Introduction to Chapter 7: Discrete-Time and Digital Systems
Human and Machine Hearing: Extracting Meaning from Sound
by Richard F. Lyon
2018

Julius Smith
CCRMA, Stanford University

Signal Processing for Hearing Lectures 2023

December 5, 2023

Overview

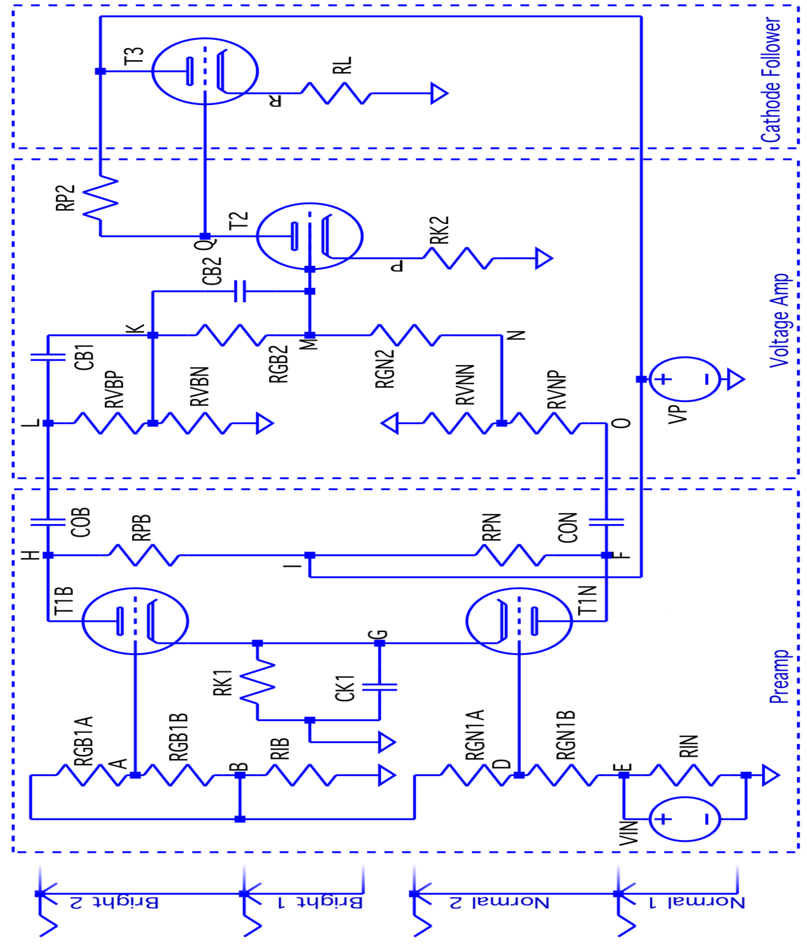
Goal: An *elementary introduction* to the topics of Chapter 7:

- Discrete-Time LTI Systems (Linear and Time-Invariant, or “shift-invariant” for images)
- Impulse Response and Convolution in *discrete time*
- Frequency in Discrete-Time Systems
- z Transform and its Inverse
- Unit Advance and Unit Delay Operators (Shift Operators)
- Filter Transfer Functions for the discrete-time case
- Sampling and Aliasing
- Mappings from Continuous-Time to Digital Systems
- Filter Design (mostly references)
- Digital Filters (a minimal start, but good pointers)
- Multiple Inputs and Outputs (see State Space)
- Fourier Analysis and Spectrograms (one good example)
- Additional References

Where I Am Coming From

Musician : Math : Physics : EE : Control : DSP : System ID : SAIL/CCRMA

(a) Some Gig



(b) Tube Amp

Center for Computer Research in Music and Acoustics (CCRMA)

Stanford AI Lab (SAIL) 60s-80s

Stanford Knoll (main campus)

John Chowning

Max Mathews

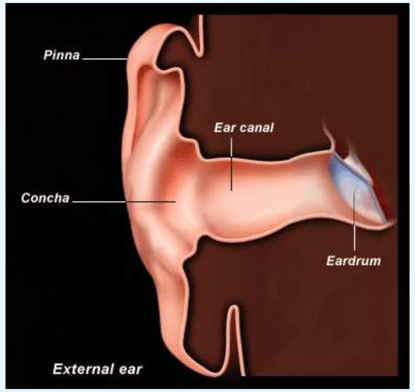
Also: John Grey, John Pierce, Roger Shepard, Earl Schubert, Ben Knapp, Malcolm Slaney, Takako Fujioka, . . . (Hearing Related)

JOS Courses Developed for CCRMA (and EE)

- **Music 320A**: Audio Spectrum Analysis
- **Music 320B**: Audio Filter Analysis and Structures
- **Music 420A**: **Physical** Audio Signal Processing
- **Music 421A**: **Time-Frequency** Audio Signal Processing

420A

$\longrightarrow$



421A

All four textbooks **free online**

Sampling Continuous-Time Signals

Let $x_c(t)$ denote a real-valued function of a continuous time variable t.
We often think of $x_c(t)$ as a *signal* in the *time domain*.

- *Sampling* can be expressed as $\boxed{x_d[n] = x_c(nT)}$, $n = [\ldots, -2, -1,]\, 0, 1, 2, \ldots$ where T denotes a fixed *sampling interval* in seconds
- n is an *integer* usually starting at time 0 and is called the *discrete-time index*
- Subscript c denotes functions of *continuous time* t while d denotes *discrete time* n
- We use *square brackets* $[\cdot]$ to indicate *discrete-time (integer) indexing*

Main Theorem:

Sampling in the time-domain corresponds to *aliasing* in the frequency domain:

$$\boxed{x_d[n] \;\longleftrightarrow\; X_d(e^{j\omega T}) = \frac{1}{T}\sum_{m=-\infty}^{\infty} X_c\left[j\left(\frac{\omega T}{T} + m\frac{2\pi}{T}\right)\right]}$$

The *Fourier dual* of this is also true
(sampling in the frequency domain $\leftrightarrow$ aliasing in the time domain)

Sampling of Properly Bandlimited Signals

If the signal $x_c(t)$ is *bandlimited* to *less than the sampling rate*, i.e., $X_c(\omega)$ is *zero* outside of the frequency interval $(-f_s/2, f_s/2)$, then $x_c(t)$ may be sampled at f_s samples per second with *no aliasing:*

$$\begin{aligned} X_d(e^{j\omega T}) &= \frac{1}{T}\sum_{m=-\infty}^{\infty} X_c\left[j\left(\frac{\omega T}{T} + m\frac{2\pi}{T}\right)\right] \\ &\approx \frac{1}{T} X_c(j\omega) \end{aligned}$$

for $\omega T \in (-\pi, \pi)$.
$X_d(e^{j\omega T})$ must *repeat periodically* outside of that interval
(along the unit circle of the z plane).

The s and z Planes

Generalized sinusoids in continuous and discrete time:

Continuous Time

$$
\begin{aligned}
e^{st} &= e^{(\sigma+j\omega)t} \\
&= e^{\sigma t} e^{j\omega t} \\
&= e^{-t/\tau}\left[\cos(\omega t) + j\sin(\omega t)\right]
\end{aligned}
$$

Discrete Time when $z = e^{sT}$

$$
\begin{aligned}
z^n &= \left(e^{sT}\right)^n = \left(e^{\sigma T + j\omega T}\right)^n \\
&= e^{\sigma n T} e^{j\omega n T} \\
&= e^{-nT/\tau}\left[\cos(\omega\, nT) + j\sin(\omega\, nT)\right]
\end{aligned}
$$

Laplace Transform

$$X_c(s) = \int_0^\infty x_c(t) e^{-st} dt$$

z Transform

$$X_d(z) = \sum_{n=0}^{\infty} x_d(n) z^{-n}$$

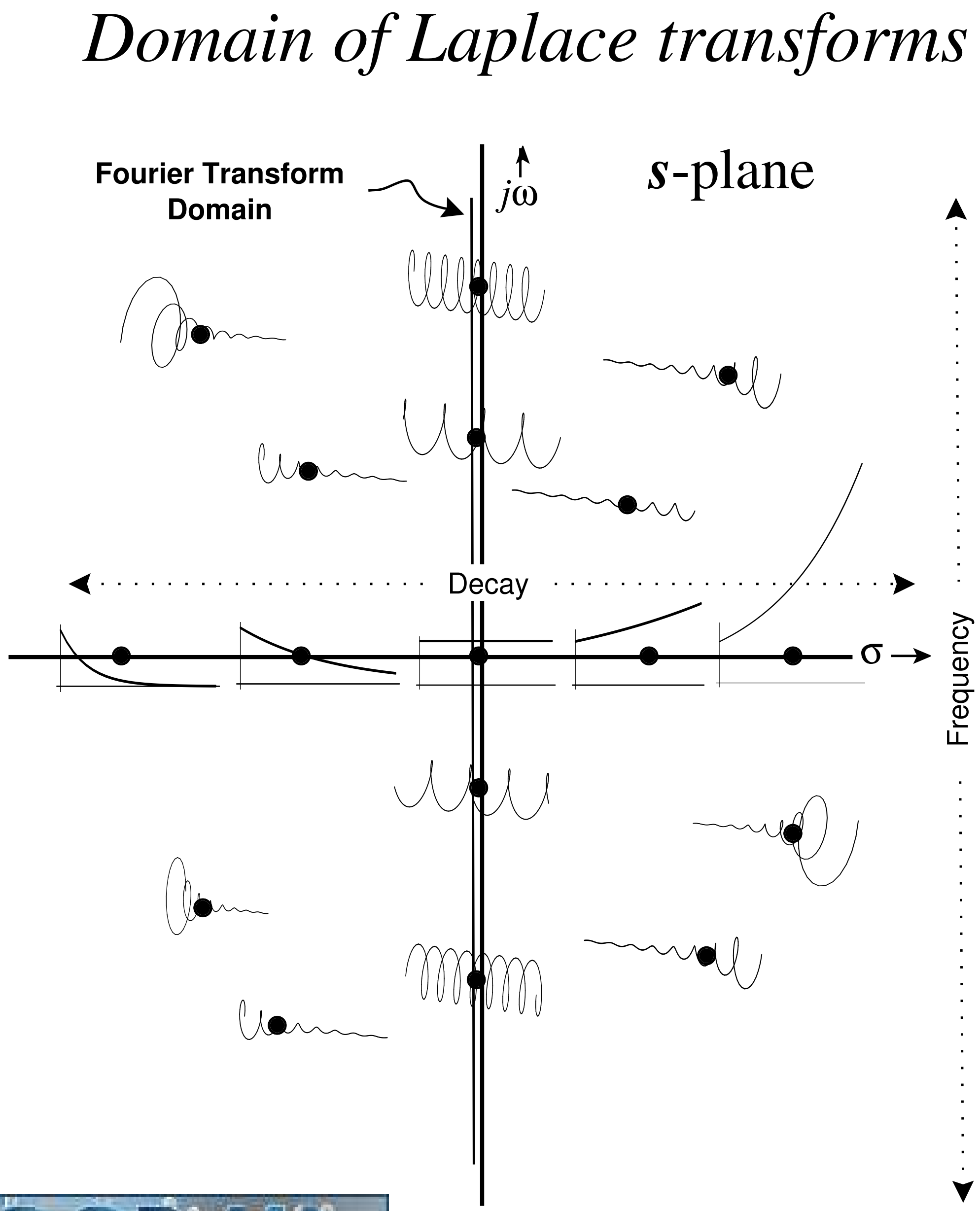
Fourier Transform (FT) ($s = j\omega$)

$$X_c(j\omega) = \int_0^\infty x_c(t) e^{-j\omega t} dt$$

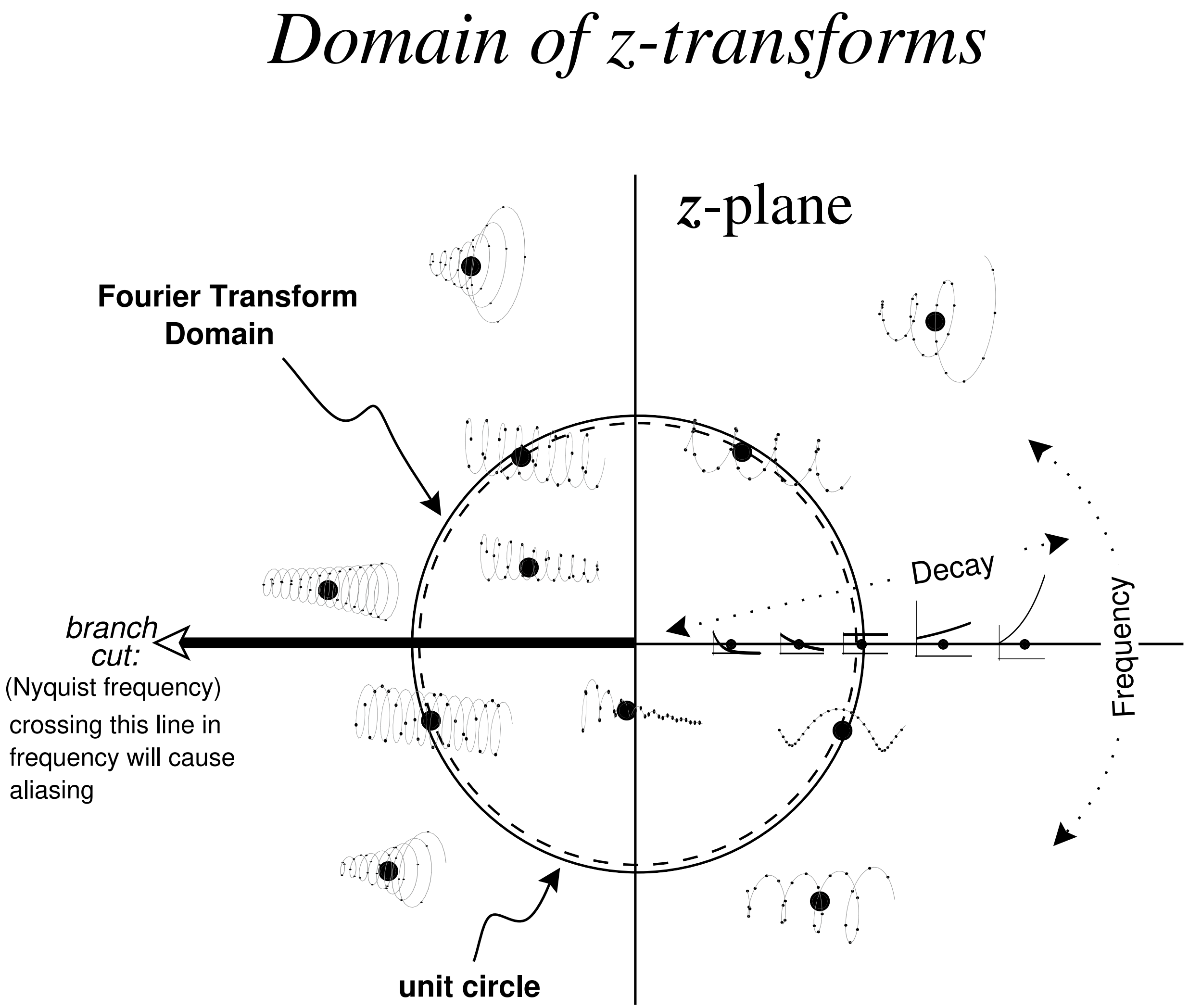
Discrete Time FT (DTFT) ($z = e^{j\omega T}$)

$$X_d(e^{j\omega T}) = \sum_{n=0}^{\infty} x_d(n) e^{-j\omega T}$$

Generalized Sinusoids e^{st} in the s Plane



Generalized Sinusoids z^n in the z Plane



Bilinear Transform

An alternative to *sampling* in the time-domain for *systems* (as opposed to signals) is to start in the *frequency domain* and apply the *Bilinear Transform:*

$$\boxed{s = \alpha \frac{1 - z^{-1}}{1 + z^{-1}}} \qquad z^{-1} = \frac{1 - s/\alpha}{1 + s/\alpha}$$

- α is any positive constant
- Setting $\alpha = 2/T$ matches *low frequencies* relative to the sampling rate f_s
- More generally, α can map *any one frequency* exactly
- See also Cayley (1846) and Möbius transforms
- Can show:
  - Analog frequency axis $s = j\omega$ (vertical axis in the s plane) maps exactly *once* to the digital frequency axis $z = e^{j\omega T}$ (unit circle in the z plane) $\Rightarrow$ *no aliasing*
  - The *left half* of the s plane (stability region for *poles*) maps to the *interior* of the unit circle in the z plane (its stability region) $\Rightarrow$ *stability preserved*

Oversampling Gives $z \approx 1 + sT$

At low frequencies and dampings, *i.e.*, near $s \approx 0$ and $z \approx 1$, we have the following low-frequency approximations (low relative to the sampling rate):

- **Bilinear Transform:**

$$= \frac{1+s/\alpha}{1-s/\alpha} = \left(1+\frac{s}{\alpha}\right)\left[1+\frac{s}{\alpha}+\left(\frac{s}{\alpha}\right)^2+\cdots\right] \approx 1+2\frac{s}{\alpha} = \boxed{1+sT}$$

when $\alpha = 2/T$

- **Basic Sampling:**

$$z = e^{sT} = 1+sT+\frac{(sT)^2}{2!}+\frac{(sT)^3}{3!}+\cdots \approx \boxed{1+sT}$$

It is good to oversample sufficiently so that there is no audible difference between the z-planes of signals and systems digitized separately by ordinary sampling and the bilinear transform (or multiple bilinear transforms as in Wave Digital Filters)

Impulse Response and Convolution

Recall the *Convolution Representation* from Section 7.3 of (Lyon 2018):

$$y[k] = (h * x)[k] = \text{“}h[k] * x[k]\text{”} = \sum_{n=-\infty}^{\infty} x[k-n]h[n]$$

This Convolution Representation exists for every *linear*, *time-invariant* (LTI) system

Linearity:

$$\mathcal{L}\{\alpha\, x_1 + \beta\, x_2\} = \alpha\, \mathcal{L}\{x_1\} + \beta\, \mathcal{L}\{x_2\}$$

for any (complex) scalars α and β, and any signals x_1, and x_2

Time Invariance:

$$\mathcal{L}_n\{\text{SHIFT}_N\{x_1\} = \mathcal{L}_{n-N}\{x\}$$

where

$$\text{SHIFT}_{N,n}\{x\} \triangleq x(n-N)$$

Shift and Convolution Theorems for z Transforms

It is easy to prove the z-transform

- **Shift Theorem:**
  `https://ccrma.stanford.edu/~jos/filters/Shift_Theorem.html`

- **Convolution Theorem:**
  `https://ccrma.stanford.edu/~jos/filters/Convolution_Theorem.html`

Spectrum Analysis

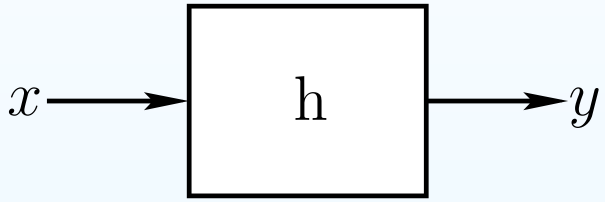
The *spectrogram* is said to have been invented at Bell Labs during World War II

- Speech Spectrogram Example in MATLAB/Octave:
  `https://ccrma.stanford.edu/~jos/mdft/Spectrogram_Speech.html`

- Same example translated to Python by ChatGPT-4
  `https://chat.openai.com/share/adfb9774-6fd6-40c1-b12a-4b6b8775a7a6`

Digital Filters

In a *convolution* $y(t) = (x * h)(t)$, if x is considered an *input signal*, then h can be described as a linear, time-invariant (LTI) *filter*:



- The roles of input and filter are interchangeable since $h * x = x * h$.
- In discrete-time, the *sampled* impulse response $h[n]$ gives a *digital filter:*

$$y[n] \;=\; (h * x)[n] \;=\; \sum_{m=-\infty}^{\infty} h[m]\, x[n-m]$$

- In practice, h must be a *finite impulse reponse (FIR)*.
- We create *infinite impulse reponse (IIR)* filters using *feedback*.

Recursive Digital Filters (“Infinite Impulse Reponse” (IIR))

Example: One-pole recursive smoother, with gain g and pole at $z = p > 0$:

$$
\begin{aligned}
y[n] &= g\,x[n] + p\,y[n-1] \\
\Longleftrightarrow\; Y(z) &= g\,X(z) + p\,z^{-1}Y(z) \\
\Longleftrightarrow\; Y(z) &= \frac{g}{1 - p\,z^{-1}} X(z) \\
\Longleftrightarrow\; H(z) &= \frac{g}{1 - p\,z^{-1}} \;=\; g\left[1 + p\,z^{-1} + (p\,z^{-1})^2 + \cdots\right] \\
\Longleftrightarrow\; h[n] &= g\,p^n
\end{aligned}
$$

- This happens to be a first-order Butterworth lowpass filter for $p > 0$ (among others)
- The FAUST distribution supports real-time Butterworth digital filters of all orders
- FAUST compiles to C, C++, Java, JAX, Julia, Rust, VHDL, Web Assembly, and more

Supplementary Online References

- Introduction to the Discrete Fourier Transform (DFT):
  `https://ccrma.stanford.edu/~jos/mdft/`

- Introduction to Digital Filters:
  `https://ccrma.stanford.edu/~jos/filters/`

- Audio Signal Processing in Faust:
  `https://ccrma.stanford.edu/~jos/aspf/`

Download These Overheads:
Web-search for "Julius Smith CCRMA" and scroll to the bottom for
`https://ccrma.stanford.edu/~jos/pdf/SPFH7.pdf`

Overview

Goal: An *elementary introduction* to the topics of Chapter 7:

- Discrete-Time LTI Systems (Linear and Time-Invariant, or “shift-invariant” for images)
- Impulse Response and Convolution in *discrete time*
- Frequency in Discrete-Time Systems
- z Transform and its Inverse
- Unit Advance and Unit Delay Operators (Shift Operators)
- Filter Transfer Functions for the discrete-time case
- Sampling and Aliasing
- Mappings from Continuous-Time to Digital Systems
- Filter Design (mostly references)
- Digital Filters (a minimal start, but good pointers)
- Multiple Inputs and Outputs (see State Space)
- Fourier Analysis and Spectrograms (one good example)
- Additional References