

Real-time Programming & Processing of Music Signals

Habilitation à diriger la recherche
Université Pierre et Marie Curie (UPMC)

Arshia Cont

Équipe-projet MuTant

Plan

(I) Practical & Scientific Context

(II) Real-time Machine Listening

- Score Following
- ~~Multiple-pitch Recognition~~
- Music Information Geometry

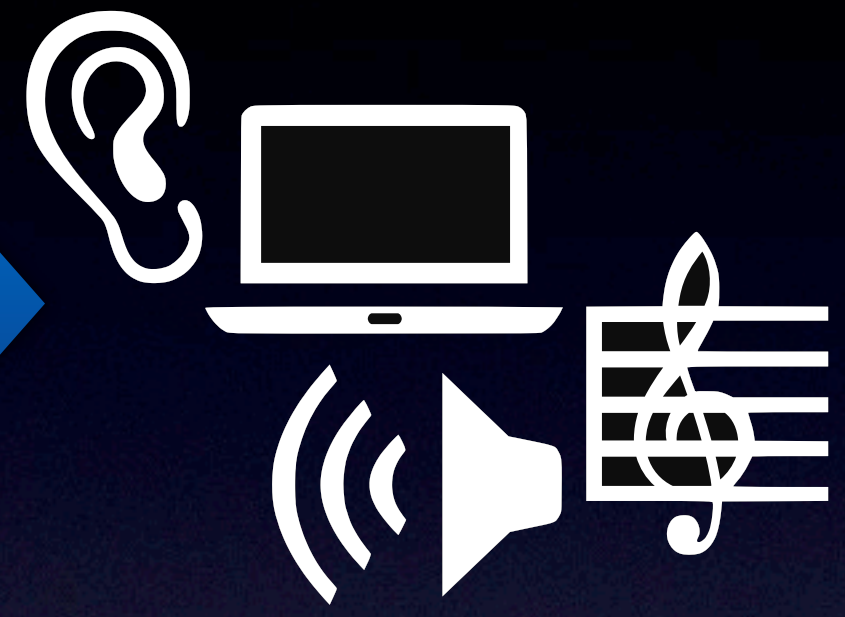
(III) Reactive Synchronous Programming

- Antescofo language

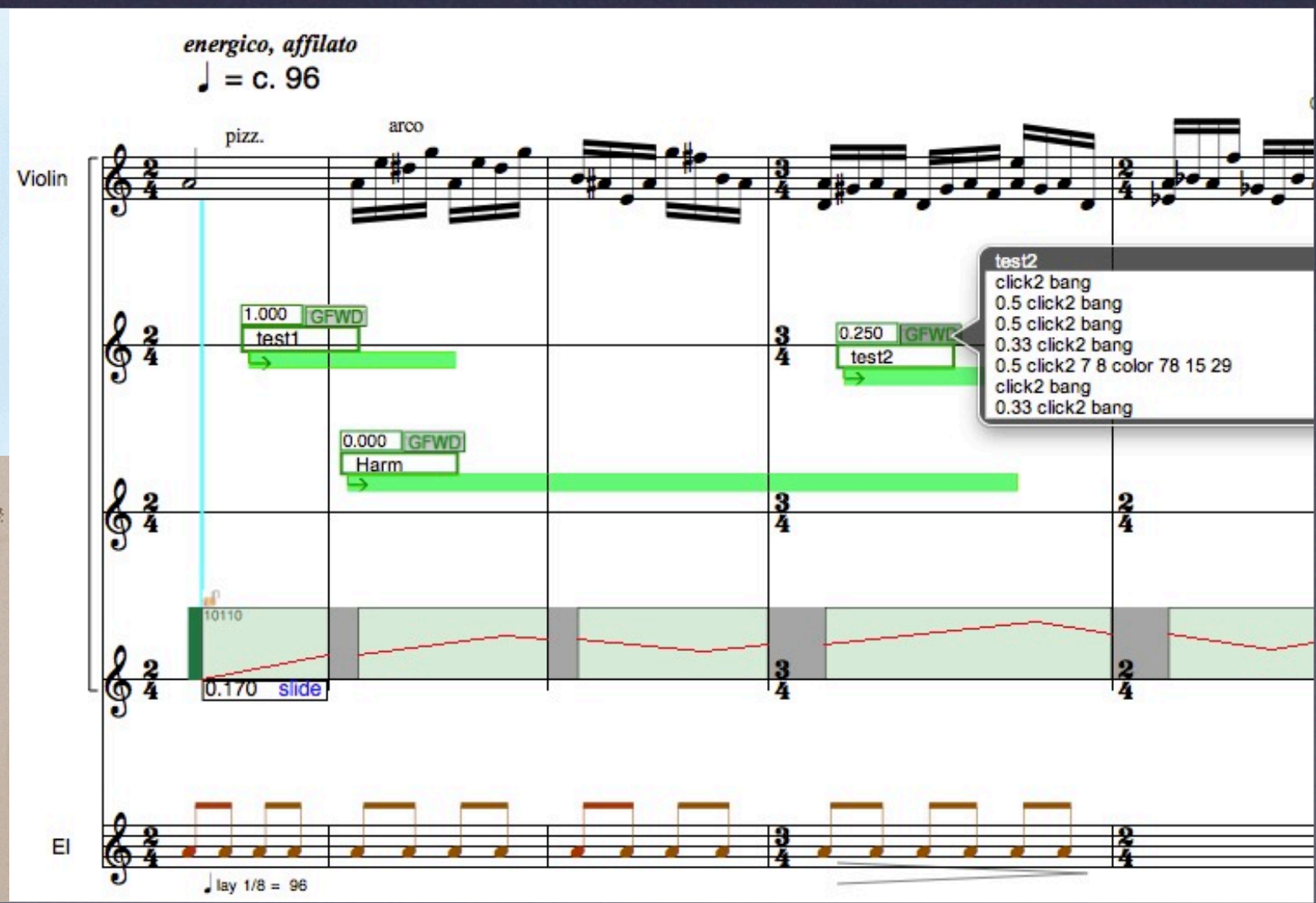
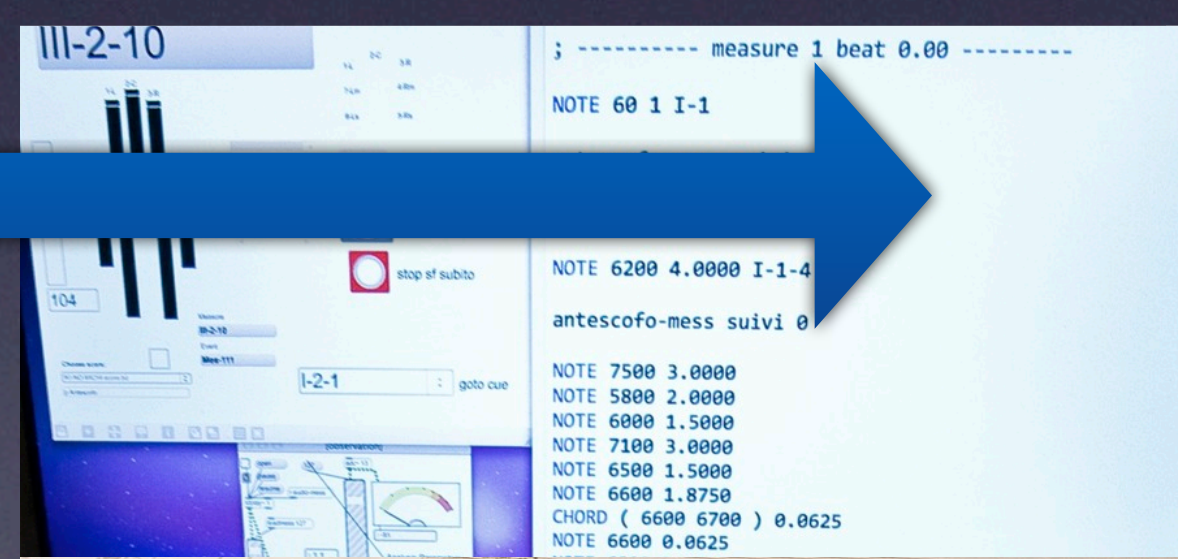
(IV) Towards Cyber-Physical Music Systems

Practical Context

Performance / Realtime

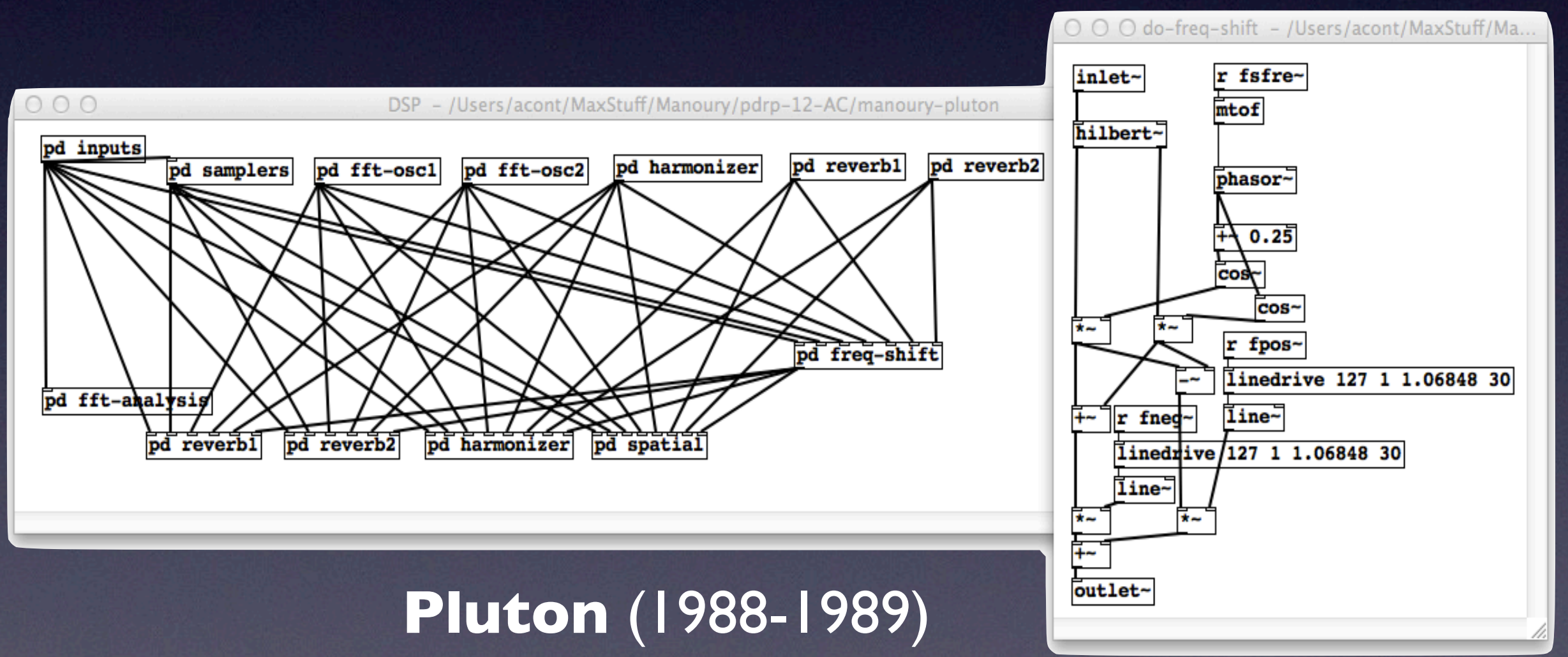
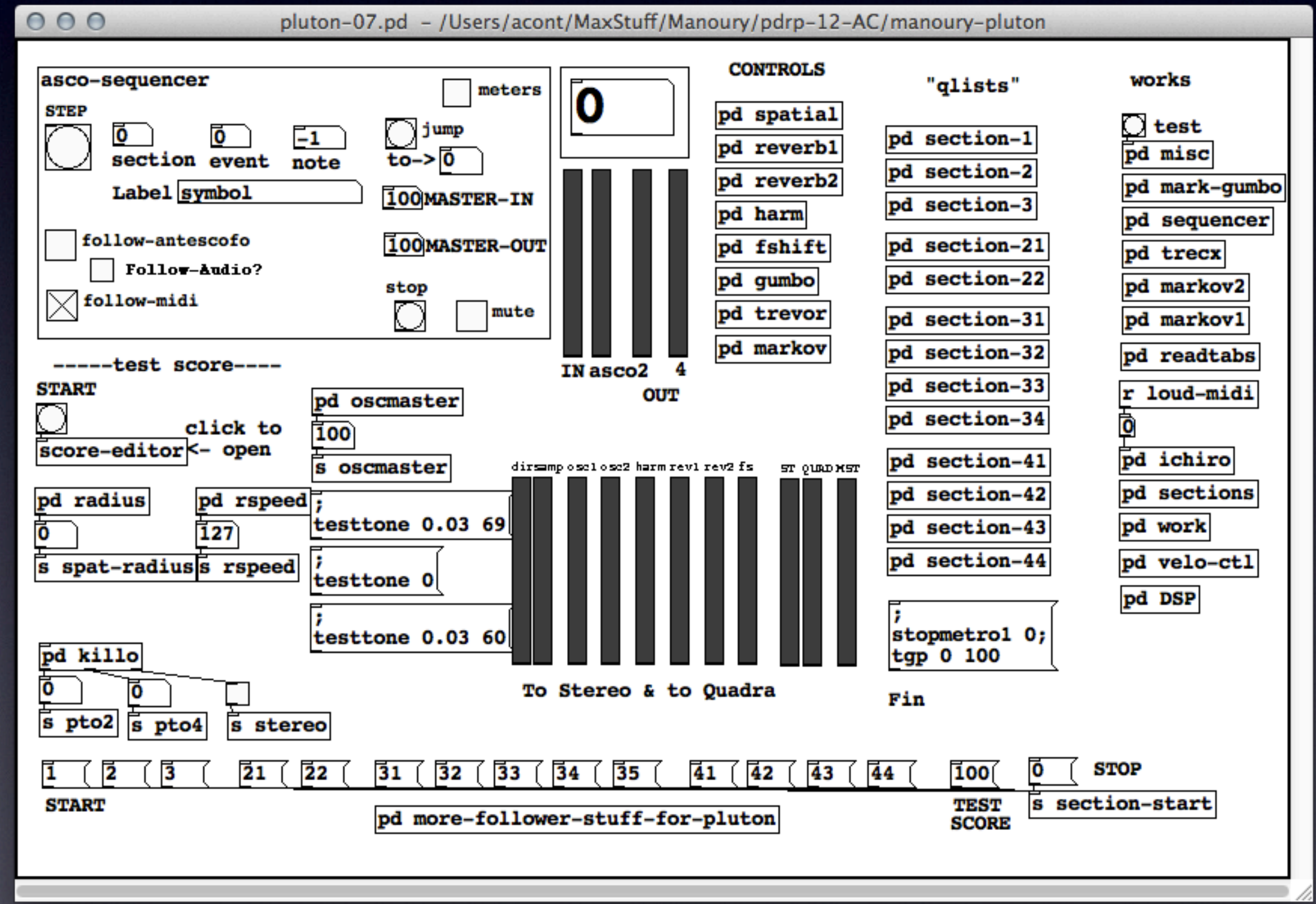


Composition / Authoring



Practical Context

- Composition / Authoring for mixed music
 - Programming style in the 1980s-90s



Pluton (1988-1989)
for piano & live electronics
Manoury, Lippe, Puckette

Practical Context

- Composition / Authoring for mixed music
- Programming style in the 1980s-90s

The image displays several Pure Data patches and control panels for the 'Pluton' software. The patches are arranged in a collage, showing various signal processing blocks and control elements.

- Top Left:** A patch titled 'sec 2' showing a network of 'pd' objects connected to an 'r cue02' object. The patch is organized into columns of objects, with some objects numbered 1 through 30.
- Top Middle:** A control panel titled 'pluton-07.pd' with various knobs, buttons, and meters. It includes sections for 'quencer', 'CONTROLS', and 'works'. The 'works' section lists various patch objects like 'pd spatial', 'pd reverb1', 'pd reverb2', etc.
- Top Right:** A patch titled 'do-freq-shift' showing a signal chain starting with an 'inlet~', followed by 'hilbert~', 'r fsfre~', 'mtof', 'phasor~', and several 'cos~' objects, leading to 'linedrive' objects and an 'outlet~'.
- Bottom Left:** A large patch titled '(subpatch) - /Users/acont/MaxStuff/Manoury/pdrp-12-AC/manoury-pluton'. It features an 'inlet' connected to a 'sel' object, which triggers a 'reset' and a 'del 5000' object. The patch includes several signal processing blocks like 'ptor', 'rgate', 'rto4', 'fto4', 'spaton', 'clrosc', 'mark', 'rtof', 'fsfre', 'Revfb', 'Rto4', 'hto4', 'tto4', 'stopdel', 'stopmetro2', and 'stopl.1.25 bang;'. It also includes a 'gumbo3' and 'gumbo2' object with specific frequency and skew settings.
- Bottom Middle:** A control panel titled 'DSP' with various meters and buttons. It includes sections for 'CONTROLS', 'works', and 'DSP'. The 'works' section lists various patch objects like 'pd spatial', 'pd reverb1', 'pd reverb2', etc.

Pluton (1988-1989)
for piano & live electronics
Manoury, Lippe, Puckette

Practical Context

- Composition / Authoring for mixed music
 - Programming style today

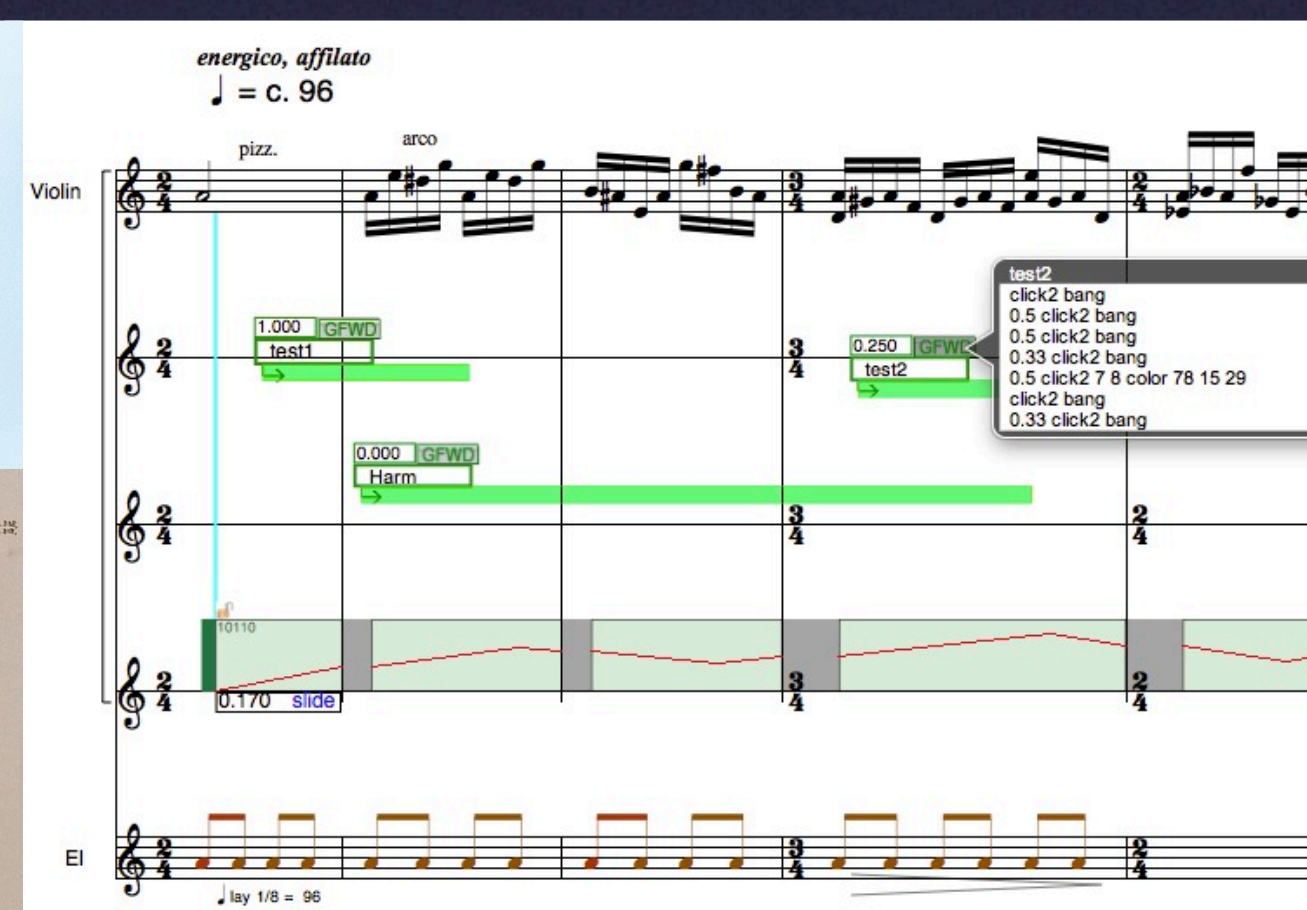
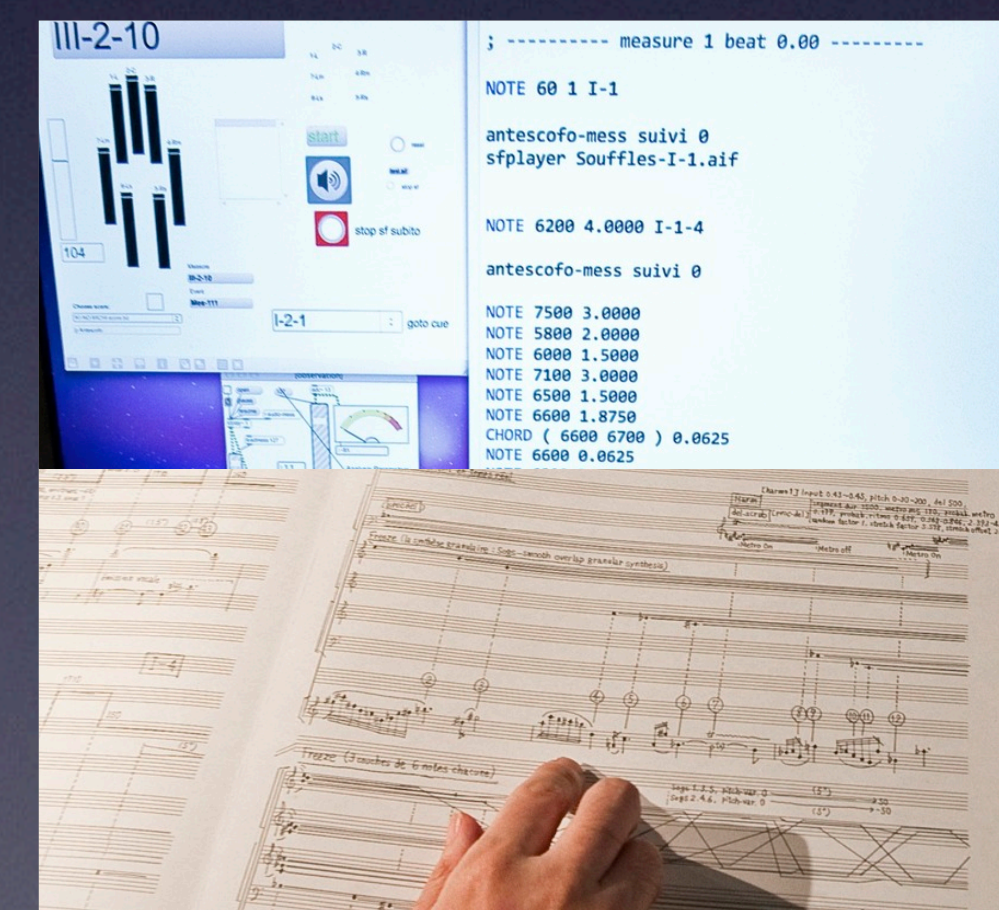
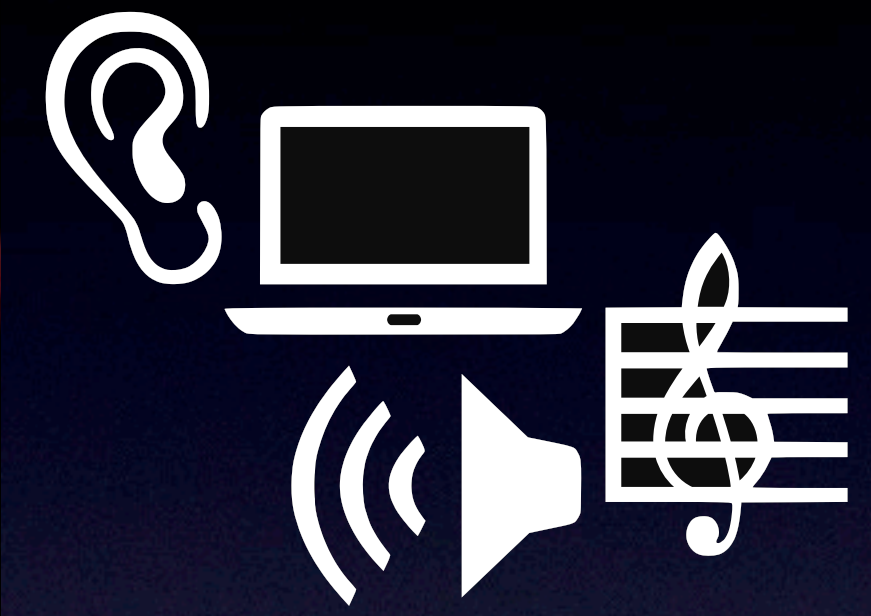
The screenshot shows a Pure Data patch for 'Antescofo'. On the left is a 'matrix' window with a grid of points for various channels: HP1-4, SPAT1-3, SD, HARM1-4, BUF1-4, FS, DEL, CYCPN, SD2, and BUF3-4. Below the matrix is a 'faders (presentation)' window with vertical sliders for spat1-3, Cycpan, del, sd1-2, harm1-4, and buf1. The main patch area includes a 'HistWist_DSP1_04' window with a 'Loadbang' and 'system un' objects, and an 'Antescofo' window with a 'Movement1' dropdown, 'start' and 'stop' buttons, and various signal processing objects like 'p observation', 'p menu.ctrl', and 'p mackeys'. A 'Tempogram' window is also visible.

The screenshot shows a Pure Data patch for 'Hist whist'. It features a complex signal flow with multiple DSP objects including 'poly~ spectraldelay_04.monopoly~.maxpat 1 args sd1 8192', 'poly~ freqshifter_01.monopoly~ 1 args fs', 'poly~ svptrans.polyphony3~ 4 args harm 4096 4', and 'poly~ hw_tapdelays_02 1 args del'. There are also several 'receive~' and 'ro.bcf.fader' objects for control. The patch is organized into several sub-windows: '[sound-processes]', '[dsp]', and '[out_processes]'. A 'matrix' window is also visible on the right side.

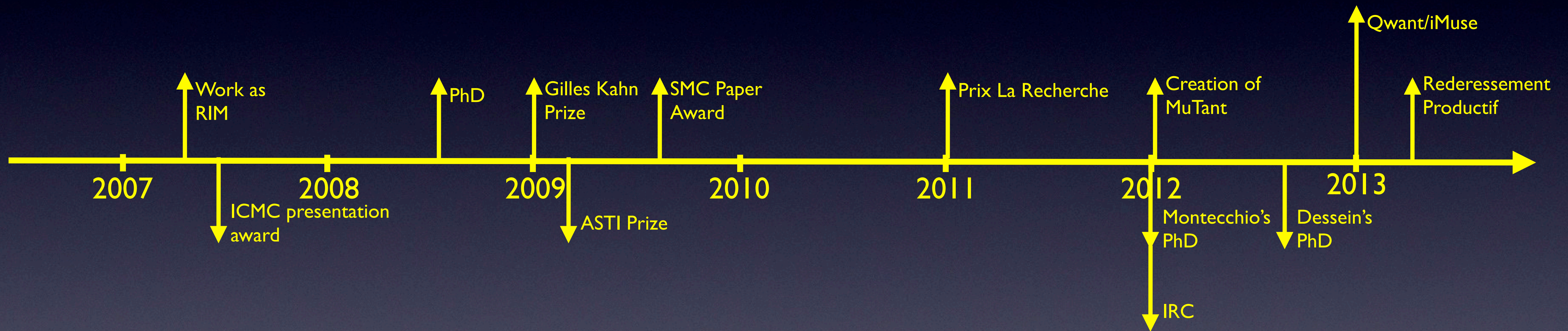
Hist whist (2009)
for violin & chamber electronics
Stroppa, Cont

Scientific Challenges

- Real-time Coordination and Synchrony between Human & Computer Mediums
- Real-time Computer Audition
- Authoring heterogeneous mediums, computations and times
- Critical Safetiness: Assuring what is written is what is performed!
 - Correctness in Timing
 - Correctness in Computing



Social Timeline



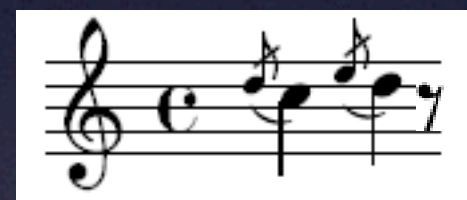


Real-time Machine Listening

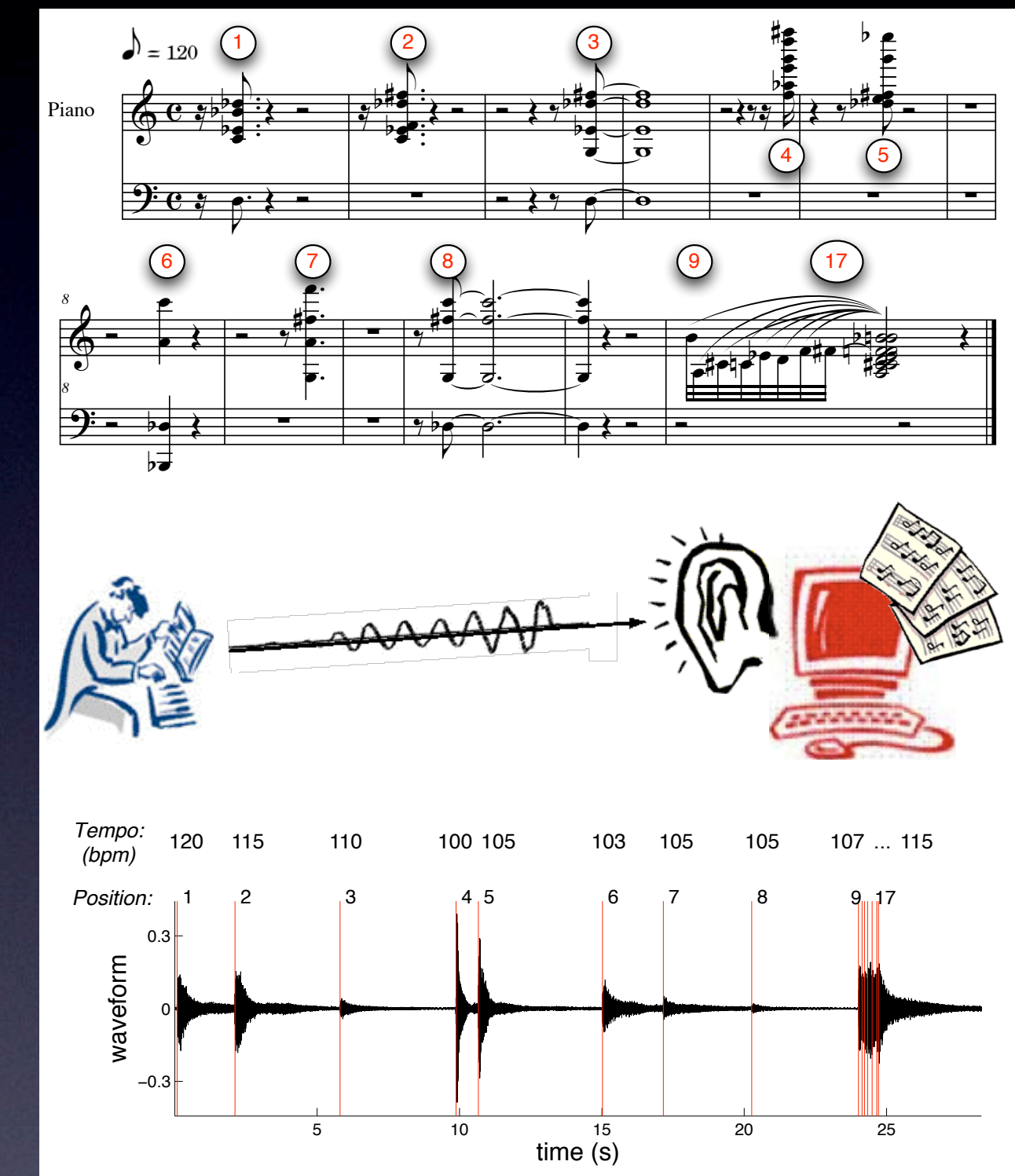
Real-time Machine Listening

- **Score Following**

- Real-time alignment of audio to music score + Extracting interpretation parameters
- Dominating approach:
 - Represent the score & latent variables using generative models



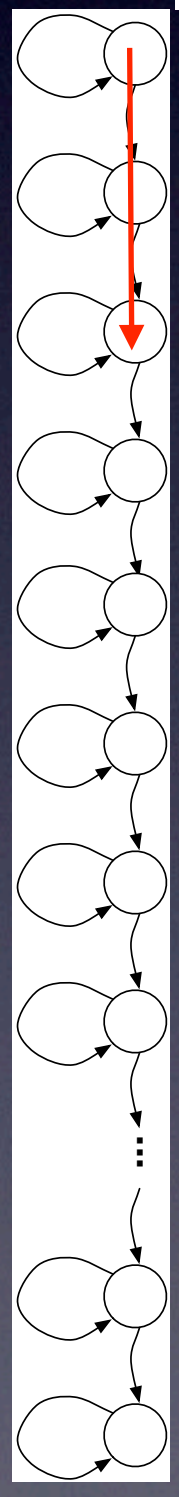
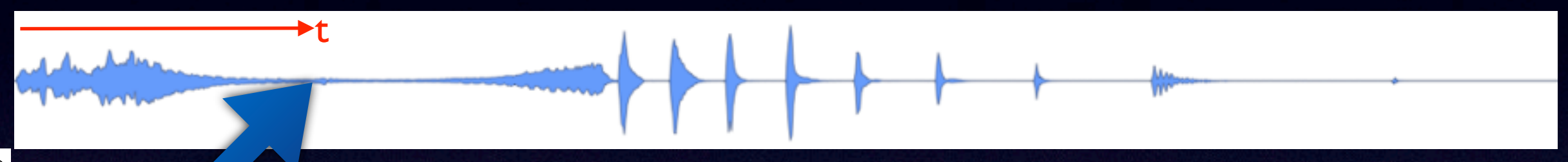
- Define an inference framework for the inverse problem
 - DTW, HMM, DBN, RBF, etc.



Real-time Machine Listening

- **Score Following**

- Inference formulation simplified:



- For each (real)time audio at time t , X_t , find the corresponding state S_j , representing the best path from 0 to t ,

Or
$$\max_{s_0, \dots, s_{t-1}} P(S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t)$$

- This is heavy to compute in real-time. We make assumptions to make it tractable.
 - The Markovian assumption: $\tilde{\alpha}_j(t) = b_j(x_t) \max_i (p_{ij} \tilde{\alpha}_i(t-1))$
 - Requires that the stochastic process be **memoryless**
 - Ideally should be complemented with future beliefs

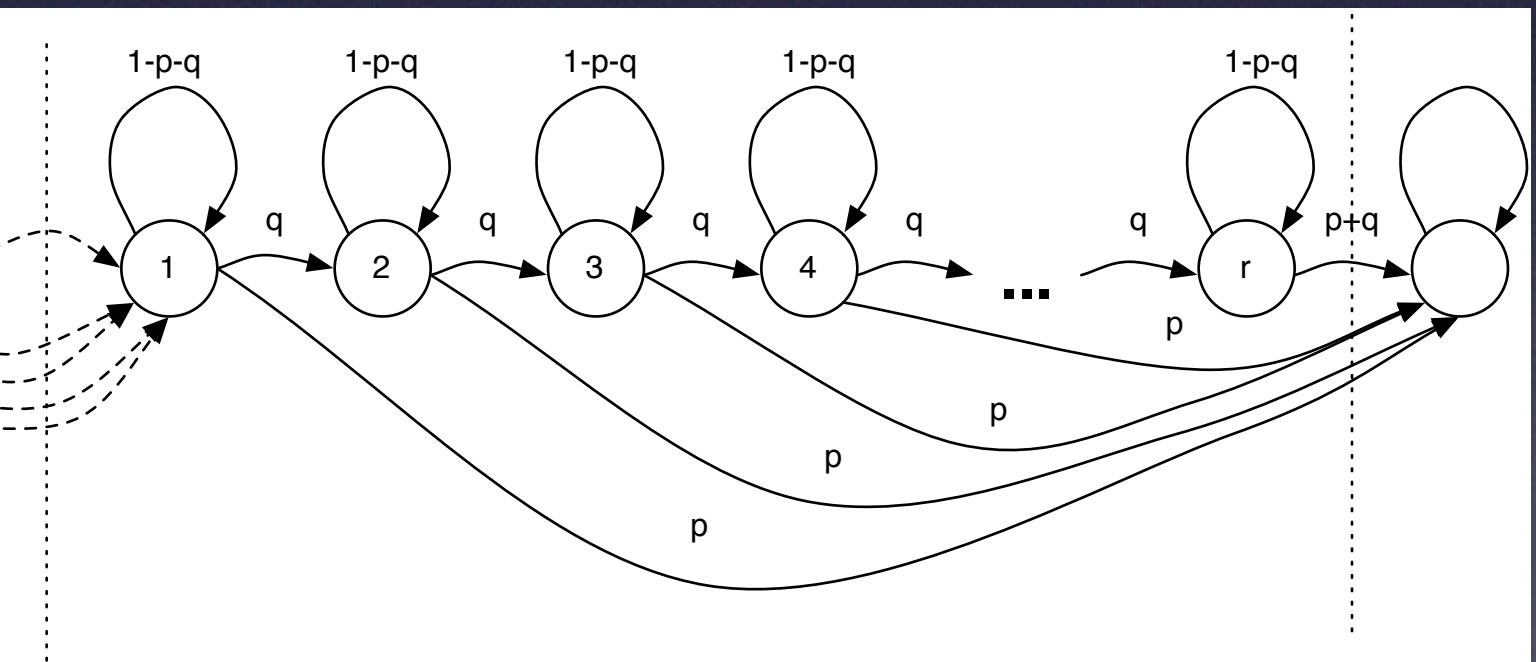
Problem #1:
Music performance is **NOT** a memoryless process

Problem #2:
Inference in music performance is **NOT** causal but anticipatory

Real-time Machine Listening

- **Score Following**

- Do not ignore “time”!
- Markovian Occupancy:
 - The probability of staying d discrete time in a Markovian state is **implicit** and exponential.
 - In speech (and music) an event with expected duration is often modeled as follows:



$$P(U = u) = \sum_{n=1}^{r-1} \binom{u-1}{n-1} (1-p-q)^{u-n} q^{n-1} p + \binom{u-1}{r-1} (1-p-q)^{u-r} q^{r-1} (p+q)$$

- The number of states and parameters are **static** and often learned from performance.

Problem #1:
Music performance is NOT a memoryless process

Problem #2:
Inference in music performance is NOT causal but anticipatory

Problem #3:
Music performance is NOT a static but dynamic process

Real-time Machine Listening

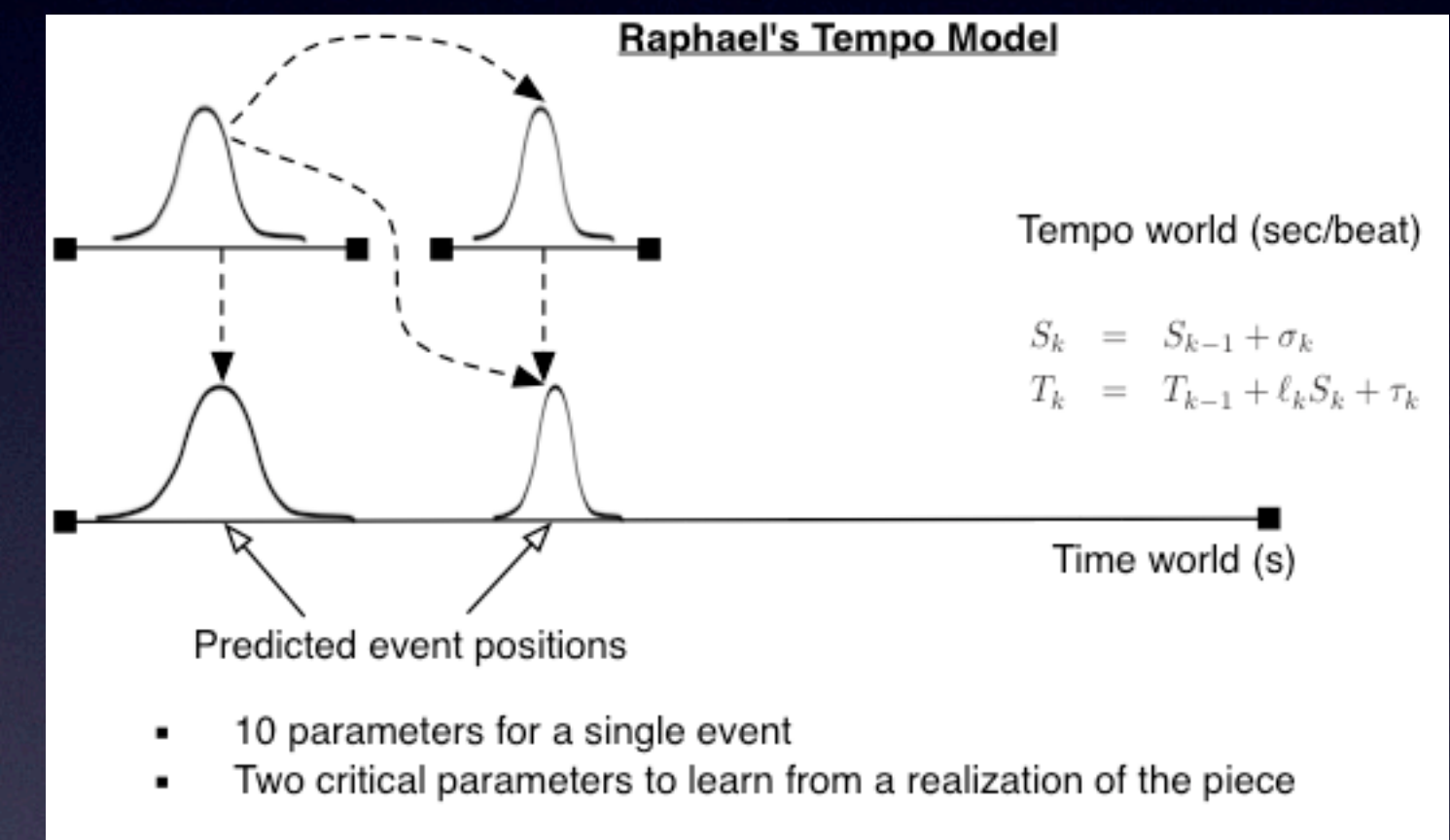
- **Score Following**

- 2006:

- The Suivi~ system
- ICASSP paper on polyphonic score following: NMF + Hierarchical HMM + Particle Filtering!

- 2007:

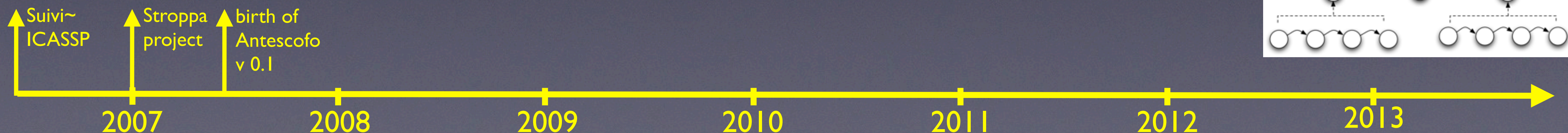
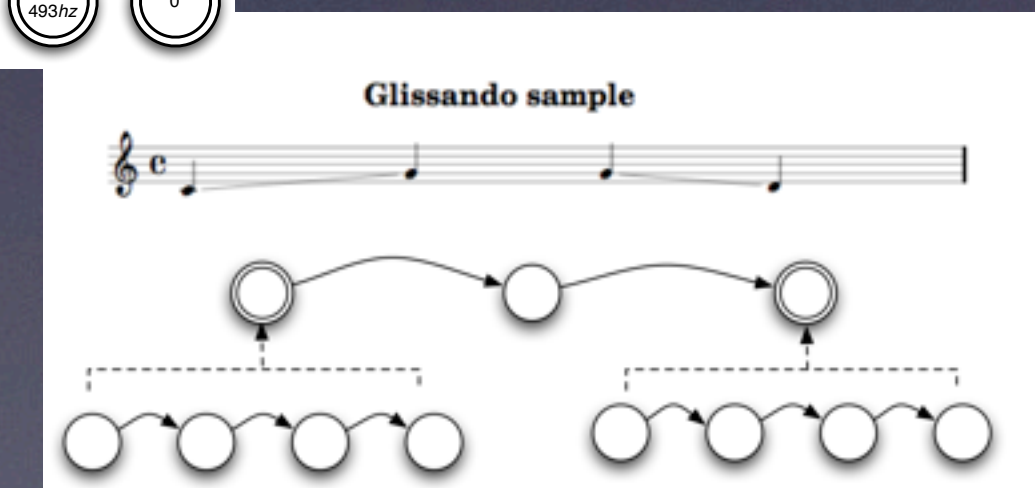
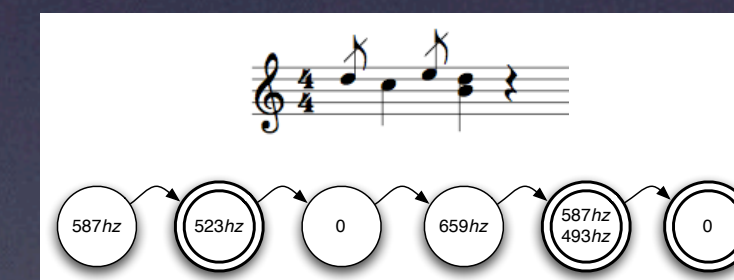
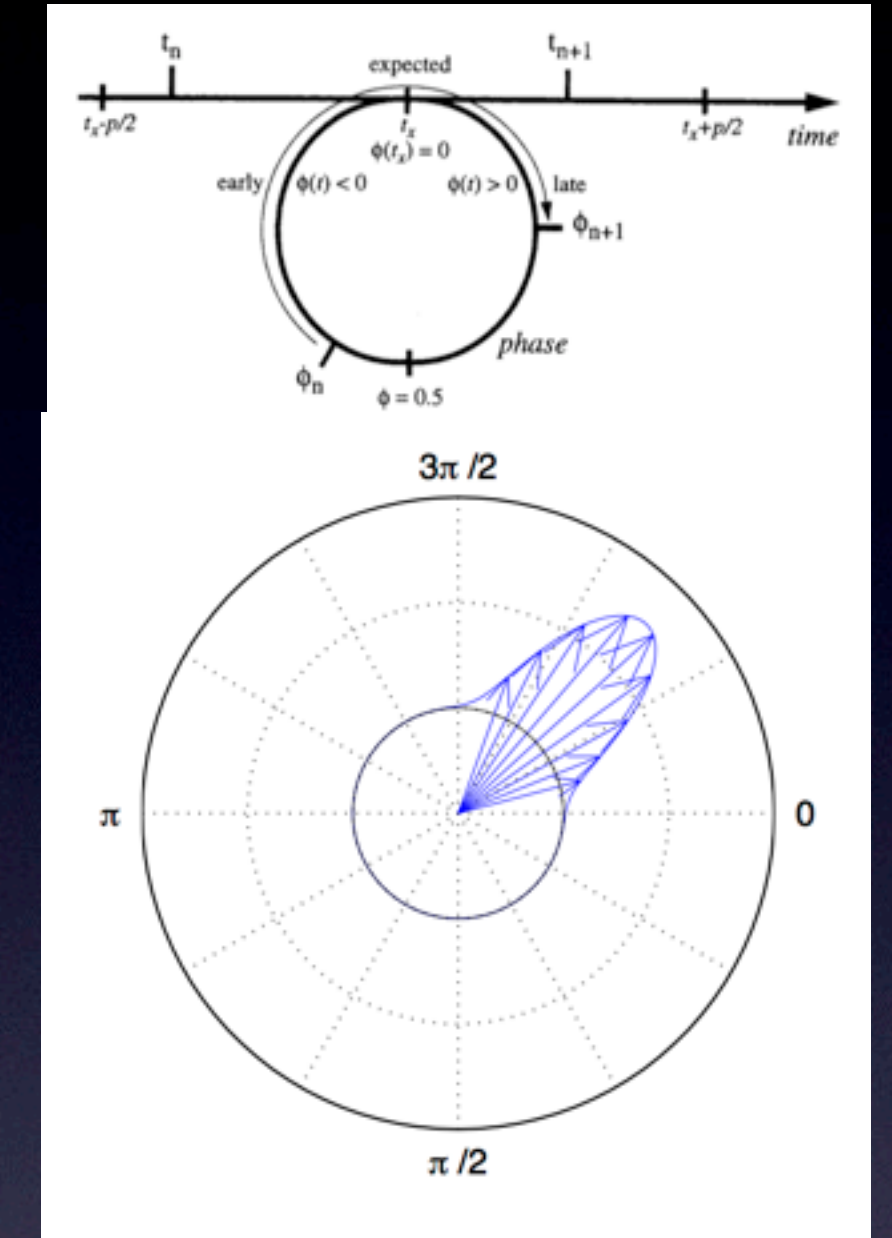
- Stroppa project + Puckette's Qualifying exam: Decoding position *and* tempo in real-time
- Christopher Raphael's Model
 - *Wise*: Inferring discrete position and continuous tempo
 - *Not wise*: Cascaded position and tempo inference, Static parameters, Linear model of time



Real-time Machine Listening

- **Score Following**

- Cognitive foundations of musical time
 - “The dynamics of attending: How people track time-varying events” by Large & Jones (1999)
 - Time is circular! (Directional Statistics)
 - Use entrainment (sympathy of clocks), extended Kalman Filter. Two **dynamic** parameters only.
- Abandon of Markovian principle
 - Use of Variable length memory Semi-Markov Chains
 - Hybrid Markov + Semi-Markov Inference, Heterogeneous times
- Coupled agents (as contrary to cascaded decoding)



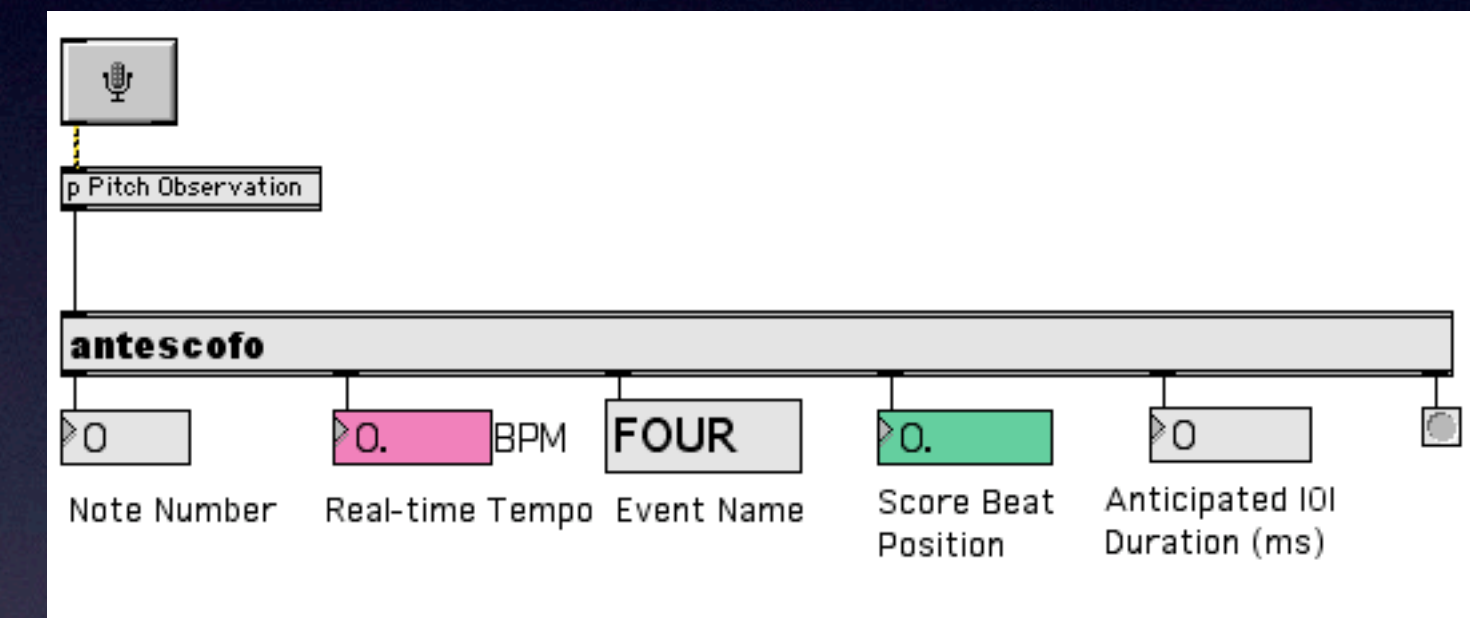
Real-time Machine Listening

- **Score Following**

- Antescofo version 0.1 (2007-08)

$$\alpha_j(t) = b_j(x_t) \times \max_{1 \leq u \leq t} \left[\left\{ \prod_{v=1}^{u-1} b_j(x_{t-v}) \right\} d_j(u) \max_{i \neq j} (p_{ij} \alpha_i(t-u)) \right] \quad \text{VS} \quad \tilde{\alpha}_j(t) = b_j(x_t) \max_i (p_{ij} \tilde{\alpha}_i(t-1))$$

- Explicit occupancy distribution: $d_j(u)$
- Explicit time-memory capacity per state: U
 - Both coupled with *tempo* and thus *dynamic* (learned online)
- Polyphonic?!
 - The concert experience with Los Angeles Phil. (01/2008)
 - Piece for Vibraphone, Composer in Research, Vassos Nicolaou with Serge Lemouton (2008-09)
 - Solution: Use faulty and cheap observations!



Real-time Machine Listening

- **Score Following**

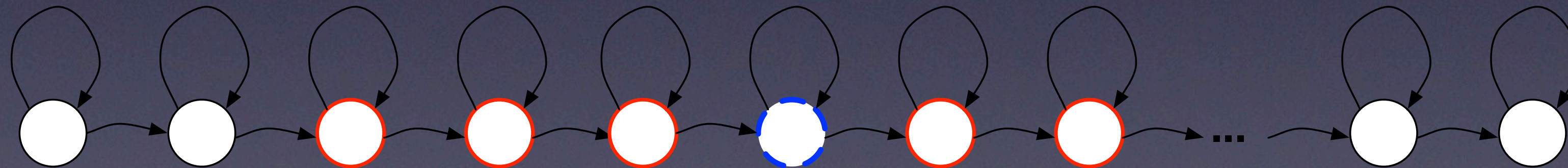
- By 2009-10: 30+ world-class performances, 12 creations at Ircam
- The *Pluton* disaster in 2011
 - Pedal, repeated notes, very low frequencies, high resonance
 - Complete rethink of the inference formulation

propagation in Antescofo 0.4

propagation in Antescofo 0.3

$$\alpha_j(t) = \frac{b_j(x_t)}{N_t} \times \left[\sum_{u=1}^t \left\{ \prod_{v=1}^{u-1} b_j(x_{t-v}) \right\} d_j(u) \sum_{i \neq j} (p_{ij} \alpha_i(t-u)) \right] \quad \text{VS} \quad \alpha_j(t) = b_j(x_t) \times \max_{1 \leq u \leq t} \left(\left\{ \prod_{v=1}^{u-1} b_j(x_{t-v}) \right\} d_j(u) \max_{i \neq j} (p_{ij} \alpha_i(t-u)) \right)$$

- + make it more aware of “now” and “future”



Real-time Machine Listening

- **Score Following**

- Perspectives:

- Multiple-target Following

- Polyphonic vs. Multiple-voice following
- Voice asynchrony
- Philippe Cuvillier's MS Thesis 2012

- The Tautology of Generative Modeling

- Mixture of discriminative and generative models?

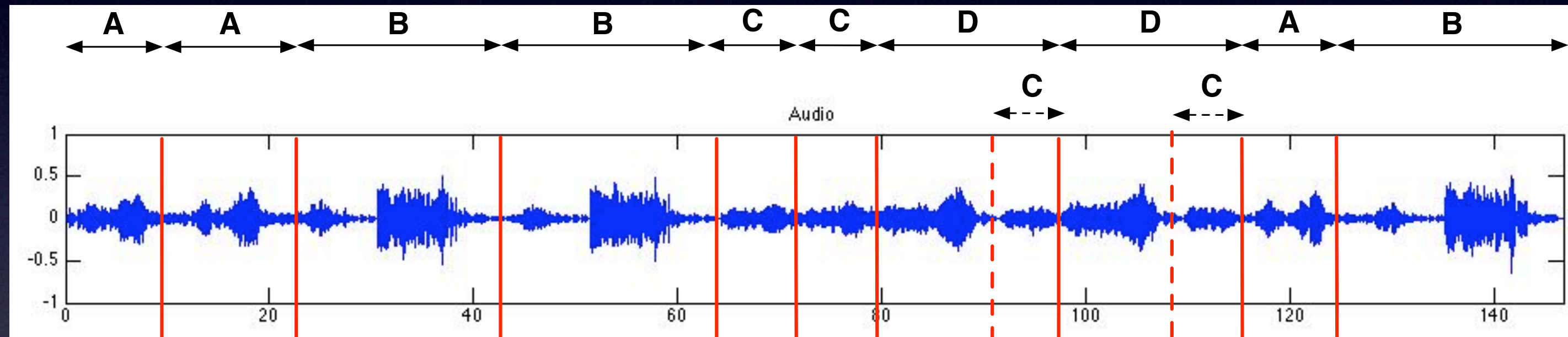
- Multi-modal tracking



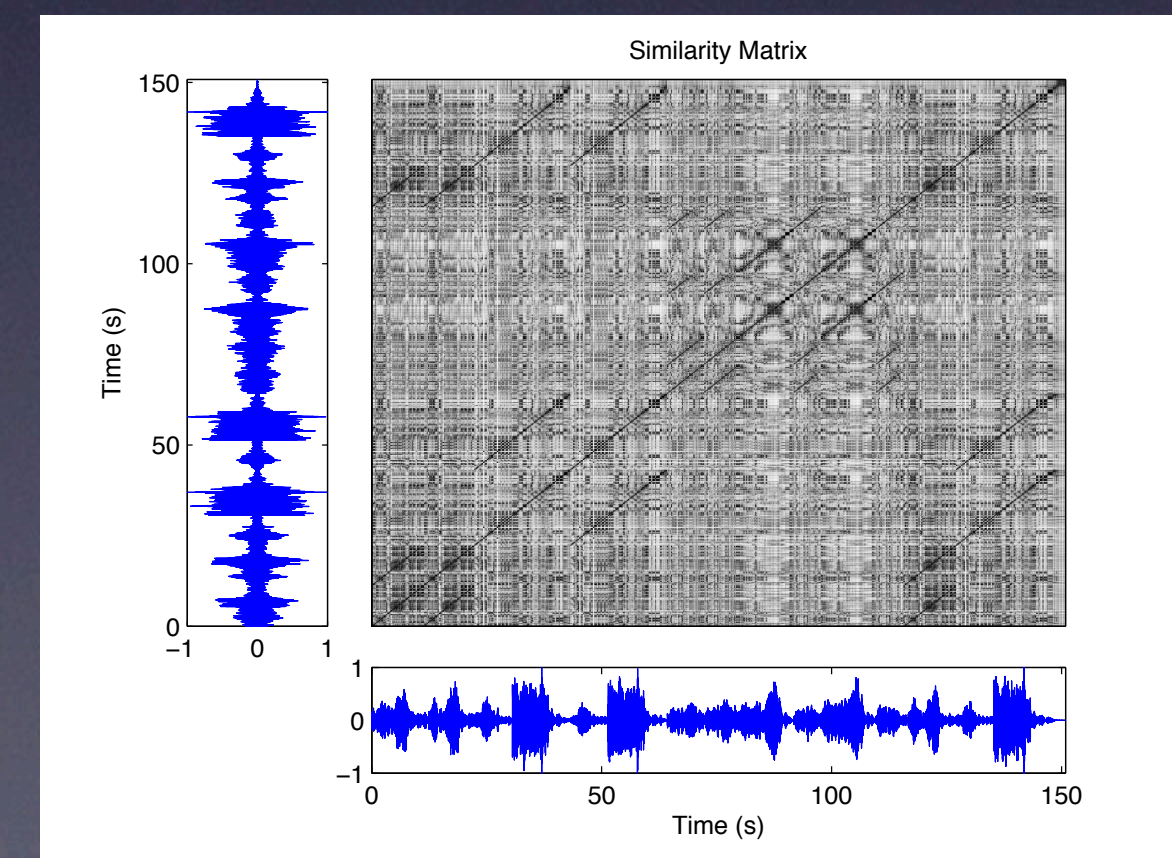
Real-time Machine Listening

- **Music Information Geometry**

- Motivation: “Effortless” ability among humans that pose challenging problems for machine intelligence.
Ex: Blind Structure Recovery



- Bridging the gap between “symbol” & “signal”
- Common approach: Self-similarity analysis (MIR)
- Non-causal, requires further processing, ignores “time”
- Machine learning with implicit geometrical processing



AudioOracle
ICMC Award



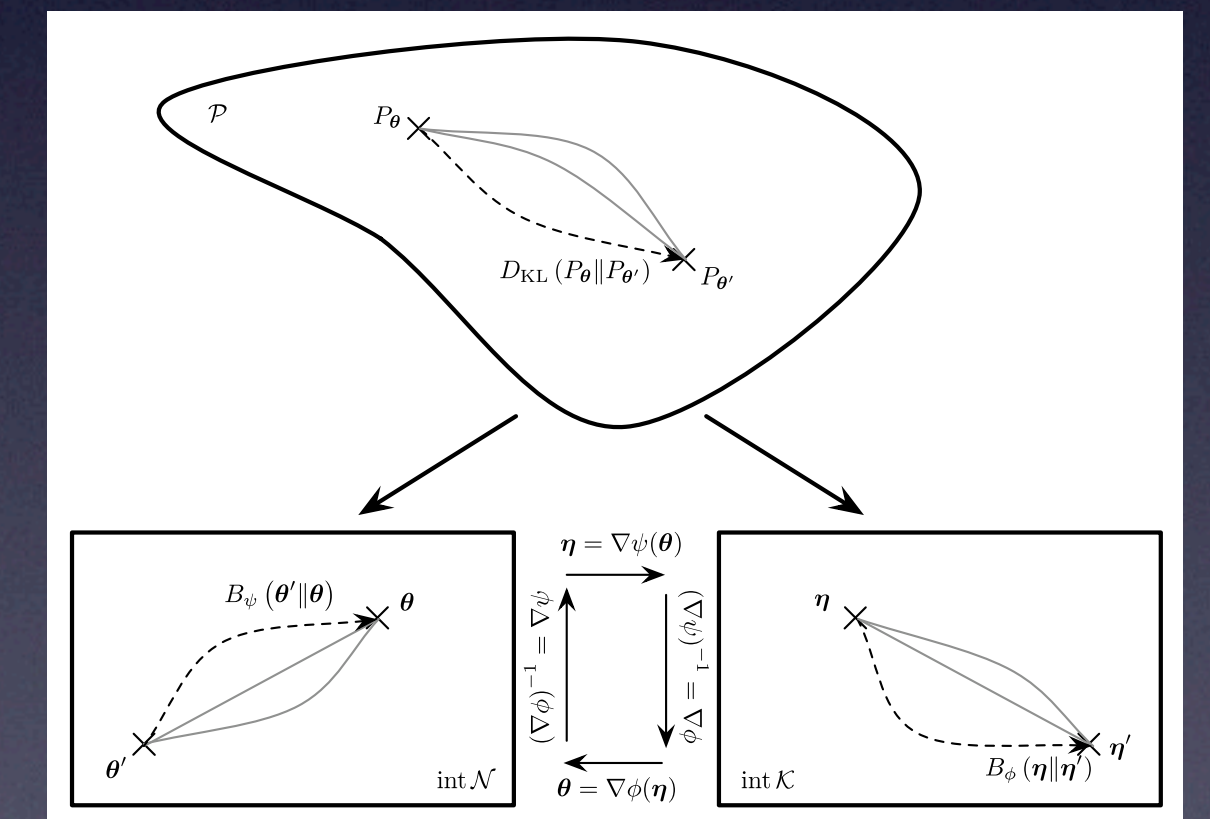
Real-time Machine Listening

- **Music Information Geometry**

- Basic idea: Attack the very fundamentals of “distance” and “information relevance”
- Geometry + Information Theory: Construct a dedicated geometric space!

- Common constructions:

- Bottom-up: Define points, connections, distances, transports towards metrics (Amari et al., Pennec et al.)
- Top-down: Derive the geometry from data represented on prior engineering assumptions (Nielsen & Nock, Banerjee)



Real-time Machine Listening

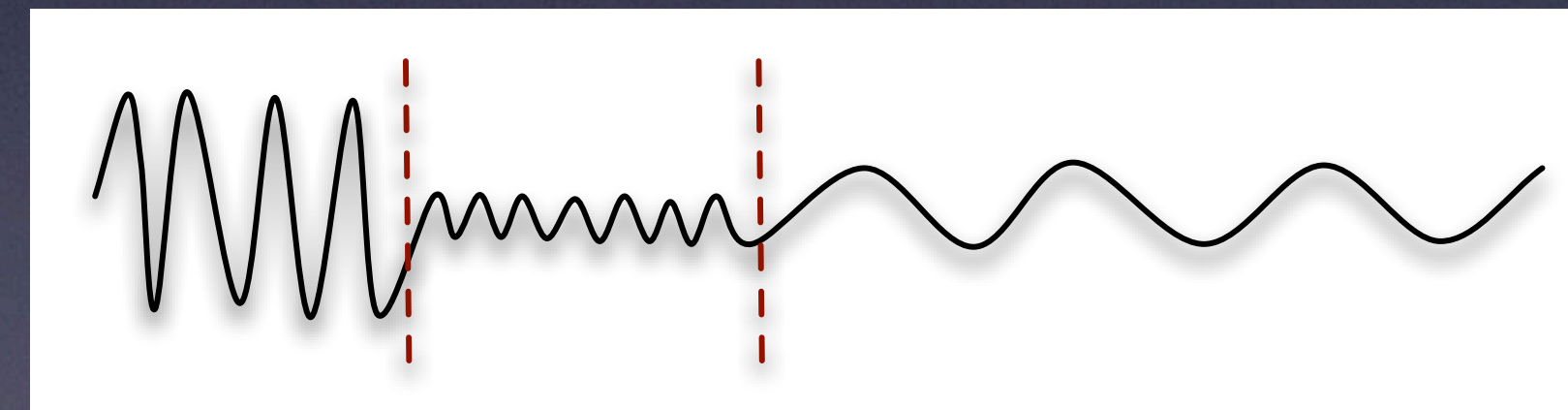
- **Music Information Geometry**

- General approach:
 - Do NOT formalize information content!
 - Control *changes* of information content incrementally

Definition Two entities $\theta_0, \theta_1 \in \mathcal{X}$ are assumed to be *similar* if the information gain by passing from one representation to other is zero or minimal; quantified by $d_X(\theta_0, \theta_1) < \epsilon$ which depends not on the signal itself, but on the probability functions $p_X(x; \theta_0)$ and $p_X(x; \theta_1)$.

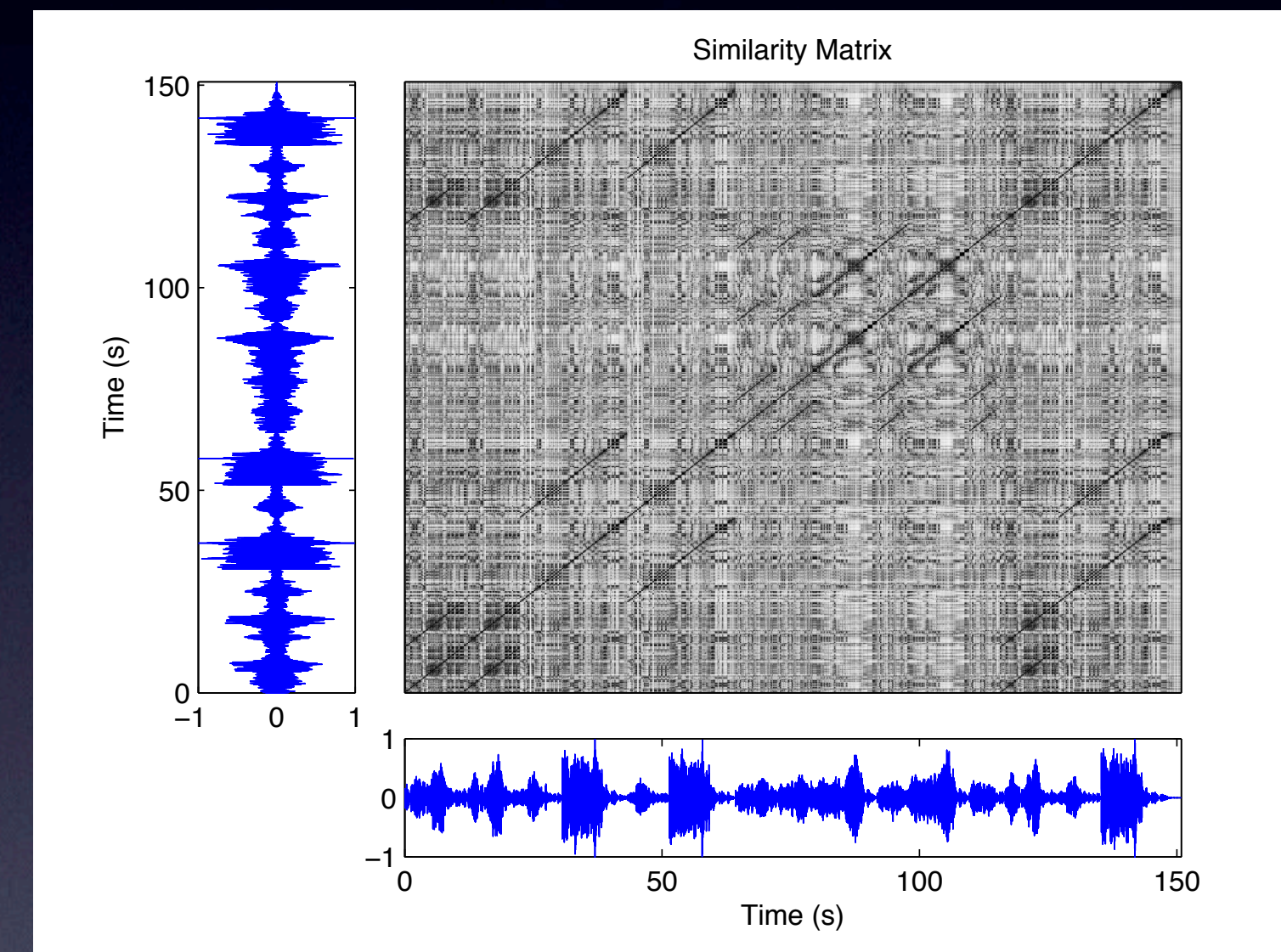
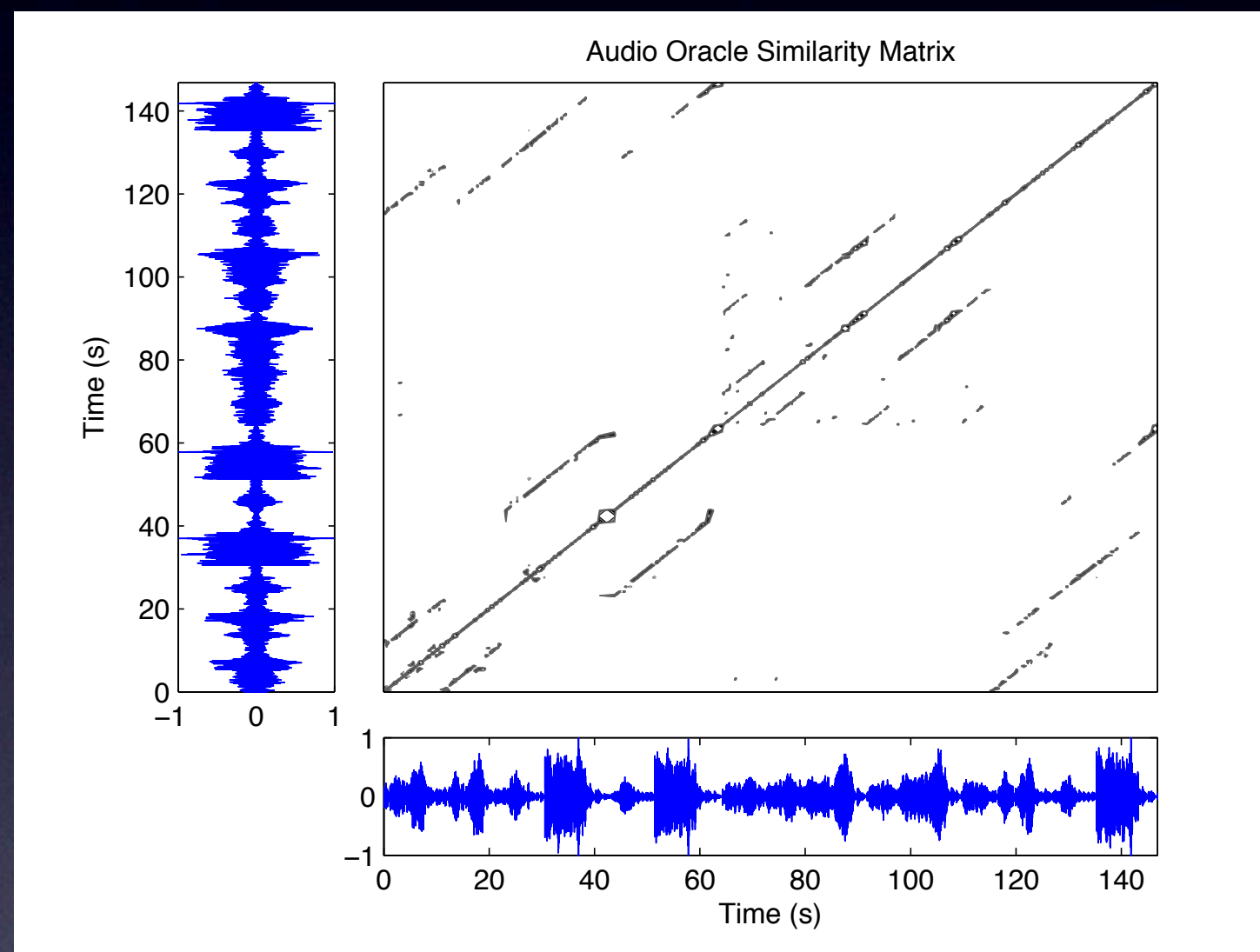
- Parametric approach:

Definition Given a dual structure manifold $(\mathcal{S}, g, \Delta^D, \Delta^{D^*})$ derived on a regular exponential family formed on data-stream X_k , a *model* θ_i consist of a set $\mathcal{X}_i = \{x_k | k \in \mathcal{N}, \mathcal{N} \subset \mathbb{N}\}$ that forms a *Bregman Ball* $B_r(\mu_i, R_i)$ with center μ_i and radius R_i .



Real-time Machine Listening

- **Music Information Geometry**
 - Interesting results for incremental/real-time system



- Major problem: robustness..



Real-time Machine Listening

- **Music Information Geometry**

- Perspectives:

- Non-parametric methods
- Optimal Transport Theory on Wasserstein Geometries of Sound
 - Candidate application: *Sound morphing*
- Formalize sound invariance

- Thesis proposal on EDITE





Reactive Synchronous Programming

Reactive Synchronous Programming

- Interactive Music Design Issue #1
- Authoring: Transcription of thought

```

irl-param reverb tr0 60, reverb trh 0.2, reverb G1 -27,
reverb fl 127.;
irl-param reverb delay minmax 43.7 497.8;
pd-on-off 1;
pd-grain 150;
pd-prime bang;
ir2-param reverb tr0 60, reverb t
reverb fl 127.;
toSynth 93 90 85 84 82 80 79 77 7

```

```

fd1_del 25;
fd1_fre -347;
fd1_db 0;
s1-t1-trait2 bang
src4 fact prer 90
src4 fact pres 105;
src4 source Az 0;
ir2-param reverb tr0 3;
toSynth 74 120 4;
200 toSynth 73 120 4;
200 toSynth 70 120 4;
200 toSynth 69 120 4;
200 toSynth 68 120 4;
200 toSynth 67 120 4;
200 toSynth 66 120 4;
200 toSynth 63 120 4;

```


Reactive Synchronous Programming

- Interactive Music Design Issue #2

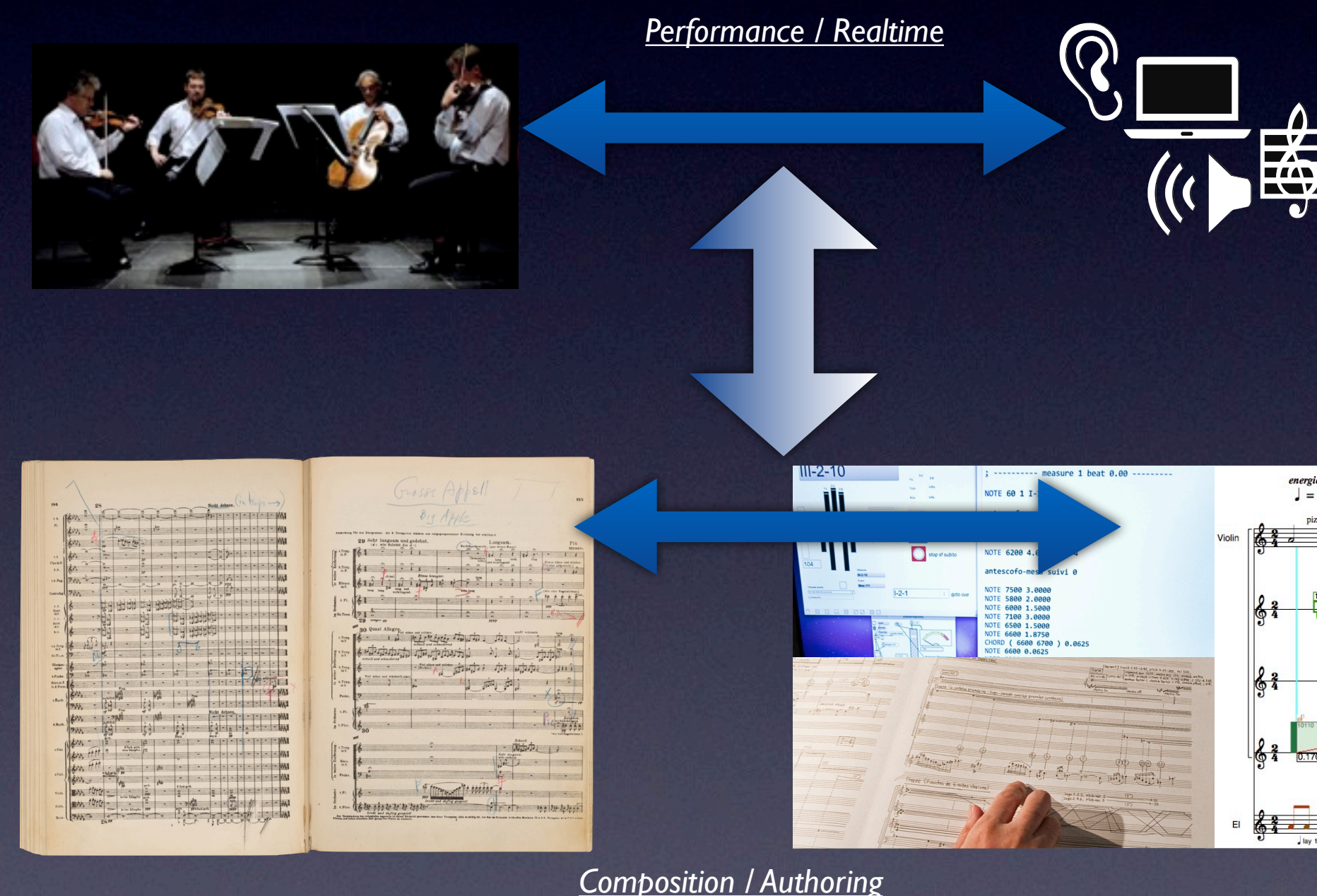
The image displays several Pure Data patches and a control interface for the Pluton system. On the left, a patch titled 'section-2' shows a network of 'pd' objects connected to an 'r cue02' object. Below it, a 'subpatch' window shows a more complex patch with an 'inlet', 'sel' objects, and various signal processing objects like 'ptor', 'rgate', 'rto4', 'fto4', 'spat', 'clrosc', 'mark', 'rtof', 'fto2', 'mark2', 'revfb', 'fsfre', 'Rto4', 'hto4', 'tto4', 'stopdel', 'stopmetro2', and 'stop1.1.25'. In the center, a control interface for 'pluton-07.pd' features a 'sequencer' section with a 'note' field and a 'jump' button, a 'CONTROLS' section with a '0' knob and 'IN asco2 4' meters, and a 'works' section with various checkboxes and buttons. On the right, a 'DSP' patch window shows a dense network of objects including 'pd inputs', 'pd samplers', 'pd fft-osc1', 'pd fft-osc2', 'pd harmonizer', 'pd reverb1', 'pd reverb2', 'pd freq-shift', 'pd reverbl', 'pd reverb2', 'pd harmonizer', and 'pd spatial'. Further right, a 'do-freq-shift' patch window shows an 'inlet', 'hilbert', 'r fsfre', 'mtof', 'phasor', '+ 0.25', 'cos', 'r fpos', 'linedrive 127 1 1.06848 30', 'r fnes', 'line', and 'outlet' objects.

Pluton (1988-1989)
for piano & live electronics
Manoury, Lippe, Puckette



Reactive Synchronous Programming

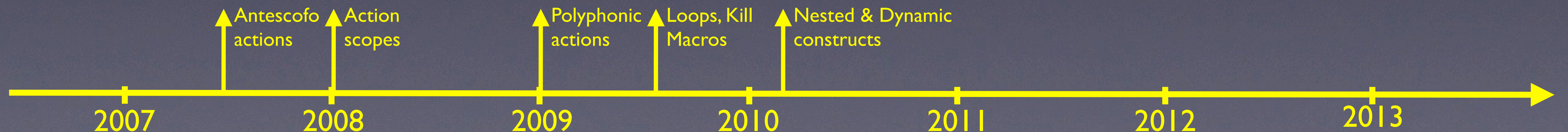
- Interactive Music Design Issue #3
 - The divide between “performative” and “compositional” aspects in Computer Music [Puckette, 2004]
 - Computer Assisted Composition: Rich representation and data-structure, no or few real-time support
 - OpenMusic, PWGL
 - Domain-specific Computing/Practice:
 - Physical Modeling: Waveguides, Modal Synthesis (Modalys)
 - Sound synthesis: CSound, SuperCollider
 - DSP: FAUST
 - Live Coding: Chuck, SuperCollider, Impromptu
- Max & Pd secret: Combining event & signal processing
 - But still “performance oriented”



Reactive Synchronous Programming

- **Storyline:**

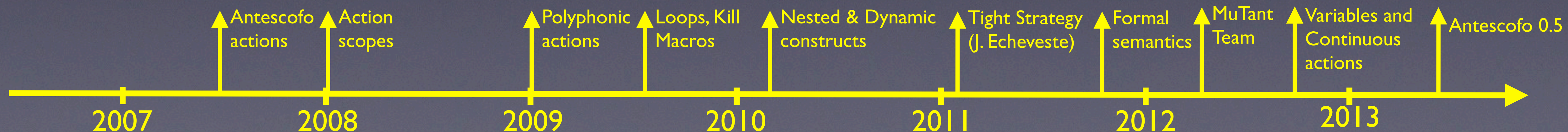
- 2007, Stroppa's "... of Silence" *Antescofo 0.1*
Putting actions next to instrumental events, use decoded tempo to evaluate relative delays
- 2008, Boulez' "... Explosante-Fixe..." in Los Angeles
Action scopes & fault-tolerance
- 2009: *Antescofo 0.3, seperating event & signal proc.*
 - Stroppa's "Hist whist": Polyphonic actions, loops, macros
 - Harvey's "Speakings": loops and fault-tolerance on concurrent actions
 - Nicolaou's "Otemo": Extensive polyphonic actions
- 2010: *Antescofo 0.4, Hierarchical clocks trees*
 - Event and time triggering idea set forth by P. Boulez
 - Nested and dynamics timing in Manoury's "Tensio"
 - Meeting with G. Berry; Involvement with the Synchronous Community



Reactive Synchronous Programming

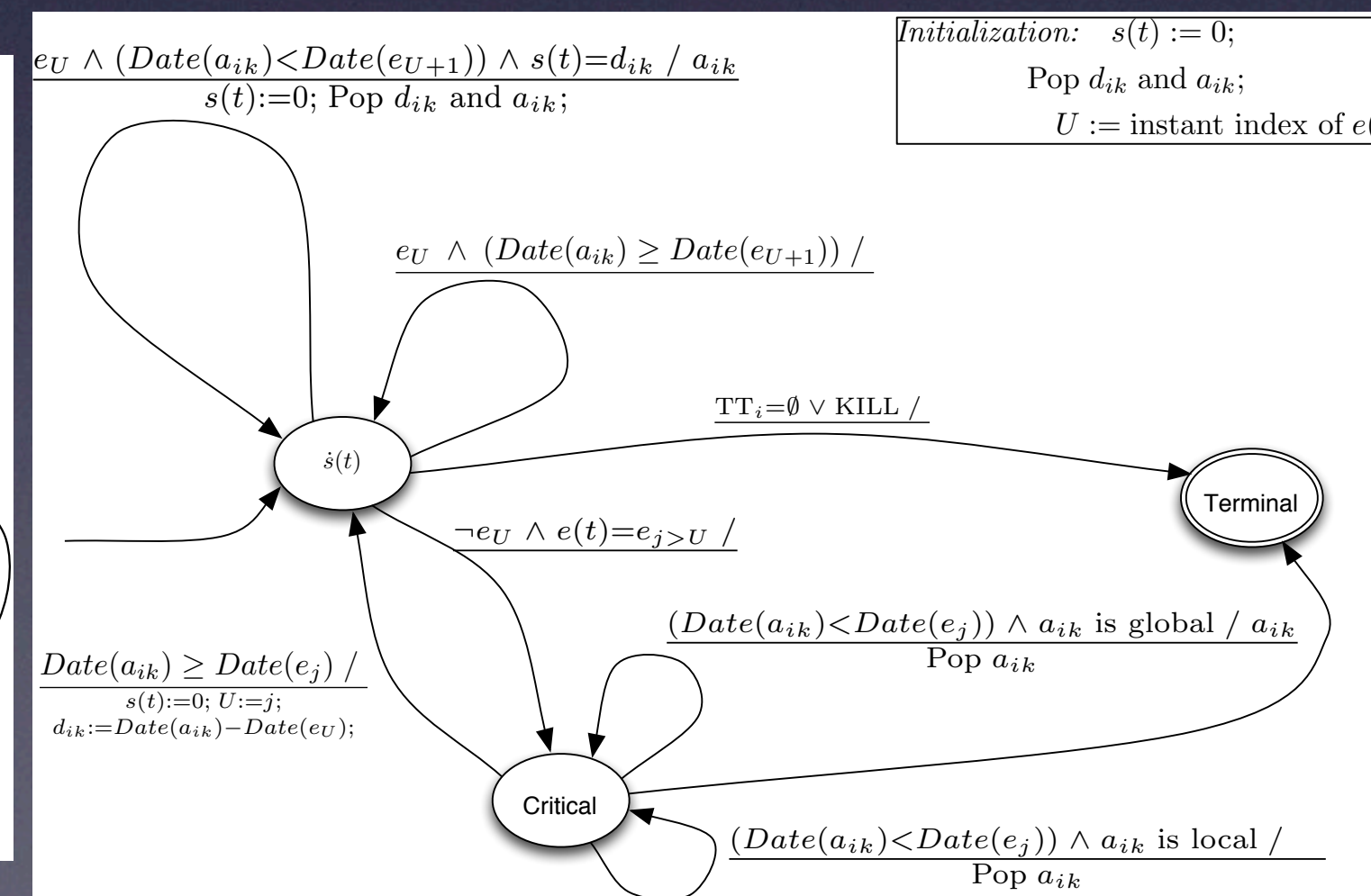
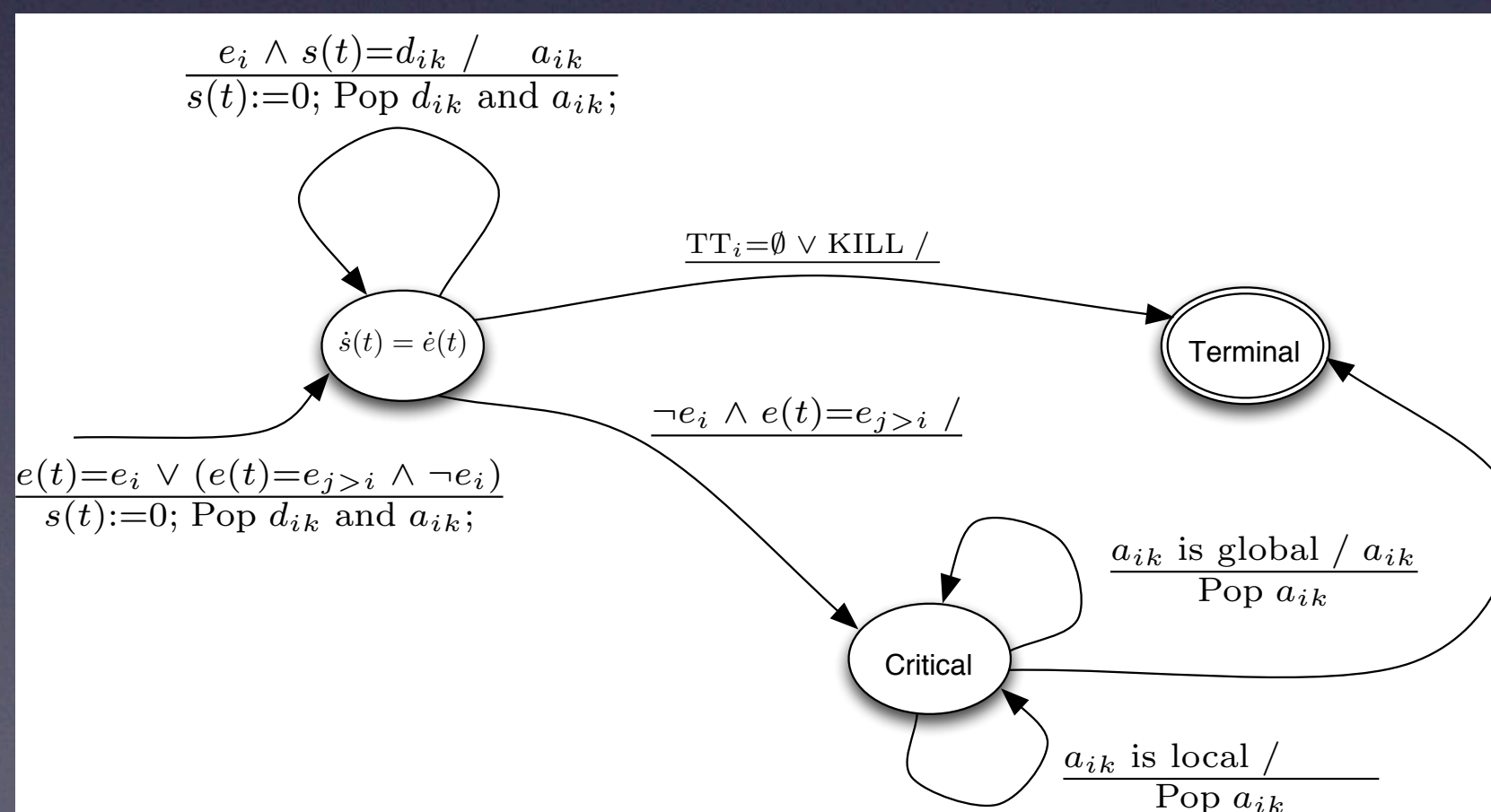
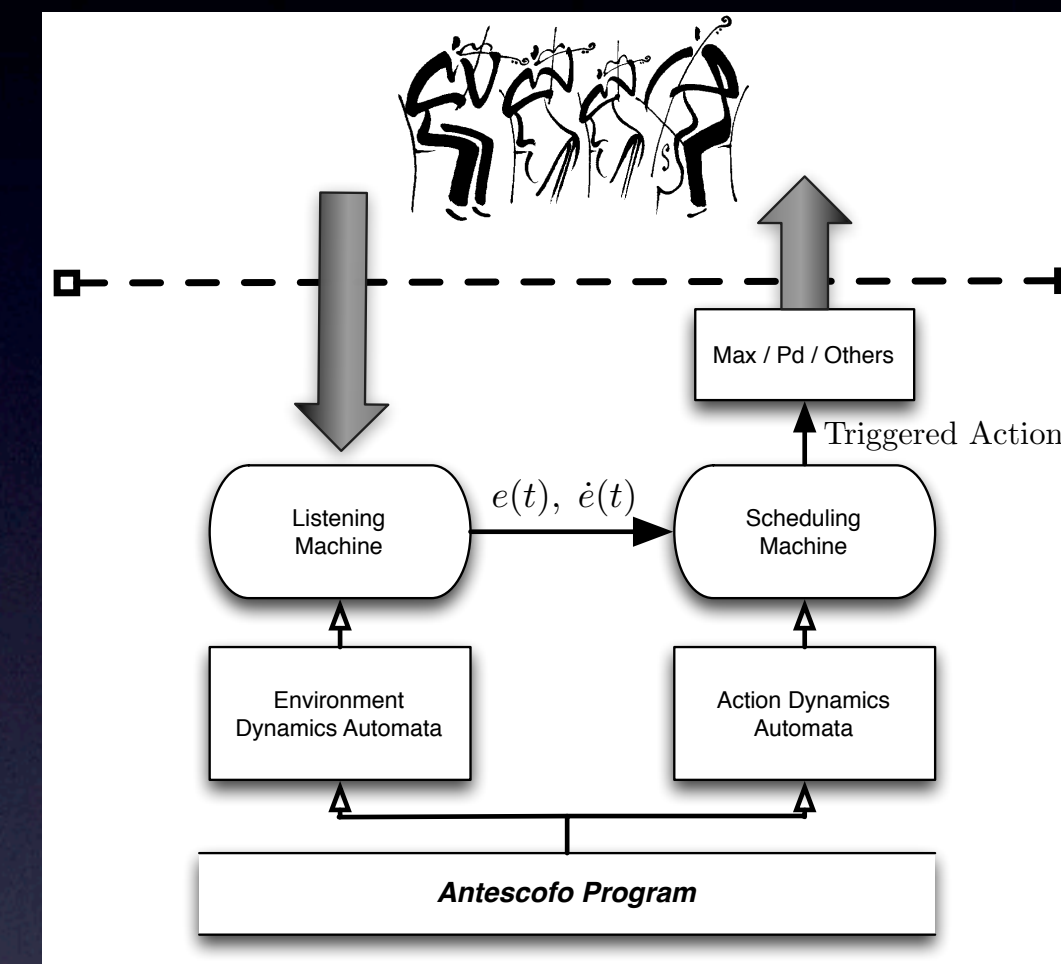
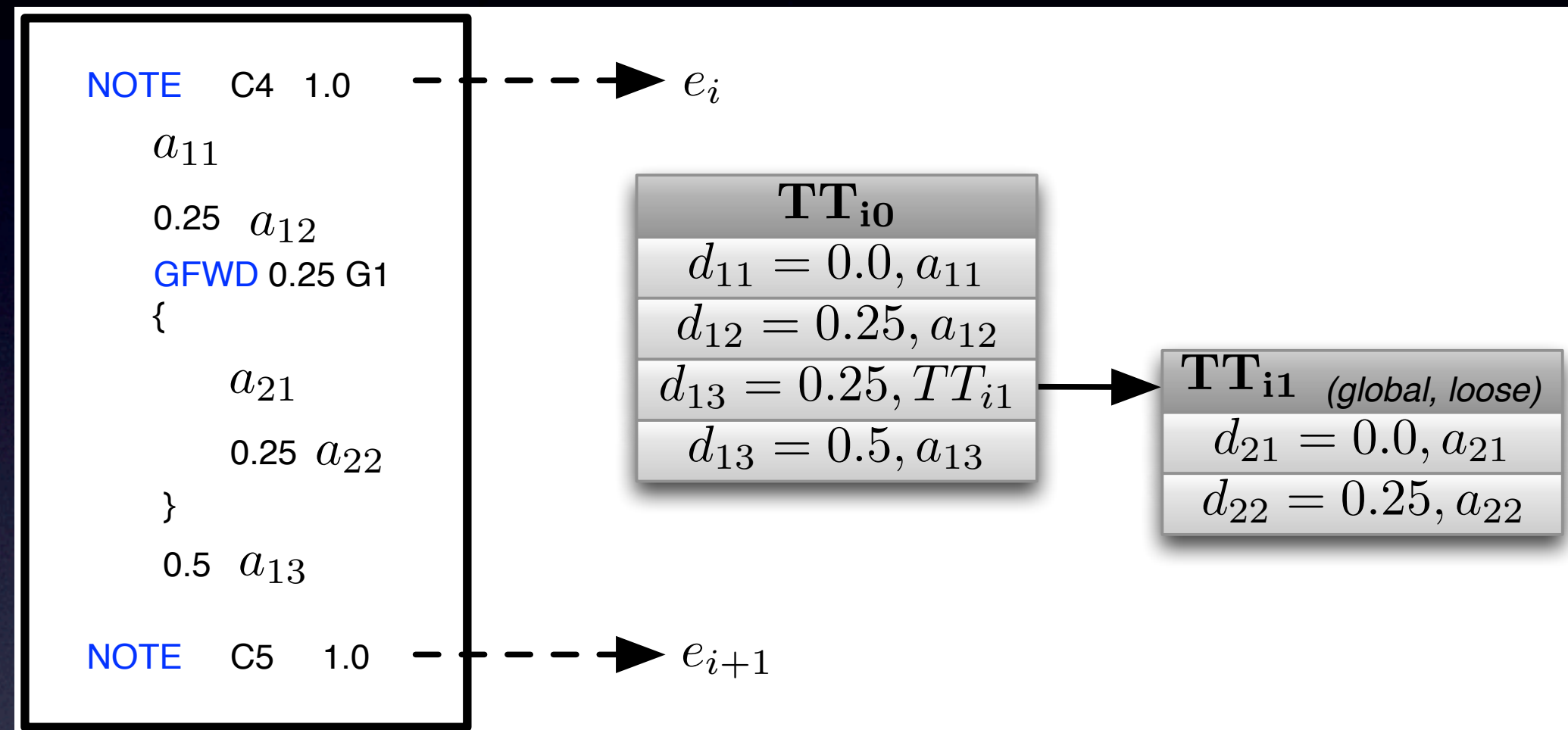
- **Storyline**

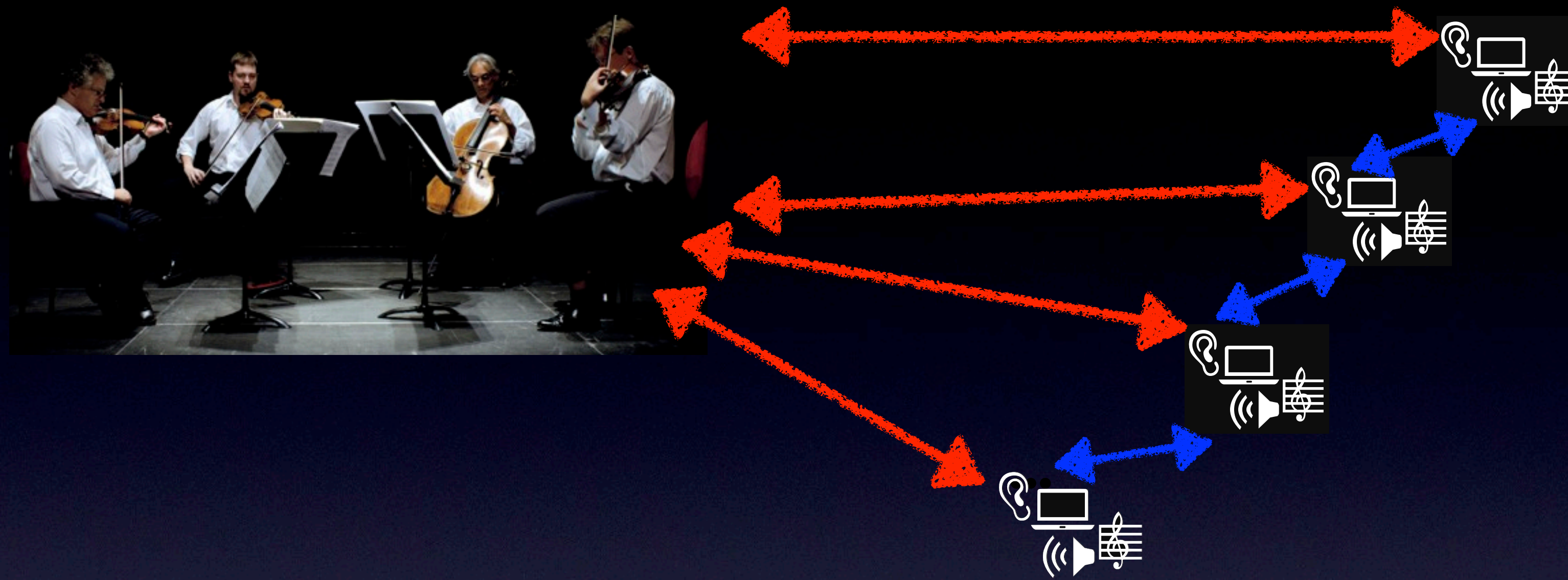
- 2011:
 - Synchronization strategies (José Echeveste)
 - First formal semantic of Antescofo language (MSR'2011)
- 2012:
 - Creation of MuTant team (Inria): Two “real” computer scientists join the team (J.-L. Giavitto, F. Jacquemard)
 - Nodaira’s “Iki-no-michi”
 - Dynamic variables and local tempi (José Echeveste)
 - Continuous Actions
 - First schemes of Verification Analysis (F. Jacquemard)
 - Thomas Coffy joins the team (Inria)
- 2013:
 - [last week!] Release of the new architecture v0.5



Runtime Semantics

- Antescofo's Virtual Machine: Event and Time Triggered Architecture





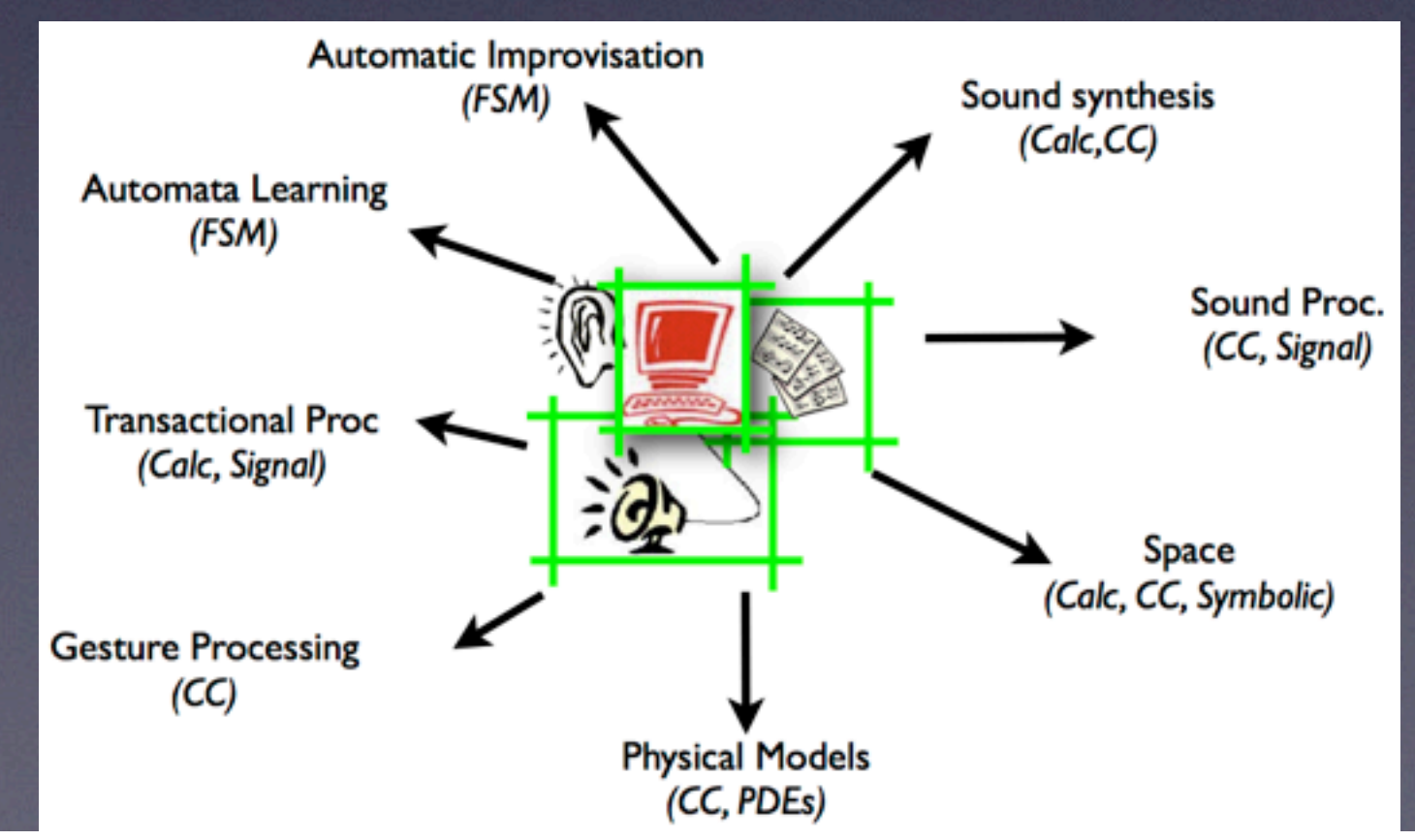
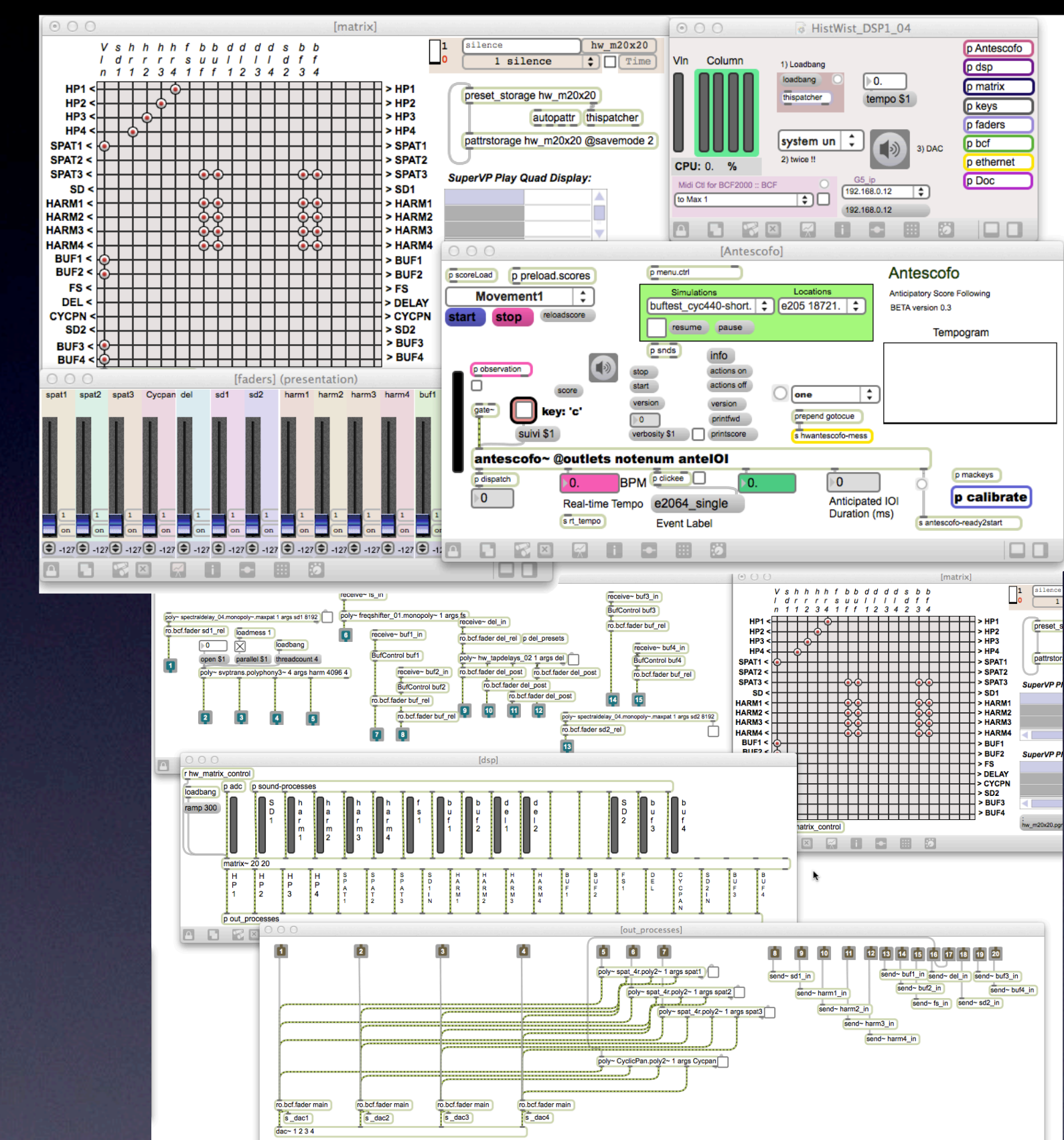
Cyber-Physical Music Systems

so far...

- Three design reasons for the [unexpected] success of Antescofo:
 1. Human Centered Computing
 2. Coupling of Action & Perception
 3. Reliability despite fallible components
- Three practical reasons for the success of Antescofo:
 1. Bridging Compositional & Performative aspects in Computer Music
 2. Linking implementation to behavior
 3. Incremental R&D, strong team involvement with composers and in studios!
- Three missing & blocking features (milestone for Antescofo v1.0)
 1. Coordinating heterogeneous computing paradigms
 2. Beyond “score alignment” for machine listening
 3. Distributed & embedded coordination/orchestration

Practical Situation Today

- The Max paradigm dilemma
 - Gaining CPU power, loosing time correctness
 - “Time has become an accident of implementation”
 - *Scalability*: Chaotic behavior from programs to systems
- The importance of computational abstractions
 - Max: Signal + Event
 - Pure “Functional View” of computing
- No explicit support for interaction between the **cyber** and the **physical** worlds



Scientific Situation Today

- Multimedia Systems Literature
 - The roots of (our accepted) real-time systems for Audio [Steinmetz & Nahrstedt 2004]
 - Focus on data-delivery and end-synchronization
 - Ad-hoc support for “process synchronization”
- Embedded system literature
 - Traditional real-time school vs. Cyber-physical System school
 - One regards **real-time** as a performance issue in terms of feasibility and scheduling (Buttazzo 2005; Burns et al. 2009; etc.)
 - The other concerned with the use of *formally defined language semantics and timing requirements* (Synchronous School, TTA School, Ptolemy, etc.)

Cyber-Physical Systems

- Integration of both **cyber** (software, hardware, network) and **physical** components
- whose dynamics are modeled **jointly**
- New relationships between the Cyber & Physical components require new architectural models that re-define **form** and **function**
- Explicit links between heterogeneous computational elements and physical environment
- Support system engineering & authorship for **high-confidence** real-time audio
- **Cyber-Physical Music Systems**
 - An integrated perspective of real-time audio computing, machine listening, dynamics and control

Cyber-Physical Music Systems

Explicit Coupling of real-time *Machine Listening* and *Sound Computing*, to provide safe and high-confidence *Cyber-Physical Music Systems* to users and designers.

(I) New generation of Machine Listening algorithms

- Methods of *information fusion* for multiple-paradigm listening
- With explicit models of *time*
- The missing OpenCV for Sound
- Embeddable Listening modules

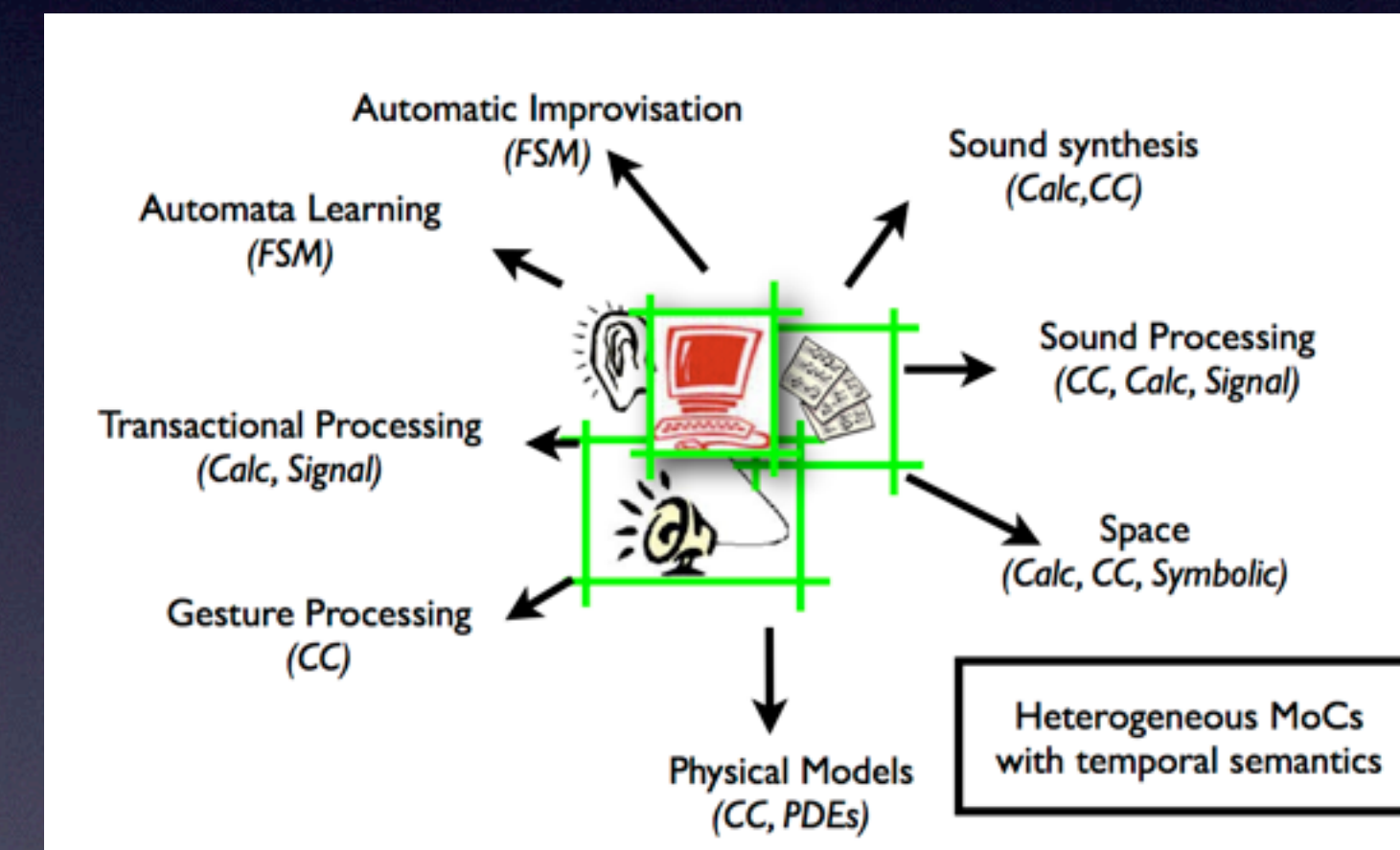
Cyber-Physical Music Systems

(II) New generation of Real-time Audio Computing

- Provably correct abstractions for Sound and Music Computations (MOCs)
- with heterogeneous *temporal semantics*
- leveraging *functional requirements to reaction & execution requirements*
- Dynamic multi-form scheduling mechanisms

(III) Integration

- Joint dynamics between listening and computing
- Provide chain of tools and building blocks supporting end-to-end assurance
- Link implementation to behavior for large systems
- Leverage hardware compatibility



ERC StG Proposal 2013 (pending)

Habilitation à diriger la recherche
Université Pierre et Marie Curie (UPMC)

Arshia Cont

Équipe-projet MuTant

Thank you!