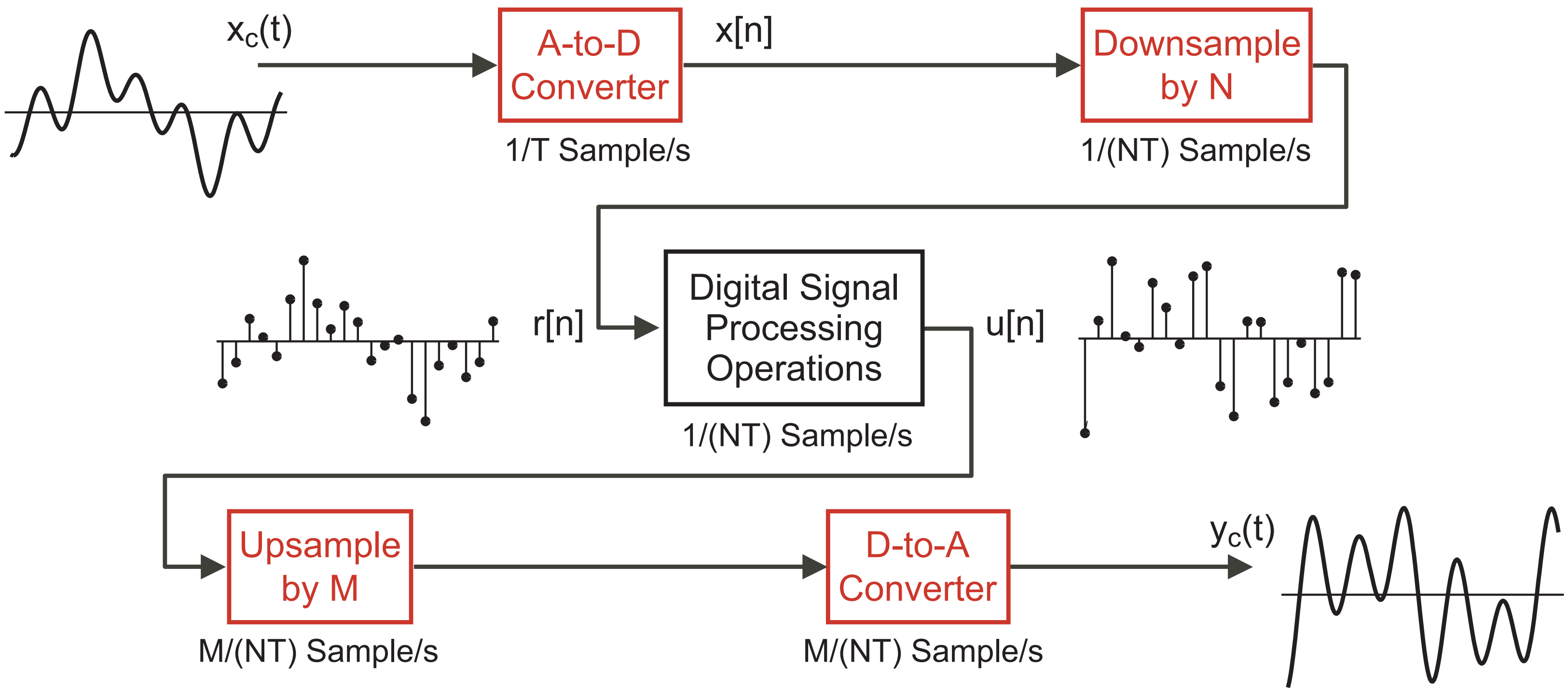
Downsampling, Upsampling, and Reconstruction

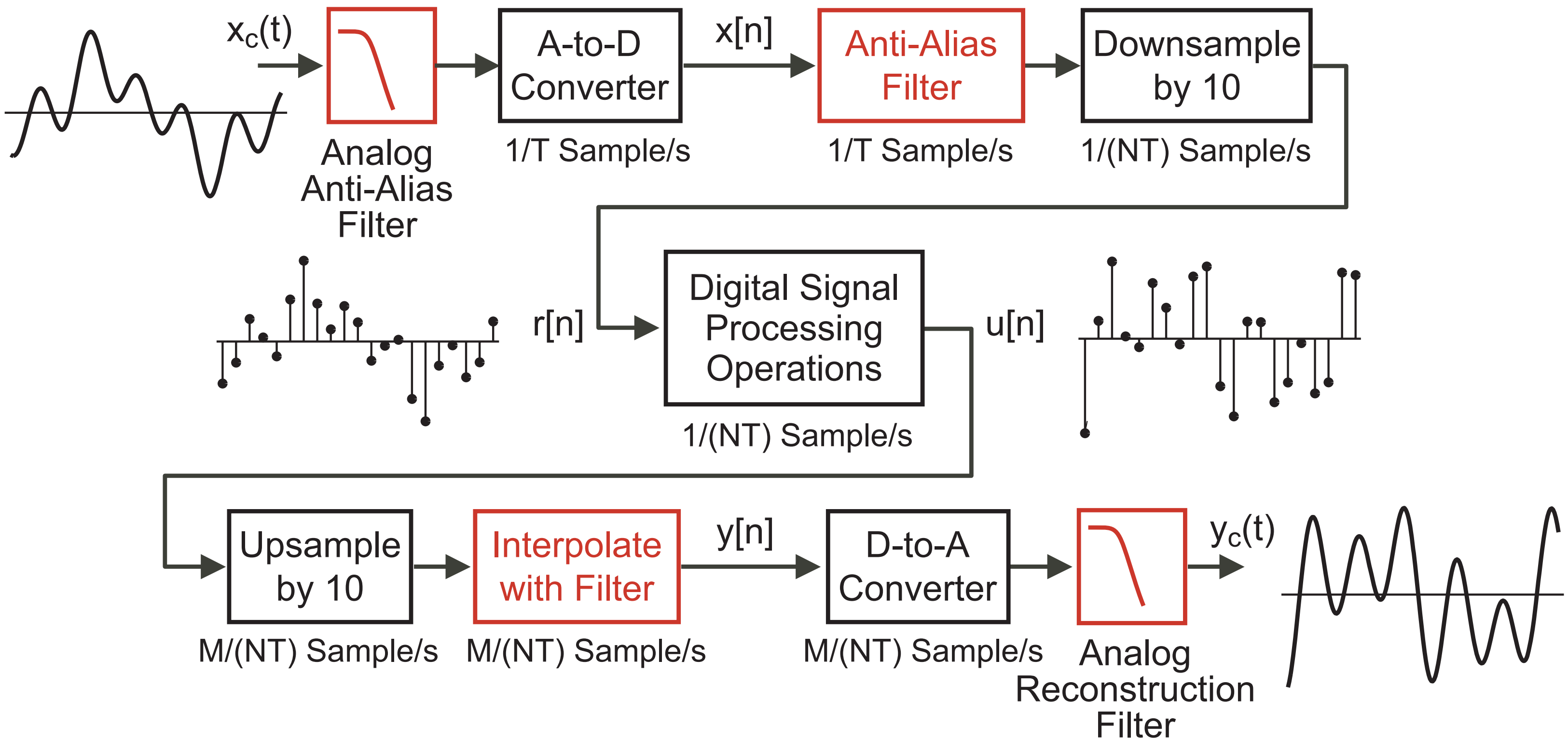
- A-to-D and its relation to sampling
- Downsampling and its relation to sampling
- Upsampling and interpolation
- D-to-A and reconstruction filtering
- Filters and their relation to convolution

Digital Processing of Analog Signals



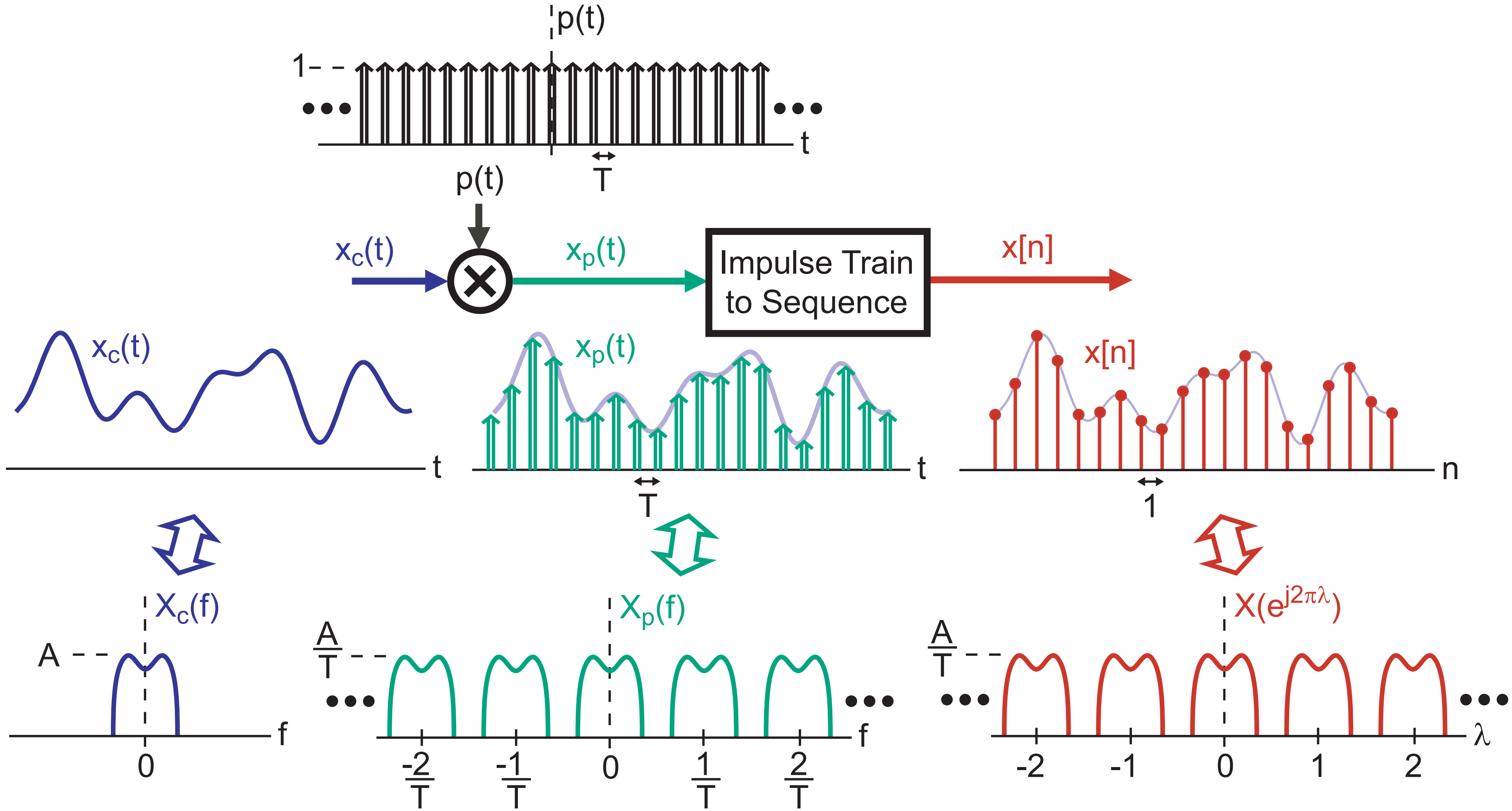
- *Digital* circuits can perform very complex processing of *analog* signals, but require
  - Conversion of analog signals to the digital domain
  - Conversion of digital signals to the analog domain
  - Downsampling and upsampling to match sample rates of A-to-D, digital processor, and D-to-A

Inclusion of Filtering Operations



- **A-to-D and downsampler require *anti-alias* filtering**
  - **Prevents aliasing**
- **D-to-A and upsampler require *interpolation* (i.e., *reconstruction*) filtering**
  - **Provides \`smoothly' changing waveforms**

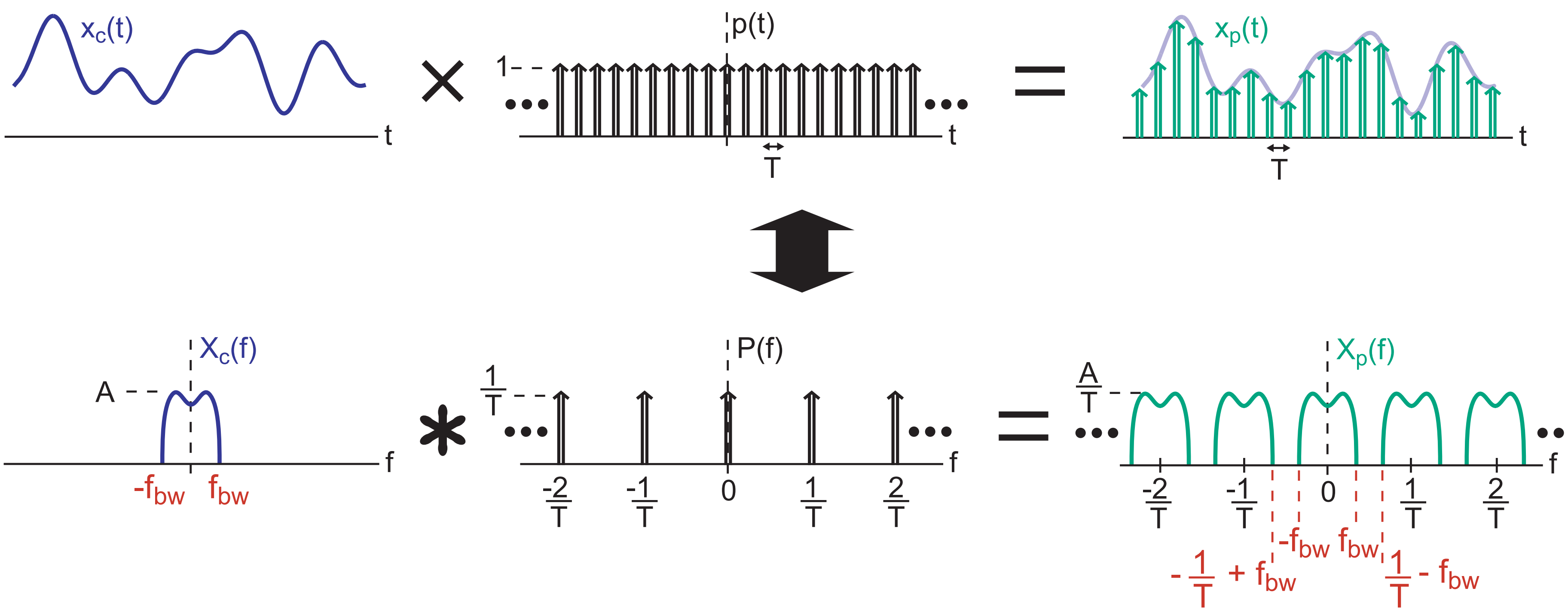
Summary of Sampling Process (Review)



- Sampling leads to periodicity in frequency domain

We need to avoid overlap of replicated signals in frequency domain (i.e., aliasing)

The Sampling Theorem (Review)

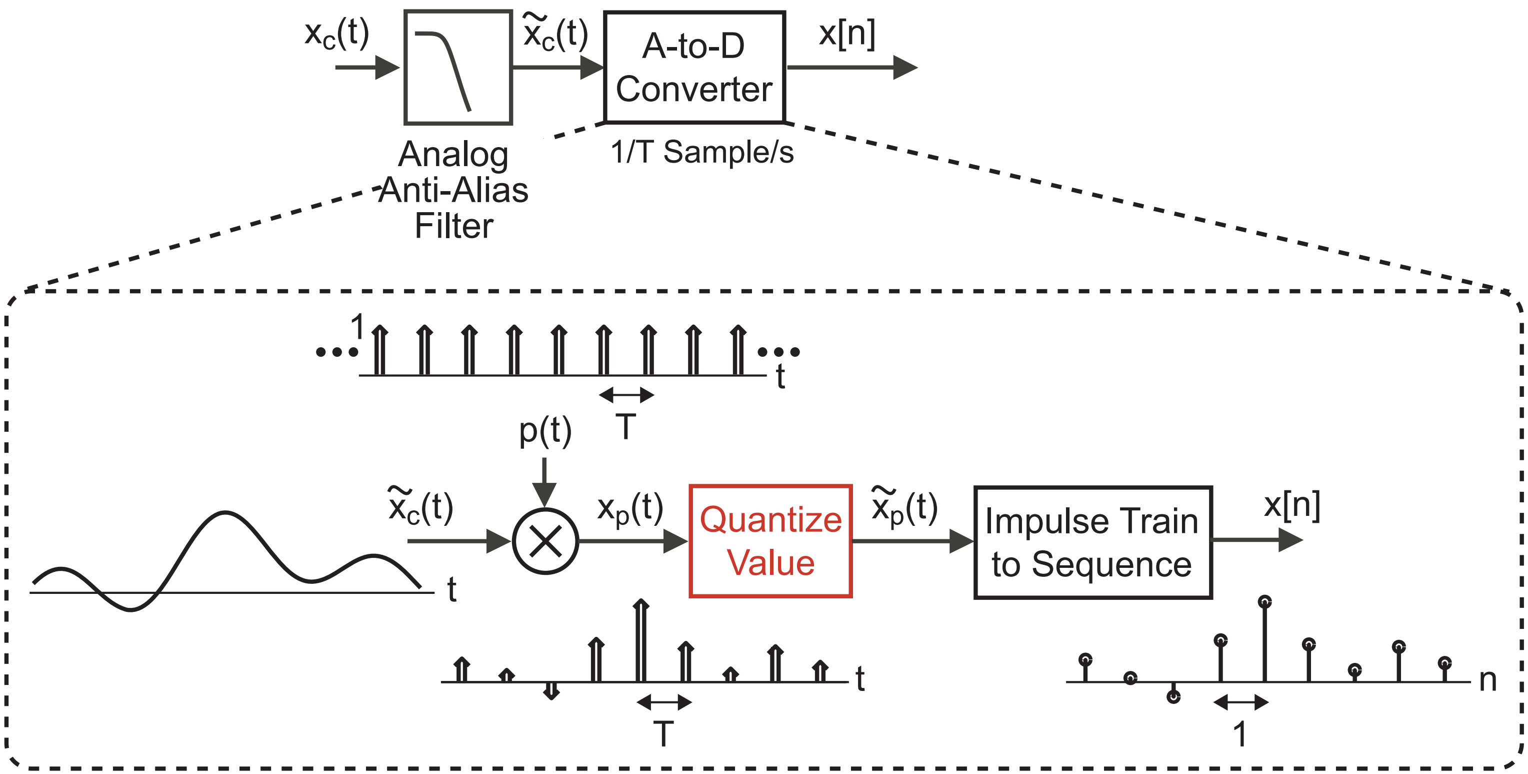


- **Overlap in frequency domain (i.e., aliasing) is avoided if:**

$$\frac{1}{T} - f_{bw} \geq f_{bw} \quad \Rightarrow \quad \boxed{\frac{1}{T} \geq 2 f_{bw}}$$

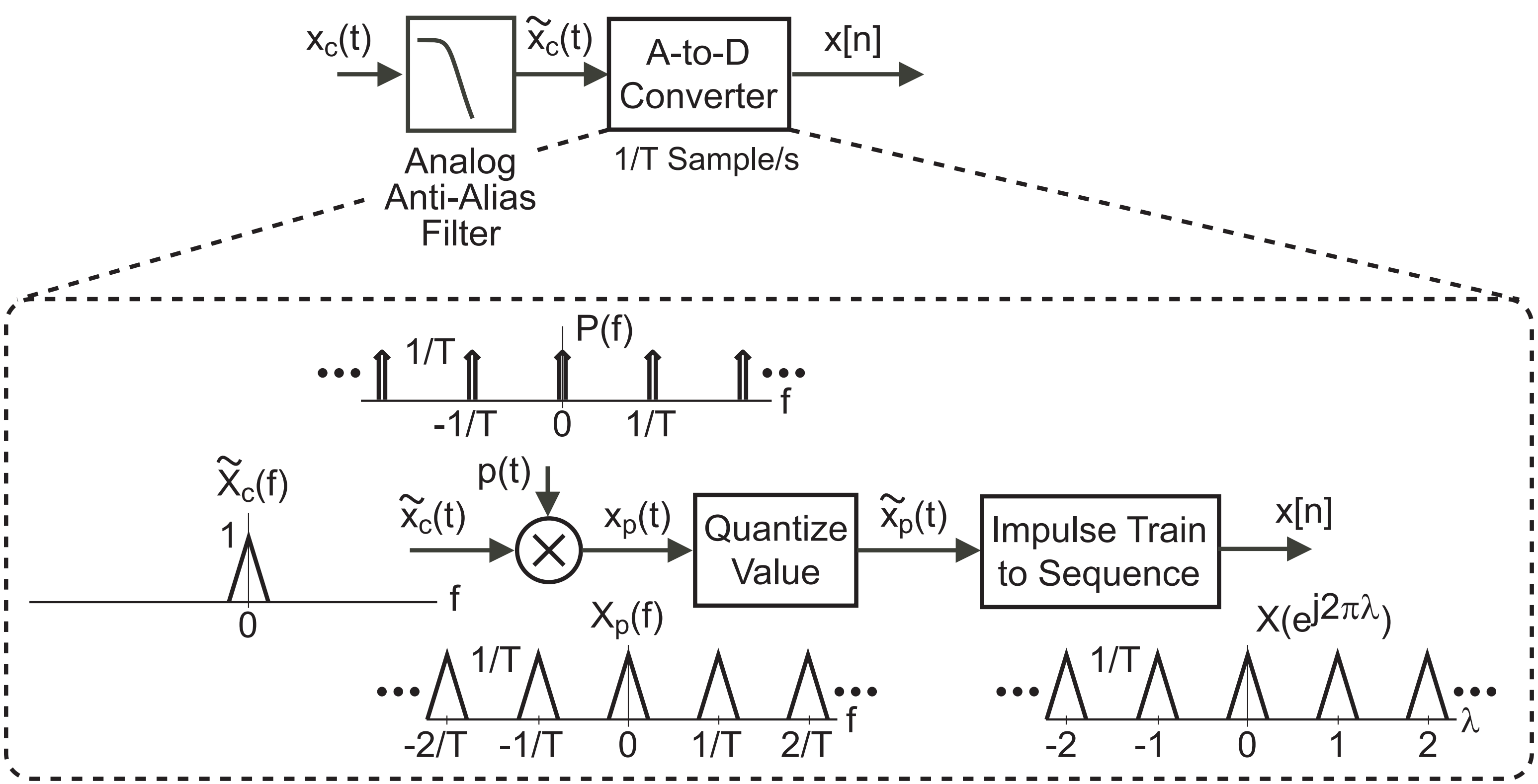
- **We refer to the minimum *1/T* that avoids aliasing as the *Nyquist* sampling frequency**

A-to-D Converter



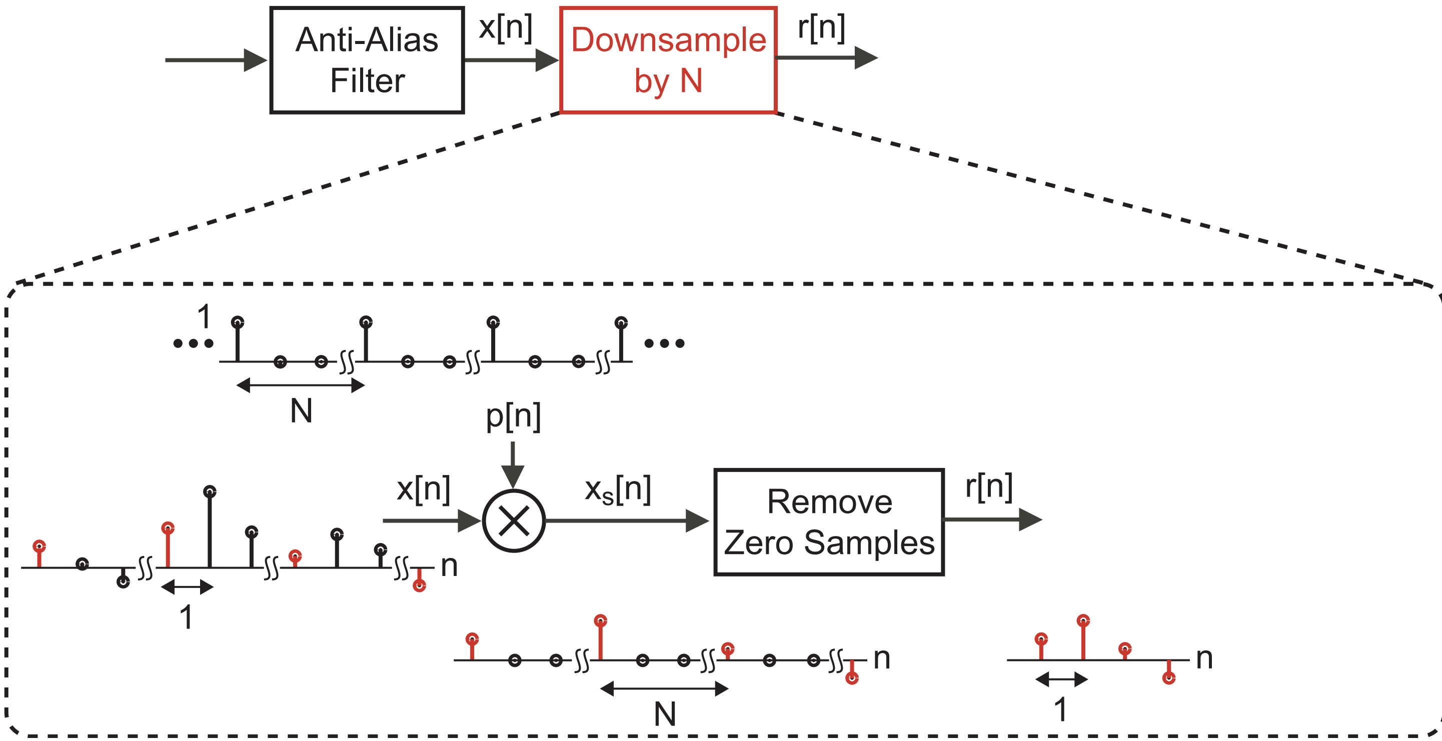
- Operates using both a *sampler* and *quantizer*
  - Sampler converts *continuous-time* input signal into a *discrete-time* sequence
  - Quantizer converts *continuous-valued* signal/sequence into a *discrete-valued* signal/sequence
    - Introduces *quantization noise* as discussed in Lab 4

Frequency Domain View of A-to-D



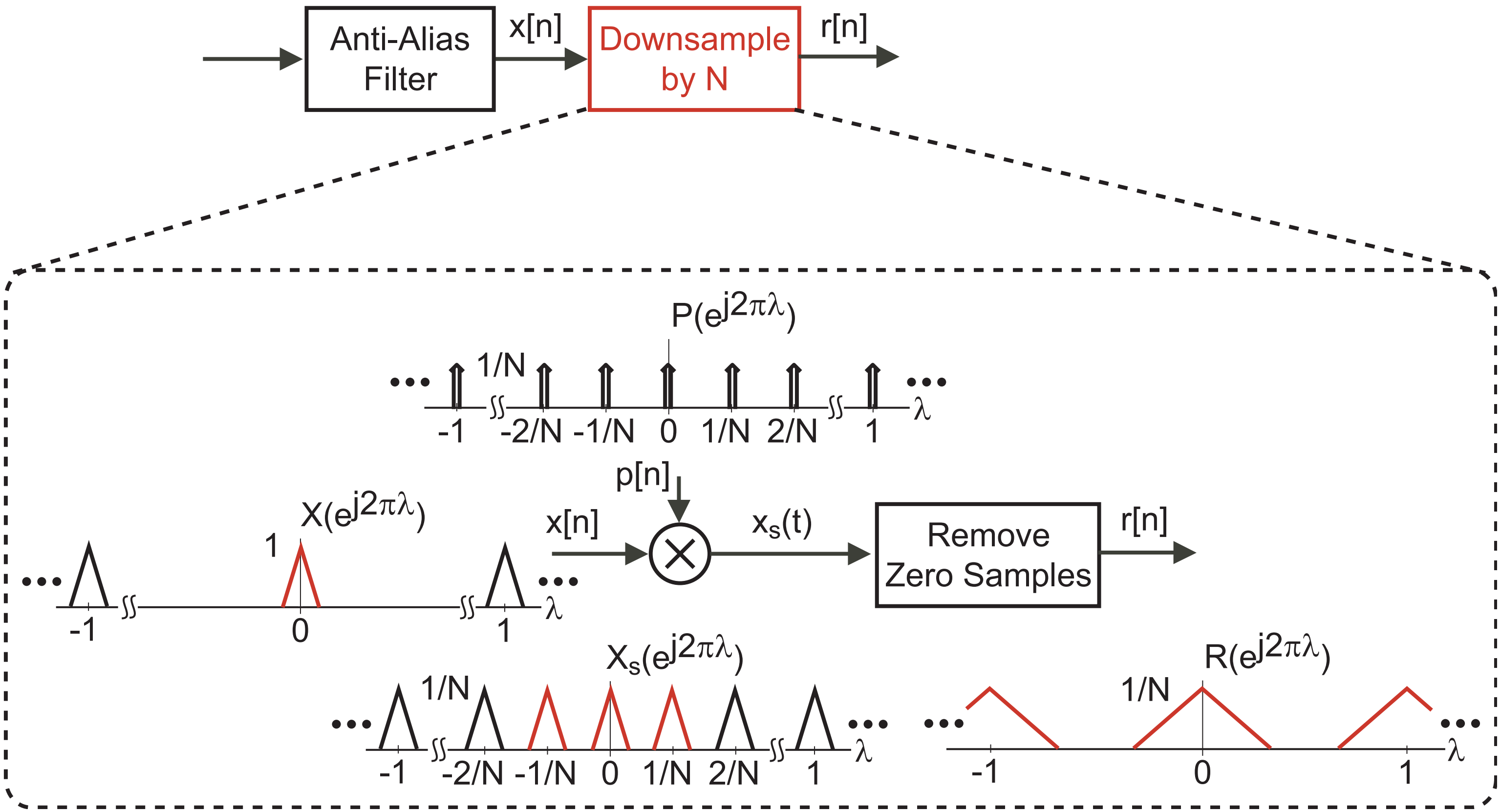
- **Analysis of A-to-D same as for sampler**
  - **For simplicity, we will ignore the influence of quantization noise in our picture analysis**
    - **In lab 4, we will explore the influence of quantization noise using Matlab**

Downsampling



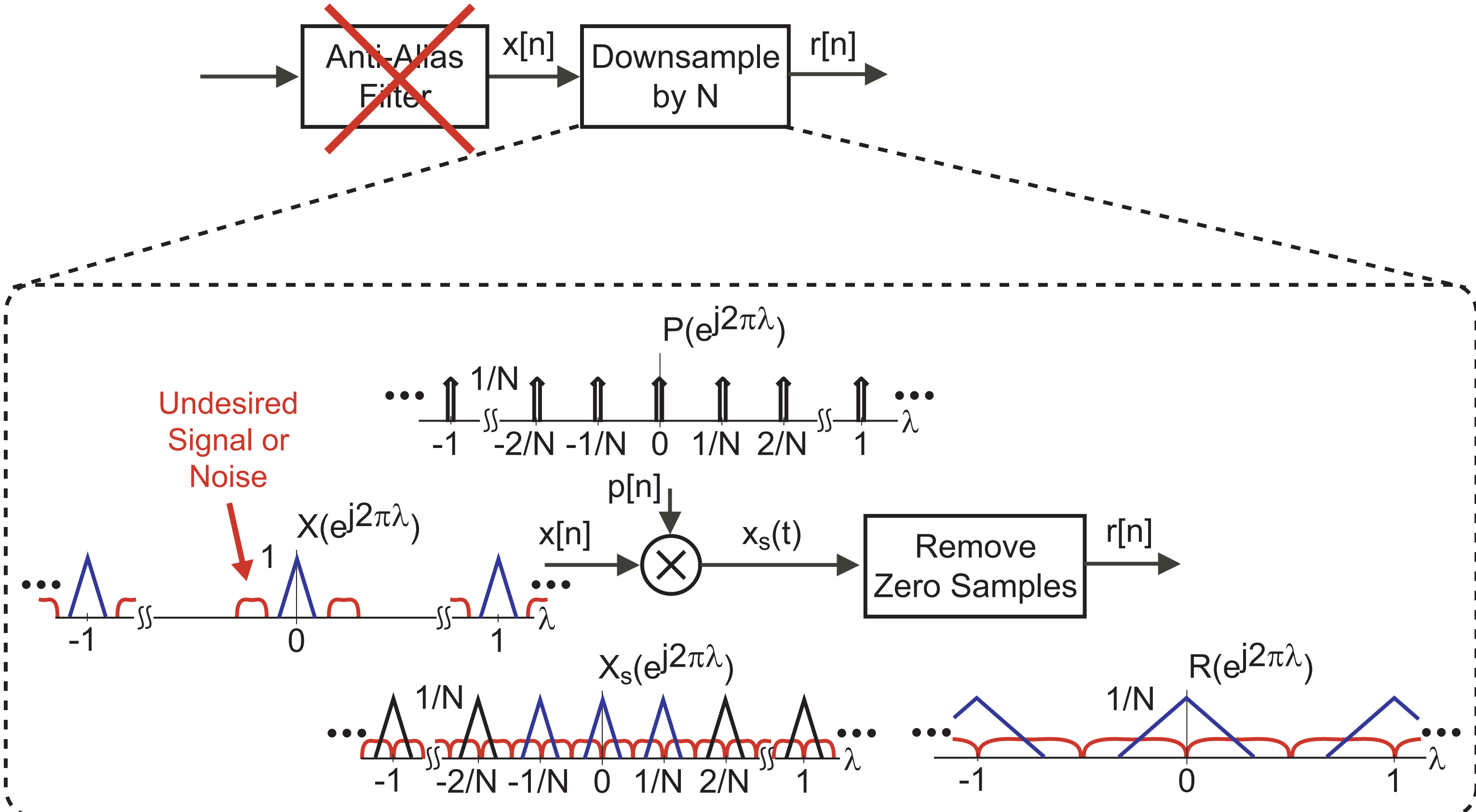
- Similar to sampling, but operates on *sequences*
- Analysis is simplified by breaking into two steps
  - *Multiply* input by impulse sequence of period *N* samples
  - Remove all samples of $x_s[n]$ associated with the zero-valued samples of the impulse sequence, *p[n]*
    - Amounts to *scaling* of time axis by factor *1/N*

Frequency Domain View of Downsampling



- **Multiplication by impulse sequence leads to replicas of input transform every *1/N Hz* in frequency**
- **Removal of zero samples (i.e., scaling of time axis) leads to scaling of frequency axis by factor *N***

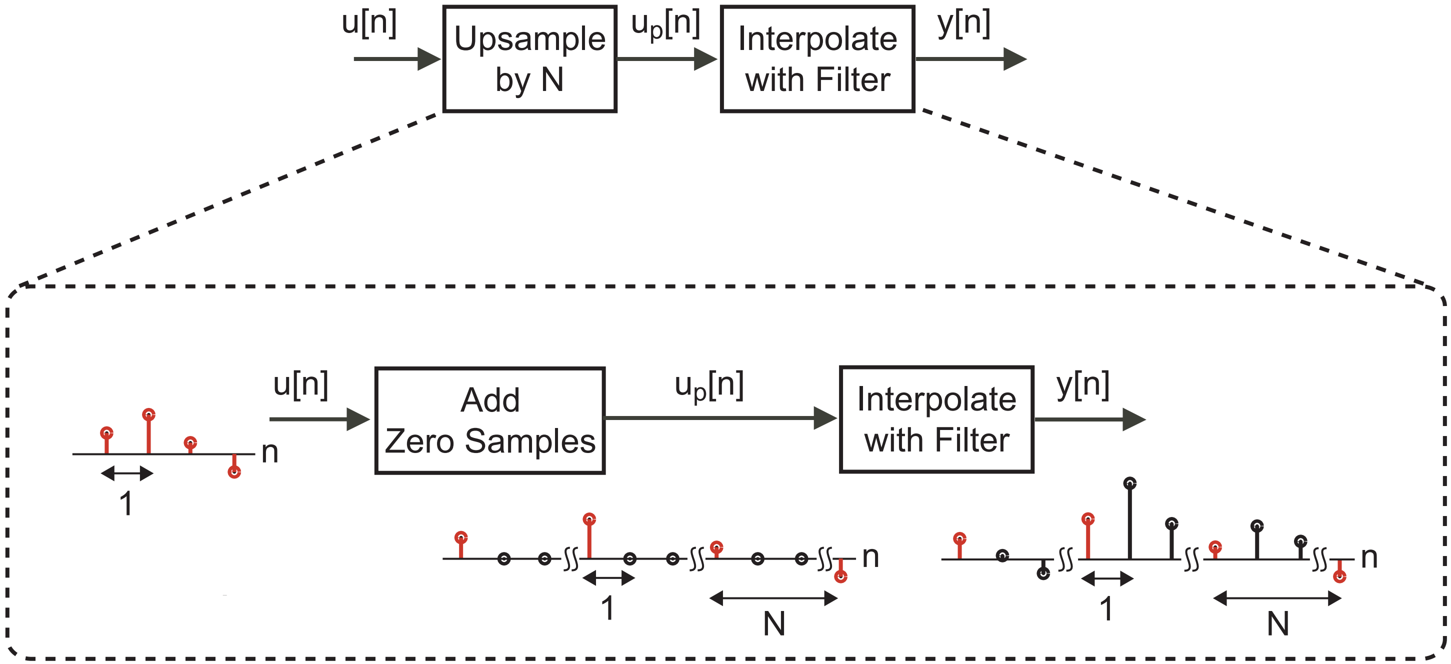
The Need for Anti-Alias Filtering



- Removal of anti-alias filter would allow undesired signals or noise to alias into desired signal band

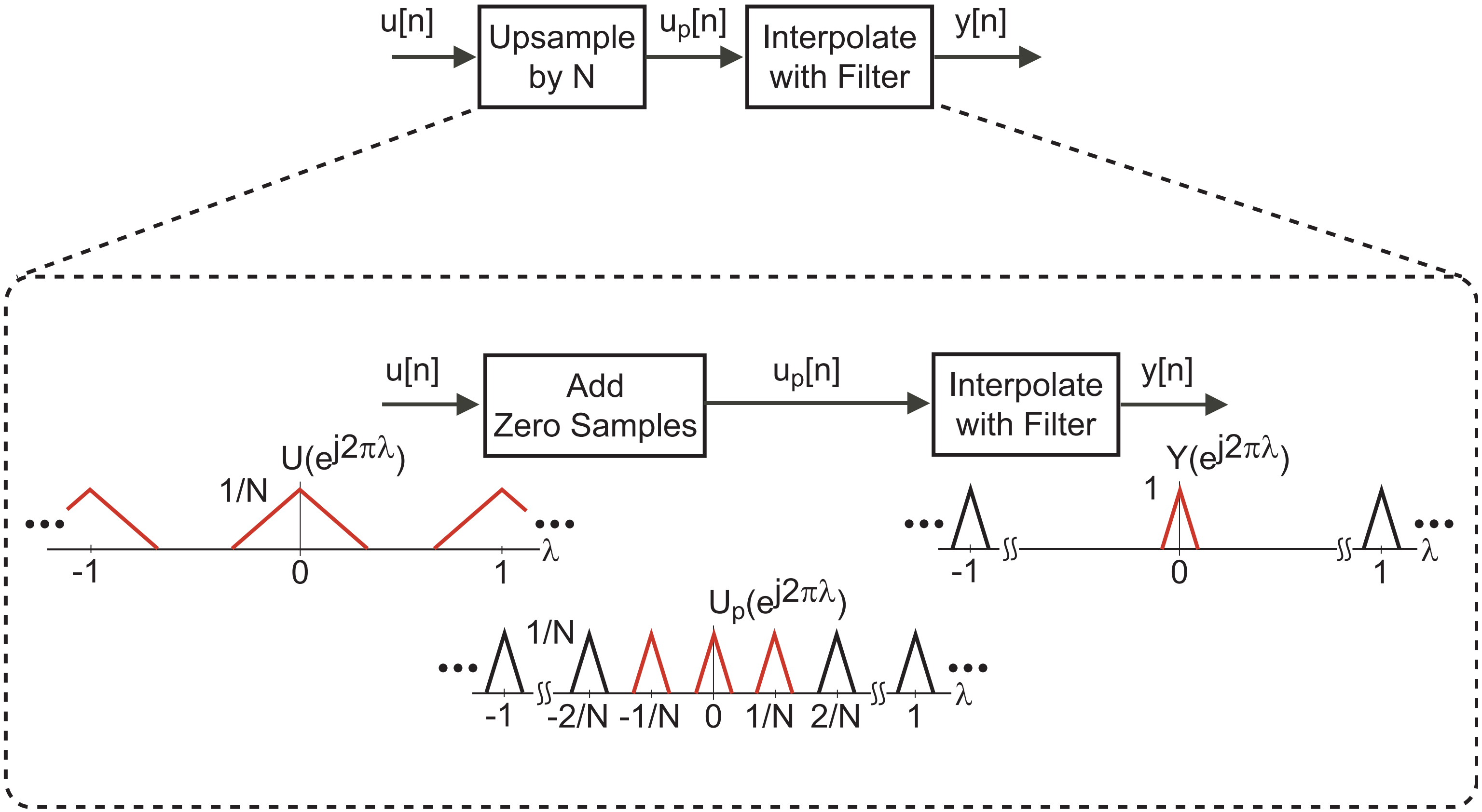
What is the appropriate bandwidth of the anti-alias lowpass filter?

Upsampler



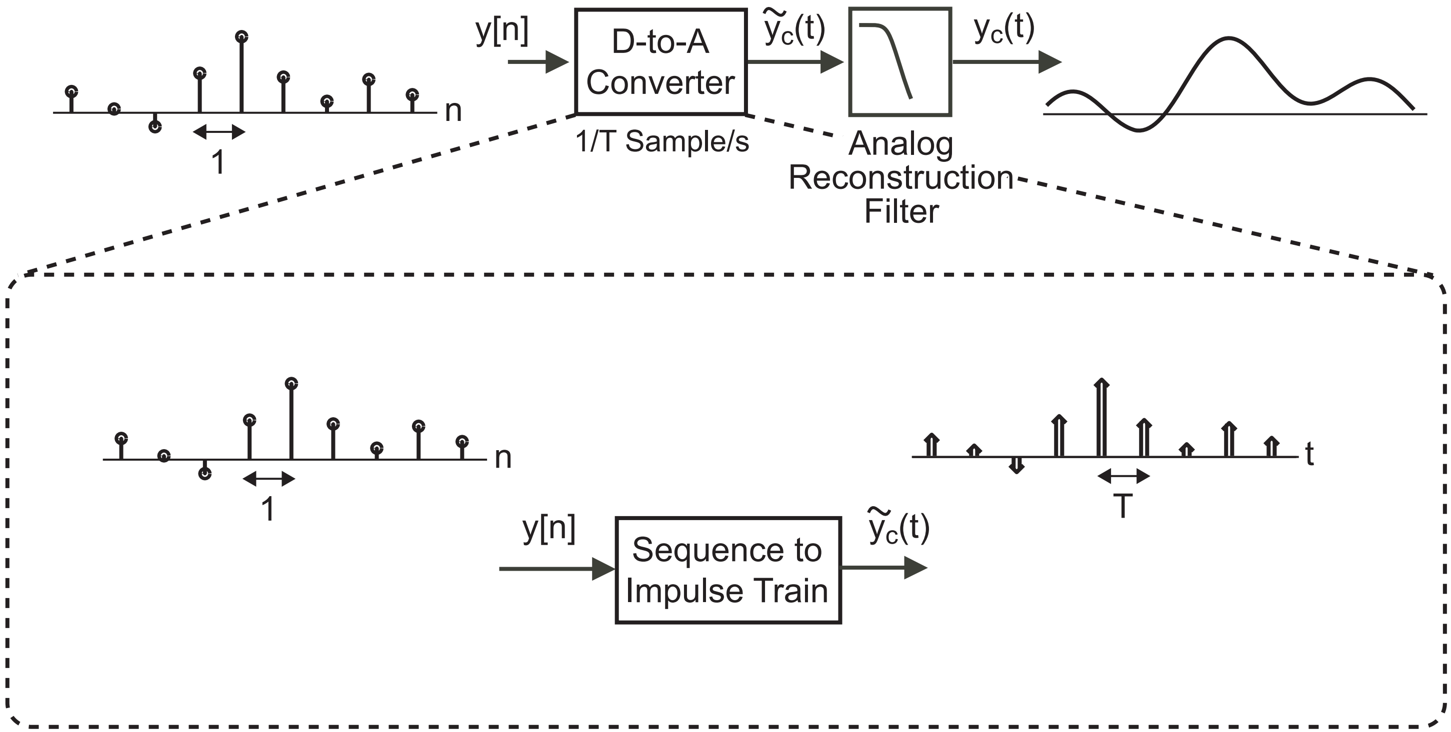
- **Consists of two operations**
  - **Add *N-1* zero samples between every sample of the input**
    - **Effectively scales time axis by factor *N***
  - **Filter the resulting sequence, $u_p[n]$, in order to create a *smoothly* varying set of sequence samples**
    - **Proper choice of the filter leads to *interpolation* between the non-zero samples of sequence $u_p[n]$ (discussed in Lab 5)**

Frequency Domain View of Upsampling



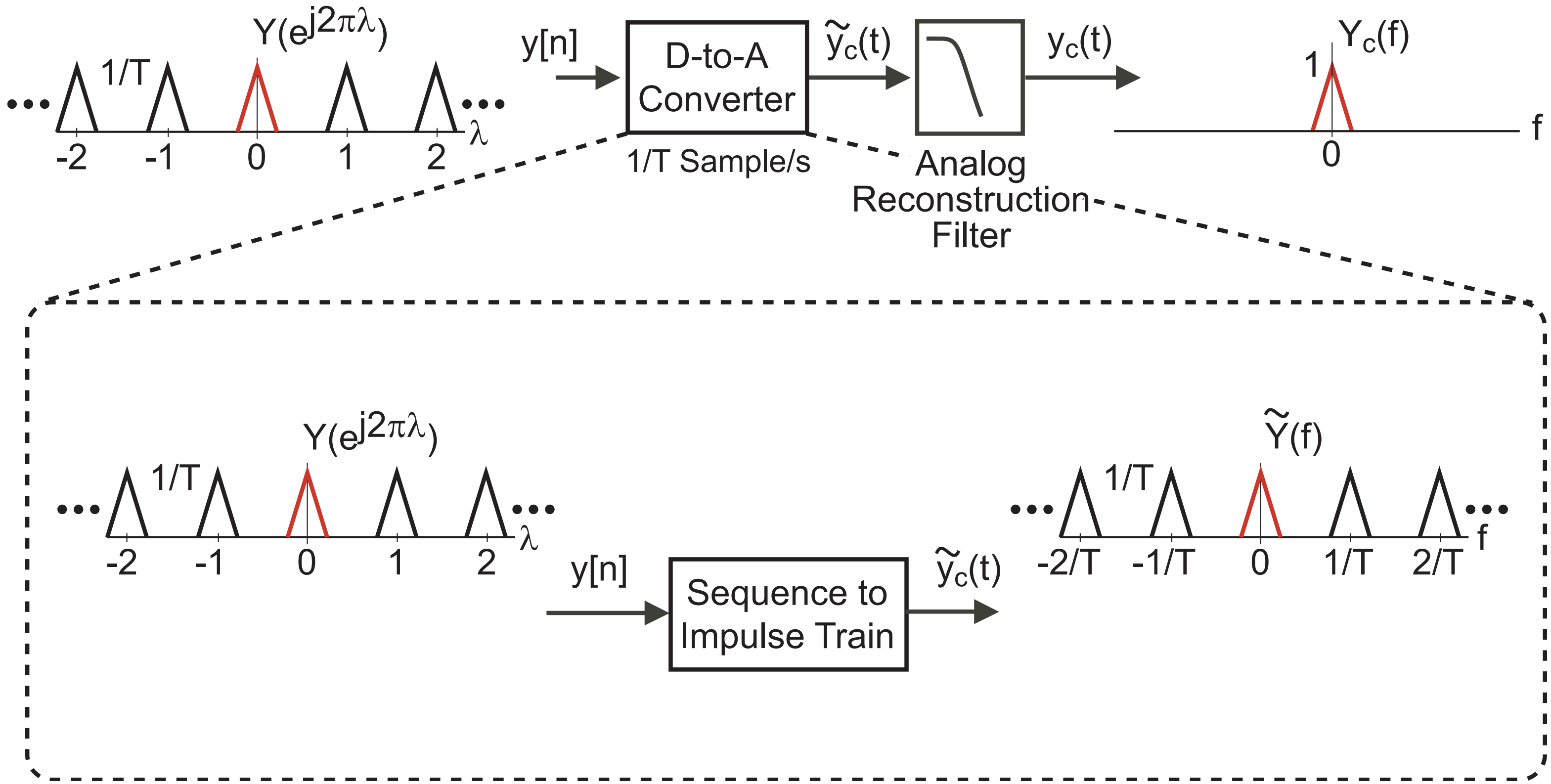
- Addition of zero samples (scaling of time axis) leads to scaling of frequency axis by factor *1/N*
- Interpolation filter removes all replicas of the signal transform *except* for the baseband copy

D-to-A Converter



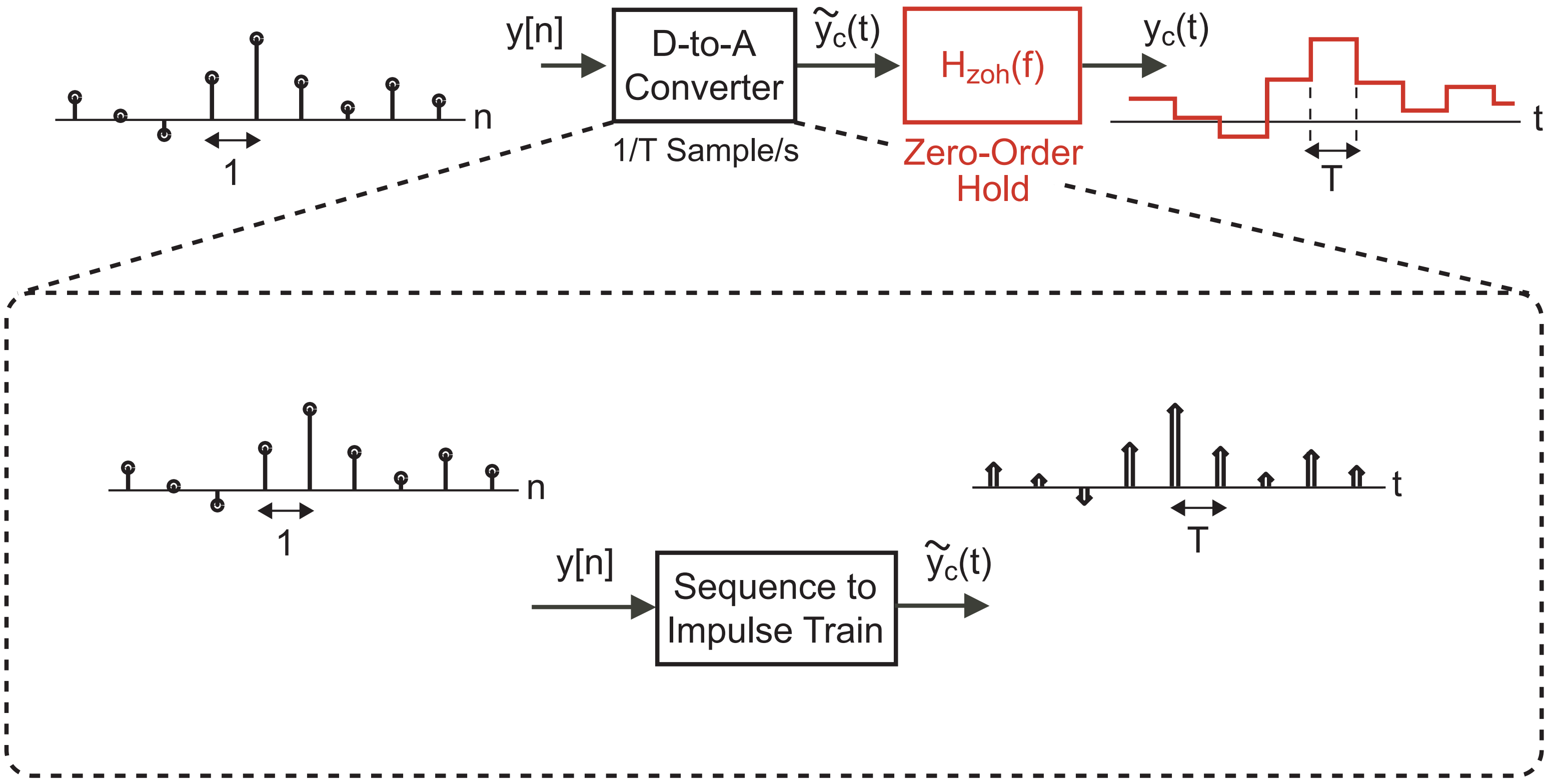
- **Simple analytical model includes two operations**
  - **Convert input sequence samples into corresponding impulse train**
  - **Filter impulse train to create a smoothly varying signal**
    - **Proper choice of the *reconstruction filter* leads to *interpolation* between impulse train values**

Frequency Domain View of D-to-A



- **Conversion from sequence to impulse train amounts to scaling the frequency axis by sample rate of D-to-A (*1/T*)**
- **Reconstruction filter removes all replicas of the signal transform *except* for the baseband copy**

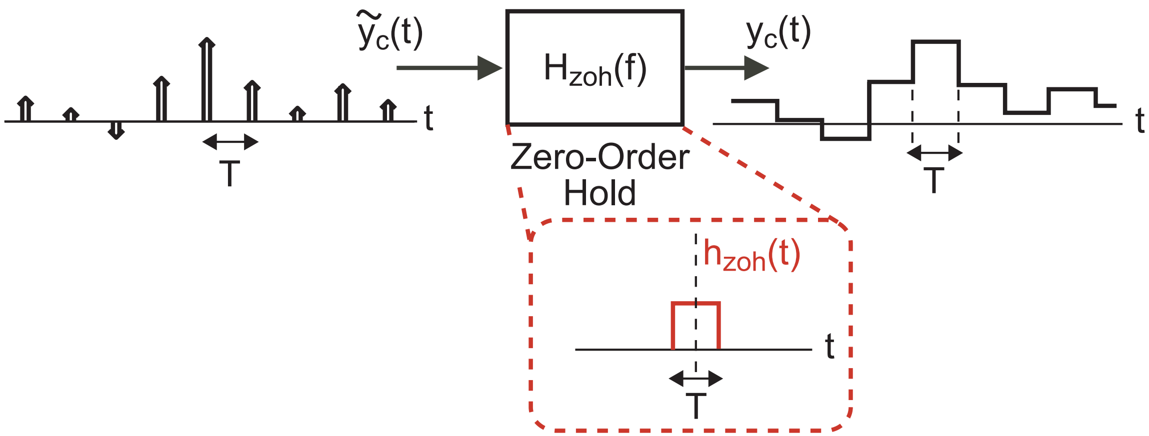
A Common Reconstruction Filter

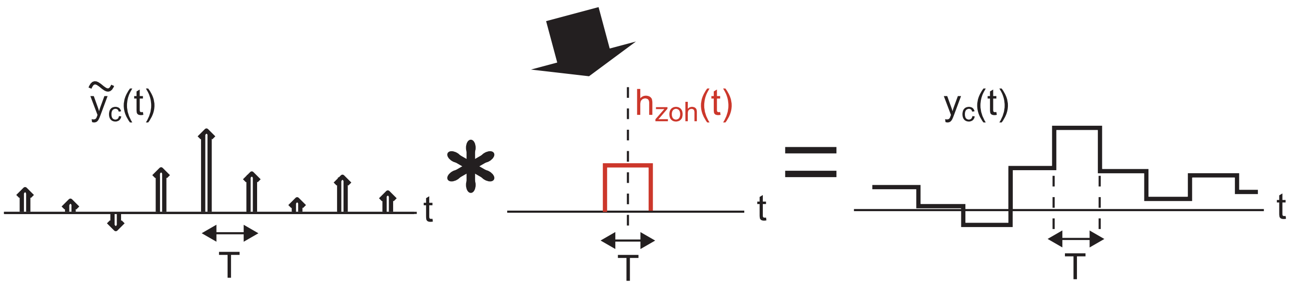


- Zero-order hold circuit operates by maintaining the impulse value across the D-to-A sample period
  - Easy to implement in hardware

How do we analyze this?

Filtering is Convolution in Time



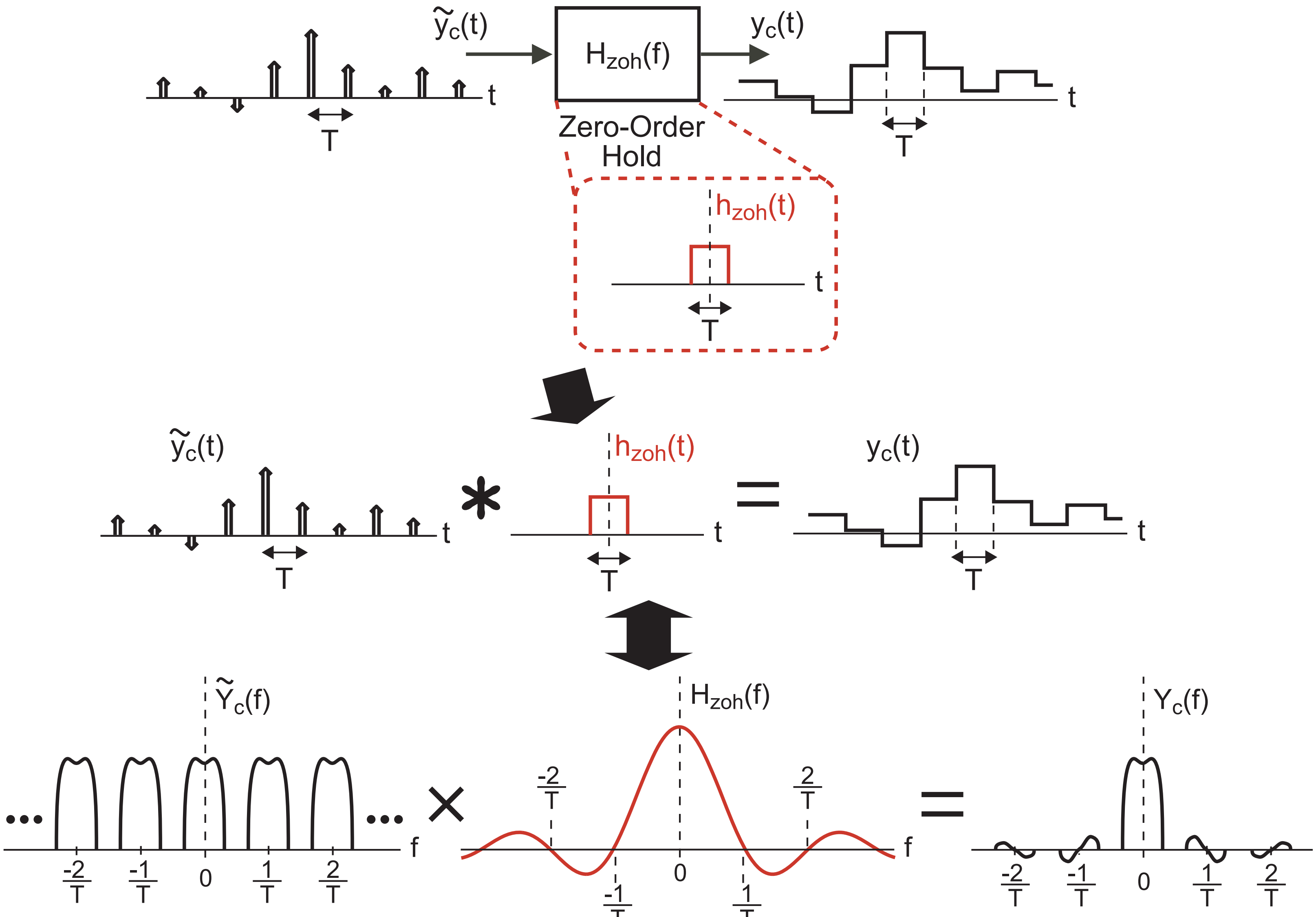


- **Recall that *multiplication* in *frequency* corresponds to *convolution* in *time***

$$x(t) * y(t) \;\Leftrightarrow\; X(f)Y(f)$$

- **Filtering corresponds to convolution in time between the input and the filter *impulse response***

Frequency Domain View of Filtering



- **Zero-order hold is not a great filter, but it's simple…**

Summary

- A-to-D converters convert continuous-time signals into sequences with discrete sample values
  - Operates with the use of sampling and quantization
- D-to-A converters convert sequences with discrete sample values into continuous-time signals
  - Analyzed as conversion to impulse train followed by reconstruction filtering
    - Zero-order hold is a simple but low performance filter
- Upsampling and downsampling allow for changes in the effective sample rate of sequences
  - Allows matching of sample rates of A-to-D, D-to-A, and digital processor
  - Analysis: downsampler/upsampler similar to A-to-D/D-to-A
- Up next: digital modulation