Sampling and reconstruction

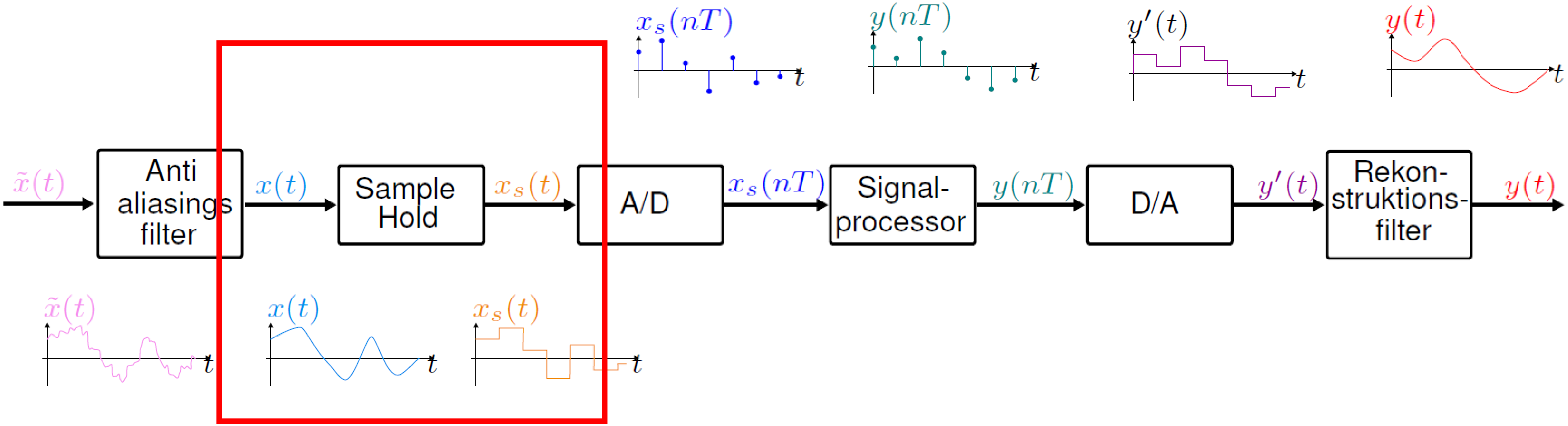
Zhuoqi Cheng
zch@mmmi.sdu.dk
SDU Robotics

SDU

Topics to be covered in this course

→ Sampling and reconstruction
→ Aliasing
→ Quantization and dynamic range
→ Implementation
→ Conversion time-frequency domain
→ Z transform
→ Linear Time Invariant system (LTI)
→ System analysis
→ Window functions
→ Filter design
→ Impulse response (FIR and IIR)

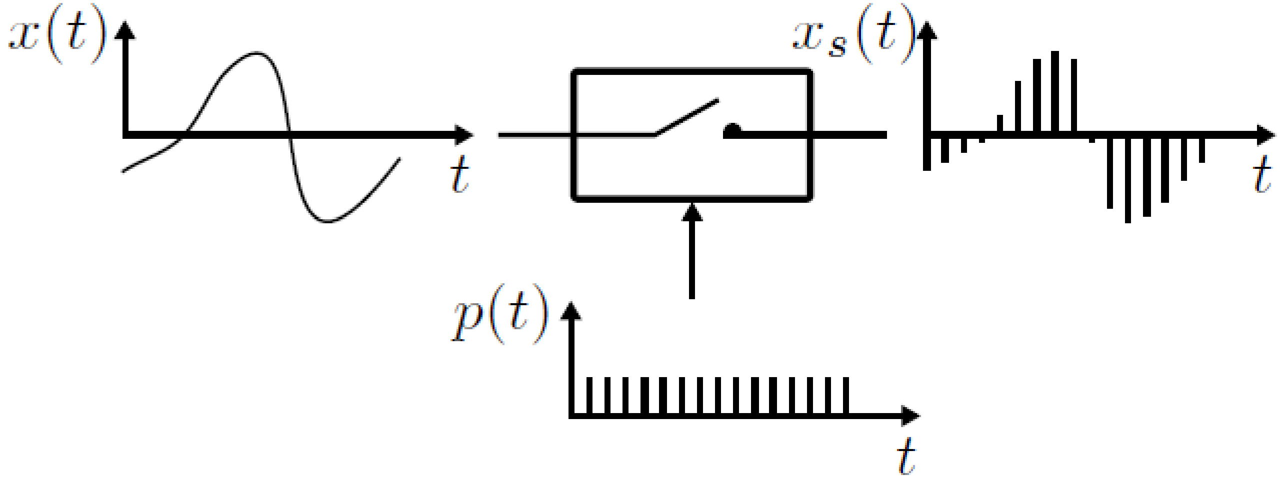
Digital signal processing workflow



Continuous– discrete (C/D) converter

The input signal to the sampler is the time-continuous signal $x_c(t)$
the sampler's output signal $x_s(t)$ is time-discrete (a periodic pulse train).
The sample switch is controlled by the pulse sampling signal $p(t)$.

Definition
T: sampling period (interval)
$f_s = \frac{1}{T}$: sampling frequency (Hz)



Time domain

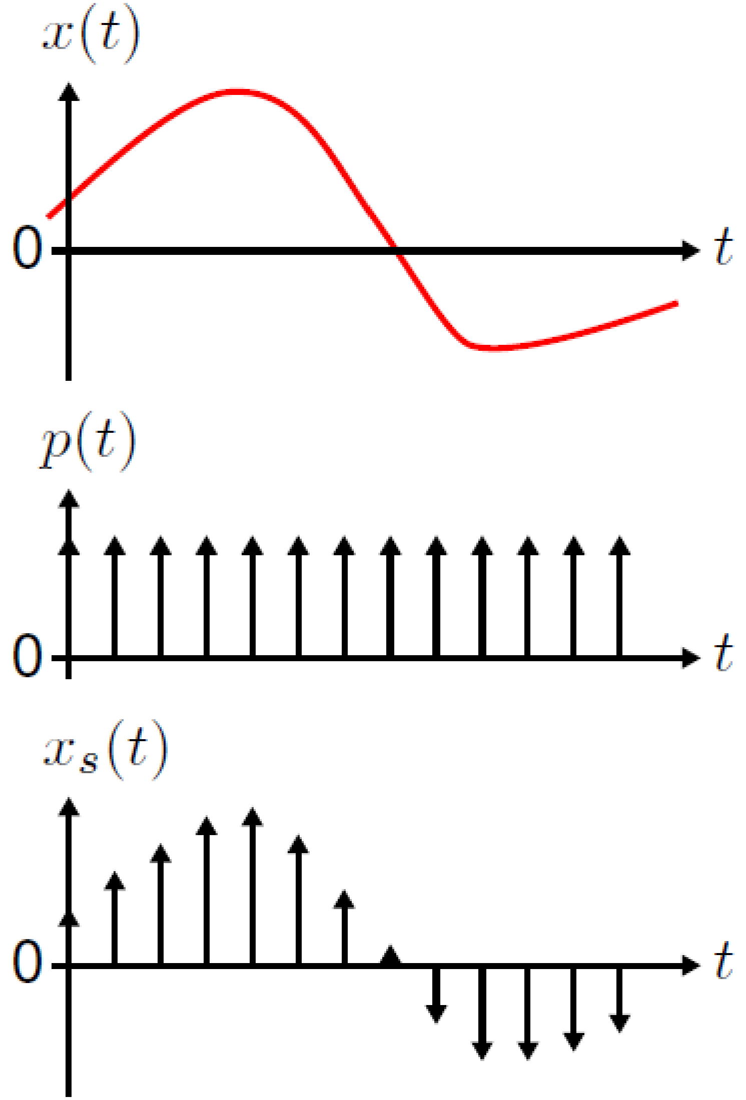
The sampling signal $p(t)$ can be written as
$p(t) = \sum_{n=-\infty}^{\infty} \delta \quad (t - nT)$

The sampled signal can be described as

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

It can be rewritten as

$x_s(n) = \sum_{n=-\infty}^{\infty} x(nT)\delta \quad (t - nT)$



Frequency domain

Recall:
For T-periodic signal $\mathrm{x}(t)$, it can be represented by **Fourier series:**

$$x(t) = \sum_{m=-\infty}^{\infty} C_m e^{jm\omega_0 t}$$

where the **Fourier coefficient**

$$C_m = \frac{1}{T}\int_0^T x(t) e^{-j\omega_0 t}\, dt$$

$\omega_0 = \frac{2\pi}{T}$ is the signal frequency

Applying the above on the pulse sampling signal $\mathrm{p}(t)$, we can have

$$\boxed{p(t) = \sum_{n=-\infty}^{\infty} \delta\ (t - nT) = \sum_{m=-\infty}^{\infty} C_m e^{jm\omega_s t} = \frac{1}{T}\sum_{m=-\infty}^{\infty} e^{jm\omega_s t}}$$

where $\omega_s = 2\pi f_s$

Also, we can see that $C_m = \frac{1}{T}$

From the previous slide, the impulse sampled signal $x_s(t)$ can be written as

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

Through Fourier transform

$$X_s(\omega_s) = \mathcal{F}\big(x_s(t)\big) = \sum_{n=-\infty}^{\infty} x(nT)\, e^{-j\omega_s nT} = \frac{1}{T}\sum_{m=-\infty}^{\infty} X(\omega - m\omega_s)$$

Rewrite

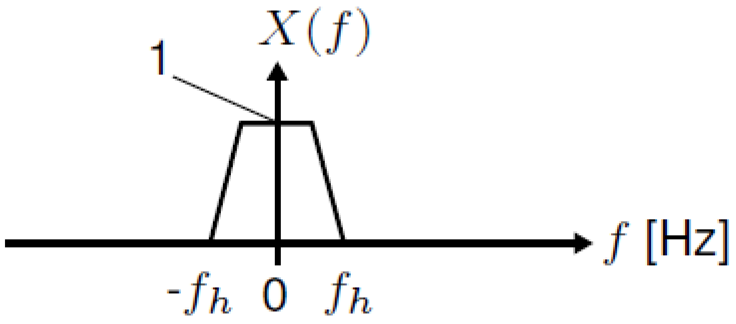
$$X_s(f) = \frac{1}{T}\sum_{m=-\infty}^{\infty} X(f - mf_s)$$

What does this equation mean??

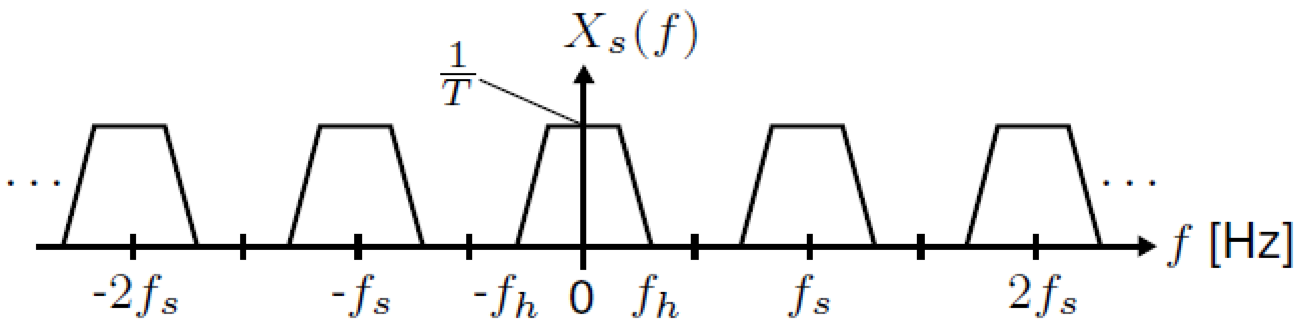
Continuous signal after pulse sampling:
- The original spectrum $X(f)$ is repeated an infinite number of times with distance f_s
- The amplitude is scaled by a factor 1/T .

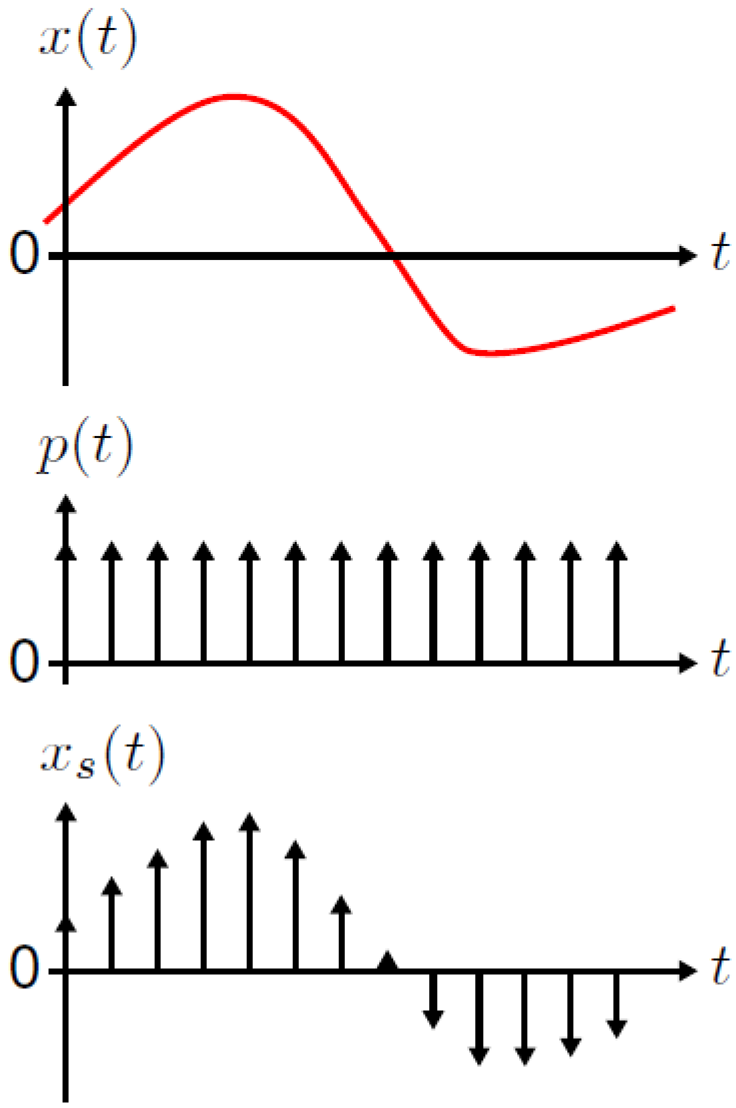
Graphical explanation

Example spectrum of continuous signal $X(f)$



After impulse sampling, the amplitude spectrum becomes $X_s(f)$





Relationship btw time domain and freq domain

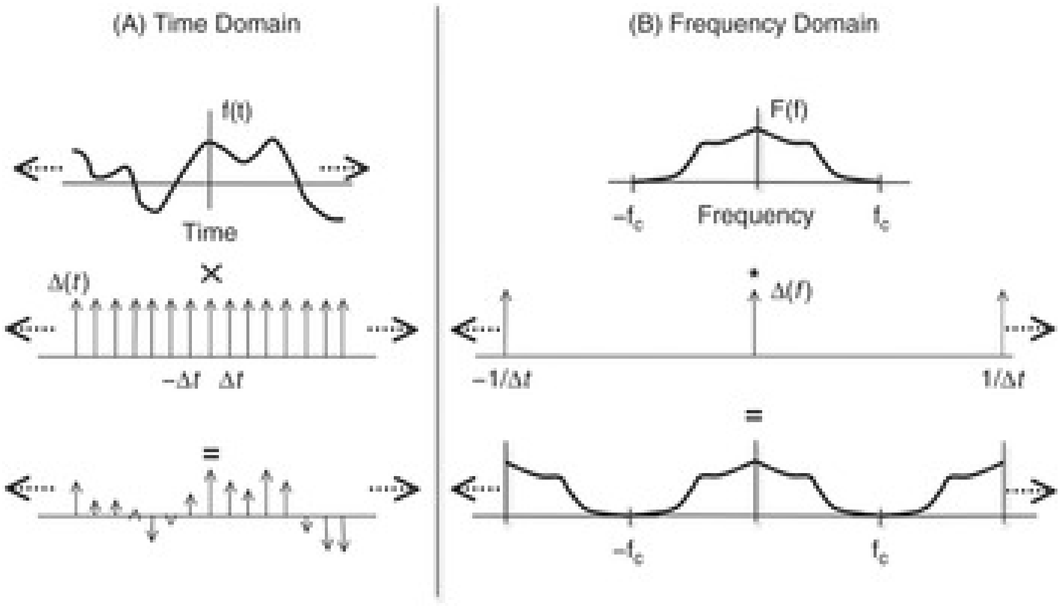
Discrete signal in time domain -> periodic frequency spectra

Periodic signal in time domain -> discrete frequency spectra

A signal that is time-limited -> it must have infinite spectral content.

A signal that is band-limited -> it must extend infinitely in time.

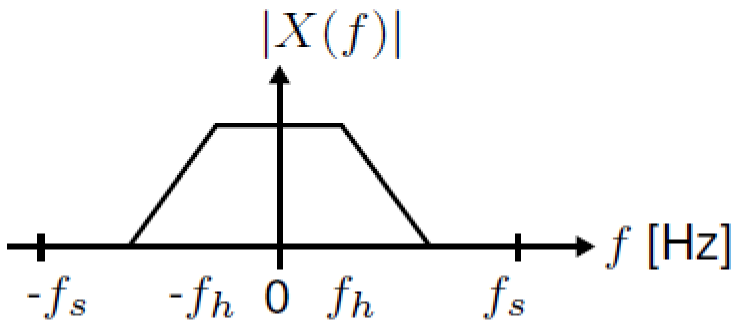
Multiply in frequency domain -> Convolution in time domain
Convolution in frequency domain -> Multiply in time domain



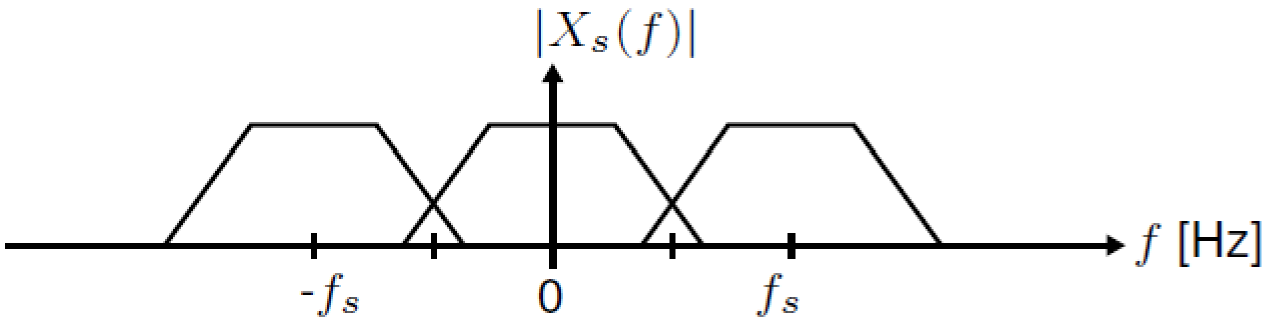
Band limited

How should the sampling frequency f_s be chosen so that the original spectrum can be recovered from the sampled signal?

Let’s consider the follow example continuous signal spectrum

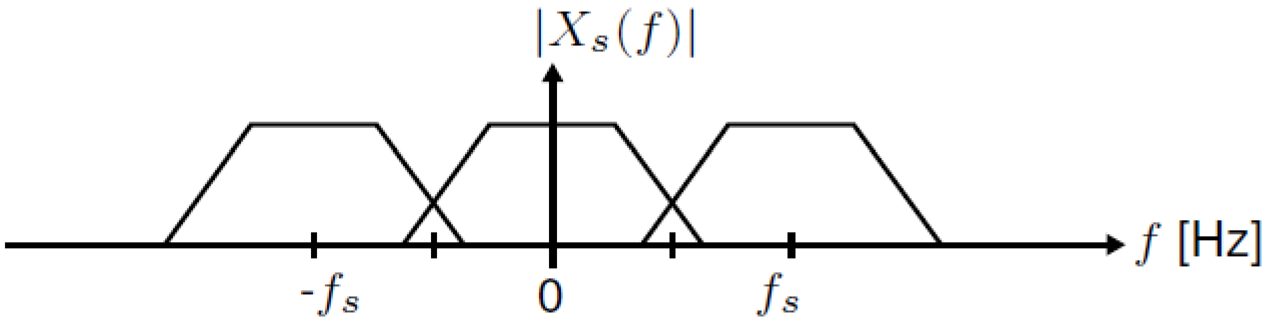


If the sample frequency is chosen too low, the repeated spectra overlap, and thus **aliasing** occurs.

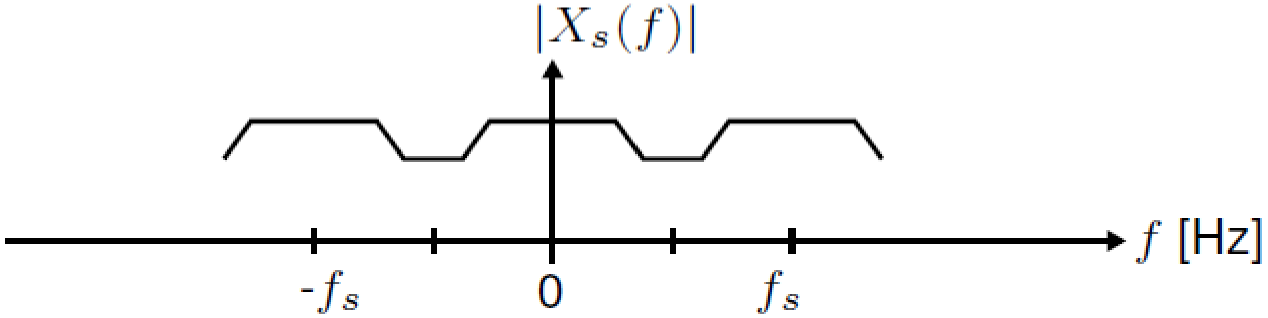


SDU

Aliasing



The resulting amplitude spectrum for $x_s(t)$ becomes as below.



This is called the **aliasing noise.**

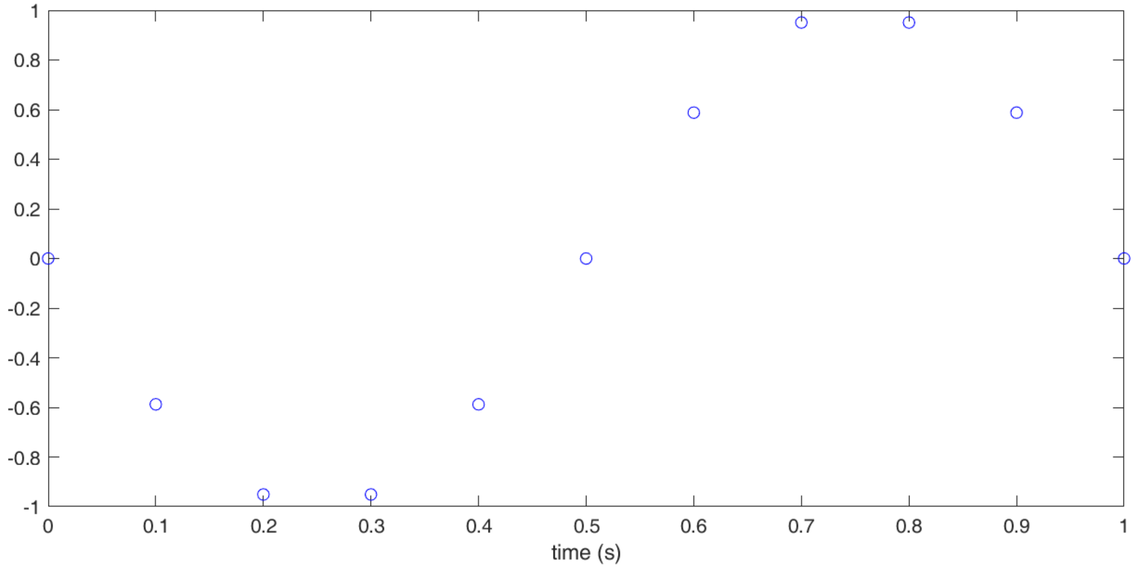
SDU

Nyquist-Shannon Theorem

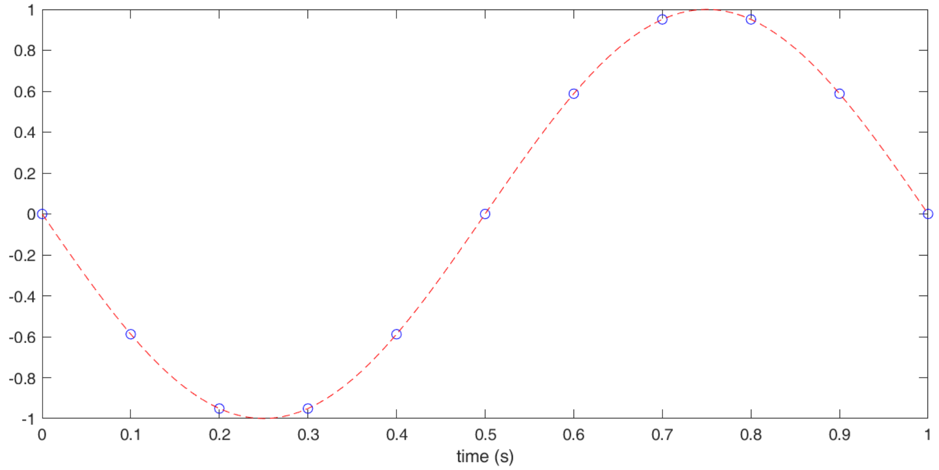
A time-continuous signal $x(t)$ can only be correctly reconstructed from $x_s(t)$ if the sample frequency is at least twice the highest frequency in the spectrum for $x(t)$.

The input discrete signal is as follows. Assuming that it is a sinusoidal signal.

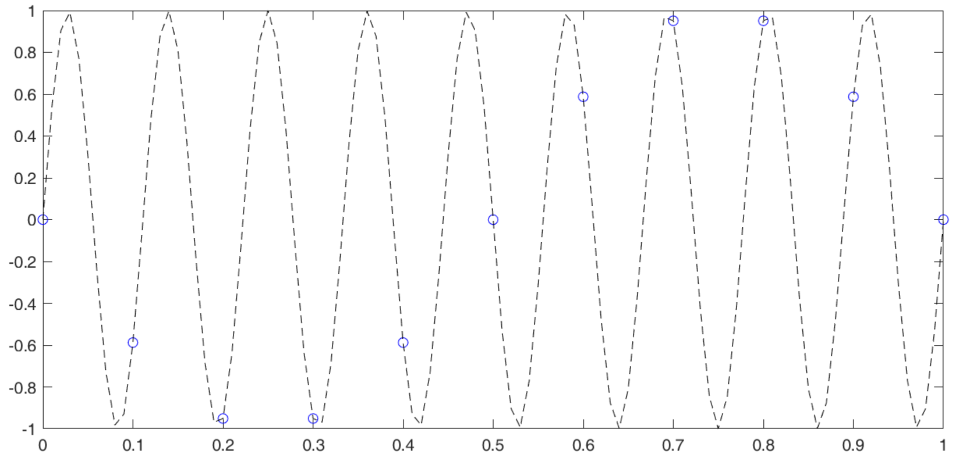
Q1: what is the sampling frequency f_s?



Q2: what is your guess of the signal's frequency?



f=1Hz

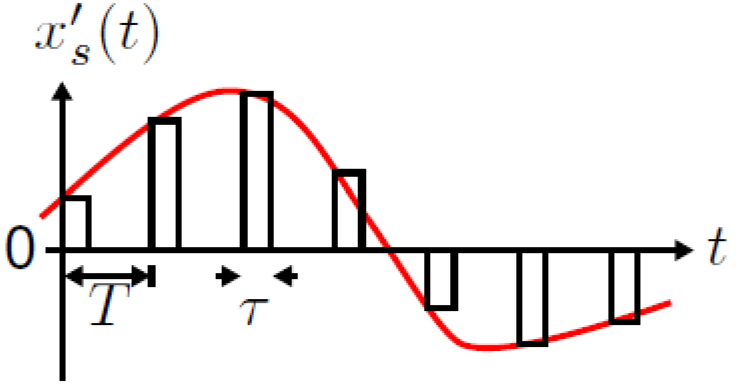


f=9Hz

SDU

Pulse sampling

Impulse sampling cannot be realized in practice, as the pulse width will be greater than zero.
In practice, we should consider a pulse sampled signal as shown below.

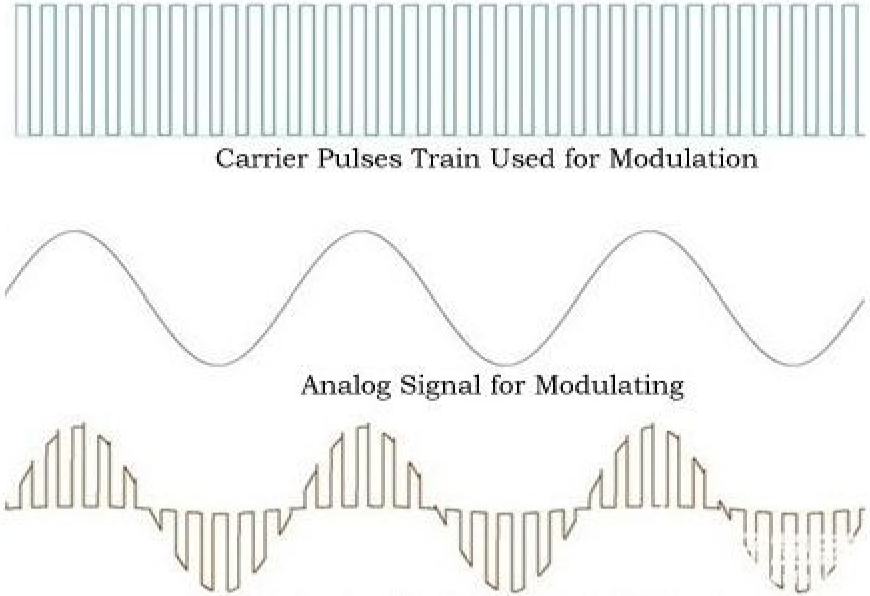


We use T to denote sampling interval [s], and τ to denote the pulse width [s].

The duty factor of the pulse sampled signal $d = \frac{\tau}{T}$

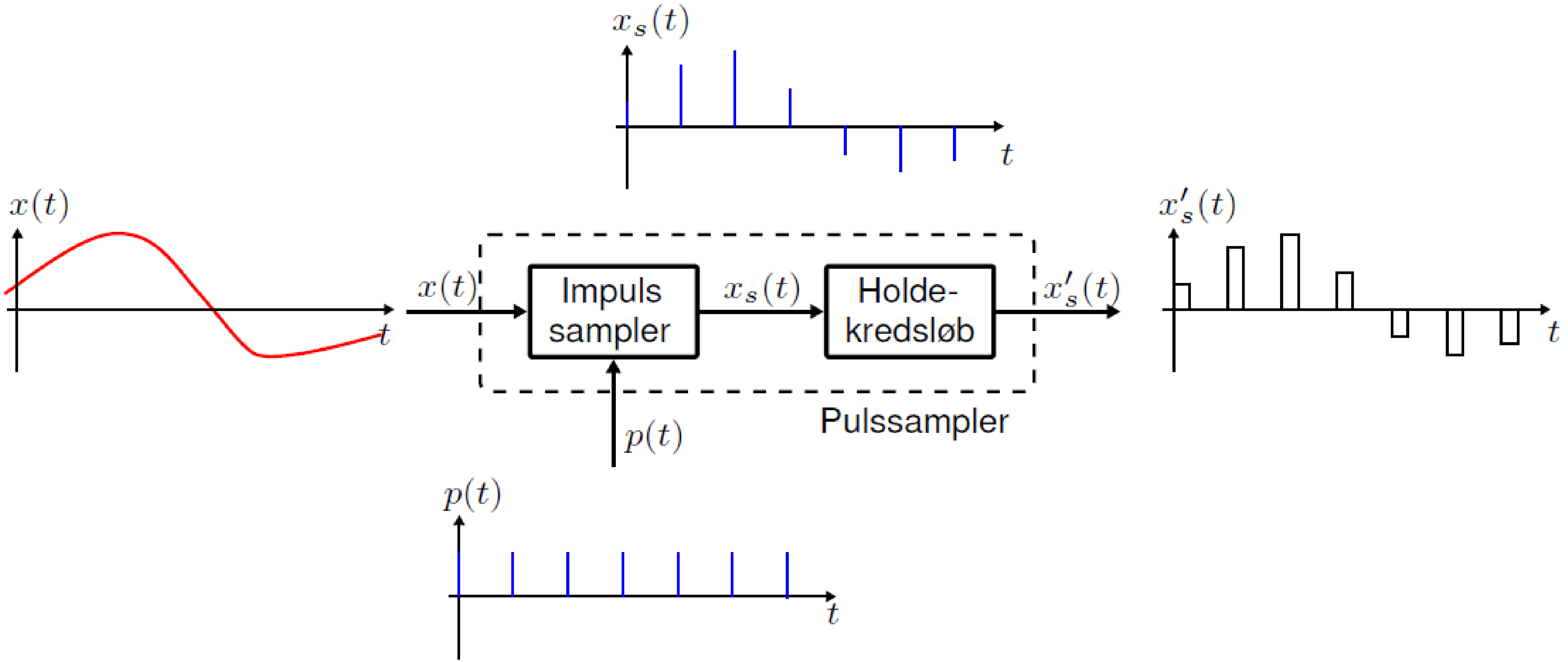
Different types of pulse modulators

→ **Pulse Amplitude Modulation**
→ Pulse Width Modulation
→ Pulse Position Modulation
→ Pulse Density Modulation
→ etc



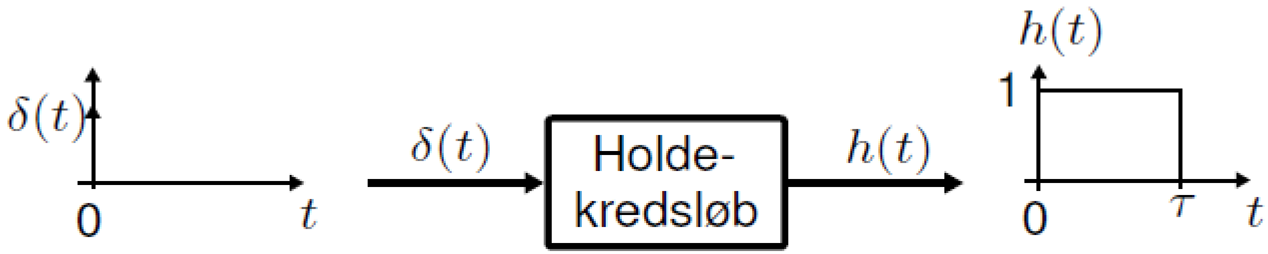
Pulse modulation

A pulse sampler can be modeled using a sample-and-hold circuit



Holding circuit

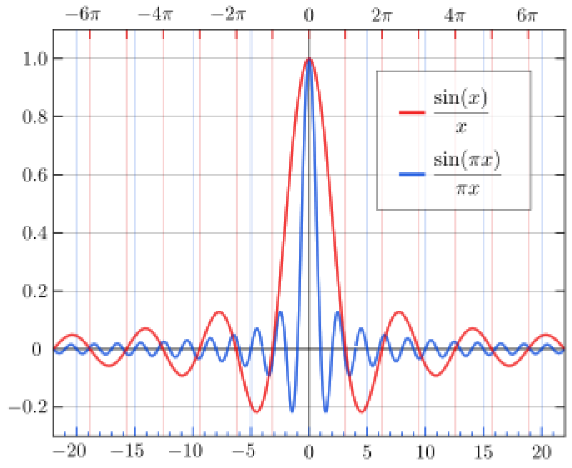
The operation of a pulse sampler with pulse width τ is shown as follows.



It is known that the spectrum function for a **rectangular pulse** is

$$F(\omega) = \mathcal{F}\big(f(t)\big) = \tau \, \mathrm{sinc}(\omega \frac{\tau}{2})$$

$$\mathrm{sinc}(x) = \frac{\sin x}{x}$$



Pulse Modulation

$$x_s{}'(t) = x(t)p(t)\mathrm{f}(\mathrm{t})$$

Frequency spectrum of the pulse sampled signal is $\boxed{X\ \ (f)}$ $\boxed{F(f)}$ $\boxed{P(f)}$

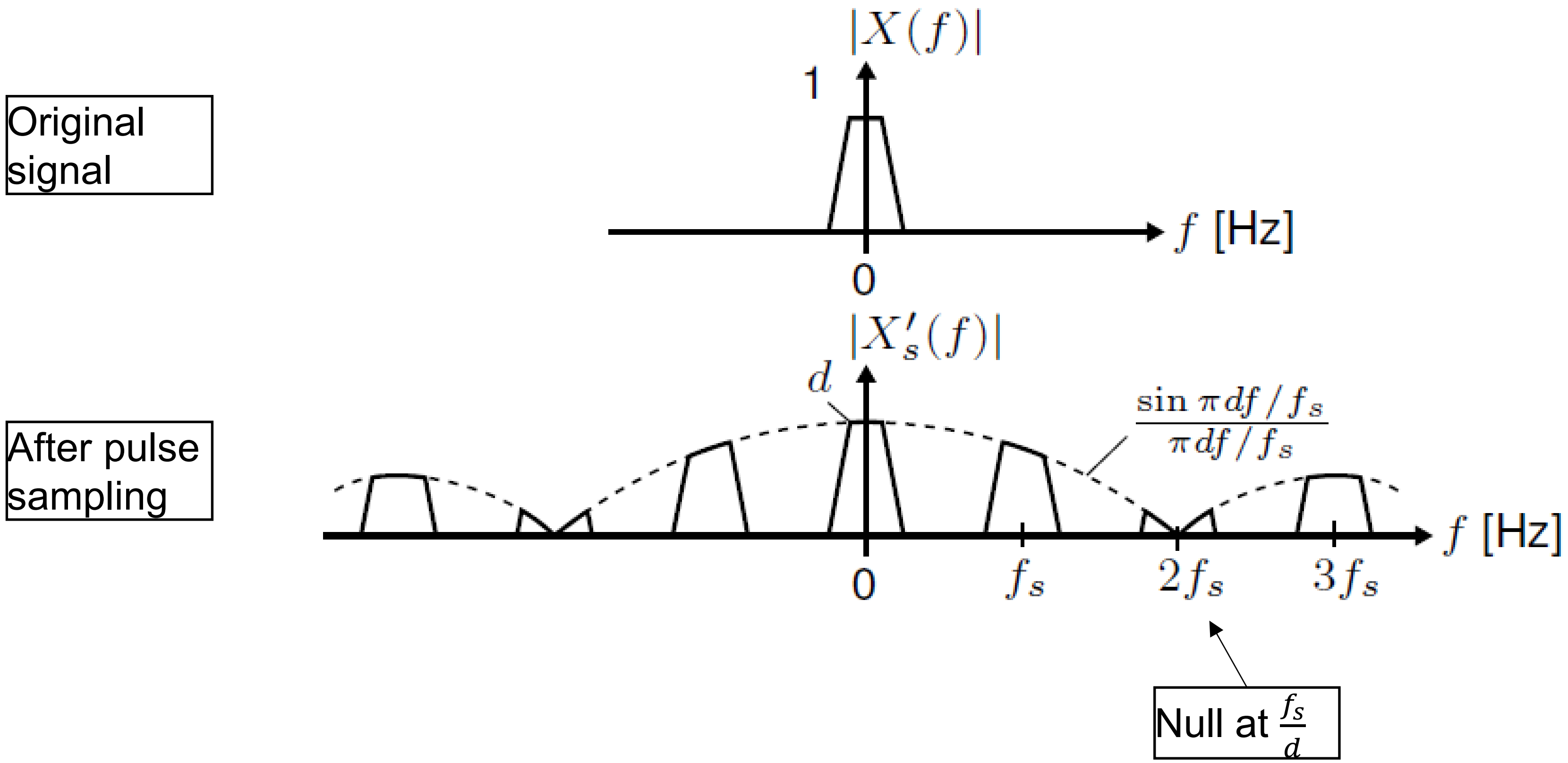
$$X_s'(f) = \frac{1}{T}\sum_{m=-\infty}^{\infty} X(f - mf_s)\,\tau\ \mathrm{sinc}(\pi d\frac{f}{f_s})e^{-j\pi d\frac{f}{f_s}}$$

Amplitude spectrum thus becomes

$$|X_s'(f)| = d\left|\mathrm{sinc}\left(\pi d\frac{f}{f_s}\right)\right|\sum_{m=-\infty}^{\infty} X(f - mf_s)$$

where duty factor of the pulse sampled signal $d = \frac{\tau}{T}$

Pulse-sampling signal with $d = 0.5$

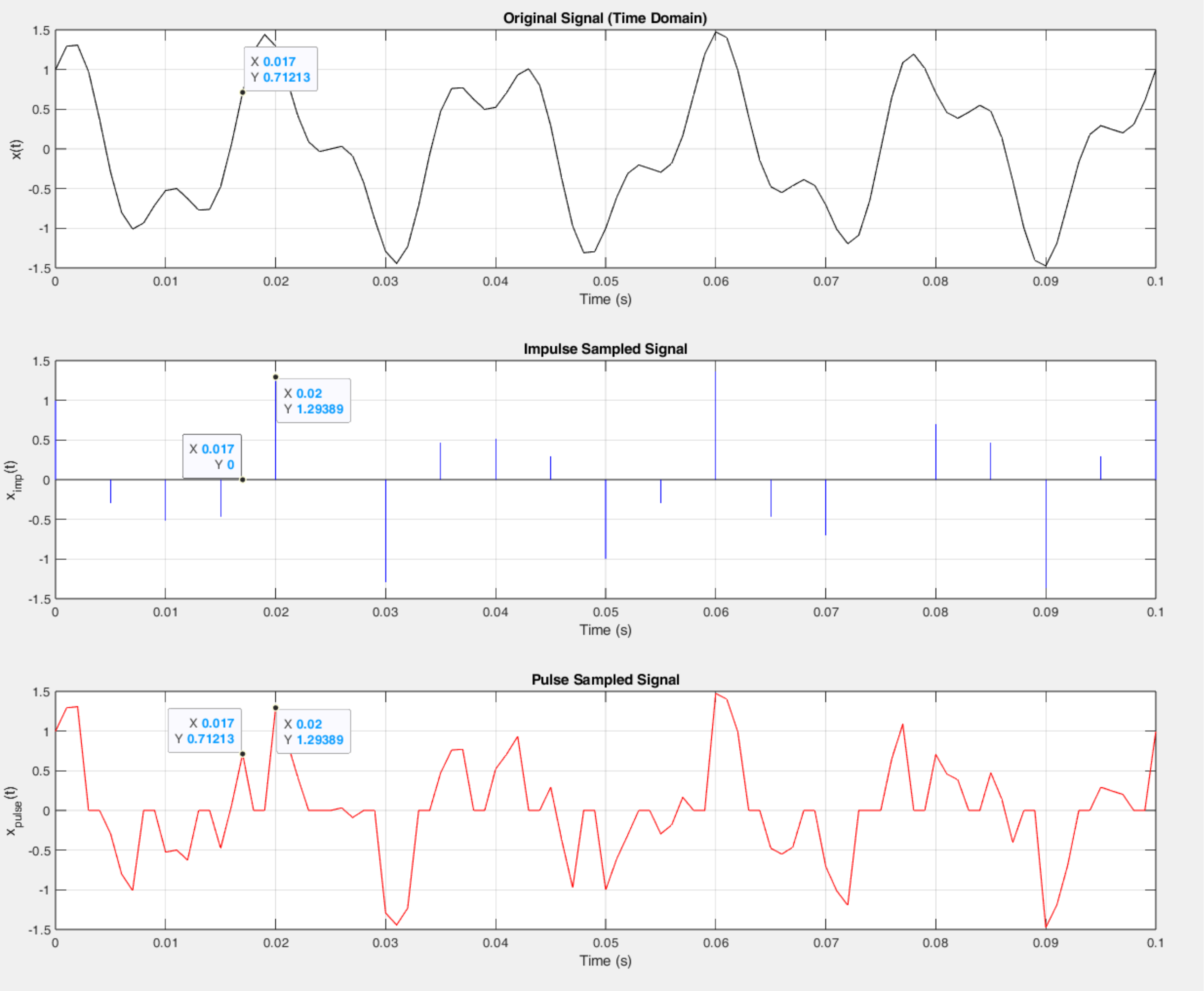


Example

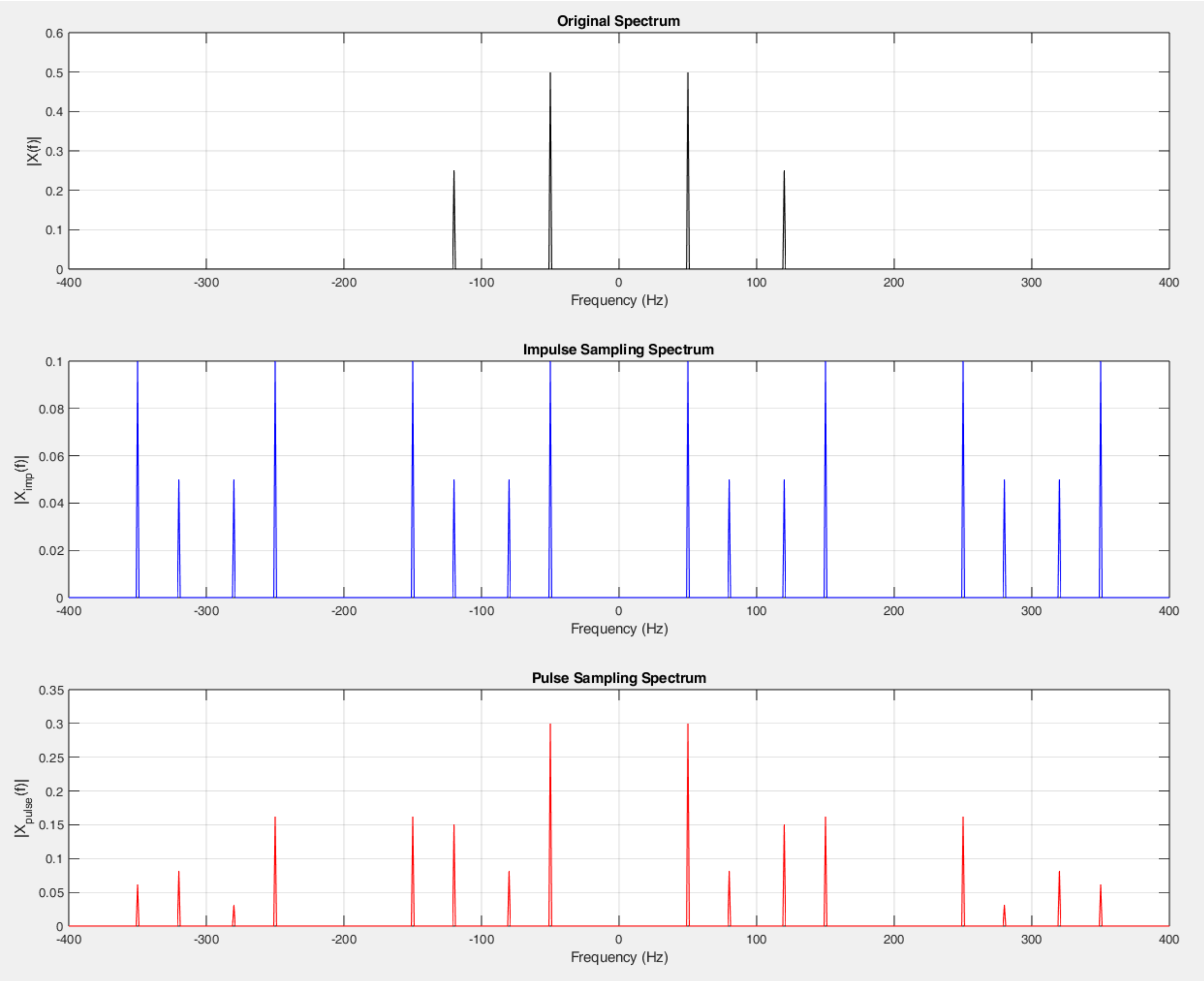
1. Use Matlab to generate a 50Hz + 120Hz sine wave;
2. Impulse sampling at 200Hz, show its amplitude spectrum.
3. Pulse sampling at 200Hz and duty factor 1/2, show its amplitude spectrum.

```
fs = 1000; % "Continuous-time" reference sampling rate (Hz)
T = 1/fs;
t = 0:T:1-T; % 1 second duration
N = length(t);
f = (-N/2:N/2-1)*(fs/N);

% Test Signal (sum of sinusoids)
x = cos(2*pi*50*t) + 0.5*sin(2*pi*120*t); % signal with 50Hz and 120Hz components

% Impulse Sampling
fs_sample = 200; % Sampling frequency (Hz)
Ts = 1/fs_sample;
impulse_train = zeros(size(t));
impulse_train(1:fs/fs_sample:end) = 1; % Dirac comb
x_imp = x .* impulse_train; % Impulse-sampled signal

% Pulse Sampling (with duty factor)
duty = 0.5; % duty cycle (50%)
pulse_train = square(2*pi*fs_sample*t, duty*100);
pulse_train = (pulse_train+1)/2; % Make it 0/1 rectangular pulses
x_pulse = x .* pulse_train; % Pulse-sampled signal

% FFT
X = abs(fftshift(fft(x)))/N;
X_imp = abs(fftshift(fft(x_imp)))/N;
X_pulse = abs(fftshift(fft(x_pulse)))/N;
```

```
% Plot Time Domain
figure;
subplot(3,1,1);
plot(t, x, 'k');
title('Original Signal (Time Domain)');
xlabel('Time (s)'); ylabel('x(t)'); grid on; xlim([0 0.1]);

subplot(3,1,2);
stem(t, x_imp, 'b', 'Marker','none');
title('Impulse Sampled Signal');
xlabel('Time (s)'); ylabel('x_{imp}(t)'); grid on; xlim([0 0.1]);

subplot(3,1,3);
plot(t, x_pulse, 'r');
title('Pulse Sampled Signal');
xlabel('Time (s)'); ylabel('x_{pulse}(t)'); grid on; xlim([0 0.1]);

% Plot Frequency Spectrum
figure;
subplot(3,1,1);
plot(f, X, 'k');
title('Original Spectrum');
xlabel('Frequency (Hz)'); ylabel('|X(f)|'); grid on; xlim([-400 400]);

subplot(3,1,2);
plot(f, X_imp, 'b');
title('Impulse Sampling Spectrum');
xlabel('Frequency (Hz)'); ylabel('|X_{imp}(f)|'); grid on; xlim([-400 400]);

subplot(3,1,3);
plot(f, X_pulse, 'r');
title('Pulse Sampling Spectrum');
xlabel('Frequency (Hz)'); ylabel('|X_{pulse}(f)|'); grid on; xlim([-400 400]);
```



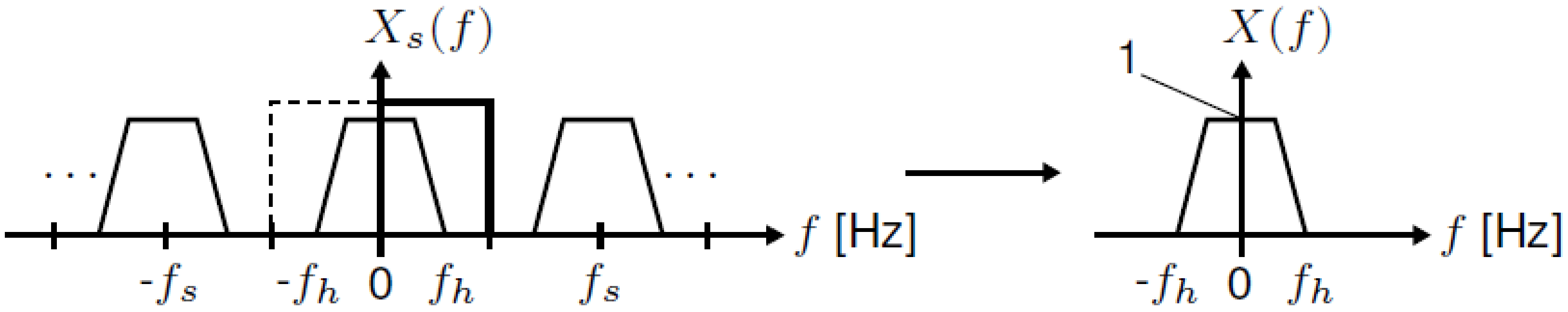
Original Signal (Time Domain)
x(t)
Time (s)
X 0.017
Y 0.71213
Impulse Sampled Signal
$x_{imp}(t)$
Time (s)
X 0.02
Y 1.29389
X 0.017
Y 0
Pulse Sampled Signal
$x_{pulse}(t)$
Time (s)
X 0.017
Y 0.71213
X 0.02
Y 1.29389


Original Spectrum
|X(f)|
Frequency (Hz)
Impulse Sampling Spectrum
$|X_{imp}(f)|$
Frequency (Hz)
Pulse Sampling Spectrum
$|X_{pulse}(f)|$
Frequency (Hz)

SDU

Reconstruction

The amplitude spectrum X(f) can be recreated from Xs(f) by low-pass filtering.



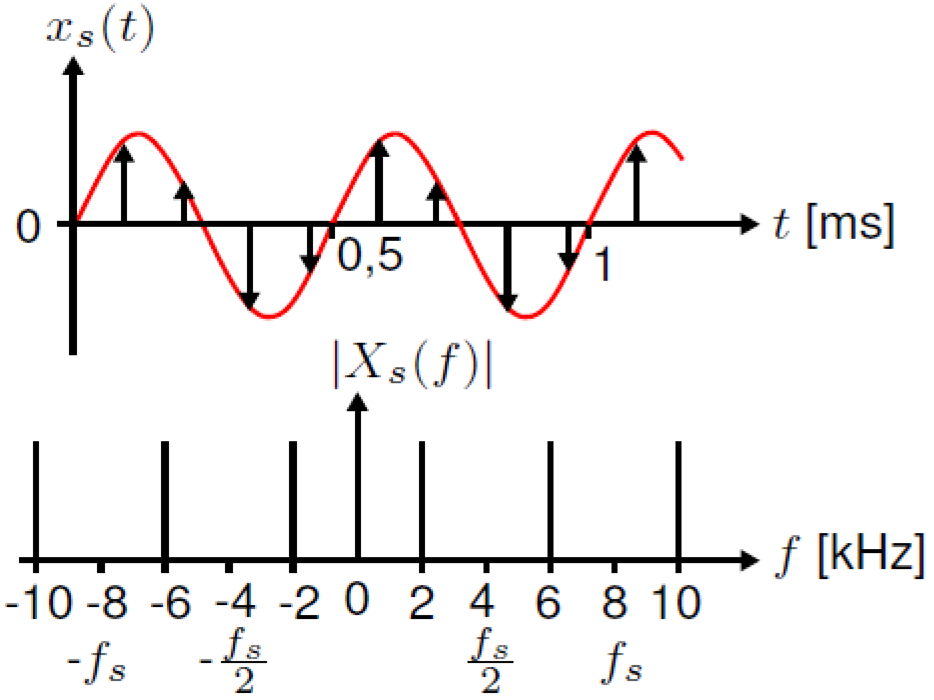
Example

Let's take a look the following signal

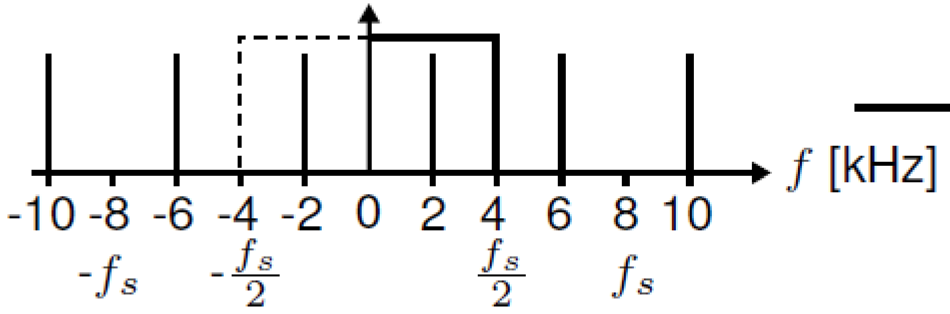
$$x(t) = \sin(2\pi f t)$$

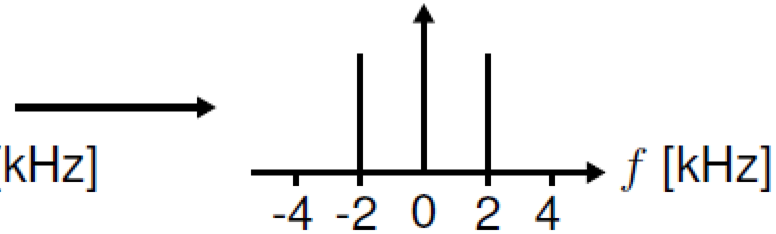
where the signal frequency $f = 2\ \text{kHz}$

We use a sampling frequency $f_s = 8\ \text{kHz}$



And it can be reconstructed as follows



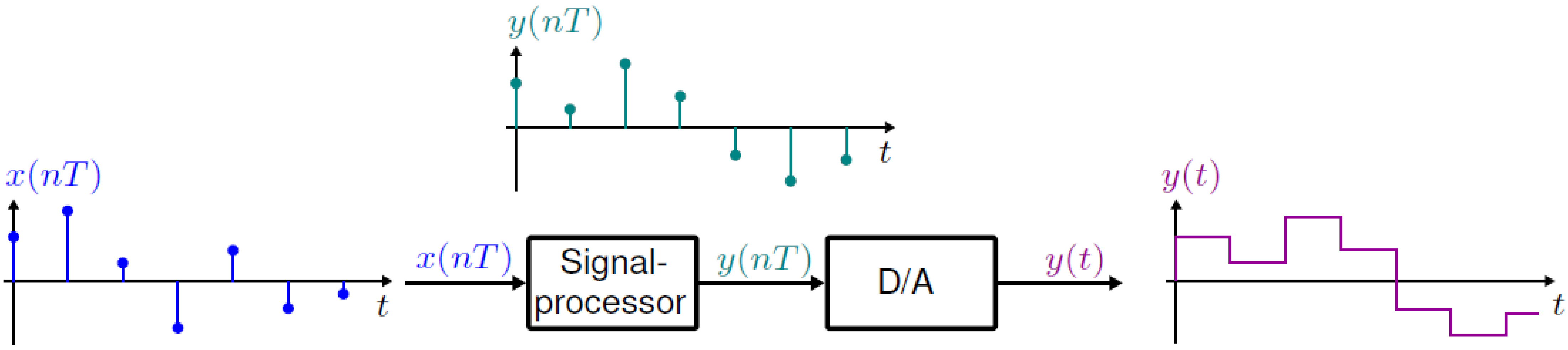


SDU

Reconstruction

Reconstruction means that all the values between 0 and T are interpolated.

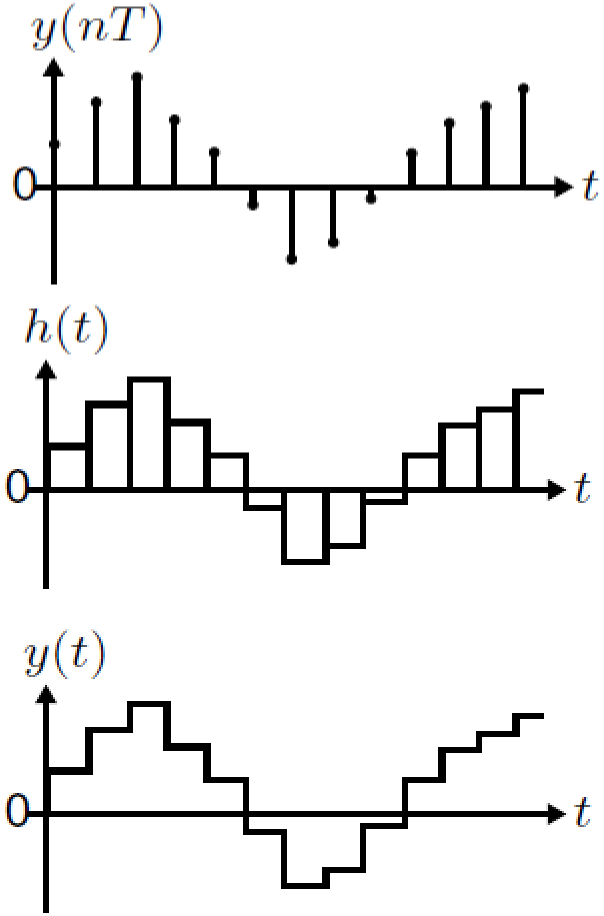
A digital-analog converter (D/A converter) reconstructs the analog signal from the digital sequence, and is therefore called a reconstruction circuit.



Since the output signal of the D/A converter is piecewise **constant** with an amplitude given by the amplitude of the digital sequence, this is a **zero-order hold (ZOH)** circuit.

Zero-order hold circuit

When a digital sequence is D/A converted using zero-order hold circuit acquisition, we get the following



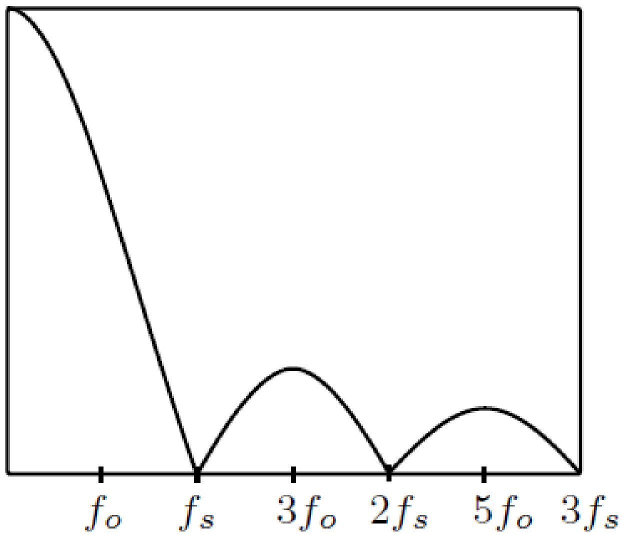
Note that the impulse response of the holding circuit is like that of a pulse sampler, where $\tau = T$ (duty factor of 1).

SDU

Amplitude characteristic for zero-order hold circuit

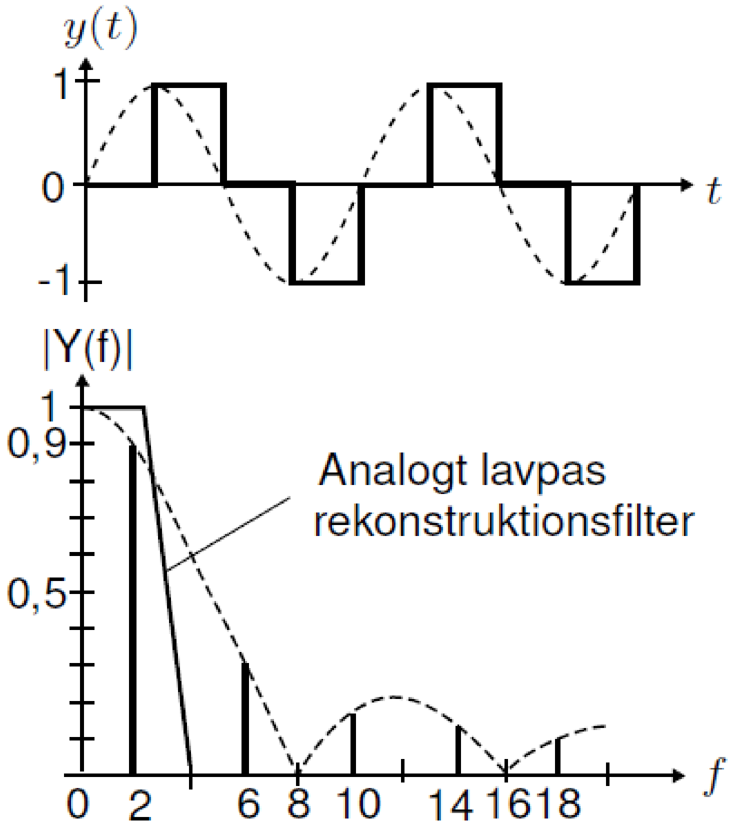
The frequency response of a zero-order holding circuit with input signal $x(t)$ and output signal $y(t)$ is

$$|Y(f)| = \left|\frac{\sin\left(\frac{\pi f}{f_s}\right)}{\frac{\pi f}{f_s}}\right| \sum_{m=-\infty}^{\infty} X(f - m f_s)$$



Example

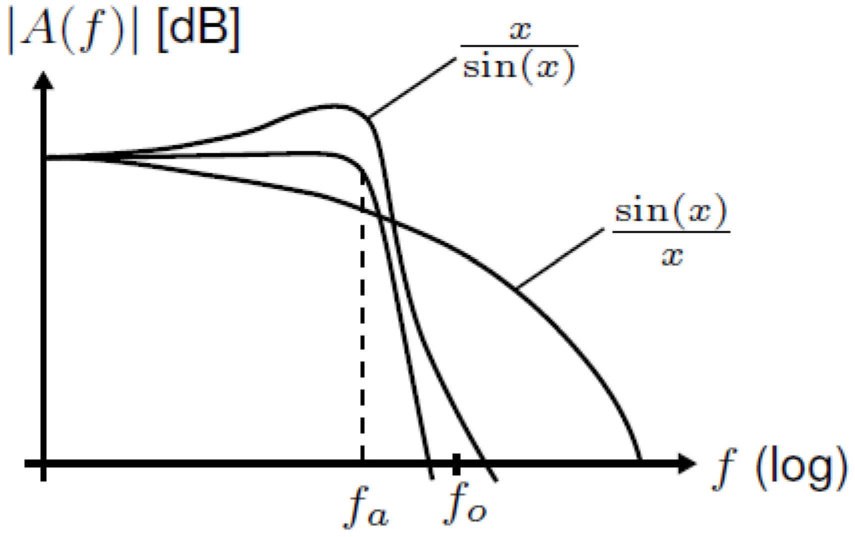
The following shows the reconstruction of a 2 kHz sine wave sampled at 8 kHz.



A low-pass filter is often inserted after the D/A converter to attenuate the high-frequency components at $f > f_o$.

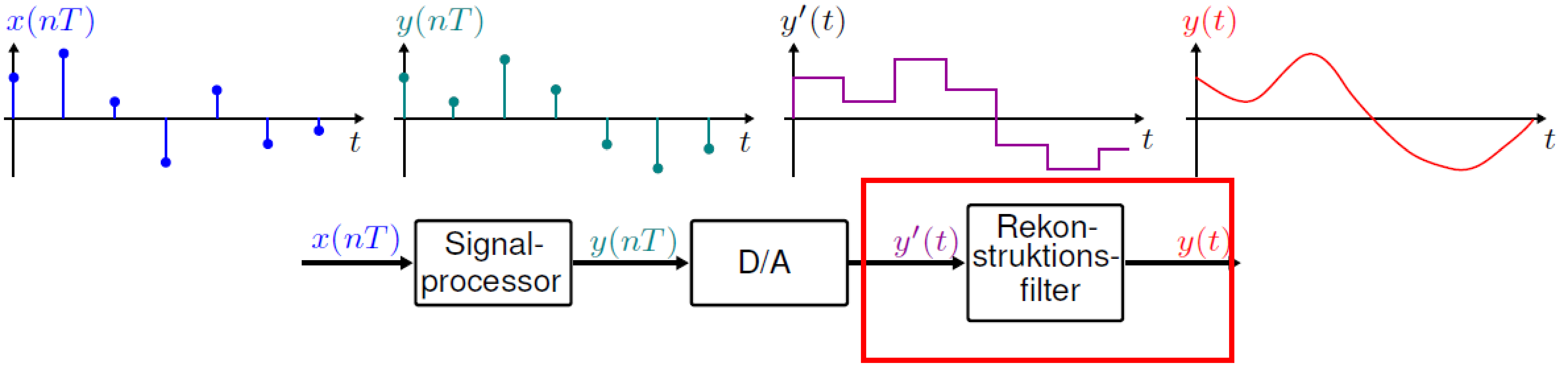
Correction of amplitude response

Since the $\mathrm{sinc}()$ function is multiplied on the amplitude spectrum of the signal, a reconstruction filter with the reciprocal amplitude characteristic $\frac{x}{\sin(x)}$ can be added.



Reconstruction filter

Reconstruction filter: to remove frequency components with frequency above f_o, a low-pass filter is used.

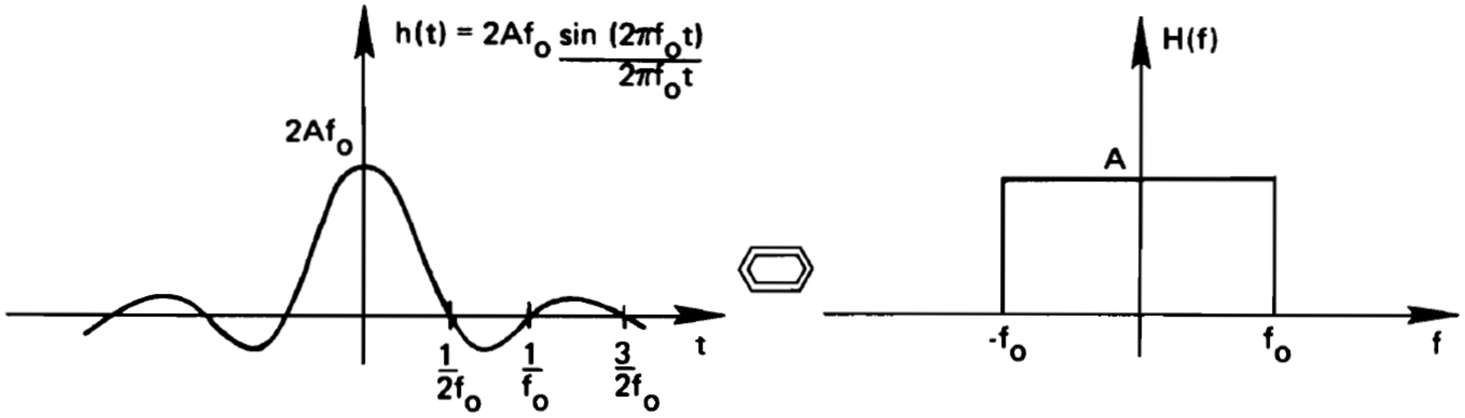


Reconstruction in the time domain

$$x_r(t) = h_r(t) * x_s(t) = h_r(t) * x_s(t) = h_r(t) * \sum_n x(n)\ \delta_c(t - nT) = \sum_n x(n)\ h_r(t - nT)$$

$$= \cdots + x(-1)h_r(t+T) + x(0)h_r(t) + x(1)h_r(t-T) + \cdots$$

where $h_r(t)$ is called the reconstruction filter



Multiply in frequency domain
Means
Convolution in time domain

Example - reconstruction

→ Generate an original signal 1Hz sin wave of 4s.
→ Do an impulse sample of 28 samples
→ Present the zero-order hold result
→ Reconstruct analog signal from the discrete time domain samples

```
clc; clear; close all;

t = 0:0.001:4-0.001;
x_org = sin(2*pi*t);

t_smp = linspace(0, 4, 29); % 28+1
f_s = 28/4; % sampling freq
x_n = sin(2*pi*t_smp);
figure();
plot(t,x_org)
hold on
stem(t_smp, x_n)

% Zero-order hold (ZOH)
t_hold = linspace(0, 4, 1001); % Continuous time vector for holding
y_zoh = interp1(t_smp, x_n, t_hold, 'previous'); % ZOH interpolation

% Sinc Interpolation for Reconstruction
T = 1/f_s;
y_rec = zeros(size(t_hold)); % Initialize reconstructed signal
for k = 1:length(t_smp)-1
y_rec = y_rec + x_n(k) * sinc((t_hold - t_smp(k)) / T);
end

figure()
subplot(2,1,1)
plot(t_hold, y_zoh)

subplot(2,1,2)
plot(t_hold, y_rec)
```

Exercise time