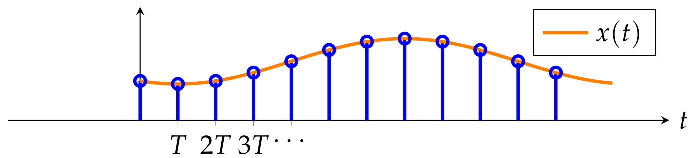
EE120 - Fall'15 - Lecture 11 Notes[1]

Murat Arcak

7 October 2015

Sampling Continued

Chapter 7 *in* Oppenheim & Willsky



$$x_d[n] = x(nT) \tag{1}$$

How are the Fourier Transforms of $x(t)$ and $x_d[n]$ related?
Use impulse train sampling: $x_p(t) = \sum_n x_d[n]\delta(t - nT)$ to relate the two:

$$X_p(j\omega) = \sum_{n=-\infty}^{\infty} x_d[n]e^{-j\omega Tn}$$

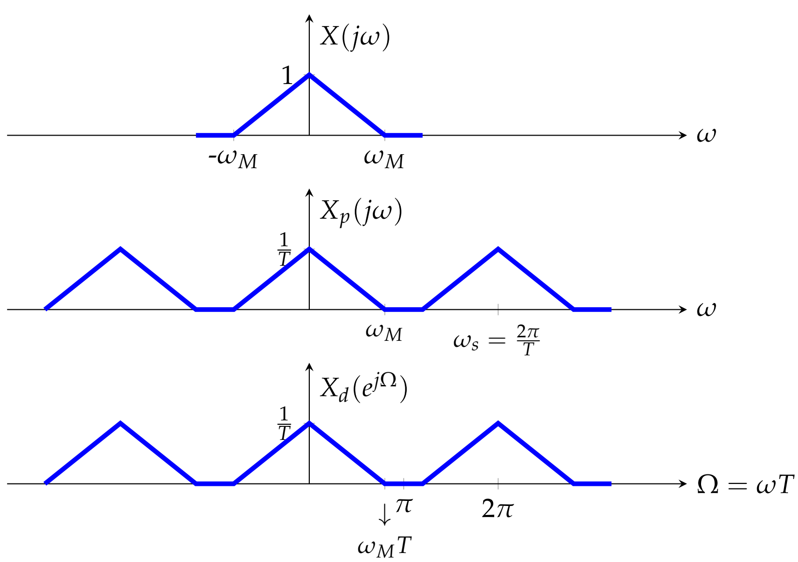
$$X_d(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x_d[n]e^{-j\Omega n}$$

$\Omega = \omega T$
ω : radians/sec.
Ω : radians

$$\boxed{X_d(e^{j\Omega})\Big|_{\Omega=\omega T} = X_p(j\omega)} \tag{2}$$

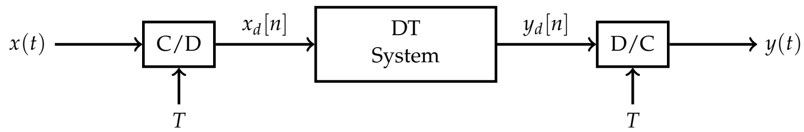
Last lecture: $$\boxed{X_p(j\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s))} \tag{3}$$

Combine the two:

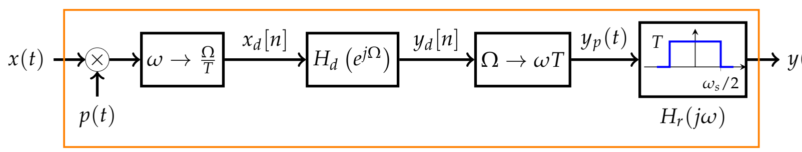


DT Processing of CT Signals

Section 7.4 *in* Oppenheim & Willsky



This is not a LTI system (why not?), therefore it does not possess a well-defined frequency response $H(j\omega)$.
However, if $x(t)$ is bandlimited by $w_s/2 = \pi/T$, an "effective" $H(j\omega)$ can be calculated:



$$Y_d(e^{j\Omega}) = H_d(e^{j\Omega})X_d(e^{j\Omega}) = H_d(e^{j\Omega})X_p(j\Omega/T) \tag{4}$$

$$Y_p(j\omega) = Y_d(e^{j\omega T}) = H_d(e^{j\omega T})X_p(j\omega) \tag{5}$$

$$Y(j\omega) = \begin{cases} TH_d(e^{j\omega T})X_p(j\omega) & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2. \end{cases} \tag{6}$$

$$X_p(j\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s)) \tag{7}$$

Combining (6) and (7):

$$Y(j\omega) = \begin{cases} H_d(e^{j\omega T})\displaystyle\sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s)) & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2. \end{cases} \tag{8}$$

If $x_c(t)$ is bandlimited by $\omega_s/2$, no aliasing:

$$\sum_{k=-\infty}^{\infty} X(j(\omega - k\omega_s)) = X(j\omega) \qquad |\omega| < \omega_s/2 \tag{9}$$

$$Y(j\omega) = \begin{cases} H_d(e^{j\omega T})X(j\omega) & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2. \end{cases} \tag{10}$$

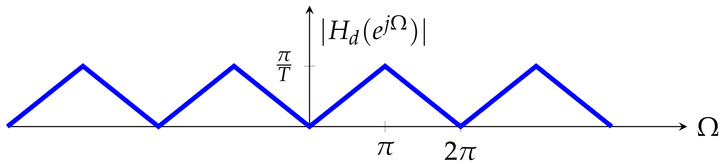
$$H_{\text{eff}}(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \begin{cases} H_d(e^{j\omega T}) & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2 \end{cases} \tag{11}$$

Effective freq. resp. valid for inputs with bandwidth $< \omega_s/2$

Example: Digital differentiator

$$\text{We want} \qquad H_{\text{eff}} = \begin{cases} j\omega & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2 \end{cases}$$

$$\text{therefore,} \quad H_d(e^{j\Omega}) = j\left(\frac{\Omega}{T}\right) \qquad |\Omega| < \pi$$



From Inverse Fourier Transform:

$$h_d[n] = \begin{cases} \frac{\cos \pi n}{nT} = \frac{(-1)^n}{nT}, & n \neq 0 \\ 0, & n = 0. \end{cases} \tag{12}$$

$$y_d[n] = h_d[n] * x_d[n] = \sum_k x_d[n-k]h_d[k] \tag{13}$$

$$\begin{aligned} &= \ldots h_d[-2]x_d[n+2] + h_d[-1]x_d[n+1] \\ &\quad + \underbrace{h_d[1]}_{-h_d[-1]} x_d[n-1] + \underbrace{h_d[2]}_{-h_d[-2]} x_d[n-2] + \ldots \end{aligned} \tag{14}$$

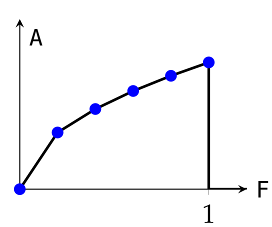
$$\begin{aligned} &= h_d[-1]\left(x_d[n+1] - x_d[n-1]\right) \\ &\quad + h_d[-2]\left(x_d[n+2] - x_d[n-2]\right) + \ldots \end{aligned} \tag{15}$$

Note that this is a form of numerical differentiation.
We can truncate $h_d[n]$ with an appropriate window and implement the resulting FIR filter.

MATLAB commands:

- `fir1` for lowpass, highpass, bandpass, *etc.*
- `fir2` for arbitrary shapes:

`fir2(M,F,A,window(M+1))` — plot (`F`,`A`) defines filter shape



Example above: $\mathtt{F} = [0, 1]$ $\quad \mathtt{A} = [0, j\pi/T]$

Try a simpler Euler approximation instead:

$$y_d[n] = \frac{x_d[n] - x_d[n-1]}{T} = h_d[n] * x_d[n] \tag{16}$$

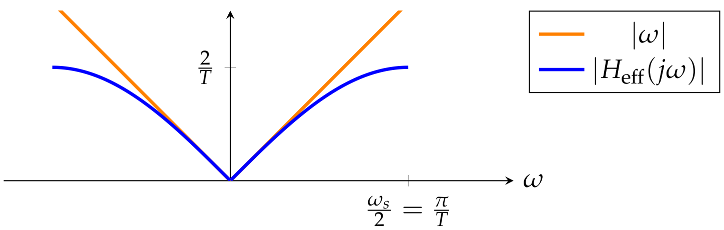
where $h_d[0] = \frac{1}{T}$, $h_d[1] = \frac{-1}{T}$, $h_d[n] = 0$ for $n < 0$ and $n > 1$.

$$H_d(e^{j\Omega}) = \sum_n h_d[n]e^{-j\Omega n} = \frac{1}{T}(1 - e^{-j\Omega}) \tag{17}$$

$$H_{\text{eff}}(j\omega) = \begin{cases} \frac{1}{T}(1 - e^{-j\omega T}) & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2 \end{cases} \tag{18}$$

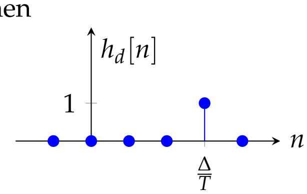
$$|H_{\text{eff}}(j\omega)| = \frac{1}{T}\sqrt{(1-\cos\omega T)^2 + (\sin\omega T)^2} \qquad |\omega| < \omega_s/2 \tag{19}$$

$$= \frac{1}{T}\sqrt{2(1-\cos\omega T)} \tag{20}$$



Example:
Digital implementation of a delay: $y(t) = x(t - \Delta)$
How should we design the DT filter?
If Δ is an integer multiple of T, then

$$y_d[n] = x_d[n - \underbrace{\tfrac{\Delta}{T}}_{\text{integer}}] \quad \rightarrow$$



What if $\frac{\Delta}{T}$ is not an integer?
We want $y(t) = x(t - \Delta)$, i.e. $Y(j\omega) = e^{-j\omega\Delta}X(j\omega)$.
Therefore, the desired $H_{\text{eff}}(j\omega)$ is:

$$H_{\text{eff}}(j\omega) = \begin{cases} e^{-j\omega\Delta} & |\omega| < \omega_s/2 \\ 0 & |\omega| > \omega_s/2 \end{cases} \tag{21}$$

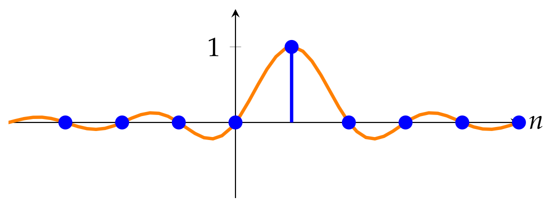
and $H_d(e^{j\Omega}) = e^{-j\frac{\Omega}{T}\Delta} \quad |\Omega| < \pi$.
Then inverse Fourier Transform gives:

$$h_d[n] = \frac{\sin\left(\left(n - \frac{\Delta}{T}\right)\pi\right)}{\left(n - \frac{\Delta}{T}\right)\pi} = \text{sinc}\left(n - \frac{\Delta}{T}\right) \tag{22}$$

Examples:

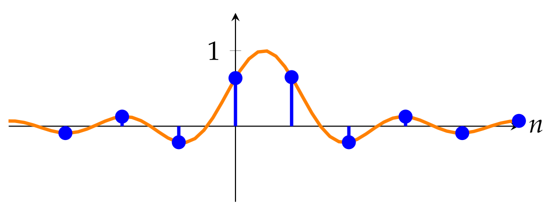
1) $\Delta = T$

$$h_d[n] = \frac{\sin((n-1)\pi)}{(n-1)\pi} = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases} \tag{23}$$



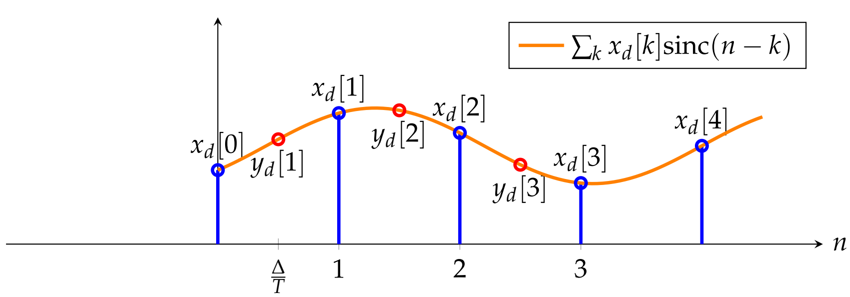
2) $\Delta = \frac{T}{2}$

$$h_d[n] = \frac{\sin\left(\left(n - \frac{1}{2}\right)\pi\right)}{\left(n - \frac{1}{2}\right)\pi} \tag{24}$$



$$y_d[n] = h_d[n] * x_d[n] = \sum_k x_d[k] h_d[n-k] \tag{25}$$

$$= \underbrace{\sum_k x_d[k] \text{sinc}\left(n - \frac{\Delta}{T} - k\right)}_{\text{Visualize this as if } n \text{ is a continuous variable: bandlimited interpolation of } x_d[k] \text{ shifted by } \Delta/T} \tag{26}$$



2D Sampling

Given $x(t_1, t_2)$ and sampling periods T_1, T_2:

$$x_d[n_1, n_2] \triangleq x(n_1T_1, n_2T_2).$$

Impulse train sampling:

$$x_p(t_1, t_2) = x(t_1, t_2)p(t_1, t_2)$$

where

$$p(t_1, t_2) \triangleq \sum_{n_1}\sum_{n_2} \delta(t_1 - n_1T_1, t_2 - n_2T_2).$$

2D CTFT gives:

$$X_p(j\omega_1, j\omega_2) = \frac{1}{T_1T_2}\sum_{k_1}\sum_{k_2} X\left(j\left(\omega_1 - k_1\omega_{s_1}\right), j\left(\omega_2 - k_2\omega_{s_2}\right)\right)$$

where

$$\omega_{s_1} = \frac{2\pi}{T_1} \quad \text{and} \quad \omega_{s_2} = \frac{2\pi}{T_2}.$$

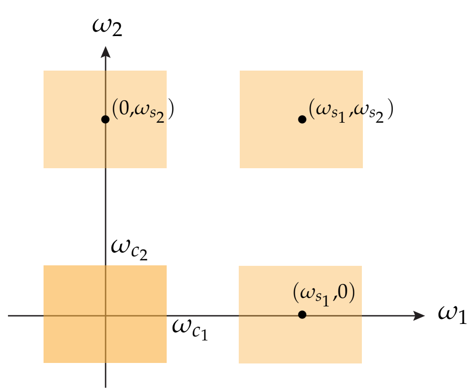
Therefore, if $x(t_1, t_2)$ is bandlimited:

$$X(j\omega_1, j\omega_2) = 0 \quad \text{when } |\omega_1| > \omega_{c_1} \text{ or } |\omega_2| > \omega_{c_2}$$

and

$$\omega_{s_1} > 2\omega_{c_1}, \qquad \omega_{s_2} > 2\omega_{c_2},$$

then there is no aliasing upon sampling:



Thus, $x(t_1, t_2)$ can be reconstructed from its samples with a low pass filter.