Project 5. Real-Time Performance Interim Report Due: April 15
Preview Due: Week of April 21
Concert: April 27 (afternoon)
Final Report Due: April 28

Who is in the group?

Eric Dissinger, Noelle Jung, Matt Kent, David Lu, Vivian Wong

What do you plan to perform?

We will be performing a piece for voice, synth, and vocoder.
(Original) We may cover our own arrangement of an existing recording.
(New) Professor Dannenberg suggested that we not do a cover of an existing song, so we decided to do an improvisational piece.

What is the challenge or theme that place your project above and beyond what you did for Projects 3 or 4?

We will be implementing a complex signal processor along with a synthesizer

What roles will each person play in creating the performance? Roles include design of software components, sound design, and composition. Most group members will be working on specific software components.

Task	**Group Member**
Volume detector	Vivian
Gain module	Eric
Bandpass filter	David
TouchOSC interface for synth and vocoder	Noelle & Matt
Synthesizer	Matt & Noelle
Track arrangement	Vivian

For each role, (1) what artifacts will be created (e.g. design documents, API specification, code, test cases, user manual, music scores, hardware), and (2) what grading criteria should be applied (e.g. "I will create clock synchronization code and a test case that runs

on 2 to 5 computers," or "I will compose all the music, which will consist of a score and 5 parts for the 5 of us," or "I will implement all of the signal processing to be used and document the OSC interface to Pd."

Task	**Created artifacts**	**Grading Criteria**
Volume detector	Code	I will create a function that detects the volume of audio input samples. *Working:* The volume detector computes an average of the signal amplitude *Fully working:* The volume detector is an envelope detector for the signal
Gain module (Attenuator)	Code	I will create a function that transitions audio signals from the synthesizer to volumes determined by a corresponding volume detector. *Working:* Transitions buffers to new volume levels with some audible jumps in volume *Fully working:* Transitions buffers smoothly with no audible jumps in volume
Bandpass filter	Code	*Working:* Distorted/non-normalized bandpass filtering *Fully working:* frequency and Q parameters reflect the filter
TouchOSC interface for synth and vocoder	Code, user manual	(Original) I will program controls on a pre-loaded TouchOSC layout to alter the vocoder output. *Minimum:* 5 controls with documentation for use (New) I decided to use a TouchOSC interface for the synthesizer. However, I did not provide an interface for controlling the vocoder because we had a working polyphonic synth, which provided enough depth to the sound on its own.
Synthesizer	Code	*Minimum:* Saw tooth wave at a fixed

		frequency *Fully working:* Saw tooth wave oscillator (monophonic) that takes frequency from pitch and note on/off messages *Extra credit:* polyphonic synth; takes frequency from pitch and note on/off messages from an external keyboard
Track arrangement	Music score, instructions for performers	(Original) I will choose a song and create an arrangement of it for our performance. *Fully working:* 2 voices (New) Altered to an improv performance. However, an original score was still created.