Contents

1 Course Description 1

2 Administrative Information 1
   2.1 Announcements ................................. 1
   2.2 Assignments ..................................... 2
   2.3 Exams ............................................. 2
   2.4 Grading ........................................... 2
   2.5 Office Hours and Getting Help .................... 2
   2.6 Computer Usage .................................. 3

3 Textbooks 3

4 The Partially Flipped Classroom 3

5 A Recipe for Learning 4

6 320B Schedule and Pointers 4
   6.1 Section 1: Linearity and Time Invariance: Time-Domain Representations ............. 4
   6.2 Section 3: Digital Filter Design ......................................................... 5
   6.3 Section 4: Quality Factor Q, Allpass Filters, State Space, State Variable Filter .... 6
   6.4 Section 5: Voice Synthesis, F0 Estimation, Cepstra, Converting to Minimum Phase ... 9
Music 320 A & B: Introduction to Digital Audio Signal Processing

1 Course Description

Music 320 is a two-quarter first-course in digital signal processing with applications in computer music and audio.

The lectures present fundamental elements of digital audio signal processing, such as sinusoids, spectra, the Discrete Fourier Transform (DFT), digital filters, z transforms, transfer-function analysis, and basic Fourier analysis in the discrete-time case. Matlab is used for in-class demonstrations and homework/lab assignments. The labs focus on practical applications of the theory, with emphasis on working with waveforms and spectra, "getting sound", and developing proficiency in the matlab language.

Prerequisites: High-school level algebra and trigonometry, some calculus, and prior exposure to complex numbers.

2 Time and Place

Term: Autumn and Winter Quarters
Location: CCRMA Classroom (Knoll 217)
Lectures: Tuesdays and Thursdays 3:00–4:50 PM
Units: 2–4
Instructor: Julius O. Smith (jos@ccrma.stanford.edu)
TA: Orchi Das (orchi@ccrma.stanford.edu)
Office Hours: See “Office Hours and Getting Help” below
Schedule: See “Schedule and Pointers” below

2 Administrative Information

2.1 Announcements

Class announcements are often made via email. For this we are presently using Piazza:

https://piazza.com/stanford/fall2017/music320a/home

If you signed up for the class in axess before the first day of classes, you should receive an invitation from Piazza to join the class (using the email address known to axess). Otherwise, please join by visiting the above URL and entering your preferred email address.

1http://ccrma.stanford.edu/~jos/intro320/Office_Hours_Getting_Help.html
2http://ccrma.stanford.edu/~jos/intro320/
2.2 Assignments

There are five homework/lab assignments, each covering roughly two weeks of the course. In each two-week “section”, the first week is devoted primarily to theory while the second week is focused more on software and applications. Thus, each assignment contains both a theory and laboratory part. The lab portion typically requires programming in matlab.

Each assignment is typically announced on Tuesday in the first week of the section. The theory part is normally due the following Tuesday at 3:15 pm in the 320 mailbox (located in the Knoll, central wing, second floor, facing the printer). The lab part is normally due by midnight the following Friday, i.e., at the end of the two-week section.

For lab assignments, we will be using the Canvas website. To sign up, go to the Canvas website and find Music320A. Once you are enrolled in the class, you can upload your matlab files in the “drop box” on the left menu.

See §2.5 below regarding obtaining help with theory and lab assignments.

Regarding late homeworks, 7 free late days are allowed (with hours rounded up to the nearest day). Late homeworks beyond this will be penalized at 5% per day. When using late days, write the number of late days used at the top of the assignment (date and time).

Students are encouraged to discuss the homework assignments with each other. It is fine to learn from a classmate how to solve any of the homework problems, but each student is responsible for carrying out and writing up the assignments individually. It is an honor code violation to copy the work of others.

2.3 Exams

The final examination will be held in the CCRMA Classroom (Knoll 217) on the University-assigned date, also listed for convenience in the class schedule (§6 on page 4).

2.4 Grading

Grades are based on the homeworks/labs (60%), and the final exam (40%). There are also bonus points available based on general participation. The weightings may be changed as we see fit.

2.5 Office Hours and Getting Help

We will be using Piazza for sharing answers to posted questions with the whole class. To sign up, see the 320 Piazza site. It is free and allows you to view past questions from other students, and discuss questions together. Try it first for any homework questions you may have. You are also welcome, of course, to catch us whenever you see us at CCRMA, such as during office hours, etc.

TA weekly office hours are Wednesday evenings 7:30-9:30 PM in the CCRMA Ballroom (2nd floor). Meetings with JOS are arranged via email for half-hour slots after class, or other times when necessary.

https://canvas.stanford.edu
https://www.piazza.com
https://piazza.com/stanford/fall2017/music320a/home
2.6 Computer Usage

Lab exercises will be computer based. All students may obtain a computer account at CCRMA in order to use the computer facilities. It is also possible to work entirely on your own computer, as long as you have the necessary software. However, note that some course materials are restricted to on-campus access, so you should have at least one Stanford computer account from which you access those.

Here is how to obtain a CCRMA computer account:

https://cm-knoll.stanford.edu/usersignup

Note: This link only works at CCRMA.

Once you have your account, please log in at CCRMA and take a look at the User’s guides tab in the left-frame menu of the main CCRMA website to learn more about computer usage and other facilities at CCRMA.

3 Textbooks

Music 320A (fall) is based on assigned chapters of

Mathematics of the Discrete Fourier Transform (DFT) by Julius O. Smith

Music 320B (winter) is based on assigned chapters of

Introduction to Digital Filters by Julius O. Smith

See §6 for the list of assigned chapters. Both books are fully available on-line. Softcover versions are available from Amazon.com.

4 The Partially Flipped Classroom

With the lectures recorded, class time is freed up for other activities. Here is how a typical “partially flipped class” is organized:

• Q&A session on the reading/video content
• Review of main points in the reading/videos
• Demos in support of the reading/videos
• Presentation of the homework/lab assignment
• Worked problems similar to those in the homework
• Matlab session on theory/lab-related topics
• Live coding in matlab

Additional available time may be devoted to

• More demos
• More discussion

http://ccrma.stanford.edu/guides/
http://ccrma.stanford.edu/~jos/mdft/
http://ccrma.stanford.edu/~jos/filters/
5 A Recipe for Learning

Learning something new requires multiple passes on the material. For example:

1. Do the assigned reading at a fixed pace to get a picture of what’s covered
2. Watch the lecture videos, pausing and taking notes on anything newly learned
3. Make a first pass on the homework, flagging and skipping when stuck on a problem
4. Discuss nonobvious homework problems with other students, the TA, and/or JOS
5. Write up the homework problems, everything now understood
6. Exam prep: Reread the text for full comprehension
7. Exam prep: Reread your notes
8. Prepare your one-page summary of the course allowed in the exam
9. Exam experience: Exercise in problem solving using the material

These multiple engagements result in a good amount of learning.

6 320B Schedule and Pointers

6.1 Section 1: Linearity and Time Invariance; Time-Domain Representations

- Reading:
  - Chapters 1 and 2 of Introduction to Digital Filters
  - Chapter 4 (Linearity and Time Invariance) and Chapter 5 (Time Domain Filter Representations) of Introduction to Digital Filters
  - First section of Chapter 9 (Implementation Structures) on the Four Direct Forms
  - Optionally peruse the Music 421 overheads pertaining to acyclic convolution
  - Assignment 1

- Lecture Videos (Total Viewing Time ≈ 2 Hours):

  IMPORTANT NOTICE: The videos are hosted on YouTube and they use annotations for corrections and supplementary information. These annotations are not supported on mobile devices. It is therefore unfortunately important to view these videos in a Web browser on a desktop/laptop computer.

  "https://ccrma.stanford.edu/~jos/filters/filters.html"
– Linear Time-Invariant (LTI) Filters, Convolution, Ideal Lowpass, Guard Band, Transition Band, Simplest Lowpass Filter, Impulse Response, DTFT, Frequency Response, Amplitude Response, Phase Response, Linear Phase, Sinewave Analysis,[10] [38:36]
– Derivation of Convolution from Linearity and Time-Invariance (LTI) (Superposition) [2015][1] [29:08]
– Recursive Filters, Simplest Lowpass, Phase Delay, Group Delay[12] [28:01]
– Supplementary: FAUST in the Classroom[13] [41:00]
– Supplementary: FAUST Intro[14] [26:00]
– Supplementary: FAUST Implementation of the Simplest Lowpass Filter[15] [18:22]
– Simplest RECURSIVE LPF, Pole Gain, PFE, Time-Constant of a Pole, Stability Pole, Bandwidth, Laplace Transform, s-plane poles and zeros, s-plane pole corresponds to exponential[16] [38:37]
– Direct Form Digital Filters, Transposing a Flow Graph, Transposed Direct Forms 1 and 2, Direct Form 1 Biquad, Direct Form 2 Biquad, Transposed Direct Form 2 Biquad, Interpolated Delay-Line Read, Interpolated Delay-Line Write = Transpose of Read[17] [14:35]
– Simplest Mechanical LPF: Ideal Mass on Frictionless Surface, Newton’s law of motion f=ma, Analog Transfer Function for Driving-Force Input, Velocity Output, Admittance (Mobility) of a Mass[18] [5:31]

6.2 Section 3: Digital Filter Design

• Reading:

  – Chapter 10 (Elementary Audio Digital Filters)
  – Chapter 11 (Filters Preserving Phase)
  – Appendix I (Recursive Digital Filter Design)
  – Appendix I.2 (Butterworth Filters)
  – Appendix K (Digital Filtering in FAUST and Pd)
  Supplementary: Audio Signal Processing in FAUST

[10] https://www.youtube.com/watch?v=p19QzBxnhvG
[12] https://www.youtube.com/watch?v=r0fg8eZAKGs
[13] https://www.youtube.com/watch?v=21Et7dszI0O
[14] https://www.youtube.com/watch?v=qE1_UzQZmmN
[15] https://www.youtube.com/watch?v=jRcKG1MHE9A
[16] https://www.youtube.com/watch?v=1J7mmqVBFk
[17] https://www.youtube.com/watch?v=q2UcTsHkHBQ
[18] https://www.youtube.com/watch?v=BUkMast6_U
[19] https://www.youtube.com/watch?v=VWJpHhjf8c
• **Assignment 2**

• **Lecture Videos (Total Viewing Time ≈ 2 Hours):**
  
  – Analog Filters Reviewed: Transfer Function, Frequency Response, Power Response; Analog Lowpass Design, Maximally Flat Passband, Butterworth Filter[^20] [30:46]
  
  
  – Supplementary: Introduction to Functional Audio Stream (FAUST): Simplest Lowpass, Utilities in Faust’s filter.lib[^22] [37:30]
  
  – Supplementary: More FAUST: Testing filters using faust2octave[^23] [27:32]
  
  – Butterworth Power Response, Analytic Continuation, Butterworth Poles, Matlab butter() Function[^24] [7:02]
  
  – Example Butterworth Filter of Order 2, Digitization via Bilinear Transform, Frequency Prewarping[^25] [18:49]
  
  – Digital Filter Design and Implementation in Matlab: Noise Removal via Lowpass Filtering, Create Sinusoid and Noise, Matlab’s butter(), filter(), fdatool (Filter Design and Analysis tool), Simulating a Telephone Channel Bandwidth[^26] [10:30]
  
  – Matlab: freqz(), freqs(): Continuous Butterworth Filter Analysis; Converting to Second-Order Sections in Matlab using tf2sos(); Viewing Butterworth Poles in Matlab using zplane(); Excess Delay at Filter Cutoff; grpdelay(); Elliptic Filters using ellip(); Ripple; impz(): Zero Phase versus Minimum Phase (Pre-Ring versus Post-Ring); Minimum-Delay Property of Minimum-Phase Filters; Partial Fraction Expansion in Matlab using residue(), residuez(), or residued[^27] [31:10]

6.3 **Section 4: Quality Factor Q, Allpass Filters, State Space, State Variable Filter**

• **Reading:**
  
  – Appendix E.7 (Quality Factor (Q))
  
  – Appendix C (Allpass Filters)
  
  – Laplace Analysis of a Force-Driven Mass[^28]
  
  – State-Space Formulation of Digital Filters[^29]

[^20]: https://www.youtube.com/watch?v=doDMmZfEfbg
[^21]: https://www.youtube.com/watch?v=pUtUrzVHF3Q
[^22]: https://www.youtube.com/watch?v=FEm7dpAdp6I
[^23]: https://www.youtube.com/watch?v=Ao1ZriZI8nY
[^24]: https://www.youtube.com/watch?v=hhhuAzBU1eU
[^25]: https://www.youtube.com/watch?v=UcThn4B5tuU
[^26]: https://www.youtube.com/watch?v=io8g5x0_EgRM
[^27]: https://www.youtube.com/watch?v=MxxDS01EaSo
- **State-Variable Filter**

- **Supplementary:** State-Space Introduction in Music 420

- **Supplementary:** State-Space Canonical Forms

- **Supplementary:** Linkwitz-Riley Crossovers: A Primer

- **Assignment 8**

- **Lecture Videos (Total Viewing Time ≈ 3 Hours):**

  - Quality Factor (Q) of a Resonator²⁸ [7:12]
  - Complex One-Pole Resonator and its Q; Canonical Form of a Biquad (s-plane); Mechanical and Electrical Resonators; Limiters, Compressors, Expanders²⁹ [39:30]
  - Filter Decay Time is about Q Periods³⁰ [3:16]
  - Bilinear Transform Frequency Mapping, Analog Computers, State Space Formulation, Physical Derivation of Bilinear Transform, State Variable³¹ [38:29]
  - Minimum Phase Filters and Signals; Allpass Filters: Poles and Zeros, Graphical Amplitude and Phase Response, Biquad Realization, Phasing; Allpass-Minimum-Phase Decomposition³² [28:44]
  - Allpass Filters in z and s Planes; Instability as Noncausality; Laurent Series; Bilateral DTFT; Cepstrum; Converting Arbitrary Spectra to Minimum-Phase Form³³ [39:13]
  - Repeated Poles at \( s = 0 \)
    - One Pole at DC in the s Plane³⁶ [14:17]
    - Mechanical Integrator using a Mass³⁵ [3:46]
    - Integrator made by a Spring or Inductor³⁶ [5:46]
    - One Pole at DC in the s Plane, Continued³⁵ [1:31]
    - Frequency Response of an Integrator³⁸ [5:05]
    - Repeated Poles at DC³⁹ [4:16]
    - General Transfer Function of a Pile of Poles at DC⁴⁰ [3:38]
    - Impulse Response of a Pile of Poles at DC⁴¹ [3:22]

²⁸ https://www.youtube.com/watch?v=V04yxqgBrYu
²⁹ https://www.youtube.com/watch?v=xeClzRkUfQI
³⁰ https://www.youtube.com/watch?v=kPKZQ16EdcU
³¹ https://www.youtube.com/watch?v=GRpAqevbUWs
³² https://www.youtube.com/watch?v=Cj6Vj3p6kXH
³³ https://www.youtube.com/watch?v=mCwj1VeXq44
³⁴ https://www.youtube.com/watch?v=D1HfzJHuWzE
³⁵ https://www.youtube.com/watch?v=c11Iv2YbJ3U
³⁶ https://www.youtube.com/watch?v=qPLyNgE7rF4
³⁷ https://www.youtube.com/watch?v=1qXHf2a3yw
³⁸ https://www.youtube.com/watch?v=VLtpw4eU4uc
³⁹ https://www.youtube.com/watch?v=5s6umF7I4T4
⁴⁰ https://www.youtube.com/watch?v=6_VRX5Fvdiw
⁴¹ https://www.youtube.com/watch?v=d6uQYhQteaw
– The *State Space* Formulation of Linear Systems [2016 flipped-class review]
  * Adding Feedback around the Integrator Chain, Derivation of the State Space Formulation [42] [10:46]
  * State Space Formulation, Continued [43] [1:33]
  * State Space Overview [44] [11:50]
  * Force Driven Mass [45] [4:49]
  * General Discussion of State Space [46] [8:48]
  * Defining State Variables [47] [4:51]
  * State Variable Choice Summary [48] [5:14]
  * General State Space Model and Digitization via Backward Euler [49] [5:16]
  * State Space Converts Nth-Order to Vector First-Order [50] [2:08]

– Moog VCF [51] [23:29]
– Supplementary: Moog VCF Live-Coded in Faust
  * Moog VCF Live-Coded in Faust [52] [30:49]
  * Moog VCF in Faust, Review [53] [5:02]
  * Moog VCF in Faust, Frequency Responses [54] [7:39]
  * [Can Skip] Q-Correction and Gain-Correction Tables [55] [3:38]
– State Variable Lowpass, Bandpass, and Highpass
  * Normalized Biquad Lowpass Filter, Continuous Time [56] [19:29]
  * State Variable Realization of Normalized Biquad Lowpass [57] [31:22]
  * State Variable Filter Lowpass/Bandpass/Highpass [58] [3:00]
  * State Variable Filter LP/BP/HP Frequency Scaling and Digitization [59] [11:35]
– Note on Repeated Poles [60] [4:40]

[42] https://www.youtube.com/watch?v=a_oM8rYPHc
[43] https://www.youtube.com/watch?v=ZNQ26879Ch0
[44] https://www.youtube.com/watch?v=Ygq66m9WrBk
[45] https://www.youtube.com/watch?v=zUlbjg6pQQ8
[46] https://www.youtube.com/watch?v=F0ku7LwHr10
[47] https://www.youtube.com/watch?v=45QwWhbJMVU
[48] https://www.youtube.com/watch?v=IgggFhhoUo
[49] https://www.youtube.com/watch?v=rbrJnnh_dU
[50] https://www.youtube.com/watch?v=zd02nKQeAY
[51] https://www.youtube.com/watch?v=Kxo1BNsZHY
[52] https://www.youtube.com/watch?v=WlvpGN_U11A
[53] https://www.youtube.com/watch?v=VZ2e0ChjTqe
[54] https://www.youtube.com/watch?v=GzG1b1Ji
[55] https://www.youtube.com/watch?v=QmOnjDeQYY
[56] https://www.youtube.com/watch?v=hjH2M0a3yk
[57] https://www.youtube.com/watch?v=9cxaELOwU
[58] https://www.youtube.com/watch?v=Zpot57KOE
[59] https://www.youtube.com/watch?v=Lz2SwTFB=0
[60] https://www.youtube.com/watch?v=D6_AK7mFqQ
6.4 Section 5: Voice Synthesis, F0 Estimation, Cepstra, Converting to Minimum Phase

- Reading:
  - Chapter 12 (Minimum Phase Digital Filters)
  - Supplementary: Spectral Envelopes via Cepstrum or LPC [from Music 421 overheads]

- Lecture Videos:
  - Minimum-Phase Spectra Play-List
  - Complex and Real Cepstrum, Quefrency [7:54]
  - Mel Frequency Cepstral Coefficients (MFCC); Bark and Equivalent Rectangular Bandwidth (ERB) Psychoacoustic Frequency Scales based on Critical Bands of hearing [7:54]
  - Complex Cepstrum Derived; Converting Mixed-Phase Signals to Minimum Phase [5:39]
  - Series Expansion of Log of $1/(1-x)$ [7:57]
  - Series Expansion of Log Transfer Function in Factored Form [5:26]
  - Contribution of Zeros to the Complex Cepstrum [3:15]
  - Contribution of Poles and Zeros (Inside and Outside the Unit Circle) to the Complex Cepstrum; Nonparametric Cepstral Folding Method for Converting Mixed Phase to Minimum Phase using the FFT; Testing for Time Aliasing [9:14]
  - Nonparametric Cepstral Folding Method in Matlab: \texttt{minphasespec()}, \texttt{fold()}, \texttt{invfreqz()} [6:11]
  - Review of another Cepstral Folding Example [19:08]
  - Minimum Phase Conversion by Spectral Factorization or Cepstral Method [5:45]
  - Minimum Phase Conversion by the Cepstral Method, Continued [12:10]
  - Cepstral Method Code, then CollideFX Demo by Chet Gnegy [55:51]
  - Supplementary: Voice Vowel Synthesis in Faust [11:26]
  - Supplementary: Voice Vowel Synthesis in Faust, Continued []

---

[61] https://www.youtube.com/watch?v=Tlk6CLc1PPU
[62] https://www.youtube.com/watch?v=201BBFeQCGE
[63] https://www.youtube.com/watch?v=OH1z4047U
[64] https://www.youtube.com/watch?v=Vrs1b9U7HaI
[65] https://www.youtube.com/watch?v=3gCA_rj13k
[66] https://www.youtube.com/watch?v=nIY8EAA4peA
[67] https://www.youtube.com/watch?v=ya01UUTCko
[68] https://www.youtube.com/watch?v=PTx0capRmMU
[69] https://www.youtube.com/watch?v=V7K4reT94PE
[70] https://www.youtube.com/watch?v=7GcKMcqVao
[71] https://www.youtube.com/watch?v=E58NEeaFxFVg
[72] https://www.youtube.com/watch?v=mkt3bR0t14
[73] https://www.youtube.com/watch?v=GR97SMvS4Fw
[74] https://www.youtube.com/watch?v=
– Supplementary: Voder [0:43]
– Supplementary: Phonem for iPad by Wolfgang Palm

\[7\] https://www.youtube.com/watch?v=e5gQBei-z-c