Sampling

Yinan Huang

Firstly it seems an amazing conclusion to me that, though given a continuous time signal $x(t)$, the whole information is only stored in its samples $x(nT)$, if we know the highest frequency of $x(t)$. How can infinite information (sample $x(nT)$) represent a continuous time signal with finite dimension?

Without any help of math, we first think it with intuition. Given series of points $a_n = x(nT)$, the original signal $x(t)$ must meet the constraint that it goes through all these points. The set $\{x(t)|x(nT) = a_n\}$ consists of all possible $x(t)$. A possible $x(t)$ could change slowly, passing its samples smoothly, or could change fastly and oscilate within two sample points. Knowing the highest frequency of real $x(t)$, we could determine $x(t)$. But be careful, if our sampling rate is too slow, the possible set $\{x(t)\}$ is too dense to distinguish.

1 Continuous time sampling and restoration

For a continuous time signal $x(t)$, we usually use a sampling function $p(t)$ (periodical for uniform sampling) to model a sampling process:

$$x_p(t) = x(t)p(t). \tag{1}$$

A periodical $p(t)$ has a spectrum of the following form:

$$P(i\omega) = 2\pi \sum_k a_k \delta(\omega - k\omega_s), \tag{2}$$

where a_k is the fourier coefficient and ω_s is the sampling (angular) frequency. Because $X_p(i\omega) = \frac{1}{2\pi} X(i\omega) * P(i\omega)$, so

$$X_p(i\omega) = \sum_k a_k X(i(\omega - k\omega_s)). \tag{3}$$

The spectrum of signal after sampling is just copying (with weights) and moving the original spectrum. As a simple example, considering impulse sampling $p(t) = \sum_k \delta(t - kT)$, then $a_k = 1/T$, we have

$$X_p(i\omega) = \frac{1}{T} \sum_k X(i(\omega - k\omega_s)). \tag{4}$$

We find we can easily reconstruct $x(t)$ from $x_p(t)$ by using a low-pass filter. But do be careful that to keep the spctrum $X(i(\omega - k\omega_s))$ seperated, we must require $\omega_s > 2\omega_M$, with ω_M being the highest frequency of $x(t)$. This requirement is called sampling theorem and $2\omega_M$ is called nyquist sampling rate.

Applying a filter $h(t)$ to $x_p(t)$ can be written as

$$x_r(t) = x_p(t) * h(t) = \sum_n x(nT)h(t - nT), \tag{5}$$

which can be seen as a interpolation. A ideal low-pass filter $h(t) = \frac{\sin \frac{\omega_s}{2} t}{\pi t}$ ensures

$$x_r(t) = \sum_n x(nT) \frac{\sin \frac{\omega_s}{2} t}{\pi t} = x(t), \tag{6}$$

as long as the sampling theorem is meet. The interpolation corresponding to ideal low-pass filter is perfect, but hard to achive. An attempt to use a non-ideal low-pass filter $h(t)$, can be seen as an approximate interpolation method. For instance, let $h(t)$ be a square wave, $H(i\omega) = e^{-i\omega T/2} \frac{2 \sin \omega T/2}{\omega}$ is an approximate low-pass filter, and in the point of view of interpolation, it is a zero-order interpolation.

2 Discrete time sampling and restoration

Discrete time sampling is similar to continuous time case. The spectrum is copyed and moved within $(-\pi, \pi)$. Suppose $x_p[nN] = x[nN]$ is the signal after sampling, then we notice x_p will be very sparse and thus waste memories. So we apply a process call decimation, which defines a decimation signal $x_b[n] = x_p[nN] = x[nN]$. It leads to the expansion of spectrum

$$X_b(e^{i\omega}) = \sum_n x_b[n] e^{-i\omega n} = \sum_n x_p[nN] e^{-i\omega n} = \sum_n x_p[n] e^{-i\omega n/N} = X_p(e^{i\omega/N}). \tag{7}$$

If we assume $x[n]$ is sampled from a continuous time signal $x(t)$, then sampling and then decimation is equivalent to decrese the sampling rate, and thus this process is also called downsampling. Downsampling expands the spectrum of the original signal. To restore the original signal, we first add zero points to the signal, and then apply a low-pass filter. This inverse process is also called upsampling, which constracts the spectrum.