Review of Signal Processing

This contains a brief review of

- Sampling and Reconstruction
- Decimation and Interpolation and Resampling

Sampling and Reconstruction

- I give a review of important facts about 1D theory, the 2D theory is analogous and available in the textbook (AK Jain, Chapter 2, 4).
- Some references
  - The review below is based on the book: Signals and Systems, by Oppenheim, Wilsky and Yoeng.
  - Also see an excellent tutorial on the web (for figures, examples, and many more details): `http://ccrma.stanford.edu/~jos/mdft/mdft.html`
  - Perfect interpolation of discrete periodic signals is possible to do. See S. Schanze, Sinc interpolation of discrete periodic signals, IEEE Trans Signal Processing, June 1995. Available at: `http://ieeexplore.ieee.org/iel4/78/8826/00388863.pdf?arnumber=388863`
- Disclaimer: This is a first draft, may have errors and typos, please email me (namrata@iastate.edu) if you find them.
- $\int$ without limits denotes $\int_{-\infty}^{\infty}$. Similarly for $\sum_n$.
- Let f_s is sampling frequency, let $T_s = 1/f_s$ is the sampling interval.
- Know/revise the following
  1. Definition of CTFT (Continuous time Fourier transform)
  2. Definition of DTFT (Discrete time Fourier transform)
  3. Definition of DFT (Discrete Fourier transform)
  4. Fourier transform properties and know common signal and CTFT/DTFT/DFT pairs.
  5. Nyquist's theorem, concept of aliasing.
- To sample a signal at frequency f_s, without aliasing (i.e. to be able to perfectly reconstruct it), it should be bandlimited to $f_s/2$. If not, apply an appropriate Low Pass Filter.
- Some important FT pairs
  1. Comb function: $x(t) = \sum_n \delta(t - nT_s)$, then $X(f) = f_s \sum_k \delta(f - kf_s)$
  2. Interpolation function: $x(t) = sinc(2\pi W t) = \frac{sin(2\pi W t)}{2\pi W t}$, then $X(f) = rect_W(f)$, i.e. $X(f) = 1$ for $|f| < W$ and 0 otherwise (perfect LPF with cutoff W).
  3. For bandlimited signals sampled above Nyquist, DTFT in one period $(-0.5, 0.5)$ is the CTFT with frequency scaled appropriately (explained below). But this does not hold in general (because of aliasing), e.g. the CTFT of $x(t) = rect_{T_0}(t)$ is a sinc function, while the DTFT of a sampled rectangular pulse of length N_1 is $\frac{sin(2\pi F(N_1+1/2))}{sin(\pi F)}$.

- Consider a continuous signal $x(t)$ that is sampled to give $x_s(t) = \sum_n x(nT_s)\delta(t - nT_s)$. This can be written as a discrete sequence $x_d[n] = x_s(nT_s) = x(nT_s)$.

- Let $X(f) = \int x(t)e^{-j2\pi ft}dt$ is CTFT of $x(t)$.

  1. Then $X_s(f) = \sum_n x(nT_s)e^{-j2\pi(f/f_s)n} = (f_s)\sum_m X(f - mf_s)$ is CTFT of x_s.
  2. The DTFT of $x_d[n]$ is $X_d(F) = \sum_n x_d[n]e^{-j2\pi Fn} = X_s(Ff_s) = (f_s)\sum_m X((F - m)f_s)$
  3. DFT assumes the signal $x_d[n]$ is time limited to $N - 1$ (i.e. original signal time limited to $(N - 1)T_s$).
  4. The DFT of $x_d[n]$ is $X_{d,DFT}(k) = \sum_{n=0}^{N-1} x_d[n]e^{-j2\pi kn/N} = X_d(k/N) = X_s(kf_s/N)$.

- Thus the CTFT of $x_s(t)$ is a periodic summation of the CTFT of $x(t)$ with period f_s. The DTFT of $x_d[n]$ is obtained by scaling down the frequency axis of CTFT by $1/f_s$. In other words, $f_s/2$ in CTFT corresponds to $1/2$ in DTFT. The DFT assumes a time limited signal. It is the DTFT "sampled" in the frequency domain. N point DFT (N even) corresponds to scaling the frequency axis of DTFT by N. In other words, $1/2$ in DTFT corresponds to $N/2$ in DFT and $f_s/2$ in the CTFT.

- In short, the DFT value $X(k)$ (k^{th} DFT value) is the frequency response of the sampled signal at kf_s/N.

- In MATLAB, we usually compute the DFT (due to ease of computation, and fast FFT algorithm available). Carefully understand effects of sampling, time windowing, zero padding:

  1. DFT assumes a time limited signal. For a signal that is not time limited, this corresponds to multiplying it by a square window (i.e. the DFT will be the DFT of the original signal convolved with a sinc function, width of sinc inversely proportional to the time window). This is called "leakage". Try computing DFT of a sine wave for different time windowing for a fixed but large value of N. Try zero padding each time. Does it help? Notice that what helps is keeping more periods of the sine wave (longer time window implies narrower convolving sinc).
  2. DFT $X(k)$ is frequency response at frequency kf_s/N. So if you're sampling above Nyquist rate (or above approx Nyquist rate), changing f_s should not change (not change by much) how the freq response plot looks. If it does, then you're plotting incorrectly!
  3. N point DFT: changing N only changes the frequency resolution, i.e. you get frequency response at all frequencies kf_s/N. So a $2N$ point DFT should only be an interpolated version of the N point DFT and should equal it at all points kf_s/N. One often zero pads the signal to obtain better frequency resolution.
  4. When dealing with decimation (subsampling a discrete signal/image), use $f_s = 1$ cycle/pixel, i.e. $T_s = 1$ pixel. We assume 1 pixel corresponds to some unit length (which we do not know).
  5. Remember DFT assumes uniform sampling of the original signal. If that is not the case, you should first re-sample (e.g. using interp1) with time interval T_s and then take N point DFT. When plotting DFT, scale frequency by f_s/N. Alternatively, write code to compute the DTFT of the non-uniformly sampled data. In image processing, this problem occurs when analyzing boundary contours of objects (or contour deformations). One example is the Fourier descriptor (FD) method of representing "shape" of a contour. FD is the Fourier series of the following 1D function: the tangent angle at different

points on the contour as a function of contour arclength. Another example is normal deformation (deformation along contour normal) between two contours as a function of arclength of the first one. In both these cases, arclength replaces "time". But because the image is defined on a rectangular grid, you do not get samples at uniform arclength distance apart. An overview of Fourier descriptors available at:
`http://www.tsi.enst.fr/tsi/enseignement/ressources/mti/descript_fourier/Part1.html`

- Reconstruction of continuous signal from $x_s(nT_s)$
  1. Perfect reconstruction corresponds to applying a perfect Low Pass Filter with cutoff $W = f_s/2$, i.e. it corresponds to convolving with a sinc function $h(t) = sinc(2\pi(f_s/2)t) = sinc(\pi f_s t)$.
  2. The sinc function has a response that goes from $-\infty$ to ∞, so it cannot be used in practice (except for periodic signals). See discussion below in the interpolation section.

Decimation, Interpolation, Resampling

- Decimation by D: subsampling a discrete time signal by D.
- It can always be interpreted as sampling the original signal at f_s/D.
- Decimation: interpret as a two step process.
  1. Take signal $x[n]$, keep only the values at $n = mD$, zero out rest, i.e. get $x_p[n] = \sum_m x[mD]\delta(n - mD)$. Here δ is Kronecker delta.
  2. Shrink the time axis, i.e. $x_D[m] = x_p[mD] = x[mD]$.
  3. In DTFT domain, $X_d(\omega) = X_p(\omega/D) = (1/D)\sum_{k=0}^{D-1} X(\omega - 2\pi k/D)$
  4. To prevent aliasing, the original signal $x[n]$ should be bandlimited to $1/2D$ (or π/D). If not, apply a Low Pass Filter (LPF), which is called anti-aliasing filter.
- Perfect Interpolation of a signal $y[n]$ by D: interpret as a two step process again
  1. Insert $D - 1$ zeros between every two samples of $y[n]$, to get $y_Z[n]$. DTFT: $Y_Z(\omega) = Y(D\omega)$.
  2. Low Pass Filter by $rect_W(\omega)$ with $W = \pi/D$ i.e. $Y_I(\omega) = Y_Z(\omega)rect_W(\omega) = Y(D\omega)rect_W(\omega)$. In time domain, this corresponds to multiplication by discretized sinc, i.e. $y_I[n] = y_Z[n] * h[n]$ where $h[n] = sinc(2\pi n/D)$ and $*$ denotes convolution.
  3. Since the sinc function has an infinite impulse response, it cannot be used in practice. An exception is periodic signals. See the reference:
     S. Schanze, Sinc interpolation of discrete periodic signals, IEEE Trans Signal Processing, June 1995. Available at:
     `http://ieeexplore.ieee.org/iel4/78/8826/00388863.pdf?arnumber=388863`
- Practical Interpolation
  1. We often use zero order hold, linear (1st order hold), quadratic (2nd order hold) or cubic (3rd order hold) interpolation. These are not perfect interpolators, i.e. they are not bandlimited (results in "interpolation error") and their gain is not uniform in the passband (results in "resolution loss").

2. Zero order hold is averaging filter, its FT is a *sinc* function. FT of linear interpolator is $sinc^2$, and so on. Thus zero order hold has minimum resolution loss, but maximum interpolation error. Linear is a good trade off and so is used often in practice.
3. n^{th} order hold for $n \to \infty$ approaches a Gaussian function.
4. Other popular interpolators are the class of n^{th} order Lagrange functions or B-spline (Bezier spline) functions. The n^{th} order Lagrange function approaches a sinc function as $n \to \infty$.
5. Read Section 4.2 of AK Jain.

- Resampling by a rational factor, P/Q: first interpolate P times (insert zeros and LPF), apply anti-aliasing LPF and then decimate Q times.