MUSICAL BEHAVIORS:

LAYERED COMPOSITIONAL ALGORITHMS AS PLUGINS FOR THE TRANSFORMATION ENGINE

ALGORITHMIC COMPOSITION IN THE CONTEXT OF "PRACTICAL CREATIVITY"

- "Practical Creativity" - the hands-on creation of any musical work, but especially for:
  - accompaniment of a film, video or play;
  - music that follows a narrative structure;
  - live instrumental performance.

- there are many situations where it is desirable to use algorithmic processes ranging in complexity from *"raise pitch logarithmically one octave over 8 measures",* to an astro-physics simulation (e.g. planetary motions) or fractal structure, mathematical process (Euclidean rhythms).

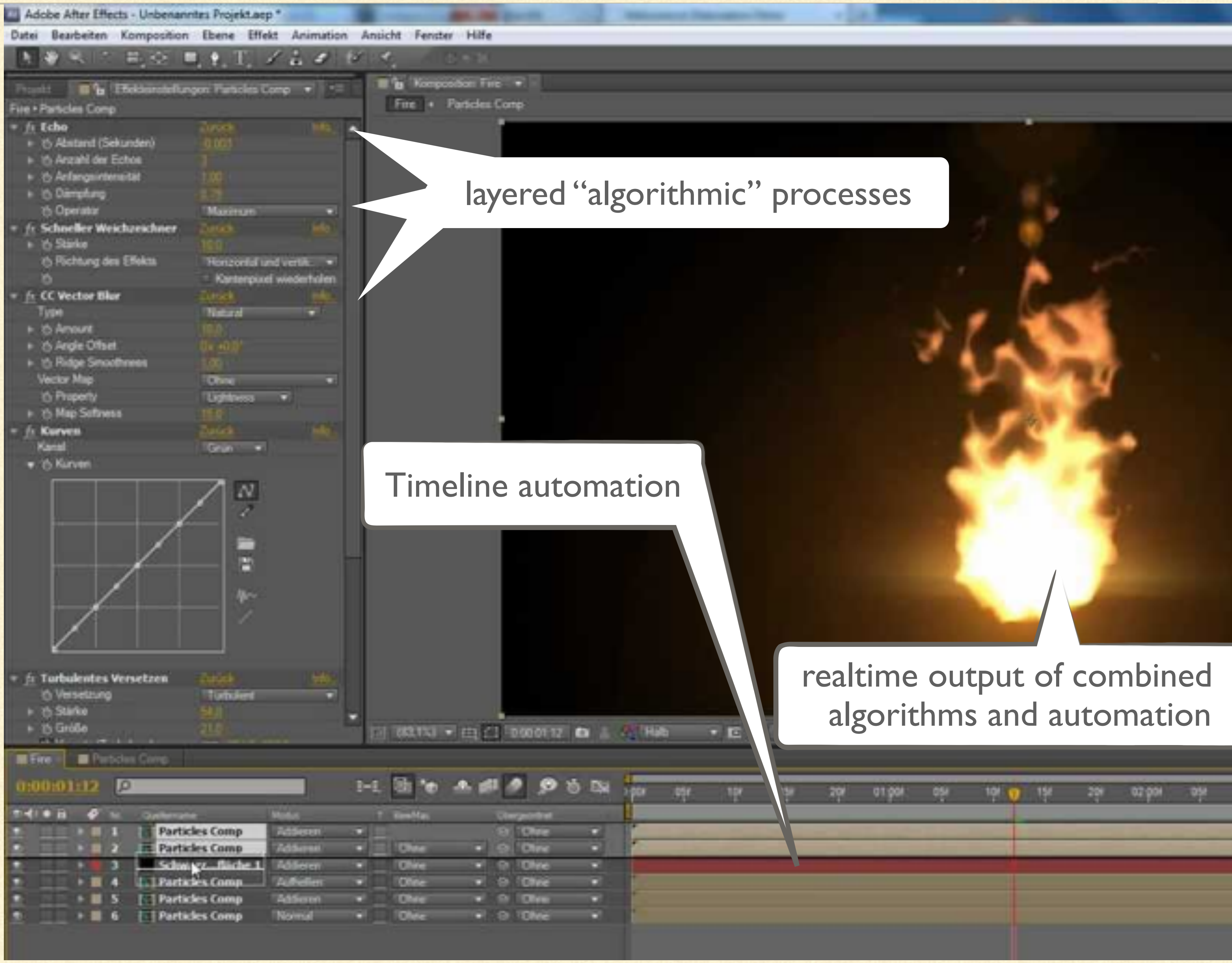
- BUT these algorithmic processes must be further modified, custom shaped, to fit the narrative, or accommodate the limitations of acoustic instruments.

SOFTWARE SUPPORT FOR ALGORITHMIC COMPOSITION

- TWO PROBLEMS:
    - TOO LITTLE: Commercial composition software (Cubase, Logic, etc.) dominated by simulation of multi-track recorder for > 30 years. Only timeline-based modification is supported. Little or no support for algorithmic composition.
    - TOO MUCH: Experimental composition software (e.g. Max-MSP, Processing, PD, etc.) sometimes supports algorithmic approach, but only with global scope. Algorithm 'takes over' all musical processes, prohibiting custom shaping, a requirement for "practical creativity".

VISUAL ANIMATION SOFTWARE (ADOBE AFTER EFFECTS)

- e.g. Adobe After Effects, Autodesk Maya, Apple Motion
- incorporate BOTH modes of control:
  - algorithmic processes as plugins - e.g. particle system
  - timeline-based automation (for "hand shaping")


Adobe After Effects - Unbenanntes Projekt.aep *
Datei Bearbeiten Komposition Ebene Effekt Animation Ansicht Fenster Hilfe
Projekt
Effekteinstellungen: Particles Comp
Fire • Particles Comp
Komposition: Fire
Fire • Particles Comp
Fire
Particles Comp
0:00:01:12
Quellenname
Modus
BewMas
Übergeordnet

Echo — Zurück — Info...
Abstand (Sekunden) -0.001
Anzahl der Echos 3
Anfangsintensität 1.00
Dämpfung 0.75
Operator: Maximum

Schneller Weichzeichner — Zurück — Info...
Stärke 10.0
Richtung des Effekts: Horizontal und vertik...
Kantenpixel wiederholen

CC Vector Blur — Zurück — Info...
Type: Natural
Amount 10.0
Angle Offset 0x +0.0°
Ridge Smoothness 1.00
Vector Map: Ohne
Property: Lightness
Map Softness 15.0

Kurven — Zurück — Info...
Kanal: Grün
Kurven

Turbulentes Versetzen — Zurück — Info...
Versetzung: Turbulent
Stärke 58.0
Größe 21.0

1 Particles Comp — Addieren — Ohne
2 Particles Comp — Addieren — Ohne — Ohne
3 Schwarze Fläche 1 — Addieren — Ohne — Ohne
4 Particles Comp — Aufhellen — Ohne — Ohne
5 Particles Comp — Addieren — Ohne — Ohne
6 Particles Comp — Normal — Ohne — Ohne

layered "algorithmic" processes

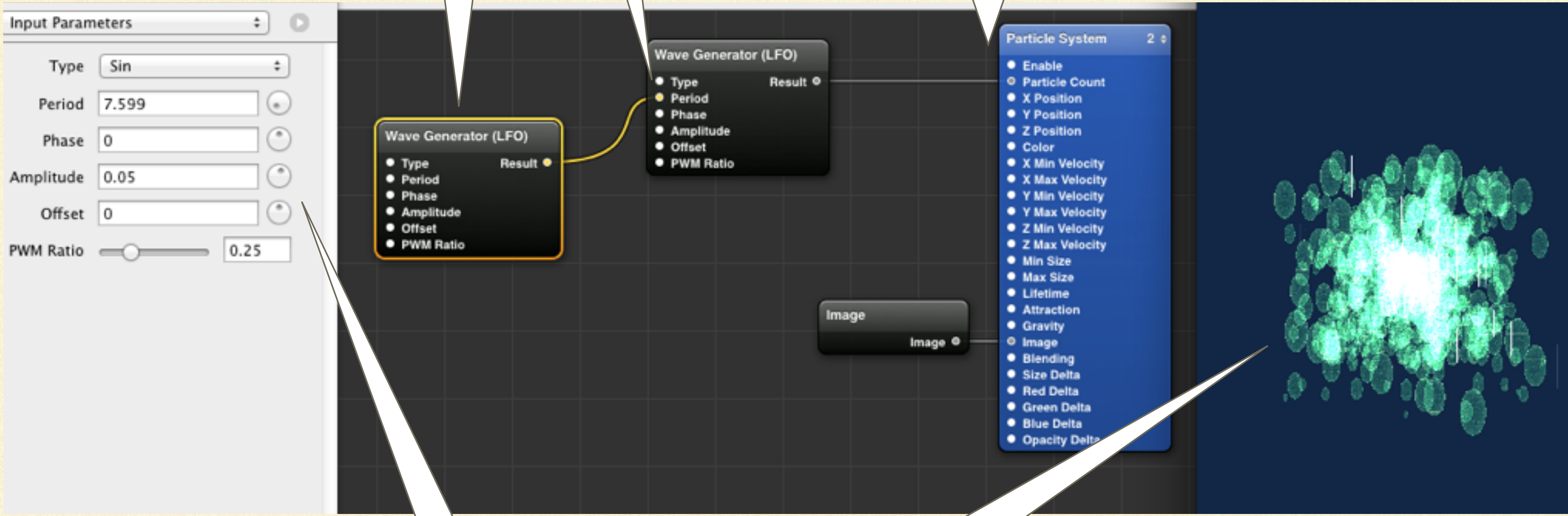
Timeline automation

realtime output of combined algorithms and automation

image: https://www.youtube.com/watch?v=1z9ti_dGgQE

VISUAL ANIMATION SOFTWARE (QUARTZ COMPOSER)

algorithmic processes can control other processes (e.g. frequency modulated LFO controls particle count)



Input Parameters

Type	Sin
Period	7.599
Phase	0
Amplitude	0.05
Offset	0
PWM Ratio	0.25

realtime interactive input;
realtime display of output

PRIOR WORK

- Apple Logic Pro X - includes a MIDI Scripting language
  - seems to be limited to echo effects and arpeggiators (?)
- Cakewalk - Cakewalk Application Language
  - non-realtime only (?)

SOFTWARE REQUIREMENTS FOR MUSICAL BEHAVIORS (NAME FROM APPLE *MOTION* SOFTWARE)

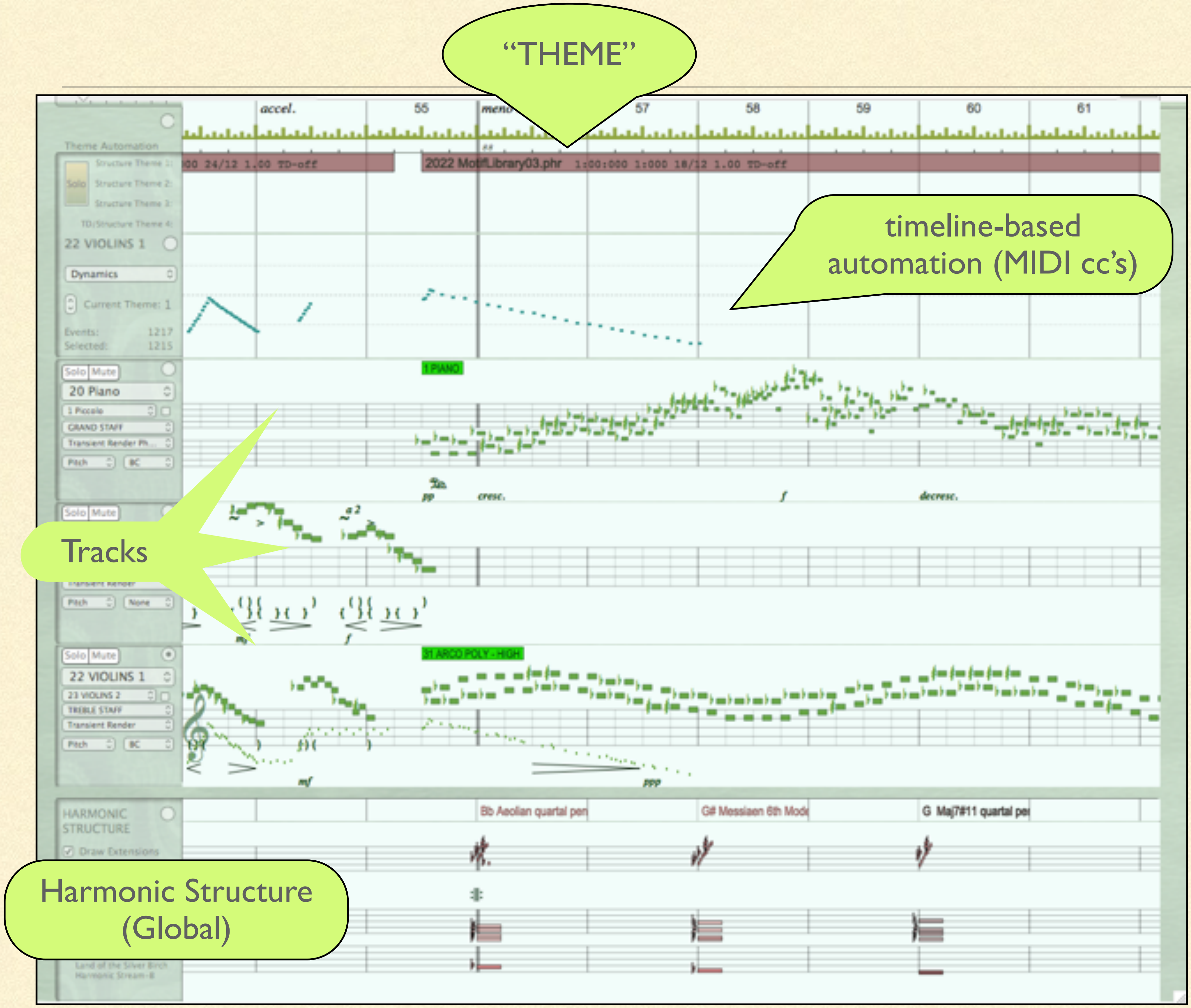
- **DESIDERATA:**
  - **MUSICAL BEHAVIORS MUST:**

    1. co-exist with timeline-based automation
    2. be selectable (i.e. plugin format)
    3. combine correctly with one another (i.e. be layerable)
    4. have clearly defined scope (i.e. limited to a specific time-segment and instrument)
    5. be interactive in realtime, with realtime audio output and graphic display
    6. provide full-featured programming language structures (IF-THEN, LOOPs, etc) and access to sequencer data

  - **BEHAVIORS SHOULD:**

    7. allow programmable interconnection between one another

THE TRANSFORMATION ENGINE



"THEME"

timeline-based automation (MIDI cc's)

Tracks

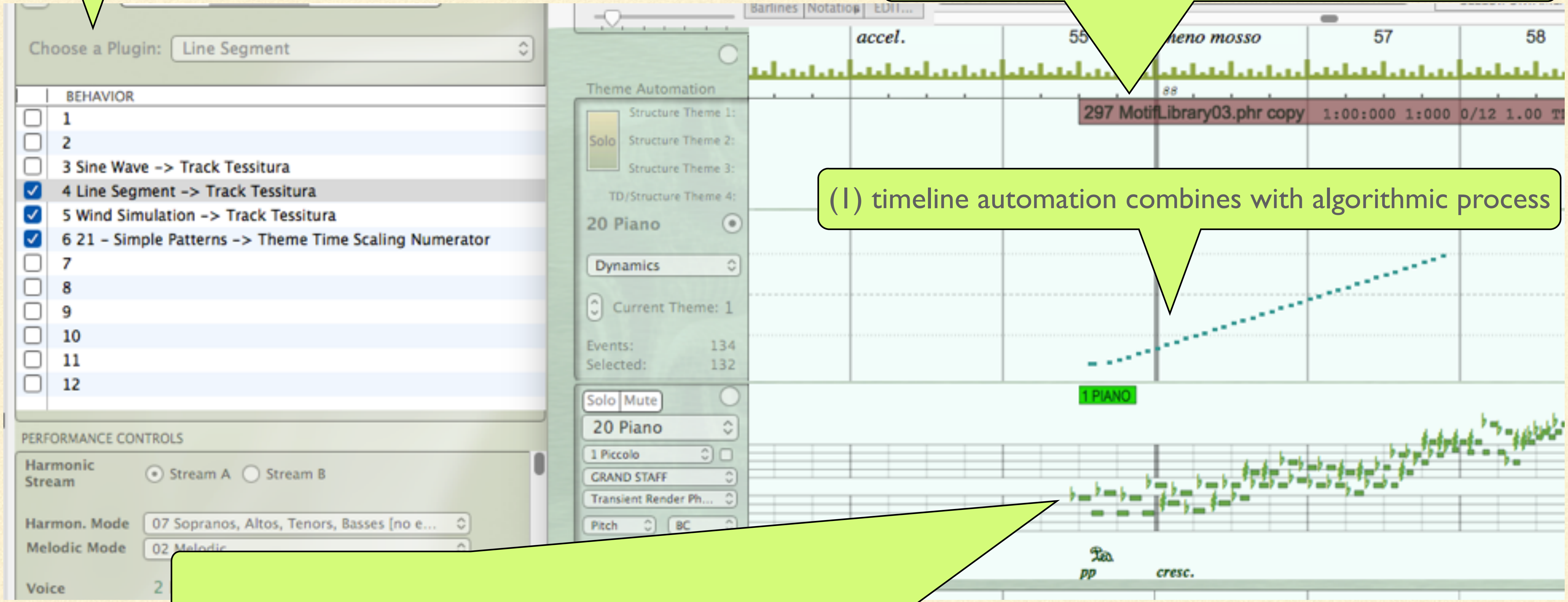
Harmonic Structure (Global)

- personal composition software
- oriented to traditional music composition (i.e. themes, motivic development, harmonic structure)
- MIDI-based, with extensions for Open Sound Control (OSC), OpenGL & MusicXML

BEHAVIORS IN THE TRANSFORMATION ENGINE

selectable (2), layered (3) "algorithmic" processes

(4) scope of algorithm is limited to theme & track



Choose a Plugin: Line Segment

BEHAVIOR

- □ 1
- □ 2
- □ 3 Sine Wave -> Track Tessitura
- ☑ 4 Line Segment -> Track Tessitura
- ☑ 5 Wind Simulation -> Track Tessitura
- ☑ 6 21 - Simple Patterns -> Theme Time Scaling Numerator
- □ 7
- □ 8
- □ 9
- □ 10
- □ 11
- □ 12

PERFORMANCE CONTROLS

Harmonic Stream: ● Stream A ○ Stream B

Harmon. Mode: 07 Sopranos, Altos, Tenors, Basses [no e...

Melodic Mode: 02 Melodic

Voice: 2

Theme Automation

Solo Structure Theme 1: Structure Theme 2: Structure Theme 3: TD/Structure Theme 4:

20 Piano

Dynamics

Current Theme: 1

Events: 134

Selected: 132

Solo Mute

20 Piano

1 Piccolo

GRAND STAFF

Transient Render Ph...

Pitch

BC

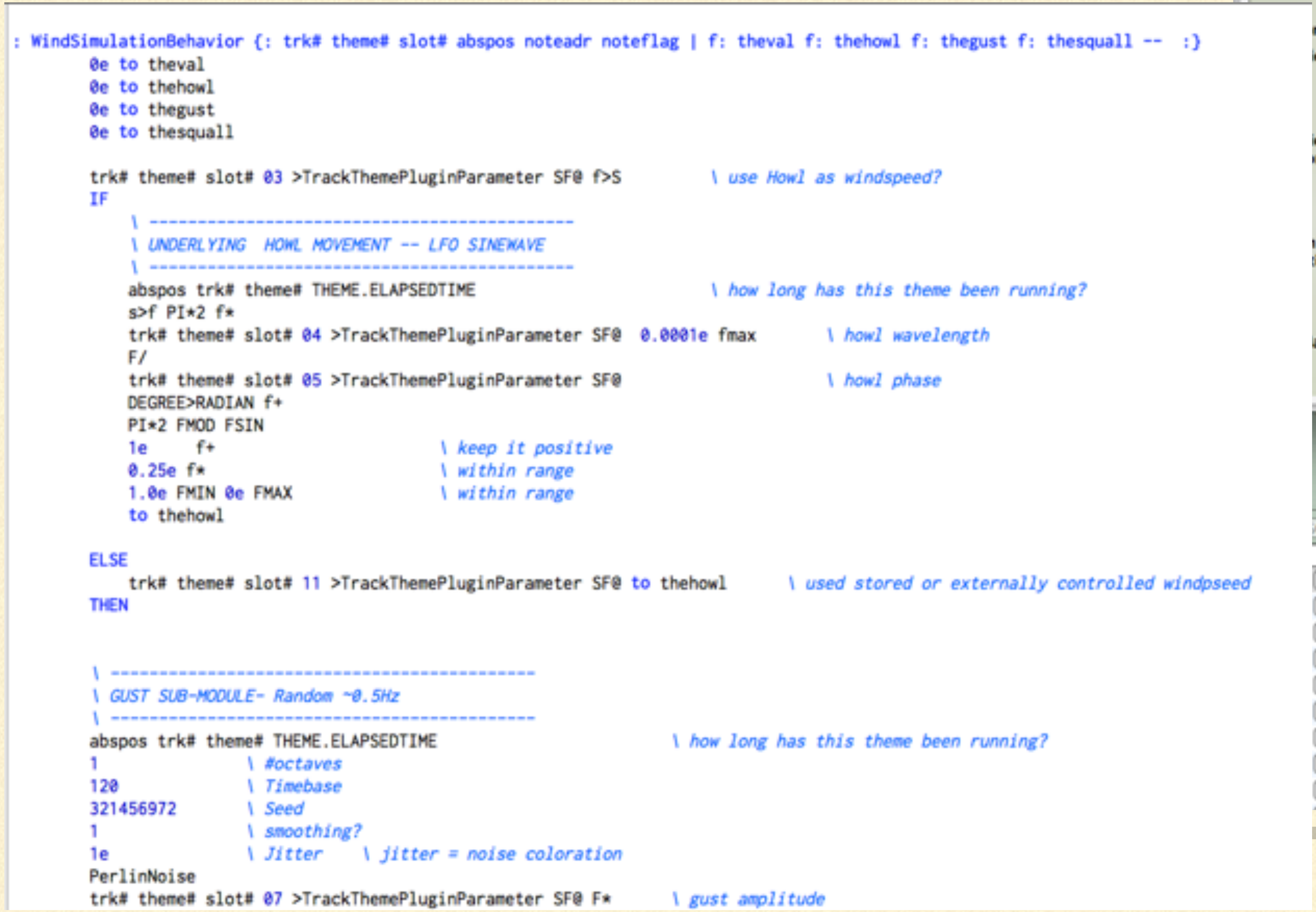
297 MotifLibrary03.phr copy 1:00:000 1:000 0/12 1.00

(1) timeline automation combines with algorithmic process

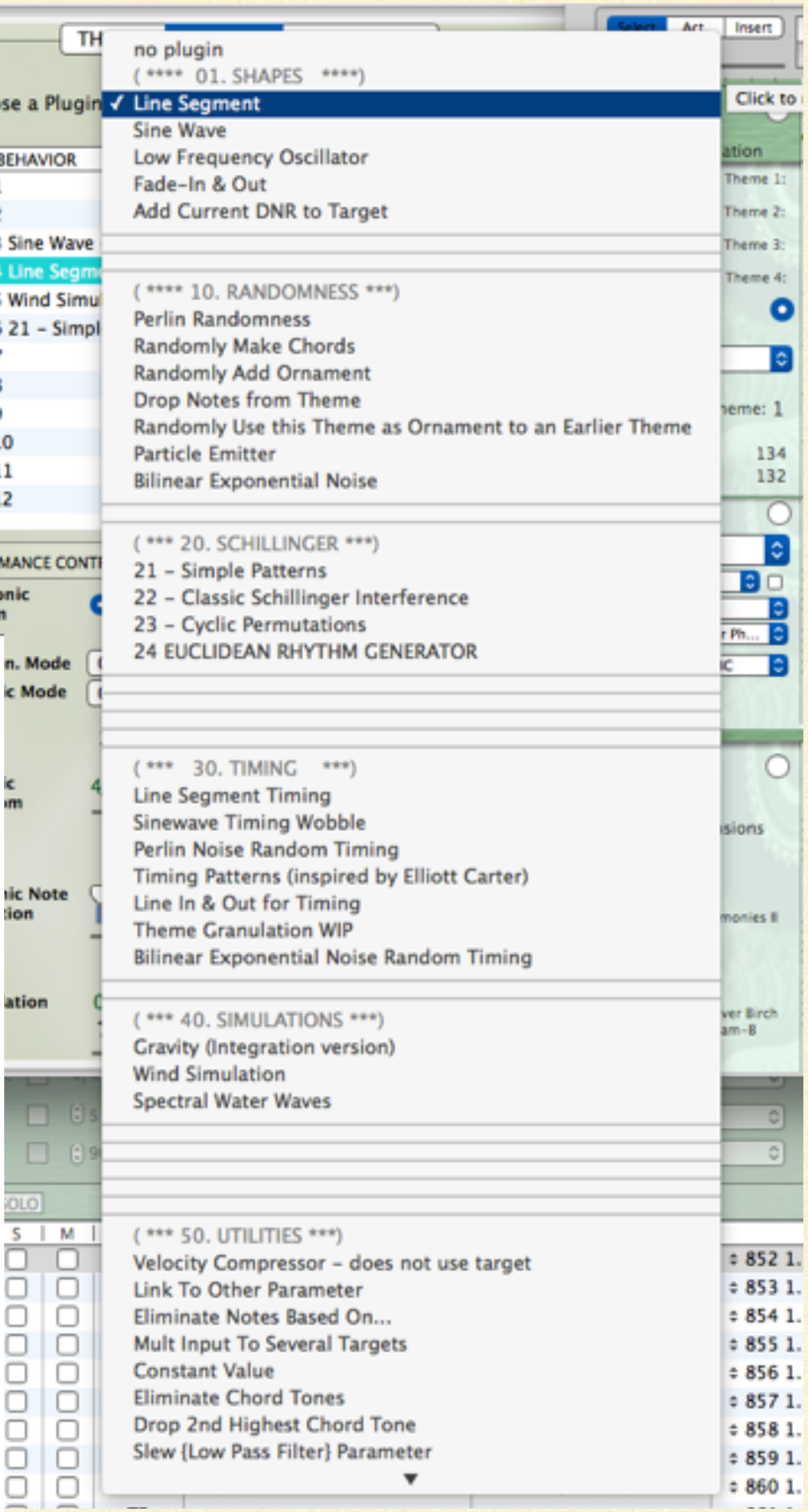
(5) realtime display of automation + algorithmic output

FULL PROGRAMMING LANGUAGE SUPPORT

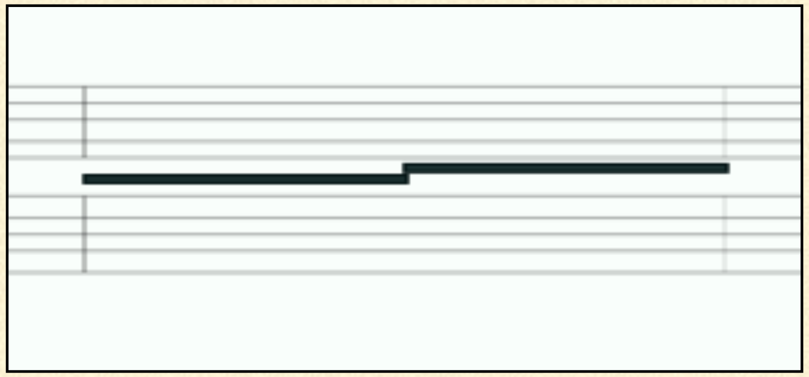
- (6) plugins are programmed in the host language, VFXForth, compiled from text source-code



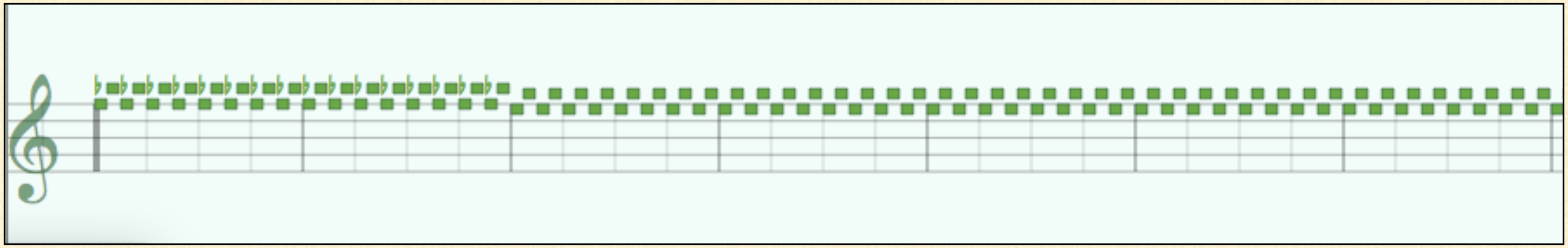
```
: WindSimulationBehavior {: trk# theme# slot# abspos noteadr noteflag | f: theval f: thehowl f: thegust f: thesquall --  :}
        0e to theval
        0e to thehowl
        0e to thegust
        0e to thesquall

        trk# theme# slot# 03 >TrackThemePluginParameter SF@ f>S          \ use Howl as windspeed?
        IF
            \ ---------------------------------------------
            \ UNDERLYING  HOWL MOVEMENT -- LFO SINEWAVE
            \ ---------------------------------------------
            abspos trk# theme# THEME.ELAPSEDTIME                       \ how long has this theme been running?
            s>f PI*2 f*
            trk# theme# slot# 04 >TrackThemePluginParameter SF@  0.0001e fmax        \ howl wavelength
            F/
            trk# theme# slot# 05 >TrackThemePluginParameter SF@                       \ howl phase
            DEGREE>RADIAN f+
            PI*2 FMOD FSIN
            1e      f+                        \ keep it positive
            0.25e f*                          \ within range
            1.0e FMIN 0e FMAX                 \ within range
            to thehowl

        ELSE
            trk# theme# slot# 11 >TrackThemePluginParameter SF@ to thehowl        \ used stored or externally controlled windpseed
        THEN


        \ ---------------------------------------------
        \ GUST SUB-MODULE- Random ~0.5Hz
        \ ---------------------------------------------
        abspos trk# theme# THEME.ELAPSEDTIME                       \ how long has this theme been running?
        1               \ #octaves
        120             \ Timebase
        321456972       \ Seed
        1               \ smoothing?
        1e              \ Jitter     \ jitter = noise coloration
        PerlinNoise
        trk# theme# slot# 07 >TrackThemePluginParameter SF@ F*         \ gust amplitude
```



DEMONSTRATION: LFO BEHAVIOR

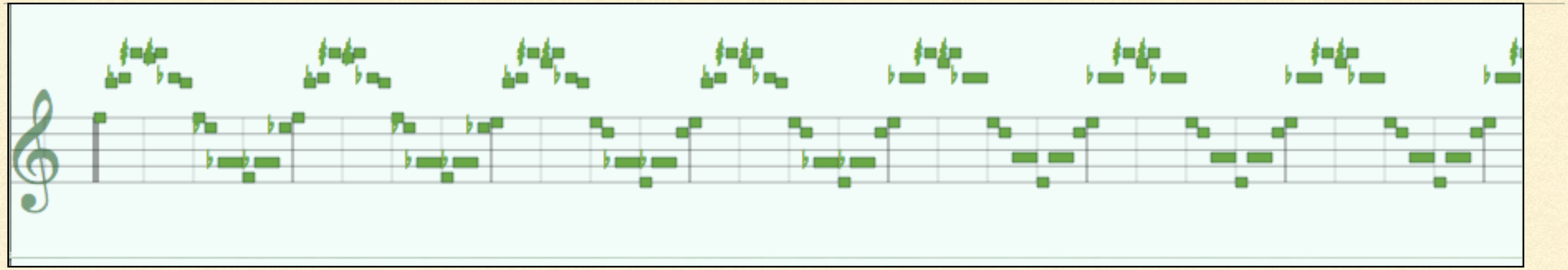


‘theme’ consists of two eighth notes

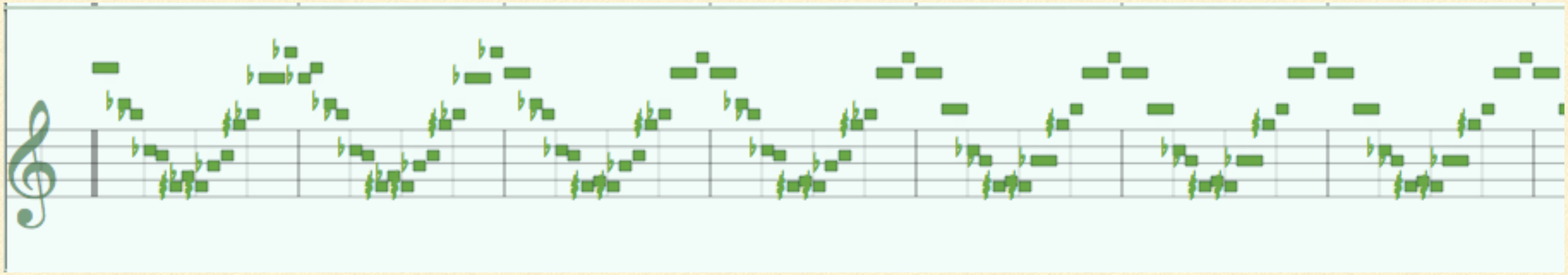


NO BEHAVIORS - theme repeats verbatim, with harmonic changes (doubled speed is due to timeline automation settings)

DEMONSTRATION: LFO BEHAVIOR

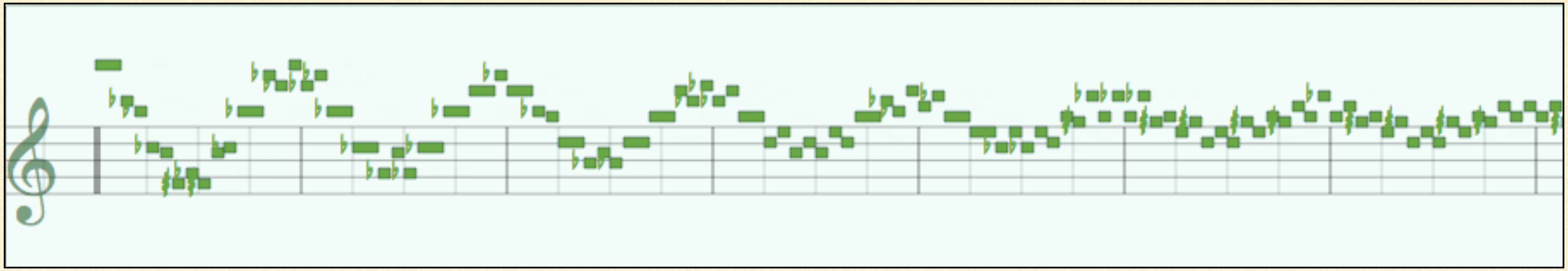


add LFO BEHAVIOR - Shape: Sine Wave 100%;
Range - +- 12 semitones;
Wavelength: 960 ticks = one measure
Phase: 0 degrees



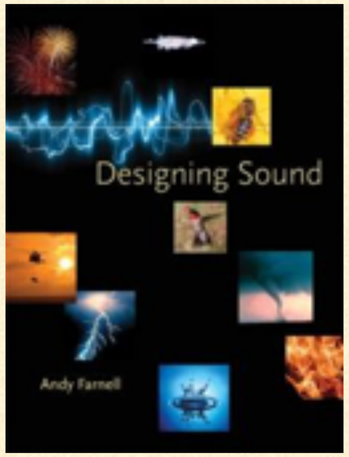
adjust Phase: 110 degrees

DEMONSTRATION: LFO BEHAVIOR



adjust Damping: -2.5%

DEMONSTRATION: WIND SIMULATION BEHAVIOR



- algorithm derived from Andy Farnell, *Designing Sound* (MIT Press) , pp.475 ff

- algorithm originally written in *PureData*

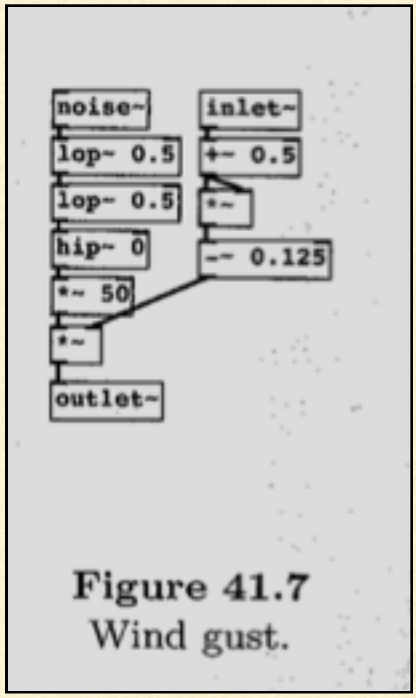


Figure 41.7
Wind gust.

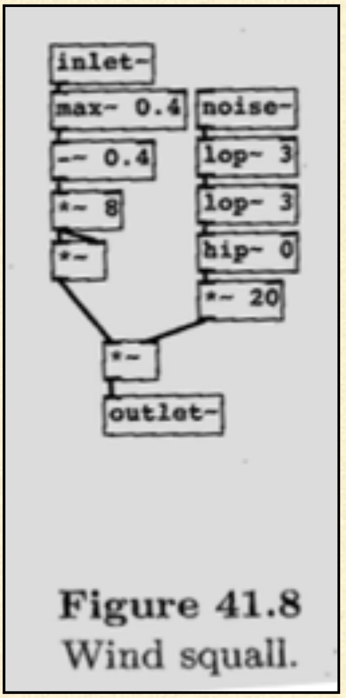
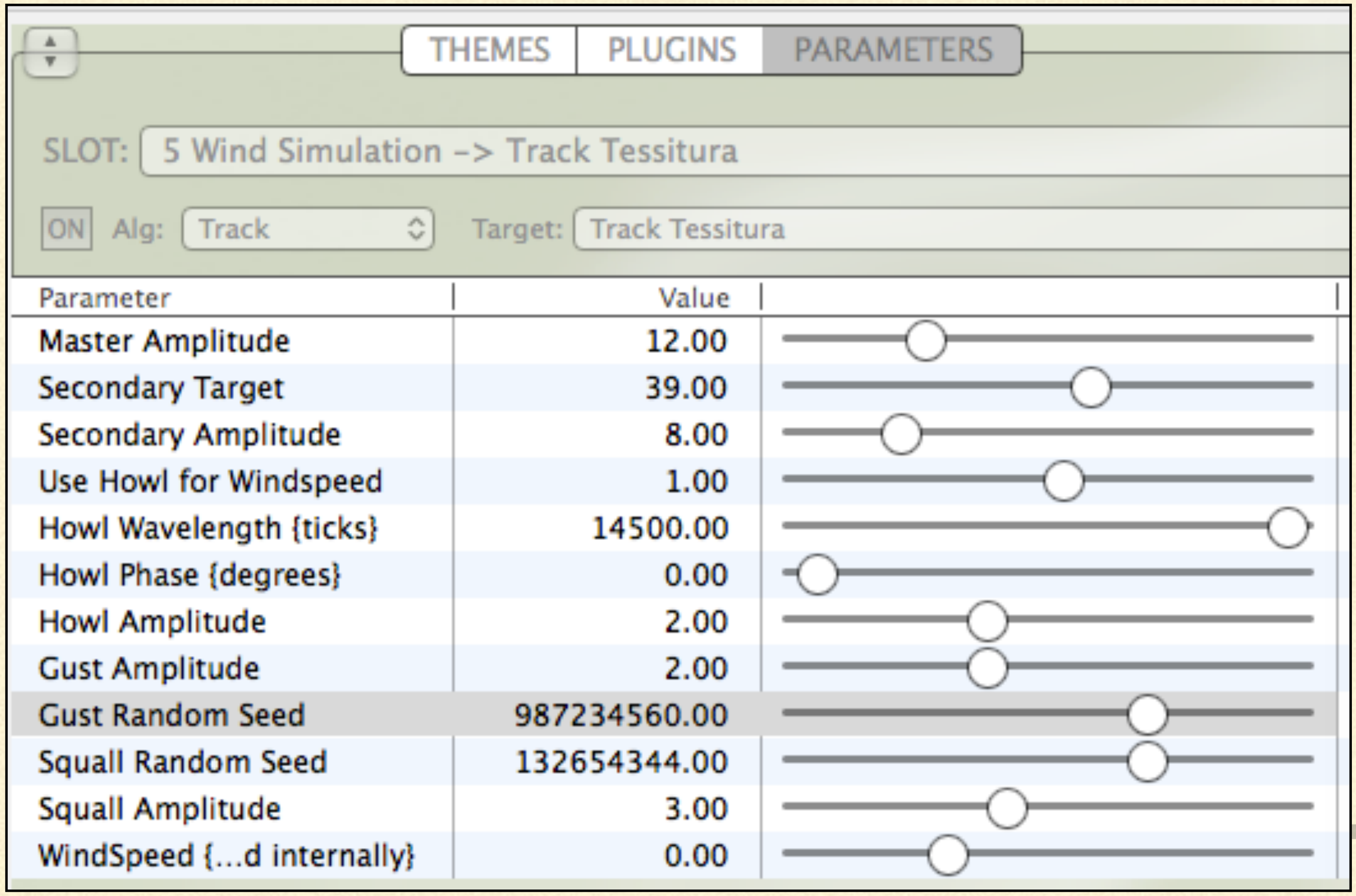


Figure 41.8
Wind squall.

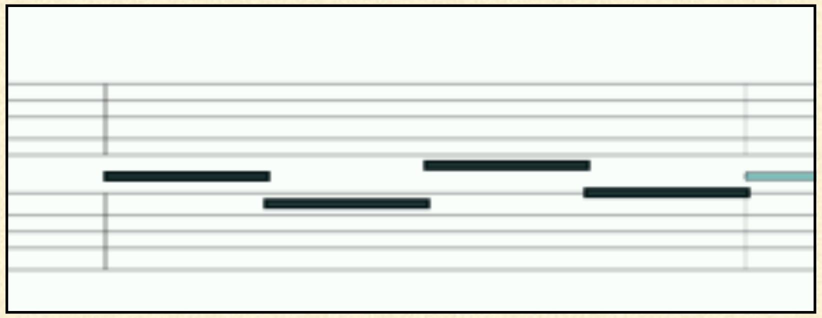


SLOT: 5 Wind Simulation -> Track Tessitura

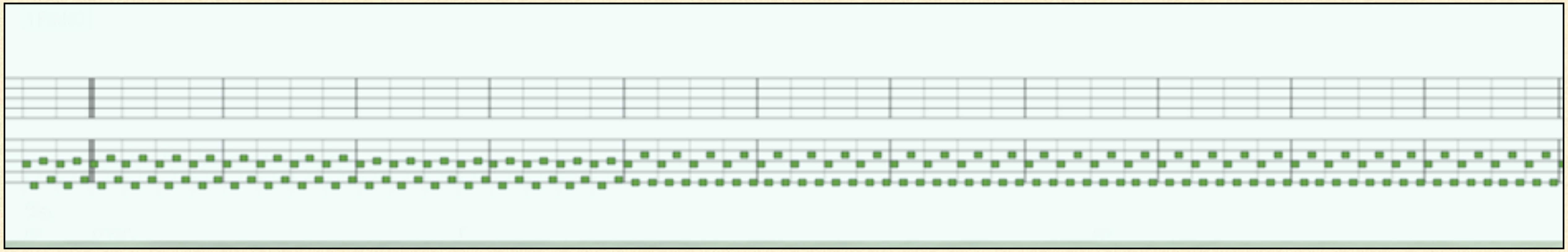
ON Alg: Track Target: Track Tessitura

Parameter	Value
Master Amplitude	12.00
Secondary Target	39.00
Secondary Amplitude	8.00
Use Howl for Windspeed	1.00
Howl Wavelength {ticks}	14500.00
Howl Phase {degrees}	0.00
Howl Amplitude	2.00
Gust Amplitude	2.00
Gust Random Seed	987234560.00
Squall Random Seed	132654344.00
Squall Amplitude	3.00
WindSpeed {...d internally}	0.00

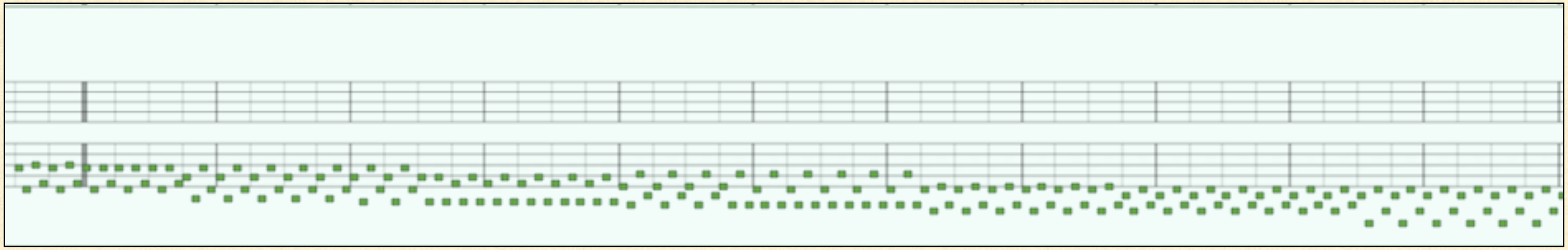
- three components of wind are: *Howl*, *Gust* and *Squall.* Each component has separate amplitude control.



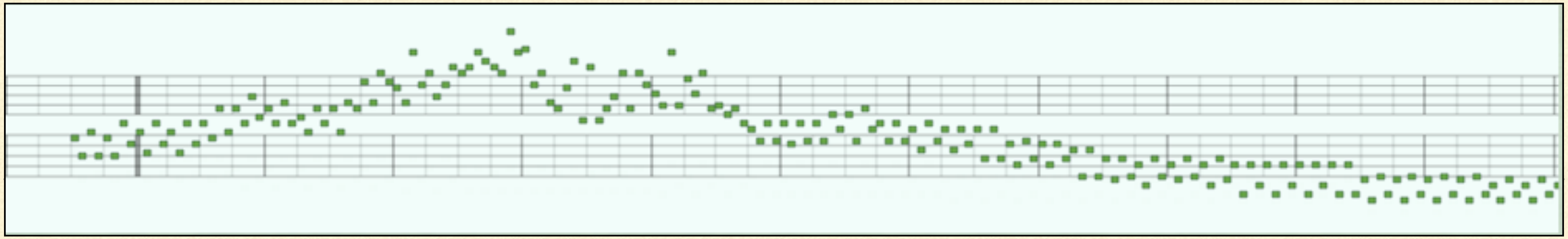
‘theme’ consists of four sixteenth notes



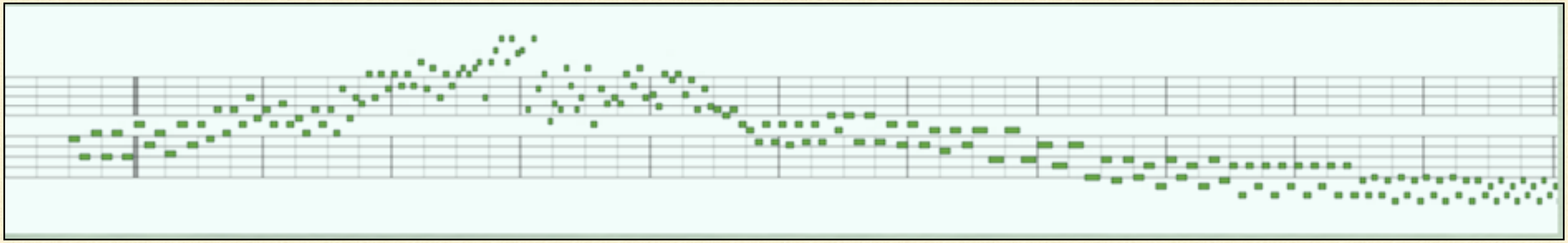
NO BEHAVIORS - theme repeats verbatim, with harmonic changes



LINE SEGMENT BEHAVIOR - adds a one octave drop over phrase

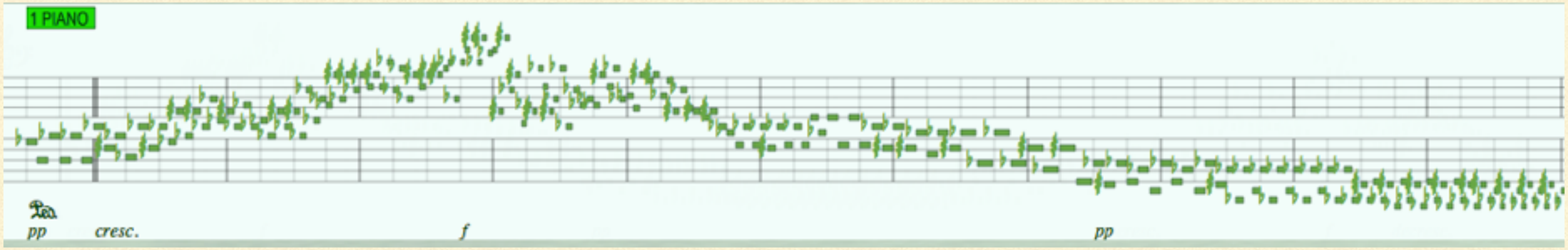


WIND SIMULATION - gives new contour, with gust and squall detail

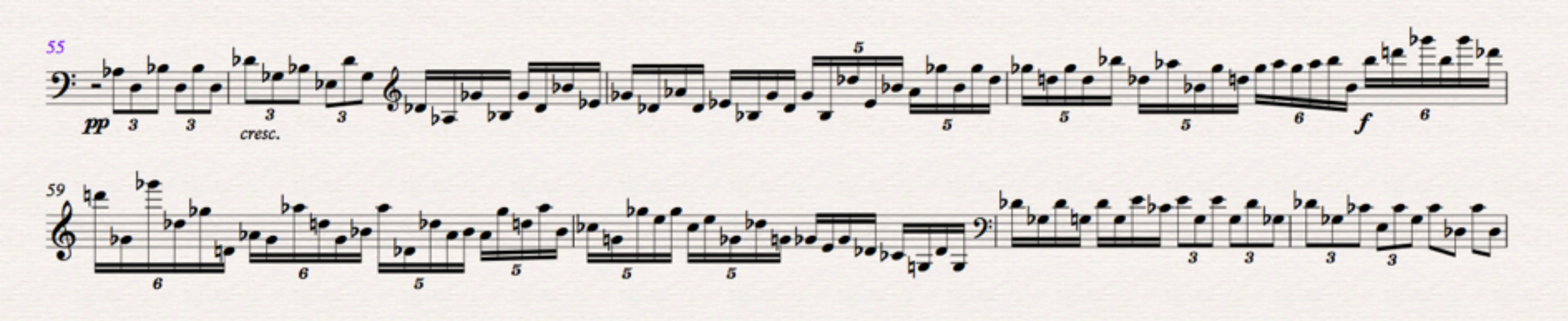


SIMPLE PATTERN BEHAVIOR - gives variety of rhythmic pace

DEMONSTRATION: WIND SIMULATION BEHAVIOR



graphic display, including MusicXML notation



MusicXML output converted to CMN via Sibelius

CONCLUSION

- "Musical Behaviors" in The Transformation Engine provide a software composition environment suitable for "practical creativity" by fulfilling the desired characteristics:

  - √ co-exist with timeline-based automation
  - √ individually selectable
  - √ layer-able
  - √ have clearly defined scope (i.e. limited to a specific time-segment and instrument)
  - √ be interactive in realtime, with realtime audio output and graphic display
  - √ provide full-featured programming language structures (IF-THEN, LOOPs, etc) and access to sequencer data
  - √ allow programmable interconnection between one another (not demonstrated)