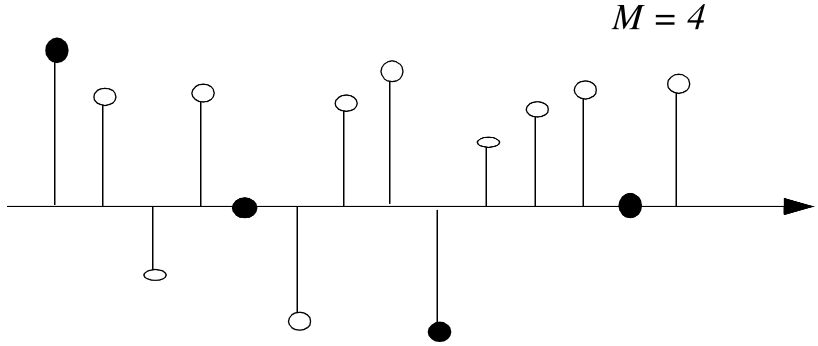
# Lecture 6: Down-Sampling and Up-Sampling

**Reading: Sec. 4.6.**

- Decimation (down-sampling) and interpolation (up-sampling) are two very useful concepts in discrete-time signal processing. They find applications in digital DLL, rate conversion, high-performance A/D and D/A converters, and recently image and video compression.
- **Down-sampling**: Given $x(n)$, define $x_d(n)$ to be the sequence down-sampled from $x(n)$ by a factor of $M$: $x_d(n) = x(nM)$.

  *Example:*

- First let's define the relationship among $X(e^{j\omega})$, $X_d(e^{j\omega})$, and $X_c(j\Omega)$.

  We know that $X(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{j\omega}{T} - \frac{j2\pi k}{T}\right)$ . Similarly, we have

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_{r=-\infty}^{\infty} X_c\left(\frac{j\omega}{MT} - \frac{j2\pi r}{MT}\right)$$

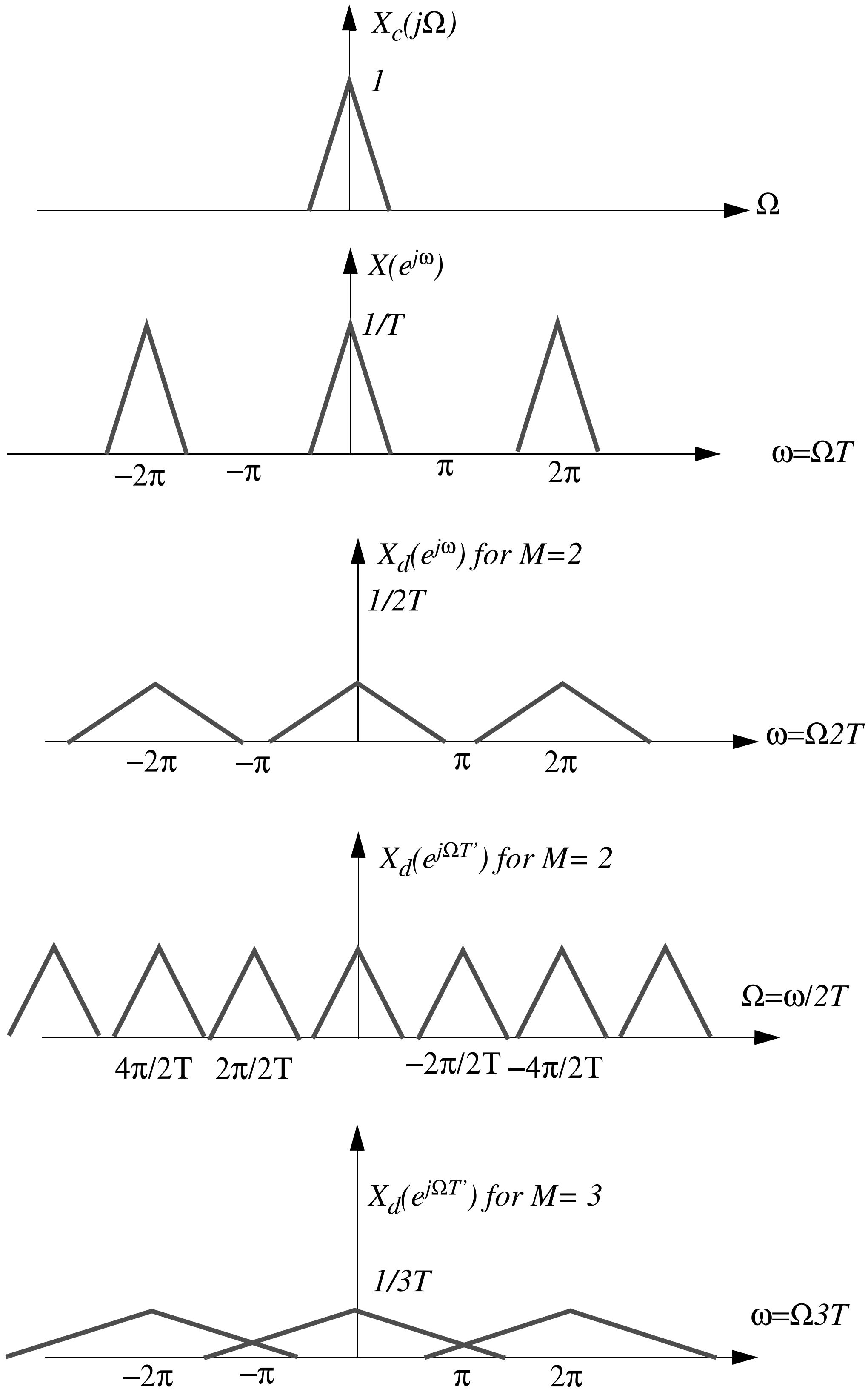
  Replacing $r = i + kM$,

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(\frac{j\omega}{MT} - \frac{j2\pi i}{MT} - \frac{j2\pi k}{T}\right)\right]$$

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{\omega - 2\pi i}{M}\right)}\right) \quad !$$

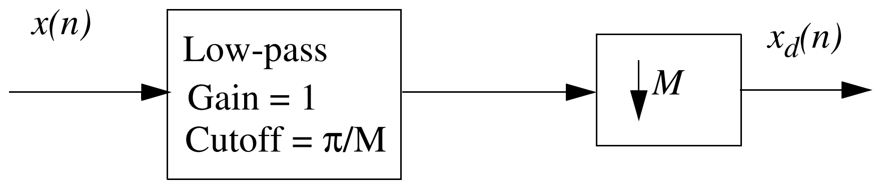
which represents $M$ copies of shifted and frequency-scaled $X(e^{j\omega})$.

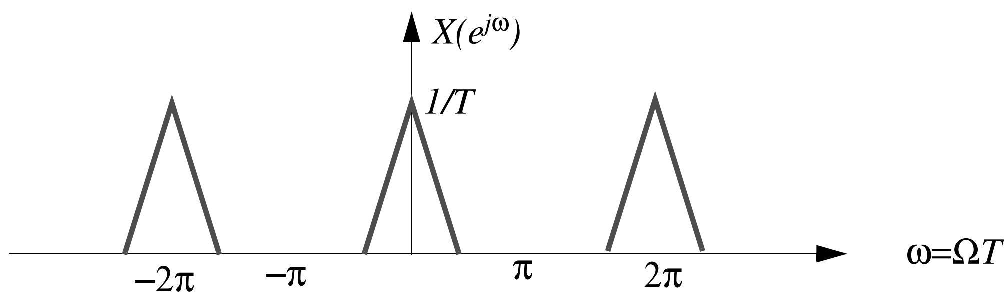
*Example:* It's easier to see frequency scaling in a graph.

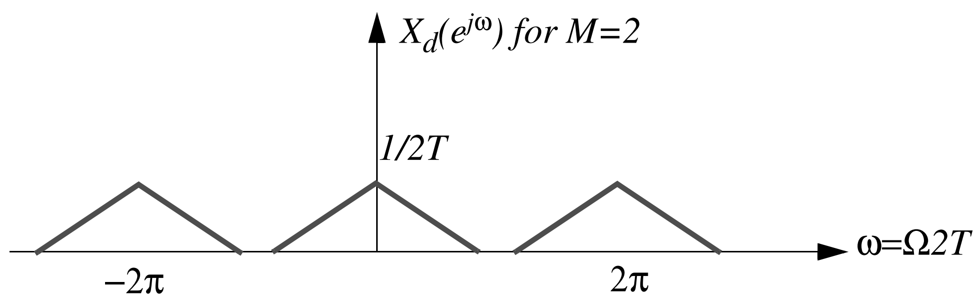


Well, what do we do now?

- Down-sampling is usually preceded by a low-pass filter to prevent aliases in the baseband:



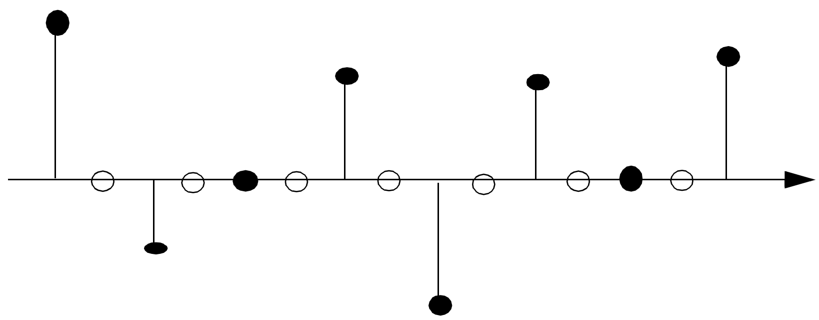

- Filtering is performed on *x(n)*, before down-sampling. If some guard band is allowed in the original signal *x(n)*, for example using a sampling rate higher than twice the signal bandwidth, then the design of the “decimation filter” would be alot easier.

- The effect of down-sampling is equivalent to reduced sampling rate (increased sampling interval). Hence the frequency response in Ω simply has more copies of aliases (no big deal). The frequency response in ω, however, seems “flattened”, which is just an artifact of a larger *T*’ (*T’=MT)*, making the normalized response stretched by a factor of *M* (frequency scaling).

- **Up-sampling** is often referred to as interpolation, which is a process of generating intermediate samples between discrete samples.

- Given $x(n)$, define $x_u(n) = \begin{cases} x(n/L), & n=0, L, -L, 2L, -2L, \ldots\ldots \\ 0, & otherwise \end{cases}$

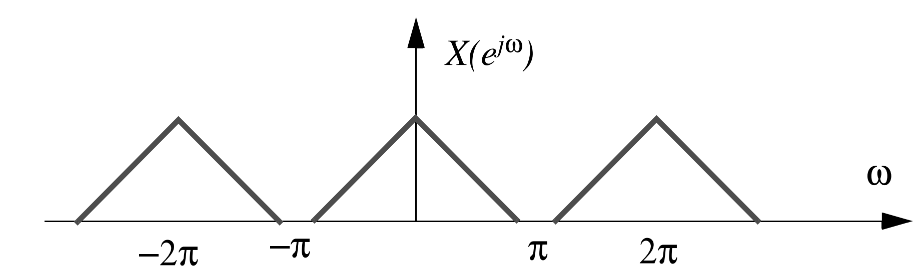
  *Example:* $x_u(n)$ is generated by padding zeros in between samples of $x(n)$:

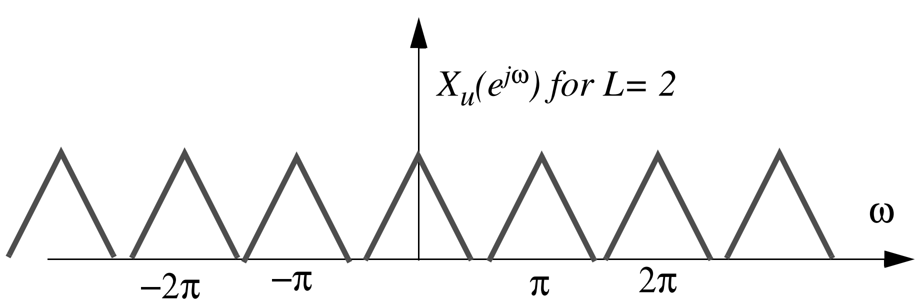


$$X_u(e^{j\omega}) = \sum_{n=-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty} x(k)\delta(n-kL)\right)e^{-j\omega n}$$

$$X_u(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x(k) \sum_{n=-\infty}^{\infty} \delta(n-kL)e^{-j\omega n} = \sum_{k=-\infty}^{\infty} x(k)e^{-j\omega kL} = X(e^{j\omega L})$$

A frequency scaled $X(e^{j\omega})$!



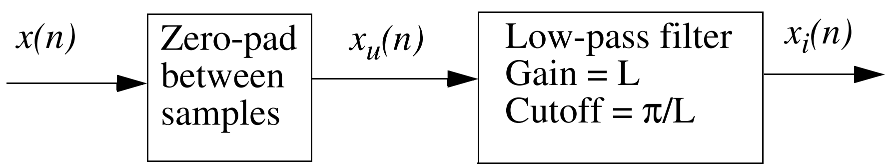



- We make two observations of $X_u(e^{j\omega})$:

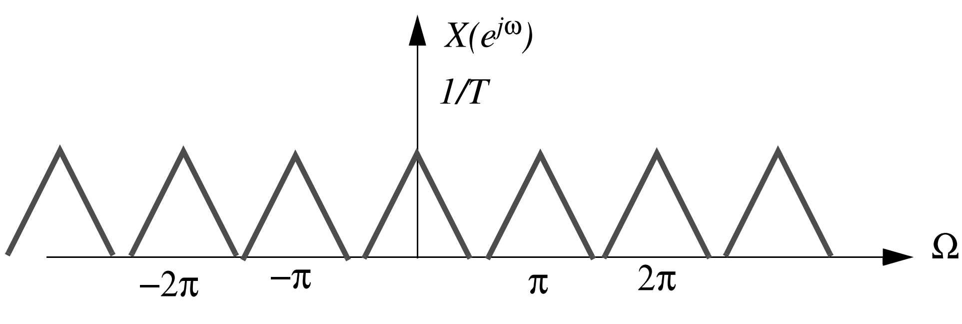
  1. Padding zeros in between samples in the time domain obviously introduces some high frequency components into the frequency spectrum, hence the fairly large frequency magnitude around $\pi$ (the highest frequency representable after sampling, sometimes termed the Nyquist frequency).

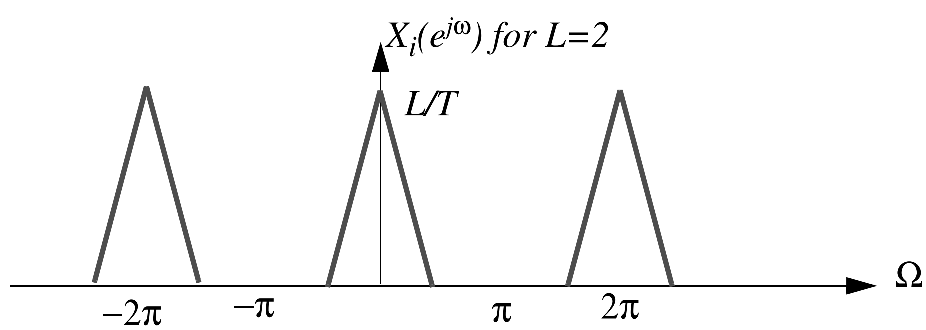
  2. Adding uniformly spaced zeros in the time domain shouldn't change the "shape" of the frequency spectrum (just take a look at its DTFT definition), hence the scaled frequency spectrum of the same shape.

- To generate the final up-sampled sequence, denoted by $x_i(n)$, we need to remove all the frequency scaled copies of $X(e^{j\omega L})$ except those at the integer multiples of $2\pi$, which implies the use of a low-pass filter.



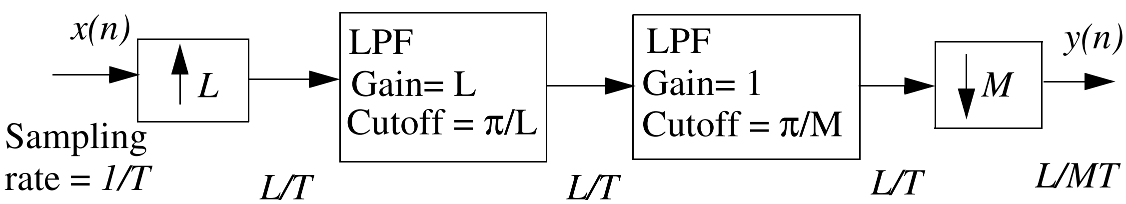

The low-pass filter has a gain of $L$ to make sure that $X_i(e^{j\omega})$ has the same spectrum as if obtained from sampling the original continuous-time signal $x_c(t)$ at a sampling rate of $L/T$.



## Application 1: Rate Conversion.

- Rate conversion is a process that converts the sampling rate of a discrete-time signal to a different one without changing the signal's analog waveform, e.g. from the US NTSC TV standards to the European PAM TV standards, which operates at a different frame rate than the NTSC. Another example is the different frame rates used by computer monitors, such as the ones used by SUNs and MACs. In order to display a MAC-generated video on a SUN workstation, rate conversion must be employed.
- Basic rate conversion involves the following steps:



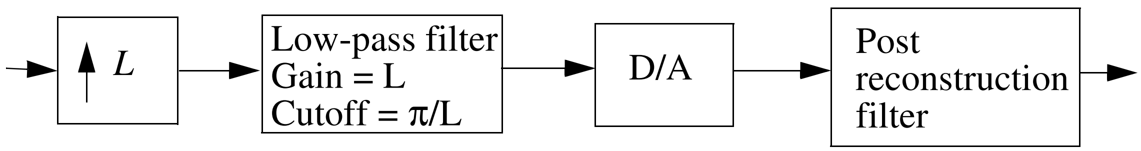

The two low-pass filters can be combined to be one with a gain of $L$ and a bandwidth of the smaller of $\pi/L$ and $\pi/M$.

## Application 2: Over-Sampled A/D Converters.

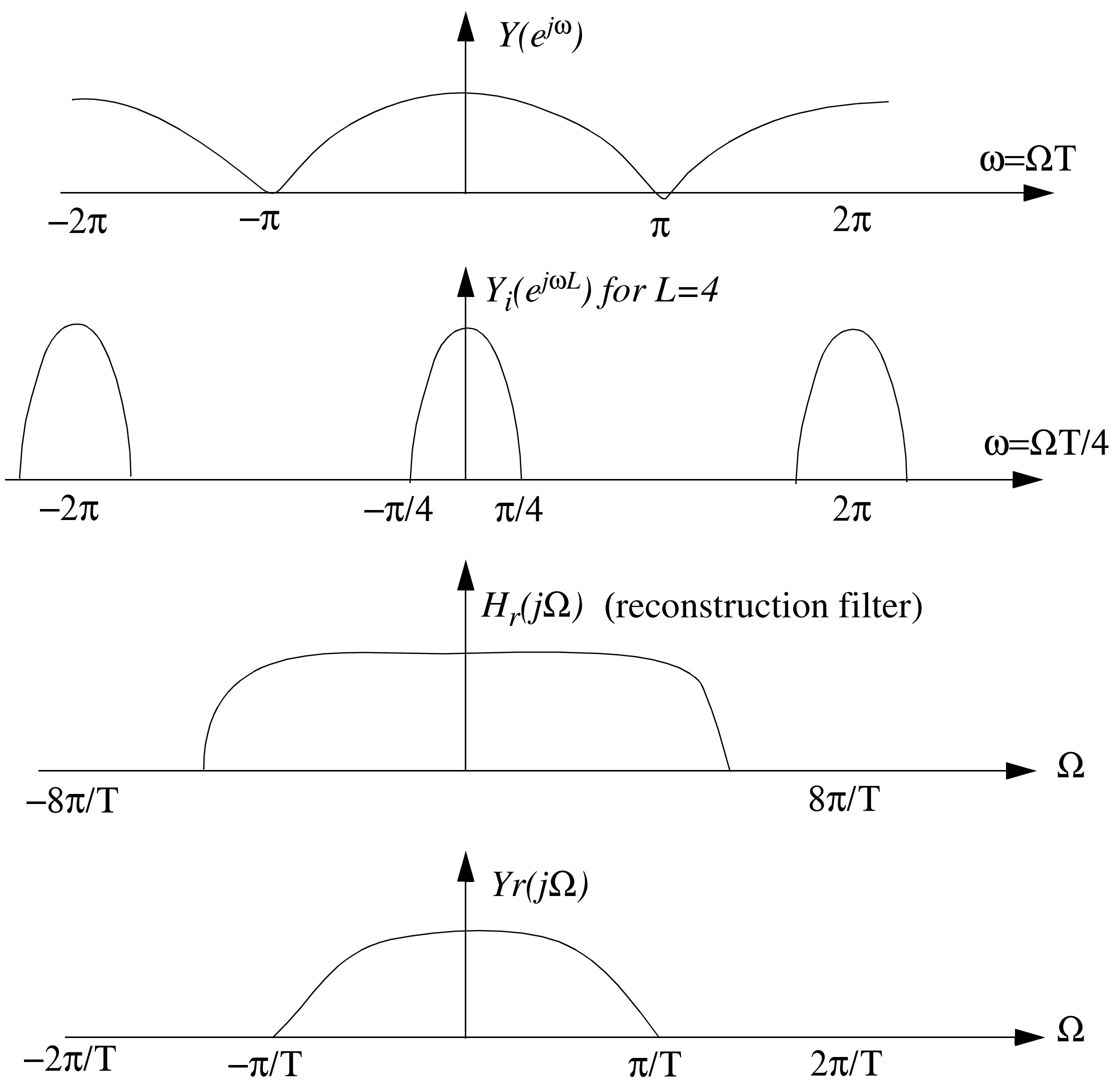
- To relax the specs of the analog anti-aliasing filter in the design of an A/D converter, we can transfer the burden of the designing a sharp cutoff analog low-pass filter to the design of a discrete-time filter, which is better controlled. One example is the sigma-delta A/D conversion scheme.
- In over-sampled A/D, we sample the analog signal at a rate much higher than twice the bandwidth of the signal, usually in the range of 4 to 256. By doing so the design of the analog anti-aliasing filter can be much relaxed. Down-sampling is then performed in the discrete-time domain to filter out the residue noise in high frequencies.
- Because of the use of the more robust discrete-time filter over the hard-to-control analog filter, over-sampled A/D converters usually result in better SNR (noise being the high-frequency noise aliased to the baseband due to the imperfect analog anti-aliasing filter).

## Application 3: Over-sampled D/A Converters.

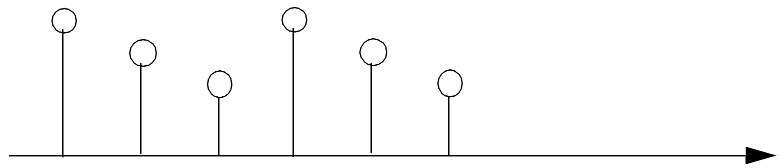
- Similar to the use of down-sampling filter in A/D conversion, up-sampling can be used in D/A conversion to relax the design of the final analog reconstruction filter. One example is the 8 times over-sampling filter advertized in the CD player.




- Up-sampling takes place before the final analog reconstruction filter so that the analog filter can have a much wider guard band, allowing a very easy design.

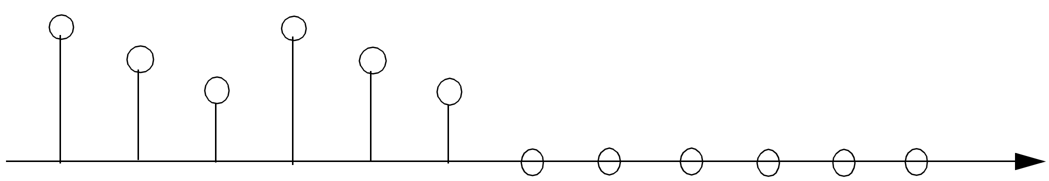


- **Quiz: Up-sampling**. Assume that *x(n)* and *X[k]* are a DFT pair and *x(n)* is given by the following 6 samples.

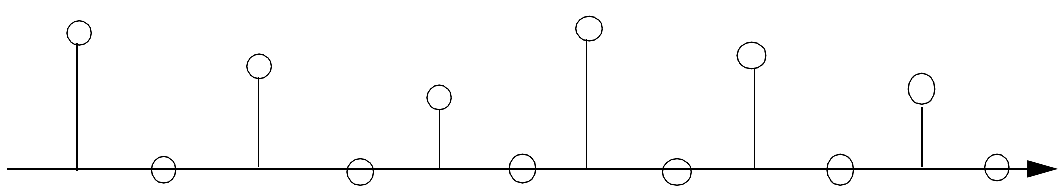


In each of the following cases, draw the corresponding DFT.

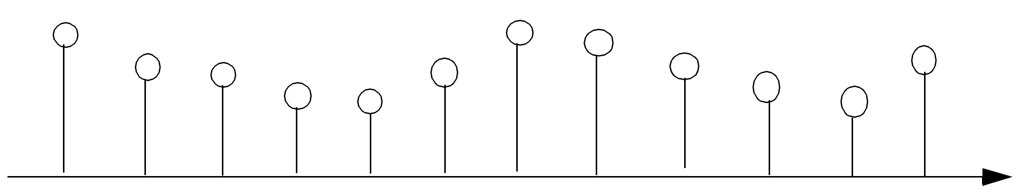
- If $x_1(n)$ is *x(n)* zero-padded at the tail, what is $X_1[k]$?



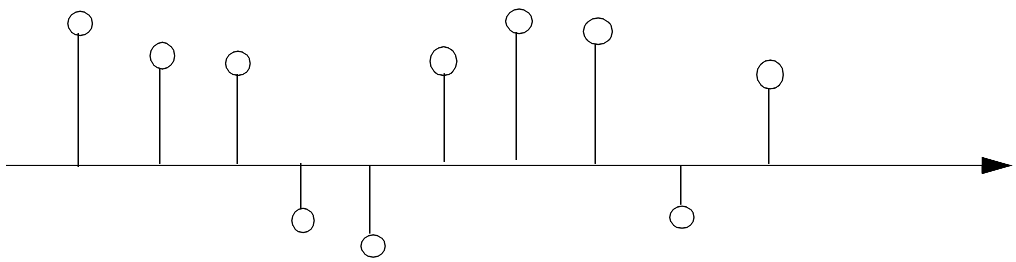
- If $x_2(n)$ is *x(n)* zero-padded in between samples, what is $X_2[k]$?



- If $x_3(n)$ is perfectly interpolated *x(n)*, what is $X_3[k]$?

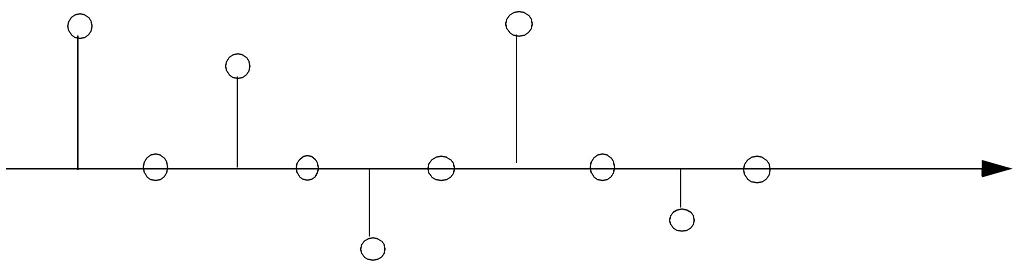


- **Quiz: Down-sampling**. Assume that $x(n)$ and $X[k]$ are a DFT pair and $x(n)$ is given by the following 10 samples.

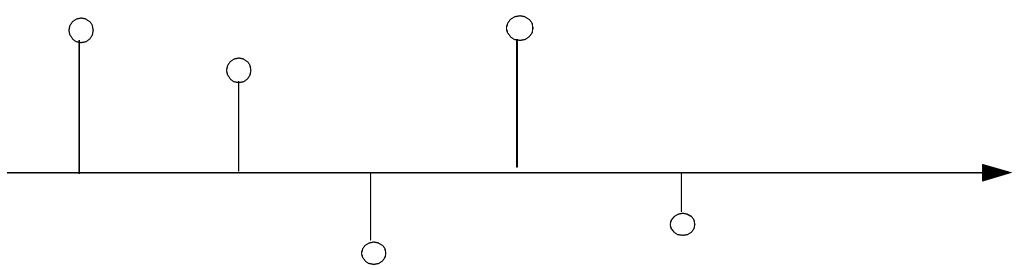


In each of the following cases, draw the corresponding DFT.

- If $x_1(n)$ is $x(n)$ with every odd sample set to zero, what is $X_1[k]$?



- If $x_2(n)$ is $x(n)$ with every odd sample deleted, what is $X_2[k]$?



- If $x_3(n)$ is $x(n)$ with every even sample set to zero, what is $X_3[k]$? Show that $X_1[k] + X_3[k] = X[k]$.