle in the complex plane; these are the zeros of the transfer function. The denominator is zero at $z^K = 0$, giving K poles at $z = 0$. This leads to a pole-zero plot like the ones shown below.

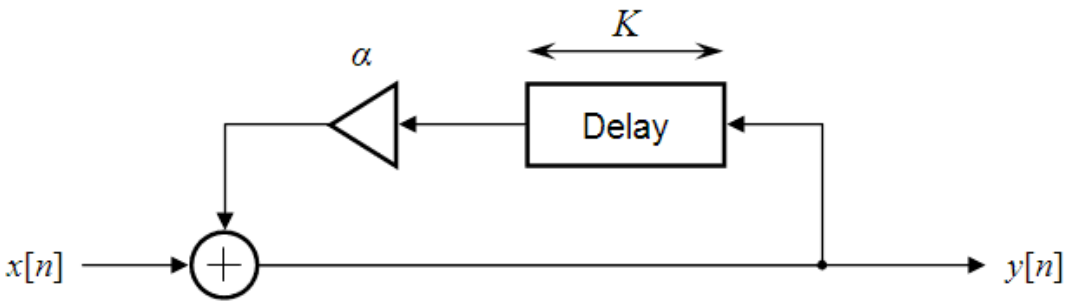


Pole-zero plot of feedfoward comb filter with $K = 8$ and $\alpha = 0.5$



Pole-zero plot of feedfoward comb filter with $K = 8$ and $\alpha = -0.5$

Feedback form



Feedback comb filter structure

Similarly, the general structure of a feedback comb filter is shown on the right. It may be described by the following difference equation:

$$y[n] = x[n] + \alpha y[n - K]$$

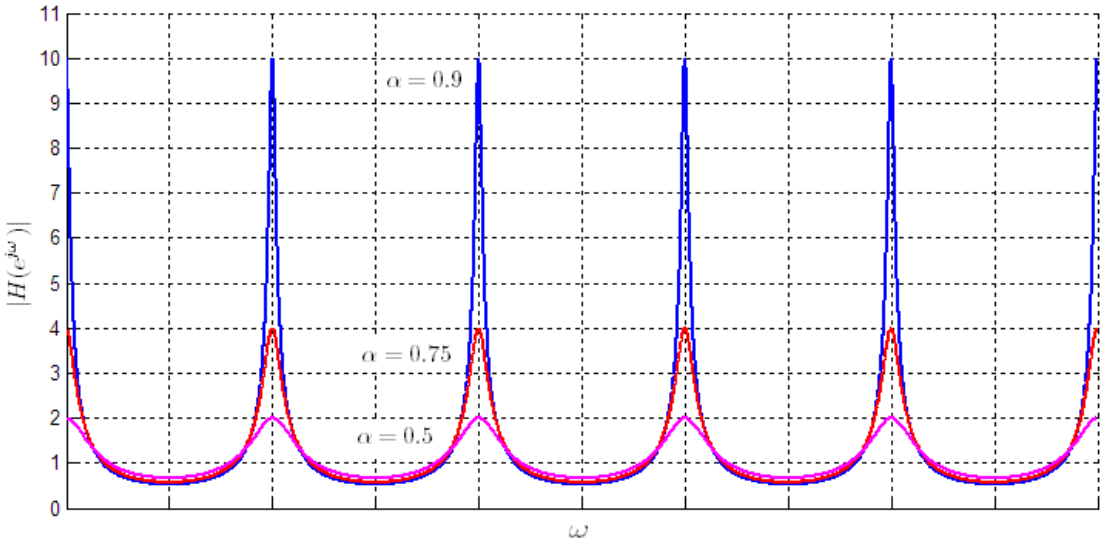
If we rearrange this equation so that all terms in *y* are on the left-hand side, and then take the Z transform, we obtain:

$$(1 - \alpha z^{-K})Y(z) = X(z)$$

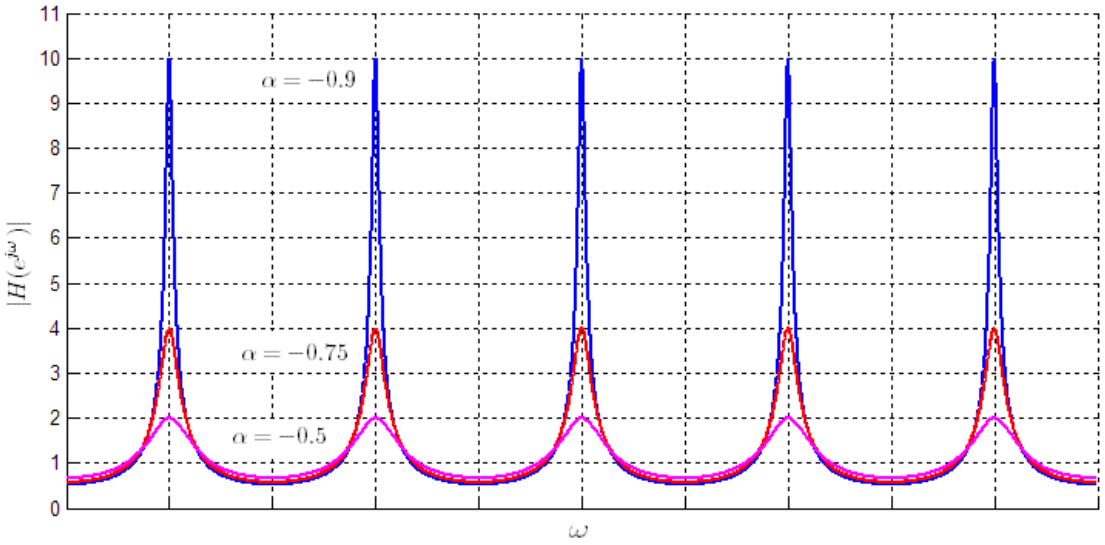
The transfer function is therefore:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \alpha z^{-K}} = \frac{z^K}{z^K - \alpha}$$

Frequency response



Feedback magnitude response for various positive values of α



Feedback magnitude response for various negative values of α

If we make the substitution $z = e^{j\omega}$ into the Z-domain expression for the feedback comb filter, we get:

$$H(e^{j\omega}) = \frac{1}{1 - \alpha e^{-j\omega K}}$$

The magnitude response is as follows:

$$|H(e^{j\omega})| = \frac{1}{\sqrt{(1+\alpha^2) - 2\alpha\cos(\omega K)}}$$

Again, the response is periodic, as the graphs to the right demonstrate. The feedback comb filter has some properties in common with the feedforward form:

- The response periodically drops to a local minimum and rises to a local maximum.
- The maxima for positive values of α coincide with the minima for negative values of α, and vice versa.

However, there are also some important differences because the magnitude response has a term in the denominator:

- The levels of the maxima and minima are no longer equidistant from 1.
- The filter is only stable if | α | is strictly less than 1. As can be seen from the graphs, as | α | increases, the amplitude of the maxima rises increasingly rapidly.

Impulse response

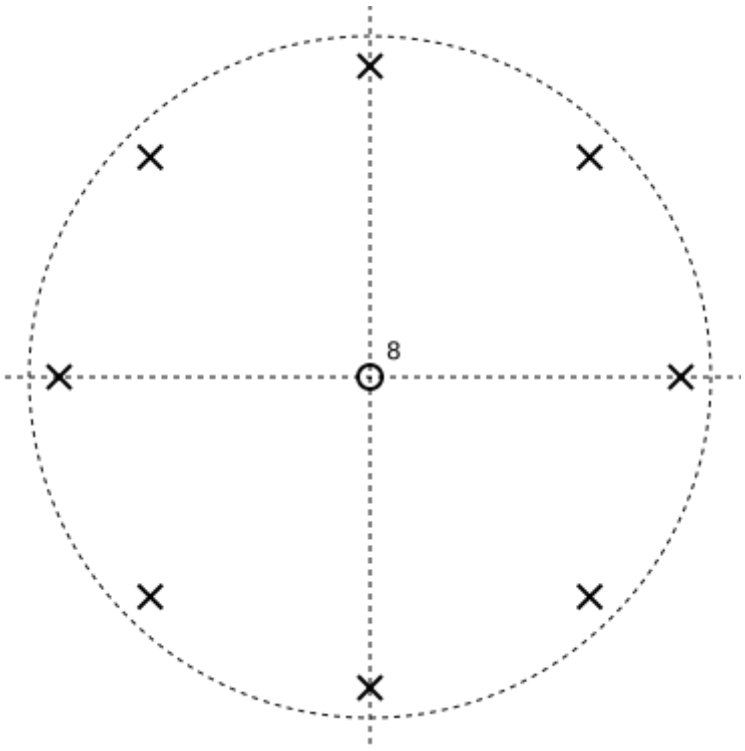
The feedback comb filter is a simple type of infinite impulse response filter. If stable, the response simply consists of a repeating series of impulses decreasing in amplitude over time.

Pole-zero interpretation

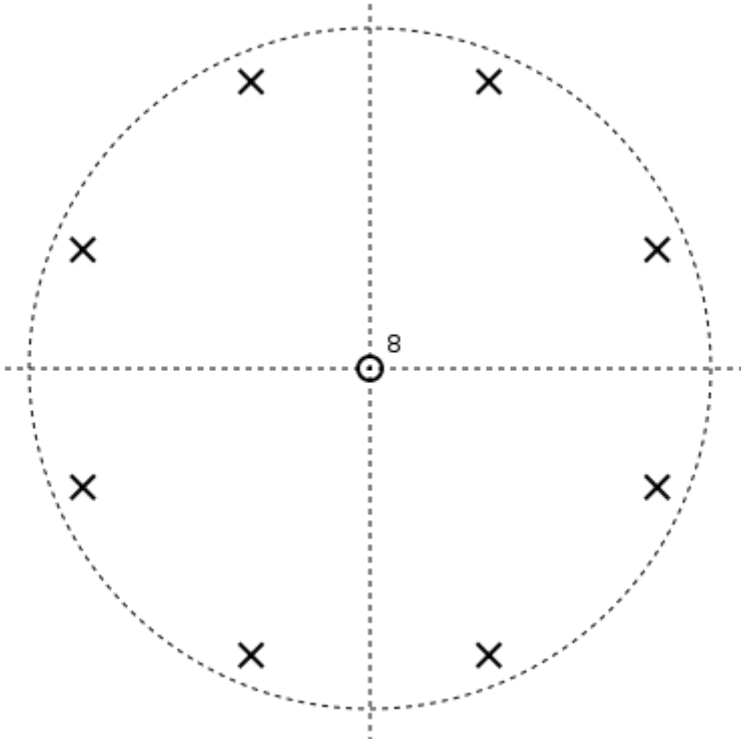
Looking again at the Z-domain transfer function of the feedback comb filter:

$$H(z) = \frac{z^K}{z^K - \alpha}$$

This time, the numerator is zero at $z^K = 0$, giving K zeros at $z = 0$. The denominator is equal to zero whenever $z^K = \alpha$. This has K solutions, equally spaced around a circle in the complex plane; these are the poles of the transfer function. This leads to a pole-zero plot like the ones shown below.



Pole-zero plot of feedback comb filter with $K = 8$ and $\alpha = 0.5$



Pole-zero plot of feedback comb filter with $K = 8$ and $\alpha = -0.5$

Continuous-time comb filters

Comb filters may also be implemented in continuous time. The feedforward form may be described by the following equation:

$$y(t) = x(t) + \alpha x(t - \tau)$$

and the feedback form by:

$$y(t) = x(t) + \alpha y(t - \tau)$$

where τ is the delay (measured in seconds).

They have the following frequency responses, respectively:

$$H(\omega) = 1 + \alpha e^{-j\omega\tau}$$
$$H(\omega) = \frac{1}{1 - \alpha e^{-j\omega\tau}}$$

Continuous-time implementations share all the properties of the respective discrete-time implementations.

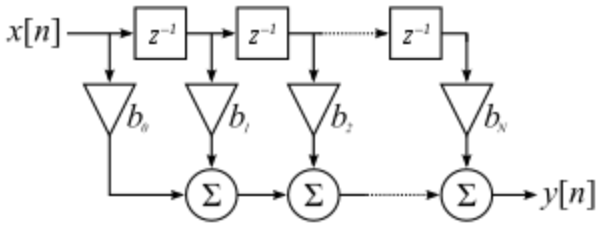
Chapter-3

Finite Impulse Response

A **finite impulse response (FIR)** filter is a type of a signal processing filter whose impulse response (or response to any finite length input) is of *finite* duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which have internal feedback and may continue to respond indefinitely (usually decaying). The impulse response of an Nth-order discrete-time FIR filter (i.e. with a Kronecker delta impulse input) lasts for N+1 samples, and then dies to zero.

FIR filters can be discrete-time or continuous-time, and digital or analog.

Definition



A discrete-time FIR filter of order *N*. The top part is an *N*-stage delay line with *N*+1 taps. Each unit delay is a z^{-1} operator in Z-transform notation.

The output *y* of a linear time invariant system is determined by convolving its input signal *x* with its impulse response *b*.

For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence *y[n]* in terms of its input sequence *x[n]*:

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$

$$y[n] = \sum_{i=0}^{N} b_i x[n-i]$$

where:

- $x[n]$ is the input signal,
- $y[n]$ is the output signal,
- b_i are the **filter coefficients**, also known as **tap weights**, that make up the impulse response,
- N is the filter order; an Nth-order filter has $(N + 1)$ terms on the right-hand side. The $x[n - i]$ in these terms are commonly referred to as **taps**, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a "5th order/6-tap filter", for instance.

Properties

An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter. FIR filters:

- Are inherently stable. This is due to the fact that, because there is no feedback, all the poles are located at the origin and thus are located within the unit circle.
- Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric; linear phase, or phase change proportional to frequency, corresponds to equal delay at all frequencies. This property is sometimes desired for phase-sensitive applications, for example data communications, crossover filters, and mastering.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed. However many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

Impulse response

The impulse response $h[n]$ can be calculated if we set $x[n] = \delta[n]$ in the above relation, where $\delta[n]$ is the Kronecker delta impulse. The impulse response for an FIR filter then becomes the set of coefficients b_n, as follows

$$\begin{aligned} h[n] &= \sum_{i=0}^{N} b_i \delta[n - i] \\ &= b_n. \end{aligned}$$

for $n = 0$ to N.

The Z-transform of the impulse response yields the transfer function of the FIR filter

$$\begin{aligned} H(z) &= Z\{h[n]\} \\ &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \\ &= \sum_{n=0}^{N} b_n \, z^{-n}. \end{aligned}$$

FIR filters are clearly *bounded-input bounded-output* (BIBO) stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than $\sum |b_i|$ times the largest value appearing in the input.

Filter design

To design a filter means to select the coefficients such that the system has specific characteristics. The required characteristics are stated in filter specifications. Most of the time filter specifications refer to the frequency response of the filter. There are different methods to find the coefficients from frequency specifications:

1. Window design method
2. Frequency Sampling method
3. Weighted least squares design
4. Parks-McClellan method (also known as the Equiripple, Optimal, or Minimax method). The Remez exchange algorithm is commonly used to find an optimal equiripple set of coefficients. Here the user specifies a desired frequency response, a weighting function for errors from this response, and a filter order *N*. The algorithm then finds the set of $(N + 1)$ coefficients that minimize the maximum deviation from the ideal. Intuitively, this finds the filter that is as close as you can get to the desired response given that you can use only $(N + 1)$ coefficients. This method is particularly easy in practice since at least one text includes a program that takes the desired filter and *N*, and returns the optimum coefficients.
5. Equiripple FIR filters can be designed using the FFT algorithms as well. The algorithm is iterative in nature. You simply compute the DFT of an initial filter design that you have using the FFT algorithm (if you don't have an initial estimate you can start with h[n]=delta[n]). In the Fourier domain or FFT domain you correct the frequency response according to your desired specs and compute the inverse FFT. In time-domain you retain only N of the coefficients (force the other coefficients to zero). Compute the FFT once again. Correct the frequency response according to specs.

Software packages like MATLAB, GNU Octave, Scilab, and SciPy provide convenient ways to apply these different methods.

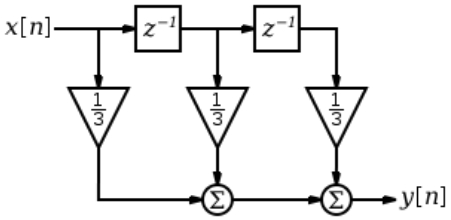
Some filter specifications refer to the time-domain shape of the input signal the filter is expected to "recognize". The optimum matched filter for separating any waveform from white noise is obtained by sampling that shape and using those samples in reverse order as the coefficients of the filter -- giving the filter an impulse response that is the time-reverse of the expected input signal.

Window design method

In the Window Design Method, one designs an ideal IIR filter, then applies a window function to it – in the time domain, multiplying the infinite impulse by the window function. This results in the frequency response of the IIR being convolved with the frequency response of the window function – thus the imperfections of the FIR filter (compared to the ideal IIR filter) can be understood in terms of the frequency response of the window function.

The ideal frequency response of a window is a Dirac delta function, as that results in the frequency response of the FIR filter being identical to that of the IIR filter, but this is not attainable for finite windows, and deviations from this yield differences between the FIR response and the IIR response.

Moving average example



Block diagram of a simple FIR filter (2nd-order/3-tap filter in this case, implementing a moving average)

A moving average filter is a very simple FIR filter. It is sometimes called a *boxcar filter*, especially when followed by decimation. The filter coefficients are found via the following equation:

$$b_i = \frac{1}{N+1} \text{for } i = 0, 1, \ldots, N$$

To provide a more specific example, we select the filter order:

$$N = 2$$

The impulse response of the resulting filter is:

$$h[n] = \frac{1}{3}\delta[n] + \frac{1}{3}\delta[n-1] + \frac{1}{3}\delta[n-2]$$

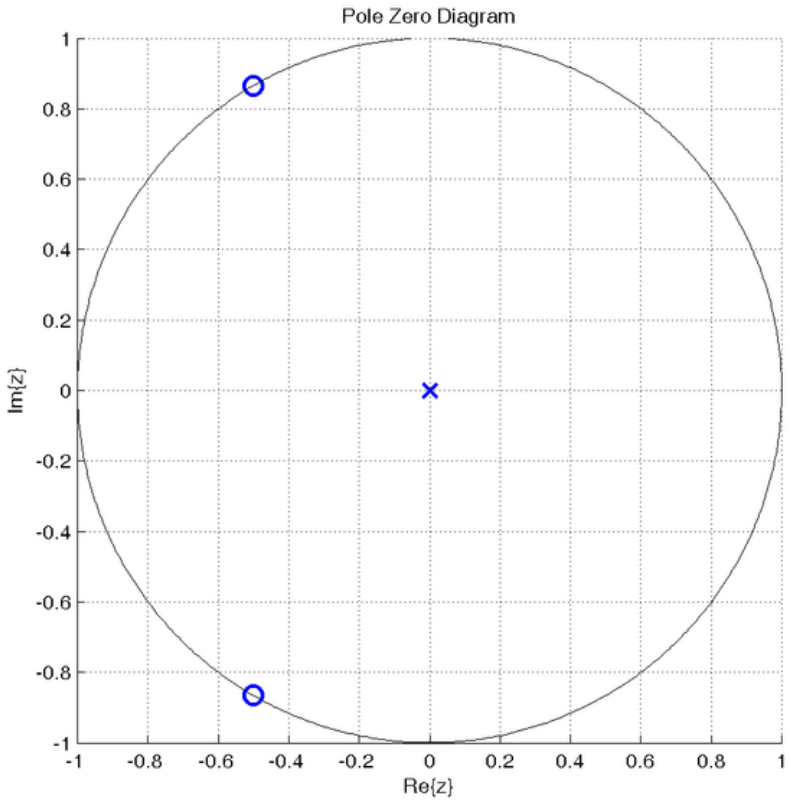
The following figure shows the block diagram of such a 2nd-order moving-average filter.

To discuss stability and spectral topics we take the z-transform of the impulse response:

$$H(z) = \frac{1}{3} + \frac{1}{3}z^{-1} + \frac{1}{3}z^{-2} = \frac{1}{3}\frac{z^2 + z + 1}{z^2}$$

The following figure shows the pole-zero diagram of the filter. Zero frequency (DC) corresponds to (1,0), positive frequencies advancing counterclockwise around the circle to (-1,0) at half the sample frequency.

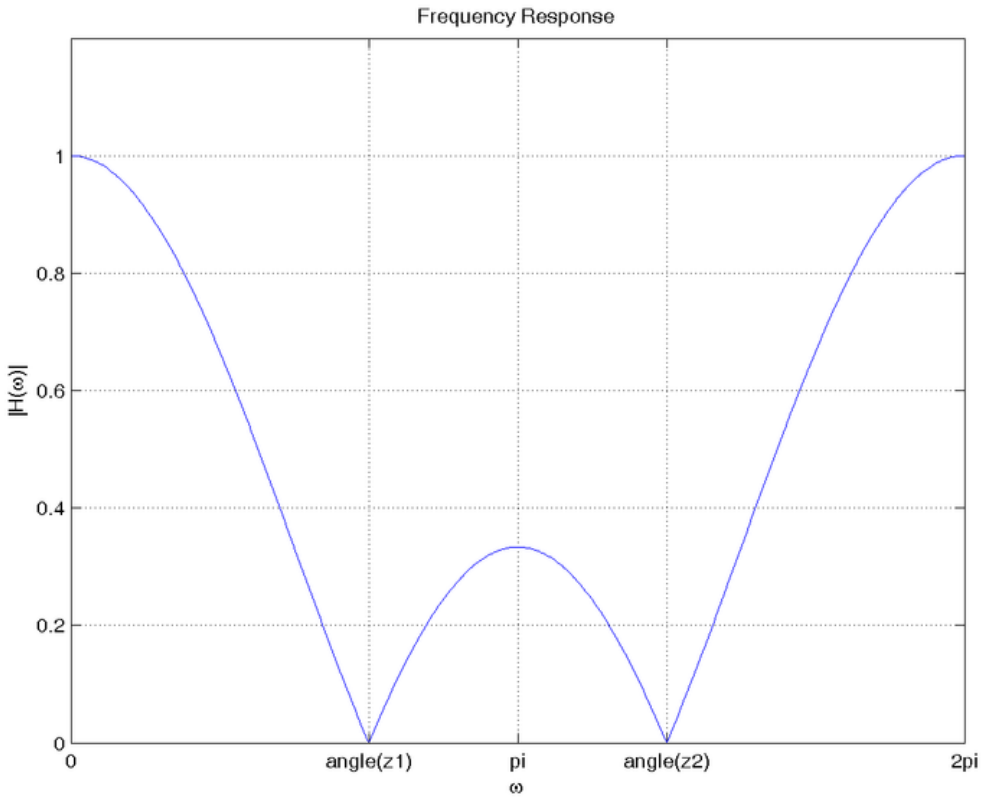
Two poles are located at the origin, and two zeros are located at $z_1 = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$, $z_2 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$



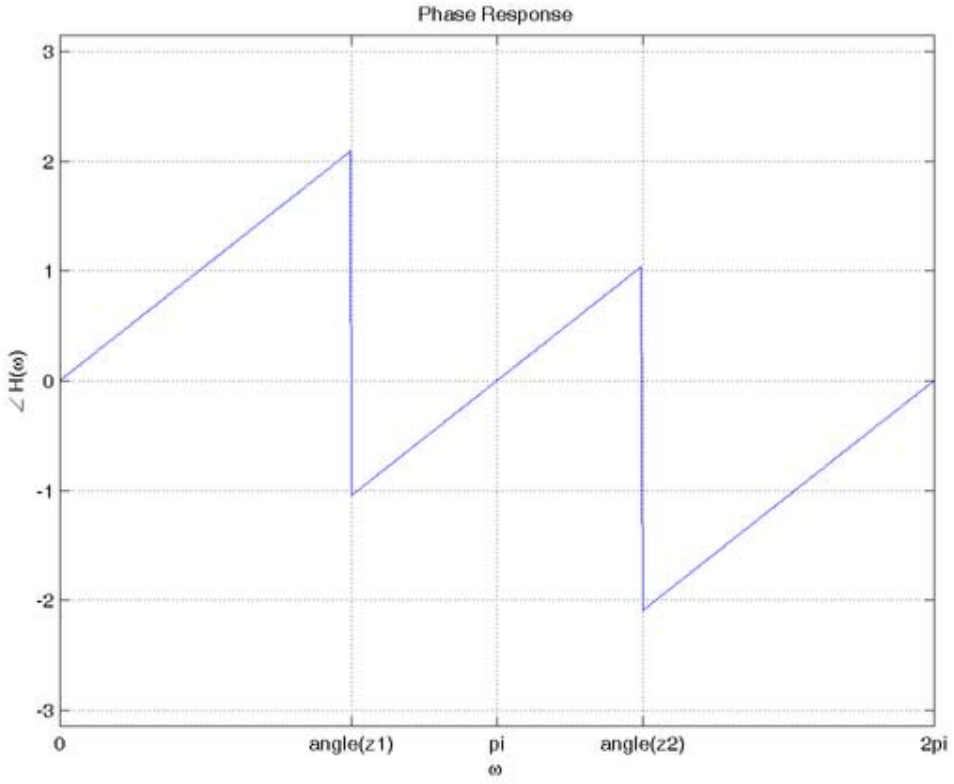
The frequency response, for frequency ω in radians per sample, is:

$$H(e^{j\omega}) = \frac{1}{3} + \frac{1}{3}e^{-j\omega} + \frac{1}{3}e^{-j2\omega}$$

The following figure shows the absolute value of the frequency response. Clearly, the moving-average filter passes low frequencies with a gain near 1, and attenuates high frequencies. This is a typical low-pass filter characteristic. Frequencies above π are aliases of the frequencies below π, and are generally ignored or filtered out if reconstructing a continuous-time signal.



The following figure shows the phase response. Since the phase always follows a straight line except where it has been reduced modulo π radians (should be 2π), the linear phase property is demonstrated.


Phase Response
3
2
1
0
-1
-2
-3
∠ H(ω)
0
angle(z1)
pi
angle(z2)
2pi
ω

Chapter-4

Dual Impedance

Dual impedance and dual network are terms used in electronic network analysis. The dual of an impedance Z is its algebraic inverse $Z' = \frac{1}{Z}$. Note that Z and Z' are the duals of each other, that is, they are reciprocal. For this reason the dual impedance is also called the inverse impedance. The dual of a network of impedances is that network whose impedance is Z'. In the case of a network with more than one port the impedance looking into each of the ports must simultaneously be dual.

Another way of stating this is that the dual of Z is the admittance $Y = Z$.

This is consistent with the definition of dual as being that circuit whose voltages and currents are interchanged since $Z = \frac{V}{I}$ and $Z' = \frac{1}{Z} = \frac{I}{V}$

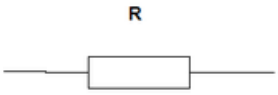
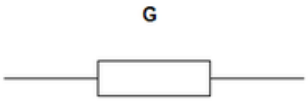
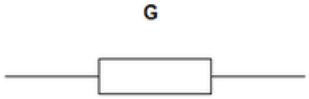
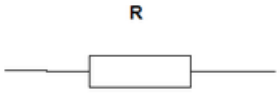
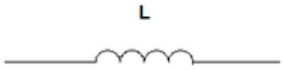
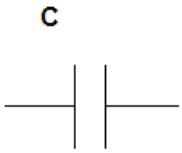
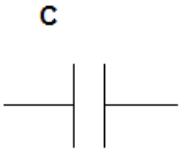
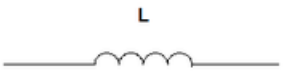
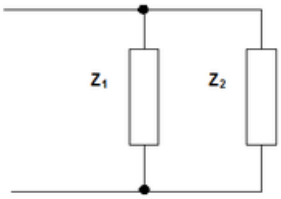
Scaled and normalised duals

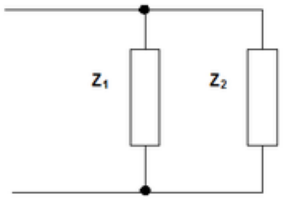
In a real design situation it is usually desired to find the dual of an impedance with respect to some nominal or characteristic impedance. To do this, Z and Z' are scaled to the nominal impedance Z_0 so that;

$$\frac{Z'}{Z_0} = \frac{Z_0}{Z}$$

Z_0 is usually taken to be a purely real number R_0, so Z' is only changed by a real factor of R_0^2. In other words, the dual remains qualitatively the same circuit but all the component values must be scaled quantitively by R_0^2.

Duals of basic circuit elements

Element	Z	Dual	Z'
 Resistor R	R	 Conductor G = R	$\frac{1}{R}$
 Conductor G	$\frac{1}{G}$	 Resistor R = G	G
 Inductor L	$i\omega L$	 Capacitor C = L	$\frac{1}{i\omega L}$
 Capacitor C	$\frac{1}{i\omega C}$	 Inductor L = C	$i\omega C$
 Series impedances Z = Z_1 + Z_2	$Z_1 + Z_2$	 Parallel admittances Y = Z_1 + Z_2	$\frac{1}{Z_1 + Z_2}$

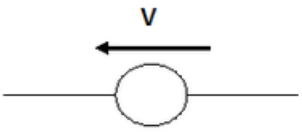


$$Z = \frac{Z_1 Z_2}{Z_1 + Z_2}$$

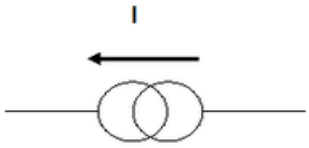
$$\frac{1}{Z_1} + \frac{1}{Z_2}$$

Parallel impedances $1/Z = 1/Z_1 + 1/Z_2$

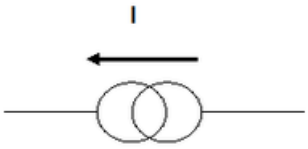
Series admittances $1/Y = 1/Z_1 + 1/Z_2$



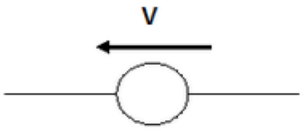
Voltage generator V



Current generator I = V



Current generator I



Voltage generator V = I

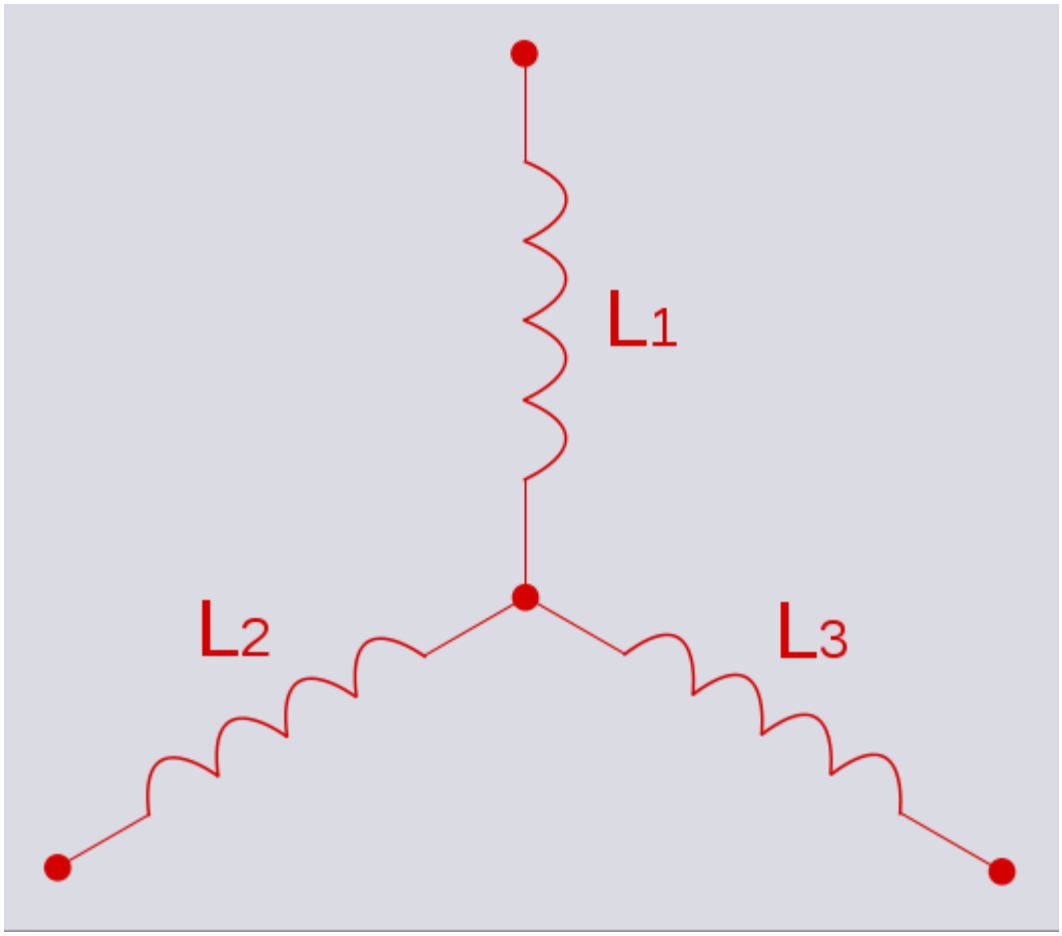
Graphical Method

There is a graphical method of obtaining the dual of a network which is often easier to use than the mathematical expression for the impedance. Starting with a circuit diagram of the network in question, Z, the following steps are drawn on the diagram to produce Z' superimposed on top of Z. Typically, Z' will be drawn in a different colour to help distinguish it from the original, or, if using CAD, Z' can be drawn on a different layer.

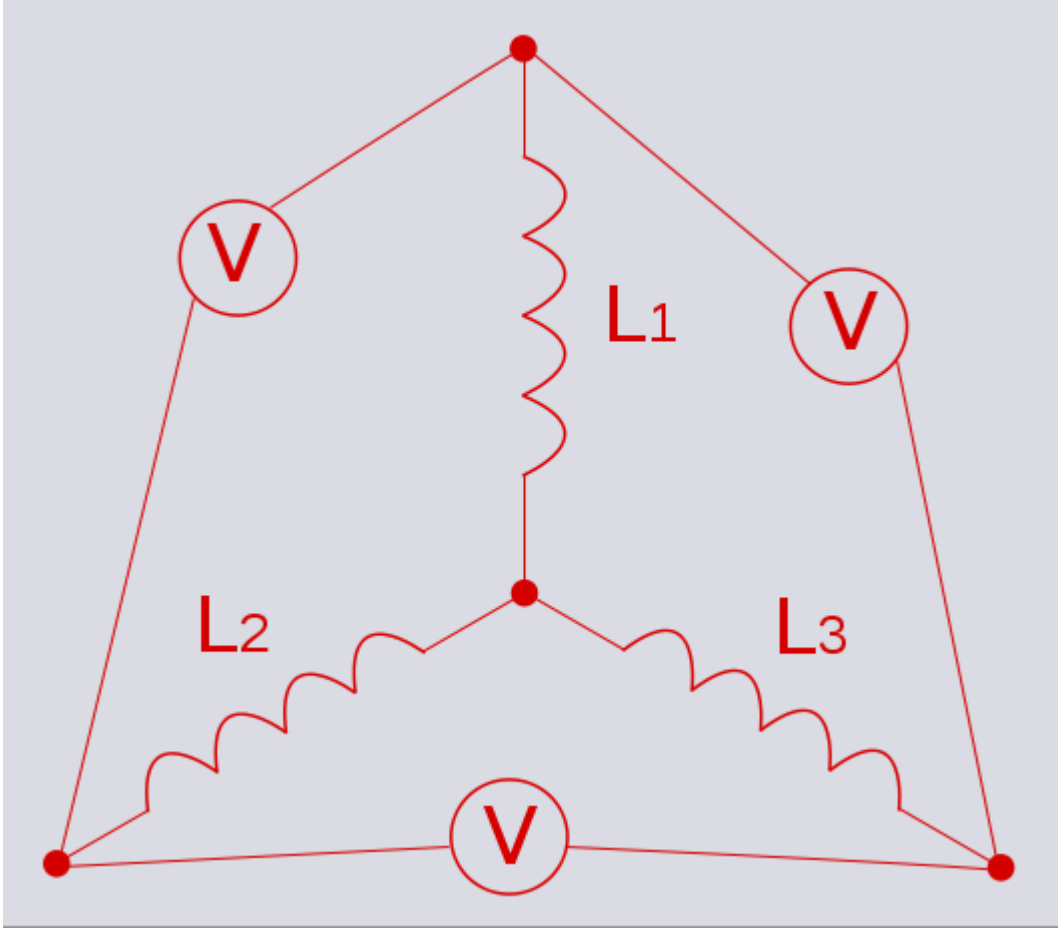
1. A generator is connected to each port of the original network. The purpose of this step is to prevent the ports from being "lost" in the inversion process. This happens because a port left open circuit will transform into a short circuit and disappear.
2. A dot is drawn at the centre of each mesh of the network Z. These dots will become the circuit nodes of Z'.
3. A conductor is drawn which entirely encloses the network Z. This conductor also becomes a node of Z'.
4. For each circuit element of Z, its dual is drawn between the nodes in the centre of the meshes either side of Z. Where Z is on the edge of the network, one of these nodes will be the enclosing conductor from the previous step.

This completes the drawing of Z'. This method also serves to demonstrate that the dual of a mesh transforms in to a node and the dual of a node transforms in to a mesh. Two useful examples are given below, both to illustrate the process and to give some further examples of dual networks.

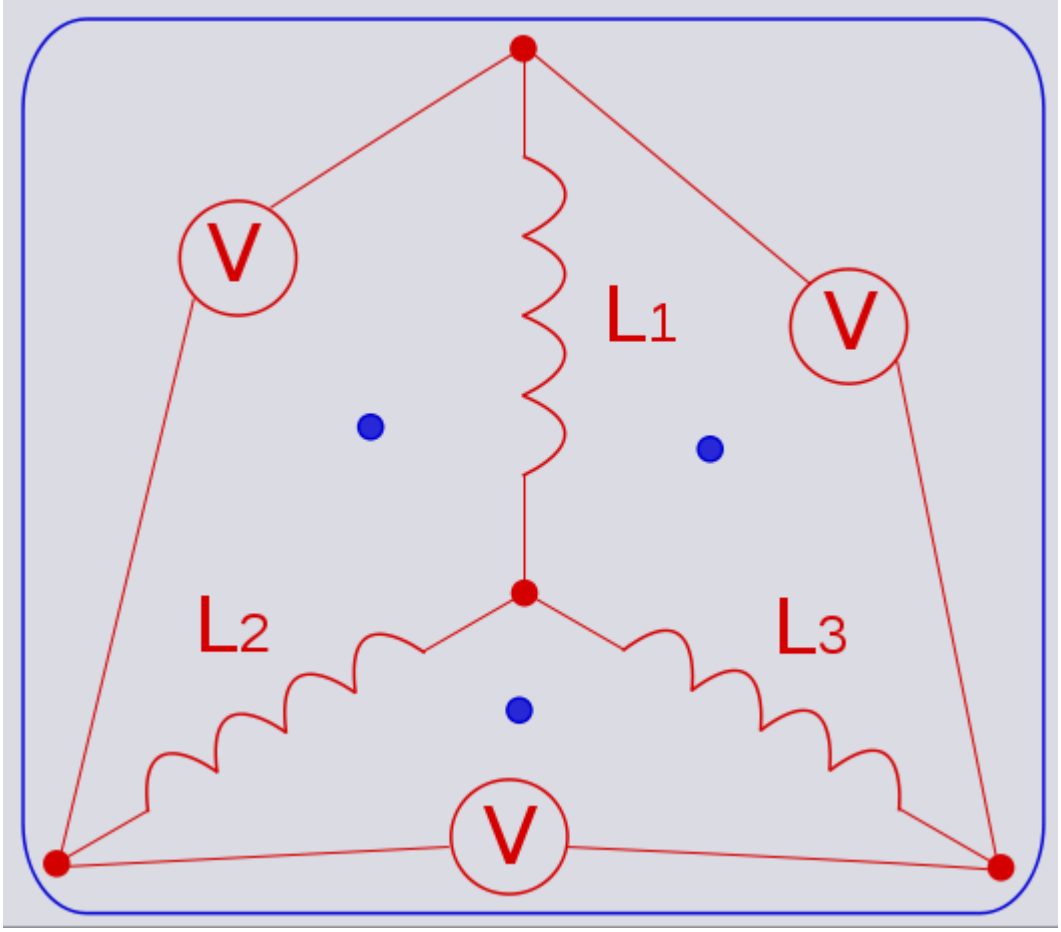
Example - star network



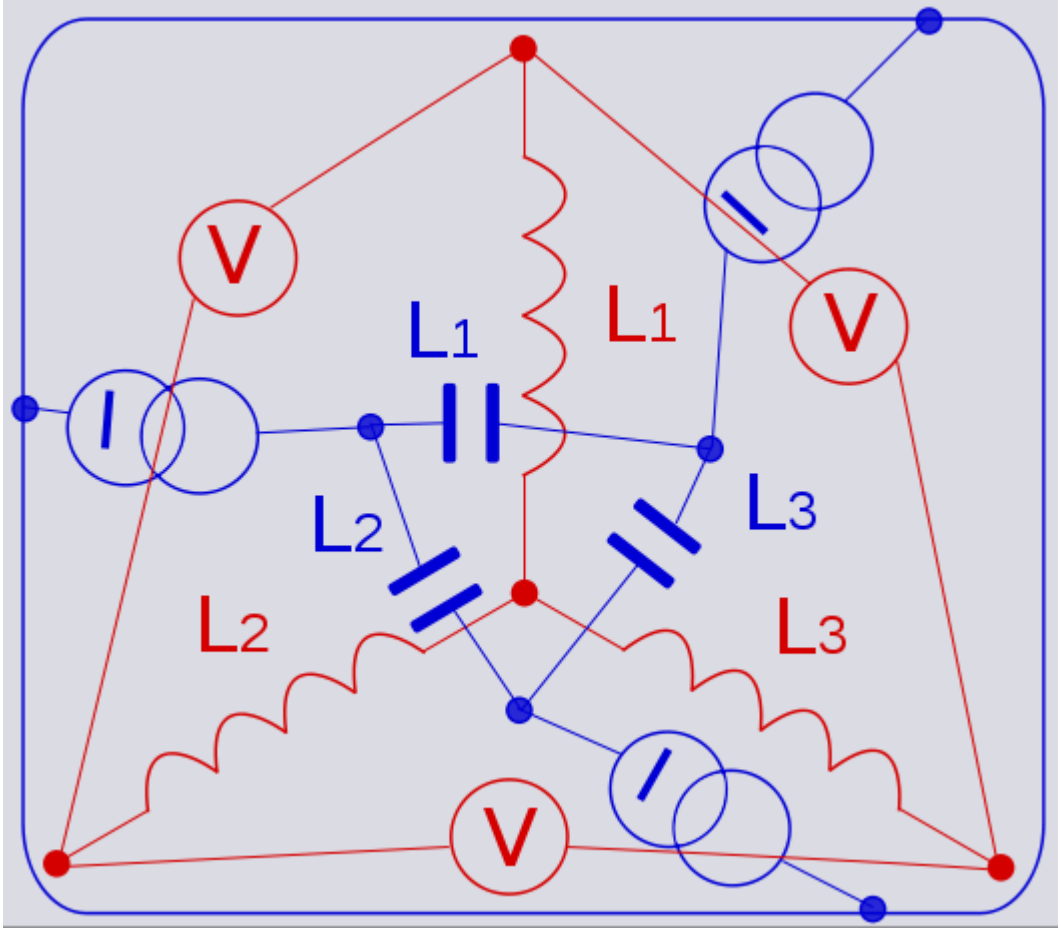
A star network of inductors, such as might be found on a three-phase transformer



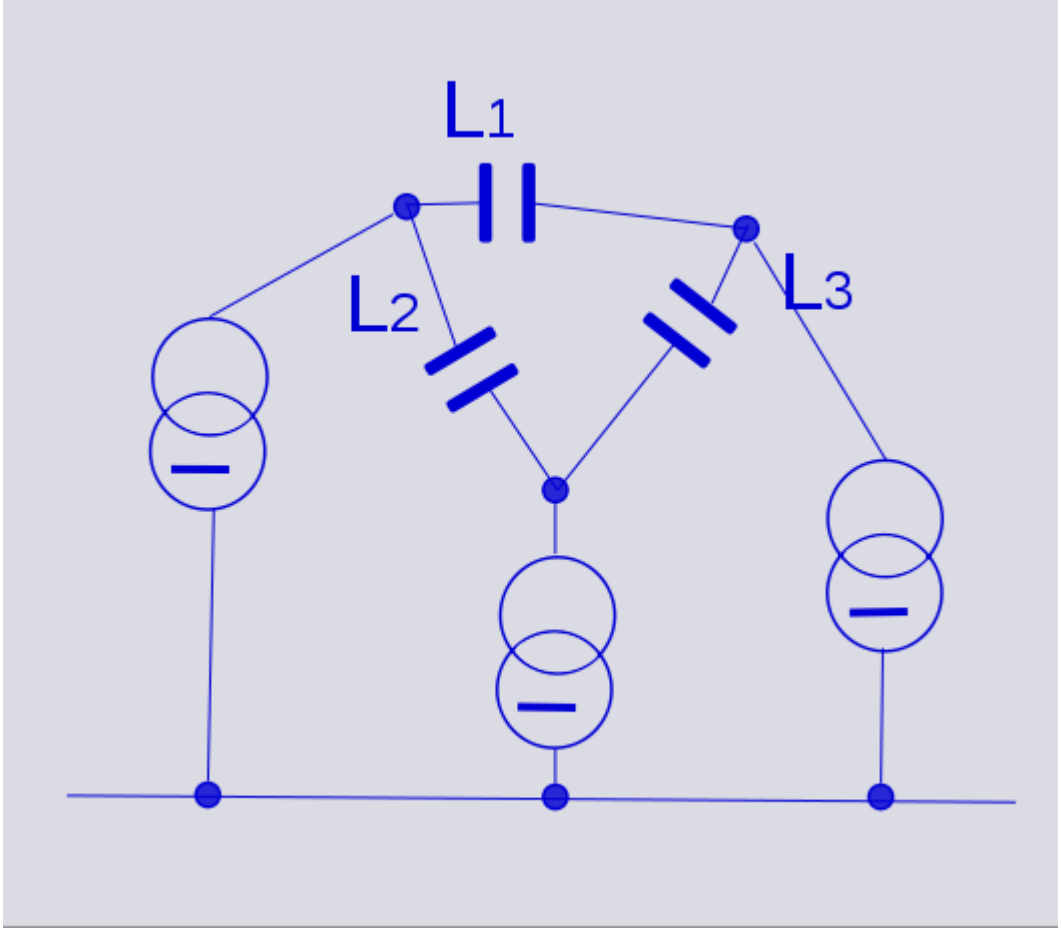
Attaching generators to the three ports



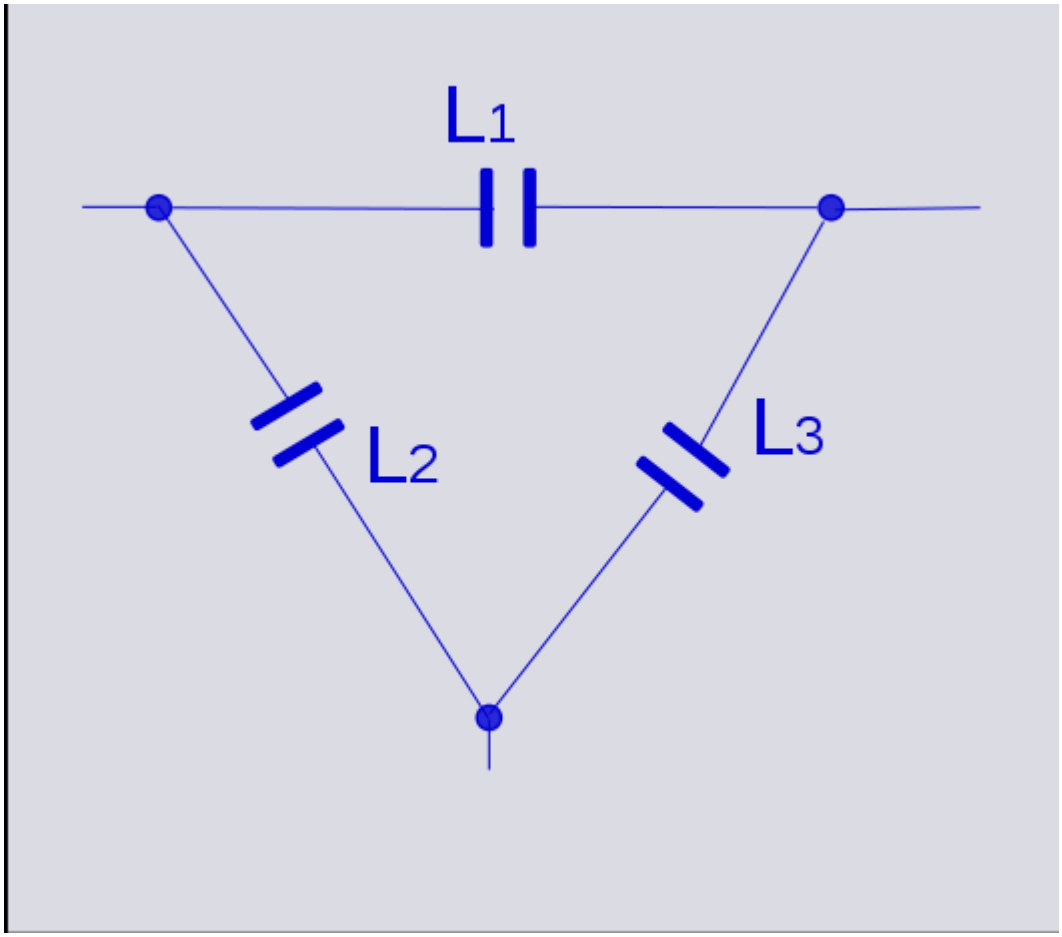
Nodes of the dual network



Components of the dual network



The dual network with the original removed and slightly redrawn to make the topology clearer

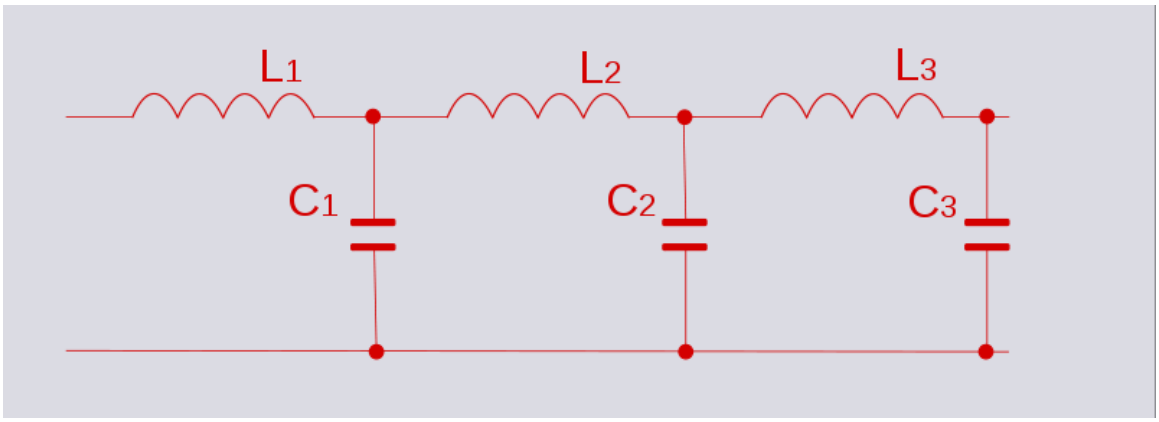


The dual network with the notional generators removed

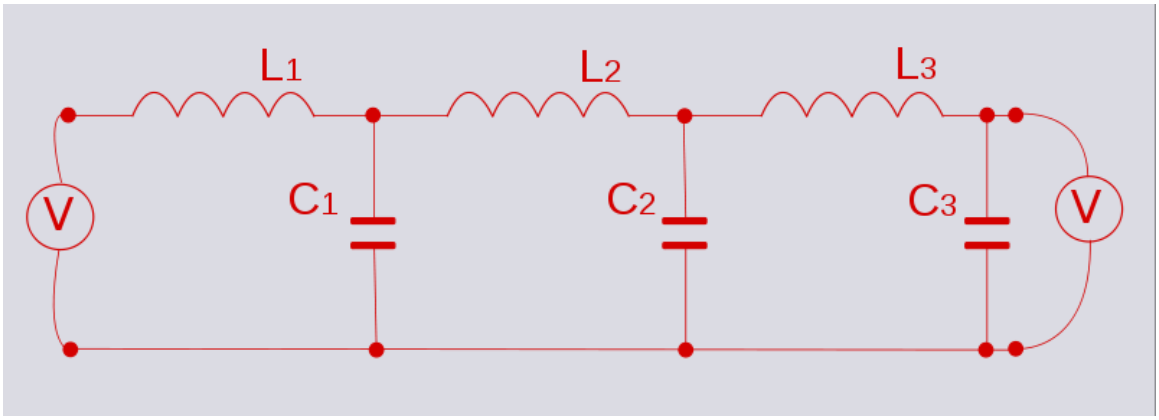
It is now clear that the dual of a star network of inductors is a delta network of capacitors. This dual circuit is not the same thing as a star-delta (Y-Δ) transformation. A Y-Δ transform results in an *equivalent* circuit, not a dual circuit.

Example - Cauer network

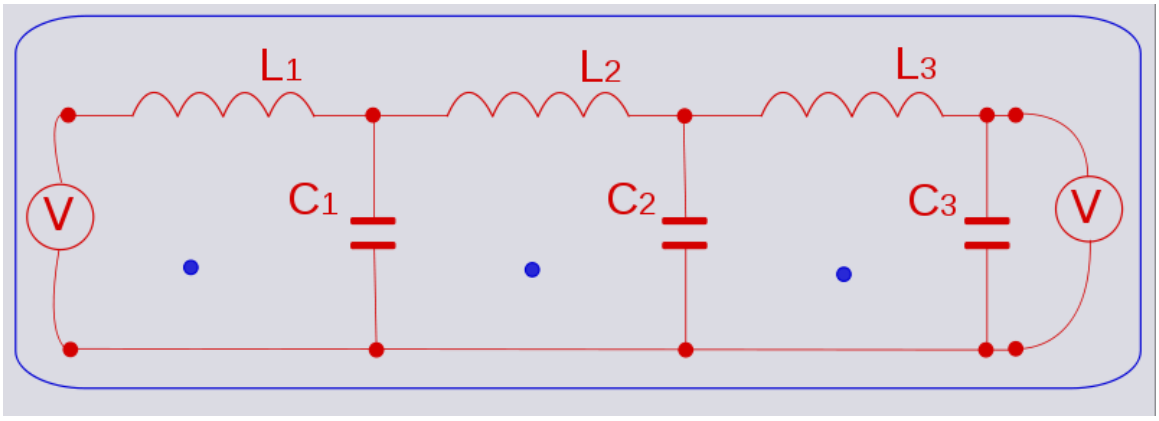
Filters designed using Cauer's topology of the first form are low-pass filters consisting of a ladder network of series inductors and shunt capacitors.



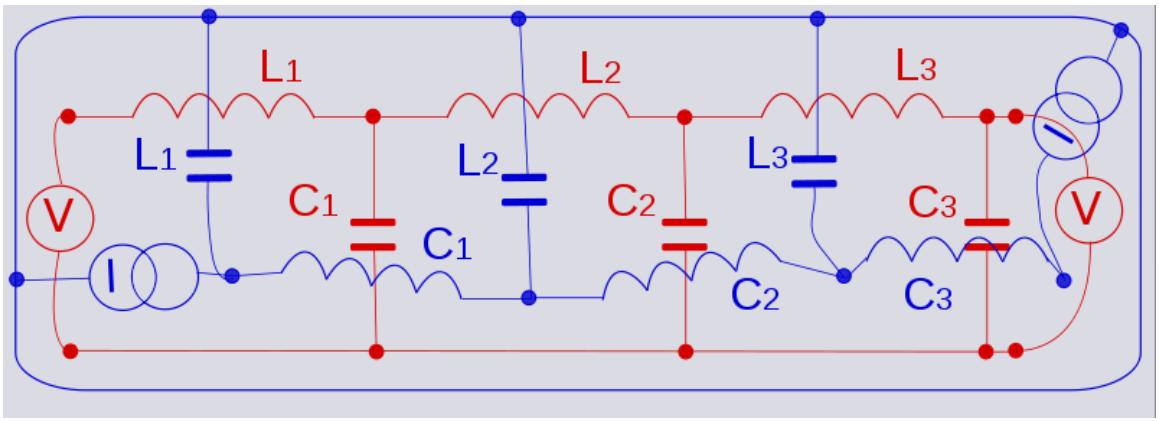
A low-pass filter implemented in Cauer topology



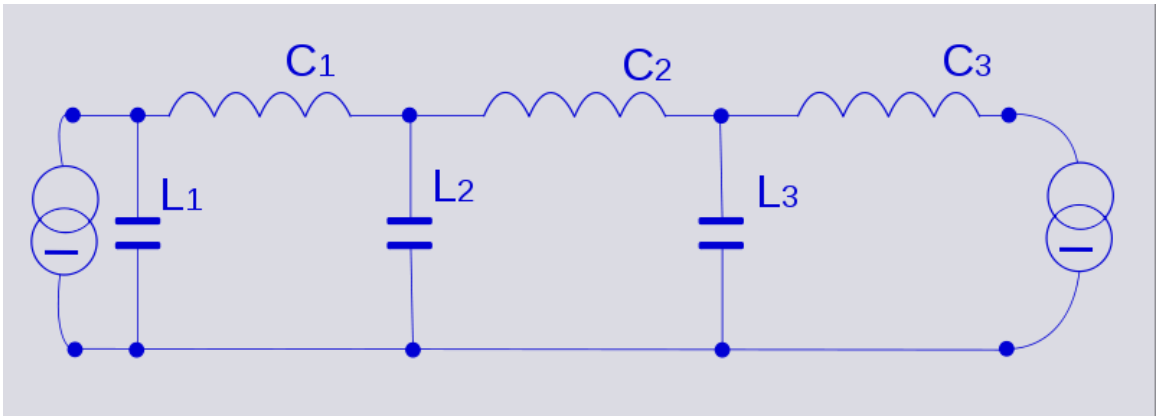
Attaching generators to the input and output ports



Nodes of the dual network



Components of the dual network



The dual network with the original removed and slightly redrawn to make the topology clearer

It can now be seen that the dual of a Cauer low-pass filter is still a Cauer low-pass filter. It does not transform into a high-pass filter as might have been expected. Note, however, that the first element is now a shunt component instead of a series component.

Chapter-5

Alpha Beta Filter and Cutoff Frequency

Alpha beta filter

An **alpha beta filter** (or alpha-beta filter) is a simplified form of observer for estimation, data smoothing and control applications. It is closely related to Kalman filters and to linear state observers used in control theory. Its principal advantage is that it does not require a detailed system model.

Filter equations

An alpha beta filter presumes that a system is adequately approximated by a model having two internal states, where the first state is obtained by integrating the value of the second state over time. Measured system output values correspond to observations of the first model state, plus disturbances. This very low order approximation is adequate for many simple systems, for example, mechanical systems where position is obtained as the time integral of velocity. Based on a mechanical system analogy, the two states can be called *position x* and *velocity v*. Assuming that velocity remains approximately constant over the small time interval ΔT between measurements, the position state is projected forward to predict its value at the next sampling time using equation 1.

$$(1) \quad \hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k-1} + \Delta \mathrm{T}\ \mathbf{v}_{k-1}$$

Since velocity variable *v* is presumed constant, so its projected value at the next sampling time equals the current value.

$$(2) \quad \hat{\mathbf{v}}_k \leftarrow \hat{\mathbf{v}}_{k-1}$$

If additional information is known about how a driving function will change the *v* state during each time interval, equation 2 can be modified to include this.

The output measurement is expected to deviate from the prediction because of noise and dynamic effects not included in the simplified dynamic model. This prediction error *r* is

also called the *residual* or *innovation*, based on statistical or Kalman filtering interpretations

$$\mathbf{(3)} \quad \hat{\mathbf{r}}_k \leftarrow \mathbf{x}_k - \hat{\mathbf{x}}_k$$

Suppose that residual *r* is positive. This could result because the previous *x* estimate was low, the previous *v* was low, or some combination of the two. The alpha beta filter takes selected *alpha* and *beta* constants (from which the filter gets its name), uses *alpha* times the deviation *r* to correct the position estimate, and uses *beta* times the deviation *r* to correct the velocity estimate. An extra ΔT factor conventionally serves to normalize magnitudes of the multipliers.

$$\mathbf{(4)} \quad \hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k + (\alpha)\ \mathbf{r}_k$$
$$\mathbf{(5)} \quad \hat{\mathbf{v}}_k \leftarrow \hat{\mathbf{v}}_k + (\beta/[\Delta \mathrm{T}])\ \mathbf{r}_k$$

The corrections can be considered small steps along an estimate of the gradient direction. As these adjustments accumulate, error in the state estimates is reduced. For convergence and stability, the values of the *alpha* and *beta* multipliers should be positive and small.

$$0 < \alpha < 1$$
$$0 < \beta < 1$$

Values of *alpha* and *beta* typically are adjusted experimentally. In general, larger *alpha* and *beta* gains tend to produce faster response for tracking transient changes, while smaller *alpha* and *beta* gains reduce the level of noise in the state estimates. If a good balance between accurate tracking and noise reduction is found, and the algorithm is effective, filtered estimates are more accurate than the direct measurements. This motivates calling the alpha-beta process a *filter*.

Algorithm Summary

Initialize.

- Set the initial values of state estimates *x* and *v*, using prior information or additional measurements; otherwise, set the initial state values to zero.
- Select values of the *alpha* and *beta* correction gains.

Update. Repeat for each time step ΔT:

```
Project state estimates x and v using equations 1 and 2
Obtain a current measurement of the output value
Compute the residual r using equation 3
Correct the state estimates using equations 4 and 5
Send updated x and optionally v as the filter outputs
```

Relationship to general state observers

More general state observers, such as the Luenberger observer for linear control systems, use a rigorous system model. Linear observers use a gain matrix to determine state estimate corrections from multiple deviations between measured variables and predicted outputs that are linear combinations of state variables. In the case of alpha beta filters, this gain matrix reduces to two terms. There is no general theory for determining the best observer gain terms, and typically gains are adjusted experimentally for both.

The linear Luenberger observer equations reduce to the alpha beta filter by applying the following specializations and simplifications.

- The discrete state transition matrix **A** is a square matrix of dimension 2, with all main diagonal terms equal to 1, and the first super-diagonal terms equal to ΔT.
- The observation equation matrix **C** has one row that selects the value of the first state variable for output.
- The filter correction gain matrix **L** has one column containing the alpha and beta gain values.
- Any known driving signal for the second state term is represented as part of the input signal vector **u**, otherwise the **u** vector is set to zero.
- Input coupling matrix **B** has a non-zero gain term as its last element if vector **u** is non-zero.

Relationship to Kalman Filters

A Kalman filter estimates the values of state variables and corrects them in a manner similar to an alpha beta filter or a state observer. However, a Kalman filter does this in a much more formal and rigorous manner. The principal differences between Kalman filters and alpha beta filters are the following.

- Like state observers, Kalman filters use a detailed dynamic system model that is not restricted to two states.
- Like state observers, Kalman filters in general use multiple observed variables to correct state variable estimates, and these do not have to be direct measurements of individual system states.
- A Kalman filter uses covariance noise models for states and observations. Using these, a time-dependent estimate of state covariance is updated automatically, and from this the Kalman gain matrix terms are calculated. Alpha beta filter gains are manually selected and static.

- For certain classes of problems, a Kalman filter is Wiener optimal, while alpha beta filtering is in general suboptimal.

A Kalman filter designed to track a moving object using a constant-velocity target dynamics (process) model (i.e., constant velocity between measurement updates) with process noise covariance and measurement covariance held constant will converge to the same structure as an alpha-beta filter. However, a Kalman filter's gain is computed recursively at each time step using the assumed process and measurement error statistics, whereas the alpha-beta's gain is computed ad hoc.

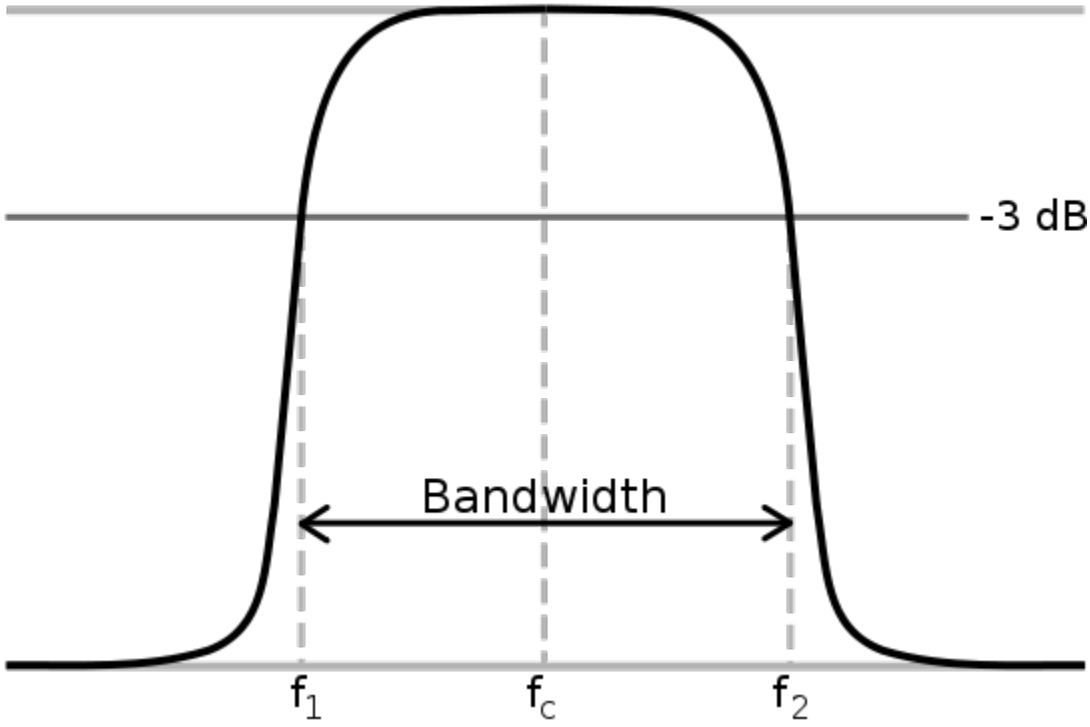
The alpha beta gamma extension

It is sometimes useful to extend the assumptions of the alpha beta filter one level. The second state variable *v* is presumed to be obtained from integrating a third *acceleration* state, analogous to the way that the first state is obtained by integrating the second. An equation for the *a* state is added to the equation system. A third multiplier, *gamma*, is selected for applying corrections to the new *a* state estimates. This yields the alpha beta gamma update equations.

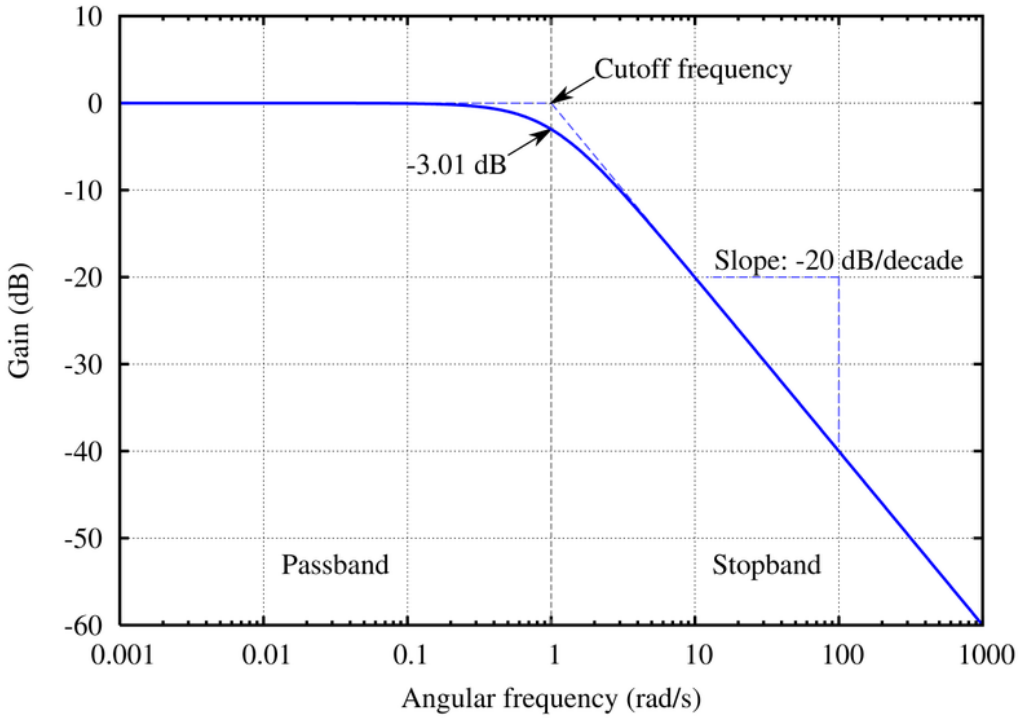
$$\begin{aligned}
\hat{\mathbf{x}}_k &\leftarrow \hat{\mathbf{x}}_k + (\alpha)\ \mathbf{r}_k \\
\hat{\mathbf{v}}_k &\leftarrow \hat{\mathbf{v}}_k + (\beta/[\Delta \mathrm{T}])\ \mathbf{r}_k \\
\hat{\mathbf{a}}_k &\leftarrow \hat{\mathbf{a}}_k + (\gamma/[2\ \Delta \mathrm{T}^2])\ \mathbf{r}_k
\end{aligned}$$

Similar extensions to additional higher orders are possible, but most systems of higher order tend to have significant interactions among the multiple states, so approximating the system dynamics as a simple integrator chain is less likely to prove useful.

Cutoff frequency



Magnitude transfer function of a bandpass filter with lower 3dB cutoff frequency f_1 and upper 3dB cutoff frequency f_2



A bode plot of the Butterworth filter's frequency response, with corner frequency labeled. (The slope −20 dB per decade also equals −6 dB per octave.)

In physics and electrical engineering, a **cutoff frequency**, **corner frequency**, or **break frequency** is a boundary in a system's frequency response at which energy flowing through the system begins to be reduced (attenuated or reflected) rather than passing through.

Typically in electronic systems such as filters and communication channels, cutoff frequency applies to an edge in a lowpass, highpass, bandpass, or band-stop characteristic – a frequency characterizing a boundary between a passband and a stopband. It is sometimes taken to be the point in the filter response where a transition band and passband meet, for example as defined by a 3 dB corner, a frequency for which the output of the circuit is -3 dB of the nominal passband value. Alternatively, a stopband corner frequency may be specified as a point where a transition band and a stopband meet: a frequency for which the attenuation is larger than the required stopband attenuation, which for example may be 30 dB or 100 dB.

In the case of a waveguide or an antenna, the cutoff frequencies correspond to the lower and upper **cutoff wavelengths**.

Cutoff frequency can also refer to the plasma frequency.

Electronics

In electronics, cutoff frequency or corner frequency is the frequency either above or below which the power output of a circuit, such as a line, amplifier, or electronic filter has fallen to a given proportion of the power in the passband. Most frequently this proportion is one half the passband power, also referred to as the 3dB point since a fall of 3dB corresponds approximately to half power. As a voltage ratio this is a fall to $\sqrt{1/2} \approx 0.707$ of the passband voltage.

However, other ratios are sometimes more convenient. For instance, in the case of the Chebyshev filter it is usual to define the cutoff frequency as the point after the last peak in the frequency response at which the level has fallen to the design value of the passband ripple. The amount of ripple in this class of filter can be set by the designer to any desired value, hence the ratio used could be any value.

Communications

In communications, the term cutoff frequency can mean the frequency below which a radio wave fails to penetrate a layer of the ionosphere at the incidence angle required for transmission between two specified points by reflection from the layer.

Waveguides

The cutoff frequency of an electromagnetic waveguide is the lowest frequency for which a mode will propagate in it. In fiber optics, it is more common to consider the **cutoff wavelength**, the maximum wavelength that will propagate in an optical fiber or waveguide. The cutoff frequency is found with the characteristic equation of the Helmholtz equation for electromagnetic waves, which is derived from the electromagnetic wave equation by setting the longitudinal wave number equal to zero and solving for the frequency. Thus, any exciting frequency lower than the cutoff frequency will attenuate, rather than propagate. The following derivation assumes lossless walls. The value of c, the speed of light, should be taken to be the group velocity of light in whatever material fills the waveguide.

For a rectangular waveguide, the cutoff frequency is

$$\omega_c = c\sqrt{\left(\frac{n\pi}{a}\right)^2 + \left(\frac{m\pi}{b}\right)^2},$$

where $n, m \geq 0$ are the mode numbers and *a* and *b* the lengths of the sides of the rectangle.

The cutoff frequency of the TM_{01} mode in a waveguide of circular cross-section (the transverse-magnetic mode with no angular dependence and lowest radial dependence) is given by

$$\omega_c = c\frac{\chi_{01}}{r} = c\frac{2.4048}{r},$$

where r is the radius of the waveguide, and χ_{01} is the first root of $J_0(r)$, the bessel function of the first kind of order 1.

For a single-mode optical fiber, the cutoff wavelength is the wavelength at which the normalized frequency is approximately equal to 2.405.

Mathematical analysis

The starting point is the wave equation (which is derived from the Maxwell equations),

$$\left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\psi(\mathbf{r},t) = 0,$$

which becomes a Helmholtz equation by considering only functions of the form

$$\psi(x,y,z,t) = \psi(x,y,z)e^{i\omega t}.$$

Substituting and evaluating the time derivative gives

$$(\nabla^2 + \frac{\omega^2}{c^2})\psi(x,y,z) = 0.$$

The function ψ here refers to whichever field (the electric field or the magnetic field) has no vector component in the longitudinal direction - the "transverse" field. It is a property of all the eigenmodes of the electromagnetic waveguide that at least one of the two fields is transverse. The z axis is defined to be along the axis of the waveguide.

The "longitudinal" derivative in the Laplacian can further be reduced by considering only functions of the form

$$\psi(x,y,z,t) = \psi(x,y)e^{i(\omega t - k_z z)},$$

where k_z is the longitudinal wavenumber, resulting in

$$(\nabla_T^2 - k_z^2 + \frac{\omega^2}{c^2})\psi(x,y,z) = 0,$$

where subscript T indicates a 2-dimensional transverse Laplacian. The final step depends on the geometry of the waveguide. The easiest geometry to solve is the rectangular waveguide. In that case the remainder of the Laplacian can be evaluated to its characteristic equation by considering solutions of the form

$$\psi(x,y,z,t) = \psi_0 e^{i(\omega t - k_z z - k_x x - k_y y)}.$$

Thus for the rectangular guide the Laplacian is evaluated, and we arrive at

$$\frac{\omega^2}{c^2} = k_x^2 + k_y^2 + k_z^2$$

The transverse wavenumbers can be specified from the standing wave boundary conditions for a rectangular geometry crossection with dimensions *a* and *b*:

$$k_x = \frac{n\pi}{a},$$
$$k_y = \frac{m\pi}{b},$$

where *n* and *m* are the two integers representing a specific eigenmode. Performing the final substitution, we obtain

$$\frac{\omega^2}{c^2} = \left(\frac{n\pi}{a}\right)^2 + \left(\frac{m\pi}{b}\right)^2 + k_z^2,$$

which is the dispersion relation in the rectangular waveguide. The cutoff frequency ω_c is the critical frequency between propagation and attenuation, which corresponds to the frequency at which the longitudinal wavenumber k_z is zero. It is given by

$$\omega_c = c\sqrt{\left(\frac{n\pi}{a}\right)^2 + \left(\frac{m\pi}{b}\right)^2}$$

The wave equations are also valid below the cutoff frequency, where the longitudinal wave number is imaginary. In this case, the field decays exponentially along the waveguide axis.

Chapter-6

Passive Analogue Filter Development

Analogue filters are a basic building block of signal processing much used in electronics. Amongst their many applications are the separation of an audio signal before application to bass, mid-range and tweeter loudspeakers; the combining and later separation of multiple telephone conversations onto a single channel; the selection of a chosen radio station in a radio receiver and rejection of others.

Passive linear electronic analogue filters are those filters which can be described with linear differential equations (linear); they are composed of capacitors, inductors and, sometimes, resistors (passive) and are designed to operate on continuously varying (analogue) signals. There are many linear filters which are not analogue in implementation (digital filter), and there are many electronic filters which may not have a passive topology – both of which may have the same transfer function of the filters described here. Analogue filters are most often used in wave filtering applications, that is, where it is required to pass particular frequency components and to reject others from analogue (continuous-time) signals.

Analogue filters have played an important part in the development of electronics. Especially in the field of telecommunications, filters have been of crucial importance in a number of technological breakthroughs and have been the source of enormous profits for telecommunications companies. It should come as no surprise, therefore, that the early development of filters was intimately connected with transmission lines. Transmission line theory gave rise to filter theory, which initially took a very similar form, and the main application of filters was for use on telecommunication transmission lines. However, the arrival of network synthesis techniques greatly enhanced the degree of control of the designer.

Today, it is often preferred to carry out filtering in the digital domain where complex algorithms are much easier to implement, but analogue filters do still find applications, especially for low-order simple filtering tasks and are often still the norm at higher frequencies where digital technology is still impractical, or at least, less cost effective. Wherever possible, and especially at low frequencies, analogue filters are now implemented in a filter topology which is active in order to avoid the wound components required by passive topology.

It is possible to design linear analogue mechanical filters using mechanical components which filter mechanical vibrations or acoustic waves. While there are few applications for such devices in mechanics per se, they can be used in electronics with the addition of transducers to convert to and from the electrical domain. Indeed some of the earliest ideas for filters were acoustic resonators because the electronics technology was poorly understood at the time. In principle, the design of such filters can be achieved entirely in terms of the electronic counterparts of mechanical quantities, with kinetic energy, potential energy and heat energy corresponding to the energy in inductors, capacitors and resistors respectively.

Historical overview

There are three main stages in the history of **passive analogue filter development**:

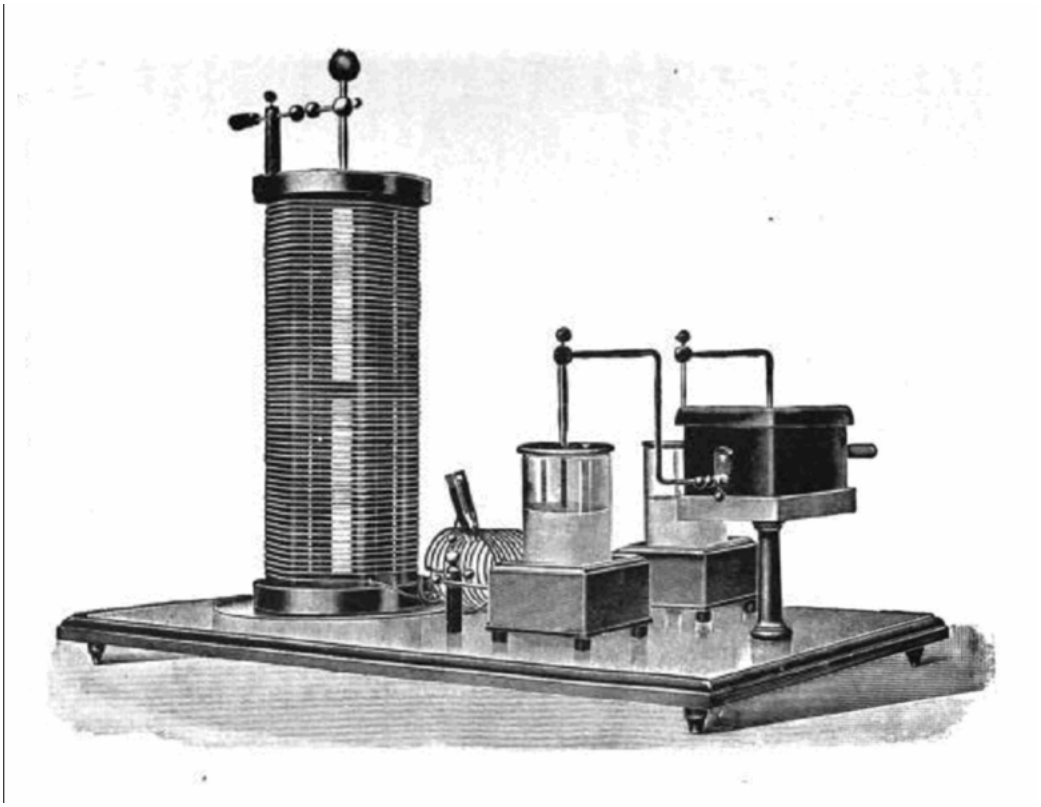
1. **Simple filters**. The frequency dependence of electrical response was known for capacitors and inductors from very early on. The resonance phenomenon was also familiar from an early date and it was possible to produce simple, single-branch filters with these components. Although attempts were made in the 1880s to apply them to telegraphy, these designs proved inadequate for successful frequency division multiplexing. Network analysis was not yet powerful enough to provide the theory for more complex filters and progress was further hampered by a general failure to understand the frequency domain nature of signals.
2. **Image filters**. Image filter theory grew out of transmission line theory and the design proceeded in a similar manner to transmission line analysis. For the first time filters could be produced that had precisely controllable passbands and other parameters. These developments took place in the 1920s and filters produced to these designs were still in widespread use in the 1980s, only declining as the use of analogue telecommunications has declined. Their immediate application was the economically important development of frequency division multiplexing for use on intercity and international telephony lines.
3. **Network synthesis filters**. The mathematical bases of network synthesis were laid in the 1930s and 1940s. After the end of World War II network synthesis became the primary tool of filter design. Network synthesis put filter design on a firm mathematical foundation, freeing it from the mathematically sloppy techniques of image design and severing the connection with physical lines. The essence of network synthesis is that it produces a design that will (at least if implemented with ideal components) accurately reproduce the response originally specified in black box terms.

Throughout the letters R,L and C are used with their usual meanings to represent resistance, inductance and capacitance, respectively. In particular they are used in combinations, such as LC, to mean, for instance, a network consisting only of inductors and capacitors. Z is used for electrical impedance, any 2-terminal combination of RLC elements and in some sections D is used for the rarely seen quantity elastance, which is the inverse of capacitance.

Resonance

Early filters utilised the phenomenon of resonance to filter signals. Although electrical resonance had been investigated by researchers from a very early stage, it was at first not widely understood by electrical engineers. Consequently, the much more familiar concept of acoustic resonance (which in turn, can be explained in terms of the even more familiar mechanical resonance) found its way into filter design ahead of electrical resonance. Resonance can be used to achieve a filtering effect because the resonant device will respond to frequencies at, or near, to the resonant frequency but will not respond to frequencies far from resonance. Hence frequencies far from resonance are filtered out from the output of the device.

Electrical resonance



A 1915 example of an early type of resonant circuit known as an Oudin coil which uses Leyden jars for the capacitance.

Resonance was noticed early on in experiments with the Leyden jar, invented in 1746. The Leyden jar stores electricity due to its capacitance, and is, in fact, an early form of capacitor. When a Leyden jar is discharged by allowing a spark to jump between the electrodes, the discharge is oscillatory. This was not suspected until 1826, when Felix Savary in France, and later (1842) Joseph Henry in the US noted that a steel needle placed close to the discharge does not always magnetise in the same direction. They both independently drew the conclusion that there was a transient oscillation dying with time.

Hermann von Helmholtz in 1847 published his important work on conservation of energy in part of which he used those principles to explain why the oscillation dies away, that it is the resistance of the circuit which dissipates the energy of the oscillation on each successive cycle. Helmholtz also noted that there was evidence of oscillation from the electrolysis experiments of William Hyde Wollaston. Wollaston was attempting to decompose water by electric shock but found that both hydrogen and oxygen were present at both electrodes. In normal electrolysis they would separate, one to each electrode.

Helmholtz explained why the oscillation decayed but he had not explained why it occurred in the first place. This was left to Sir William Thomson (Lord Kelvin) who, in 1853, postulated that there was inductance present in the circuit as well as the capacitance of the jar and the resistance of the load. This established the physical basis for the phenomenon - the energy supplied by the jar was partly dissipated in the load but also partly stored in the magnetic field of the inductor.

So far, the investigation had been on the natural frequency of transient oscillation of a resonant circuit resulting from a sudden stimulus. More important from the point of view of filter theory is the behaviour of a resonant circuit when driven by an external AC signal: there is a sudden peak in the circuits response when the driving signal frequency is at the resonant frequency of the circuit. James Clerk Maxwell heard of the phenomenon from Sir William Grove in 1868 in connection with experiments on dynamos, and was also aware of the earlier work of Henry Wilde in 1866. Maxwell explained resonance mathematically, with a set of differential equations, in much the same terms that an RLC circuit is described today.

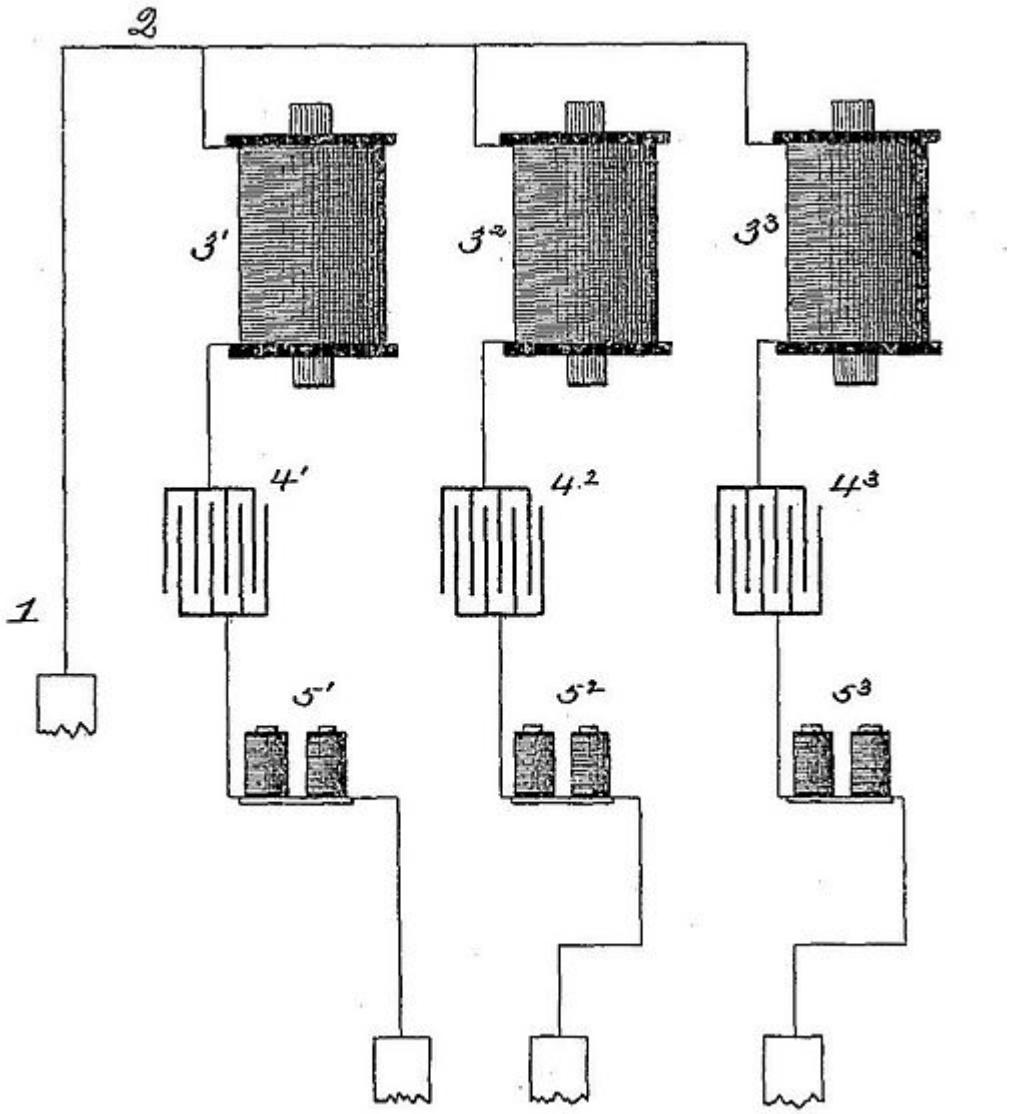
Heinrich Hertz (1887) experimentally demonstrated the resonance phenomena by building two resonant circuits, one of which was driven by a generator and the other was tunable and only coupled to the first electromagnetically (i.e., no circuit connection). Hertz showed that the response of the second circuit was at a maximum when it was in tune with the first. The diagrams produced by Hertz in this paper were the first published plots of an electrical resonant response.

Acoustic resonance

As mentioned earlier, it was acoustic resonance that inspired filtering applications, the first of these being a telegraph system known as the "harmonic telegraph". Versions are due to Elisha Gray, Alexander Graham Bell (1870s), Ernest Mercadier and others. Its purpose was to simultaneously transmit a number of telegraph messages over the same line and represents an early form of frequency division multiplexing (FDM). FDM requires the sending end to be transmitting at different frequencies for each individual communication channel. This demands individual tuned resonators, as well as filters to separate out the signals at the receiving end. The harmonic telegraph achieved this with electromagnetically driven tuned reeds at the transmitting end which would vibrate similar reeds at the receiving end. Only the reed with the same resonant frequency as the transmitter would vibrate to any appreciable extent at the receiving end.

Incidentally, the harmonic telegraph directly suggested to Bell the idea of the telephone. The reeds can be viewed as transducers converting sound to and from an electrical signal. It is no great leap from this view of the harmonic telegraph to the idea that speech can be converted to and from an electrical signal.

Early multiplexing



Hutin and Leblanc's multiple telegraph filter of 1891 showing the use of resonant circuits in filtering.

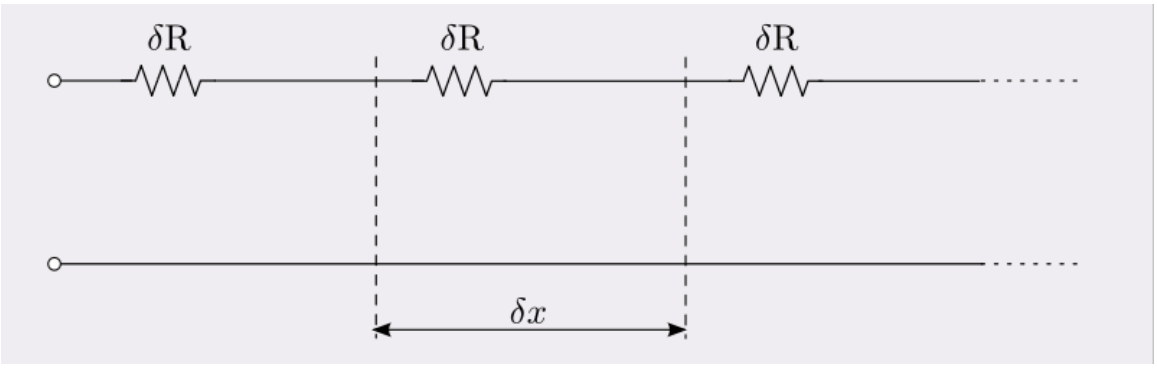
By the 1890s electrical resonance was much more widely understood and had become a normal part of the engineer's toolkit. In 1891 Hutin and Leblanc patented an FDM scheme for telephone circuits using resonant circuit filters. Rival patents were filed in 1892 by Michael Pupin and John Stone Stone with similar ideas, priority eventually being

awarded to Pupin. However, no scheme using just simple resonant circuit filters can successfully multiplex (i.e. combine) the wider bandwidth of telephone channels (as opposed to telegraph) without either an unacceptable restriction of speech bandwidth or a channel spacing so wide as to make the benefits of multiplexing uneconomic.

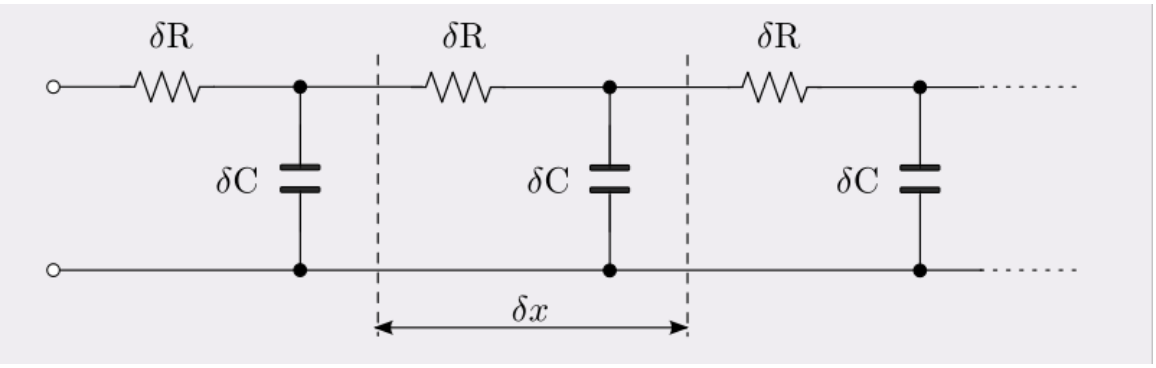
The basic technical reason for this difficulty is that the frequency response of a simple filter approaches a fall of 6 dB/octave far from the point of resonance. This means that if telephone channels are squeezed in side-by-side into the frequency spectrum, there will be crosstalk from adjacent channels in any given channel. What is required is a much more sophisticated filter that has a flat frequency response in the required passband like a low-Q resonant circuit, but that rapidly falls in response (much faster than 6 dB/octave) at the transition from passband to stopband like a high-Q resonant circuit. Obviously, these are contradictory requirements to be met with a single resonant circuit. The solution to these needs was founded in the theory of transmission lines and consequently the necessary filters did not become available until this theory was fully developed. At this early stage the idea of signal bandwidth, and hence the need for filters to match to it, was not fully understood; indeed, it was as late as 1920 before the concept of bandwidth was fully established. For early radio, the concepts of Q-factor, selectivity and tuning sufficed. This was all to change with the developing theory of transmission lines on which image filters are based, as explained in the next section.

At the turn of the century as telephone lines became available, it became popular to add telegraph on to telephone lines with an earth return phantom circuit. An LC filter was required to prevent telegraph clicks being heard on the telephone line. From the 1920s onwards, telephone lines, or balanced lines dedicated to the purpose, were used for FDM telegraph at audio frequencies. The first of these systems in the UK was a Siemens and Halske installation between London and Manchester. GEC and AT&T also had FDM systems. Separate pairs were used for the send and receive signals. The Siemens and GEC systems had six channels of telegraph in each direction, the AT&T system had twelve. All of these systems used electronic oscillators to generate a different carrier for each telegraph signal and required a bank of band-pass filters to separate out the multiplexed signal at the receiving end.

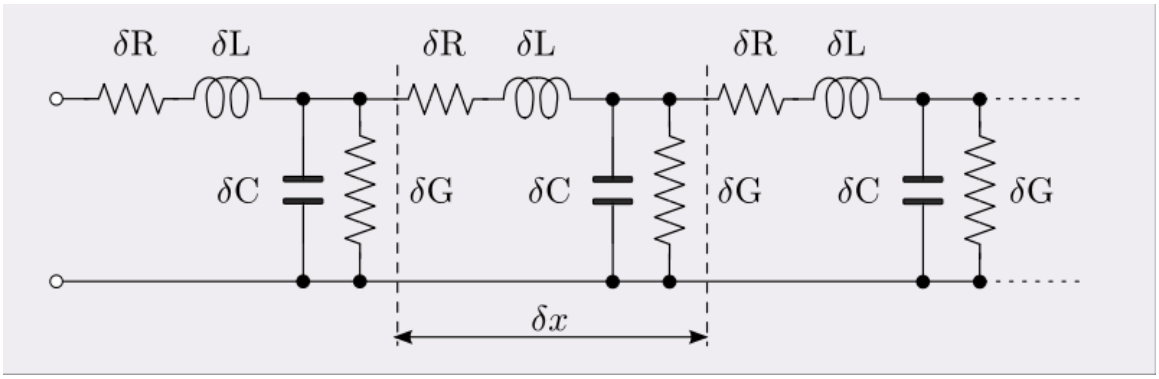
Transmission line theory



Ohm's model of the transmission line was simply resistance.



Lord Kelvin's model of the transmission line accounted for capacitance and the dispersion it caused. The diagram represents Kelvin's model translated into modern terms using infinitesimal elements, but this was not the actual approach used by Kelvin.

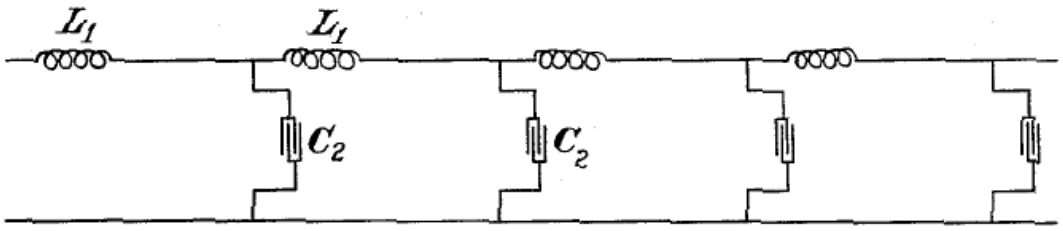


Heaviside's model of the transmission line. L, R, C and G in all three diagrams are the primary line constants. The infinitesimals δL, δR, δC and δG are to be understood as Lδx, Rδx, Cδx and Gδx respectively.

The earliest model of the transmission line was probably described by Georg Ohm (1827) who established that resistance in a wire is proportional to its length. The Ohm model thus included only resistance. Latimer Clark noted that signals were delayed and elongated along a cable, an undesirable form of distortion now called dispersion but then called retardation, and Michael Faraday (1853) established that this was due to the capacitance present in the transmission line. Lord Kelvin (1854) found the correct mathematical description needed in his work on early transatlantic cables; he arrived at an equation identical to the conduction of a heat pulse along a metal bar. This model incorporates only resistance and capacitance, but that is all that was needed in undersea cables dominated by capacitance effects. Kelvin's model predicts a limit on the telegraph signalling speed of a cable but Kelvin still did not use the concept of bandwidth, the limit was entirely explained in terms of the dispersion of the telegraph symbols. The mathematical model of the transmission line reached its fullest development with Oliver Heaviside. Heaviside (1881) introduced series inductance and shunt conductance into the model making four distributed elements in all. This model is now known as the telegrapher's equation and the distributed elements are called the primary line constants.

From the work of Heaviside (1887) it had become clear that the performance of telegraph lines, and most especially telephone lines, could be improved by the addition of inductance to the line. George Campbell at AT&T implemented this idea (1899) by inserting loading coils at intervals along the line. Campbell found that as well as the desired improvements to the line's characteristics in the passband there was also a definite frequency beyond which signals could not be passed without great attenuation. This was a result of the loading coils and the line capacitance forming a low-pass filter, an effect that is only apparent on lines incorporating lumped components such as the loading coils. This naturally led Campbell (1910) to produce a filter with ladder topology, a glance at the circuit diagram of this filter is enough to see its relationship to a loaded transmission line. The cut-off phenomenon is an undesirable side-effect as far as loaded lines are concerned but for telephone FDM filters it is precisely what is required. For this application, Campbell produced band-pass filters to the same ladder topology by replacing the inductors and capacitors with resonators and anti-resonators respectively. Both the loaded line and FDM were of great benefit economically to AT&T and this led to fast development of filtering from this point onwards.

Image filters



Campbell's sketch of the low-pass version of his filter from his 1915 patent showing the now ubiquitous ladder topology with capacitors for the ladder rungs and inductors for the stiles. Filters of more modern design also often adopt the same ladder topology as used by Campbell. It should be understood that although superficially similar, they are really quite different. The ladder construction is essential to the Campbell filter and all the sections have identical element values. Modern designs can be realised in any number of topologies, choosing the ladder topology is merely a matter of convenience. Their response is quite different (better) than Campbell's and the element values, in general, will all be different.

The filters designed by Campbell were named wave filters because of their property of passing some waves and strongly rejecting others. The method by which they were designed was called the image parameter method and filters designed to this method are called image filters. The image method essentially consists of developing the transmission constants of an infinite chain of identical filter sections and then terminating the desired finite number of filter sections in the image impedance. This exactly corresponds to the way the properties of a finite length of transmission line are derived

from the theoretical properties of an infinite line, the image impedance corresponding to the characteristic impedance of the line.

From 1920 John Carson, also working for AT&T, began to develop a new way of looking at signals using the operational calculus of Heaviside which in essence is working in the frequency domain. This gave the AT&T engineers a new insight into the way their filters were working and led Otto Zobel to invent many improved forms. Carson and Zobel steadily demolished many of the old ideas. For instance the old telegraph engineers thought of the signal as being a single frequency and this idea persisted into the age of radio with some still believing that frequency modulation (FM) transmission could be achieved with a smaller bandwidth than the baseband signal right up until the publication of Carson's 1922 paper. Another advance concerned the nature of noise, Carson and Zobel (1923) treated noise as a random process with a continuous bandwidth, an idea that was well ahead of its time, and thus limited the amount of noise that it was possible to remove by filtering to that part of the noise spectrum which fell outside the passband. This too, was not generally accepted at first, notably being opposed by Edwin Armstrong (who ironically, actually succeeded in reducing noise with wide-band FM) and was only finally settled with the work of Harry Nyquist whose thermal noise power formula is well known today.

Several improvements were made to image filters and their theory of operation by Otto Zobel. Zobel coined the term constant k filter (or k-type filter) to distinguish Campbell's filter from later types, notably Zobel's m-derived filter (or m-type filter). The particular problems Zobel was trying to address with these new forms were impedance matching into the end terminations and improved steepness of roll-off. These were achieved at the cost of an increase in filter circuit complexity.

A more systematic method of producing image filters was introduced by Hendrik Bode (1930), and further developed by several other investigators including Piloty (1937-1939) and Wilhelm Cauer (1934-1937). Rather than enumerate the behaviour (transfer function, attenuation function, delay function and so on) of a specific circuit, instead a requirement for the image impedance itself was developed. The image impedance can be expressed in terms of the open-circuit and short-circuit impedances of the filter as $Z_i=\sqrt{Z_oZ_s}$. Since the image impedance must be real in the passbands and imaginary in the stopbands according to image theory, there is a requirement that the poles and zeroes of Z_o and Z_s cancel in the passband and correspond in the stopband. The behaviour of the filter can be entirely defined in terms of the positions in the complex plane of these pairs of poles and zeroes. Any circuit which has the requisite poles and zeroes will also have the requisite response. Cauer pursued two related questions arising from this technique: what specification of poles and zeroes are realisable as passive filters; and what realisations are equivalent to each other. The results of this work led Cauer to develop a new approach, now called network synthesis.

This "poles and zeroes" view of filter design was particularly useful where a bank of filters, each operating at different frequencies, are all connected across the same transmission line. The earlier approach was unable to deal properly with this situation,

but the poles and zeroes approach could embrace it by specifying a constant impedance for the combined filter. This problem was originally related to FDM telephony but frequently now arises in loudspeaker crossover filters.

Network synthesis filters

The essence of network synthesis is to start with a required filter response and produce a network that delivers that response, or approximates to it within a specified boundary. This is the inverse of network analysis which starts with a given network and by applying the various electric circuit theorems predicts the response of the network. The term was first used with this meaning in the doctoral thesis of Yuk-Wing Lee (1930) and apparently arose out of a conversation with Vannevar Bush. The advantage of network synthesis over previous methods is that it provides a solution which precisely meets the design specification. This is not the case with image filters, a degree of experience is required in their design since the image filter only meets the design specification in the unrealistic case of being terminated in its own image impedance, to produce which would require the exact circuit being sought. Network synthesis on the other hand, takes care of the termination impedances simply by incorporating them into the network being designed.

The development of network analysis needed to take place before network synthesis was possible. The theorems of Gustav Kirchhoff and others and the ideas of Charles Steinmetz (phasors) and Arthur Kennelly (complex impedance) laid the groundwork. The concept of a port also played a part in the development of the theory, and proved to be a more useful idea than network terminals. The first milestone on the way to network synthesis was an important paper by Ronald Foster (1924), *A Reactance Theorem*, in which Foster introduces the idea of a driving point impedance, that is, the impedance that is connected to the generator. The expression for this impedance determines the response of the filter and vice versa, and a realisation of the filter can be obtained by expansion of this expression. It is not possible to realise any arbitrary impedance expression as a network. Foster's reactance theorem stipulates necessary and sufficient conditions for realisability: that the reactance must be algebraically increasing with frequency and the poles and zeroes must alternate.

Wilhelm Cauer expanded on the work of Foster (1926) and was the first to talk of realisation of a one-port impedance with a prescribed frequency function. Foster's work considered only reactances (i.e., only LC-kind circuits). Cauer generalised this to any 2-element kind one-port network, finding there was an isomorphism between them. He also found ladder realisations of the network using Thomas Stieltjes' continued fraction expansion. This work was the basis on which network synthesis was built, although Cauer's work was not at first used much by engineers, partly because of the intervention of World War II, partly for reasons explained in the next section and partly because Cauer presented his results using topologies that required mutually coupled inductors and ideal transformers. Although on this last point, it has to be said that transformer coupled double tuned amplifiers are a common enough way of widening bandwidth without sacrificing selectivity.

Image method versus synthesis

Image filters continued to be used by designers long after the superior network synthesis techniques were available. Part of the reason for this may have been simply inertia, but it was largely due to the greater computation required for network synthesis filters, often needing a mathematical iterative process. Image filters, in their simplest form, consist of a chain of repeated, identical sections. The design can be improved simply by adding more sections and the computation required to produce the initial section is on the level of "back of an envelope" designing. In the case of network synthesis filters, on the other hand, the filter is designed as a whole, single entity and to add more sections (i.e., increase the order) the designer would have no option but to go back to the beginning and start over. The advantages of synthesised designs are real, but they are not overwhelming compared to what a skilled image designer could achieve, and in many cases it was more cost effective to dispense with time-consuming calculations. This is simply not an issue with the modern availability of computing power, but in the 1950s it was non-existent, in the 1960s and 1970s available only at cost, and not finally becoming widely available to all designers until the 1980s with the advent of the desktop personal computer. Image filters continued to be designed up to that point and many remained in service into the 21st century.

The computational difficulty of the network synthesis method was addressed by tabulating the component values of a prototype filter and then scaling the frequency and impedance and transforming the bandform to those actually required. This kind of approach, or similar, was already in use with image filters, for instance by Zobel, but the concept of a "reference filter" is due to Sidney Darlington. Darlington (1939), was also the first to tabulate values for network synthesis prototype filters, nevertheless it had to wait until the 1950s before the Cauer-Darlington elliptic filter first came into use.

Once computational power was readily available, it became possible to easily design filters to minimise any arbitrary parameter, for example time delay or tolerance to component variation. The difficulties of the image method were firmly put in the past, and even the need for prototypes became largely superfluous. Furthermore, the advent of active filters eased the computation difficulty because sections could be isolated and iterative processes were not then generally necessary.

Realisability and equivalence

Realisability (that is, which functions are realisable as real impedance networks) and equivalence (which networks equivalently have the same function) are two important questions in network synthesis. Following an analogy with Lagrangian mechanics, Cauer formed the matrix equation,

$$[\mathbf{A}] = s^2[\mathbf{L}] + s[\mathbf{R}] + [\mathbf{D}] = s[\mathbf{Z}]$$

where [**Z**],[**R**],[**L**] and [**D**] are the *n*x*n* matrices of, respectively, impedance, resistance, inductance and elastance of an *n*-mesh network and *s* is the complex frequency operator

$s=\sigma+i\omega$. Here [**R**],[**L**] and [**D**] have associated energies corresponding to the kinetic, potential and dissipative heat energies, respectively, in a mechanical system and the already known results from mechanics could be applied here. Cauer determined the driving point impedance by the method of Lagrange multipliers;

$$Z_{\mathrm{p}}(s) = \frac{\det[\mathbf{A}]}{s\, a_{11}}$$

where a_{11} is the complement of the element A_{11} to which the one-port is to be connected. From stability theory Cauer found that [**R**], [**L**] and [**D**] must all be positive-definite matrices for $Z_{\mathrm{p}}(s)$ to be realisable if ideal transformers are not excluded. Realisability is only otherwise restricted by practical limitations on topology. This work is also partly due to Otto Brune (1931), who worked with Cauer in the US prior to Cauer returning to Germany. A well known condition for realisability of a one-port rationalimpedance due to Cauer (1929) is that it must be a function of s that is analytic in the right halfplane ($\sigma>0$), have a positive real part in the right halfplane and take on real values on the real axis. This follows from the Poisson integral representation of these functions. Brune coined the term positive-real for this class of function and proved that it was a necessary and sufficient condition (Cauer had only proved it to be necessary) and they extended the work to LC multiports. A theorem due to Sidney Darlington states that any positive-real function $Z(s)$ can be realised as a lossless two-port terminated in a positive resistor R. No resistors within the network are necessary to realise the specified response.

As for equivalence, Cauer found that the group of real affine transformations,

$$[\mathbf{T}]^T[\mathbf{A}][\mathbf{T}]$$

where,

$$[\mathbf{T}] = \begin{bmatrix} 1 & 0\cdots 0 \\ T_{21} & T_{22}\cdots T_{2n} \\ \cdot & \cdots \\ T_{n1} & T_{n2}\cdots T_{nn} \end{bmatrix}$$

is invariant in $Z_{\mathrm{p}}(s)$, that is, all the transformed networks are equivalents of the original.

Approximation

The approximation problem in network synthesis is to find functions which will produce realisable networks approximating to a prescribed function of frequency within limits arbitrarily set. The approximation problem is an important issue since the ideal function of frequency required will commonly be unachievable with rational networks. For instance, the ideal prescribed function is often taken to be the unachievable lossless transmission in the passband, infinite attenuation in the stopband and a vertical transition between the two. However, the ideal function can be approximated with a rational function, becoming ever closer to the ideal the higher the order of the polynomial. The

first to address this problem was Stephen Butterworth (1930) using his Butterworth polynomials. Independently, Cauer (1931) used Chebyshev polynomials, initially applied to image filters, and not to the now well-known ladder realisation of this filter.

Butterworth filter

Butterworth filters are an important classof filters due to Stephen Butterworth (1930) which are now recognised as being a special case of Cauer's elliptic filters. Butterworth discovered this filter independently of Cauer's work and implemented it in his version with each section isolated from the next with a valve amplifier which made calculation of component values easy since the filter sections could not interact with each other and each section represented one term in the Butterworth polynomials. This gives Butterworth the credit for being both the first to deviate from image parameter theory and the first to design active filters. It was later shown that Butterworth filters could be implemented in ladder topology without the need for amplifiers, possibly the first to do so was William Bennett (1932) in a patent which presents formulae for component values identical to the modern ones. Bennett, at this stage though, is still discussing the design as an artificial transmission line and so is adopting an image parameter approach despite having produced what would now be considered a network synthesis design. He also does not appear to be aware of the work of Butterworth or the connection between them.

Insertion-loss method

The insertion-loss method of designing filters is, in essence, to prescribe a desired function of frequency for the filter as an attenuation of the signal when the filter is inserted between the terminations relative to the level that would have been received were the terminations connected to each other via an ideal transformer perfectly matching them. Versions of this theory are due to Sidney Darlington, Wilhelm Cauer and others all working more or less independently and is often taken as synonymous with network synthesis. Butterworth's filter implementation is, in those terms, an insertion-loss filter, but it is a relatively trivial one mathematically since the active amplifiers used by Butterworth ensured that each stage individually worked into a resistive load. Butterworth's filter becomes a non-trivial example when it is implemented entirely with passive components. An even earlier filter which influenced the insertion-loss method was Norton's dual-band filter where the input of two filters are connected in parallel and designed so that the combined input presents a constant resistance. Norton's design method, together with Cauer's canonical LC networks and Darlington's theorem that only LC components were required in the body of the filter resulted in the insertion-loss method. However, ladder topology proved to be more practical than Cauer's canonical forms.

Darlington's insertion-loss method is a generalisation of the procedure used by Norton. In Norton's filter it can be shown that each filter is equivalent to a separate filter unterminated at the common end. Darlington's method applies to the more straightforward and general case of a 2-port LC network terminated at both ends. The procedure consists of the following steps:

1. determine the poles of the prescribed insertion-loss function,
2. from that find the complex transmission function,
3. from that find the complex reflection coefficients at the terminating resistors,
4. find the driving point impedance from the short-circuit and open-circuit impedances,
5. expand the driving point impedance into an LC (usually ladder) network.

Darlington additionally used a transformation found by Hendrik Bode that predicted the response of a filter using non-ideal components but all with the same *Q*. Darlington used this transformation in reverse to produce filters with a prescribed insertion-loss with non-ideal components. Such filters have the ideal insertion-loss response plus a flat attenuation across all frequencies.

Elliptic filters

Elliptic filters are filters produced by the insertion-loss method which use elliptic rational functions in their transfer function as an approximation to the ideal filter response and the result is called a Chebyshev approximation. This is the same Chebyshev approximation technique used by Cauer on image filters but follows the Darlington insertion-loss design method and uses slightly different elliptic functions. Cauer had some contact with Darlington and Bell Labs before WWII (for a time he worked in the US) but during the war they worked independently, in some cases making the same discoveries. Cauer had disclosed the Chebyshev approximation to Bell Labs but had not left them with the proof. Sergei Schelkunoff provided this and a generalisation to all equal ripple problems. Elliptic filters are a general class of filter which incorporate several other important classes as special cases: Cauer filter (equal ripple in passband and stopband), Chebyshev filter (ripple only in passband), reverse Chebyshev filter (ripple only in stopband) and Butterworth filter (no ripple in either band).

Generally, for insertion-loss filters where the transmission zeroes and infinite losses are all on the real axis of the complex frequency plane (which they usually are for minimum component count), the insertion-loss function can be written as;

$$\frac{1}{1 + JF^2}$$

where *F* is either an even (resulting in an antimetric filter) or an odd (resulting in an symmetric filter) function of frequency. Zeroes of *F* correspond to zero loss and the poles of *F* correspond to transmission zeroes. *J* sets the passband ripple height and the stopband loss and these two design requirements can be interchanged. The zeroes and poles of *F* and *J* can be set arbitrarily. The nature of *F* determines the class of the filter;

- if *F* is a Chebyshev approximation the result is a Chebyshev filter,
- if *F* is a maximally flat approximation the result is a passband maximally flat filter,
- if 1/*F* is a Chebyshev approximation the result is a reverse Chebyshev filter,

- if $1/F$ is a maximally flat approximation the result is a stopband maximally flat filter,

A Chebyshev response simultaneously in the passband and stopband is possible, such as Cauer's equal ripple elliptic filter.

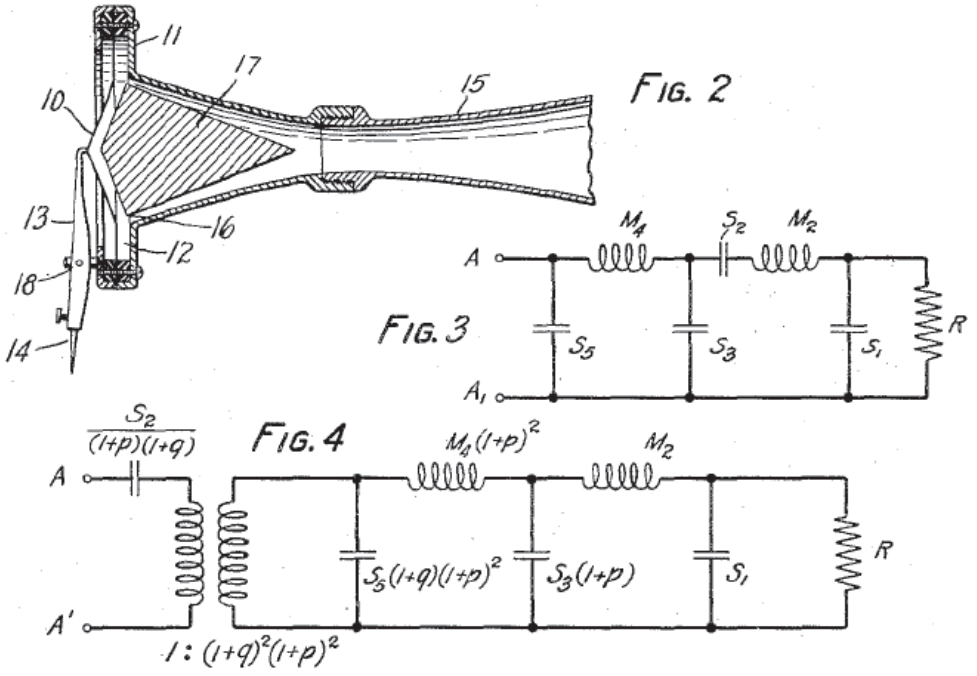
Darlington relates that he found in the New York City library Carl Jacobi's original paper on elliptic functions, published in Latin in 1829. In this paper Darlington was surprised to find foldout tables of the exact elliptic function transformations needed for Chebyshev approximations of both Cauer's image parameter, and Darlington's insertion-loss filters.

Other methods

Darlington considers the topology of coupled tuned circuits to involve a separate approximation technique to the insertion-loss method, but also producing nominally flat passbands and high attenuation stopbands. The most common topology for these is shunt anti-resonators coupled by series capacitors, less commonly, by inductors, or in the case of a two-section filter, by mutual inductance. These are most useful where the design requirement is not too stringent, that is, moderate bandwidth, roll-off and passband ripple.

Other notable developments and applications

Mechanical filters



Norton's mechanical filter together with its electrical equivalent circuit. Two equivalents are shown, "Fig.3" directly corresponds to the physical relationship of the mechanical

components; "Fig.4" is an equivalent transformed circuit arrived at by repeated application of a well known transform, the purpose being to remove the series resonant circuit from the body of the filter leaving a simple *LC* ladder network.

Edward Norton, around 1930, designed a mechanical filter for use on phonograph recorders and players. Norton designed the filter in the electrical domain and then used the correspondence of mechanical quantities to electrical quantities to realise the filter using mechanical components. Mass corresponds to inductance, stiffness to elastance and damping to resistance. The filter was designed to have a maximally flat frequency response.

In modern designs it is common to use quartz crystal filters, especially for narrowband filtering applications. The signal exists as a mechanical acoustic wave while it is in the crystal and is converted by transducers between the electrical and mechanical domains at the terminals of the crystal.

Transversal filters

Transversal filters are not usually associated with passive implementations but the concept can be found in a Wiener and Lee patent from 1935 which describes a filter consisting of a cascade of all-pass sections. The outputs of the various sections are summed in the proportions needed to result in the required frequency function. This works by the principle that certain frequencies will be in, or close to antiphase, at different sections and will tend to cancel when added. These are the frequencies rejected by the filter and can produce filters with very sharp cut-offs. This approach did not find any immediate applications, and is not common in passive filters. However, the principle finds many applications as an active delay line implementation for wide band discrete-time filter applications such as television, radar and high-speed data transmission.

Matched filter

The purpose of matched filters is to maximise the signal-to-noise ratio (S/N) at the expense of pulse shape. Pulse shape, unlike many other applications, is unimportant in radar while S/N is the primary limitation on performance. The filters were introduced during WWII (described 1943) by Dwight North and are often eponymously referred to as "North filters".

Filters for control systems

Control systems have a need for smoothing filters in their feedback loops with criteria to maximise the speed of movement of a mechanical system to the prescribed mark and at the same time minimise overshoot and noise induced motions. A key problem here is the extraction of Gaussian signals from a noisy background. An early paper on this was published during WWII by Norbert Wiener with the specific application to anti-aircraft fire control analogue computers. Rudy Kalman (Kalman filter) later reformulated this in terms of state-space smoothing and prediction where it is known as the linear-quadratic-

Gaussian control problem. Kalman started an interest in state-space solutions, but according to Darlington this approach can also be found in the work of Heaviside and earlier.

Modern practice

LC passive filters gradually became less popular as active amplifying elements, particularly operational amplifiers, became cheaply available. The reason for the change is that wound components (the usual method of manufacture for inductors) are far from ideal, the wire adding resistance as well as inductance to the component. Inductors are also relatively expensive and are not "off-the-shelf" components. On the other hand, the function of LC ladder sections, LC resonators and RL sections can be replaced by RC components in an amplifier feedback loop (active filters). These components will usually be much more cost effective, and smaller as well. Cheap digital technology, in its turn, has largely supplanted analogue implementations of filters. However, there is still an occasional place for them in the simpler applications such as coupling where sophisticated functions of frequency are not needed.

Chapter-7

Impulse Invariance and Infinite Impulse Response

Impulse invariance

Impulse invariance is a technique for designing discrete-time infinite-impulse-response (IIR) filters from continuous-time filters in which the impulse response of the continuous-time system is sampled to produce the impulse response of the discrete-time system. The frequency response of the discrete-time system will be a sum of shifted copies of the frequency response of the continuous-time system; if the continuous-time system is approximately band-limited to a frequency less than the Nyquist frequency of the sampling, then the frequency response of the discrete-time system will be approximately equal to it for frequencies below the Nyquist frequency.

Discussion

The continuous-time system's impulse response, $h_c(t)$, is sampled with sampling period T to produce the discrete-time system's impulse response, $h[n]$.

$$h[n] = Th_c(nT)$$

Thus, the frequency responses of the two systems are related by

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c\left(j\frac{\omega}{T} + j\frac{2\pi}{T}k\right)$$

If the continuous time filter is approximately band-limited (i.e. $H_c(j\Omega) < \delta$ when $|\Omega| \geq \pi/T$), then the frequency response of the discrete-time system will be approximately the continuous-time system's frequency response for frequencies below π radians per sample (below the Nyquist frequency $1/(2T)$ Hz):

$$H(e^{j\omega}) = H_c(j\omega/T) \text{for } |\omega| \leq \pi$$

Comparison to the bilinear transform

Note that aliasing will occur, including aliasing below the Nyquist frequency to the extent that the continuous-time filter's response is nonzero above that frequency. The bilinear transform is an alternative to impulse invariance that uses a different mapping that maps the continuous-time system's frequency response, out to infinite frequency, into the range of frequencies up to the Nyquist frequency in the discrete-time case, as opposed to mapping frequencies linearly with circular overlap as impulse invariance does.

Effect on poles in system function

If the continuous poles at $s = s_k$, the system function can be written in partial fraction expansion as

$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k}$$

Thus, using the inverse Laplace transform, the impulse response is

$$h_c(t) = \begin{cases} \sum_{k=1}^{N} A_k e^{s_k t}, & t \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

The corresponding discrete-time system's impulse response is then defined as the following

$$h[n] = T h_c(nT)$$
$$h[n] = T \sum_{k=1}^{N} A_k e^{s_k nT} u[n]$$

Performing a z-transform on the discrete-time impulse response produces the following discrete-time system function

$$H(z) = T \sum_{k=1}^{N} \frac{A_k}{1 - e^{s_k T} z^{-1}}$$

Thus the poles from the continuous-time system function are translated to poles at $z = e^{s_k T}$. The zeros, if any, are not so simply mapped.

Poles and zeros

If the system function has zeros as well as poles, they can be mapped the same way, but the result is no longer an impulse invariance result: the discrete-time impulse response is

not equal simply to samples of the continuous-time impulse response. This method is known as the matched Z-transform method, or pole–zero mapping. In the case of all-pole filters, the methods are equivalent.

Stability and causality

Since poles in the continuous-time system at $s = s_k$ transform to poles in the discrete-time system at $z = \exp(s_k T)$, poles in the left half of the s-plane map to inside the unit circle in the z-plane; so if the continuous-time filter is causal and stable, then the discrete-time filter will be causal and stable as well.

Corrected formula

When a causal continuous-time impulse response has a discontinuity at $t = 0$, the expressions above are not consistent. This is because $h_c(0)$ should really only contribute half its value to $h[0]$.

Making this correction gives

$$h[n] = T\left(h_c(nT) - \frac{1}{2}h_c(0)\delta[n]\right)$$
$$h[n] = T\sum_{k=1}^{N} A_k e^{s_k nT}\left(u[n] - \frac{1}{2}\delta[n]\right)$$

Performing a z-transform on the discrete-time impulse response produces the following discrete-time system function

$$H(z) = T\sum_{k=1}^{N}\frac{A_k}{1 - e^{s_k T}z^{-1}} - \frac{T}{2}\sum_{k=1}^{N} A_k.$$

Infinite impulse response

Infinite impulse response (IIR) is a property of signal processing systems. Systems with this property are known as *IIR systems* or, when dealing with filter systems, as *IIR filters*. IIR systems have an impulse response function that is non-zero over an infinite length of time. This is in contrast to finite impulse response (FIR) filters, which have fixed-duration impulse responses. The simplest analog IIR filter is an RC filter made up of a single resistor (R) feeding into a node shared with a single capacitor (C). This filter has an exponential impulse response characterized by an RC time constant.

IIR filters may be implemented as either analog or digital filters. In digital IIR filters, the output feedback is immediately apparent in the equations defining the output. Note that

unlike FIR filters, in designing IIR filters it is necessary to carefully consider the "time zero" case in which the outputs of the filter have not yet been clearly defined.

Design of digital IIR filters is heavily dependent on that of their analog counterparts because there are plenty of resources, works and straightforward design methods concerning analog feedback filter design while there are hardly any for digital IIR filters. As a result, usually, when a digital IIR filter is going to be implemented, an analog filter (e.g. Chebyshev filter, Butterworth filter, Elliptic filter) is first designed and then is converted to a digital filter by applying discretization techniques such as Bilinear transform or Impulse invariance.

Example IIR filters include the Chebyshev filter, Butterworth filter, and the Bessel filter.

Transfer function derivation

Digital filters are often described and implemented in terms of the difference equation that defines how the output signal is related to the input signal:

$$\begin{aligned} y[n] = \frac{1}{a_0}(b_0x[n] + b_1x[n-1] + \cdots + b_Px[n-P] \\ - a_1y[n-1] - a_2y[n-2] - \cdots - a_Qy[n-Q]) \end{aligned}$$

where:

- P is the feedforward filter order
- b_i are the feedforward filter coefficients
- Q is the feedback filter order
- a_i are the feedback filter coefficients
- $x[n]$ is the input signal
- $y[n]$ is the output signal.

A more condensed form of the difference equation is:

$$y[n] = \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j]\right)$$

which, when rearranged, becomes:

$$\sum_{j=0}^{Q} a_j y[n-j] = \sum_{i=0}^{P} b_i x[n-i]$$

To find the transfer function of the filter, we first take the Z-transform of each side of the above equation, where we use the time-shift property to obtain:

$$\sum_{j=0}^{Q} a_j z^{-j} Y(z) = \sum_{i=0}^{P} b_i z^{-i} X(z)$$

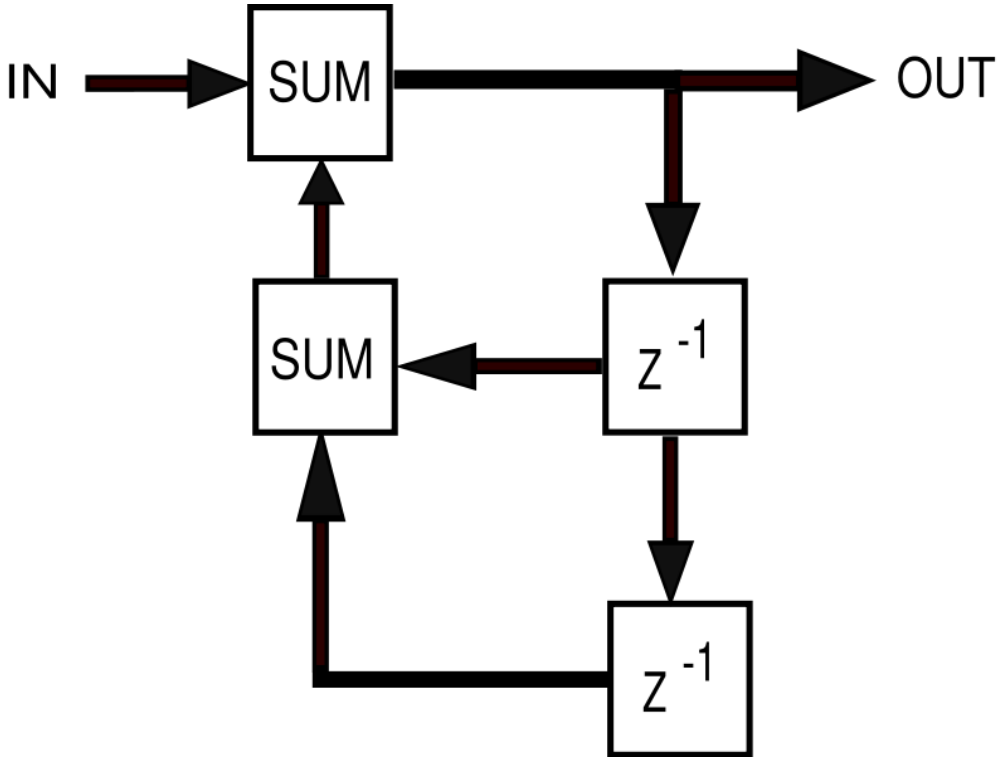
We define the transfer function to be:

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{\sum_{i=0}^{P} b_i z^{-i}}{\sum_{j=0}^{Q} a_j z^{-j}} \end{aligned}$$

Considering that in most IIR filter designs coefficient a_0 is 1, the IIR filter transfer function takes the more traditional form:

$$H(z) = \frac{\sum_{i=0}^{P} b_i z^{-i}}{1 + \sum_{j=1}^{Q} a_j z^{-j}}$$

Description of block diagram



Simple IIR filter block diagram

A typical block diagram of an IIR filter looks like the following. The z^{-1} block is a unit delay. The coefficients and number of feedback/feedforward paths are implementation-dependent.

Stability

The transfer function allows us to judge whether or not a system is bounded-input, bounded-output (BIBO) stable. To be specific, the BIBO stability criteria requires that the ROC of the system includes the unit circle. For example, for a causal system, all poles of the transfer function have to have an absolute value smaller than one. In other words, all poles must be located within a unit circle in the *z*-plane.

The poles are defined as the values of *z* which make the denominator of *H*(*z*) equal to 0:

$$0 = \sum_{j=0}^{Q} a_j z^{-j}$$

Clearly, if $a_j \neq 0$ then the poles are not located at the origin of the z-plane. This is in contrast to the FIR filter where all poles are located at the origin, and is therefore always stable.

IIR filters are sometimes preferred over FIR filters because an IIR filter can achieve a much sharper transition region roll-off than FIR filter of the same order.

Example

Let the transfer function of a filter *H* be

$$H(z) = \frac{B(z)}{A(z)} = \frac{1}{1 - az^{-1}}$$ with ROC $a < |z|$ and $0 < a < 1$

which has a pole at *a*, is stable and causal. The time-domain impulse response is

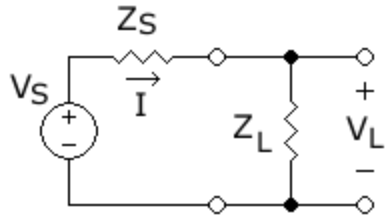
$$h(n) = a^n u(n)$$

which is non-zero for $n >= 0$.

Chapter-8

Impedance Matching

In electronics, **impedance matching** is the practice of designing the input impedance of an electrical load or the output impedance of its corresponding signal source in order to maximize the power transfer and minimize reflections from the load.



In the case of a complex source impedance Z_S and load impedance Z_L, matching is obtained when

$$Z_S = Z_L^*$$

where * indicates the complex conjugate.

The concept of impedance matching was originally developed for electrical power, but can be applied to any other field where a form of energy (not necessarily electrical) is transferred between a source and a load.

An alternative to impedance matching is impedance bridging, where the load impedance is chosen to be much larger than the source impedance and maximizing voltage transfer, rather than power, is the goal.

Explanation

The term impedance is used for the resistance of a system to an energy source. For constant signals, this resistance can also be constant. For varying signals, it usually changes with frequency. The energy involved can be electrical, mechanical, magnetic or

even thermal. The concept of electrical impedance is perhaps the most commonly known. Electrical impedance, like electrical resistance, is measured in ohms. In general, impedance has a complex value, which means that loads generally have a resistance to the source that is in phase with a sinusoidal source signal **and** reactance that is out of phase with a sinusoidal source signal. The total impedance (symbol: Z) is the vector sum of the resistance (symbol: R; a real number) and the reactance (symbol: X; an imaginary number).

In simple cases, such as low-frequency or direct-current power transmission, the reactance is negligible or zero and the impedance can be considered a pure resistance, expressed as a real number. In the following summary, we will consider the general case when the resistance and reactance are both significant, and also the special case in which the reactance is negligible.

Reflectionless or broadband matching

Impedance matching to minimize reflections and maximize power transfer over a (relatively) large bandwidth (also called **reflectionless matching** or **broadband matching**) is the most commonly used. To prevent all reflections of the signal back into the source, the load (which must be totally resistive) must be matched exactly to the source impedance (which again must be totally resistive). In this case, if a transmission line is used to connect the source and load together, it must also be the same impedance: $Z_{\text{load}} = Z_{\text{line}} = Z_{\text{source}}$, where Z_{line} is the characteristic impedance of the transmission line. Although source and load should each be totally resistive for this form of matching to work, the more general term 'impedance' is still used to describe the source and load characteristics. Any and all reactance actually present in the source or the load will affect the 'match'.

Complex conjugate matching

This is used in cases in which the source and load are reactive. This form of impedance matching can only maximize the power transfer between a reactive source and a reactive load at a *single* frequency. In this case,

$$Z_{\text{load}} = Z_{\text{source}}^{*}$$

(where * indicates the complex conjugate).

If the signals are kept within the narrow frequency range for which the matching network was designed, reflections (in this narrow frequency band only) are also minimized. For the case of purely resistive source and load impedances, all reactance terms are zero and the formula above reduces to

$$Z_{\text{load}} = Z_{\text{source}}$$

as would be expected.

Power transfer

Whenever a source of power *with a fixed output impedance*, such as an electric signal source, a radio transmitter, or even mechanical sound (e.g., a loudspeaker) operates into a load, the maximum possible power is delivered to the load when the impedance of the load (load impedance or input impedance) is equal to the *complex conjugate* of the impedance of the source (that is, its internal impedance or output impedance). For two impedances to be complex conjugates, their resistances must be equal, and their reactances must be equal in magnitude but of opposite sign.

In low-frequency or DC systems, or in systems with purely resistive sources and loads, the reactances are zero, or small enough to be ignored. In this case, maximum power transfer occurs when the resistance of the load is equal to the resistance of the source.

Impedance matching is not always desirable. For example, if a source with a low impedance is connected to a load with a high impedance, then the power that can pass through the connection is limited by the higher impedance, but the electrical voltage transfer is higher and less prone to corruption than if the impedances had been matched. This maximum voltage connection is a common configuration called **impedance bridging** or **voltage bridging** and is widely used in signal processing. In such applications, delivering a high voltage (to minimize signal degradation during transmission and/or to consume less power by reducing currents) is often more important than maximum power transfer.

In older audio systems, reliant on transformers and passive filter networks, and based on the telephone system, the source and load resistances were matched at 600 ohms. One reason for this was to maximize power transfer, as there were no amplifiers available that could restore lost signal. Another reason was to ensure correct operation of the hybrid transformers used at central exchange equipment to separate outgoing from incoming speech so that these could be amplified or fed to a four-wire circuit. Most modern audio circuits, on the other hand, use active amplification and filtering, and they can use voltage bridging connections for best accuracy.

Strictly speaking, impedance matching only applies when both source and load devices are linear, however useful matching may be obtained between nonlinear devices with certain operating ranges.

Impedance matching devices

Adjusting the source impedance or the load impedance, in general, is called "impedance matching".

There are three possible ways to improve an impedance mismatch, all of which are called "impedance matching":

- devices intended to present an apparent load to the source of $R_{load} = R_{source}*$ (complex conjugate matching). Given a source with a fixed voltage and fixed source impedance, the maximum power theorem says this is the only way to extract the maximum power from the source.
- devices intended to present an apparent load of $R_{load} = R_{line}$ (complex impedance matching), to avoid echoes. Given a transmission line source with a fixed source impedance, this "reflectionless impedance matching" at the end of the transmission line is the only way to avoid reflecting echoes back to the transmission line.
- devices intended to present an apparent source resistance as close to zero as possible, or presenting an apparent source voltage as high as possible. This is the only way to maximize energy efficiency, and so it is used at the beginning of electrical power lines. Such an impedance bridging connection also minimizes distortion and electromagnetic interference, and so it is also used in modern audio amplifiers and signal processing devices.

There are a variety of devices that are used between some source of energy and some load that perform "impedance matching".

To match electrical impedances, engineers use combinations of transformers, resistors, inductors, capacitors and transmission lines.

These passive and active impedance matching devices are optimized for different applications, and are called baluns, antenna tuners (sometimes called ATUs or roller coasters because of their appearance), acoustic horns, matching networks, and terminators.

Transformers

Transformers are sometimes used to match the impedances of circuits with different impedances. A transformer converts alternating current at one voltage to the same waveform at another voltage. The power input to the transformer and output from the transformer is the same (except for conversion losses). The side with the lower voltage is at low impedance, because this has the lower number of turns, and the side with the higher voltage is at a higher impedance as it has more turns in its coil.

Resistive network

Resistive impedance matches are easiest to design and can be achieved with a simple L pad consisting of only two resistors. Power loss is an unavoidable consequence of using resistive networks and they are consequently only usually used to transfer line level signals.

Stepped transmission line

Most lumped element devices can match a specific range of load impedance. For example, in order to match an inductive load into a real impedance, a capacitor needs be used. And if the load impedance becomes capacitive for some reason, the matching element must be replaced by an inductor. In many practical cases however, there is a need to use the same circuit to match a broad range of load impedance, thus simplify the circuit design. This issue was addressed by the stepped transmission line where multiple serially placed quarter wave dielectric slugs are used to vary transmission line's characteristic impedance. By controlling the position of each individual element, a broad range of load impedance can be matched without having to reconnect the circuit.

Some special situations - such as radio tuners and transmitters - use tuned filters such as stubs to match impedances at specific frequencies. These can distribute different frequencies to different places in the circuit.

In addition, there is the closely related idea of

- power factor correction devices intended to cancel out the reactive and nonlinear characteristics of a load at the end of a power line. This causes the load seen by the power line to be purely resistive. For a given true power required by a load, this minimizes the true current supplied through the power lines, and so minimizes the power wasted in the resistance of those power lines.

For example, a maximum power point tracker is used to extract the maximum power from a solar panel, and efficiently transfer it to batteries, the power grid, or other loads. The maximum power theorem applies to its "upstream" connection to the solar panel, so it emulates a load resistance equal to the solar panel source resistance. However, the maximum power theorem does not apply to its "downstream" connection, so that connection is an impedance bridging connection—it emulates a high-voltage, low-resistance source, to maximize efficiency.

L-section

One simple electrical impedance matching network requires one capacitor and one inductor. One reactance is in parallel with the source (or load) and the other is in series with the load (or source). If a reactance is in parallel *with the source*, the effective network matches from high impedance to low impedance. The L-section is inherently a narrowband matching network.

The analysis is as follows. Consider a real source impedance of R_1 and real load impedance of R_2. If a reactance X_1 is in parallel with the source impedance, the combined impedance can be written as:

$$\frac{jR_1X_1}{R_1 + jX_1}$$

If the imaginary part of the above impedance is completely canceled by the series reactance, the real part is

$$R_2 = \frac{R_1 X_1^2}{R_1^2 + X_1^2}$$

Solving for X_1

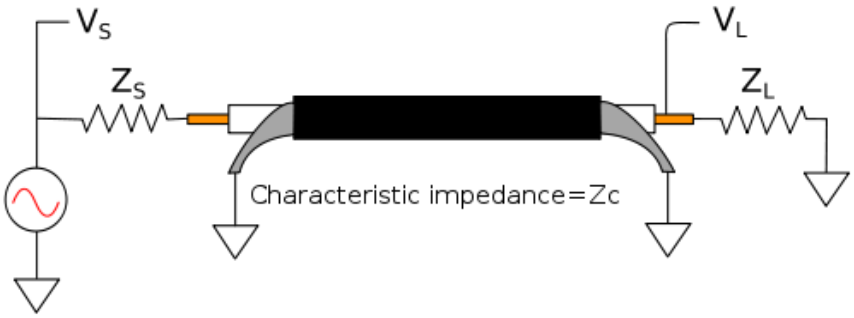
$$X_1 = \sqrt{\frac{R_2 R_1^2}{R_1 - R_2}}$$

If $R_1 \gg R_2$ the above equation can be approximated as

$$X_1 \approx \sqrt{R_1 R_2}$$

The inverse connection, impedance step up, is simply the reverse, e.g. reactance in series with the source. The magnitude of the impedance ratio is limited by reactance losses such as the Q of the inductor. Multiple L-sections can be wired in cascade to achieve higher impedance ratios or greater bandwidth. Transmission line matching networks can be modeled as infinitely many L-sections wired in cascade. Optimal matching circuits can be designed for a particular system with the use of the Smith chart.

Transmission lines



Coaxial transmission line with one source and one load.

Impedance bridging is unsuitable for RF connections because it causes power to be reflected back to the source from the boundary between the high impedance and the low impedance. The reflection creates a standing wave if there is a reflection at both ends of the transmission line, which leads to further power waste and may cause frequency dependent loss. In these systems, impedance matching is desirable.

In electrical systems involving transmission lines, such as radio and fiber optics, where the length of the line is long compared to the wavelength of the signal (the signal changes rapidly compared to the time it takes to travel from source to load), the impedances at each end of the line must be matched to the transmission line's characteristic impedance, Z_c to prevent reflections of the signal at the ends of the line. (When the length of the line is short compared to the wavelength, impedance mismatch is the basis of transmission line impedance transformers, a topic that is addressed in the previous section.) In radio-frequency (RF) systems, a common value for source and load impedances is 50 ohms. A typical RF load is a quarter-wave ground plane antenna (37 ohms with an ideal ground plane but can be matched to 50 ohms by using a modified ground plane or a coaxial matching section, i.e. part or all the feeder being of higher impedance).

The general form of the voltage reflection coefficient for a wave moving from medium 1 to medium 2 is given by

$$\Gamma_{12} = \frac{Z_2 - Z_1}{Z_2 + Z_1}$$

while the voltage reflection coefficient for a wave moving from medium 2 to medium 1 is

$$\Gamma_{21} = \frac{Z_1 - Z_2}{Z_1 + Z_2}$$
$$\Gamma_{21} = -\Gamma_{12}$$

so the reflection coefficient is the same except for sign no matter from which direction the wave approaches the boundary.

There is also a current reflection coefficient. It is the same as the voltage coefficient except that it has opposite sign. Thus if the wave encounters an open at the load end, a positive voltage pulse and a negative current pulse are transmitted back toward the source. Negative current means the current is going the opposite direction.

Thus, at each boundary there are four reflection coefficients (voltage and current on one side and voltage and current on the other side). All four are the same except that two are positive and two are negative. Voltage reflection coefficient and current reflection coefficient on the same side have opposite signs. Voltage reflection coefficients on opposite sides of the boundary have opposite signs.

Because they are all the same except for sign, it is traditional to take reflection coefficient to mean voltage reflection coefficient unless otherwise indicated. Either or both ends of a transmission line can be a source or load or both, so there is no inherent preference for which side of the boundary is medium 1 and which side is medium 2. In the case where there is only one transmission line, it is customary to define the voltage reflection coefficient for a wave incident on the boundary from the transmission line side without regard to whether a source or load is connected on the other side.

Transmission line with single source driving a load

Conditions at the load end

In a transmission line, a wave travels from the source along the line. Suppose the wave hits a boundary (an abrupt change in impedance). Some of the wave is reflected back, while some keeps moving onwards. (Assume there is only one boundary and it is at the load.)

Let:

V_i and I_i be the voltage and current that is incident on the boundary from the source side.
V_t and I_t be the voltage and current that is transmitted to the load.
V_r and I_r be the voltage and current that is reflected back toward the source.

On the line side of the boundary $V_i = Z_c I_i$ and $V_r = -Z_c I_r$ and on the load side $V_t = Z_L I_t$ where V_i, V_r, V_t, I_i, I_r, I_t, and Z_c are phasors.

At a boundary, voltage and current must be continuous, therefore

$$V_t = V_i + V_r$$
$$I_t = I_i + I_r$$

All these conditions are satisfied by

$$V_r = \Gamma_{TL} V_i$$
$$I_r = -\Gamma_{TL} I_i$$
$$V_t = (1 + \Gamma_{TL}) V_i$$
$$I_t = (1 - \Gamma_{TL}) I_i$$

where : Γ_{TL} the reflection coefficient going from the transmission line to the load.

$$\Gamma_{TL} = \frac{Z_L - Z_c}{Z_L + Z_c} = \Gamma_L$$

The purpose of a transmission line is to get the maximum amount of energy to the other end of the line, or to transmit information with minimal error, so the reflection should be as small as possible. This is achieved by matching the impedances Z_L and Z_c so that they are equal ($\Gamma = 0$).

Conditions at the source end

At the source end of the transmission line, there may be waves incident both from the source and from the line and one can compute a reflection coefficient for each direction

with $-\Gamma_{ST} = \Gamma_{TS} = \frac{Z_s - Z_c}{Z_s + Z_c} = \Gamma_S$, where **Zs** is the source impedance. The source of waves that are incident from the line are the reflections from the load end. If the source impedance matches the line, then reflections from the load end will be absorbed at the source end. If the transmission line is not matched at both ends then reflections from the load will be re-reflected at the source and re-re-reflected at the load etc. losing energy on each transit of the transmission line. This can cause a resonance condition that can cause strong frequency dependent behavior. In a narrow band system this can be desirable for matching, but it is generally undesirable in a wide band system.

Impedance at the source end

$$Z_{in} = Z_C \frac{(1 + T^2\Gamma_L)}{(1 - T^2\Gamma_L)}$$

where T is the one way transfer function from either end to the other end when the transmission line is exactly matched at source and load. T accounts for everything that happens to the signal in transit including delay, attenuation and dispersion. Note that if there is a perfect match at the load then $\Gamma_L = 0$ and $Z_{in} = Z_C$

Overall transfer function

$$V_L = V_S \frac{T(1 - \Gamma_S)(1 + \Gamma_L)}{2(1 - T^2\Gamma_S\Gamma_L)}$$

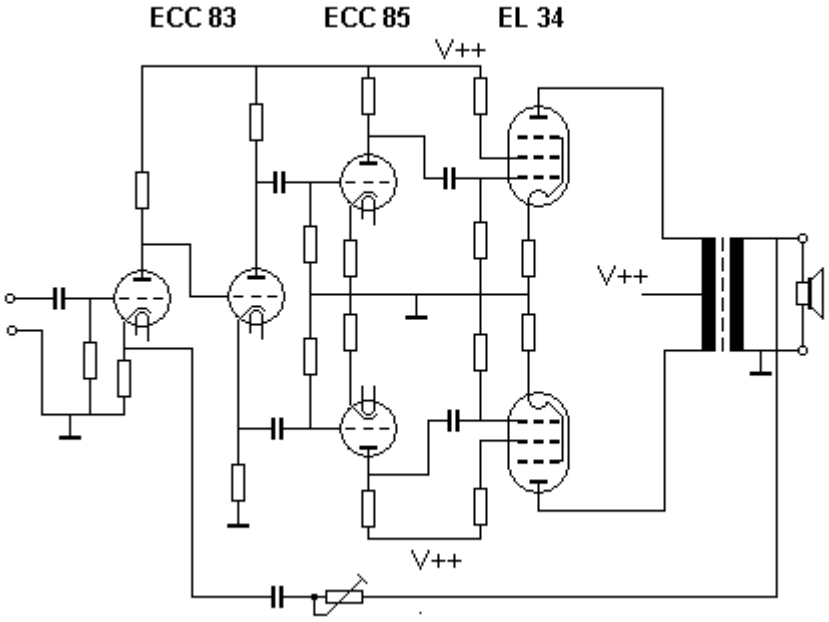
where V_S is the open circuit (or unloaded) output voltage from the source.

Note that if there is a perfect match at both ends $\Gamma_L = 0$ and $\Gamma_S = 0$ and then $V_L = V_S \frac{T}{2}$

Electrical examples

Telephone systems

Telephone systems also use matched impedances to minimise echoes on long distance lines. This is related to transmission lines theory. Matching also enables the telephone *hybrid coil* (2 to 4 wire conversion) to operate correctly. As the signals are sent and received on the same two-wire circuit to the central office (or exchange), cancellation is necessary at the telephone earpiece so that excessive sidetone is not heard. All devices used in telephone signal paths are generally dependent on using matched cable, source and load impedances. In the local loop, the impedance chosen is 600 ohm (nominal). Terminating networks are installed at the exchange to try to offer the best match to their subscriber lines. Each country has its own standard for these networks but they are all designed to approximate to about 600 ohms over the voice frequency band.



Typical push-pull audio tube power amplifier matched to the loudspeaker with an impedance matching transformer.
Loudspeaker amplifiers

Many modern solid state audio amplifiers do not use matched impedances, because semiconductor based amplifiers do not have output transformers. The driver amplifier has a low output impedance, such as < 0.1 ohm, and the loudspeaker usually has an input impedance of 4, 8, or 16 ohms, which is many times larger than the former. This type of connection is impedance bridging, and it provides better damping of the loudspeaker cone to minimize distortion. The misconception arises from tube audio amplifiers, which required impedance matching for proper, reliable operation. Most of these had output transformer taps to approximately match the amplifier output to typical loudspeaker impedances.

The output transformer in the vacuum tube based amplifiers has two basic functions:

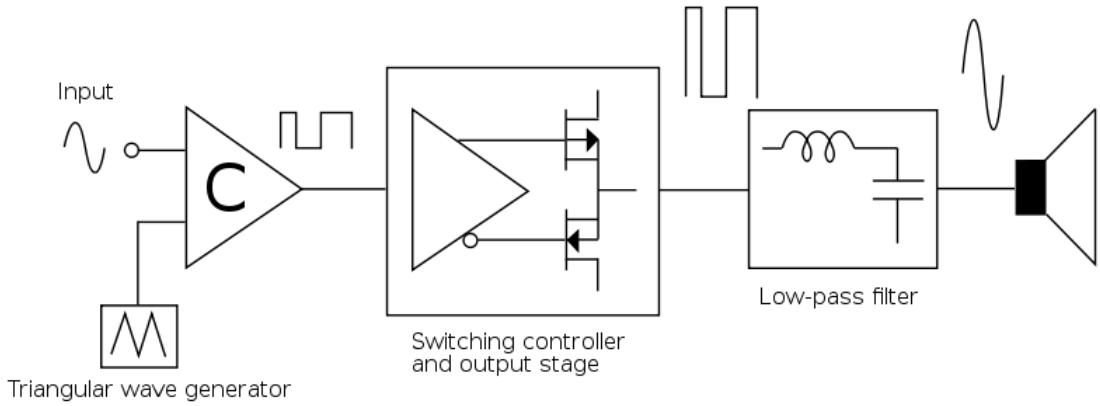
1. Separation of the AC component, which contains the audio signals, from the DC component, supplied by the power supply, in the anode circuit of vacuum tube based power stage. A loudspeaker should not be subjected to DC current.
2. Matching of the low impedance of popular loudspeakers to the high internal resistance of power pentodes such as the EL34.

The impedance of the loudspeaker on the secondary coil of the transformer will be transformed to a higher impedance on the primary coil in the circuit of the power pentodes by the square of the turns ratio, which forms the *impedance scaling factor*.

The secondary impedance of the loudspeaker is frequently moved (or "referred") to the primary side after multiplying the components by the impedance scaling factor $\left(\frac{N_P}{N_S}\right)^2$.

- *Np* = the number of windings of the primary coil
- *Ns* = the number of windings of the secondary coil

The required turns ratio can be computed with a given internal resistance of power pentodes in parallel = 3500 Ohm and a given loudspeaker impedance = 4 Ohm:



Class D amplifier with integrator in the endstage

$$\sqrt{\frac{3500}{4}} = 29.6$$

is the turns ratio of the output transformer. A careless selection of the tap of the output transformer to match the impedance of the loudspeaker used will result in power loss.

The output stage in semiconductor based endstages with MOSFETs or power transistors, do have a very low internal resistances. If they are properly balanced, then there is no need for a device as a transformer or a (big) electrolytic capacitor to separate AC current from DC current. In class D amplifiers, based on pulse conversion, is only a integrator necessary to generate the audio signal and to block the sample frequencies .

Non-electrical examples

Acoustics

Similar to electrical transmission lines, the impedance matching problem exists when transferring sound energy from one medium to another. If the acoustic impedance of the two media are very different, then most of the sound energy will be reflected or absorbed, rather than transferred across the border.

The gel used in medical ultrasonography helps transfer acoustic energy from the transducer to the body and back again. Without the gel, the "impedance mismatch" in the transducer-to-air and the air-to-body discontinuity reflects almost all the energy, leaving very little to go into the body.

Horns are used like transformers, matching the impedance of the transducer to the impedance of the air. This principle is used in both horn loudspeakers and musical instruments.

Most loudspeaker systems themselves contain impedance matching mechanisms, especially for low frequencies. Because most driver impedances are poorly matched to the impedance of free air at low frequencies, and because of out-of-phase cancellations between output from the front of a speaker cone and from the rear, loudspeaker enclosures serve both to match impedances and prevent the interference. Sound coupling into air from a loudspeaker is related to the ratio of the diameter of the speaker to the wavelength of the sound being reproduced. That is, larger speakers can produce lower frequencies at higher levels than smaller speakers for this reason. Elliptical speakers are a complex case, acting like large speakers lengthwise, and like small speakers crosswise.

Acoustic impedance matching (or the lack of it) affects the operation of a megaphone, an echo, and soundproofing.

Optics

A similar effect occurs when light (or any electromagnetic wave) hits the interface between two media with different refractive indices. For non-magnetic materials, refractive index is inversely proportional to the material's characteristic impedance. An *optical* or *wave impedance* that depends on the propagation direction can be calculated for each medium, and may be used in the usual transmission line reflection equation

$$r = \frac{Z_2 - Z_1}{Z_1 + Z_2}$$

to calculate the reflection and transmission coefficients for the interface. For non-magnetic dielectrics, this equation is equivalent to the Fresnel equations. Unwanted reflections can be reduced by the use of an anti-reflection optical coating.

Mechanics

If a body of mass m collides elastically with a second body, the maximum energy transferred to the second body will occur when the second body has the same mass m. For a head-on collision, with equal masses, the energy of the first body will be completely transferred to the second body. In this case, the masses act as "mechanical impedances" which must be matched. If m_1 and m_2 are the masses of the moving and the stationary body respectively, and P is the momentum of the system, which remains

constant throughout the collision, then the energy of the second body after the collision will be E_2:

$$E_2 = \frac{2P^2 m_2}{(m_1 + m_2)^2}$$

which is analogous to the power transfer equation in the above "mathematical proof" section.

These principles are useful in the application of highly energetic materials (explosives). If an explosive charge is placed upon a target, the sudden release of energy causes compression waves to propagate through the target radially from the point charge contact. When the compression waves reach areas of high acoustic impedance mismatch (like the other side of the target), tension waves reflect back and create spalling. The greater the mismatch, the greater the effect of creasing and spalling will be. A charge initiated against a wall with air behind it will do more damage to the wall than a charge initiated against a wall with soil behind it.

Chapter-9

Propagation Constant and Multidelay Block Frequency Domain Adaptive Filter

Propagation constant

The **propagation constant** of an electromagnetic wave is a measure of the change undergone by the amplitude of the wave as it propagates in a given direction. The quantity being measured can be the voltage or current in a circuit or a field vector such as electric field strength or flux density. The propagation constant itself measures change per metre but is otherwise dimensionless.

The propagation constant is expressed logarithmically, almost universally to the base e, rather than the more usual base 10 used in telecommunications in other situations. The quantity measured, such as voltage, is expressed as a sinusoidal phasor. The phase of the sinusoid varies with distance which results in the propagation constant being a complex number, the imaginary part being caused by the phase change.

Alternative names

The term propagation constant is somewhat of a misnomer as it usually varies strongly with ω. It is probably the most widely used term but there are a large variety of alternative names used by various authors for this quantity. These include, **transmission parameter**, **transmission function**, **propagation parameter**, **propagation coefficient** and **transmission constant**. In plural, it is usually implied that α and β are being referenced separately but collectively as in **transmission parameters**, **propagation parameters**, **propagation coefficients**, **transmission constants** and **secondary coefficients**. This last occurs in transmission line theory, the term *secondary* being used to contrast to the *primary line coefficients*. The primary coefficients being the physical properties of the line; R,C,L and G, from which the secondary coefficients may be derived using the telegrapher's equation. Note that, at least in the field of transmission

lines, the term transmission coefficient has a different meaning despite the similarity of name. Here it is the corollary of reflection coefficient.

Definition

The propagation constant, symbol γ, for a given system is defined by the ratio of the amplitude at the source of the wave to the amplitude at some distance *x*, such that,

$$\frac{A_0}{A_x} = e^{\gamma x}$$

Since the propagation constant is a complex quantity we can write:

$$\gamma = \alpha + i\beta$$

where

α, the real part, is called the attenuation constant
β, the imaginary part, is called the phase constant

That β does indeed represent phase can be seen from Euler's formula;

$$e^{i\theta} = \cos\theta + i\sin\theta$$

which is a sinusoid which varies in phase as θ varies but does not vary in amplitude because;

$$\left|e^{i\theta}\right| = \sqrt{\cos^2\theta + \sin^2\theta} = 1$$

The reason for the use of base e is also now made clear. The imaginary phase constant, iβ, can be added directly to the attenuation constant, α, to form a single complex number that can be handled in one mathematical operation provided they are to the same base. Angles measured in radians require base e, so the attenuation is likewise in base e.

The propagation constant for copper (or any other conductor) lines can be calculated from the primary line coefficients by means of the relationship;

$$\gamma = \sqrt{ZY}$$

where;

$Z = R + i\omega L$, the series impedance of the line per metre and,
$Y = G + i\omega C$, the shunt admittance of the line per metre.

Attenuation constant

In telecommunications, the term **attenuation constant**, also called **attenuation parameter** or **coefficient**, is the attenuation of an electromagnetic wave propagating through a medium per unit distance from the source. It is the real part of the propagation constant and is measured in nepers per metre. A neper is approximately 8.7dB. Attenuation constant can be defined by the amplitude ratio;

$$\left|\frac{A_0}{A_x}\right| = e^{\alpha x}$$

The propagation constant per unit length is defined as the natural logarithmic of ratio of the sending end current or voltage to the receiving end current or voltage.

Copper lines

The attenuation constant for copper lines (or ones made of any other conductor) can be calculated from the primary line coefficients as shown above. For a line meeting the distortionless condition, with a conductance G in the insulator, the attenuation constant is given by;

$$\alpha = \sqrt{RG}$$

however, a real line is unlikely to meet this condition without the addition of loading coils and, furthermore, there are some frequency dependant effects operating on the primary "constants" which cause a frequency dependence of the loss. There are two main components to these losses, the metal loss and the dielectric loss.

The loss of most transmission lines are dominated by the metal loss, which causes a frequency dependency due to finite conductivity of metals, and the skin effect inside a conductor. The skin effect causes R along the conductor to be approximately dependent on frequency according to;

$$R \propto \sqrt{\omega}$$

Losses in the dielectric depend on the loss tangent (*tan*δ) of the material, which depends inversely on the wavelength of the signal and is directly proportional to the frequency.

$$\alpha_d = \frac{\pi\sqrt{\varepsilon_r}}{\lambda}\tan\delta$$

Optical fibre

The attenuation constant for a particular propagation mode in an optical fiber, the real part of the axial propagation constant.

Phase constant

In electromagnetic theory, the **phase constant**, also called **phase change constant**, **parameter** or **coefficient** is the imaginary component of the propagation constant for a plane wave. It represents the change in phase per metre along the path travelled by the wave at any instant and is equal to the angular wavenumber of the wave. It is represented by the symbol β and is measured in units of radians per metre.

From the definition of angular wavenumber;

$$k = \frac{2\pi}{\lambda} = \beta$$

This quantity is often (strictly speaking incorrectly) abbreviated to wavenumber. Properly, wavenumber is given by,

$$\tilde{\nu} = 1/\lambda$$

which differs from angular wavenumber only by a constant multiple of 2π, in the same way that angular frequency differs from frequency.

For a transmission line, the Heaviside condition of the telegrapher's equation tells us that the wavenumber must be proportional to frequency for the transmission of the wave to be undistorted in the time domain. This includes, but is not limited to, the ideal case of a lossless line. The reason for this condition can be seen by considering that a useful signal is composed of many different wavelengths in the frequency domain. For there to be no distortion of the waveform, all these waves must travel at the same velocity so that they arrive at the far end of the line at the same time as a group. Since wave phase velocity is given by;

$$v_p = \frac{\lambda}{T} = \frac{f}{\tilde{\nu}} = \frac{\omega}{\beta}$$

it is proved that β is required to be proportional to ω. In terms of primary coefficients of the line, this yields from the telegrapher's equation for a distortionless line the condition;

$$\beta = \omega\sqrt{LC}$$

However, practical lines can only be expected to approximately meet this condition over a limited frequency band.

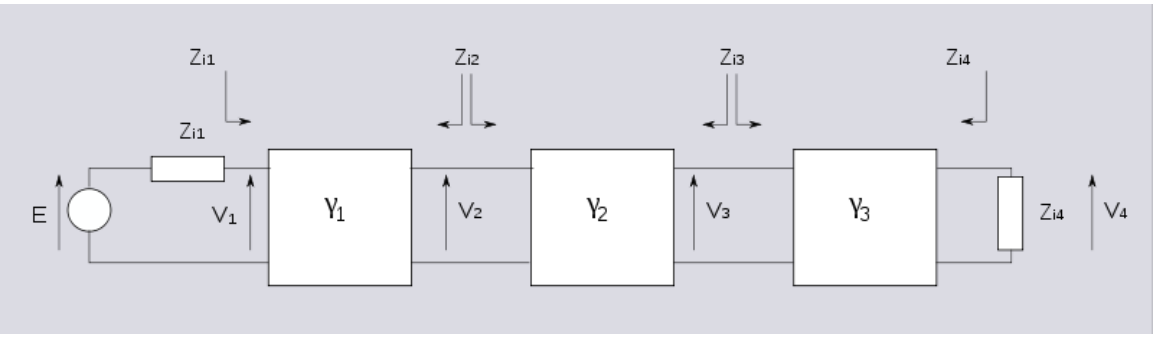
Filters

The term propagation constant or propagation function is applied to filters and other two-port networks used for signal processing. In these cases, however, the attenuation and

phase coefficients are expressed in terms of nepers and radians per network section rather than per metre. Some authors make a distinction between per metre measures (for which "constant" is used) and per section measures (for which "function" is used).

The propagation constant is a useful concept in filter design which invariably uses a cascaded section topology. In a cascaded topology, the propagation constant, attenuation constant and phase constant of individual sections may be simply added to find the total propagation constant etc.

Cascaded networks



Three networks with arbitrary propagation constants and impedances connected in cascade. The Z_i terms represent image impedance and it is assumed that connections are between matching image impedances.

The ratio of output to input voltage for each network is given by,

$$\frac{V_1}{V_2} = \sqrt{\frac{Z_{I1}}{Z_{I2}}} e^{\gamma_1}$$

$$\frac{V_2}{V_3} = \sqrt{\frac{Z_{I2}}{Z_{I3}}} e^{\gamma_2}$$

$$\frac{V_3}{V_4} = \sqrt{\frac{Z_{I3}}{Z_{I4}}} e^{\gamma_3}$$

The terms $\sqrt{\frac{Z_{In}}{Z_{Im}}}$ are impedance scaling terms and their use is explained in the image impedance article.

The overall voltage ratio is given by,

$$\frac{V_1}{V_4} = \frac{V_1}{V_2} \cdot \frac{V_2}{V_3} \cdot \frac{V_3}{V_4} = \sqrt{\frac{Z_{I1}}{Z_{I4}}} e^{\gamma_1 + \gamma_2 + \gamma_3}$$

Thus for *n* cascaded sections all having matching impedances facing each other, the overall propagation constant is given by,

$$\gamma_{Tot} = \gamma_1 + \gamma_2 + \gamma_3 + \cdots + \gamma_n$$

Multidelay block frequency domain adaptive filter

The **Multidelay block frequency domain adaptive filter** (MDF) algorithm is a block-based frequency domain implementation of the (normalised) Least mean squares filter (LMS) algorithm.

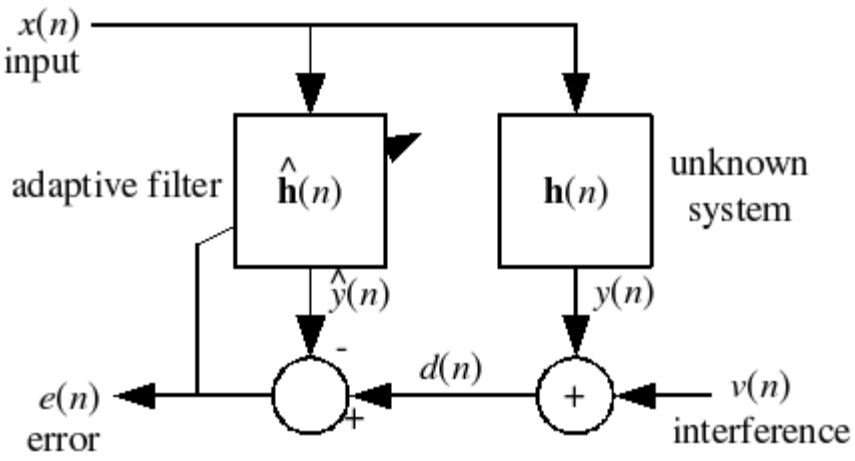
Introduction

The MDF algorithm is based on the fact that convolutions may be efficiently computed in the frequency domain (thanks to the Fast Fourier Transform). However, the algorithm differs from the Fast LMS algorithm in that block size it uses may be smaller than the filter length. If both are equal, then MDF reduces to the FLMS algorithm.

The advantages of MDF over the (N)LMS algorithm are:

- Lower algorithmic complexity
- Partial de-correlation of the input (which 'may' lead to faster convergence)

Variable definitions



Let *N* be the length of the processing blocks, *K* be the number of blocks and $\mathbf{F}$ denote the 2Nx2N Fourier transform matrix. The variables are defined as:

$$\underline{\mathbf{e}}(\ell) = \mathbf{F}\left[\mathbf{0}_{1xN}, e(\ell N), \ldots, e(\ell N - N - 1)\right]^T$$
$$\underline{\mathbf{x}}_k(\ell) = \mathrm{diag}\left\{\mathbf{F}\left[x((\ell - k + 1)N), \ldots, x((\ell - k - 1)N - 1)\right]^T\right\}$$
$$\underline{\mathbf{X}}(\ell) = \left[\underline{\mathbf{x}}_0(\ell), \underline{\mathbf{x}}_1(\ell), \ldots, \underline{\mathbf{x}}_{K-1}(\ell)\right]$$
$$\underline{\mathbf{d}}(\ell) = \mathbf{F}\left[\mathbf{0}_{1xN}, d(\ell N), \ldots, d(\ell N - N - 1)\right]^T$$

With normalisation matrices $\mathbf{G}_1$ and $\mathbf{G}_2$:

$$\mathbf{G}_1 = \mathbf{F}\begin{bmatrix}\mathbf{0}_{NxN} & \mathbf{0}_{NxN} \\ \mathbf{0}_{NxN} & \mathbf{I}_{NxN}\end{bmatrix}\mathbf{F}^{-1}$$
$$\tilde{\mathbf{G}}_2 = \mathbf{F}\begin{bmatrix}\mathbf{I}_{NxN} & \mathbf{0}_{NxN} \\ \mathbf{0}_{NxN} & \mathbf{0}_{NxN}\end{bmatrix}\mathbf{F}^{-1}$$
$$\mathbf{G}_2 = \mathrm{diag}\left\{\tilde{\mathbf{G}}_2, \tilde{\mathbf{G}}_2, \ldots, \tilde{\mathbf{G}}_2\right\}$$

In practice, when multiplying a column vector $\mathbf{x}$ by $\mathbf{G}_1$, we take the inverse FFT of $\mathbf{x}$, set the first *N* values in the result to zero and then take the FFT. This is meant to remove the effects of the circular convolution.

Algorithm description

For each block, the MDF algorithm is computed as:

$$\hat{\underline{\mathbf{y}}}(\ell) = \mathbf{G}_1 \underline{\underline{\mathbf{X}}}(\ell) \hat{\underline{\mathbf{h}}}(\ell - 1)$$
$$\underline{\mathbf{e}}(\ell) = \underline{\mathbf{d}}(\ell) - \hat{\underline{\mathbf{y}}}(\ell)$$
$$\mathbf{\Phi}_{\mathbf{xx}} = \underline{\underline{\mathbf{X}}}(\ell) \underline{\underline{\mathbf{X}}}(\ell)^H$$
$$\hat{\underline{\mathbf{h}}}(\ell) = \hat{\underline{\mathbf{h}}}(\ell - 1) + \mu \mathbf{G}_2 \mathbf{\Phi}_{\mathbf{xx}}^{-1}(\ell) \underline{\underline{\mathbf{X}}}^H(\ell) \underline{\mathbf{e}}(\ell)$$

It is worth noting that, while the algorithm is more easily expressed in matrix form, the actual implementation requires no matrix multiplications. For instance the normalisation matrix computation $\mathbf{\Phi}_{\mathbf{xx}} = \underline{\underline{\mathbf{X}}}(\ell) \underline{\underline{\mathbf{X}}}(\ell)^H$ reduces to an element-wise vector multiplication because $\underline{\underline{\mathbf{X}}}(\ell)$ is block-diagonal. The same goes for other multiplications.

Chapter-10

Linear Filter

Linear filters in the time domain process time-varying input signals to produce output signals, subject to the constraint of linearity. This results from systems composed solely of components (or digital algorithms) classified as having a linear response.

Most filters implemented in analog electronics, in digital signal processing, or in mechanical systems are classified as causal, time invariant, and linear. However the general concept of linear filtering is broader, also used in statistics, data analysis, and mechanical engineering among other fields and technologies. This includes noncausal filters and filters in more than one dimension such as would be used in image processing; those filters are subject to different constraints leading to different design methods, which are discussed elsewhere.

A linear time-invariant (LTI) filter can be uniquely specified by its impulse response h, and the output of any filter is mathematically expressed as the convolution of the input with that impulse response. The frequency response, given by the filter's transfer function $H(\omega)$, is an alternative characterization of the filter. The frequency response may be tailored to, for instance, eliminate unwanted frequency components from an input signal, or to limit an amplifier to signals within a particular band of frequencies.

Among the time-domain filters we here consider, there are two general classes of filter transfer functions that can approximate a desired frequency response. Very different mathematical treatments apply to the design of filters termed infinite impulse response (IIR) filters, characteristic of mechanical and analog electronics systems, and finite impulse response (FIR) filters, which can be implemented by discrete time systems such as computers (then termed *digital signal processing*).

Impulse response and transfer function

The impulse response h of a linear time-invariant causal filter specifies the output that the filter would produce if it were to receive an input consisting of a single impulse at time 0. An "impulse" in a continuous time filter means a Dirac delta function; in a discrete time filter the Kronecker delta function would apply. The impulse response completely

characterizes the response of any such filter, inasmuch as any possible input signal can be expressed as a (possibly infinite) combination of weighted delta functions. Multiplying the impulse response shifted in time according to the arrival of each of these delta functions by the amplitude of each delta function, and summing these responses together (according to the superposition principle, applicable to all linear systems) yields the output waveform.

Mathematically this is described as the convolution of a time-varying input signal *x(t)* with the filter's impulse response *h*, defined as:

$$y(t) = \int_0^T x(t-\tau)\, h(\tau)\, d\tau$$

$$y_k = \sum_{i=0}^{N} x_{k-i}\, h_i$$

The first form is the continuous-time form which describes mechanical and analog electronic systems, for instance. The second equation is a discrete-time version used, for example, by digital filters implemented in software, so-called *digital signal processing*. The impluse response *h* completely characterizes any linear time-invariant (or shift-invariant in the discrete-time case) filter. The input *x* is said to be "convolved" with the impulse response *h* having a (possibly infinite) duration of time *T* (or of *N* sampling periods).

The filter response can also be completely characterized in the frequency domain by its transfer function $H(\omega)$, which is the Fourier transform of the impulse response *h*. Typical filter design goals are to realize a particular frequency response, that is, the magnitude of the transfer function $| H(\omega) |$; the importance of the phase of the transfer function varies according to the application, inasmuch as the shape of a waveform can be distorted to a greater or lesser extent in the process of achieving a desired (amplitude) response in the frequency domain.

Filter design consists of finding a possible transfer function that can be implemented within certain practical constraints dictated by the technology or desired complexity of the system, followed by a practical design that realizes that transfer function using the chosen technology. The complexity of a filter may be specified according to the order of the filter, which is specified differently depending on whether we are dealing with an IIR or FIR filter. We will now look at these two cases.

Infinite impulse response filters

Consider a physical system that acts as a linear filter, such as a system of springs and masses, or an analog electronic circuit that includes capacitors and/or inductors (along with other linear components such as resistors and amplifiers). When such a system is subject to an impulse (or any signal of finite duration) it will respond with an output waveform which lasts past the duration of the input, eventually decaying exponentially in

one or another manner, but never completely settling to zero (mathematically speaking). Such a system is said to have an infinite impulse response (IIR). The convolution integral (or summation) above extends over all time: T (or N) must be set to infinity.

For instance, consider a damped harmonic oscillator such as a pendulum, or a resonant L-C tank circuit. If the pendulum has been at rest and we were to strike it with a hammer (the "impulse"), setting it in motion, it would swing back and forth ("resonate"), say, with an amplitude of 10cm. But after 10 minutes, say, it would still be swinging but the amplitude would have decreased to 5cm, half of its original amplitude. After another 10 minutes its amplitude would be only 2.5cm, then 1.25cm, etc. However it would never come to a complete rest, and we therefore call that response to the impulse (striking it with a hammer) "infinite" in duration.

The complexity of such a system is specified by its order N. N is often a constraint on the design of a transfer function since it specifies the number of reactive components in an analog circuit; in a digital IIR filter the number of computations required is proportional to N.

Finite impulse response filters

A filter implemented in a computer program (or a so-called digital signal processor) is a discrete-time system; a different (but parallel) set of mathematical concepts defines the behavior of such systems. Although a digital filter can be an IIR filter if the algorithm implementing it includes feedback, it is also possible to easily implement a filter whose impulse truly goes to zero after N time steps; this is called a finite impulse response (FIR) filter.

For instance, suppose we have a filter which, when presented with an impulse in a time series:

0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.....

will output a series which responds to that impulse at time 0 until time 4, and has no further response, such as:

0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0.....

Although the impulse response has lasted 4 time steps after the input, starting at time 5 it has truly gone to zero. The extent of the impulse response is *finite*, and this would be classified as a 4^{th} order FIR filter. The convolution integral (or summation) above need only extend to the full duration of the impulse response T, or the order N in a discrete time filter.

Implementation issues

Classical analog filters are IIR filters, and classical filter theory centers on the determination of transfer functions given by low order rational functions, which can be synthesized using the same small number of reactive components. Using digital computers, on the other hand, both FIR and IIR filters are straightforward to implement in software.

A digital IIR filter can generally approximate a desired filter response using less computing power than a FIR filter, however this advantage is more often unneeded given the increasing power of digital processors. The ease of designing and characterizing FIR filters makes them preferable to the filter designer (programmer) when ample computing power is available. Another advantage of FIR filters is that their impulse response can be made symmetric, which implies a response in the frequency domain which has zero phase at all frequencies (not considering a finite delay), which is absolutely impossible with any IIR filter.

Frequency response

The frequency response or transfer function $| H(\omega) |$ of a filter can be obtained if the impulse response is known, or directly through analysis using Laplace transforms, or in discrete-time systems the Z-transform. The frequency response also includes the phase as a function of frequency, however in many cases the phase response is of little or no interest. FIR filters can be made to have zero phase, but with IIR filters that is generally impossible With most IIR transfer functions there are related transfer functions having a frequency response with the same magnitude but a different phase; in most case the so-called minimum phase transfer function is preferred.

Filters in the time domain are most often requested to follow a specified frequency response. Then a mathematical procedure is used to find a filter transfer function which can be realized (within some constraints) and which approximates the desired response to within some criterion. Common filter response specifications are described as follows:

- A low-pass filter passes low frequencies while blocking higher frequencies.
- A high-pass filter passes high frequencies.
- A band-pass filter passes a band (range) of frequencies.
- A band-stop filter passes high and low frequencies outside of a specified band.
- A notch filter has a null response at a particular frequency. This function may be combined with one of the above responses.
- An all-pass filter passes all frequencies equally well, but alters the phase relationship among them.
- An equalization filter is not designed to fully pass or block any frequency, but instead to gradually vary the amplitude response as a function of frequency: filters used as pre-emphasis filters, equalizers, or tone controls are good examples.

FIR transfer functions

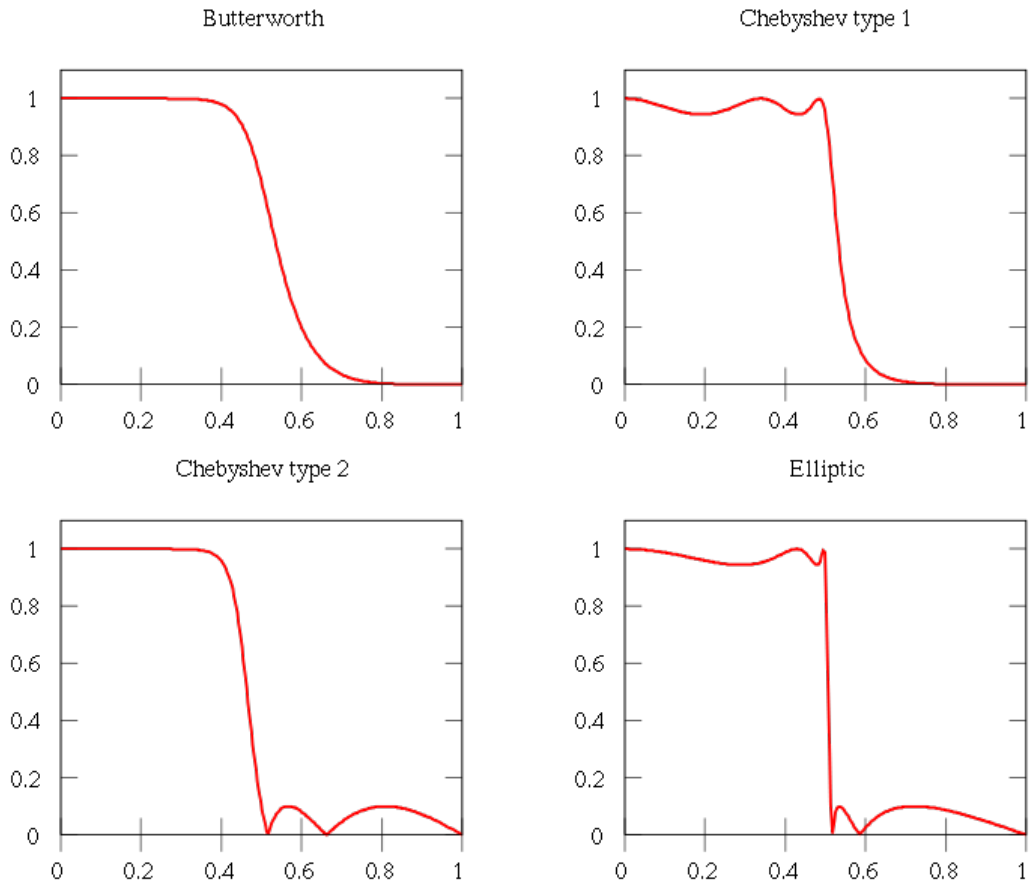
Meeting a frequency response requirement with an FIR filter uses relatively straight-forward procedures. In the most basic form, the desired frequency response itself can be sampled with a resolution of Δf and fourier transformed to the time domain. This will obtain the filter coefficients h_i which will implement a zero phase FIR filter which matches the frequency response at the sampled frequencies used. In order to better match a desired response, Δf must be reduced. However the duration of the filter's impulse response, and the number of terms which must be summed for each output value (according to the above discrete time convolution) is given by $N = 1/(\Delta f\, T)$ where T is the sampling period of the discrete time system (N-1 is also termed the *order* of an FIR filter). Thus the complexity of a digital filter and the computing time involved, grows inversely with Δf, placing a higher cost on filter functions which better approximate the desired behavior. For the same reason, filter functions whose critical response is at lower frequencies (compared to the sampling frequency *1/T*) require a higher order, more computationally intensive FIR filter. An IIR filter can thus be much more efficient in such cases.

Elsewhere the reader may find further discussion of design methods for practical FIR filter design.

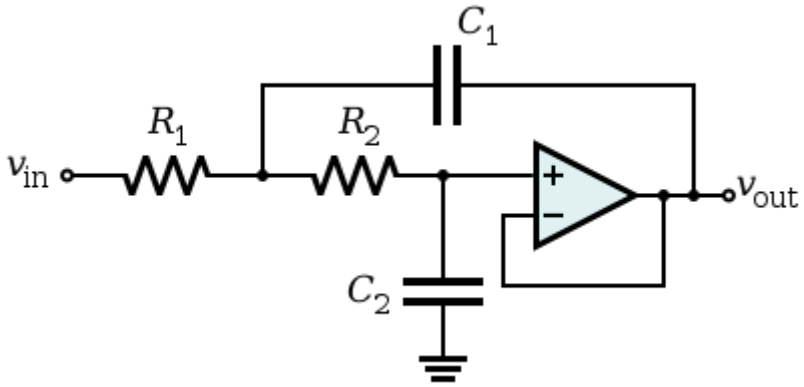
IIR transfer functions

Since classical analog filters are IIR filters, there has been a long history of studying the range of possible transfer functions implementing various of the above desired filter responses in continuous time systems. Using transforms it is possible to convert these continuous time frequency responses to ones that are implemented in discrete time, for use in digital IIR filters. The complexity of any such filter is given by the *order* N, which describes the order of the rational function describing the frequency response. The order N is of particular importance in analog filters, because an N^{th} order electronic filter requires N reactive elements (capactors and/or inductors) to implement. If a filter is implemented using, for instance, biquad stages using op-amps, N/2 stages will be needed. In a digital implementation, the number of computations performed per sample is proportional to N. Thus the mathematical problem is to obtain the best approximation (in some sense) to the desired response using a smaller N, as we shall now illustrate.

Below are the frequency responses of several standard filter functions which approximate a desired response, optimized according to some criterion. These are all fifth-order low-pass filters, designed for a cutoff frequency of .5 in normalized units. Frequency responses are shown for the Butterworth, Chebyshev, inverse Chebyshev, and elliptic filters.



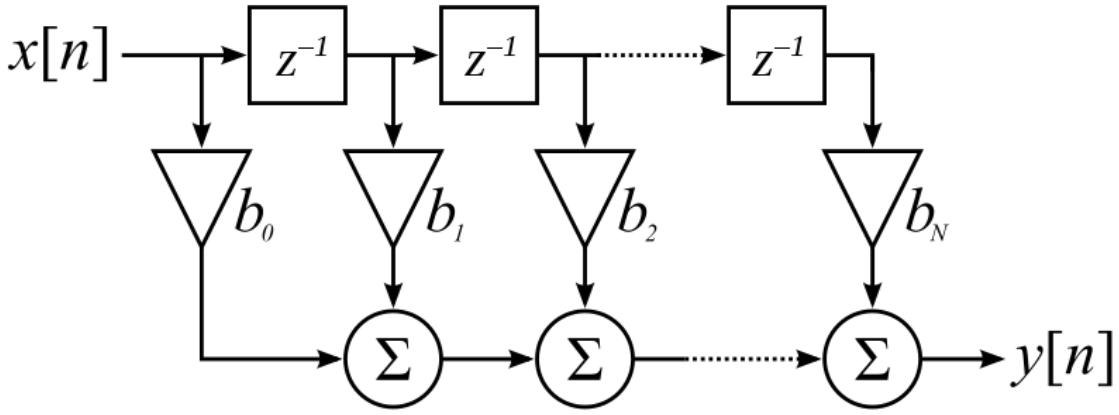
As is clear from the image, the elliptic filter is sharper than the others, but at the expense of ripples in both its passband and stopband. The Butterworth filter has the poorest transition but has a more even response, avoiding ripples in either the passband or stopband. A Bessel filter (not shown) has an even poorer transition in the frequency domain, but maintains the best phase fidelity of a waveform. Different applications will emphasize different design requirements, leading to different choices among these (and other) optimizations, or requiring a filter of a higher order.



Low-pass filter implemented with a Sallen–Key topology

Example implementations

A popular circuit implementing a second order active R-C filter is the Sallen-Key design, whose schematic diagram is shown here. This topology can be adapted to produce low-pass, band-pass, and high pass filters.



A discrete-time FIR filter of order *N*. The top part is an *N*-sample delay line; each delay step is denoted z^{-1}.

An N^{th} order FIR filter can be implemented in a discrete time system using a computer program or specialized hardware in which the input signal is subject to N delay stages. The output of the filter is formed as the weighted sum of those delayed signals, as is depicted in the accompanying signal flow diagram. The response of the filter depends on the weighting coefficients denoted b_0, b_1, ... b_N. For instance, if all of the coefficients were equal to unity, a so-called boxcar function, then it would implement a low-pass filter with a low frequency gain of N+1 and a frequency response given by the sinc function. Superior shapes for the frequency response can be obtained using coefficients derived from a more sophisticated design procedure.

Mathematics of filter design

LTI system theory describes linear *time-invariant* (LTI) filters of all types. LTI filters can be completely described by their frequency response and phase response, the specification of which uniquely defines their impulse response, and *vice versa*. From a mathematical viewpoint, continuous-time IIR LTI filters may be described in terms of linear differential equations, and their impulse responses considered as Green's functions of the equation. Continuous-time LTI filters may also be described in terms of the Laplace transform of their impulse response, which allows all of the characteristics of the filter to be analyzed by considering the pattern of poles and zeros of their Laplace transform in the complex plane. Similarly, discrete-time LTI filters may be analyzed via the Z-transform of their impulse response.

Before the advent of computer filter synthesis tools, graphical tools such as Bode plots and Nyquist plots were extensively used as design tools. Even today, they are invaluable tools to understanding filter behavior. Reference books had extensive plots of frequency response, phase response, group delay, and impulse response for various types of filters, of various orders. They also contained tables of values showing how to implement such filters as RLC ladders - very useful when amplifying elements were expensive compared to passive components. Such a ladder can also be designed to have minimal sensitivity to component variation a property hard to evaluate without computer tools.

Many different analog filter designs have been developed, each trying to optimise some feature of the system response. For practical filters, a custom design is sometimes desirable, that can offer the best tradeoff between different design criteria, which may include component count and cost, as well as filter response characteristics.

These descriptions refer to the *mathematical* properties of the filter (that is, the frequency and phase response). These can be *implemented* as analog circuits (for instance, using a Sallen Key filter topology, a type of active filter), or as algorithms in digital signal processing systems.

Digital filters are much more flexible to synthesize and use than analog filters, where the constraints of the design permits their use. Notably, there is no need to consider component tolerances, and very high Q levels may be obtained.

FIR digital filters may be implemented by the direct convolution of the desired impulse response with the input signal. They can easily be designed to give a matched filter for any arbitrary pulse shape.

IIR digital filters are often more difficult to design, due to problems including dynamic range issues, quantization noise and instability. Typically digital IIR filters are designed as a series of digital biquad filters.

All low-pass second-order continuous-time filters have a transfer function given by

$$H(s) = \frac{K\omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}.$$

All band-pass second-order continuous-time have a transfer function given by

$$H(s) = \frac{K\frac{\omega_0}{Q}s}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}.$$

where

- K is the gain (low-pass DC gain, or band-pass mid-band gain) (K is 1 for passive filters)
- Q is the Q factor
- ω_0 is the center frequency
- $s = \sigma + j\omega$ is the complex frequency

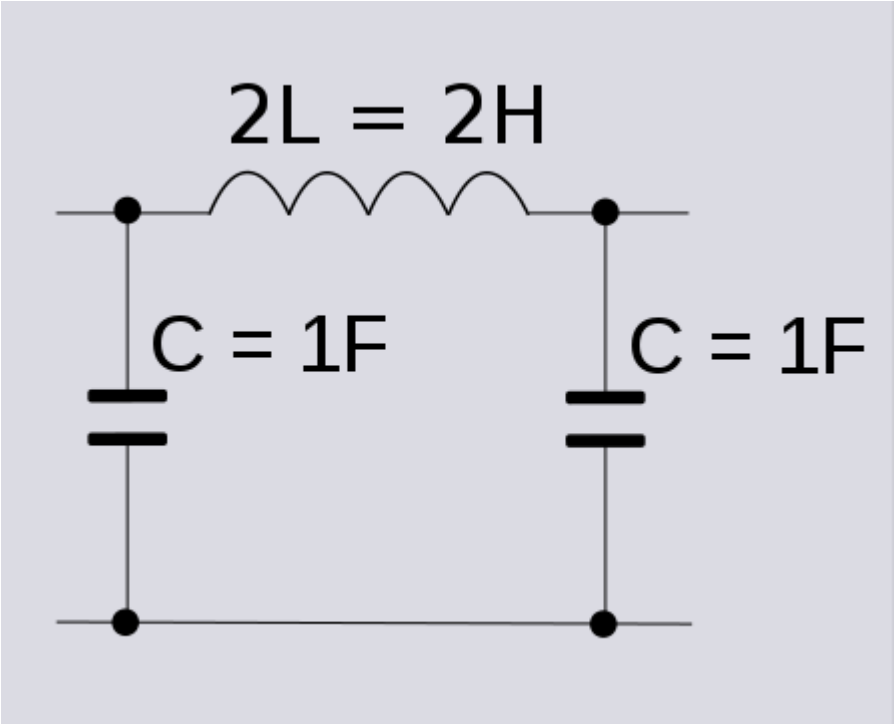
Chapter-11

Prototype Filter

Prototype filters are electronic filter designs that are used as a template to produce a modified filter design for a particular application. They are an example of a nondimensionalised design from which the desired filter can be scaled or transformed. They are most often seen in regards to electronic filters and especially linear analogue passive filters. However, in principle, the method can be applied to any kind of linear filter or signal processing, including mechanical, acoustic and optical filters.

Filters are required to operate at many different frequencies, impedances and bandwidths. The utility of a prototype filter comes from the property that all these other filters can be derived from it by applying a scaling factor to the components of the prototype. The filter design need thus only be carried out once in full, with other filters being obtained by simply applying a scaling factor.

Especially useful is the ability to transform from one bandform to another. In this case, the transform is more than a simple scale factor. Bandform here is meant to indicate the category of passband that the filter possesses. The usual bandforms are lowpass, highpass, bandpass and bandstop, but others are possible. In particular, it is possible for a filter to have multiple passbands. In fact, in some treatments, the bandstop filter is considered to be a type of multiple passband filter having two passbands. Most commonly, the prototype filter is expressed as a lowpass filter, but other techniques are possible.



A low pass prototype constant k Π filter

Low-pass prototype

The prototype is most often a low-pass filter with a 3dB corner frequency of angular frequency $\omega_c' = 1$ rad/s. Occasionally, frequency $f' = 1$ Hz is used instead of $\omega_c' = 1$. Likewise, the nominal or characteristic impedance of the filter is set to $R' = 1\ \Omega$.

In principle, any non-zero frequency point on the filter response could be used as a reference for the prototype design. For example, for filters with ripple in the passband the corner frequency is usually defined as the highest frequency at maximum ripple rather than 3dB. Another case is in image parameter filters (an older design method than the more modern network synthesis filters) which use the cut-off frequency rather than the 3dB point since cut-off is a well defined point in this types of filter.

The prototype filter can only be used to produce other filters of the same class and order. For instance, a fifth order Bessel filter prototype can be converted into any other fifth order Bessel filter, but it cannot be transformed into a third order Bessel filter or a fifth order Tchebyscheff filter.

Frequency scaling

The prototype filter is scaled to the frequency required with the following transformation:

$$i\omega \rightarrow \left(\frac{\omega_c'}{\omega_c}\right) i\omega$$

where ω_c' is the value of the frequency parameter (e.g. cut-off frequency) for the prototype and ω_c is the desired value. So if $\omega_c' = 1$ then the transfer function of the filter is transformed as:

$$A(i\omega) \rightarrow A\left(i\frac{\omega}{\omega_c}\right)$$

It can readily be seen that to achieve this, the non-resistive components of the filter must be transformed by:

$$L \rightarrow \frac{\omega_c'}{\omega_c} L \quad \text{and,} \quad C \rightarrow \frac{\omega_c'}{\omega_c} C$$

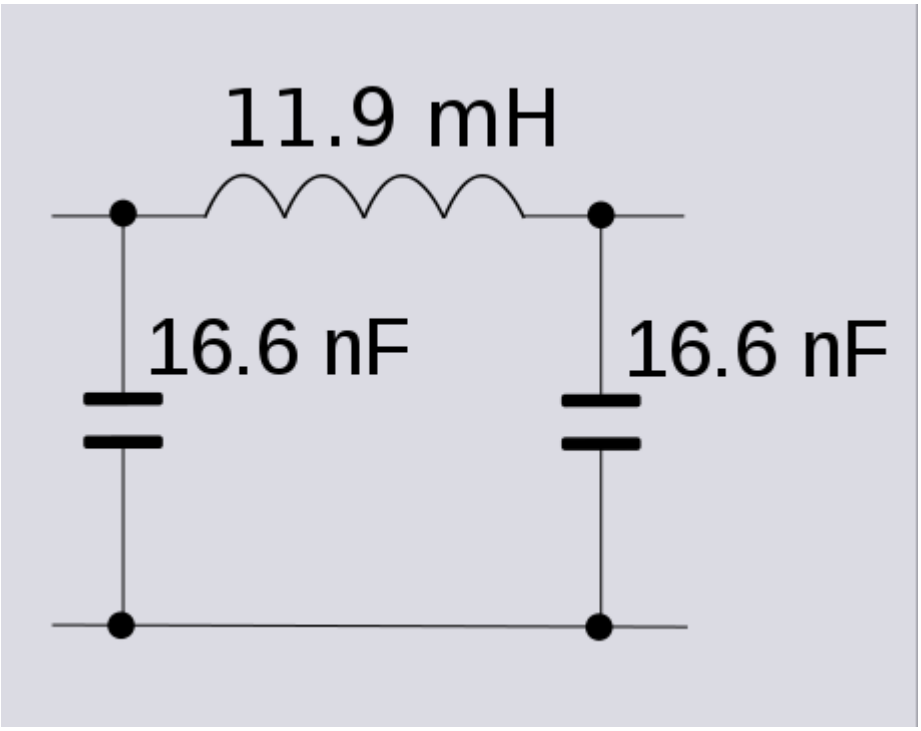
Impedance scaling

Impedance scaling is invariably a scaling to a fixed resistance. This is because the terminations of the filter, at least nominally, are taken to be a fixed resistance. To carry out this scaling to a nominal impedance *R*, each impedance element of the filter is transformed by:

$$Z \rightarrow \frac{R}{R'} Z$$

It may be more convenient on some elements to scale the admittance instead:

$$Y \rightarrow \frac{R'}{R} Y$$



The prototype filter above, transformed to a 600Ω, 16kHz lowpass filter

It can readily be seen that to achieve this, the non-resistive components of the filter must be scaled as:

$$L \to \frac{R}{R'} L \quad \text{and,} \quad C \to \frac{R'}{R} C$$

Impedance scaling by itself has no effect on the transfer function of the filter (providing that the terminating impedances have the same scaling applied to them). However, it is usual to combine the frequency and impedance scaling into a single step:

$$L \to \frac{\omega_c'}{\omega_c} \frac{R}{R'} L \quad \text{and,} \quad C \to \frac{\omega_c'}{\omega_c} \frac{R'}{R} C$$

Bandform transformation

In general, the bandform of a filter is transformed by replacing $i\omega$ where it occurs in the transfer function with a function of $i\omega$. This in turn leads to the transformation of the impedance components of the filter into some other component(s). The frequency scaling above is a trivial case of bandform transformation corresponding to a lowpass to lowpass transformation.

Lowpass to highpass

The frequency transformation required in this case is:

$$\frac{i\omega}{\omega_c'} \rightarrow \frac{\omega_c}{i\omega}$$

where ω_c is the point on the highpass filter corresponding to ω_c' on the prototype. The transfer function then transforms as:

$$A(i\omega) \rightarrow A\left(\frac{\omega_c\,\omega_c'}{i\omega}\right)$$

Inductors are transformed into capacitors according to,

$$L' \rightarrow C = \frac{1}{\omega_c\,\omega_c'\,L'}$$

and capacitors are transformed into inductors,

$$C' \rightarrow L = \frac{1}{\omega_c\,\omega_c'\,C'}$$

the primed quantities being the component value in the prototype.

Lowpass to bandpass

In this case, the required frequency transformation is:

$$\frac{i\omega}{\omega_c'} \rightarrow Q\left(\frac{i\omega}{\omega_0} + \frac{\omega_0}{i\omega}\right)$$

where Q is the Q-factor and is equal to the inverse of the fractional bandwidth:

$$Q = \frac{\omega_0}{\Delta\omega}$$

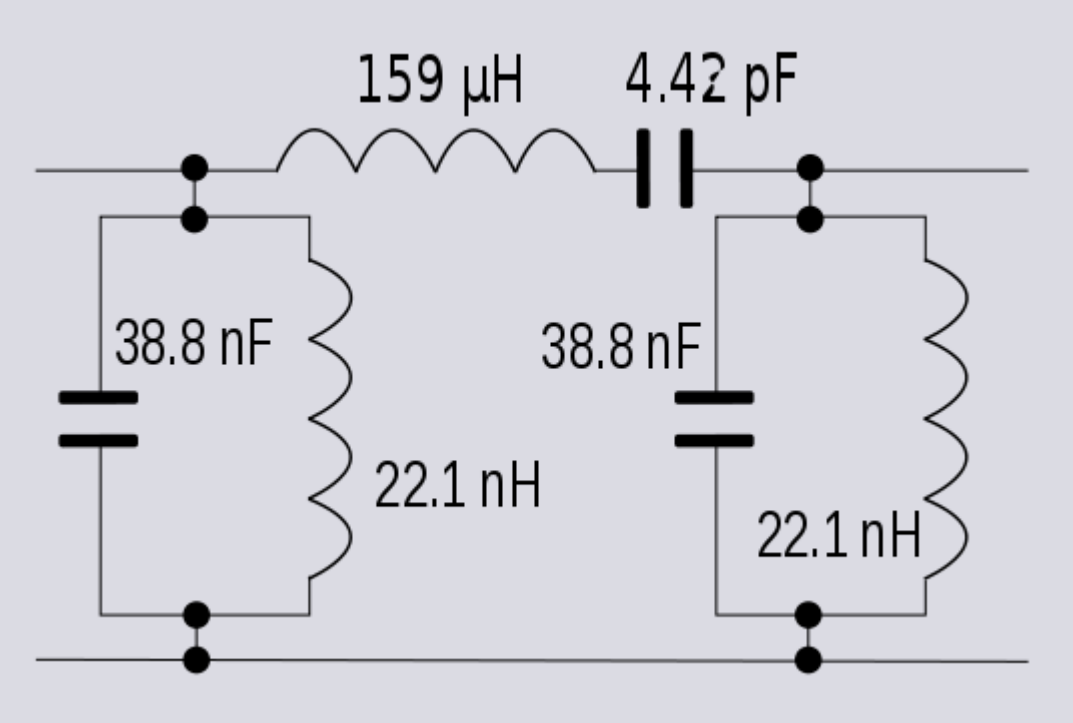
If ω_1 and ω_2 are the lower and upper frequency points (respectively) of the bandpass response corresponding to ω_c' of the prototype, then,

$\Delta\omega = \omega_2 - \omega_1$ and $\omega_0 = \sqrt{\omega_1\omega_2}$

$\Delta\omega$ is the absolute bandwidth, and ω_0 is the resonant frequency of the resonators in the filter. Note that frequency scaling the prototype prior to lowpass to bandpass transformation does not affect the resonant frequency, but instead affects the final bandwidth of the filter.

The transfer function of the filter is transformed according to:

$$A(i\omega) \rightarrow A\left(\omega_c' Q\left[\frac{i\omega}{\omega_0} + \frac{\omega_0}{i\omega}\right]\right)$$



The prototype filter above, transformed to a 50Ω, 6MHz bandpass filter with 100kHz bandwidth

Inductors are transformed into series resonators,

$$L' \rightarrow L = \frac{\omega_c' Q}{\omega_0} L' \, , \, C = \frac{1}{\omega_0 \omega_c' Q} \frac{1}{L'}$$

and capacitors are transformed into parallel resonators,

$$C' \rightarrow C = \frac{\omega_c' Q}{\omega_0} C' \parallel L = \frac{1}{\omega_0 \omega_c' Q} \frac{1}{C'}$$

Lowpass to bandstop

The required frequency transformation for lowpass to bandstop is:

$$\frac{\omega_c'}{i\omega} \to Q\left(\frac{i\omega}{\omega_0} + \frac{\omega_0}{i\omega}\right)$$

Inductors are transformed into parallel resonators,

$$L' \to L = \frac{\omega_c'}{\omega_0 Q} L' \parallel C = \frac{Q}{\omega_0 \omega_c'} \frac{1}{L'}$$

and capacitors are transformed into series resonators,

$$C' \to C = \frac{\omega_c'}{\omega_0 Q} C' \,,\; L = \frac{1}{\omega_0 Q \omega_c'} \frac{1}{C'}$$

Lowpass to multi-band

Filters with multiple passbands may be obtained by applying the general transformation:

$$\frac{\omega_c'}{i\omega} \to \frac{1}{Q_1\left(\frac{i\omega}{\omega_{01}} + \frac{\omega_{01}}{i\omega}\right)} + \frac{1}{Q_2\left(\frac{i\omega}{\omega_{02}} + \frac{\omega_{02}}{i\omega}\right)} + \cdots$$

The number of resonators in the expression corresponds to the number of passbands required. Lowpass and highpass filters can be viewed as special cases of the resonator expression with one or the other of the terms becoming zero as appropriate. Bandstop filters can be regarded as a combination of a lowpass and a highpass filter. Multiple bandstop filters can always be expressed in terms of a multiple bandpass filter. In this way it, can be seen that this transformation represents the general case for any bandform, and all the other transformations are to be viewed as special cases of it.

The same response can equivalently be obtained, sometimes with a more convenient component topology, by transforming to multiple stopbands instead of multiple passbands. The required transformation in those cases is:

$$\frac{i\omega}{\omega_c'} \to \frac{1}{Q_1\left(\frac{i\omega}{\omega_{01}} + \frac{\omega_{01}}{i\omega}\right)} + \frac{1}{Q_2\left(\frac{i\omega}{\omega_{02}} + \frac{\omega_{02}}{i\omega}\right)} + \cdots$$

Alternative prototype

In his treatment of image filters, Zobel provided an alternative basis for constructing a prototype which is not based in the frequency domain. The Zobel prototypes do not, therefore, correspond to any particular bandform, but they can be transformed into any of them. Not giving special significance to any one bandform makes the method more mathematically pleasing; however, it is not in common use.

The Zobel prototype considers filter sections, rather than components. That is, the transformation is carried out on a two-port network rather than a two-terminal inductor or capacitor. The transfer function is expressed in terms of the product of the series impedance, Z, and the shunt admittance Y of a filter half-section. This quantity is nondimensional, adding to the prototype's generality. Generally, ZY is a complex quantity,

$ZY = U + iV$ and as U and V are both, in general, functions of ω we should properly write,

$$ZY = U(\omega) + iV(\omega)$$

With image filters, it is possible to obtain filters of different classes from the constant k filter prototype by means of a different kind of transformation, constant k being those filters for which Z/Y is a constant. For this reason, filters of all classes are given in terms of $U(\omega)$ for a constant k, which is notated as,

$$ZY = U_k(\omega) + iV_k(\omega)$$

In the case of dissipationless networks, i.e. no resistors, the quantity $V(\omega)$ is zero and only $U(\omega)$ need be considered. $U_k(\omega)$ ranges from 0 at the centre of the passband to -1 at the cut-off frequency and then continues to increase negatively into the stopband regardless of the bandform of the filter being designed. To obtain the required bandform, the following transforms are used:

For a lowpass constant k prototype that is scaled:

$$R_0 = 1\,,\ \omega_c = 1$$

the independent variable of the response plot is,

$$U_k(\omega) = -\omega^2$$

The bandform transformations from this prototype are,

for lowpass, $$U_k(\omega) \rightarrow \left(\frac{i\omega}{\omega_c}\right)^2$$

for highpass, $$U_k(\omega) \to \left(\frac{\omega_c}{i\omega}\right)^2$$

and for bandpass, $$U_k(\omega) \to Q^2 \left(\frac{i\omega}{\omega_0} + \frac{\omega_0}{i\omega}\right)^2$$

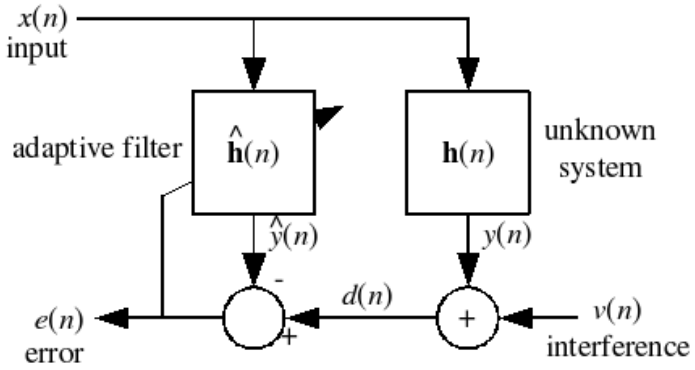
Chapter-12

Least Mean Squares Filter and Quarter Wave Impedance Transformer

Least mean squares filter

Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. It was invented in 1960 by Stanford University professor Bernard Widrow and his first Ph.D. student, Ted Hoff.

Problem formulation



Most linear adaptive filtering problems can be formulated using the block diagram above. That is, an unknown system $\mathbf{h}(n)$ is to be identified and the adaptive filter attempts to adapt the filter $\hat{\mathbf{h}}(n)$ to make it as close as possible to $\mathbf{h}(n)$, while using only observable signals $x(n)$, $d(n)$ and $e(n)$; but $y(n)$, $v(n)$ and $h(n)$ are not directly observable. Its solution is closely related to the Wiener filter.

definition of symbols

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$$
$$\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{p-1}(n)]^T, \quad \mathbf{h}(n) \in \mathbb{C}^p$$
$$y(n) = \mathbf{h}^H(n) \cdot \mathbf{x}(n)$$
$d(n) = y(n) + \nu(n)$
$$e(n) = d(n) - \hat{y}(n) = d(n) - \hat{\mathbf{h}}^H(n) \cdot \mathbf{x}(n)$$

Idea

The idea behind LMS filters is to use steepest descent to find filter weights $\mathbf{h}(n)$ which minimize a cost function. We start by defining the cost function as

$$C(n) = E\left\{|e(n)|^2\right\}$$

where $e(n)$ is the error at the current sample 'n' and $E\{.\}$ denotes the expected value.

This cost function ($C(n)$) is the mean square error, and it is minimized by the LMS. This is where the LMS gets its name. Applying steepest descent means to take the partial derivatives with respect to the individual entries of the filter coefficient (weight) vector

$$\nabla_{\hat{\mathbf{h}}^H} C(n) = \nabla_{\hat{\mathbf{h}}^H} E\left\{e(n)\, e^*(n)\right\} = 2E\left\{\nabla_{\hat{\mathbf{h}}^H}(e(n))\, e^*(n)\right\}$$

where ∇ is the gradient operator.

$$\nabla_{\hat{\mathbf{h}}^H} e(n) = \nabla_{\hat{\mathbf{h}}^H}\left(d(n) - \hat{\mathbf{h}}^H \cdot \mathbf{x}(n)\right) = -\mathbf{x}(n)$$
$$\nabla C(n) = -2E\left\{\mathbf{x}(n)\, e^*(n)\right\}$$

Now, $\nabla C(n)$ is a vector which points towards the steepest ascent of the cost function. To find the minimum of the cost function we need to take a step in the opposite direction of $\nabla C(n)$. To express that in mathematical terms

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) - \frac{\mu}{2}\nabla C(n) = \hat{\mathbf{h}}(n) + \mu\, E\left\{\mathbf{x}(n)\, e^*(n)\right\}$$

where $\frac{\mu}{2}$ is the step size(adaptation constant). That means we have found a sequential update algorithm which minimizes the cost function. Unfortunately, this algorithm is not realizable until we know $E\left\{\mathbf{x}(n)\, e^*(n)\right\}$.

Generally, the expectation above is not computed. Instead, to run the LMS in an online (updating after each new sample is received) environment, we use an instantaneous estimate of that expectation.

Simplifications

For most systems the expectation function $E\left\{\mathbf{x}(n)\,e^*(n)\right\}$must be approximated. This can be done with the following unbiased estimator

$$\hat{E}\left\{\mathbf{x}(n)\,e^*(n)\right\} = \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{x}(n-i)\,e^*(n-i)$$

where *N* indicates the number of samples we use for that estimate. The simplest case is *N* = 1

$$\hat{E}\left\{\mathbf{x}(n)\,e^*(n)\right\} = \mathbf{x}(n)\,e^*(n)$$

For that simple case the update algorithm follows as

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu\mathbf{x}(n)\,e^*(n)$$

Indeed this constitutes the update algorithm for the LMS filter.

LMS algorithm summary

The LMS algorithm for a *p*th order algorithm can be summarized as

Parameters: p = filter order

μ = step size

Initialisation: $\hat{\mathbf{h}}(0) = 0$

Computation: For $n = 0,1,2,\ldots$

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$$
$$e(n) = d(n) - \hat{\mathbf{h}}^H(n)\mathbf{x}(n)$$
$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu\,e^*(n)\mathbf{x}(n)$$

where $\hat{\mathbf{h}}^H(n)$denotes the Hermitian transpose of $\hat{\mathbf{h}}(n)$.

Convergence and stability in the mean

Assume that the true filter $\mathbf{h}(n) = \mathbf{h}$ is constant, and that the input signal $x(n)$ is wide-sense stationary. Then $E\{\hat{\mathbf{h}}(n)\}$ converges to $\mathbf{h}$ as $n \to \infty$ if and only if

$$0 < \mu < \frac{2}{\lambda_{\max}},$$

where λ_{max} is the greatest eigenvalue of the autocorrelation matrix $R = E\{\mathbf{x}(n)\mathbf{x}^H(n)\}$. If this condition is not fulfilled, the algorithm becomes unstable and $\hat{\mathbf{h}}(n)$ diverges.

Maximum convergence speed is achieved when

$$\mu = \frac{2}{\lambda_{\max} + \lambda_{\min}},$$

where λ_{min} is the smallest eigenvalue of R. Given that μ is less than or equal to this optimum, the convergence speed is determined by $\mu\lambda_{min}$, with a larger value yielding faster convergence. This means that faster convergence can be achieved when λ_{max} is close to λ_{min}, that is, the maximum achievable convergence speed depends on the eigenvalue spread of R.

A white noise signal has autocorrelation matrix $R = \sigma^2 I$, where σ^2 is the variance of the signal. In this case all eigenvalues are equal, and the eigenvalue spread is the minimum over all possible matrices. The common interpretation of this result is therefore that the LMS converges quickly for white input signals, and slowly for colored input signals, such as processes with low-pass or high-pass characteristics.

It is important to note that the above upperbound on μ only enforces stability in the mean, but the coefficients of $\hat{\mathbf{h}}(n)$ can still grow infinitely large, i.e. divergence of the coefficients is still possible. A more practical bound is

$$0 < \mu < \frac{2}{tr\,[R]},$$

where $tr\,[R]$ denotes the trace of R. This bound guarantees that the coefficients of $\hat{\mathbf{h}}(n)$ do not diverge (in practice, the value of μ should not be chosen close to this upper bound, since it is somewhat optimistic due to approximations and assumptions made in the derivation of the bound).

Normalised least mean squares filter (NLMS)

The main drawback of the "pure" LMS algorithm is that it is sensitive to the scaling of its input $x(n)$. This makes it very hard (if not impossible) to choose a learning rate μ that guarantees stability of the algorithm (Haykin 2002). The *Normalised least mean squares filter* (NLMS) is a variant of the LMS algorithm that solves this problem by normalising with the power of the input. The NLMS algorithm can be summarised as:

Parameters: p = filter order

μ = step size

Initialization: $\hat{\mathbf{h}}(0) = 0$

Computation: For $n = 0,1,2,...$

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-p+1)]^T$$

$$e(n) = d(n) - \hat{\mathbf{h}}^H(n)\mathbf{x}(n)$$

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu\, e^*(n)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}$$

Optimal learning rate

It can be shown that if there is no interference ($v(n) = 0$), then the optimal learning rate for the NLMS algorithm is

$$\mu_{opt} = 1$$

and is independent of the input $x(n)$ and the real (unknown) impulse response $\mathbf{h}(n)$. In the general case with interference ($v(n) \neq 0$), the optimal learning rate is

$$\mu_{opt} = \frac{E\left[|y(n) - \hat{y}(n)|^2\right]}{E\left[|e(n)|^2\right]}$$

The results above assume that the signals $v(n)$ and $x(n)$ are uncorrelated to each other, which is generally the case in practice.

Proof

Let the filter misalignment be defined as $\Lambda(n) = \left|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\right|^2$, we can derive the expected misalignment for the next sample as:

$$E\left[\Lambda(n+1)\right] = E\left[\left|\hat{\mathbf{h}}(n) + \frac{\mu\, e^*(n)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)} - \mathbf{h}(n)\right|^2\right]$$

$$E\left[\Lambda(n+1)\right] = E\left[\left|\hat{\mathbf{h}}(n) + \frac{\mu\, \left(v^*(n) + y^*(n) - \hat{y}^*(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)} - \mathbf{h}(n)\right|^2\right]$$

Let $\delta = \hat{\mathbf{h}}(n) - \mathbf{h}(n)$ and $r(n) = \hat{y}(n) - y(n)$

$$E\left[\Lambda(n+1)\right] = E\left[\left|\delta(n) - \frac{\mu\, \left(v(n) + r(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}\right|^2\right]$$

$$E\left[\Lambda(n+1)\right] = E\left[\left(\delta(n) - \frac{\mu\, \left(v(n) + r(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}\right)^H \left(\delta(n) - \frac{\mu\, \left(v(n) + r(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}\right)\right]$$

Assuming independence, we have:

$$E\left[\Lambda(n+1)\right] = \Lambda(n) + E\left[\left(\frac{\mu\, \left(v(n) - r(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}\right)^H \left(\frac{\mu\, \left(v(n) - r(n)\right)\mathbf{x}(n)}{\mathbf{x}^H(n)\mathbf{x}(n)}\right)\right] - 2E\left[\frac{\mu|r(n)|^2}{\mathbf{x}^H(n)\mathbf{x}(n)}\right]$$

$$E\left[\Lambda(n+1)\right] = \Lambda(n) + \frac{\mu^2 E\left[|e(n)|^2\right]}{\mathbf{x}^H(n)\mathbf{x}(n)} - \frac{2\mu E\left[|r(n)|^2\right]}{\mathbf{x}^H(n)\mathbf{x}(n)}$$

The optimal learning rate is found at $\frac{dE\left[\Lambda(n+1)\right]}{d\mu} = 0$, which leads to:

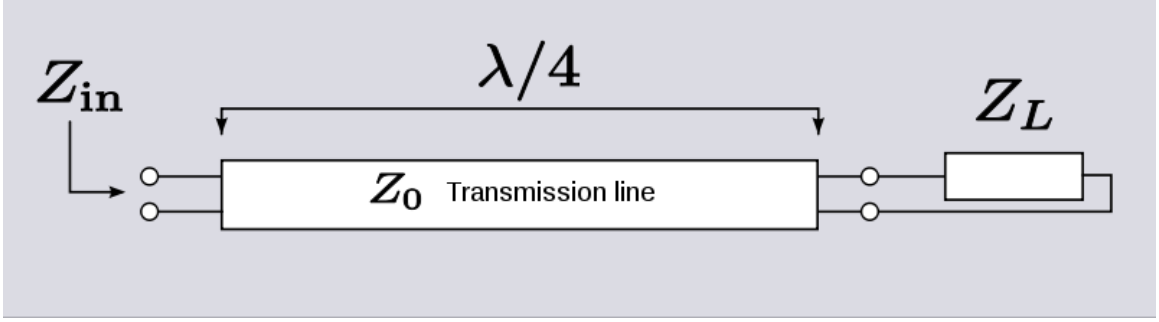
$$2\mu E\left[|e(n)|^2\right] - 2E\left[|r(n)|^2\right] = 0$$

$$\mu = \frac{E\left[|r(n)|^2\right]}{E\left[|e(n)|^2\right]}$$

Quarter wave impedance transformer

A **quarter wave impedance transformer**, often written as **λ/4 impedance transformer**, is a component used in electrical engineering consisting of a length of transmission line or waveguide exactly one quarter of a wavelength (λ) long and terminated in some known impedance. The device presents at its input the dual of the impedance with which it is terminated. It is a similar concept to a stub; but whereas a stub is terminated in a short (or open) circuit and the length designed to produce the required impedance, the λ/4 transformer is the other way around; it is a pre-determined length and the termination is

designed to produce the required impedance. The relationship between the characteristic, Z_0, input, Z_{in} and load, Z_L, impedances is;

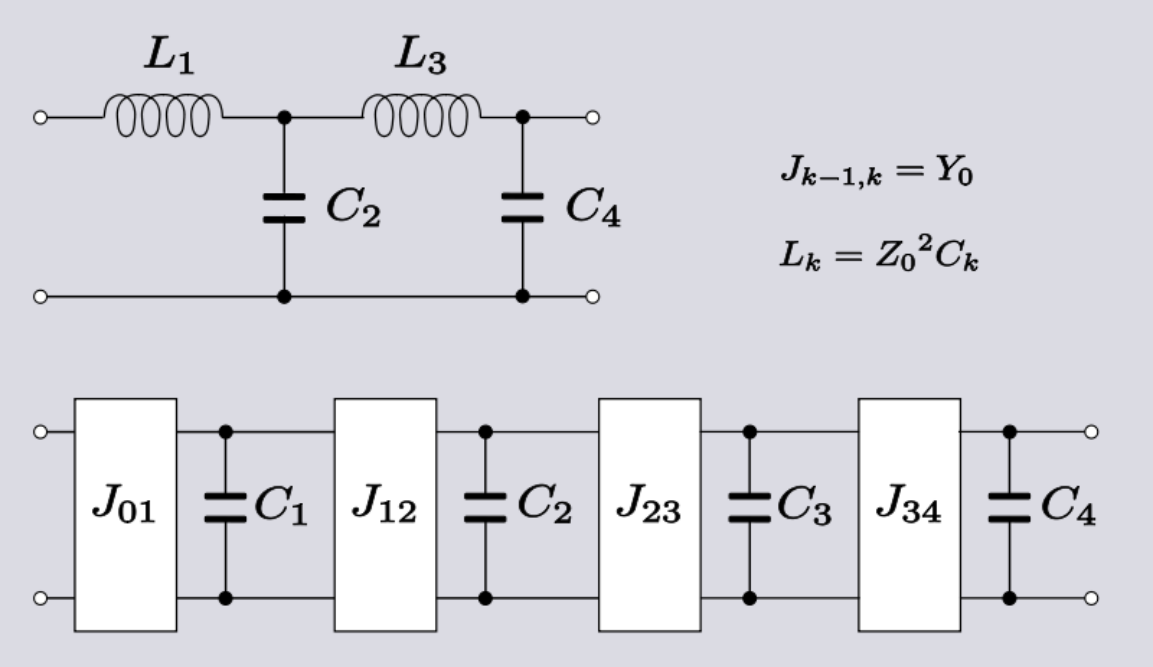


Using a transmission line as an impedance transformer.

$$\frac{Z_{\text{in}}}{Z_0} = \frac{Z_0}{Z_L}$$

Applications

At radio frequencies of upper VHF or higher up to microwave frequencies one quarter wavelength is conveniently short enough to incorporate the component within many products, but not so small that it cannot be manufactured using normal engineering tolerances, and it is at these frequencies where the device is most often encountered. It is especially useful for making an inductor out of a capacitor, since designers have a preference for the latter. Another application is when DC power needs to be fed into a transmission line, which may be necessary to power an active device connected to the line, such as a switching transistor or a varactor diode for instance. An ideal DC voltage source has zero impedance, that is, it presents a short circuit and it is not useful to connect a short circuit directly across the line. Feeding in the DC via a λ/4 transformer will transform the short circuit into an open circuit which has no effect on the signals on the line. Likewise, an open circuit can be transformed into a short circuit.



The lumped element low-pass filter (top) can be converted to a design that eliminates the inductors and contains capacitors only by the use of *J*-inverters, resulting in a mixed lumped element and distributed element design.

The device can be used as a component in a filter and in this application it is sometimes known as an inverter because it produces the mathematical inverse of an impedance. Impedance inverters are not to be confused with the more common meaning of inverter for a device that has the inverse function of a rectifier. Inverter is a general term for the class of circuits that have the function of inverting an impedance. There are many such circuits and the term does not necessarily imply a λ/4 transformer. The most common use for inverters is to convert a 2-element-kind *LC* filter design such as a ladder network into a one-element-kind filter. Equally, for bandpass filters, a two-resonator-kind (resonators and anti-resonators) filter can be converted to a one-resonator-kind. Inverters are classified as *K*-inverters or *J*-inverters depending on whether they are inverting a series impedance or a shunt admittance. Filters incorporating λ/4 inverters are only suitable for narrow band applications. This is because the impedance transformer line only has the correct electrical length of λ/4 at one specific frequency. The further the signal is from this frequency the less accurately the impedance transformer will be reproducing the impedance inverter function and the less accurately it will be representing the element values of the original lumped element filter design.

Theory of operation

A transmission line that is terminated in some impedance, Z_L, that is different from the characteristic impedance, Z_0, will result in a wave being reflected from the termination back to the source. At the input to the line the reflected voltage adds to the incident

voltage and the reflected current subtracts (because the wave is travelling in the opposite direction) from the incident current. The result is that the input impedance of the line (ratio of voltage to current) differs from the characteristic impedance and for a line of length *l* is given by;

$$Z_{\text{in}} = Z_0 \frac{Z_L + Z_0 \tanh(\gamma l)}{Z_0 + Z_L \tanh(\gamma l)}$$

where γ is the line propagation constant.

A very short transmission line, such as those being considered here, in many situations will have no appreciable loss along the length of the line and the propagation constant can be considered to be purely imaginary phase constant, $i\beta$ and the impedance expression reduces to,

$$Z_{\text{in}} = Z_0 \frac{Z_L + iZ_0 \tan(\beta l)}{Z_0 + iZ_L \tan(\beta l)}$$

Since β is the same as the angular wavenumber,

$$\beta = \frac{2\pi}{\lambda} \ ,$$

for a quarter wavelength line,

$$l = \frac{\lambda}{4} \ , \beta l = \frac{\pi}{2} \ ,$$

and the impedance becomes,

$$Z_{\text{in}} = Z_0 \frac{Z_L + iZ_0 \tan(\frac{\pi}{2})}{Z_0 + iZ_L \tan(\frac{\pi}{2})} = Z_0 \frac{iZ_0 \tan(\frac{\pi}{2})}{iZ_L \tan(\frac{\pi}{2})} = \frac{{Z_0}^2}{Z_L}$$

which is the same as the condition for dual impedances;

$$\frac{Z_{\text{in}}}{Z_0} = \frac{Z_0}{Z_L}$$

Chapter-13

Recursive Least Squares Filter and Ripple (electrical)

Recursive least squares filter

The **Recursive least squares (RLS)** adaptive filter is an algorithm which recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is in contrast to other algorithms such as the least mean squares (LMS) that aim to reduce the mean square error. In the derivation of the RLS, the input signals are considered deterministic, while for the LMS and similar algorithm they are considered stochastic. Compared to most of its competitors, the RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity, and potentially poor tracking performance when the filter to be estimated (the "true system") changes.

Motivation

In general, the RLS can be used to solve any problem that can be solved by adaptive filters. For example, suppose that a signal d(n) is transmitted over an echoey, noisy channel that causes it to be received as

$$x(n) = \sum_{k=0}^{q} b_n(k) d(n-k) + v(n+1)$$

where $v(n)$ represents additive noise. We will attempt to recover the desired signal $d(n)$ by use of a p-tap FIR filter, $\mathbf{w}$:

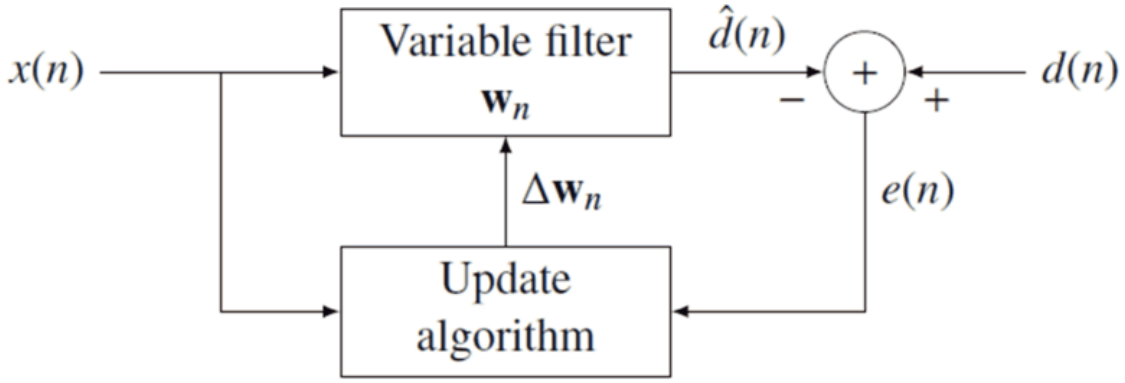
$$\hat{d}(n) = \sum_{k=0}^{p-1} w_n(k) x(n-k) = \mathbf{w}_n^T \mathbf{x}(n)$$

where $\mathbf{x}_n = [x(n) \quad x(n-1) \quad \dots \quad x(n-p+1)]^T$ is the vector containing the *p* most recent samples of *x*(*n*). Our goal is to estimate the parameters of the filter $\mathbf{w}$, and at each time *n* we refer to the new least squares estimate by $\mathbf{w_n}$. As time evolves, we would like to avoid completely redoing the least squares algorithm to find the new estimate for $\mathbf{w}_{n+1}$, in terms of $\mathbf{w}_n$.

The benefit of the RLS algorithm is that there is no need to invert matrices, thereby saving computational power. Another advantage is that it provides intuition behind such results as the Kalman filter.

Discussion

The idea behind RLS filters is to minimize a cost function *C* by appropriately selecting the filter coefficients $\mathbf{w}_n$, updating the filter as new data arrives. The error signal *e*(*n*) and desired signal *d*(*n*) are defined in the negative feedback diagram below:



The error implicitly depends on the filter coefficients through the estimate $\hat{d}(n)$:

$$e(n) = d(n) - \hat{d}(n)$$

The weighted least squares error function *C*—the cost function we desire to minimize—being a function of e(n) is therefore also dependent on the filter coefficients:

$$C(\mathbf{w_n}) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$$

where $0 < \lambda \leq 1$ is the "forgetting factor" which gives exponentially less weight to older error samples.

The cost function is minimized by taking the partial derivatives for all entries *k* of the coefficient vector $\mathbf{w}_n$ and setting the results to zero

$$\frac{\partial C(\mathbf{w}_n)}{\partial w_n(k)} = \sum_{i=0}^{n} 2\lambda^{n-i} e(i) \frac{\partial e(i)}{\partial w_n(k)} = \sum_{i=0}^{n} 2\lambda^{n-i} e(i)\, x(i-k) = 0$$

Next, replace *e*(*n*) with the definition of the error signal

$$\sum_{i=0}^{n} \lambda^{n-i} \left[d(i) - \sum_{l=0}^{p} w_n(l) x(i-l) \right] x(i-k) = 0$$

Rearranging the equation yields

$$\sum_{l=0}^{p} w_n(l) \left[\sum_{i=0}^{n} \lambda^{n-i}\, x(i-l) x(i-k) \right] = \sum_{i=0}^{n} \lambda^{n-i} d(i) x(i-k)$$

This form can be expressed in terms of matrices

$$\mathbf{R}_x(n)\, \mathbf{w}_n = \mathbf{r}_{dx}(n)$$

where $\mathbf{R}_x(n)$ is the weighted sample correlation matrix for *x*(*n*), and $\mathbf{r}_{dx}(n)$ is the equivalent estimate for the cross-correlation between *d*(*n*) and *x*(*n*). Based on this expression we find the coefficients which minimize the cost function as

$$\mathbf{w}_n = \mathbf{R}_x^{-1}(n)\, \mathbf{r}_{dx}(n)$$

This is the main result of the discussion.

Choosing λ

The smaller λ is, the smaller contribution of previous samples. This makes the filter *more* sensitive to recent samples, which means more fluctuations in the filter co-efficients. The λ = 1 case is referred to as the *growing window RLS algorithm*.

Recursive algorithm

The discussion resulted in a single equation to determine a coefficient vector which minimizes the cost function. Here we want to derive a recursive solution of the form

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \Delta \mathbf{w}_{n-1}$$

where $\Delta \mathbf{w}_{n-1}$ is a correction factor at time *n-1*. We start the derivation of the recursive algorithm by expressing the cross correlation $\mathbf{r}_{dx}(n)$ in terms of $\mathbf{r}_{dx}(n-1)$

$$\begin{aligned}\mathbf{r}_{dx}(n) &= \sum_{i=0}^{n} \lambda^{n-i} d(i)\mathbf{x}(i) \\ &= \sum_{i=0}^{n-1} \lambda^{n-i} d(i)\mathbf{x}(i) + \lambda^0 d(n)\mathbf{x}(n) \\ &= \lambda \mathbf{r}_{dx}(n-1) + d(n)\mathbf{x}(n)\end{aligned}$$

where $\mathbf{x}(i)$ is the *p+1* dimensional data vector

$$\mathbf{x}(i) = [x(i), x(i-1), \ldots, x(i-p)]^T$$

Similarly we express $\mathbf{R}_x(n)$ in terms of $\mathbf{R}_x(n-1)$ by

$$\begin{aligned}\mathbf{R}_x(n) &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \\ &= \lambda \mathbf{R}_x(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)\end{aligned}$$

In order to generate the coefficient vector we are interested in the inverse of the deterministic autocorrelation matrix. For that task the Woodbury matrix identity comes in handy. With

$A = \lambda \mathbf{R}_x(n-1)$ is *(p+1)*-by-*(p+1)*

$U = \mathbf{x}(n)$ is *(p+1)*-by-1

$V = \mathbf{x}^T(n)$ is 1-by-*(p+1)*

$C = I_1$ is the 1-by-1 identity matrix

The Woodbury matrix identity follows

$$\begin{aligned}\mathbf{R}_x^{-1}(n) &= \left[\lambda \mathbf{R}_x(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)\right]^{-1} \\ &= \lambda^{-1}\mathbf{R}_x^{-1}(n-1) \\ &\quad -\lambda^{-1}\mathbf{R}_x^{-1}(n-1)\mathbf{x}(n) \\ &\quad \left\{1 + \mathbf{x}^T(n)\lambda^{-1}\mathbf{R}_x^{-1}(n-1)\mathbf{x}(n)\right\}^{-1} \mathbf{x}^T(n)\lambda^{-1}\mathbf{R}_x^{-1}(n-1)\end{aligned}$$

To come in line with the standard literature, we define

$$\begin{aligned}\mathbf{P}(n) &= \mathbf{R}_x^{-1}(n) \\ &= \lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\end{aligned}$$

where the *gain vector* $g(n)$ is

$$\begin{aligned}\mathbf{g}(n) &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)\left\{1 + \mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)\right\}^{-1} \\ &= \mathbf{P}(n-1)\mathbf{x}(n)\left\{\lambda + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)\right\}^{-1}\end{aligned}$$

Before we move on, it is necessary to bring $\mathbf{g}(n)$ into another form

$$\begin{aligned}\mathbf{g}(n)\left\{1 + \mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)\right\} &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n) \\ \mathbf{g}(n) + \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n) &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)\end{aligned}$$

Subtracting the second term on the left side yields

$$\begin{aligned}\mathbf{g}(n) &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n) \\ &= \lambda^{-1}\left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right]\mathbf{x}(n)\end{aligned}$$

With the recursive definition of $\mathbf{P}(n)$ the desired form follows

$$\mathbf{g}(n) = \mathbf{P}(n)\mathbf{x}(n)$$

Now we are ready to complete the recursion. As discussed

$$\begin{aligned}\mathbf{w}_n &= \mathbf{P}(n)\,\mathbf{r}_{dx}(n) \\ &= \lambda\mathbf{P}(n)\,\mathbf{r}_{dx}(n-1) + d(n)\mathbf{P}(n)\,\mathbf{x}(n)\end{aligned}$$

The second step follows from the recursive definition of $\mathbf{r}_{dx}(n)$. Next we incorporate the recursive definition of $\mathbf{P}(n)$ together with the alternate form of $\mathbf{g}(n)$ and get

$$\begin{aligned}\mathbf{w}_n &= \lambda\left[\lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)\right]\mathbf{r}_{dx}(n-1) + d(n)\mathbf{g}(n) \\ &= \mathbf{P}(n-1)\mathbf{r}_{dx}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{r}_{dx}(n-1) + d(n)\mathbf{g}(n) \\ &= \mathbf{P}(n-1)\mathbf{r}_{dx}(n-1) + \mathbf{g}(n)\left[d(n) - \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{r}_{dx}(n-1)\right]\end{aligned}$$

With $\mathbf{w}_{n-1} = \mathbf{P}(n-1)\mathbf{r}_{dx}(n-1)$ we arrive at the update equation

$$\begin{aligned}\mathbf{w}_n &= \mathbf{w}_{n-1} + \mathbf{g}(n)\left[d(n) - \mathbf{x}^T(n)\mathbf{w}_{n-1}\right] \\ &= \mathbf{w}_{n-1} + \mathbf{g}(n)\alpha(n)\end{aligned}$$

where $\alpha(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}_{n-1}$ is the *a priori* error. Compare this with the *a posteriori* error; the error calculated *after* the filter is updated:

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}_n$$

That means we found the correction factor

$$\Delta\mathbf{w}_{n-1} = \mathbf{g}(n)\alpha(n)$$

This intuitively satisfying result indicates that the correction factor is directly proportional to both the error and the gain vector, which controls how much sensitivity is desired, through the weighting factor, λ.

RLS algorithm summary

The RLS algorithm for a *p*-th order RLS filter can be summarized as

Parameters: p = filter order
λ = forgetting factor
δ = value to initialize $\mathbf{P}(0)$

Initialization: $\mathbf{w}_n = 0$
$\mathbf{P}(0) = \delta^{-1}I$ where I is the (p + 1)-by-(p + 1) identity matrix

Computation: For $n = 0, 1, 2, \ldots$

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-p) \end{bmatrix}$$

$$\alpha(n) = d(n) - \mathbf{w}(n-1)^T\mathbf{x}(n)$$

$$\mathbf{g}(n) = \mathbf{P}(n-1)\mathbf{x}(n)\left\{\lambda + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)\right\}^{-1}$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha(n)\mathbf{g}(n)$$

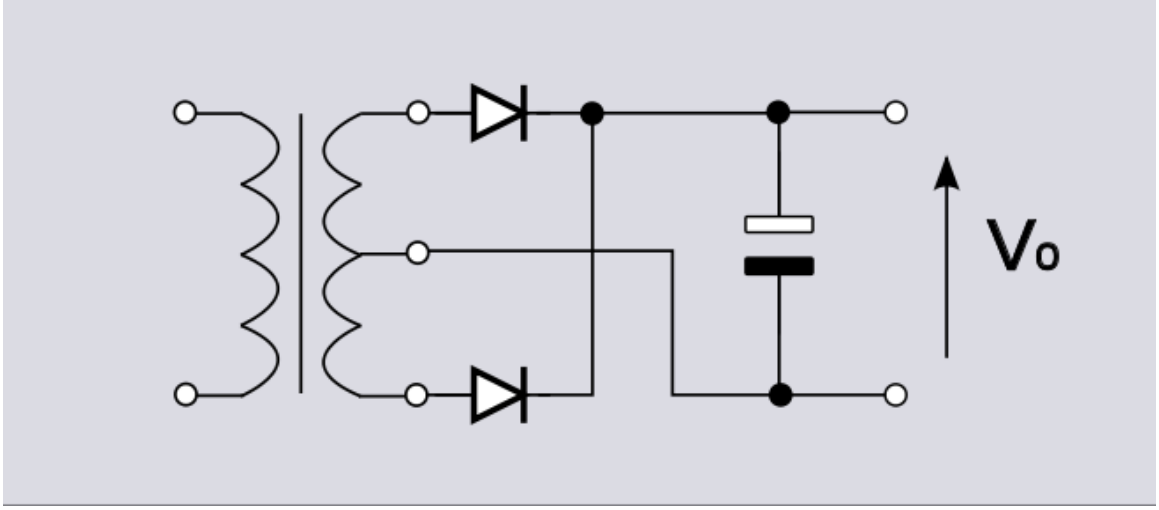
Note that the recursion for P follows a Riccati equation and thus draws parallels to the Kalman filter.

Ripple (electrical)

The most common meaning of **ripple** in electrical science, is the small unwanted residual periodic variation of the direct current (dc) output of a power supply which has been derived from an alternating current (ac) source. This ripple is due to incomplete suppression of the alternating waveform within the power supply.

As well as this time-varying phenomenon, there is a **frequency domain ripple** that arises in some classes of filter and other signal processing networks. In this case the periodic variation is a variation in the insertion loss of the network against increasing frequency. The variation may not be strictly linearly periodic. In this meaning also, ripple is usually to be considered an unwanted effect, its existence being a compromise between the amount of ripple and other design parameters.

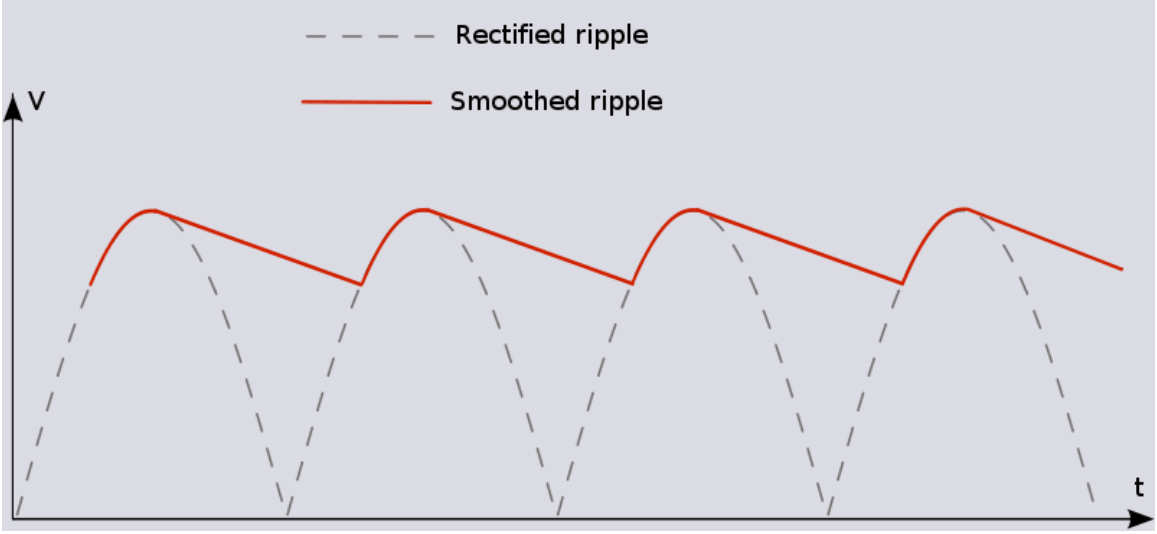
Time-domain ripple



Full-wave rectifier circuit with a reservoir capacitor on the output for the purpose of smoothing ripple

Ripple factor (γ) may be defined as the ratio of the root mean square (rms) value of the ripple voltage to the absolute value of the dc component of the output voltage, usually expressed as a percentage. However, ripple voltage is also commonly expressed as the peak-to-peak value. This is largely because peak-to-peak is both easier to measure on an oscilloscope and is simpler to calculate theoretically. Filter circuits intended for the reduction of ripple are usually called smoothing circuits.

The simplest scenario in ac to dc conversion is a rectifier without any smoothing circuitry at all. The ripple voltage is very large in this situation; the peak-to-peak ripple voltage is equal to the peak ac voltage. A more common arrangement is to allow the rectifier to work into a large smoothing capacitor which acts as a reservoir. After a peak in output voltage the capacitor (C) supplies the current to the load (R) and continues to do so until the capacitor voltage has fallen to the value of the now rising next half-cycle of rectified voltage. At that point the rectifiers turn on again and deliver current to the reservoir until peak voltage is again reached. If the time constant, CR, is large in comparison to the period of the ac waveform, then a reasonable accurate approximation can be made by assuming that the capacitor voltage falls linearly. A further useful assumption can be made if the ripple is small compared to the dc voltage. In this case the phase angle through which the rectifiers conduct will be small and it can be assumed that the capacitor is discharging all the way from one peak to the next with little loss of accuracy.



Ripple voltage from a full-wave rectifier, before and after the application of a smoothing capacitor

With the above assumptions the peak-to-peak ripple voltage can be calculated as:

For a full-wave rectifier:

$$V_{\mathrm{pp}} = \frac{I}{2fC}$$

For a half-wave rectification:

$$V_{\mathrm{pp}} = \frac{I}{fC}$$

where

- V_{pp} is the peak-to-peak ripple voltage
- I is the current in the circuit
- f is the frequency of the ac power
- C is the capacitance

For the rms value of the ripple voltage, the calculation is more involved as the shape of the ripple waveform has a bearing on the result. Assuming a sawtooth waveform is a similar assumption to the ones above and yields the result:

$$\gamma = \frac{1}{4\sqrt{3}fCR}$$

where

- γ is the ripple factor
- R is the resistance of the load

Another approach to reducing ripple is to use a series choke. A choke has a filtering action and consequently produces a smoother waveform with less high-order harmonics. Against this, the dc output is close to the average input voltage as opposed to the higher voltage with the reservoir capacitor which is close to the peak input voltage. With suitable approximations, the ripple factor is given by:

$$\gamma = \frac{0.236R}{\omega L}$$

where

- ω is the angular frequency $2\pi f$
- L is the inductance of the choke

More complex arrangements are possible; the filter can be an LC ladder rather than a simple choke or the filter and the reservoir capacitor can both be used to gain the benefits of both. The most commonly seen of these is a low-pass Π-filter consisting of a reservoir capacitor followed by a series choke followed by a further shunt capacitor However, use of chokes is deprecated in contemporary designs for economic reasons. A more common solution where good ripple rejection is required is to use a reservoir capacitor to reduce the ripple to something manageable and then pass through a voltage regulator circuit. The regulator circuit, as well as regulating the output, will incidentally filter out nearly all of the ripple as long as the minimum level of the ripple waveform does not go below the voltage being regulated to.

The majority of power supplies are now switched mode. The filtering requirements for such power supplies are much easier to meet due to the frequency of the ripple waveform

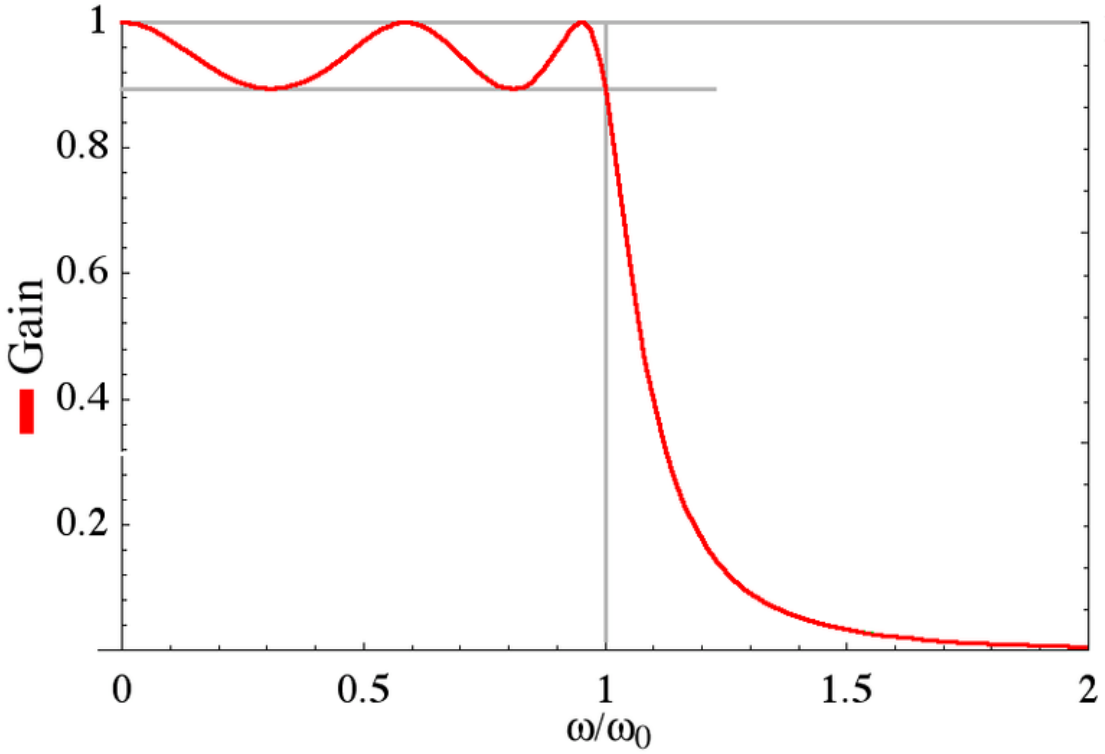
being very high. In traditional power supply designs the ripple frequency is either equal to (half-wave), or twice (full-wave) the ac line frequency. With switched mode power supplies the ripple frequency is not related to the line frequency, but is instead related to the frequency of the chopper circuit.

Effects of ripple

Ripple is undesirable in many electronic applications for a variety of reasons:

- The ripple frequency and its harmonics are within the audio band and will therefore be audible on equipment such as radio receivers, equipment for playing recordings and professional studio equipment.
- The ripple frequency is within television video bandwidth. Analogue TV receivers will exhibit a pattern of moving wavy lines if too much ripple is present.
- The presence of ripple can reduce the resolution of electronic test and measurement instruments. On an oscilloscope it will manifest itself as a visible pattern on screen.
- Within digital circuits, it reduces the threshold, as does any form of supply rail noise, at which logic circuits give incorrect outputs and data is corrupted.
- High amplitude ripple currents reduce the life of electrolytic capacitors.

Frequency-domain ripple



Ripple on a fifth order prototype Chebyshev filter

Ripple in the context of the frequency domain is referring to the periodic variation in insertion loss with frequency of a filter or some other two-port network. Not all filters exhibit ripple, some have monotonically increasing insertion loss with frequency such as the Butterworth filter. Common classes of filter which exhibit ripple are the Chebyshev filter, inverse Chebyshev filter and the Elliptical filter. The ripple is not usually strictly linearly periodic as can be seen from the example plot. Other examples of networks exhibiting ripple are impedance matching networks that have been designed using Chebyshev polynomials. The ripple of these networks, unlike regular filters, will never reach 0dB at minimum loss if designed for optimum transmission across the passband as a whole.

The amount of ripple can be traded for other parameters in the filter design. For instance, the rate of roll-off from the passband to the stopband can be increased at the expense of increasing the ripple without increasing the order of the filter (that is, the number of components has stayed the same). On the other hand, the ripple can be reduced by increasing the order of the filter while at the same time maintaining the same rate of roll-off.