8th exam item

András Mátyás Biricz

June 24, 2019

Abstract

Jelfeldolgozás és idősor-analízis – Fourier-módszerek, FFT, a spektrum és a spektrogram, az átviteli és ablakfüggvények, Wiener-szűrő. Korrelációs függvények, a Wiener–Hincsin-tétel és a teljesítményspektrum. Konvolúció és dekonvolúció. Szűrők analóg és digitális megvalósítása, RLC-körök, FIR- és IIR-szűrők.

Signal processing and analysis of time series - Fourier methods, FFT, spectrum and spectrogram, transfer- and window functions, Wiener-filter. Correlation functions, Wiener-Khinchin theorem, power spectrum. Convolution, deconvolution. Realization of analog and digital filters, RLC-circuit, FIR and IIR filters.

1   Introduction

Signal processing is a subfield of electrical engineering that concerns the analysis, synthesis, and modification of signals[1] in order to extract information of phenomena through processing of audio, video and the outcome of various measurements. Signal processing techniques can be used to improve signal transmission, emphasize or detect interesting components of measured signal and to improve storage efficiency.

As a consequence of the technological advancement tremendous amount of data can be measured in various fields of science and it becomes even bigger challenge to deal with that vast amount of data. Fortunately, the application of the signal processing methods can help.

2   Fourier methods

Fourier methods play a key role in analysing signals, especially in case of time series data. These methods involve the Fourier-series and Fourier transform. As a result we can acquire valuable information as a (frequency) spectrum. In addition, these can be applicable more broadly, even to analyse spatial properties of 2 or 3 dimensional signals or to solve differential equations. Last but not least, it comes in handy in calculation of correlation and convolution.

2.1   Fourier series

A Fourier series is an expansion of a periodic function $f(x)$ in terms of an infinite sum of sines and cosines. Fourier series make use of the orthogonality relationships of the sine and cosine functions. The computation and study of Fourier series is known as harmonic analysis and is extremely useful as a way to break up an arbitrary periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical.

In particular, since the superposition principle holds for solutions of a linear homogeneous ordinary differential equation, if such an equation can be solved in the case of a single sinusoid, the solution for an arbitrary function is immediately available by expressing the original function as a Fourier series and then plugging in the solution for each sinusoidal component. In some special cases where the Fourier series can be summed in closed form, this technique can even yield analytic solutions.

---

[1]https://en.wikipedia.org/wiki/Signal_processing.

In addition, any set of functions that form a complete orthogonal system has a corresponding generalized Fourier series analogous to the Fourier series. For example, using orthogonality of the roots of a Bessel function of the first kind gives a so-called Fourier-Bessel series.

The computation of the Fourier series is based on integral identities (follow source link to view). For a 2π periodic function $f(x)$, the series expansion is the following:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(nx) + \sum_{n=1}^{\infty} b_n sin(nx), \tag{1}$$

where

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\mathrm{d}x, \tag{2}$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)cos(nx)\mathrm{d}x, \tag{3}$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)sin(nx)\mathrm{d}x, \tag{4}$$

where a_0 is the coefficient of the constant term, a_n and b_n are the coefficients of the periodic terms and $n = 1, 2, 3, ...$

The notion of a Fourier series can also be extended to complex coefficients. In case of a real-valued function $f(x)$:

$$f(x) = \sum_{n=-\infty}^{\infty} A_n e^{inx}. \tag{5}$$

It can be showed (follow source), that:

$$A_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)e^{-inx}\mathrm{d}x, \tag{6}$$

which are coefficients hiding the a_0, a_n, b_n coefficients in a compact form.

Note, that not only 2π periodic functions can be expanded in a Fourier series, it can be generalized to the case of a function periodic in $[-L/2, L/2]$:

$$f(x) = \sum_{n=-\infty}^{\infty} A_n e^{i2\pi nx/L}, \tag{7}$$

$$A_n = \frac{1}{L} \int_{-L/2}^{L/2} f(x)e^{i2\pi nx/L}\mathrm{d}x. \tag{8}$$

The source and more information with examples can be found by following this link[2].

---

[2]http://mathworld.wolfram.com/FourierSeries.html.

2.2 Fourier transform

The Fourier transform is a generalization of the complex Fourier series in the limit as $L \to \infty$. Replace the discrete A_n with the continuous $F(k)dk$ while letting $n/L \to k$. Then change the sum to an integral, and the equations become:

$$f(x) = \int_{-\infty}^{\infty} F(k)e^{i2\pi kx}\mathrm{d}k, \tag{9}$$

$$f(k) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi kx}\mathrm{d}x. \tag{10}$$

Note that some authors (especially physicists) prefer to write the transform in terms of angular frequency $\omega \equiv 2\pi\nu$) instead of the oscillation frequency (ν). However, this destroys the symmetry, resulting in the transform pair:

$$H(\omega) = \mathcal{F}[h(t)] = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}\mathrm{d}t, \tag{11}$$

$$h(t) = \mathcal{F}^{-1}[H(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega)e^{i\omega t}\mathrm{d}\omega. \tag{12}$$

A function $f(x)$ has a forward and inverse Fourier transform if:

- $\int_{-\infty}^{\infty} |f(x)|\mathrm{d}x$ exists,

- there are a finite number of discontinuities,

- the function has bounded variation. (A sufficient weaker condition is fulfillment of the Lipschitz condition.[3])

The Fourier transform is linear, which means the following for functions $f(x)$ and $g(x)$:

$$\mathcal{F}[af(x) + bg(x)] = a\mathcal{F}[f(x)] + b\mathcal{F}[g(x)] = aF(k) + bG(k). \tag{13}$$

The source and more information with examples can be found by following this link[4].

2.3 Fast Fourier Transform

A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the complexity of computing the DFT from $\mathcal{O}(N^2)$, which arises if one simply applies the definition of DFT, to $\mathcal{O}(NlogN)$, where N is the data size. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In the presence of round-off error, many FFT algorithms are much more accurate than evaluating the DFT definition directly. There are many different

---

[3]http://mathworld.wolfram.com/LipschitzCondition.html,
[4]http://mathworld.wolfram.com/FourierTransform.html.

FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory.[5]

Let $x_0, ..., x_{N-1}$ be complex numbers. The DFT is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \quad k = 0, \ldots, N-1, \tag{14}$$

where $e^{i2\pi/N}$ is a primitive N^{th} root of 1.

Evaluating this definition directly requires $\mathcal{O}(N^2)$ operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in $\mathcal{O}(NlogN)$ operations. All known FFT algorithms require $\mathcal{O}(NlogN)$ operations, although there is no known proof that a lower complexity score is impossible.

To illustrate the savings of an FFT, consider the count of complex multiplications and additions for $N = 4096$ data points. Evaluating the DFT's sums directly involves N^2 complex multiplications and $N(N-1)$ complex additions, of which $\mathcal{O}(N)$ operations can be saved by eliminating trivial operations such as multiplications by 1. On the other hand, the Radix-2 (James W. Cooley and John W. Tukey (1965)) algorithm, for N a power of 2, can compute the same result with only $N/2 log_2 N$ complex multiplications (again, ignoring simplifications of multiplications by 1 and similar) and $N log_2 N$ complex additions. In practice, actual performance on modern computers is usually dominated by factors other than the speed of arithmetic operations and the analysis is a complicated subject, but the overall improvement from $\mathcal{O}(N^2)$ to $\mathcal{O}(NlogN)$ remains.

A Radix-2 decimation-in-time (DIT) FFT is the simplest and most common form of the Cooley–Tukey algorithm, although highly optimized Cooley–Tukey implementations typically use other forms of the algorithm as described below. Radix-2 DIT divides a DFT of size N into two interleaved DFTs (hence the name "Radix-2") of size $N/2$ with each recursive stage.

The discrete Fourier transform (DFT) is defined by the formula:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}, \tag{15}$$

where k is an integer ranging from 0 to $N-1$.

Radix-2 DIT first computes the DFTs of the even-indexed inputs $(x_{2m} = x_0, x_2, \ldots, x_{N-2})$ and of the odd-indexed inputs $(x_{2m+1} = x_1, x_3, \ldots, x_{N-1})$, and then combines those two results to produce the DFT of the whole sequence. This idea can then be performed recursively to reduce the overall runtime to $\mathcal{O}(NlogN)$. This simplified form assumes that N is a power of two, since the number of sample points N can usually be chosen freely by the application (e.g. by changing the sample rate or window, zero-padding, etcetera), this is often not an important restriction.

The Radix-2 DIT algorithm rearranges the DFT of the function x_n into two parts: a sum over the even-numbered indices $(n = 2m)$ and a sum over the odd-numbered indices $(n = 2m + 1)$:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k}. \tag{16}$$

One can factor a common multiplier $e^{-\frac{2\pi i}{N}k}$ out of the second sum, as shown in the equation below. It is then clear that the two sums are the DFT of the even-indexed part x_{2m} and the

---

[5]https://en.wikipedia.org/wiki/Fast_Fourier_transform.

DFT of odd-indexed part x_{2m+1} of the function x_n. Denote the DFT of the even-indexed inputs x_{2m} by E_k and the DFT of the odd-indexed inputs x_{2m+1} by O_k and we obtain:

$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of even-indexed part of } x_n} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}}_{\text{DFT of odd-indexed part of } x_n} = E_k + e^{-\frac{2\pi i}{N}k}O_k. \tag{17}$$

Thanks to the periodicity of the complex exponentials, $X_{k+\frac{N}{2}}$ is also obtained from E_k and O_k:

$$
\begin{aligned}
X_{k+\frac{N}{2}} &= \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}m(k+\frac{N}{2})} + e^{-\frac{2\pi i}{N}(k+\frac{N}{2})} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}m(k+\frac{N}{2})} \\
&= \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk} e^{-2\pi mi} + e^{-\frac{2\pi i}{N}k} e^{-\pi i} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk} e^{-2\pi mi} \\
&= \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk} - e^{-\frac{2\pi i}{N}k} \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk} \\
&= E_k - e^{-\frac{2\pi i}{N}k}O_k.
\end{aligned}
\tag{18}
$$

We can rewrite X_k as:

$$
\begin{aligned}
X_k &= E_k + e^{-\frac{2\pi i}{N}k}O_k, \\
X_{k+\frac{N}{2}} &= E_k - e^{-\frac{2\pi i}{N}k}O_k.
\end{aligned}
\tag{19}
$$

This result, expressing the DFT of length N recursively in terms of two DFTs of size $N/2$, is the core of the Radix-2 DIT fast Fourier transform. The algorithm gains its speed by re-using the results of intermediate computations to compute multiple DFT outputs. Note that final outputs are obtained by a $\pm$ combination of E_k and $O_k \exp(-2\pi i k/N)$, which is simply a size-2 DFT (sometimes called a butterfly diagram in this context (see figure 1)); when this is generalized to larger radices, the size-2 DFT is replaced by a larger DFT (which itself can be evaluated with an FFT).

The source and more information with examples can be found by following these links[6][7].

---

[6] https://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm,
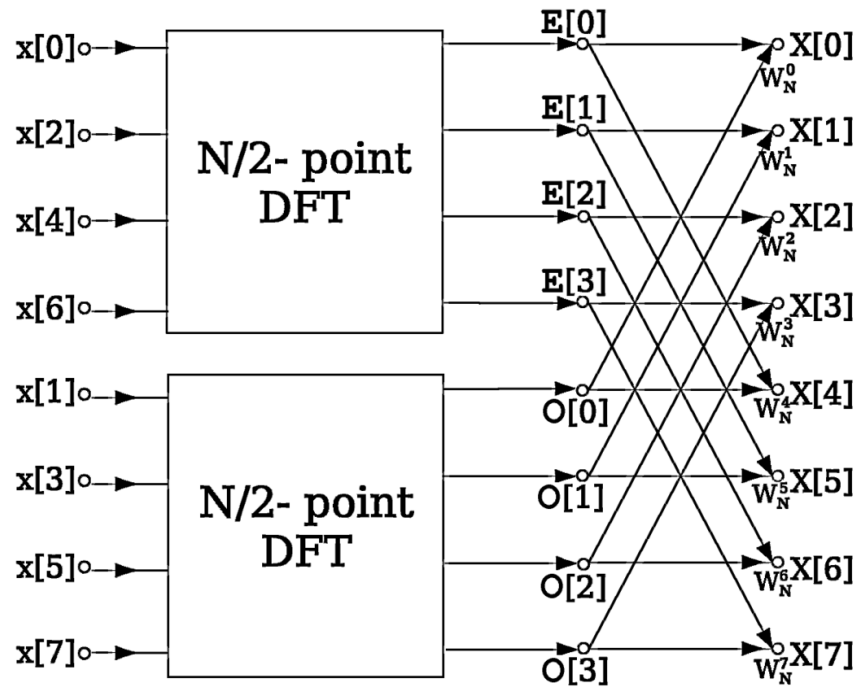[7] https://en.wikipedia.org/wiki/Fast_Fourier_transform.

Figure 1: Data flow diagram for $N = 8$: a decimation-in-time radix-2 FFT breaks a length-N DFT into two length-$N/2$ DFTs followed by a combining stage consisting of many size-2 DFTs called "butterfly" operations (so-called because of the shape of the data-flow diagrams). The source was accessed in June 2019 (https://en.wikipedia.org/wiki/File:DIT-FFT-butterfly.png).

2.4 Spectrum, spectrogram

The spectrum is the (frequency) domain representation of a signal. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time (see figures 2 and 3). When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data is represented in a 3D plot they may be called waterfalls. Spectrograms are used extensively in the fields of music, sonar, radar, and speech processing, seismology, and others. Spectrograms of audio can be used to identify spoken words phonetically, and to analyse the various calls of animals[8].
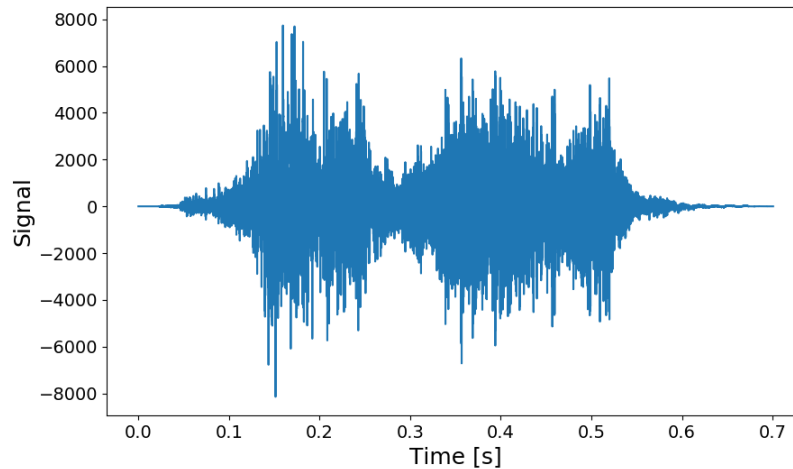

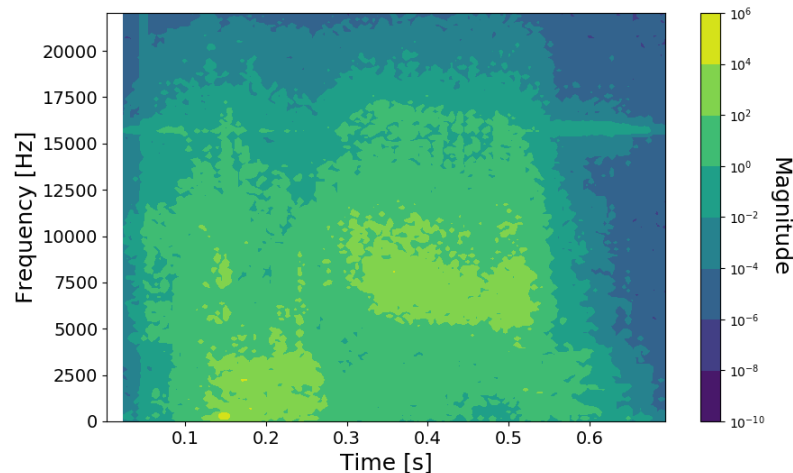Figure 2: Yes sound. Source: https://www.pacdv.com/sounds/voices/yes-2.wav.

Figure 3: Spectrogram of the yes sound.

To summarize, in this section the basics of the Fourier methods are explained. In the upcoming sections the wide usability and powerfullness of these methods will be presented.

---

[8]https://en.wikipedia.org/wiki/Spectrogram.

3 Convolution

The definition of convolution for two functions $f(x)$ and $g(x)$[9].

$$[f * g](x) = \int_\infty^\infty f(x')g(x - x')\mathrm{d}x', \tag{20}$$

The Fourier transform of convolutions of two functions $f(x)$ and $g(x)$ is defined as follows:

$$\mathcal{F}[f * g] = \int_\infty^\infty \int_\infty^\infty e^{-2\pi i k x} f(x')g(x - x')\mathrm{d}x'\mathrm{d}x, \tag{21}$$

which can be easily proved, that:

$$\mathcal{F}[f * g] = \mathcal{F}[f] \cdot \mathcal{F}[g], \tag{22}$$

or in words: the convolution of two functions becomes a simple multiplication in Fourier space. Since the Fourier transform can be calculated by an efficient numerical method, this will be a widely applicable, preferred technique in computer science.

To learn about the convolution and deconvolution, please read the 9th exam item.

4 Correlation functions

There is also a somewhat surprising and extremely important relationship between the **autocorrelation** and the Fourier transform known as the **Wiener-Khinchin theorem**[10]. Let $\mathcal{F}[f(x)](k) = F(k)$, and $\overline{f}$ denote the complex conjugate of f, then the Fourier transform of the absolute square of $F(k)$ can be given by[11]:

$$\mathcal{F}^{-1}[|F(k)|^2](x) = \int_\infty^\infty \overline{f}(\tau)f(\tau + x)\mathrm{d}\tau. \tag{23}$$

The Wiener-Khinchin theorem is a special case of the cross-correlation theorem with $f = g$. The **cross-correlation** can be defined as the following:

$$[f \bullet g](x) = \int_\infty^\infty f(x')g(x + x')\mathrm{d}x', \tag{24}$$

and can be also calculated as:

$$[f \bullet g](x) = \mathcal{F}^{-1}[\overline{F}(k)G(k)]. \tag{25}$$

5 Transfer and window functions

In engineering, a **transfer function** of an electronic or control system component is a mathematical function which theoretically models the device's output for each possible input[12]. In

---

[9]http://mathworld.wolfram.com/Convolution.html,

[10]http://mathworld.wolfram.com/Wiener-KhinchinTheorem.html,

[11]http://mathworld.wolfram.com/FourierTransform.html,

[12]https://en.wikipedia.org/wiki/Transfer_function.

signal processing, the impulse response, or impulse response function (IRF), of a dynamic system is its output when presented with a brief input signal, called an impulse. More generally, an impulse response is the reaction of any dynamic system in response to some external change. In both cases, the impulse response describes the reaction of the system as a function of time (or possibly as a function of some other independent variable that parameterizes the dynamic behavior of the system)[13]. For physical systems this process can be described by the Green's function[14].

In signal processing and statistics, a **window function** (also known as tapering function) is a mathematical function that is zero-valued outside of some chosen interval, normally symmetric around the middle of the interval, usually near a maximum in the middle, and usually tapering away from the middle. Mathematically, when another function or waveform/data-sequence is "multiplied" by a window function, the product is also zero-valued outside the interval: all that is left is the part where they overlap, the "view through the window". Equivalently, and in actual practice, the segment of data within the window is first isolated, and then only that data is multiplied by the window function values. Thus, tapering, not segmentation, is the main purpose of window functions. The reasons for examining segments of a longer function include detection of transient events and time-averaging of frequency spectra. The duration of the segments is determined in each application by requirements like time and frequency resolution. But that method also changes the frequency content of the signal by an effect called spectral leakage. Window functions allow us to distribute the leakage spectrally in different ways, according to the needs of the particular application. In typical applications, the window functions used are non-negative, smooth, "bell-shaped" curves. Rectangle, triangle, and other functions can also be used. For more information visit this page[15].

6   Filters

6.1   Analog signal

Analog signal processing is for signals that have not been digitized, as in legacy radio, telephone, radar, and television systems. This involves linear electronic circuits as well as non-linear ones. The former are, for instance, passive filters, active filters, additive mixers, integrators and delay lines. Non-linear circuits include compandors, multiplicators (frequency mixers and voltage-controlled amplifiers), voltage-controlled filters, voltage-controlled oscillators and phase-locked loops.

6.2   Digital signal

Digital signal processing is the processing of digitized discrete-time sampled signals. Processing is done by general-purpose computers or by digital circuits such as ASICs, field-programmable gate arrays or specialized digital signal processors (DSP chips). Typical arithmetical operations include fixed-point and floating-point, real-valued and complex-valued, multiplication and addition. Other typical operations supported by the hardware are circular buffers and lookup tables. Examples of algorithms are the Fast Fourier transform (FFT), finite impulse response (FIR) filter, Infinite impulse response (IIR) filter, and adaptive filters such as the Wiener filter.

---

[13]https://en.wikipedia.org/wiki/Impulse_response,
[14]http://www1.spms.ntu.edu.sg/~ydchong/teaching/10_greens_function.pdf,
[15]https://en.wikipedia.org/wiki/Window_function.

To read more about these, visit this site[16].

6.3 RLC circuit

An RLC circuit is an electrical circuit consisting of a resistor (R), an inductor (L), and a capacitor (C), connected in series or in parallel. The name of the circuit is derived from the letters that are used to denote the constituent components of this circuit, where the sequence of the components may vary from RLC.

The circuit forms a harmonic oscillator for current, and resonates in a similar way as an LC circuit. Introducing the resistor increases the decay of these oscillations, which is also known as damping. The resistor also reduces the peak resonant frequency. In ordinary conditions, some resistance is unavoidable even if a resistor is not specifically included as a component; an ideal, pure LC circuit exists only in the domain of superconductivity, a physical effect demonstrated to this point only at temperatures far below ambient temperatures found anywhere on the Earth's surface.

RLC circuits have many applications as oscillator circuits. Radio receivers and television sets use them for tuning to select a narrow frequency range from ambient radio waves. In this role, the circuit is often referred to as a tuned circuit. An RLC circuit can be used as a band-pass filter, band-stop filter, low-pass filter or high-pass filter. The tuning application, for instance, is an example of band-pass filtering. The RLC filter is described as a second-order circuit, meaning that any voltage or current in the circuit can be described by a second-order differential equation in circuit analysis.

An important property of this circuit is its ability to resonate at a specific frequency. Resonance occurs because energy is stored in two different ways: in an electric field as the capacitor is charged and in a magnetic field as current flows through the inductor. Energy can be transferred from one to the other within the circuit and this can be oscillatory. The resistance in an RLC circuit will "damp" the oscillation, diminishing it with time if there is no driving AC power source in the circuit.

The driven frequency may be called the undamped resonance frequency or undamped natural frequency and the peak frequency may be called the damped resonance frequency or the damped natural frequency. The reason for this terminology is that the driven resonance frequency in a series or parallel resonant circuit has the value:

$$\omega_0 = \frac{1}{\sqrt{LC}}. \tag{26}$$

where L is the property of the inductor and C is the capacity of the capacitor.

The resonance effect can be used for filtering, the rapid change in impedance near resonance can be used to pass or block signals close to the resonance frequency. Both b*and-pass* and *band-stop* filters can be constructed and some filter circuits are shown later in the article. A key parameter in filter design is bandwidth. The bandwidth is measured between the cutoff frequencies, most frequently defined as the frequencies at which the power passed through the circuit has fallen to half the value passed at resonance. There are two of these half-power frequencies, one above, and one below the resonance frequency:

$$\Delta\omega = \omega_2 - \omega_1, \tag{27}$$

---

[16]https://en.wikipedia.org/wiki/Signal_processing.

where $\Delta\omega$ is the bandwidth, ω_1 is the lower half-power frequency and ω_2 is the upper half-power frequency.

To get to know more about RLC circuits, please follow this link[17].

6.4 FIR

In signal processing, a finite impulse response (FIR) filter is a 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 may have internal feedback and may continue to respond indefinitely (usually decaying).

The impulse response (that is, the output in response to a Kronecker delta input) of an Nth-order discrete-time FIR filter lasts exactly N + 1 samples (from first nonzero element through last nonzero element) before it then settles to zero.

FIR filters can be discrete-time or continuous-time, and digital or analog[18].

6.5 IIR

Infinite impulse response (IIR) is a property applying to many linear time-invariant systems. Common examples of linear time-invariant systems are most electronic and digital filters. Systems with this property are known as IIR systems or IIR filters, and are distinguished by having an impulse response which does not become exactly zero past a certain point, but continues indefinitely. This is in contrast to a finite impulse response (FIR) in which the impulse response h(t) does become exactly zero at times t > T for some finite T, thus being of finite duration.

In practice, the impulse response, even of IIR systems, usually approaches zero and can be neglected past a certain point. However the physical systems which give rise to IIR or FIR responses are dissimilar, and therein lies the importance of the distinction. For instance, analog electronic filters composed of resistors, capacitors, and/or inductors (and perhaps linear amplifiers) are generally IIR filters. On the other hand, discrete-time filters (usually digital filters) based on a tapped delay line employing no feedback are necessarily FIR filters. The capacitors (or inductors) in the analog filter have a "memory" and their internal state never completely relaxes following an impulse (assuming the classical model of capacitors and inductors where quantum effects are ignored). But in the latter case, after an impulse has reached the end of the tapped delay line, the system has no further memory of that impulse and has returned to its initial state; its impulse response beyond that point is exactly zero[19].

6.6 Wiener-filter

In signal processing, the Wiener filter is a filter used to produce an estimate of a desired or target random process by linear time-invariant (LTI) filtering of an observed noisy process, assuming known stationary signal and noise spectra, and additive noise. The Wiener filter minimizes the mean square error between the estimated random process and the desired process.

The goal of the Wiener filter is to compute a statistical estimate of an unknown signal using a related signal as an input and filtering that known signal to produce the estimate as an output. For example, the known signal might consist of an unknown signal of interest that has been

---

[17]https://en.wikipedia.org/wiki/RLC_circuit,
[18]https://en.wikipedia.org/wiki/Finite_impulse_response,
[19]https://en.wikipedia.org/wiki/Infinite_impulse_response.

corrupted by additive noise. The Wiener filter can be used to filter out the noise from the corrupted signal to provide an estimate of the underlying signal of interest. The Wiener filter is based on a statistical approach, and a more statistical account of the theory is given in the minimum mean square error (MMSE) estimator article.

Typical deterministic filters are designed for a desired frequency response. However, the design of the Wiener filter takes a different approach. One is assumed to have knowledge of the spectral properties of the original signal and the noise, and one seeks the linear time-invariant filter whose output would come as close to the original signal as possible. Wiener filters are characterized by the following points.

- Assumption: signal and (additive) noise are stationary linear stochastic processes with known spectral characteristics or known autocorrelation and cross-correlation.

- Requirement: the filter must be physically realizable/causal (this requirement can be dropped, resulting in a non-causal solution).

- Performance criterion: minimum mean-square error (MMSE)

This filter is frequently used in the process of deconvolution. For more information follow this link[20].

7 Conclusion

In this exam item the basics of signal processing can be learnt: the underlying, widely applicable Fourier methods, then the convolution, correlation functions and finally various types of filters.

---

[20]https://en.wikipedia.org/wiki/Wiener_filter.

References

[1] James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series *Math. Comp.* **19** (1965), 297-301. https://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf