Ve 216: Introduction to Signals and Systems

Yong Long

The University of Michigan- Shanghai Jiao Tong University Joint Institute
Shanghai Jiao Tong University

April 4, 2018

Based on Lecture Notes by Prof. Jeffrey A. Fessler

Outline

Outline

1. 7. Sampling
   - Introduction
   - FT of impulse-train sampled signals (7.1)
   - Sampling Theorem
   - Aliasing (7.3)
   - Reconstruction via interpolation (7.2)
   - Realistic non-impulse sampling (7.1.2)
   - Discrete-time Fourier transform (DTFT) (5.1)
   - Summary

Applications of the FT

- In chapter 4 we covered all of the fundamental mathematical properties of the FT which are used in EE.
- These properties are not just "interesting math;" they are the theoretical foundation of how just about everything involving signals work, from AM radios to digital TVs to PC sound cards etc.
- Now (at last!) we can begin to use the FT tools to understand the basic principles behind some of these applications.

Applications of the FT

- In chapter 4 we covered all of the fundamental mathematical properties of the FT which are used in EE.
- These properties are not just "interesting math;" they are the theoretical foundation of how just about everything involving signals work, from AM radios to digital TVs to PC sound cards etc.
- Now (at last!) we can begin to use the FT tools to understand the basic principles behind some of these applications.

Applications of the FT

- In chapter 4 we covered all of the fundamental mathematical properties of the FT which are used in EE.
- These properties are not just “interesting math;” they are the theoretical foundation of how just about everything involving signals work, from AM radios to digital TVs to PC sound cards etc.
- Now (at last!) we can begin to use the FT tools to understand the basic principles behind some of these applications.

Overview

- Filtering (used universally)
  convolution property
- Sampling (A/D converters in sound cards)
  FT of sampled signals
- Modulation (AM radio, digital comm (modems))
  modulation property
- ...

Sampling theorem

skip : 7.4, 7.5

- In fact, whereas the preceding mathematics is circa 1800 due to Fourier.
- The **Nyquist-Shannon sampling theorem** was a breakthrough in the 20th century, discovered by Nyquist and Shannon (a UM alumnus who is the father of information theory!).
  They proved that a signal can be perfectly reconstructed from sampling if the signal's highest frequency is half (or less) of the sampling rate.

Sampling theorem

skip : 7.4, 7.5

- In fact, whereas the preceding mathematics is circa 1800 due to Fourier.
- The **Nyquist-Shannon sampling theorem** was a breakthrough in the 20th century, discovered by Nyquist and Shannon (a UM alumnus who is the father of information theory!).

  They proved that a signal can be perfectly reconstructed from sampling if the signal's highest frequency is half (or less) of the sampling rate.

Sampling theorem

skip : 7.4, 7.5

- In fact, whereas the preceding mathematics is circa 1800 due to Fourier.
- The **Nyquist-Shannon sampling theorem** was a breakthrough in the 20th century, discovered by Nyquist and Shannon (a UM alumnus who is the father of information theory!).
  They proved that a signal can be perfectly reconstructed from sampling if the signal's highest frequency is half (or less) of the sampling rate.

Compressive sensing

The **compressive sensing theorem** (also known as compressed sensing, compressive sampling, or sparse sampling) was proposed around 2004 by Candes, Tao and Donoho.

They proved that given knowledge about a signal's sparsity ***(Picture)***, the signal may be reconstructed with even fewer samples than the Nyquist-Shannon theorem requires.

Compressive sensing

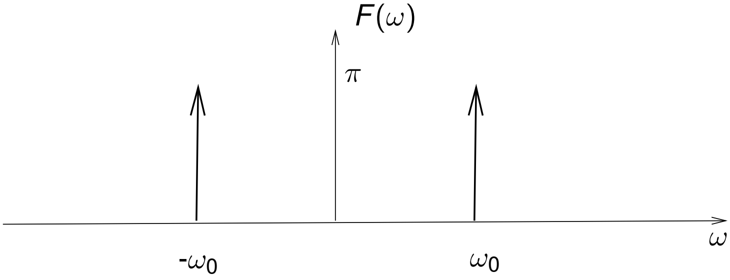
The **compressive sensing theorem** (also known as compressed sensing, compressive sampling, or sparse sampling) was proposed around 2004 by Candes, Tao and Donoho.

They proved that given knowledge about a signal's sparsity ***(Picture)***, the signal may be reconstructed with even fewer samples than the Nyquist-Shannon theorem requires.

Sparsity

$\cos(\omega_0 t)$ is sparse in the frequency domain.

$$\cos(\omega_0 t) \xleftrightarrow{\mathcal{F}} \pi\delta(\omega - \omega_0) + \pi\delta(\omega + \omega_0)$$



”single-pixel camera”

TED 2013- The Single Pixel Camera
`https://www.youtube.com/watch?v=y-jIzuHBJTo`

- The new digital image/video camera directly acquires random projections of a scene without first collecting the pixels/voxels.
- The camera architecture employs a digital micromirror array to optically calculate linear projections of the scene onto pseudorandom binary patterns.
- Its key hallmark is its ability to obtain an image or video with a single detection element (the ”single pixel”) while measuring the scene fewer times than the number of pixels/voxels.

Review of utility of unit impulse functions (1)

The simplification provided by the Dirac delta function has helped us in analyzing LTI systems like filters. It will also help us analyze sampling, even though again it is an idealization of real sampling circuits.

Property

- ***Sifting property*** *holds when $x(t)$ is continuous at t_0:*

$$\int_{-\infty}^{\infty} x(t)\delta(t-t_0)\,dt = x(t_0).$$

- ***sampling property*** *holds when $x(t)$ is continuous at t_0:*

$$x(t)\delta(t-t_0) = x(t_0)\delta(t-t_0).$$

Review of utility of unit impulse functions (1)

The simplification provided by the Dirac delta function has helped us in analyzing LTI systems like filters. It will also help us analyze sampling, even though again it is an idealization of real sampling circuits.

Property

- ***Sifting property*** *holds when $x(t)$ is continuous at t_0:*

$$\int_{-\infty}^{\infty} x(t)\delta(t - t_0)\, dt = x(t_0).$$

- ***sampling property*** *holds when $x(t)$ is continuous at t_0:*

$$x(t)\delta(t - t_0) = x(t_0)\delta(t - t_0).$$

Review of utility of unit impulse functions (2)

Property

1. *unit area property* $\int_{-\infty}^{\infty} \delta(t - t_0)\, dt = 1$ *for any* t_0
2. *scaling property* $\delta(at + b) = \frac{1}{|a|}\delta(t + b/a)$ *for* $a \neq 0$.
3. *symmetry property* $\delta(t) = \delta(-t)$
4. *support property* $\delta(t - t_0) = 0$ *for* $t \neq t_0$
5. *relationships with unit step function:* $\delta(t) = \frac{\mathrm{d}}{\mathrm{d}t}u(t)$, $u(t) = \int_{-\infty}^{t} \delta(\tau)\, d\tau$

Review of utility of unit impulse functions (3)

Property

1. $x(t) * \delta(t) = x(t)$
2. $x(t) * \delta(t - t_0) = x(t - t_0)$ ***delay property!***
3. $\delta(t - t_0) * \delta(t - t_1) = \delta(t - t_0 - t_1)$
4. $\delta(t) \rightarrow \boxed{LTI} \rightarrow h(t)$
5. *If* $y(t) = x(t) * h(t)$, *then* $x(t - t_0) * h(t - t_1) = y(t - t_0 - t_1)$.

Overview of DSP

$$x(t) \rightarrow \boxed{\text{A/D}} \rightarrow \boxed{x[n] \rightarrow \boxed{\text{DSP}} \rightarrow y[n]} \rightarrow \boxed{\text{D/A}} \rightarrow y(t)$$

The above configuration is frequently used, but not the only option.

- Often the output of the DSP is "information" rather than a signal, such as in a speech recognition system, or a radar target tracking system.
- Often (discrete-time) signals are generated digitally (like in MATLAB assignments), and then the `sound` or `image` command converts the digital representation into a continuous-time signal (audio or video) using the computers peripherals (sound card or video monitor).

Overview of DSP

$$x(t) \rightarrow \boxed{\text{A/D}} \rightarrow \boxed{x[n] \rightarrow \boxed{\text{DSP}} \rightarrow y[n]} \rightarrow \boxed{\text{D/A}} \rightarrow y(t)$$

The above configuration is frequently used, but not the only option.

- Often the output of the DSP is "information" rather than a signal, such as in a speech recognition system, or a radar target tracking system.
- Often (discrete-time) signals are generated digitally (like in MATLAB assignments), and then the `sound` or `image` command converts the digital representation into a continuous-time signal (audio or video) using the computers peripherals (sound card or video monitor).

Overview of DSP

$$x(t) \rightarrow \boxed{\text{A/D}} \rightarrow \boxed{x[n] \rightarrow \boxed{\text{DSP}} \rightarrow y[n]} \rightarrow \boxed{\text{D/A}} \rightarrow y(t)$$

The above configuration is frequently used, but not the only option.

- Often the output of the DSP is "information" rather than a signal, such as in a speech recognition system, or a radar target tracking system.
- Often (discrete-time) signals are generated digitally (like in MATLAB assignments), and then the `sound` or `image` command converts the digital representation into a continuous-time signal (audio or video) using the computers peripherals (sound card or video monitor).

Why DSP?

Question

Why DSP?

Why DSP?

Question

Why DSP?

- *DSP is everywhere due in large part to the dramatic development of digital technology over the past few decades. It hardly needs a motivating introduction these days: modems, cell phones, computer sound cards, digital video.*
- *It is inexpensive, lightweight, programmable and easily reproducible.*

Sampling Analog Signals

- A computer cannot store every value of a continuous-time signal $x(t)$.
- At most a computer can store a finite array of values, such as the values of the signal $x(t)$ at a finite collection of time instants.
- A sampler converts a CT signal into a discrete-time signal.

Sampling Analog Signals

- A computer cannot store every value of a continuous-time signal $x(t)$.
- At most a computer can store a finite array of values, such as the values of the signal $x(t)$ at a finite collection of time instants.
- A sampler converts a CT signal into a discrete-time signal.

Sampling Analog Signals

- A computer cannot store every value of a continuous-time signal $x(t)$.
- At most a computer can store a finite array of values, such as the values of the signal $x(t)$ at a finite collection of time instants.
- A sampler converts a CT signal into a discrete-time signal.

Periodic sampling

Definition

Ideal **periodic sampling** or **uniform sampling** is defined by

$$x[n] = x(nT_s),\ \ n = 0, \pm 1, \pm 2, \ldots$$

- T_s is the **sampling period** or **sampling interval**.
- $\omega_s/2\pi = 1/T_s$ is called the **sampling rate** or the **sampling frequency**, *e.g.* 44.1kHz for audio CD

Note that $x[n]$ is not a CT signal!

Periodic sampling

Definition

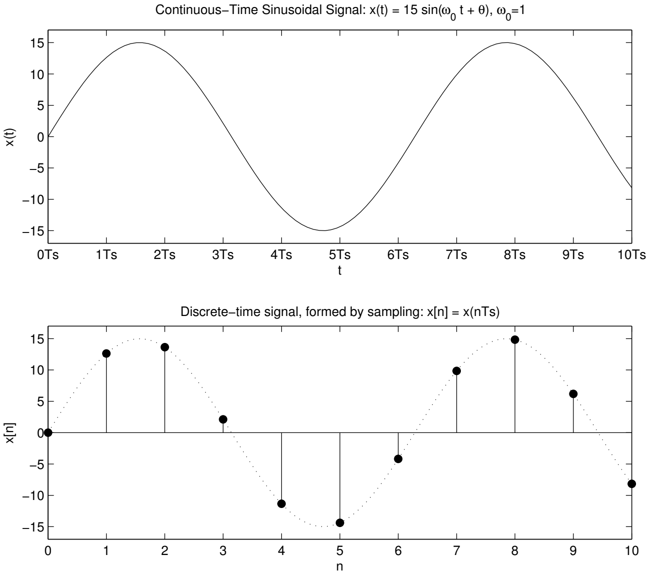
Ideal **periodic sampling** or **uniform sampling** is defined by

$$x[n] = x(nT_s), \;\; n = 0, \pm 1, \pm 2, \ldots$$

- T_s is the **sampling period** or **sampling interval**.
- $\omega_s / 2\pi = 1 / T_s$ is called the **sampling rate** or the **sampling frequency**, *e.g.* 44.1kHz for audio CD

Note that $x[n]$ is not a CT signal!

CT signal and sampled values



Outline

Impulse functions

Since we are dealing with CT signals in this class, it is convenient to create a CT signal from the samples, rather than just using $x[n]$.

Question

How to represent sampling in the time domain?

We will do this with impulse functions.

Impulse functions

Since we are dealing with CT signals in this class, it is convenient to create a CT signal from the samples, rather than just using $x[n]$.

Question

How to represent sampling in the time domain?

We will do this with impulse functions.

Impulse functions

Since we are dealing with CT signals in this class, it is convenient to create a CT signal from the samples, rather than just using $x[n]$.

Question

How to represent sampling in the time domain?

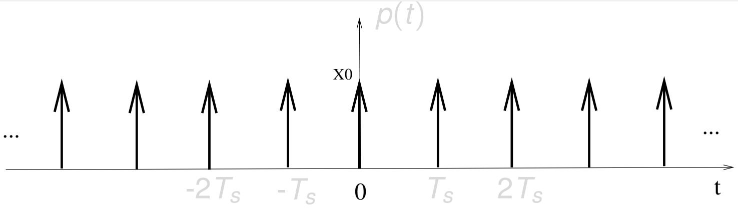
We will do this with impulse functions.

Ideal sampling function

Definition

The impulse train is sometimes called the **Dirac comb** signal or **ideal sampling function**:

$$p(t) \triangleq \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

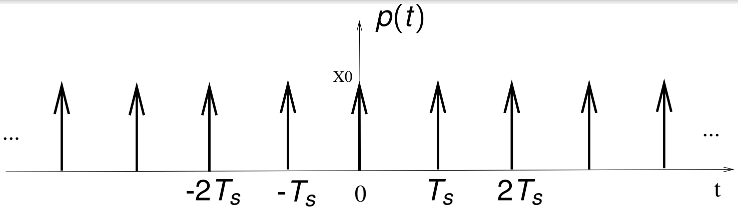


Ideal sampling function

Definition

The impulse train is sometimes called the **Dirac comb** signal or **ideal sampling function**:

$$p(t) \triangleq \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

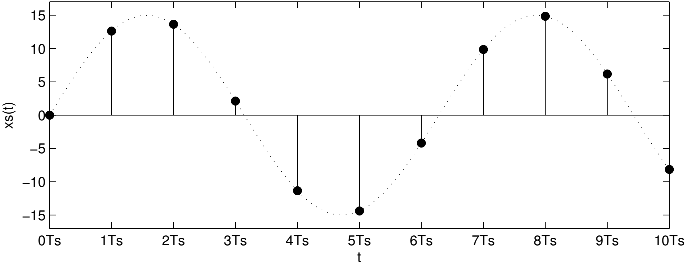


Impulse-train sampling

Suppose we have a CT signal $x(t)$ and we imagine multiplying it by $p(t)$, to form a new "sampled" signal

$$x_s(t) = x(t)p(t) = x(t)\sum_{n=-\infty}^{\infty}\delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s).$$

Note that $x_s(t)$ depends only on the original values of $x(t)$ at times nT_s. (This is why we call the above property the **sampling property** of the impulse function.)

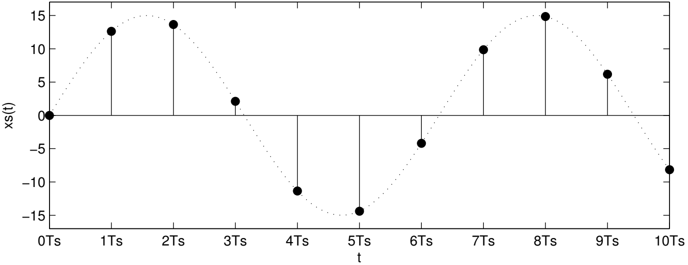


Impulse-train sampling

Suppose we have a CT signal $x(t)$ and we imagine multiplying it by $p(t)$, to form a new "sampled" signal

$$x_s(t) = x(t)p(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s).$$

Note that $x_s(t)$ depends only on the original values of $x(t)$ at times nT_s. (This is why we call the above property the sampling property of the impulse function.)

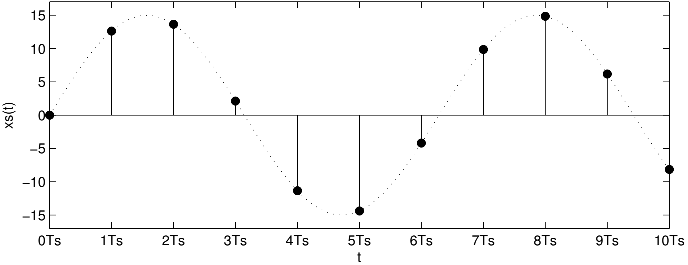


Impulse-train sampling

Suppose we have a CT signal $x(t)$ and we imagine multiplying it by $p(t)$, to form a new "sampled" signal

$$x_s(t) = x(t)p(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s).$$

Note that $x_s(t)$ depends only on the original values of $x(t)$ at times nT_s. (This is why we call the above property the **sampling property** of the impulse function.)



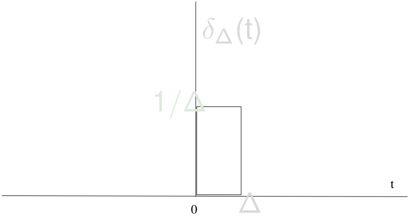
Practical sampling

Note: real systems (*e.g.* A/D converters) do not actually use impulse functions to sample a signal. This is a convenient mathematical idealization of a sampling circuit.

$$x(t) \rightarrow \boxed{\text{switch: rect}(t/\Delta - 1/2)} \rightarrow \boxed{\text{amplifier } 1/\Delta} \rightarrow x_s(t)$$

Practical impulse function

$$\delta(t) = \lim_{\Delta \to 0} \delta_\Delta(t) = \lim_{\Delta \to 0} \frac{1}{\Delta} \operatorname{rect}\left(\frac{t}{\Delta} - \frac{1}{2}\right)$$



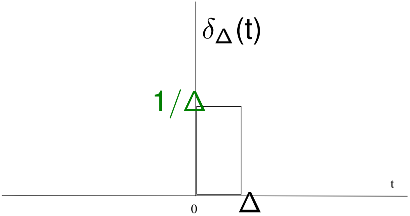
Practical sampling

Note: real systems (*e.g.* A/D converters) do not actually use impulse functions to sample a signal. This is a convenient mathematical idealization of a sampling circuit.

$$x(t) \rightarrow \boxed{\text{switch: } \mathrm{rect}(t/\Delta - 1/2)} \rightarrow \boxed{\text{amplifier } 1/\Delta} \rightarrow x_s(t)$$

Practical impulse function

$$\delta(t) = \lim_{\Delta \to 0} \delta_\Delta(t) = \lim_{\Delta \to 0} \frac{1}{\Delta} \mathrm{rect}\left(\frac{t}{\Delta} - \frac{1}{2}\right)$$



Spectrum of the sampled signal (1)

Question

But how does the spectrum of the sampled signal $x_s(t)$ relate to the spectrum of the original signal $x(t)$? In other words, what happens in the frequency domain when we sample a CT signal?

This is extremely important for understanding how digital filtering works!

Spectrum of the sampled signal (1)

Question

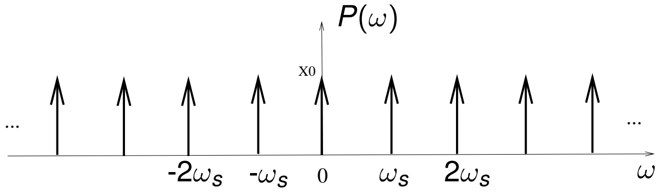
But how does the spectrum of the sampled signal $x_s(t)$ relate to the spectrum of the original signal $x(t)$? In other words, what happens in the frequency domain when we sample a CT signal?

This is extremely important for understanding how digital filtering works!

FT of impulse train

Recall FT of periodic signals (Chap. 3, p. 125)

$$\boxed{p(t) \overset{\mathcal{F}}{\longleftrightarrow} P(\omega) = \sum_{k=-\infty}^{\infty} \frac{2\pi}{T_s}\delta(\omega - k\omega_s)}$$



Spectrum of the sampled signal (2)

By time-domain multiplication property:

$$
\begin{aligned}
X_s(\omega) &= \frac{1}{2\pi} X(\omega) * P(\omega) = \frac{1}{2\pi} X(\omega) * \sum_{k=-\infty}^{\infty} \frac{2\pi}{T_s} \delta(\omega - k\omega_s) \\
&= \boxed{\frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)}
\end{aligned}
$$

which is a sum of shifted replicates of the spectrum.
(Picture)(MIT, Lecture 16-2)

Spectrum of the sampled signal (2)

Sampling in the time domain causes replication in the frequency domain.

Question

What is the dual of this statement?

Spectrum of the sampled signal (2)

Sampling in the time domain causes replication in the frequency domain.

Question

What is the dual of this statement?

Spectrum of the sampled signal (2)

Sampling in the time domain causes replication in the frequency domain.

Question

What is the dual of this statement?

Signals whose spectra are "sampled", i.e. line spectra, are periodic in the time domain.

Example (1)

Example

Consider the signal $x(t) = \text{sinc}^2(t)$. What happens when we sample it?

We have seen earlier that

$$x(t) = \text{sinc}^2(t) \xleftrightarrow{\mathcal{F}} X(\omega) = \text{rect}\left(\frac{\omega}{2\pi}\right) * \text{rect}\left(\frac{\omega}{2\pi}\right) = \text{tri}\left(\frac{\omega}{2\pi}\right)$$

(Picture)(MIT, Lecture 16.2)

Definition

A **bandlimited signal** is one whose spectrum is nonzero only over finite interval.

In this case $-\omega_{\max}$ to $\omega_{\max}$, where $\omega_{\max} = \pi$.

Example (1)

Example

Consider the signal $x(t) = \text{sinc}^2(t)$. What happens when we sample it?

We have seen earlier that

$$x(t) = \text{sinc}^2(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\omega) = \text{rect}\left(\frac{\omega}{2\pi}\right) * \text{rect}\left(\frac{\omega}{2\pi}\right) = \text{tri}\left(\frac{\omega}{2\pi}\right)$$

(Picture)(MIT, Lecture 16.2)

Definition

A **bandlimited signal** is one whose spectrum is nonzero only over finite interval.

In this case $-\omega_{\max}$ to $\omega_{\max}$, where $\omega_{\max} = \pi$.

Example (1)

Example

Consider the signal $x(t) = \mathrm{sinc}^2(t)$. What happens when we sample it?

We have seen earlier that

$$x(t) = \mathrm{sinc}^2(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\omega) = \mathrm{rect}\left(\frac{\omega}{2\pi}\right) * \mathrm{rect}\left(\frac{\omega}{2\pi}\right) = \mathrm{tri}\left(\frac{\omega}{2\pi}\right)$$

(Picture)(MIT, Lecture 16.2)

Definition

A **bandlimited signal** is one whose spectrum is nonzero only over finite interval.

In this case $-\omega_{\max}$ to $\omega_{\max}$, where $\omega_{\max} = \pi$.

Example (1)

Example

Consider the signal $x(t) = \text{sinc}^2(t)$. What happens when we sample it?

We have seen earlier that

$$x(t) = \text{sinc}^2(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\omega) = \text{rect}\left(\frac{\omega}{2\pi}\right) * \text{rect}\left(\frac{\omega}{2\pi}\right) = \text{tri}\left(\frac{\omega}{2\pi}\right)$$

(Picture)(MIT, Lecture 16.2)

Definition

A **bandlimited signal** is one whose spectrum is nonzero only over finite interval.

In this case $-\omega_{\max}$ to $\omega_{\max}$, where $\omega_{\max} = \pi$.

Example (2)

If we sample $x(t)$ at a sampling frequency ω_s,

$$x_s(t) = x(t)p(t) = \text{sinc}^2(nT_s) \sum_{n=-\infty}^{\infty} \delta(t - nT_s),$$

then from above

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s) = \boxed{\frac{1}{T_s} \sum_{k=-\infty}^{\infty} \text{tri}\left(\frac{\omega - k\omega_s}{2\pi}\right)}.$$

Example (2)

If we sample $x(t)$ at a sampling frequency ω_s,

$$x_s(t) = x(t)p(t) = \text{sinc}^2(nT_s) \sum_{n=-\infty}^{\infty} \delta(t - nT_s),$$

then from above

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s) = \boxed{\frac{1}{T_s} \sum_{k=-\infty}^{\infty} \text{tri}\left(\frac{\omega - k\omega_s}{2\pi}\right)}.$$

Example (3)

(Picture)(MIT, Lecture 16.2) with no overlap

Question

when is this picture correct?

Question

What happens if sampling rate is too low?

Example (3)

(Picture)(MIT, Lecture 16.2) with no overlap

Question

when is this picture correct?

Question

What happens if sampling rate is too low?

Example (3)

(Picture)(MIT, Lecture 16.2) with no overlap

Question

when is this picture correct?

When $\omega_s - \omega_{\max} > \omega_{\max}$, *i.e.* $\omega_s > 2\omega_{\max}$.

Question

What happens if sampling rate is too low?

Example (3)

(Picture)(MIT, Lecture 16.2) with no overlap

Question

when is this picture correct?

When $\omega_s - \omega_{\max} > \omega_{\max}$, *i.e.* $\omega_s > 2\omega_{\max}$.

Question

What happens if sampling rate is too low?

Overlap of the spectral replicates, call ***aliasing*** *occurs.*
(Picture)*(MIT, Lecture 16.3) with overlap*

Outline

Recover $x(t)$

Question

How can we choose sampling frequency ω_s so that we can recover $x(t)$ from its samples $x(nT_s)$?

Sampling theorem

Theorem

Sampling theorem

Let $x(t)$ is a band-limited signal with $X(\omega) = 0$ for $|\omega| > \omega_{\max}$. Then $x(t)$ is uniquely determined by its samples $x[n] = x(nT_s)$, $n = 0, \pm 1, \pm 2, \ldots$, if

$$\omega_s > 2\omega_{\max}$$

where

$$\omega_s = \frac{2\pi}{T_s}$$

Given these samples, we can reconstruct $x(t)$ by generating a periodic impulse train in which successive impulses have amplitudes that are successive sample values. This impulse train is then processed through an ideal lowpass filter with gain T_s and cutoff frequency greater than $\omega_{\max}$ and less than $\omega_s - \omega_{\max}$. The resulting output signal will exactly equal to $x(t)$.

Sufficient condition

- The **sampling theorem** is a sufficient condition, not a necessary condition, for recovering $x(t)$ from its uniformly spaced samples.
- If we have some additional prior information about the signal, then it may be possible (but usually difficult) to recover $x(t)$.

Sufficient condition

- The **sampling theorem** is a sufficient condition, not a necessary condition, for recovering $x(t)$ from its uniformly spaced samples.
- If we have some additional prior information about the signal, then it may be possible (but usually difficult) to recover $x(t)$.

Recovering $x(t)$ from $x_s(t)$ (1)

If $x(t)$ is bandlimited and sampled at or above the **Nyquist rate** of $2\omega_{\max}$, then there is no overlap of the spectral replicated and (theoretically) we can recover $x(t)$ from $x_s(t)$ by an ideal lowpass filter.

Specifically, from the examples we see that

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

$$X(\omega) = T_s \mathrm{rect}\left(\frac{\omega}{2\omega_c}\right) X_s(\omega) = H(\omega) X_s(\omega)$$

Recovering $x(t)$ from $x_s(t)$ (1)

If $x(t)$ is bandlimited and sampled at or above the **Nyquist rate** of $2\omega_{\max}$, then there is no overlap of the spectral replicated and (theoretically) we can recover $x(t)$ from $x_s(t)$ by an ideal lowpass filter.

Specifically, from the examples we see that

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

$$X(\omega) = T_s \text{rect}\left(\frac{\omega}{2\omega_c}\right) X_s(\omega) = H(\omega) X_s(\omega)$$

Recovering $x(t)$ from $x_s(t)$ (1)

If $x(t)$ is bandlimited and sampled at or above the **Nyquist rate** of $2\omega_{\max}$, then there is no overlap of the spectral replicated and (theoretically) we can recover $x(t)$ from $x_s(t)$ by an ideal lowpass filter.

Specifically, from the examples we see that

$$X_s(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$$

$$X(\omega) = T_s \text{rect}\left(\frac{\omega}{2\omega_c}\right) X_s(\omega) = H(\omega) X_s(\omega)$$

Recovering $x(t)$ from $x_s(t)$ (2)

so to recover $x(t)$ from $x_s(t)$, we need a filter with frequency response

$$\boxed{H(\omega) = T_s \,\text{rect}\left(\frac{\omega}{2\omega_c}\right), \text{ where } \omega_{\max} < \omega_c < \omega_s - \omega_{\max}.}$$

Usually $\omega_c = \omega_s/2 = \omega_{\max}$.
(Picture)(MIT, Lecture 16.4)

Recovering $x(t)$ from $x_s(t)$ (3)

$$x(t) \rightarrow \bigotimes_{\substack{\uparrow \\ p(t)}} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

By the convolution property of the FT, the impulse response of that filter is

$$h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi} t\right).$$

This is a noncausal filter, so in practice it must be approximated by a non-ideal filter.

Recovering $x(t)$ from $x_s(t)$ (3)

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\otimes} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

By the convolution property of the FT, the impulse response of that filter is

$$h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi} t\right).$$

This is a noncausal filter, so in practice it must be approximated by a non-ideal filter.

Recovering $x(t)$ from $x_s(t)$ (3)

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

By the convolution property of the FT, the impulse response of that filter is

$$h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi} t\right).$$

This is a noncausal filter, so in practice it must be approximated by a non-ideal filter.

Example (1)

Example

Why do CD's sample at 44.1kHz?

Example (1)

Example

Why do CD's sample at 44.1kHz?

- *The human ear can only hear up to 20kHz, so even though a given musical instrument could in fact produce signal components above 20kHz, those components are irrelevant (to humans).*
- *So we can safely remove them by lowpass filtering.*
- *That filtering produces a (nearly) bandlimited signal, so the required sampling rate is* $2 \cdot 20\text{kHz} = 40\text{kHz}$.

Example (2)

Example

But why 44.1kHz rather than 40kHz then?

Example (2)

Example

But why 44.1kHz rather than 40kHz then?

So that there is room for a transition band in the lowpass filters (both anti-alias filter and reconstruction filter).

Outline

Aliasing

Definition

The phenomenon, wherein an erroneous signal is recovered from sampled data because the sampling frequency was too low, is called **aliasing**.

Example

Illustration of **aliasing** when sampling sinusoids. Suppose

$$x(t) = \cos(4\pi t) \xleftrightarrow{\mathcal{F}} \pi[\delta(\omega - 4\pi) + \delta(\omega + 4\pi)]$$

is sampled at $\omega_s = 6\pi$. What signal will come out of lowpass filter $\mathrm{rect}\left(\frac{\omega}{6\pi}\right)$?

Aliasing

Definition

The phenomenon, wherein an erroneous signal is recovered from sampled data because the sampling frequency was too low, is called **aliasing**.

Example

Illustration of **aliasing** when sampling sinusoids. Suppose

$$x(t) = \cos(4\pi t) \xleftrightarrow{\mathcal{F}} \pi[\delta(\omega - 4\pi) + \delta(\omega + 4\pi)]$$

is sampled at $\omega_s = 6\pi$. What signal will come out of lowpass filter $\mathrm{rect}\left(\frac{\omega}{6\pi}\right)$?

Illustration of aliasing (1)

Question

What is $\omega_{\max}$? Is this under-sampling?

Illustration of aliasing (1)

Question

What is $\omega_{\max}$? Is this under-sampling?

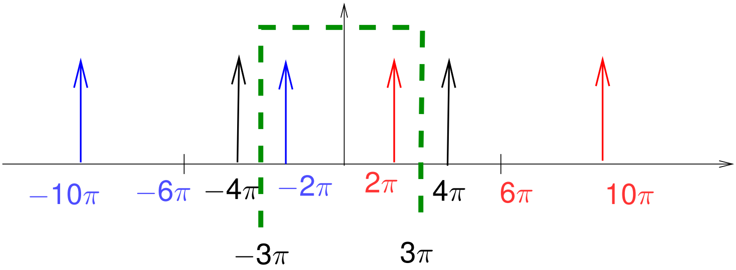
$$\omega_{\max} = 4\pi$$

$$\omega_s = 6\pi < 2\omega_{\max} = 8\pi$$

So this is under-sampling and there is aliasing.

Illustration of aliasing (2)

(Picture) of $X_s(\omega)$

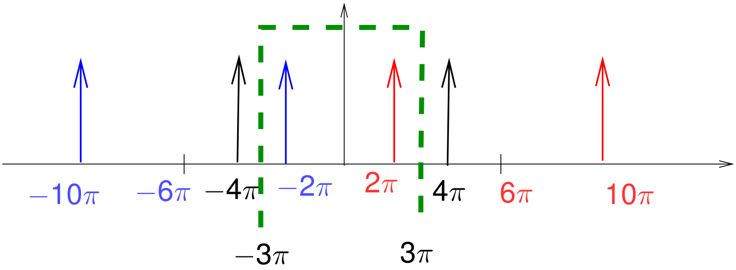


Question

What signal will come out of lowpass filter $\text{rect}\left(\frac{\omega}{6\pi}\right)$ *?*

Illustration of aliasing (2)

(Picture) of $X_s(\omega)$

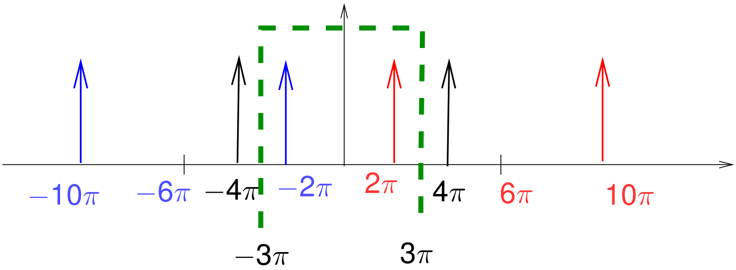


Question

What signal will come out of lowpass filter $\mathrm{rect}\left(\frac{\omega}{6\pi}\right)$ *?*

Illustration of aliasing (2)

(Picture) of $X_s(\omega)$



Question

What signal will come out of lowpass filter $\mathrm{rect}\left(\frac{\omega}{6\pi}\right)$ *?*

$$\boxed{\tilde{x}(t) = \cos(2\pi t)}$$

So input frequency was 4π*, but output was only* 2π.

Illustration of aliasing (3)

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

$$= \frac{2\pi}{\omega_s} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi - k\omega_s) + \delta(\omega + 4\pi - k\omega_s)]$$

$$= \frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi + 6\pi - (k+1)6\pi) + \delta(\omega + 4\pi - 6\pi - (k-1)6\pi)]$$

$$= \frac{1}{3} \sum_{k_1=-\infty}^{\infty} \pi\delta(\omega - k_1 6\pi + 2\pi) + \sum_{k_2=-\infty}^{\infty} \pi\delta(\omega - k_2 6\pi - 2\pi)]$$

$$= \boxed{\frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - k6\pi + 2\pi) + \delta(\omega - k6\pi - 2\pi)]}$$

Illustration of aliasing (3)

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

$$= \frac{2\pi}{\omega_s} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi - k\omega_s) + \delta(\omega + 4\pi - k\omega_s)]$$

$$= \frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi + 6\pi - (k+1)6\pi) + \delta(\omega + 4\pi - 6\pi - (k-1)6\pi)]$$

$$= \frac{1}{3} \sum_{k_1=-\infty}^{\infty} \pi\delta(\omega - k_1 6\pi + 2\pi) + \sum_{k_2=-\infty}^{\infty} \pi\delta(\omega - k_2 6\pi - 2\pi)]$$

$$= \boxed{\frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - k6\pi + 2\pi) + \delta(\omega - k6\pi - 2\pi)]}$$

Illustration of aliasing (3)

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

$$= \frac{2\pi}{\omega_s} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi - k\omega_s) + \delta(\omega + 4\pi - k\omega_s)]$$

$$= \frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi + 6\pi - (k+1)6\pi) + \delta(\omega + 4\pi - 6\pi - (k-1)6\pi)]$$

$$= \frac{1}{3} \sum_{k_1=-\infty}^{\infty} \pi\delta(\omega - k_1 6\pi + 2\pi) + \sum_{k_2=-\infty}^{\infty} \pi\delta(\omega - k_2 6\pi - 2\pi)]$$

$$= \boxed{\frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - k6\pi + 2\pi) + \delta(\omega - k6\pi - 2\pi)]}$$

Illustration of aliasing (3)

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

$$= \frac{2\pi}{\omega_s} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi - k\omega_s) + \delta(\omega + 4\pi - k\omega_s)]$$

$$= \frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi + 6\pi - (k+1)6\pi) + \delta(\omega + 4\pi - 6\pi - (k-1)6\pi)]$$

$$= \frac{1}{3} \sum_{k_1=-\infty}^{\infty} \pi\delta(\omega - k_1 6\pi + 2\pi) + \sum_{k_2=-\infty}^{\infty} \pi\delta(\omega - k_2 6\pi - 2\pi)]$$

$$= \boxed{\frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - k6\pi + 2\pi) + \delta(\omega - k6\pi - 2\pi)]}$$

Illustration of aliasing (3)

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

$$= \frac{2\pi}{\omega_s} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi - k\omega_s) + \delta(\omega + 4\pi - k\omega_s)]$$

$$= \frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - 4\pi + 6\pi - (k+1)6\pi) + \delta(\omega + 4\pi - 6\pi - (k-1)6\pi)]$$

$$= \frac{1}{3} \sum_{k_1=-\infty}^{\infty} \pi\delta(\omega - k_1 6\pi + 2\pi) + \sum_{k_2=-\infty}^{\infty} \pi\delta(\omega - k_2 6\pi - 2\pi)]$$

$$= \boxed{\frac{1}{3} \sum_{k=-\infty}^{\infty} \pi[\delta(\omega - k6\pi + 2\pi) + \delta(\omega - k6\pi - 2\pi)]}$$

Illustration of aliasing (4)

Question

Input signal was $x(t) = \cos(4\pi t)$ with a frequency of 4π, but output signal was $\tilde{x}(t) = \cos(2\pi t)$ with a frequency of only 2π. Does this violate "cosine in cosine out" property we have repeated all semester?

Sinusoidal signals through (real) LTI systems:

$$x(t) = \cos(\omega t + \phi) \rightarrow \boxed{\text{LTI } h(t)} \rightarrow y(t) = |H(j\omega)| \cos(\omega t + \phi + \angle H(j\omega))$$

Illustration of aliasing (4)

Question

Input signal was $x(t) = \cos(4\pi t)$ *with a frequency of* 4π*, but output signal was* $\tilde{x}(t) = \cos(2\pi t)$ *with a frequency of only* 2π*. Does this violate "cosine in cosine out" property we have repeated all semester?*

Sinusoidal signals through (real) LTI systems:

$$x(t) = \cos(\omega t + \phi) \to \boxed{\text{LTI } h(t)} \to y(t) = |H(j\omega)| \cos(\omega t + \phi + \angle H(j\omega))$$

Illustration of aliasing (4)

No, since system is not LTI. The overall system of recovering $x(t)$ from $x_s(t)$ is:

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

Question

Is it not linear or not time-invariant?

Illustration of aliasing (4)

No, since system is not LTI. The overall system of recovering $x(t)$ from $x_s(t)$ is:

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

Question

Is it not linear or not time-invariant?

Illustration of aliasing (4)

No, since system is not LTI. The overall system of recovering $x(t)$ from $x_s(t)$ is:

$$x(t) \to \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \to x_s(t) \to \boxed{H(\omega)} \to x(t)$$

Question

Is it not linear or not time-invariant?

Illustration of aliasing (4)

No, since system is not LTI. The overall system of recovering $x(t)$ from $x_s(t)$ is:

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\otimes} \rightarrow x_s(t) \rightarrow \boxed{H(\omega)} \rightarrow x(t)$$

Question

Is it not linear or not time-invariant?

We know the lowpass filter sub-system is LTI. Not TI due to multiplying by $p(t)$.

Aliasing (1)

Aliasing demonstrated by under sampling of a sinusoid signal (**Vedio** *MIT, Lecture 16, 9:50-19:50min*)

It is important to understand that

- In sampling and reconstruction with an ideal lowpass filter, the reconstructed output will not be equal to the original input in the presence of aliasing, but samples of the reconstructed output will always match the samples of the original signal.
- Aliasing is not necessarily undesirable.

Aliasing (1)

Aliasing demonstrated by under sampling of a sinusoid signal (**Vedio** *MIT, Lecture 16, 9:50-19:50min*)

It is important to understand that

- In sampling and reconstruction with an ideal lowpass filter, the reconstructed output will not be equal to the original input in the presence of aliasing, but samples of the reconstructed output will always match the samples of the original signal.
- Aliasing is not necessarily undesirable.

Aliasing (2)

- The human eye is an imperfect optical system. The pupil acts as a lowpass filter, So the optical image that is impinges on the retina is approximately a bandlimited signal.
- Dr. Harold Edgerton at MIT's Strobe Laboratory. Stroboscopy heavily exploits the concept of aliasing. (**Vedio** *MIT, Lecture 16, 25:22-45:30m*)
- By using sampling with light pulses, motion too fast for the eye to tract can be aliased down to much lower frequencies. In this case, the strobe light represents the sampler, and the lowpass filtering is accomplished visually.

Aliasing (2)

- The human eye is an imperfect optical system. The pupil acts as a lowpass filter, So the optical image that is impinges on the retina is approximately a bandlimited signal.
- Dr. Harold Edgerton at MIT's Strobe Laboratory. Stroboscopy heavily exploits the concept of aliasing. (**Vedio** *MIT, Lecture 16, 25:22-45:30m*)
- By using sampling with light pulses, motion too fast for the eye to tract can be aliased down to much lower frequencies. In this case, the strobe light represents the sampler, and the lowpass filtering is accomplished visually.

Outline

Frequency domain recovery

We have seen using the frequency domain that we can recover $x(t)$ from the impulse sampled version $x_s(t)$ by using an ideal lowpass filter:

$$X_r(\omega) = H(\omega) X_s(\omega)$$

$$H(\omega) = T_s \,\mathrm{rect}\left(\frac{\omega}{2\omega_c}\right), \quad \omega_{\max} < \omega_c < \omega_s - \omega_{\max}$$

Question

What does this look like in the time domain?

Frequency domain recovery

We have seen using the frequency domain that we can recover $x(t)$ from the impulse sampled version $x_s(t)$ by using an ideal lowpass filter:

$$X_r(\omega) = H(\omega)X_s(\omega)$$

$$H(\omega) = T_s \,\mathrm{rect}\left(\frac{\omega}{2\omega_c}\right), \quad \omega_{\max} < \omega_c < \omega_s - \omega_{\max}$$

Question

What does this look like in the time domain?

Time domain recovery

$$x_r(t) = h(t) * x_s(t)$$

$$= h(t) * \left[\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \right]$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s)h(t - nT_s)$$

$$= \boxed{\sum_{n=-\infty}^{\infty} x(nT_s)\frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c(t - nT_s)}{\pi}\right)}$$

$$H(\omega) = T_s \operatorname{rect}\left(\frac{\omega}{2\omega_c}\right) \stackrel{\mathcal{F}}{\longleftrightarrow} h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi}\right)$$

Time domain recovery

$$x_r(t) = h(t) * x_s(t)$$

$$= h(t) * \left[\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \right]$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) h(t - nT_s)$$

$$= \boxed{\sum_{n=-\infty}^{\infty} x(nT_s) \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c (t - nT_s)}{\pi} \right)}$$

$$H(\omega) = T_s \operatorname{rect}\left(\frac{\omega}{2\omega_c} \right) \overset{\mathcal{F}}{\longleftrightarrow} h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi} \right)$$

Time domain recovery

$$x_r(t) = h(t) * x_s(t)$$

$$= h(t) * \left[\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \right]$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s) h(t - nT_s)$$

$$= \boxed{\sum_{n=-\infty}^{\infty} x(nT_s) \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c (t - nT_s)}{\pi} \right)}$$

$$H(\omega) = T_s \operatorname{rect}\left(\frac{\omega}{2\omega_c} \right) \overset{\mathcal{F}}{\longleftrightarrow} h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi} \right)$$

Time domain recovery

$$x_r(t) = h(t) * x_s(t)$$

$$= h(t) * \left[\sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)\right]$$

$$= \sum_{n=-\infty}^{\infty} x(nT_s)h(t - nT_s)$$

$$= \boxed{\sum_{n=-\infty}^{\infty} x(nT_s)\frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c(t - nT_s)}{\pi}\right)}$$

$$H(\omega) = T_s \operatorname{rect}\left(\frac{\omega}{2\omega_c}\right) \overset{\mathcal{F}}{\longleftrightarrow} h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi}\right)$$

Sinc interpolation

$$x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c(t - nT_s)}{\pi}\right)$$

Usually one uses (approximately)

$$\omega_c = \frac{\omega_s}{2}, \quad \omega_s = \frac{2\pi}{T_s}$$

in which case

$$\boxed{x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)}$$

This is known as **sinc interpolation**, since the formula converts from the samples $x(nT_s)$ to the continuous-time signal $x_r(t)$.

Sinc interpolation

$$x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c(t - nT_s)}{\pi}\right)$$

Usually one uses (approximately)

$$\omega_c = \frac{\omega_s}{2}, \quad \omega_s = \frac{2\pi}{T_s}$$

in which case

$$\boxed{x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)}$$

This is known as **sinc interpolation**, since the formula converts from the samples $x(nT_s)$ to the continuous-time signal $x_r(t)$.

Sinc interpolation

$$x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c (t - nT_s)}{\pi}\right)$$

Usually one uses (approximately)

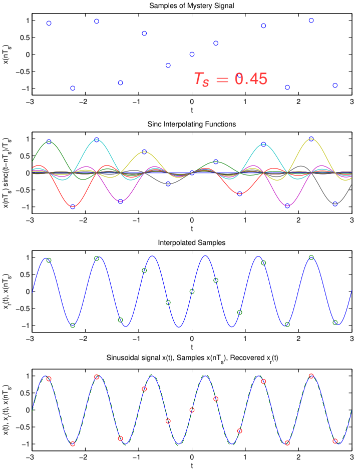
$$\omega_c = \frac{\omega_s}{2}, \quad \omega_s = \frac{2\pi}{T_s}$$

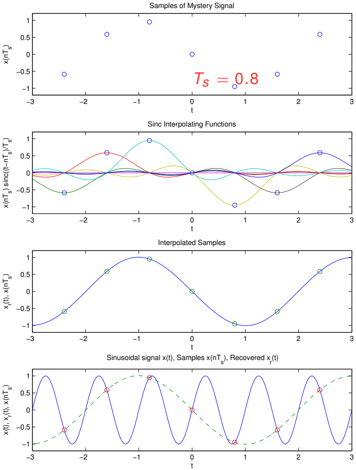
in which case

$$\boxed{x_r(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)}$$

This is known as **sinc interpolation**, since the formula converts from the samples $x(nT_s)$ to the continuous-time signal $x_r(t)$.

Sinc interpolation: example (1)





Sinc interpolation: example (2)

Example

Here are samples of two different signals, and their reconstructions using sinc interpolation.

1. Can you guess the signal from just looking at the samples?
2. Why is there imperfect agreement between the original signal and the sinc-interpolated reconstruction?

Sinc interpolation: example (2)

Example

Here are samples of two different signals, and their reconstructions using sinc interpolation.

1. Can you guess the signal from just looking at the samples?
2. Why is there imperfect agreement between the original signal and the sinc-interpolated reconstruction?

1. *No.*
2. $T_0 = 1 \Longrightarrow T_s \leq T_0/2 = 0.5$ *for exact recovery.*

Linear interpolation (1)

- In practice, the sinc interpolation formula is inconvenient because the sinc function has infinite duration.
- To save computation when using a computer for interpolation, one often simply uses **linear interpolation** (first-order hold), which essentially means "connect the dots" in the graph of $(nT_s, x(nT_s))$ pairs.
- This is exactly what programs like MATLAB do when making plots of "continuous" functions like $\sin(t)$ from a discrete array of t values.

Linear interpolation (1)

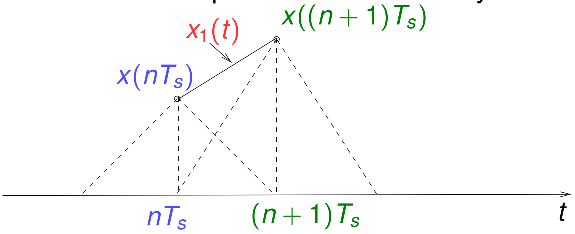
- In practice, the sinc interpolation formula is inconvenient because the sinc function has infinite duration.
- To save computation when using a computer for interpolation, one often simply uses **linear interpolation** (first-order hold), which essentially means "connect the dots" in the graph of $(nT_s, x(nT_s))$ pairs.
- This is exactly what programs like MATLAB do when making plots of "continuous" functions like $\sin(t)$ from a discrete array of t values.

Linear interpolation (1)

- In practice, the sinc interpolation formula is inconvenient because the sinc function has infinite duration.
- To save computation when using a computer for interpolation, one often simply uses **linear interpolation** (first-order hold), which essentially means "connect the dots" in the graph of $(nT_s, x(nT_s))$ pairs.
- This is exactly what programs like MATLAB do when making plots of "continuous" functions like $\sin(t)$ from a discrete array of t values.

Linear interpolation (2)

One way to write linear interpolation mathematically is as follows



For $nT_s \leq t < (n+1)T_s$, the value $x_1(t)$ along the straight line is given from the equation

$$x_1(t) = x(nT_s) + \frac{x((n+1)T_s) - x(nT_s)}{(n+1)T_s - nT_s}(t - nT_s)$$

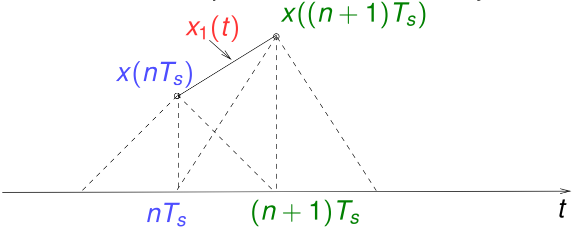
(Point-slope form of line equations $x(t) - x(t_1) = m(t - t_1)$)

$$= \boxed{(1 - \frac{t - nT_s}{T_s})x(nT_s) + \frac{t - nT_s}{T_s}x((n+1)T_s)}$$

(The MATLAB function `interp1` does this operation.)

Linear interpolation (2)

One way to write linear interpolation mathematically is as follows



For $nT_s \leq t < (n+1)T_s$, the value $x_1(t)$ along the straight line is given from the equation

$$x_1(t) = x(nT_s) + \frac{x((n+1)T_s) - x(nT_s)}{(n+1)T_s - nT_s}(t - nT_s)$$

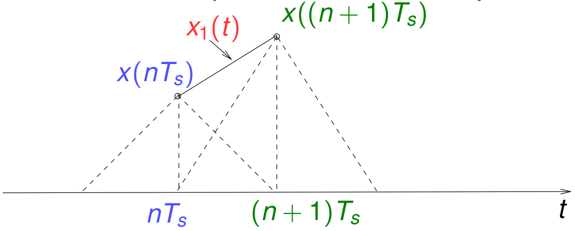
(Point-slope form of line equations $x(t) - x(t_1) = m(t - t_1)$)

$$= \boxed{(1 - \frac{t - nT_s}{T_s})x(nT_s) + \frac{t - nT_s}{T_s}x((n+1)T_s)}$$

(The MATLAB function `interp1` does this operation.)

Linear interpolation (2)

One way to write linear interpolation mathematically is as follows



For $nT_s \leq t < (n+1)T_s$, the value $x_1(t)$ along the straight line is given from the equation

$$x_1(t) = x(nT_s) + \frac{x((n+1)T_s) - x(nT_s)}{(n+1)T_s - nT_s}(t - nT_s)$$

(Point-slope form of line equations $x(t) - x(t_1) = m(t - t_1)$)

$$= \boxed{(1 - \frac{t - nT_s}{T_s})x(nT_s) + \frac{t - nT_s}{T_s}x((n+1)T_s)}$$

(The MATLAB function `interp1` does this operation.)

Linear interpolation (3)

More conveniently, we can express the linear interpolation process using convolution with a triangle function (Video MIT, Lecture 17, 22:00min)

$$x_1(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\mathrm{tri}\left(\frac{t-nT_s}{T_s}\right)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * x_s(t)$$

The impulse response of the linear interpolation filter is

$$h_1(t) = \mathrm{tri}(t/T_s) \overset{\mathcal{F}}{\longleftrightarrow} H_1(\omega) = T_s\,\mathrm{sinc}^2\left(\frac{T_s\omega}{2\pi}\right) = T_s\,\mathrm{sinc}^2\left(\frac{\omega}{\omega_s}\right).$$

Linear interpolation (3)

More conveniently, we can express the linear interpolation process using convolution with a triangle function (Video MIT, Lecture 17, 22:00min)

$$x_1(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\mathrm{tri}\left(\frac{t-nT_s}{T_s}\right)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * x_s(t)$$

The impulse response of the linear interpolation filter is

$$h_1(t) = \mathrm{tri}(t/T_s) \overset{\mathcal{F}}{\longleftrightarrow} H_1(\omega) = T_s\,\mathrm{sinc}^2\left(\frac{T_s\omega}{2\pi}\right) = T_s\,\mathrm{sinc}^2\left(\frac{\omega}{\omega_s}\right).$$

Linear interpolation (3)

More conveniently, we can express the linear interpolation process using convolution with a triangle function (Video MIT, Lecture 17, 22:00min)

$$x_1(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\text{tri}\left(\frac{t-nT_s}{T_s}\right)$$

$$= \text{tri}\left(\frac{t}{T_s}\right) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$$

$$= \text{tri}\left(\frac{t}{T_s}\right) * x_s(t)$$

The impulse response of the linear interpolation filter is

$$h_1(t) = \text{tri}(t/T_s) \overset{\mathcal{F}}{\longleftrightarrow} H_1(\omega) = T_s\,\text{sinc}^2\left(\frac{T_s\omega}{2\pi}\right) = T_s\,\text{sinc}^2\left(\frac{\omega}{\omega_s}\right).$$

Linear interpolation (3)

More conveniently, we can express the linear interpolation process using convolution with a triangle function (Video MIT, Lecture 17, 22:00min)

$$x_1(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\text{tri}\left(\frac{t-nT_s}{T_s}\right)$$

$$= \text{tri}\left(\frac{t}{T_s}\right) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$$

$$= \text{tri}\left(\frac{t}{T_s}\right) * x_s(t)$$

The impulse response of the linear interpolation filter is

$$h_1(t) = \text{tri}(t/T_s) \xleftrightarrow{\mathcal{F}} H_1(\omega) = T_s\,\text{sinc}^2\left(\frac{T_s\omega}{2\pi}\right) = T_s\,\text{sinc}^2\left(\frac{\omega}{\omega_s}\right).$$

Linear interpolation (3)

More conveniently, we can express the linear interpolation process using convolution with a triangle function (Video MIT, Lecture 17, 22:00min)

$$x_1(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\mathrm{tri}\left(\frac{t-nT_s}{T_s}\right)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t-nT_s)$$

$$= \mathrm{tri}\left(\frac{t}{T_s}\right) * x_s(t)$$

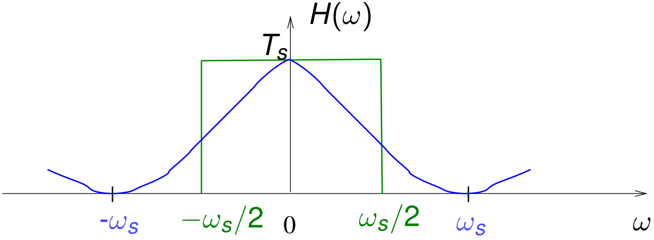
The impulse response of the linear interpolation filter is

$$h_1(t) = \mathrm{tri}(t/T_s) \overset{\mathcal{F}}{\longleftrightarrow} H_1(\omega) = T_s\,\mathrm{sinc}^2\left(\frac{T_s\omega}{2\pi}\right) = T_s\,\mathrm{sinc}^2\left(\frac{\omega}{\omega_s}\right).$$

Linear interpolation (4)

$$\boxed{H_1(\omega) = T_s \mathrm{sinc}^2\left(\frac{\omega}{\omega_s}\right)}$$

$H_1(\omega)$ with first zero at $\pm\omega_s$ vs ideal $H(\omega)$ with cutoff at $\pm\omega_s/2$.



Linear interpolation (5)

Thus in the frequency domain we have

$$X_1(\omega) = T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X_s(\omega)$$

$$= T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) \left[\frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)\right]$$

$$= \boxed{\sum_{k=-\infty}^{\infty} \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X(\omega - k\omega_s)}$$

Linear interpolation (5)

Thus in the frequency domain we have

$$X_1(\omega) = T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X_s(\omega)$$

$$= T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) \left[\frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)\right]$$

$$= \boxed{\sum_{k=-\infty}^{\infty} \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X(\omega - k\omega_s)}$$

Linear interpolation (5)

Thus in the frequency domain we have

$$X_1(\omega) = T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X_s(\omega)$$

$$= T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) \left[\frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)\right]$$

$$= \boxed{\sum_{k=-\infty}^{\infty} \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right) X(\omega - k\omega_s)}$$

Zero-order hold interpolation

The **zero-order hold** (**nearest neighbor interpolation**) system samples $x(t)$ at a given instant and holds that value until the next instant at which a sample is taken (MIT, Lecture 17.2).

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\otimes} \rightarrow x_s(t) \rightarrow \boxed{h_2(t)} \rightarrow x_2(t)$$

The impulse response of the zero-order hold filter is (MIT, Lecture 17.4) (Video, MIT, Lecture 17, 18:30min):

$$\boxed{h_2(t) = \mathrm{rect}\left(\frac{t}{T_s} - \frac{1}{2}\right)}$$

$$h_2(t) \xleftrightarrow{\mathcal{F}} H_2(\omega) = T_s \,\mathrm{sinc}\left(\frac{\omega T_s}{2\pi}\right) e^{-j\omega T_s/2} = \boxed{T_s \,\mathrm{sinc}\left(\frac{\omega}{\omega_s}\right) e^{-j\omega T_s/2}}$$

Zero-order hold interpolation

The **zero-order hold** (**nearest neighbor interpolation**) system samples $x(t)$ at a given instant and holds that value until the next instant at which a sample is taken (MIT, Lecture 17.2).

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\otimes} \rightarrow x_s(t) \rightarrow \boxed{h_2(t)} \rightarrow x_2(t)$$

The impulse response of the zero-order hold filter is (MIT, Lecture 17.4) (Video, MIT, Lecture 17, 18:30min):

$$\boxed{h_2(t) = \text{rect}\left(\frac{t}{T_s} - \frac{1}{2}\right)}$$

$$h_2(t) \xleftrightarrow{\mathcal{F}} H_2(\omega) = T_s \operatorname{sinc}\left(\frac{\omega T_s}{2\pi}\right) e^{-j\omega T_s/2} = \boxed{T_s \operatorname{sinc}\left(\frac{\omega}{\omega_s}\right) e^{-j\omega T_s/2}}$$

Zero-order hold interpolation

The **zero-order hold** (**nearest neighbor interpolation**) system samples $x(t)$ at a given instant and holds that value until the next instant at which a sample is taken (MIT, Lecture 17.2).

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \rightarrow x_s(t) \rightarrow \boxed{h_2(t)} \rightarrow x_2(t)$$

The impulse response of the zero-order hold filter is (MIT, Lecture 17.4) (Video, MIT, Lecture 17, 18:30min):

$$\boxed{h_2(t) = \text{rect}\left(\frac{t}{T_s} - \frac{1}{2}\right)}$$

$$h_2(t) \xleftrightarrow{\mathcal{F}} H_2(\omega) = T_s\,\text{sinc}\left(\frac{\omega T_s}{2\pi}\right) e^{-j\omega T_s/2} = \boxed{T_s\,\text{sinc}\left(\frac{\omega}{\omega_s}\right) e^{-j\omega T_s/2}}$$

Zero-order hold interpolation: example (1)

Example

How to recover $x(t)$ from $x_2(t)$?

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\otimes} \rightarrow x_s(t) \rightarrow \boxed{h_2(t)} \rightarrow x_2(t) \rightarrow \boxed{?} \rightarrow x(t).$$

Zero-order hold interpolation: example (1)

Example

How to recover $x(t)$ from $x_2(t)$?

$$x(t) \rightarrow \underset{\substack{\uparrow \\ p(t)}}{\bigotimes} \rightarrow x_s(t) \rightarrow \boxed{h_2(t)} \rightarrow x_2(t) \rightarrow \boxed{?} \rightarrow x(t).$$

Need ***inverse filter****.*

$$H_{2,i}(\omega) = \frac{H(\omega)}{H_2(\omega)} = H(\omega)\frac{e^{j\omega T_s/2}}{T_s \operatorname{sinc}\left(\frac{\omega}{\omega_s}\right)} = \operatorname{rect}\left(\frac{\omega}{\omega_s}\right)\frac{e^{j\omega T_s/2}}{\operatorname{sinc}\left(\frac{\omega}{\omega_s}\right)}$$

(textbook, Figure 7.8, p. 522)

Interpolations

1. Sinc interpolation (Ideal interpolation filter)

$$h(t) = \frac{\omega_c T_s}{\pi} \operatorname{sinc}\left(\frac{\omega_c t}{\pi}\right) \xleftrightarrow{\mathcal{F}} H(\omega) = T_s \operatorname{rect}\left(\frac{\omega}{2\omega_c}\right)$$

2. Linear interpolation (first-order hold filter)

$$h_1(t) = \operatorname{tri}(t/T_s) \xleftrightarrow{\mathcal{F}} H_1(\omega) = T_s \operatorname{sinc}^2\left(\frac{\omega}{\omega_s}\right)$$

3. Nearest neighbor interpolation (zero-order hold filter)

$$h_2(t) = \operatorname{rect}\left(\frac{t}{T_s} - \frac{1}{2}\right) \xleftrightarrow{\mathcal{F}} H_2(\omega) = T_s \operatorname{sinc}\left(\frac{\omega}{\omega_s}\right) e^{-j\omega T_s/2}$$

$H(\omega)$***(Picture)***MIT, Lecture 17.5
Image sampling and reconstruction example (Video: MIT Lecture 17, 28:30m)

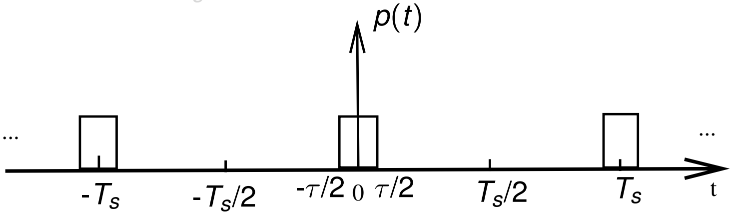
Outline

Realistic non-impulse sampling (1)

- Real A/D converters do not use ideal impulse functions, since a finite time-interval of the signal must be measured for each sample so some nonzero current can flow.
- We can model sampling more realistically by considering, for example, rectangular pulse trains with a small duty cycle.

$$p(t) = \sum_{n=-\infty}^{\infty} \mathrm{rect}\left(\frac{t - nT_s}{\tau}\right)$$

for $0 < \tau << T_s$.

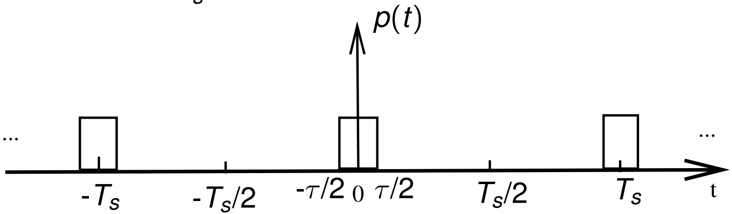


Realistic non-impulse sampling (1)

- Real A/D converters do not use ideal impulse functions, since a finite time-interval of the signal must be measured for each sample so some nonzero current can flow.
- We can model sampling more realistically by considering, for example, rectangular pulse trains with a small duty cycle.

$$p(t) = \sum_{n=-\infty}^{\infty} \mathrm{rect}\left(\frac{t - nT_s}{\tau}\right)$$

for $0 < \tau << T_s$.



Realistic non-impulse sampling (2)

Let $p(t)$ denote such a pulse train, or for that matter, any periodic signal, with period T_s, the sampling frequency. By Fourier series, we know that

$$p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}, \quad c_k = \frac{1}{T_s} \int_{-T_s/2}^{-T_s/2} p(t) e^{-jk\omega_s t} \, dt.$$

Question

Suppose we "sample" a signal $x(t)$ by multiplying by $p(t)$ to form $x_s(t) = x(t)p(t)$. What is the spectrum of $x_s(t)$?

Realistic non-impulse sampling (2)

Let $p(t)$ denote such a pulse train, or for that matter, any periodic signal, with period T_s, the sampling frequency. By Fourier series, we know that

$$p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}, \quad c_k = \frac{1}{T_s} \int_{-T_s/2}^{-T_s/2} p(t) e^{-jk\omega_s t} \, dt.$$

Question

Suppose we "sample" a signal $x(t)$ by multiplying by $p(t)$ to form $x_s(t) = x(t)p(t)$. What is the spectrum of $x_s(t)$?

Realistic non-impulse sampling (2)

Let $p(t)$ denote such a pulse train, or for that matter, any periodic signal, with period T_s, the sampling frequency. By Fourier series, we know that

$$p(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}, \quad c_k = \frac{1}{T_s} \int_{-T_s/2}^{-T_s/2} p(t) e^{-jk\omega_s t} \, dt.$$

Question

Suppose we "sample" a signal $x(t)$ by multiplying by $p(t)$ to form $x_s(t) = x(t)p(t)$. What is the spectrum of $x_s(t)$?

Realistic non-impulse sampling (3)

$$x_s(t) = x(t)p(t) = x(t)\left[\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}\right] = \sum_{k=-\infty}^{\infty} c_k \left[x(t)e^{jk\omega_s t}\right]$$

so by the frequency-shift property of the FT:

$$\boxed{X_s(\omega) = \sum_{k=-\infty}^{\infty} c_k X(\omega - k\omega_s).}$$

Realistic non-impulse sampling (3)

$$x_s(t) = x(t)p(t) = x(t)\left[\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_s t}\right] = \sum_{k=-\infty}^{\infty} c_k \left[x(t)e^{jk\omega_s t}\right]$$

so by the frequency-shift property of the FT:

$$\boxed{X_s(\omega) = \sum_{k=-\infty}^{\infty} c_k X(\omega - k\omega_s).}$$

Realistic non-impulse sampling: example (1)

Example

if $p(t)$ is the impulse train,

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

then

$$c_k = 1/T_s, \quad \text{(derived previously as FS example)}$$

so

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} c_k X(\omega - k\omega_s) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} X(\omega - k\omega_s)$$

which is identical to our earlier formula!

Realistic non-impulse sampling: example (2)

Example

$X(\omega)$ has a triangular spectrum

$$X(\omega) = \operatorname{tri}\left(\frac{\omega}{\omega_1}\right).$$

$p(t)$ is a rectangular pulse train with period T_s and width τ per cycle, and amplitude $1/\tau$. Find the spectrum of the sampled signal

$$x_s(t) = x(t)p(t).$$

Realistic non-impulse sampling: example (3)

From the FS table

$$c_k = \frac{1}{T_s} \operatorname{sinc}\left(\frac{k\tau\omega_s}{2\pi}\right)$$

so

$$X_s(\omega) = \sum_{k=-\infty}^{\infty} \frac{1}{T_s} \operatorname{sinc}\left(\frac{\tau k\omega_s}{2\pi}\right) X(\omega - k\omega_s)$$

$$= \boxed{\sum_{k=-\infty}^{\infty} \frac{1}{T_s} \operatorname{sinc}\left(\frac{\tau k\omega_s}{2\pi}\right) \operatorname{tri}\left(\frac{\omega - k\omega_s}{\omega_1}\right)}$$

with each replicate scaled down by corresponding c_k.

Outline

Computing the FT (1)

Question

How does a spectrum analyzer work?

- We have seen that a band-limited signal can be reconstructed from its samples (provided sampling rate is high enough).
- If we can find the original signal from its samples, then we should also be able to find the FT of the signal from its samples.
- We find the original signal from its sample by sinc interpolation:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

Computing the FT (1)

Question

How does a spectrum analyzer work?

- We have seen that a band-limited signal can be reconstructed from its samples (provided sampling rate is high enough).
- If we can find the original signal from its samples, then we should also be able to find the FT of the signal from its samples.
- We find the original signal from its sample by sinc interpolation:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

Computing the FT (1)

Question

How does a spectrum analyzer work?

- We have seen that a band-limited signal can be reconstructed from its samples (provided sampling rate is high enough).
- If we can find the original signal from its samples, then we should also be able to find the FT of the signal from its samples.
- We find the original signal from its sample by sinc interpolation:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\operatorname{sinc}\left(\frac{t-nT_s}{T_s}\right)$$

Computing the FT (1)

Question

How does a spectrum analyzer work?

- We have seen that a band-limited signal can be reconstructed from its samples (provided sampling rate is high enough).
- If we can find the original signal from its samples, then we should also be able to find the FT of the signal from its samples.
- We find the original signal from its sample by sinc interpolation:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\mathrm{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

Computing the FT (2)

$$\operatorname{sinc}\left(\frac{t}{T_s}\right) \overset{\mathcal{F}}{\longleftrightarrow} T_s \operatorname{rect}\left(\frac{T_s\omega}{2\pi}\right)$$

by the time-shift property of the FT:

$$\operatorname{sinc}\left(\frac{t-nT_s}{T_s}\right) \overset{\mathcal{F}}{\longleftrightarrow} e^{-j\omega nT_s} T_s \operatorname{rect}\left(\frac{T_s\omega}{2\pi}\right)$$

Thus by linearity of the FT:

$$x(t) \overset{\mathcal{F}}{\longleftrightarrow} X(\omega)$$

$$\sum_{n=-\infty}^{\infty} x(nT_s) \operatorname{sinc}\left(\frac{t-nT_s}{T_s}\right) \overset{\mathcal{F}}{\longleftrightarrow} \boxed{\sum_{n=-\infty}^{\infty} x(nT_s) e^{-j\omega nT_s} T_s \operatorname{rect}\left(\frac{T_s\omega}{2\pi}\right)}$$

DTFT

Definition

The **discrete-time Fourier transform** (**DTFT**) of the sequence $x[n] = x(nT_s), n = 0, \pm 1, \pm 2, \ldots$ is defined as follows:

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}, \quad \text{(analysis equation)}.$$

The **inverse DTFT** is

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(\Omega) e^{j\Omega n} d\Omega, \quad \text{(synthesis equation)}.$$

DTFT and FT

$$
\begin{aligned}
X(\Omega) &= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \\
X(\omega) &= \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j\omega n T_s} T_s \mathrm{rect}\left(\frac{T_s \omega}{2\pi}\right)
\end{aligned}
$$

Then the FT of the continuous-time signal $x(t)$ is related to the DTFT of the discrete-time signal $x[n]$ as follows:

$$
X(\omega) = \begin{cases} T_s \, X(\Omega)|_{\Omega=\omega T_s}, & |\omega| < \omega_s/2 \\ 0, & \text{otherwise.} \end{cases}
$$

DTFT and FT

$$
\begin{aligned}
X(\Omega) &= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \\
X(\omega) &= \sum_{n=-\infty}^{\infty} x(nT_s) e^{-j\omega n T_s}\, T_s \mathrm{rect}\left(\frac{T_s\omega}{2\pi}\right)
\end{aligned}
$$

Then the FT of the continuous-time signal $x(t)$ is related to the DTFT of the discrete-time signal $x[n]$ as follows:

$$
X(\omega) = \begin{cases} T_s\, X(\Omega)|_{\Omega=\omega T_s}, & |\omega| < \omega_s/2 \\ 0, & \text{otherwise.} \end{cases}
$$

Periodic DTFT

Question

Show the fact that the DTFT is periodic with period 2π.

Proof

$$
\begin{aligned}
X(\Omega + 2\pi) &= \sum_{n=-\infty}^{\infty} x[n] e^{-j(\Omega+2\pi)n} \\
&= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \underbrace{e^{j2\pi n}}_{1} \\
&= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} = X(\Omega).
\end{aligned}
$$

This fact is closely related to the fact that $X_s(\omega)$ is periodic.

Proof

$$
\begin{aligned}
X(\Omega + 2\pi) &= \sum_{n=-\infty}^{\infty} x[n] e^{-j(\Omega+2\pi)n} \\
&= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} \underbrace{e^{j2\pi n}}_{1} \\
&= \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n} = X(\Omega).
\end{aligned}
$$

This fact is closely related to the fact that $X_s(\omega)$ is periodic.

DTFT and FT pairs

DTFT pair:

$$\boxed{X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}, \quad x[n] = \frac{1}{2\pi}\int_{2\pi} X(\Omega)e^{j\Omega n}d\Omega}$$

FT pair

$$\boxed{X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt, \quad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\,d\omega}$$

$$x[n] = x(nT_s), \quad \Omega = \omega T_s$$

Digital spectrum analyzer

Question

How does a digital spectrum analyzer work?

- A continuous-time signal $x(t)$ is first filtered with an anti-alias filter so that it is bandlimited to $\omega_{\max} = \omega_s/2$, where ω_s is the sampling frequency of the A/D chip in the spectrum analyzer.
- Then the signal is sampled at the rate ω_s, and the discrete-time sequence $x[n] = x(nT_s)$ is stored in digital memory for $n = 0, \ldots, N-1$, for some finite number of samples N.
- Then the DTFT formula for $X(\Omega)$ above is computed digitally, with the following modifications.

Digital spectrum analyzer

Question

How does a digital spectrum analyzer work?

- A continuous-time signal $x(t)$ is first filtered with an anti-alias filter so that it is bandlimited to $\omega_{\max} = \omega_s/2$, where ω_s is the sampling frequency of the A/D chip in the spectrum analyzer.
- Then the signal is sampled at the rate ω_s, and the discrete-time sequence $x[n] = x(nT_s)$ is stored in digital memory for $n = 0, \ldots, N-1$, for some finite number of samples N.
- Then the DTFT formula for $X(\Omega)$ above is computed digitally, with the following modifications.

Digital spectrum analyzer

Question

How does a digital spectrum analyzer work?

- A continuous-time signal $x(t)$ is first filtered with an anti-alias filter so that it is bandlimited to $\omega_{\max} = \omega_s/2$, where ω_s is the sampling frequency of the A/D chip in the spectrum analyzer.
- Then the signal is sampled at the rate ω_s, and the discrete-time sequence $x[n] = x(nT_s)$ is stored in digital memory for $n = 0, \ldots, N-1$, for some finite number of samples N.
- Then the DTFT formula for $X(\Omega)$ above is computed digitally, with the following modifications.

Digital spectrum analyzer

Question

How does a digital spectrum analyzer work?

- A continuous-time signal $x(t)$ is first filtered with an anti-alias filter so that it is bandlimited to $\omega_{\max} = \omega_s/2$, where ω_s is the sampling frequency of the A/D chip in the spectrum analyzer.
- Then the signal is sampled at the rate ω_s, and the discrete-time sequence $x[n] = x(nT_s)$ is stored in digital memory for $n = 0, \ldots, N-1$, for some finite number of samples N.
- Then the DTFT formula for $X(\Omega)$ above is computed digitally, with the following modifications.

DTFT modifications

$$\boxed{\text{DTFT:} \quad X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}}$$

1. The infinite sum is replaced by a sum from 0 to $N-1$, since only a finite number of signal samples can be stored.
2. The DTFT $X(\Omega)$ is never computed for all values of Ω, since a computer can only store a finite set of values. Since $X(\Omega)$ is periodic with period 2π, typically only the values

$$\Omega = k\frac{2\pi}{N}, \quad k = 0, \ldots, N-1$$

are computed.

DTFT modifications

$$\boxed{\text{DTFT:}\quad X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}}$$

1. The infinite sum is replaced by a sum from 0 to $N-1$, since only a finite number of signal samples can be stored.
2. The DTFT $X(\Omega)$ is never computed for all values of Ω, since a computer can only store a finite set of values. Since $X(\Omega)$ is periodic with period 2π, typically only the values
$$\Omega = k\frac{2\pi}{N}, \quad k = 0, \ldots, N-1$$
are computed.

DFT

For this choice we write

$$c[k] = X(\Omega)|_{\Omega=k2\pi/N} = \boxed{\sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}},$$

which is known as the **discrete Fourier transform** or **DFT**, since both time and frequency are discrete indices.

FFT

DFT pair:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, \ldots, N-1$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}, \quad n = 0, \ldots, N-1$$

- In software, the DFT is evaluated using the **fast Fourier transform** or **FFT**, which is the `fft` routine in MATLAB.
- The FFT and DFT are the same mathematically; the FFT is just a fast algorithm for computing the DFT coefficients.

FFT

DFT pair:

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, \ldots, N-1 \\
x[n] &= \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}, \quad n = 0, \ldots, N-1
\end{aligned}
$$

- In software, the DFT is evaluated using the **fast Fourier transform** or **FFT**, which is the `fft` routine in MATLAB.
- The FFT and DFT are the same mathematically; the FFT is just a fast algorithm for computing the DFT coefficients.

FFT

DFT pair:

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, \dots, N-1 \\
x[n] &= \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}, \quad n = 0, \dots, N-1
\end{aligned}
$$

- In software, the DFT is evaluated using the **fast Fourier transform** or **FFT**, which is the `fft` routine in MATLAB.
- The FFT and DFT are the same mathematically; the FFT is just a fast algorithm for computing the DFT coefficients.

Outline

Summary

- DSP, A/D conversion
- impulse train sampling, sampling theorem
- Nyquist sampling rate
- lowpass reconstruction
- sinc interpolation
- linear interpolation (first order hold)
- nearest neighbor interpolation (zero order hold)
- non-impulse sampling
- FT vs DTFT vs DFT vs FFT