

RMS: The Revilo™ Micro-tuned Sampler

(better name to come)

DRAFT 0.1 - Stephen T. Pope - 7/6/07

**NOTE: THIS IS A PROPRIETARY INTERNAL DESIGN DOCUMENT OF
TRANSNANO AND IS NOT INTENDED FOR DISTRIBUTION**

Overview

The Transnano Revilo keyboard significantly extends the range of performance gestures available to keyboard players. To make the most of this keyboard, or to increase the musicality of a traditional MIDI-based keyboard synthesizer, we are developing an artificial intelligence based MIDI sampler that will follow a live performance and adjust the tuning of the notes played depending on the harmonic context. The goal is to develop a MIDI sampler (or, possibly, a plug-in to an existing MIDI sampler) that can dynamically adapt to a near-just-intoned tuning, i.e., it can deviate from the equal-tempered scale so that the important notes in a chord are low-order fractional pitch ratios relative to the current key or tonal center. We call this system the Revilo Micro-tuned Sampler (RMS), and this document is the draft RMS requirements specification and design overview.

Requirements

RMS is a real-time software system that must:

1. take MIDI notes (and extended messages from the Revilo keyboard) as input,
 2. interactively determine the chords being played,
 3. based on the chords played, determine the key or tonal center,
 4. based on the current key, de-tune the keys being played according to a more "natural" temperament, and
 5. play back samples (e.g., piano notes) de-tuned according to the tuning expert's advice.
- The system must respond to keyboard input by playing transposed samples, as in a traditional sampler. The response time must be comparable to a modern software synthesizer (max. 20 msec)
 - The system should respond "gracefully" to confusing input, probably sticking to equal-temperament until it has a pretty good confidence level about the current tonal center.
 - There must be user controls over the nature of the tuning rules and the range of the de-tuning effect.
 - Special inputs from the keyboard will allow the user to specify a tonal center on-the-fly
 - The initial sample set will be one of several grand pianos (Bösendorfer, Steinway, Forester, etc.)

(more to come here)

Design Description

As shown in Figure 1 below, the RMS system has several components spanning from MIDI note input to synthesized sound output. The top and bottom rows of the figure comprise a standard software MIDI sampler; MIDI input is used by the sampler program to select and process sound files in the sample library. The 3 boxes in the middle are the RMS-specific expert systems.

Chord Expert: The first of these is the chord expert; it continuously monitors the MIDI input, and, given a group of MIDI notes that arrive within a few 10s of milliseconds of one another, it uses a simple set of rules to guess what chord is being played. It needs to allow for ambiguity, both in the timing (e.g., heavily arpeggiated chords), and in the note values (e.g., non-chord groups of notes). The output of the chord expert is a real-time stream of chord guesses with associated confidence values. The chord expert is relatively simple and does not require much memory of previous notes/chords played.

Key Expert: The second stage of RMS is the key expert, it reads both the MIDI input and the output of the chord expert and determines if they indicate that the user is playing in a consistent key or tonal center. This is a bit more complex than the chord identification task, and also requires a longer-term memory of previous chords and key guesses. As illustrated in the Figure, the key rules are separate from the chord rules. the key expert generates a stream of key guesses and confidence values.

Tuning Expert: The third expert system component is the tuning expert, which takes the raw MIDI data stream and the chord and key results (and confidence values) and generates a stream of micro-tonally detuned MIDI messages as its output. The tuning rules adjust the MIDI pitches to achieve the desired near-just-intoned tuning of chords and melodic notes. As above, its rules are quite different from those used by the other two experts. For the development phase, we will require a special tool for training and adjusting the tuning expert system's rules; a simplified version of this (possibly consisting of a single "weight" slider) might be incorporated in the final product.

Sampler: The sampler in RMS is like a standard software sampler; it loads a bank of sounds (e.g., sound files with single piano notes spaced a minor 3rd apart played at 4 or 5 different volumes) and can transpose them (within a limited range) and sum several notes together to generate an audio output buffer that is sent to the output driver layer. The difference between the RMS sampler and the standard one is that RMS understands extended MIDI messages to allow precise micro-tonal playback (as specified by the tuning expert). The format of these extended messages might specify the desired frequency in Hz, as a MIDI note number with a fractional value in cents, or as a transposition ratio. The output stage is a typical call-back-oriented audio player function that mixes the processed samples of all currently active notes.

Development Plan

Expert Systems: For development and testing (esp. of the analysis expert systems), a flexible development/delivery system is desired. The final product must however run with very high performance. For this reason, the expert systems are currently being prototyped in the Siren Smalltalk system (<http://FASTLabInc.com/Siren>). Once the rules and their weightings are stable, we can translate the rules to LISP for delivery using the CLIPS (C Language Integrated Production System) expert system shell (<http://www.ghg.net/clips>). (We have done this process before in the development of the Expert Mastering Assistant, EMA [http://FASTLabInc.com/EMA_TechDoc.pdf].)

Sampler: A micro-tonal piano sampler based on the CSL (<http://FASTLabInc.com/CSL>) sound file playback class is currently under development for delivery to the development team later this month. This will allow us to deliver the test system on both Mac OS and MS-Windows platforms using PortMIDI API for

input and CoreAudio (Mac OS) or PortAudio (MS-Windows) for output. I am currently using the East/West Bösendorfer piano samples (<http://www.eastwestsamples.com>) with 5 layers of dynamics.

Development Tools: Simplified versions of the chord and key experts are currently running in Siren/Smalltalk based on a rule-based expert system engine developed for a previous project. A GUI-based Smalltalk rule editor will be provided to the developers so that they can test the system and add to the tuning rules. This editor is currently under design for delivery to the development team later this month.

System Diagram

See attached figure

Timeline

- July, 2007 - Initial prototype sampler (in CSL) and expert systems (in Siren); tuning rule development system
- August, 2007 - Tuning expert rule development (continues until November, 2007)
- September, 2007 - Sampler refinement (add other sample formats and libraries)
- October, 2007 - Port expert system rules to CLIPS in C
- November, 2007 - Testing with tweaked tuning rules and Revilo keyboard prototype
- December, 2007 - Decision about final packaging (whether to use an off-the-shelf sampler)

DRAFT FOR COMMENT

RMS: Revilo Micro-tuned Sampler Overview

