Review of Concepts

EE 20 Spring 2014
University of California, Berkeley

1 Introduction

In this note, we will discuss many of the central topics in EE 20 and how they relate to one another. Note that this is an unofficial review which should be ignored in favor of anything presented by the official course staff.

1.1 Outline

- Fourier Representations
  - Fourier Series Expansions
  - Fourier Transforms
- Multi-Rate Operations
  - Decimation
  - Interpolation
- Time-Frequency Uncertainty

2 Fourier Representations

2.1 Fourier Series Expansions

2.1.1 Continuous Time

2.1.2 Discrete Time

2.2 Fourier Transforms

2.2.1 Continuous Time

2.2.2 Discrete Time

3 Multi-Rate Operations

In digital signal processing (DSP), *multi-rate operations* are operations which can be used to change the sampling rate of a DT signal. We refer to the sampling rate $f_s = \frac{1}{T}$ of a DT signal $x[n]$ as the rate at which the CT signal $x(t)$ was sampled to yield $x[n] = x(nT)$, where T is the sampling period.

3.1 Decimation

The act of *decreasing the sampling rate* by an integer factor of M is referred to as *decimation*. If $y[n]$ is the signal produced via decimation of $x[n]$ by a factor of M, then $y[n]$ is equivalent to the DT signal that would result from sampling $x(t)$ with sampling period $T' = MT$. This corresponds to a decreased sampling rate of

$$\begin{aligned} f_s' &= \frac{1}{T'} \\ &= \frac{1}{MT} \\ &= \frac{1}{M} f_s. \end{aligned}$$

Decimation is realized by first filtering $x[n]$ with an *anti-aliasing (decimation) filter*, then *downsampling* by a factor of M.

3.1.1 Downsampling

A system which *downsamples* by an integer factor of M creates as output a new signal which contains every M^{th} sample of its input. Such a system is represented in a block diagram via the following symbol:

$$\mathbf{x[n]} \;—\; \textcircled{\downarrow \mathbf{M}} \;—\; \mathbf{y[n]}$$

Downsampling is the DT analogy to CT sampling. As in CT sampling, the time domain analysis is very simple, whereas the frequency domain analysis is a bit more involved.

- **Time Domain**
  The time domain representation of a downsampler with input $x[n]$ and output $y[n]$ is simply

  $$\boxed{y[n] = x[nM]}.$$

  As discussed above, the output is the sequence of every M^{th} sample of its input.

- **Frequency Domain**
  Let $X_c(\omega) \overset{\mathcal{F}}{\leftrightarrow} x(t)$ denote the CTFT of $x(t)$ and $X_d(\omega) \overset{\mathcal{F}}{\leftrightarrow} x[n]$ denote

the DTFT of $x[n] = x(nT)$. Recall from lecture the relationship between $X_c(\omega)$ and $X_d(\omega)$ (Theorem 22.1):

$$X_d(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega - 2\pi k}{T}\right)$$

Since $y[n] = x[nM] = x(nMT)$ is simply $x(t)$ sampled with $T' = MT$, we can write

$$\begin{aligned} Y(\omega) &= \frac{1}{T'} \sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega - 2\pi k}{T'}\right) \\ &= \frac{1}{MT} \sum_{k=-\infty}^{\infty} X_c\left(\frac{\omega - 2\pi k}{MT}\right). \end{aligned}$$

Our goal is to find the frequency domain equivalent of $y[n] = x[nM]$, i.e. the relationship between $Y(\omega)$ and $X_d(\omega)$. To do so, we will perform a series of convoluted variable substitutions. First, let $k = l + mM$ and note that summing from $k = -\infty$ to $k = \infty$ is the same as doing a double-sum over $l = 0$ to $M - 1$ and $m = -\infty$ to ∞. That is,

$$Y(\omega) = \frac{1}{MT} \sum_{l=0}^{M-1} \sum_{m=-\infty}^{\infty} X_c\left(\frac{\omega - 2\pi(l + mM)}{MT}\right).$$

Rearranging terms, we have

$$Y(\omega) = \frac{1}{M} \sum_{l=0}^{M-1} \frac{1}{T} \sum_{m=-\infty}^{\infty} X_c\left(\frac{\omega}{MT} - \frac{2\pi l}{MT} - \frac{2\pi m}{T}\right),$$

and recombining the terms in $X_c(\cdot)$ yields

$$Y(\omega) = \frac{1}{M} \sum_{l=0}^{M-1} \frac{1}{T} \sum_{m=-\infty}^{\infty} X_c\left(\frac{\left(\frac{\omega - 2\pi l}{M}\right) - 2\pi m}{T}\right).$$

Substituting in Theorem 22.1 and letting $k = l$, we see the desired relationship is

$$\boxed{Y(\omega) = \frac{1}{M} \sum_{k=0}^{M-1} X_d\left(\frac{\omega - 2\pi k}{M}\right)}.$$

3.1.2 The Anti-Aliasing (Decimation) Filter

Just as in CT sampling, downsampling a DT signal results in adding up aliased copies of the frequency-stretched (dividing ω by M stretches the frequency axis) and attenuated (the amplitude is decreased by M) spectrum of the input, so if we are to avoid aliasing, we require an *anti-aliasing filter* before performing downsampling. Since we are in DT where frequencies are 2π-periodic, we must be clear as to what we mean by aliasing: we say that aliasing occurs when the periods of the output signal's spectrum overlap as a result of downsampling. Since the spectrum is 2π-periodic, we may look at one period to see what's going on and extrapolate from there. On the interval $\omega \in [-\pi, \pi)$, aliasing occurs if frequency components of the input get stretched outside this interval – they would then overlap with neighboring periods. Since downsampling stretches the frequency axis by M, any nonzero frequency components in the input signal at $|\omega| \geq \frac{\pi}{M}$ will result in aliasing after downsampling (since $\left|\frac{\omega}{M}\right| \geq \frac{\pi}{M} \Rightarrow |\omega| \geq \pi$). Thus we define the anti-aliasing filter to be an ideal DT LTI low-pass filter with cutoff frequency $\omega_c = \frac{\pi}{M}$. If we apply this filter before downsampling, there will be no nonzero frequency components at $|\omega| \geq \frac{\pi}{M}$ which would be stretched out to $|\omega| \geq \pi$ after downsampling. Additionally, since upsampling decreases the amplitude of the input spectrum, our filter should have a gain of M at all frequencies to preserve the amplitude of the signal.

3.1.3 System Analysis

Figure 1 shows a decimator in block diagram form:

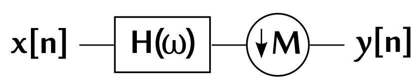


Figure 1: A decimator consisting of an ideal LPF and a downsampler.

As you've seen in class, systems that are *linear* and *time-invariant* (LTI) are of great interest to us since they can be compactly represented by an impulse response, or equivalently, a frequency response which has a very intuitive physical interpretation. Naturally, then, we would like to determine whether or not the decimator is an LTI system. Since it is the cascade of an LTI filter (...which is certainly LTI) and a downsampler, it suffices to determine whether or not the downsampler is LTI:

- **Linearity**
  Suppose $y_1[n]$ and $y_2[n]$ are the outputs of an M-downsampler corresponding to inputs $x_1[n]$ and $x_2[n]$, respectively. Let $\hat{y}[n]$ denote the output corresponding to input $\hat{x}[n] = ax_1[n] + bx_2[n]$, with $a, b \in \mathbb{R}$. From the definition of downsampling ($y[n] = x[nM]$), it is clear that $\hat{y}[n] = ay_1[n] + by_2[n]$, and thus we see that **downsampling is linear**. ✓

- **Time-Invariance**
  Suppose $y[n]$ is the output of an M-downsampler corresponding to input $x[n]$. Let $\hat{y}[n]$ denote the output corresponding to input $\hat{x}[n] = x[n - n_0]$,

with $n_0 \in \mathbb{Z}$. From the definition of downsampling, we see that

$$\begin{aligned} \hat{y}[n] &= \hat{x}[nM] \\ &= x[nM - n_0] \end{aligned}$$

and

$$\begin{aligned} y[n - n_0] &= x[(n - n_0)M] \\ &= x[nM - n_0 M]. \end{aligned}$$

Since $\hat{y}[n] \neq y[n-n_0]$, we see that **downsampling is not time-invariant**. $\ddot{\frown}$

As an intuitive example of the time-varying nature of downsampling, consider downsampling $x[n] = \delta[n]$ by a factor of $M = 2$. This results in

$$\begin{aligned} y[n] &= x[2n] \\ &= \delta[2n] \\ &= \delta[n]. \end{aligned}$$

Note that $y[n-1] = \delta[n-1]$. Now let $\hat{x}[n] = x[n-1] = \delta[n-1]$, and observe what happens when we downsample it:

$$\begin{aligned} \hat{y}[n] &= \hat{x}[2n] \\ &= \delta[2n - 1] \\ &= 0, \ \forall n, \end{aligned}$$

since $\forall n \in \mathbb{Z}, \ 2n - 1 \neq 0$. There is a clear difference between the signals that result from time shifting before and after downsampling, which results from the odd samples being thrown out (when downsampling by $M = 2$).

3.2 Interpolation

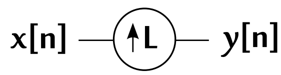
The act of *increasing the sampling rate* by an integer factor of L is referred to as *interpolation*. If $y[n]$ is the signal produced via interpolation of $x[n]$ by a factor of L, then $y[n]$ is equivalent to the DT signal that would result from sampling $x(t)$ with sampling period $T' = \frac{T}{L}$. This corresponds to an increased sampling rate of

$$\begin{aligned} f'_s &= \frac{1}{T'} \\ &= \frac{L}{T} \\ &= L f_s. \end{aligned}$$

Interpolation is realized by first *upsampling* $x[n]$ by a factor of L, then filtering with an *anti-aliasing (interpolation)* filter.

3.2.1 Upsampling

A system which *upsamples* by an integer factor of L creates as output a new signal which contains the samples of its input with $L-1$ zeroes between each input sample. Such a system is represented in a block diagram via the following symbol:



Upsampling is the DT analogy to $ImpulseGen_T$ in the $DiscToCont_T$ system presented in Chapter 11 of Lee & Varaiya. As upsampling and downsampling perform "opposite" operations, it may make some sense that the frequency domain analysis is now very simple, whereas the time domain analysis is a bit more involved.

- **Time Domain**
  The most natural time domain representation of an upsampler with input $x[n]$ and output $y[n]$ (i.e. that which most obviously conveys the idea of "inserting zeroes between input samples") is

  $$y[n] = \begin{cases} x\left[\frac{n}{L}\right], & \text{if } \frac{n}{L} \text{ is an integer} \\ 0, & \text{otherwise} \end{cases}.$$

  However, this representation does not lend itself well to analysis. Fortunately, as seen in a previous homework, we can write the above in a form that is easier to work with:

  $$\boxed{y[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]}$$

- **Frequency Domain**
  Let $X(\omega) \overset{\mathcal{F}}{\leftrightarrow} x[n]$ denote the DTFT of $x[n]$ and $Y(\omega)$ denote the DTFT of $y[n]$. To determine the relationship between $Y(\omega)$ and $X(\omega)$, we take the DTFT of both sides of the time domain relationship given above and apply the linearity of the Fourier transform:

  $$\mathcal{F}\{y[n]\} = \mathcal{F}\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]\right\}$$
  $$Y(\omega) = \sum_{k=-\infty}^{\infty} x[k]\mathcal{F}\{\delta[n-kL]\}$$

  Note that $x[k]$ is a constant inside the sum over k which allows us to pull it out of the Fourier transform. We now recall the transform pair for the Kronecker delta

  $$\delta[n] \overset{\mathcal{F}}{\leftrightarrow} 1$$

and time-shift property of the Fourier transform

$$x[n-n_0] \overset{\mathcal{F}}{\leftrightarrow} X(\omega)e^{-i\omega n_0}$$

to see that $\mathcal{F}\{\delta[n-kL]\} = e^{-i\omega kL}$. Putting it all together yields

$$Y(\omega) = \sum_{k=-\infty}^{\infty} x[k]e^{-i\omega kL}.$$

Substituting in the definition of the DTFT analysis equation, we see the desired relationship is

$$\boxed{Y(\omega) = X(\omega L)}.$$

3.2.2 The Anti-Aliasing (Interpolation) Filter

Similar to the reconstruction process in CT sampling, upsampling a DT signal results in a repetition of frequency-compressed (multiplying ω by L compresses the frequency axis) versions of the spectrum of the input. Just as in reconstruction in CT sampling, to interpolate the new samples that result from upsampling, we require an *interpolation filter*.[1] The frequency compression that results from upsampling can be viewed as "aliasing" – by limiting our view to the interval $[-\pi, \pi)$, we see that copies of the spectrum that were previously outside this interval are compressed into it. Since upsampling compresses the frequency axis by L, any nonzero frequency components at $|\omega| \geq \frac{\pi}{L}$ in the output signal must have been shifted in due to upsampling (since $|\omega L| \geq \pi \Rightarrow |\omega| \geq \frac{\pi}{L}$). Thus we define the anti-aliasing filter to be an ideal DT LTI low-pass filter with cutoff frequency $\omega_C = \frac{\pi}{M}$.

3.2.3 System Analysis

Figure 2 shows an interpolator in block diagram form:

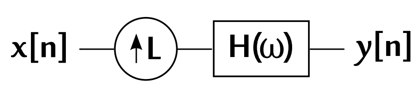


Figure 2: An interpolator consisting of an upsampler and an ideal LPF.

As in the case of the decimator, to determine whether or not the interpolator is an LTI system, it suffices to determine whether or not the upsampler is LTI:

- **Linearity**
  Suppose $y_1[n]$ and $y_2[n]$ are the outputs of an L-upsampler corresponding to inputs $x_1[n]$ and $x_2[n]$, respectively. Let $\hat{y}[n]$ denote the output

[1] In most DSP literature, this filter is also referred to as an *anti-aliasing* filter.

corresponding to input $\hat{x}[n] = ax_1[n] + bx_2[n]$, with $a, b \in \mathbb{R}$. From the definition of upsampling ($y[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$), we have

$$\begin{aligned}
\hat{y}[n] &= \sum_{k=-\infty}^{\infty} \hat{x}[k]\delta[n-kL] \\
&= \sum_{k=-\infty}^{\infty} (ax_1[n] + bx_2[n])\delta[n-kL] \\
&= a\sum_{k=-\infty}^{\infty} x_1[n]\delta[n-kL] + b\sum_{m=-\infty}^{\infty} x_2[n]\delta[n-mL] \\
&= ay_1[n] + by_2[n],
\end{aligned}$$

and thus we see that **upsampling is linear**. ✓

- **Time-Invariance**
  Suppose $y[n]$ is the output of an L-downsampler corresponding to input $x[n]$. Let $\hat{y}[n]$ denote the output corresponding to input $\hat{x}[n] = x[n-n_0]$, with $n_0 \in \mathbb{Z}$. From the definition of upsampling, we see that

$$\begin{aligned}
\hat{y}[n] &= \sum_{k=-\infty}^{\infty} \hat{x}[n]\delta[n-kL] \\
&= \sum_{k=-\infty}^{\infty} x[n-n_0]\delta[n-kL]
\end{aligned}$$

and

$$y[n-n_0] = \sum_{k=-\infty}^{\infty} x[n-n_0]\delta[n-n_0-kL].$$

Since $\hat{y}[n] \neq y[n-n_0]$, we see that **upsampling is not time-invariant**. ☹

As an intuitive example of the time-varying nature of upsampling, consider upsampling $x[n] = \delta[n]$ by a factor of $L = 2$. This results in

$$\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-2k] \\
&= \sum_{k=-\infty}^{\infty} \delta[k]\delta[n-2k] \\
&= \delta[0]\delta[n-2\cdot 0] \\
&= \delta[n].
\end{aligned}$$

Note that $y[n-1] = \delta[n-1]$. Now let $\hat{x}[n] = x[n-1] = \delta[n-1]$, and observe what happens when we upsample it:

$$\begin{aligned}
\hat{y}[n] &= \sum_{k=-\infty}^{\infty} \hat{x}[k]\delta[n-2k] \\
&= \sum_{k=-\infty}^{\infty} \delta[k-1]\delta[n-2k] \\
&= \delta[0]\delta[n-2\cdot 1] \\
&= \delta[n-2].
\end{aligned}$$

There is a clear difference between the signals that result from time shifting before and after upsampling, which results from the artificial insertion of zeroes between samples.

3.3 Example: Fractional Rate Changes

Note that downsampling and upsampling are only defined for integer factors M and L, respectively. However, there are situations in which we would like to change the sampling rate by a fractional (non-integer) factor. For example, say you wanted to change the sampling rate from f_s to $f_s' = \frac{L}{M} f_s$, for integers L and M. To accomplish this, we simply cascade an L-interpolator and an M-decimator which share the same anti-aliasing filter having cutoff frequency $\omega_c = \min\left\{\frac{\pi}{M}, \frac{\pi}{L}\right\}$.

4 Time-Frequency Uncertainty

The *time-frequency uncertainty principle* states that a function cannot be both time limited (finite-duration in time) and band limited, or alternatively, “One cannot simultaneously sharply localize a signal in both the time domain and frequency domain”. [2] While there is a well-developed theory of time-frequency uncertainty which is strongly related to a more well-known theory in quantum mechanics, its proof is a bit beyond the scope of this course. That being said, it is an important and useful concept which can provide explanations for various relationships between signals and their Fourier transforms. While this principle has not been explicitly stated in EE 20, we have seen its consequences in many forms:

- **Fourier transform pairs**:
  The notion that a signal cannot be localized in both time and frequency domains hints at an inverse relationship between the support of a time domain signal and its Fourier transform. [3] This is further exemplified by

[2] http://en.wikipedia.org/wiki/Uncertainty_principle#Signal_processing

[3] Note that we can be ambiguous as to which domain we're talking about by invoking the duality of the Fourier transform.

some common Fourier transform pairs – a delta function has the smallest possible support, whereas its Fourier transform is a constant, which has unlimited support. Similarly, a "box" signal has compact support, yet its Fourier transform is a sinc!

- **Contraction in time**:
  Downsampling a time domain signal contracts its support (since it involves keeping only every M^{th} sample and throwing everything else away), which in turn, as seen in the frequency domain analysis in Section 3, causes expansion of its spectrum.

- **Expansion in time**:
  Upsampling a time domain signal expands its support (since it involves inserting $L-1$ additional samples for each original sample), which in turn, as seen in the frequency domain analysis in Section 3, causes contraction of its spectrum.

- **Nyquist-Shannon sampling of bandlimited signals**:
  In the real world, all signals are finite-duration in time. As a result, no signals of interest are actually bandlimited! This presents a problem: according to the Nyquist-Shannon Sampling Theorem, we have no hope of accurately representing such signals by their samples or reconstructing them! Fortunately, this dilemma has motivated various methods for circumventing the worst-case, sufficient-but-not-necessary criteria of the Sampling Theorem. The easiest method we've seen is application of an anti-aliasing filter before sampling. In many cases, signals of interest are not bandlimited but have very little frequency content at high frequencies. Thus in applying an anti-aliasing filer, we are creating a bandlimited approximation to the original signal which is quite accurate and can be sampled via uniform sampling with very little loss of information. More exciting methods abandon uniform sampling entirely and try to exploit various structures (such as sparsity) in the signals of interest in order to sample "below Nyquist". For more on this, look into Finite Rate of Innovation (FRI) sampling and Compressed Sensing.

5 Closing Thoughts