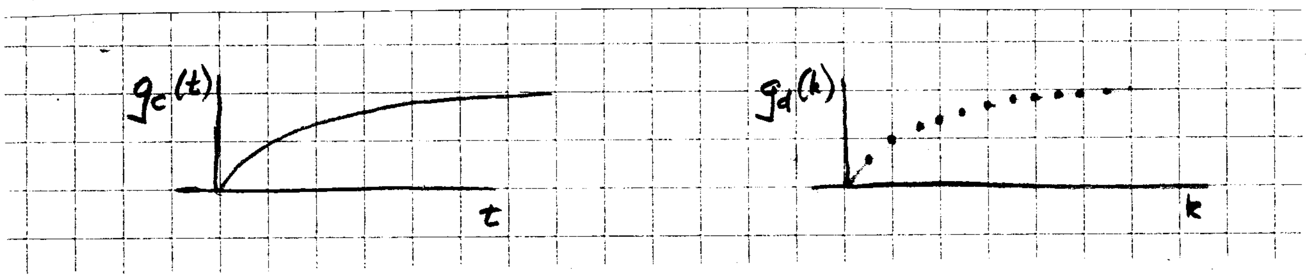
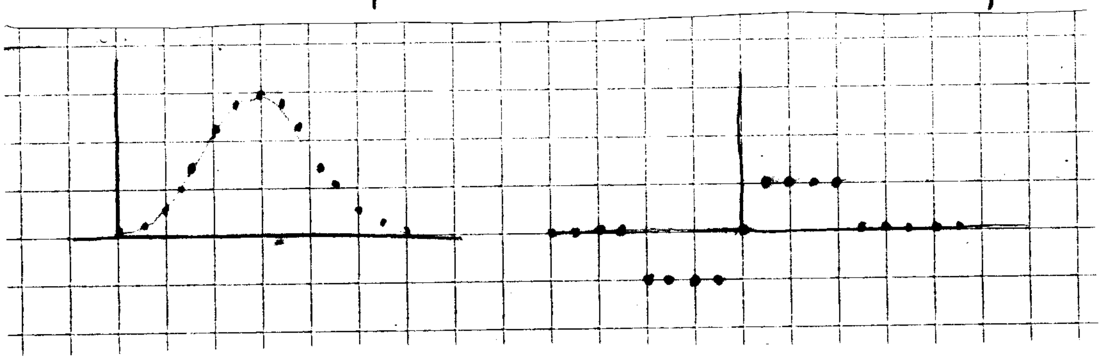
2.5 Discrete Time Approximation of Continuous Time Systems

- Although some systems are inherently discrete time (bank balance at payday, cash in till after departure of customer), many discrete time systems are designed to mimic continuous time

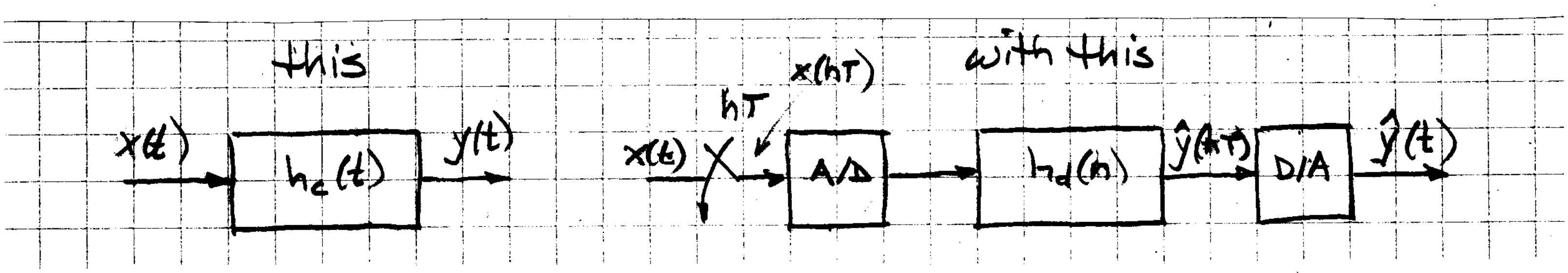
<u>example</u>: in digital filtering, get effect of RC lowpass



<u>example</u>: impulse responses difficult to realize by analog



- We want to approximate

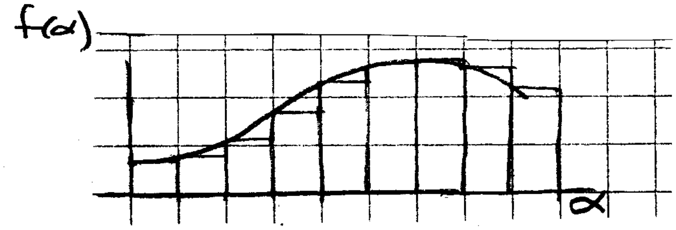


We want $h_d(n)$ such that

$$h_d(n) \circledast x(nT) \approx y(nT)$$

- The "impulse invariant approx" is a reasonable method, but not the only one, especially for finite order systems.
  - We want to approx $y(t) = \int_0^t h_c(\alpha)\, x(t-\alpha)\, d\alpha$

so
$$y(nT) = \int_0^{nT} \underbrace{h_c(\alpha)\, x(nT-\alpha)}_{f(\alpha)}\, d\alpha \approx T \sum_{i=0}^{n-1} f(iT)$$

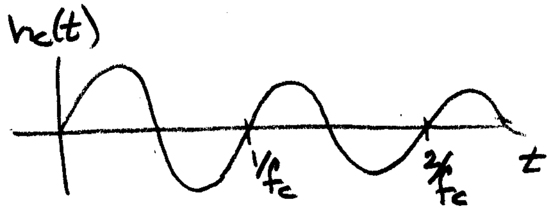


$$y(nT) = T \sum_i h_c(iT)\, x((n-i)T) = \sum_i h_d(i)\, x((n-i)T)$$

so
$$h_d(n) = T\, h_c(nT)$$

  - and sample at least twice as fast as highest frequency in the input or impulse response

- Example $\quad h_c(t) = e^{-t/\tau} \sin(2\pi f_c t)$



Find equiv digital filter by impulse invariance approx

$$h_d(n) = T\, e^{-nT/\tau} \sin(2\pi f_c T n)$$

BTW, you can write it as

$$h_d(n) = T \left(e^{-T/\tau}\right)^n \operatorname{Im}\left[\left(e^{j 2\pi f_c T}\right)^n\right]$$

$$= T \operatorname{Im}\left[\beta^n\right] \quad \text{where } \beta = \exp\left(-\frac{T}{\tau} + j\, 2\pi f_c T\right)$$