Chapter 22. Meeting 22, Approaches: Agents and Ecological Models

22.1. Announcements

- Sonic system reports due and presentations begin: 11 May
- Last quiz: Tuesday, 4 May

22.2. Workshop: Sonic System Project Drafts

- `Last two students present their draft projects`

22.3. Agents

- Software models of autonomous sub-systems
- Complexity and emergent behavior through the interaction of simple agents

22.4. Interactive Music Systems

- Computers that musically respond to MIDI messages (control data)
- Computers that musically respond to audio (sound through a microphone)
- Computers that accompany a human performance based on a shared score
- Computer (agents) that musically respond to each other (via audio or MIDI)

22.5. Analysis and Generation

- Interactive systems must have two basic components
- Components that “listen” to control data or audio information, and decode into musical models
- Components that generate musical responses based on analysis

22.6. Interactivity: Theatre

- Musical performance is theatre

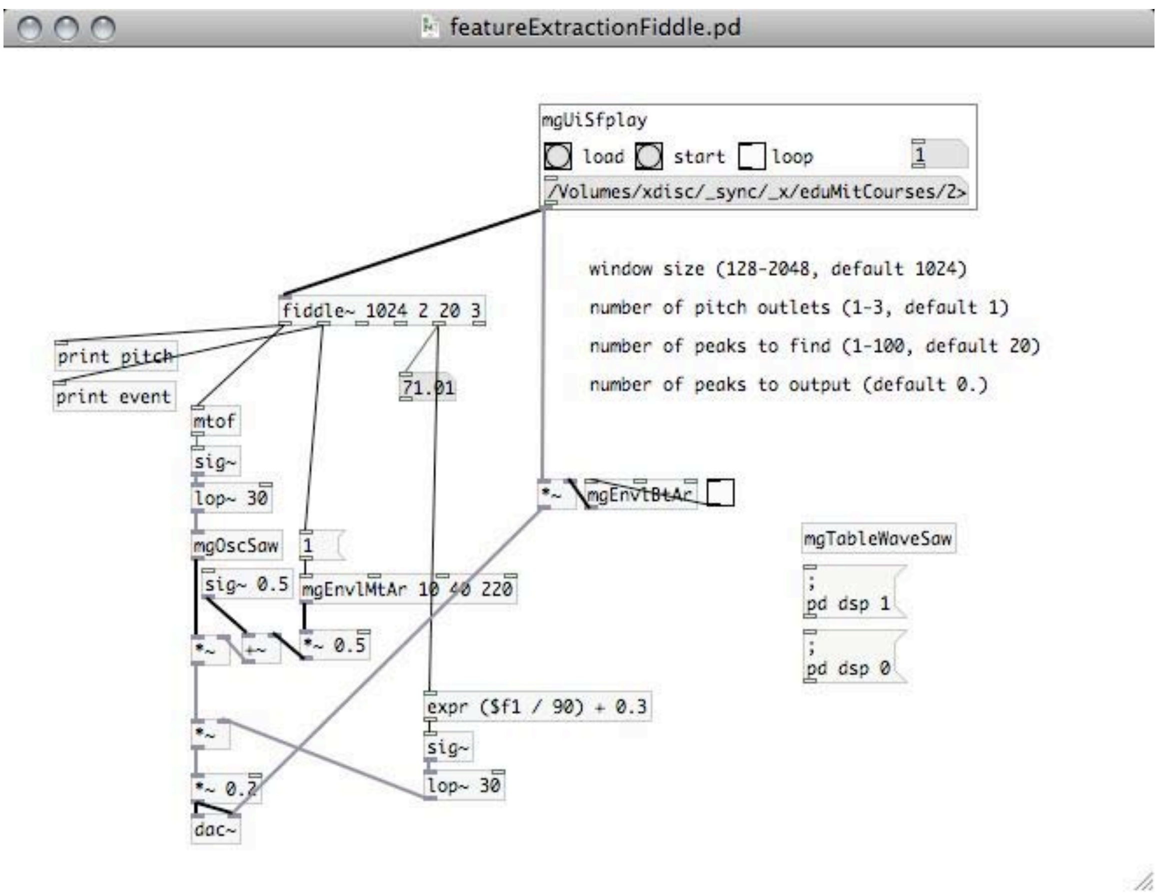
- Appeal of technological achievement or drama of technological disaster

22.7. Applications of Multi-Parameter Feature Extraction

- Detect articulation, pitch, and tempo and match to a score: score following
- Detect articulation, pitch, and rhythms, and build musical responses: interactive systems, installations

22.8. Multi-Parameter Feature Analysis in PD

- [fiddle~] object: pitch, event, and amplitude



22.9. Early Historical Examples of Interactive Music Systems

- 1967: Gordon Mumma's *Hornpipe* (1967): "an interactive live electronic work for solo hornists, cybersonic console, and a performance space"; system analyzes sound form horn and in performance space

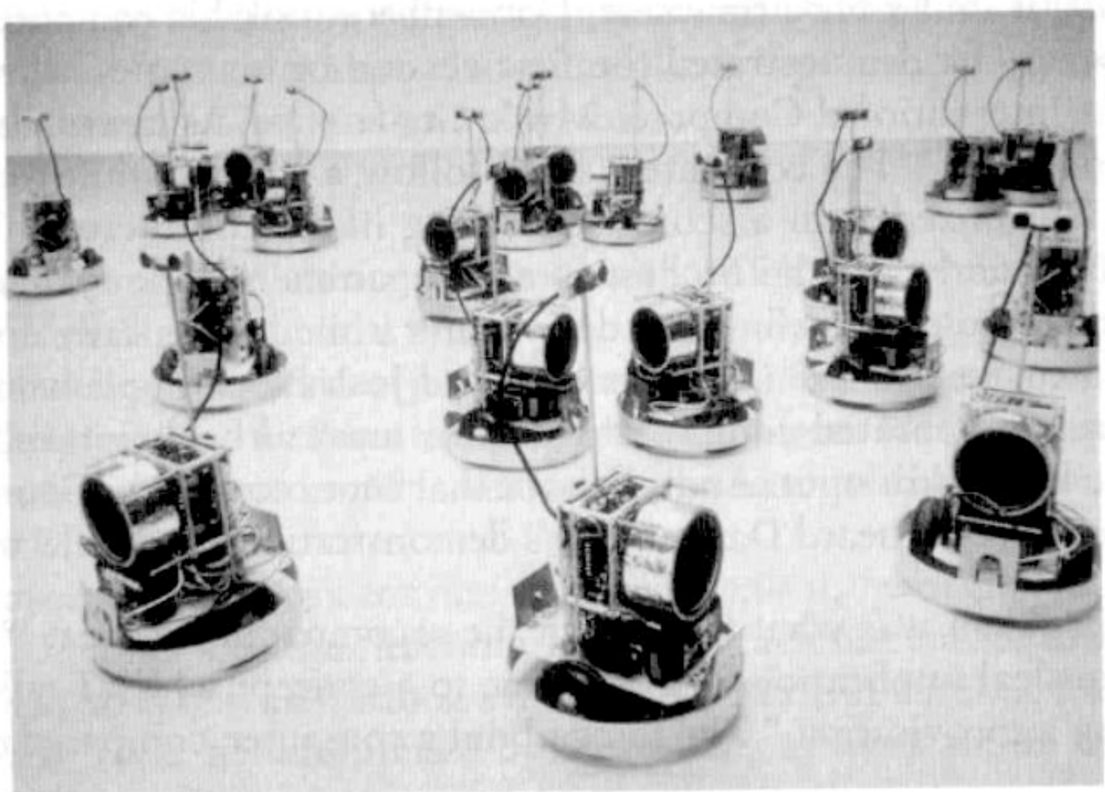
Audio: local (file://localhost/Volumes/xdisc/_sync/_x/eduMitCourses/21m380b/audio/mummaHornpipe.mp3)

- 1968: Max Mathews and F. Richard Moore develop GROOVE system at Bell Labs. Real-time performance interface to a predetermined musical score

- 1979: George Lewis, with a KIM-1 computer, develops interactive compositions designed to work with improvisation

- 1983: Felix Hess creates 40 *Electronic Sound Creatures*, small mobile machines with microphones and speakers that respond to each other an the envrionment

Felix Hess' Moving Sound Creatures in the late 1980s. Photo by John Stoel. Courtesy Felix Hess.



- 1987: Robert Rowe develops a system called Cypher, consisting of a Listener, a Player, and a Critic, used in *Flood Gate* (1989)

22.10. Reading: Rowe: Machine Listening and Composing with Cypher

- Rowe, R. 1992. “Machine Listening and Composing with Cypher.” *Computer Music Journal* 16(1): 43-63.
- What types of features are extracted during the first level of listener analysis?
- What types of features are extracted during the second level of analysis?
- How does the chord and key analysis routines work?
- What are the three compositional methods employed?
- What is the role of the critic?
- How is the large-scale behavior of the system varied over time?

22.11. Listening: Rowe

- Listening: Robert Rowe, *Shells*, 1993

22.12. Listening: Ariza

- Listening: Christopher Ariza, *to leave the best untold*, 2009

22.13. Alternative Agent Models

- Analogies to human roles
- Analogies to ecological models
- Analogies to social systems
- Analogies to physical systems

22.14. A Model of Particle Feedback Systems

- Particles in a dynamic system

- Particles
    - Have one or more states, each state with a discrete life span
    - Particle expired at termination of life span
    - Life cycle:

      ```
      [('a', 1), ('b', 2)]
      ```
- Particle Transformers
    - Have one or more states, each state with a discrete life span
    - Particle expired at termination of life span
    - State determines focus of particle
    - Focus is target state looked for in other particles; transformed with transformation map
    - Transform map:

      ```
      {'a':[(None, 3), ('a', 1)]}
      ```
    - Related to first order Markov chain
- Sensor Producers
    - Produces one type of Particle
    - Produces one type of Particle Transformer
    - Stores a threshold, a target value for a given state
    - Senses the composition of a collection of Particles
    - Stores a production count range: given difference from threshold, give a range of Particles to produce (when below threshold) or Particle Transformers to produce (when above threshold).
    - Production count range:

      ```
      {(-30,-10): [1,2], (1,10): [1, 2], (11, 20): [1, 4], None: [1, 8]}
      ```
- Environment
    - Store lists of Sensor Producers, Particles, and Particle Transformers
    - Provides model of Sensor Producer (one for now)
    - Provides an absolute discrete value range for sensed particle

- Specify number of sensors
- Can age all Particles by one or more age steps

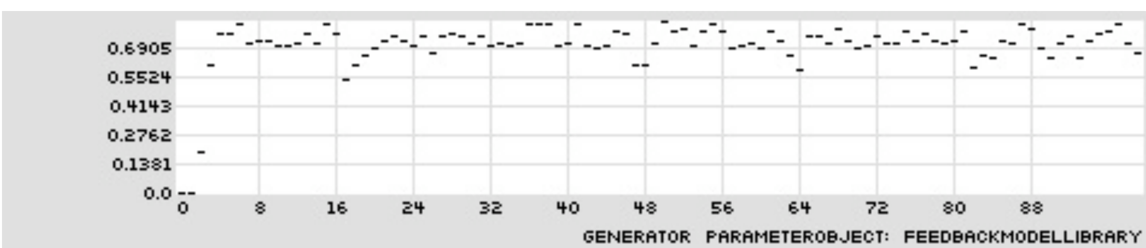
22.15. Feedback System as ParameterObject

- The feedbackModelLibrary ParameterObject

```
:: tpv fml
Generator ParameterObject
{name,documentation}
FeedbackModelLibrary feedbackModelLibrary, feedbackModelName, parameterObject,
                     parameterObject, min, max
                     Description: Produces values from a one-dimensional string
                     rewrite rule, or Lindenmayer-system generative grammar. The
                     terminus, or final result of the number of generations of
                     values specifed by the stepCount parameter, is used to
                     produce a list of defined values. Values are chosen from
                     this list using the selector specified by the
                     selectionString argument. Arguments: (1) name, (2)
                     feedbackModelName, (3) parameterObject {aging step}, (4)
                     parameterObject {threshold}, (5) min, (6) max
```

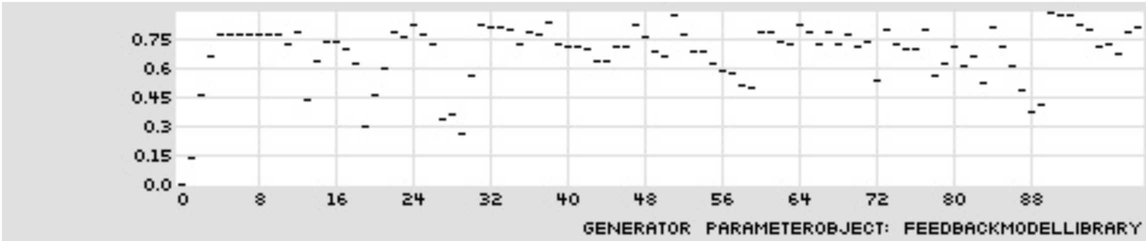
- A basic model of a Thermostat: particles as heat

```
:: tpmap 100 fml,t,(bg,rc,(1,1.5,2))
feedbackModelLibrary, thermostat, (basketGen, randomChoice, (1,1.5,2)),
(constant, 0.9), (constant, 0), (constant, 1)
TPmap display complete.
```



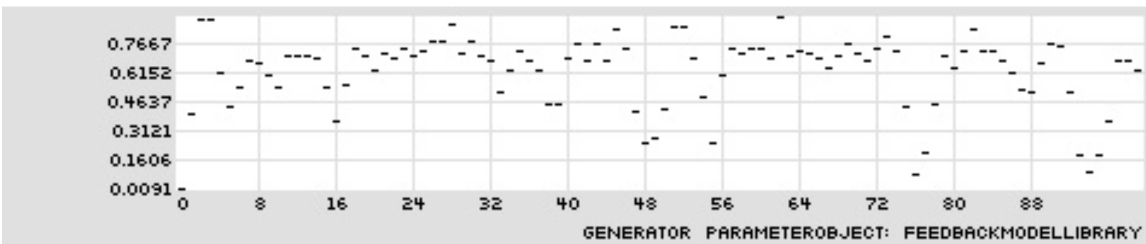
- Dynamic age values applied to Particles

```
:: tpmap 100 fml,t,(ls,e,(c,30),0,4)
feedbackModelLibrary, thermostat, (lineSegment, (constant, 30), (constant, 0),
(constant, 4)), (constant, 0.9), (constant, 0), (constant, 1)
TPmap display complete.
```



- Climate control: produce both Particles and Particle Transformers

```
:: tpmap 100 fml,cc,(bg,rc,(.5,1,1.5))
feedbackModelLibrary, climateControl, (basketGen, randomChoice, (0.5,1,1.5)),
(constant, 0.9), (constant, 0), (constant, 1)
TPmap display complete.
```



- Alternative approaches to PO interface?

22.16. Feedback System as Dynamic Contour

- Can treat the grammar alphabet as parameter values: integers, floating point values
- Command sequence:
  - emo mp
  - tmo lg
  - tin a 66
  - *constant pulse*

    tie r pt,(c,8),(c,1),(c,1)
  - *amplitude controlled by Thermostat feedback*

    tie a fml,t,(bg,rc,(1,1.5,2))
  - *using convert second to set durations*

tie r cs,(fml,t,(c,1),(c,.7),.001,.400)

- *amplitude controlled by Climate Control feedback*

  tie a fml,cc,(bg,rc,(.5,1,1.5)),(c,.7),0,1

- eln; elh

22.17. Feedback System as Path Index Values

- Feedback system states as index values from the Path
- Command sequence:
  - emo m
  - *create a single, large Multiset using a sieve*

    pin a 5@1|7@4,c2,c7

  - tmo ha
  - tin a 107
  - *constant rhythm*

    tie r pt,(c,4),(c,1),(c,1)

  - *select only Multiset 0*

    tie d0 c,0

  - *create only 1 simultaneity from each multiset; create only 1-element simultaneities*

    tie d2 c,1; tie d3 c,1

  - *select pitches from Multiset using Thermostat*

    tie d1 fml,t,(bg,rc,(1,1.5,2)),(c,.7),0,18

  - *select pitches from Multiset using Climate Control*

    tie d1 fml,cc,(bg,rc,(.5,1,1.5)),(c,.7),0,18

  - eln; elh