

Lab 04 

Sampling and Reconstruction

# Lab Objectives

* Understand the concept of sampling for converting a continuous-time signal to a discrete-time signal.
* Learn how sampling affects the frequency-domain characteristics of the signal, and what precautions must be taken to ensure that the signal obtained through sampling is an accurate representation of the original.
* Consider the issue of reconstructing an analog signal from its sampled version. Understand various interpolation methods used and the spectral relationships involved in reconstruction.

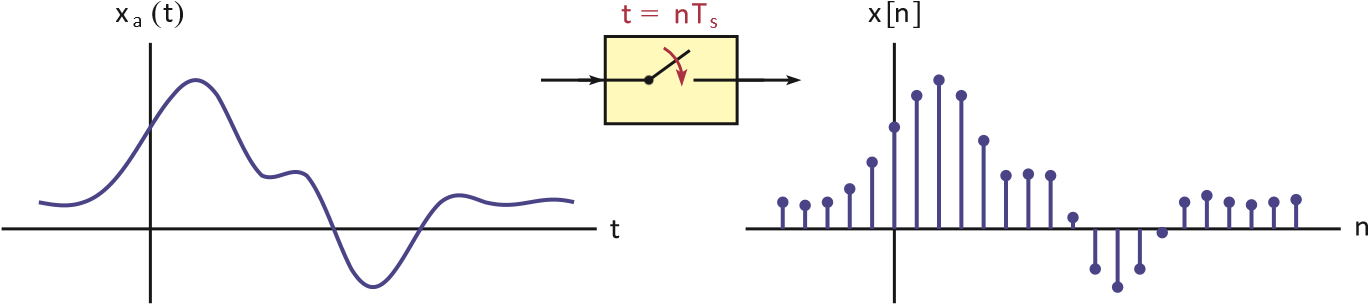
# 4.1 Introduction

The term *sampling* refers to the act of periodically measuring the amplitude of a continuous time signal and constructing a discrete-time signal with the measurements. If certain conditions are satisfied, a continuous-time signal can be completely represented by measurements (samples) taken from it at uniform intervals. This allows us to store and manipulate continuous-time signals on a digital computer.

In general relationship between the continuous time signal *xa*(*t*) and its discrete-time counterpart *x*[*n*] is

where *Ts* is the *sampling interval*, that is, the time interval between consecutive samples. It is also referred to as the *sampling period*. The reciprocal of the sampling interval is called the *sampling rate* or the *sampling frequency*:

The relationship between a continuous-time signal and its discrete-time version is illustrated in Fig. 4.1.



**Figure 4.1** – Graphical representation of sampling relationship.

In signal processing, sampling a continuous signal involves discretizing its values at specific time intervals. This process is mathematically represented by convolving the continuous signal with a series of impulses known as Dirac delta functions, positioned at each sampling instance.

Dirac delta functions are used due to their property of being zero everywhere except at one point, where they're infinitely tall, yet their integral over any interval containing the impulse is 1. This property allows for precise extraction of the signal's amplitude at each sampling instance.

Without the use of impulses, accurately capturing the signal's values at discrete time points would be challenging, akin to attempting to freeze the motion of a moving object without a high-speed camera. Impulses provide the necessary precision to capture the signal's values at specific instances in time, facilitating accurate sampling in signal processing applications.

# 4.2 Sampling of a Continuous-Time Signal

Consider a periodic impulse train *p*(*t*) with period *Ts* :

Multiplication of any signal *x* (*t*) with this impulse train ˜*p*(*t*) would result in amplitude information for *x* (*t*) being retained only at integer multiples of the period *Ts* . Let the signal *x s* (*t*) be defined as the product of the original signal and the impulse train, i.e.,

We will refer to the signal *x s* (*t*) as the *impulse-sampled* version of *x* (*t*). Fig. 4.2 illustrates the relationship between the signals involved in impulse sampling.It is important to understand that the impulse-sampled signal *xs*(*t*) is still a continuous time signal.

*x*

*a*

(

*t*

)

*t*

˜

*p*

(

*t*

)

*t*

*x*

*s*

(

*t*

)

*t*

*T*

*s*

**Figure 4.2** – Impulse-sampling a signal: (a) continuous-time signal *x*(*t*), (b) the pulse train *p* (*t*), (c) impulse-sampled signal *xs* (*t*).

Let us focus on the periodic impulse train ˜*p*(*t*) which is shown in detail in Fig. 4.3.

˜

*p*

(

*t*

)

*t*

−

*T*

*s*



2

*T*

*s*



2

*T*

*s*

*...*

*...*

**Figure 4.3** – Periodic impulse train ˜*p*(*t*).

eries expansion in the form

where is both the sampling rate in and the fundamental frequency of the impulse train. It is computed as . The EFS coefficients for are found as

Substituting the EFS coefficients found in above equations, the impulse train becomes

for the sampled signal . In order to determine the frequency spectrum of the impulse sampled signal let us take the Fourier transform of both sides of above Eqn.

Linearity property of the Fourier transform was used in obtaining the result in final Equation. Furthermore, using the frequency shifting property of the Fourier transform, the term inside the summation becomes

The Fourier transform of the impulse-sampled signal is related to the Fourier transform of the original signal by

This relationship can also be written using frequencies in Hertz as

This is a very significant result. The spectrum of the impulse-sampled signal is obtained

# 4.3 Nyquist Sampling Theorem

Aliasing, a phenomenon commonly encountered in signal processing, occurs when the frequency range of a signal is not adequately limited during sampling. This results in overlapping spectral components, distorting the original spectrum and rendering the recovery of the original signal impossible from its sampled version. Aliasing can be demonstrated when the periodic repetition of spectral components, causes overlapping regions in the spectrum. Even signals with finite frequency ranges are susceptible to aliasing if the sampling rate is not chosen carefully.

Consider a signal with a band-limited spectrum restricted to , where no frequency content exists for . To recover from its sampled version, must be retrievable from , necessitating the absence of overlaps between periodic repetitions of spectral segments. In determining an appropriate sampling rate, the Nyquist Sampling Criterion comes into play. This criterion asserts that for accurate representation, the sampling rate must be at least twice the highest frequency present in the original signal's spectrum. Mathematically, this is expressed as the equation below ensuring that spectral segments do not interfere.

*f*

*X*

*s*

(

*f*

)

*A/T*

*s*

−

*f*

*s*

−

*f*

max

*f*

max

*f*

*s*

−

*f*

*s*

−

*f*

max

−

*f*

*s*

+

*f*

max

*f*

*s*

−

*f*

max

*f*

*s*

+

*f*

max

**)**

**a**

**(**

*f*

*X*

*s*

(

*f*

)

*A/T*

*s*

−

2

*f*

*s*

−

*f*

*s*

−

*f*

max

*f*

max

*f*

*s*

2

*f*

*s*

**(**

**b**

**)**

*f*

*X*

*s*

(

*f*

)

*A/T*

*s*

−

2

*f*

*s*

−

*f*

*s*

−

*f*

max

−

*f*

*s*

+

*f*

max

*f*

*s*

−

*f*

max

*f*

max

*f*

*s*

2

*f*

*s*

**(**

**c**

**)**

**Figure 4.4- Nyquist Theorem**

# 4.4 Practical Issues in Sampling

In previous sections the issue of sampling a continuous-time signal through multiplication by an impulse train was discussed. A practical consideration in the design of samplers is that we do not have ideal impulse trains, and must therefore approximate them with pulse trains. Two important questions that arise in this context are:

1. What would happen to the spectrum if we used a pulse train instead of an impulse train
2. How would the use of pulses affect the methods used in recovering the original signal from its sampled version?

When pulses are used instead of impulses, there are two variations of the sampling operation that can be used, namely *natural sampling* and *zero-order hold sampling*. The former is easier to generate electronically while the latter lends itself better to digital coding through techniques known as *pulse-code modulation* and *delta modulation*.

# 4.4.1 Zero-order hold sampling

In natural sampling, the tops of the pulses are not flat but are rather shaped by the signal *xa*(*t*). This behavior is not always desired, especially when the sampling operation is to be followed by the conversion of each pulse to a digital format. An alternative is to hold the amplitude of each pulse constant, equal to the value of the signal at the left edge of the pulse. This is referred to as *zero-order hold sampling* or *flat-top sampling* and is illustrated in Fig. Below

*xa* (*t*) *x*¯*s* (*t*)

*t*

*t*

(a) (b)

**Figure4.5** – Illustration of zero-order hold sampling.

Conceptually the signal ¯*x s* (*t*) can be modelled as the convolution of the impulse sampled signal *x s* (*t*) and a rectangular pulse with unit amplitude and a duration of *dTs* as shown in Fig. below

*x*

*s*

(

*t*

)

*t*

¯

*x*

*s*

(

*t*

)

*t*

*h*

*zoh*

(

*t*

)

*t*

*x*

*s*

(

*t*

)

¯

*x*

*s*

(

*t*

)

Zero-order

Hold filter

1

*dT*

*s*

**Figure 4.6**– Modeling sampling operation with flat-top pulses.

The impulse response of the zero-order hold filter

Zero-order hold sampled signal ¯*xs*(*t*) can be written as

*x*¯*s* (*t*) = *hz oh* (*t*) ∗ *x s* (*t*)

where *xs*(*t*) represents the impulse sampled signal. The frequency spectrum of the zero-order hold sampled signal is found as

*X*¯*s* (*ω*) = *H z oh* (*ω*) *X s* (*ω*)

The system function for the zero-order hold filter is

The spectrum of the zero-order hold sampled signal is given as

Spectrum of the signal obtained through zero-order hold sampling:

# 4.5 Reconstruction of a Signal

*t*

*T*

*s*

2

*T*

*s*

3

*T*

*s*

4

*T*

*s*

5

*T*

*s*

6

*T*

*s*

7

*T*

*s*

8

*T*

*s*

9

*T*

*s*

10

*T*

*s*

(

a

)

*t*

*x*

*s*

(

*t*

)

*,x*

zoh

(

*t*

)

*T*

*s*

2

*T*

*s*

3

*T*

*s*

4

*T*

*s*

5

*T*

*s*

6

*T*

*s*

7

*T*

*s*

8

*T*

*s*

9

*T*

*s*

10

*T*

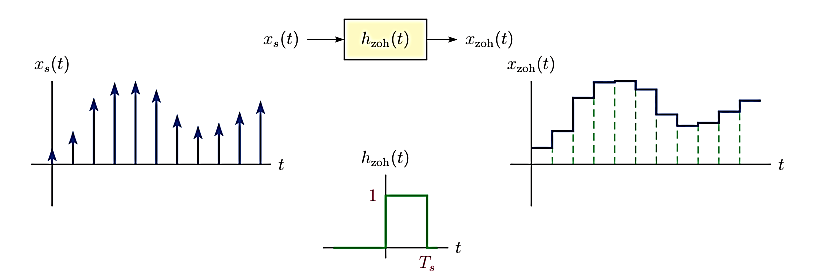
*s*

(b)

**Figure**–4.7 (a) Impulse sampling an analog signal *x* (*t*) to obtain *xs* (*t*), (b) reconstruction using zero-order hold interpolation.

Zero-order hold interpolation can be achieved by processing the impulse sampled signal *xs*(*t*) through *zero-order hold reconstruction filter*, a linear system the impulse response of which is a rectangle with unit amplitude and a duration of *Ts* .

This is illustrated in Fig bleow . The linear system that performs the interpolation is called a *reconstruction filter*.



**Figure 4.8** – Zero-order hold interpolation using an interpolation filter.

The spectral relationship between the analog signal and its naturally sampled version can be used for obtaining the relationship between the analog signal and the signal reconstructed from samples using zero-order hold:

As an alternative to zero-order hold, the gaps between the sampling instants can be filled by linear interpolation, that is, by connecting the tips of the samples with straight lines. This is also known as *first-order hold* interpolation since the straight line segments used in the interpolation correspond to first order polynomials

.

*t*

*x*

*s*

(

*t*

)

*,x*

foh

(

*t*

)

*T*

*s*

2

*T*

*s*

3

*T*

*s*

4

*T*

*s*

5

*T*

*s*

6

*T*

*s*

7

*T*

*s*

8

*T*

*s*

9

*T*

*s*

10

*T*

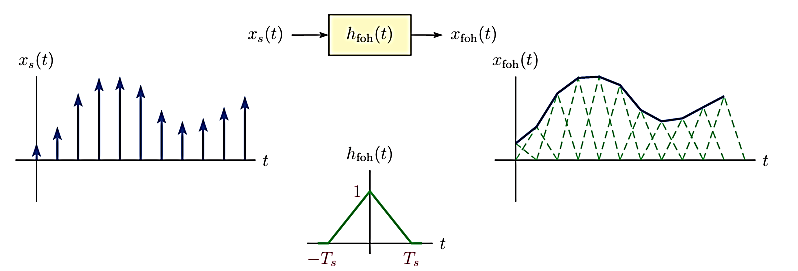
*s*

Straightline segments

**Figure 4.9** – Reconstruction using first-order hold interpolation.

First-order hold interpolation can also be implemented using a *first-order hold reconstruction filter*. The impulse response of such a filter is a triangle in the form

as illustrated in Figure



**Figure 4.10** – First-order hold interpolation using an interpolation filter.

The impulse response *h*foh (*t*) of the first-order hold interpolation filter is non-causal since it starts at *t* = − *Ts* . If a practically realizable interpolator is desired, it would be a simple matter to achieve causality by using a delayed version of *h*foh (*t*) for the impulse response:

*h*¯foh (*t*) = *h*foh (*t* − *Ts* )

In this case, the reconstructed signal would naturally lag behind the sampled signal by *Ts* .

It is insightful to derive the frequency spectra of the reconstructed signals obtained through zero-order hold and first-order hold interpolation. For convenience we will use *f* rather than *ω* in this derivation. The system function for the zero-order hold filter is

*H* zoh (*f* ) = *Ts* sinc(*f Ts* ) *e*− *j πT s*

and the spectrum of the analog signal constructed using the zero-order hold filter is

*X* zoh (*f* ) = *H* zoh (*f* ) *X s* (*f* )

Similarly, the system function for first-order hold filter is

*H* foh (*f* ) = *Ts* sinc2 (*f Ts* )

and the spectrum of the analog signal constructed using the first-order hold filter is

*X* foh (*f* ) = *H* foh (*f* ) *X s* (*f* )

The fact that both interpolation filters have lowpass characteristics warrants further exploration. The Nyquist sampling theorem states that a properly sampled signal can be recovered perfectly from its sampled version. What kind of interpolation is needed for

*f*

*X*

*s*

(

*f*

)

1

*/T*

*s*

−

3

*f*

*s*

−

2

*f*

*s*

−

*f*

*s*

0

*f*

*s*

2

*f*

*s*

3

*f*

*s*

(a)

*f*

|

*H*

*foh*

(

*f*

)

|

*T*

*s*

−

3

*f*

*s*

−

2

*f*

*s*

−

*f*

*s*

0

*f*

*s*

2

*f*

*s*

3

*f*

*s*

(b)

*f*

|

*X*

*foh*

(

*f*

)

|

1

−

3

*f*

*s*

−

2

*f*

*s*

−

*f*

*s*

0

*f*

*s*

2

*f*

*s*

3

*f*

*s*

(c)

**Figure 4.11** – (a) Sample spectrum *Xs* (*f*) for an impulse-sampled signal, (b) magnitude spectrum |*H*foh (*f*)| for the first-order hold interpolation filter, (c) magnitude spectrum |*X*foh (*f*)| for the signal reconstructed using first-order hold interpolation.

perfect reconstruction of the analog signal from its impulse-sampled version? The answer must be found through the frequency spectrum of the sampled signal. Recall that the relationship between *X s* (*f* ), the spectrum of the impulse-sampled signal, and *X a* (*f* ), the frequency spectrum of the original signal, was found in Eqn.

As long as the choice of the sampling rate satisfies the Nyquist sampling criterion, the spectrum of the impulse-sampled signal is simply a sum of frequency shifted versions of the original spectrum, shifted by every integer multiple of the sampling rate. An ideal lowpass filter that extracts the term for *k* = 0 from the summation in above and suppresses all other terms for *k* = ± 1*, . . . ,* ± ∞ would recover the original spectrum *X a* (*f* ), and therefore the original signal *x a* (*t*). Since the highest frequency in a properly sampled signal would be equal to or less than half the sampling rate, an ideal *lowpass filter* with cutoff frequency set equal to *f s /*2 is needed. The system function for such a reconstruction filter is

where we have also included a magnitude scaling by a factor of *Ts* within the system function of the lowpass filter in order to compensate for the 1*/Ts* term in Eqn.

*f*

*H*

*r*

(

*f*

)

*T*

*s*

−

*f*

*s*

*/*

2

*f*

*s*

*/*

2

**Figure 4.12**– Ideal lowpass reconstruction filter.

The frequency spectrum of the output of the filter defined by following Eqn.

This is illustrated in Fig. below

*f*

*X*

*s*

(

*f*

)

*A/T*

*s*

−

*f*

*s*

−

*f*

max

*f*

max

*f*

*s*

−

*f*

*s*

−

*f*

max

−

*f*

*s*

+

*f*

max

*f*

*s*

−

*f*

max

*f*

*s*

+

*f*

max

(a)

*f*

*H*

*r*

(

*f*

)

*T*

*s*

−

*f*

*s*

−

*f*

*s*

*/*

2

*f*

*s*

*/*

2

*f*

*s*

(b)

*f*

*X*

*r*

(

*f*

)=

*H*

*r*

(

*f*

)

*X*

*s*

(

*f*

)

*A*

−

*f*

*s*

−

*f*

max

*f*

max

*f*

*s*

(c)

**Figure 4.13** – Reconstruction using an ideal lowpass reconstruction filter: (a) sample spectrum *Xs* (*f*) for an impulse-sampled signal, (b) system function *Hr* (*f*) for the ideal lowpass filter with cutoff frequency *fs /*2, (c) spectrum *Xr* (*f*) for the signal at the output of the ideal lowpass filter.

The impulse response of the filter is

which, due to the sinc function, has equally-spaced zero crossings that coincide with the sampling instants. The signal at the output of the ideal lowpass filter is obtained by convolving with the impulse-sampled signal given by Eqn. (6.5):

The nature of interpolation performed by the ideal lowpass reconstruction filter is evident from Eqn. Let us consider the output of the filter at one of the sampling instants, say :

We also know that

and Eqn. becomes .

1. The output of the ideal lowpass reconstruction filter is equal to the sampled signal at each sampling instant.
2. Between sampling instants, is obtained by interpolation through the use of sinc functions. This is referred to as bandlimited interpolation and is illustrated in Fig.

*t*

*x*

*s*

(

*t*

)

*,x*

*r*

(

*t*

)

0

*T*

*s*

2

*T*

*s*

3

*T*

*s*

4

*T*

*s*

5

*T*

*s*

6

*T*

*s*

7

*T*

*s*

8

*T*

*s*

9

*T*

*s*

10

*T*

*s*

**Figure 4.14** – Reconstruction of a signal from its sampled version through bandlimited interpolation.

# 4.6 Real life Implementations

Implementing ideal low-pass reconstruction filters and sinc filters in real-world scenarios to obtain original signal from sampled signal. can be challenging due to several practical constraints. While these filters may offer excellent theoretical results, various factors limit their feasibility in practical applications:

4.6.1. Idealization vs. Reality:

The ideal low-pass filter has a rectangular frequency response with perfect attenuation beyond its cutoff frequency. However, achieving an ideal low-pass filter with a sharp cutoff in the real world is challenging. Real filters have non-ideal characteristics, such as finite transition bands, passband ripples, and phase distortions. Similarly the sinc function theoretically provides perfect reconstruction in a sampling system, but implementing it practically requires an infinite-duration filter, which is not feasible. Real-world systems necessitate finite-duration filters, leading to compromises in performance.

4.6.2 Infinite Impulse Response (IIR) vs. Finite Impulse Response (FIR)

The sinc function corresponds to an infinite impulse response (IIR) filter, requiring an infinite number of taps for perfect reconstruction. In practice, implementing filters with an infinite number of taps is not feasible due to computational and hardware limitations. Finite impulse response (FIR) filters are more practical but still face constraints in terms of complexity and resource requirements.

4.6.3 Computational Complexity:

Achieving a truly sharp transition as idealized in a low pass filter between the passband and stopband requires a filter with an infinite number of taps. This leads to computational challenges and increased hardware requirements, making it impractical for real-time applications.

Similarly the mathematical representation of the sinc function involves an infinite sum, and practical implementations often involve truncation. Truncating the sinc function results in a loss of its ideal characteristics, and finding an optimal truncation length is a non-trivial task.

4.6.4 Hardware Complexity :

Implementing a sinc filter with infinite precision in real-world hardware is not possible. Finite word-length effects, quantization errors, and limitations in analog components introduce distortions that deviate from the idealized mathematical model. Also on the other hand ,in real-world scenarios, the ideal low-pass filter may not provide sufficient frequency selectivity. Practical applications often require more complex filters, such as finite impulse response (FIR) filters with sophisticated designs, to meet specific performance criteria.

# 4.7 The first Order Filter

Creating a first-order triangular filter for signal reconstruction presents challenges in both theoretical and practical aspects. Let's break down the technical issues and considerations involved:

1. Theoretical Ideal vs. Real-world Constraint, Frequency Spectrum Realization

- The theoretical ideal triangular signal encompasses both positive and negative frequency spectra. However, in real-world applications, it's impractical to generate signals with negative time or negative frequency components. This constraint limits the implementation of the ideal theoretical model.

he frequency response of a first-order triangular filter ideally spans from -wmax towmax. Achieving this ideal frequency spectrum in real-world scenarios is challenging because signals are typically constructed in the positive frequency domain. This construction introduces delays in the reconstructed signal due to the inherent properties of the filter. The figure below shows an ideal first order low pass filter

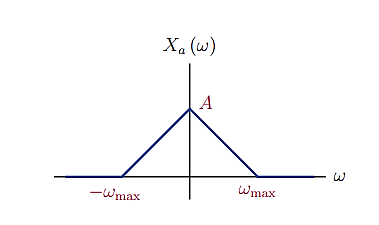


Figure: 4.15 And Ideal First order low pass filter

1. MATLAB Implementation

Due to the real life practical limitations MATLAB implementation constructs a first-order triangular filter using a piecewise linear function. This approach represents the filter's response in the positive frequency spectrum, which is a practical approximation given the constraints of real-world signal generation.

|  |
| --- |
| M=10  dt=0.1  hFOH = [linspace(0,1,M) linspace(1-1/M,0,M) 0 plot([-M:M]\*dt,hFOH,'linewidth',2); |

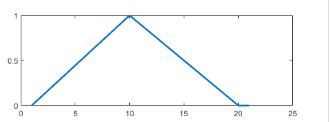


Figure: 41.6 And Ideal First order low pass filter as realized by Matlab for 20

4.. Signal Reconstruction Dely

- Due to the positive frequency spectrum realization of the filter, the reconstructed signal may exhibit delays compared to the ideal theoretical model. Minimizing these delays requires careful design and optimization of the filter parameters.

# The Ideal Low Pass Filter

The ideal low-pass filter has a frequency response characterized by a sinc function, exhibiting perfect attenuation beyond a certain cutoff frequency while preserving lower frequencies. However, realizing this ideal response in the real world is challenging

1. Frequency Spectrum Realization

The sinc function response of an ideal low-pass filter extends infinitely in both positive and negative frequency domains. In practice, implementing such a filter requires approximations and finite-length representations due to computational constraints and the impossibility of infinite response.

2. MATLAB Implementation:

- In MATLAB, one common approach to simulate an ideal low-pass filter with a sinc function response is by using the sinc function itself or windowed versions of the sinc function These functions can be truncated to a finite length to approximate the ideal response while considering computational limitations.

3. Frequency Response Approximation

While the ideal low-pass filter has a perfect sinc function response, real-world implementations typically involve compromises. Finite-length approximations, windowing effects, and practical limitations result in deviations from the ideal response, affecting the filter's frequency selectivity and attenuation characteristics.

Truncating a sinc signal to create a practical first-order filter is common practice, but it comes with consequences as shown in figure below . The truncation introduces ringing artifacts in the frequency domain, resulting from sharp discontinuities. These artifacts manifest as noise and delays in the reconstructed signal, degrading its quality and accuracy.

