Chapter 3. Meeting 3, Foundations: Envelopes, Filters, Modulation, and Mixing

3.1. Announcements

- Bring controllers (not amps) to next class on Monday; first class with amps and controllers will be meeting 5, Wednesday, 16 February

3.2. Review Waveshapes and Wavetables

- Noise, sine, waveshapes, and audio files
- Using [phasor~] to read a table

3.3. Review Pd Tutorial 1

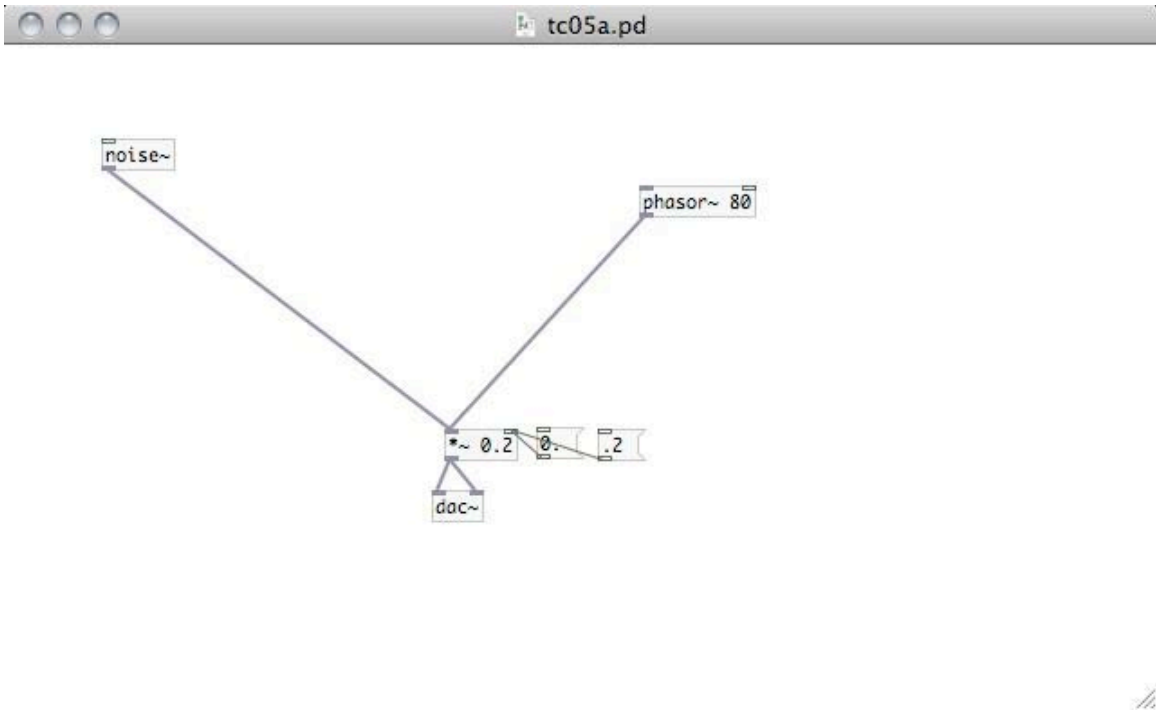
- ?

3.4. Reading: Tanaka: Sensor-Based Musical Instruments and Interactice Music

- Tanaka, A. 2009. "Sensor-Based Musical Instruments and Interactive Music." In R. T. Dean, ed. *The Oxford Handbook of Computer Music*. Oxford University Press, pp. 233-257. (233-243)
- Why might it be problematic to define an instrument?
- Why are instruments not like normal tools?
- Why might the notion of idiomatic performance technique be important when working with new musical interfaces?

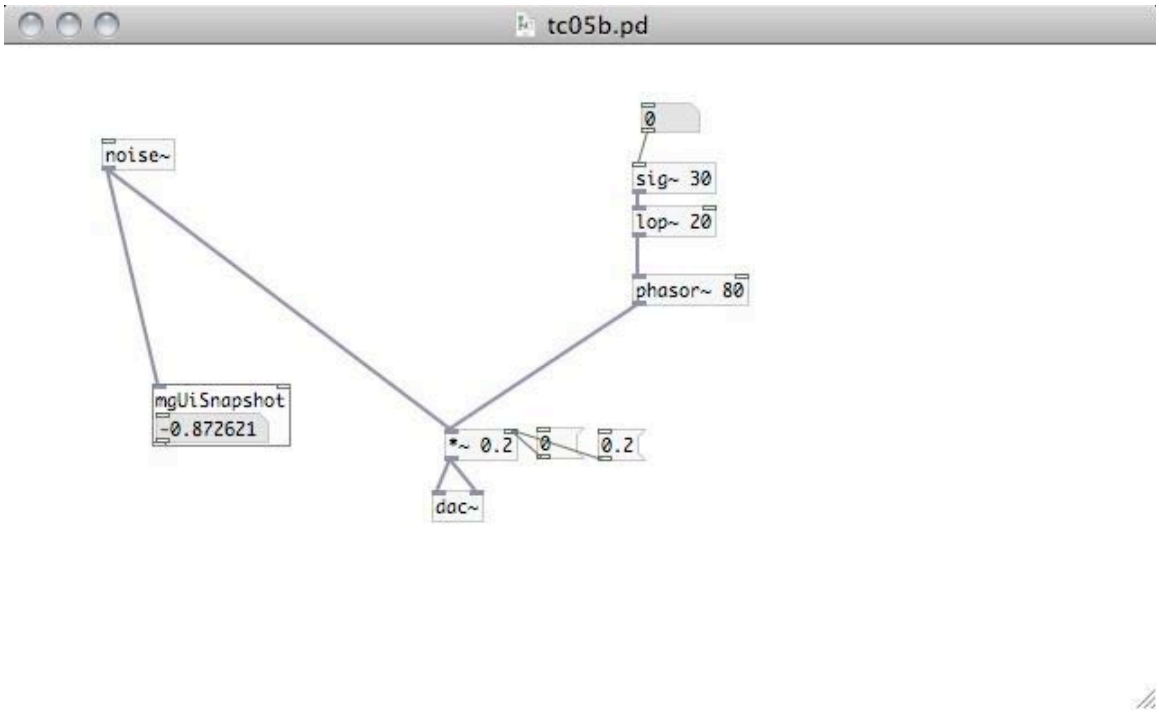
3.5. Adding and Multiplying Signals

- Adding signals [+~] is mixing
- Two signals provided into the same inlet automatically sum
- Multiplying signals [*~] is scaling
- Example: mixing [noise~] and [phasor~]



3.6. Converting Signals to Data and Data to Signals

- When converting from data to a signal, always use [sig~] to [lop~ 20]
- When converting from signal to data, use [snapshot~], [mgSnapshot], or [mgUiSnapshot]
- Example: getting values from [noise~] with [mgUiSnapshot], controlling [phasor~] frequency with [sig~] and [lop~ 20]



3.7. Envelopes

- Envelopes are signal scalars
- Generally unipolar signals that move from 0 to 1 and back to 0
- Can be used as amplitude scalars, or to provide dynamic control values to other parameters
- A number of common shapes define common types of articulations
- Can be classified as mono-triggered or bi-triggered
- Some musical events can be thought of as having a single on trigger: off is implied after a certain duration: other musical events can be thought of as having two triggers: off is trigered after an unknown duration

3.8. Envelopes: Common Shapes

- AR: Attack / Release

  Example parameters: 20 2000

  Examples: [mgEnvlBtAr], [mgEnvlMtAr]

- ADSR: Attack / Decay / Sustain / Release

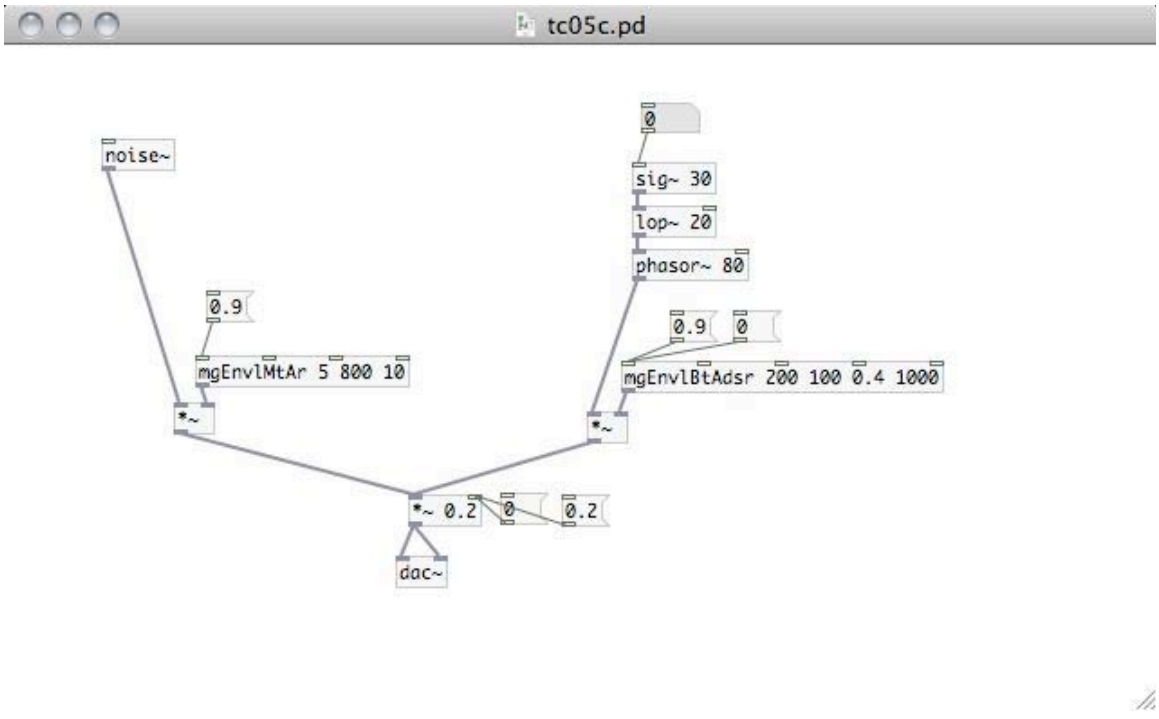
Example parameters: 200 80 .6 1200

Examples: [mgEnvlBtAdsr], [mgEnvlMtAdsr]

- Envelopes might be mixed to produce more interesting shapes
- Example: martingale/demo/envelopes.pd

3.9. Envelopes: Triggering

- Trigger start envelope
- Martingale envelopes use numerical values: non-zero sets peak amp and triggers envelopes; 0 stops envelope
- A mono trigger starts and stops with a single trigger
- A bi-trigger envelope needs a trigger to start and a trigger to end
- Example: martingale/demo/envelopes.pd
- Example: enveloping two signals



3.10. Filtering

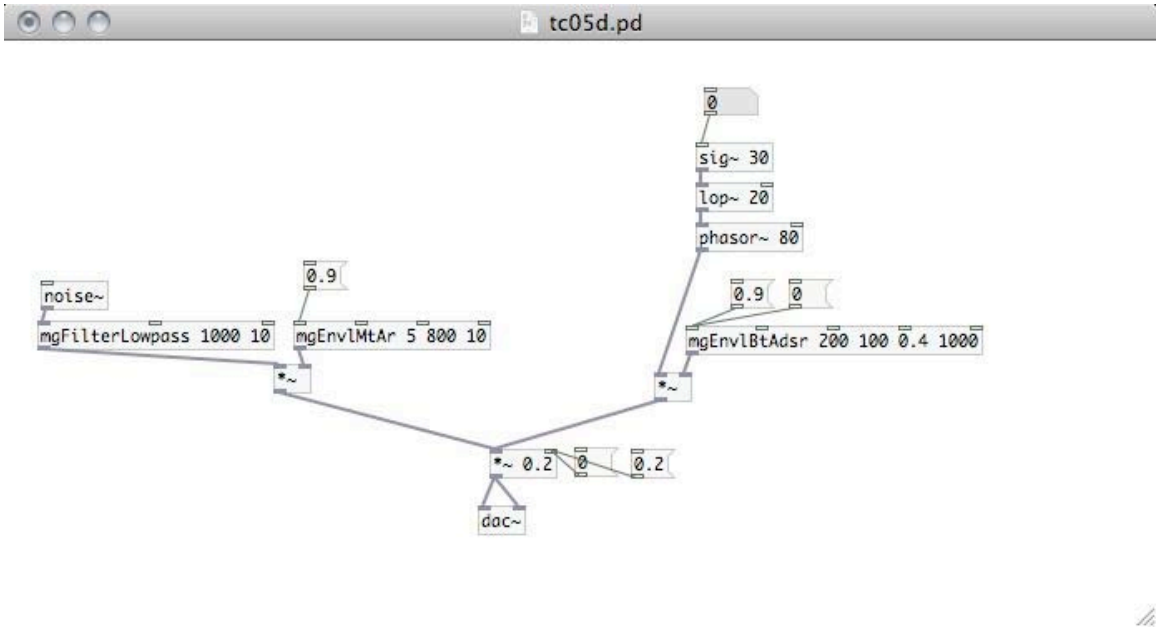
- Filters selectively boost or cut the amplitude of frequency components
- Equalizers are filters (a distinction is not useful)
- Filters cannot add frequencies that are not present in the source
- Filter shapes are depicted with frequency-domain graphs with a 0-centered amplitude change
- Filters can be used on audio signals or on control signals
- Filters can be chained in series or used in parallel

3.11. Filtering: One Parameter

- Cutoff frequency
- High pass filters (HPF), low pass filters (LPF)
- Examples: [lop~] and [hip~]
- Using an [lop~] on a control signal filters out quick changes, smoothing the signal
- Example: martingale/demo/filters.pd
- Note: may need to convert signals to event data with [mgSnapshot] to control parameters

3.12. Filtering: Two Parameter

- LPF and HPF filters with cutoff frequency and resonance
- Examples: [moog~], [mgFilterLowPass], [mgFilterHighPass]
- Examples: adding a [mgFilterLowPass]



- Bandpass and notch filters: center frequency and gain
- Examples: [mgFilterBandPass], [mgFilterNotch]
- Example: martingale/demo/filters.pd

3.13. Filtering: Three Parameters

- Parametric filter: cutoff frequency, gain, and bandwidth
- Examples: [mgFilterParametric]
- High and low shelf filters: cutoff frequency, gain, and shape
- Examples: [mgFilterLowShelf], [mgFilterHighShelf]

3.14. Modulation

- Modulation is varying a parameter over time
- Modulation can be executed with an envelope (for intermittent modulation) or a low-frequency oscillator (LFO; for continuous modulation)

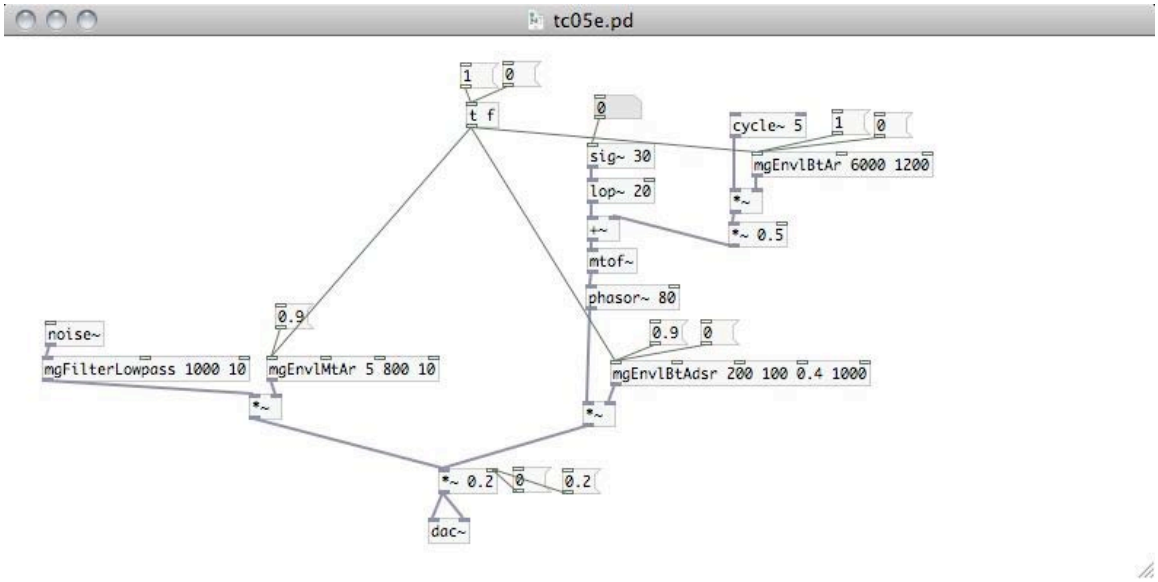
3.15. Modulation: Amplitude

- Sub-sonic AM is tremolo

- Sonic AM produces new overtones (sidebands, or rings)
- Example: martingale/demo/modulationAmBasic.pd

3.16. Modulation: Frequency

- Sub-sonic (less than 30 Hz) FM is vibrato
- Sonic FM produces new overtones
- Sonic FM produces new overtones
- Example: martingale/demo/modulationFmBasic.pd
- Example: creating a vibrato scaled by an envelope



3.17. Listening: Tenney

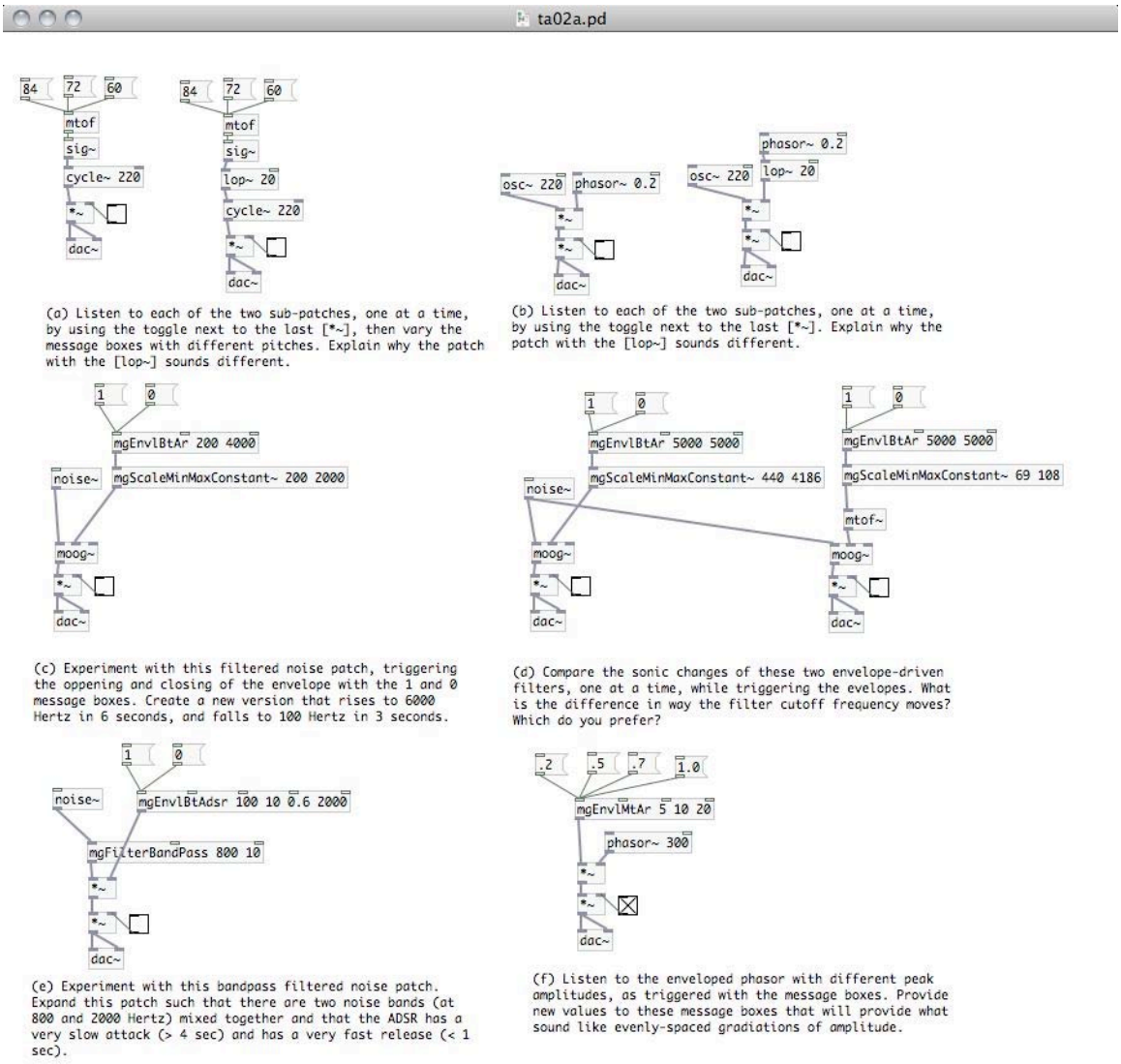
- Mathews: "to my mind, the most interesting music he did at the Laboratories involved the use of random noises of various sorts." (1980, p. 17)
- Listening: James Tenney, *Analog #1: Noise Study*, 1961

3.18. Listening: Koenig

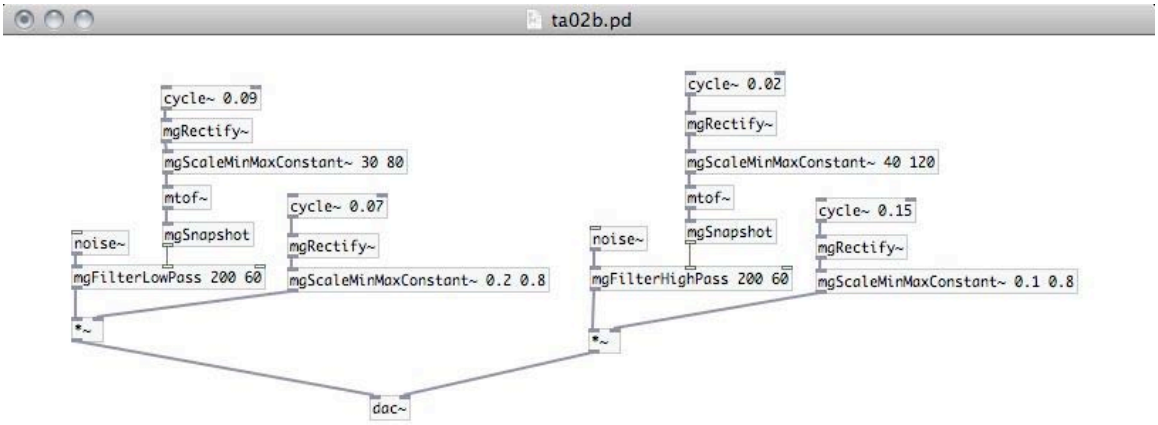
- Listening: Gottfried Michele Koenig, *Funktion Grau*, 1969

3.19. Pd Tutorial 2

1. The following examples demonstrate operations with Pd. Recreate the following patch components in a Pd file and and answer the provided questions as comments in the Pd file.



2. Create and extend the following patch.



(a) This patch creates waves of noise, where both the timbre of the noise and the amplitude of the noise are smoothly varied with [cycle~] objects. Recreate this patch and ellaborate it by adding more noise bands, using different filters, and/or controlling filter and amplitude parameters in different ways.

21M.380 Music and Technology: Live Electronics Performance Practices
Spring 2011