ChordEase: A MIDI remapper for intuitive performance of non-modal music

Chris Korda
whorld.org
80 Avon Street
Somerville, MA 02143
chordease@gmail.com

ABSTRACT
Improvising to non-modal chord progressions such as those found in jazz necessitates switching between the different scales implied by each chord. This work attempted to simplify improvisation by delegating the process of switching scales to a computer. An open-source software MIDI remapper called ChordEase was developed that dynamically alters the pitch of notes, in order to fit them to the chord scales of a predetermined song. ChordEase modifies the behavior of ordinary MIDI instruments, giving them new interfaces that permit non-modal music to be approached as if it were modal. Multiple instruments can be remapped simultaneously, using a variety of mapping functions, each optimized for a particular musical role. Harmonization and orchestration can also be automated. By facilitating the selection of scale tones, ChordEase enables performers to focus on other aspects of improvisation, and thus creates new possibilities for musical expression.

Author Keywords
Mapping, chords, jazz, improvisation, note correction, non-modal, chord progression, digital musical instrument.

ACM Classification
H.5.2 [Information Interfaces and Presentation] User Interfaces, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing–Methodologies and techniques.

1. INTRODUCTION
We sought to simplify the process of improvising to non-modal chord progressions. This led us to develop a software MIDI remapper called ChordEase that repurposes ordinary MIDI instruments by giving them new interfaces. These interfaces eliminate the need to play a different scale for each chord, the related need to play accidentals, and more generally the need to remember or even know the chord progression. ChordEase alters the pitches of notes in real time while preserving their rhythm and dynamics. Control over pitch is partially sacrificed in exchange for the freedom to approach non-modal music as if it were modal. By delegating the more computational aspects of improvisation to ChordEase, users are able to focus on other aspects, such as rhythm, dynamics, feel, group interaction, and aesthetics, even at fast tempos.

2. PRIOR WORK
Note correction schemes have been proposed before, for example a recent invention by Kinter [1] puts "an electronic MIDI keyboard in perfect pitch," while the Hotz Translator [2] alters note pitches "so that only correct values ... are available." ChordEase integrates a novel combination of schemes, and does so within the context of an open-source project in order to encourage collaboration. It also lowers the barriers to entry by requiring only ordinary MIDI instruments and a laptop, unlike most existing solutions which are embedded in specialized hardware [3][4]. Though a few existing solutions permit the chord progression to be sequenced [5], most expect the user to perform it in real time, typically by selecting from a palette of chords. ChordEase is designed to sequence chords, in part because this eliminates the need to know the chord progression, but also because features that require look-ahead, such as harmonic anticipation and bass approach, are easier to implement when the chords are predetermined. ChordEase permits manual selection of chords, but its look-ahead features are inoperative in this case. Finally, unlike most existing solutions, ChordEase supports multiple users, each having their own settings but following the same chord changes, so that an ensemble can be remapped by a single instance of the application.

It's instructive to compare ChordEase with Jean Haury's metapiano [6]. A metapiano user can perform a classical score without knowing its pitches, and similarly a ChordEase user can improvise to a song without knowing its chord progression or the scales those chords imply. Both systems reduce the amount of knowledge required to play music. In both cases the user delegates control over pitch, while retaining control over rhythm and dynamics. But unlike the metapiano, ChordEase doesn't restrict the user to a predetermined sequence of notes; instead it restricts the user to a pre-determined sequence of *scales*. The notes of each scale can be played in any order.

ChordEase can also be compared with Musician Maker [7], in which "players are limited to notes which fit the chord progression", so as to "allow novice players the opportunity to create expressive improvisational music." Musician Maker shares the goal of delegating "tedious difficulties" to a computer, and supports multiple users, but unlike ChordEase, it depends on unique custom controllers for input, and restricts users to chord tones rather than scales.

According to the mapping strategy analysis of Wanderley et al. [8], the mappings used in ChordEase are *indirect*, because multiple inputs determine the pitch of an output note. The chord progression and current position within it, the choice of mapping function and the function's parameters all combine with the pitch of the input note to determine the corresponding output note's pitch. Virtuosos may find it disconcerting that a given gesture doesn't always produce the same pitches. On the other hand, musical vocabulary is acquired more easily, because it's no longer necessary to learn it in all twelve keys. ChordEase lightens the performer's information processing load by embedding harmonic rules, but practice still makes perfect. Playing ChordEase is *skill-based* in the sense of “requiring constant performer input to sustain musical output” [9].

3. DESIGN

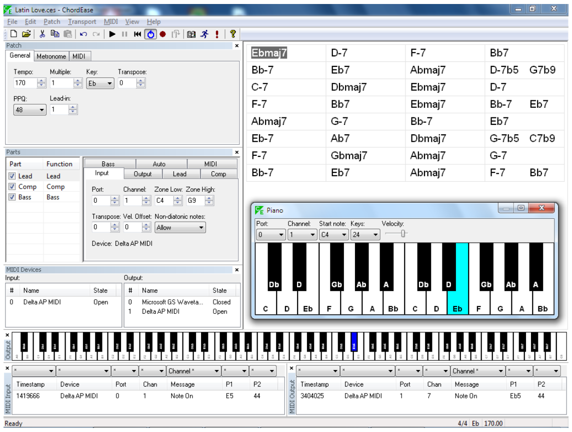

Figure 1: ChordEase screen shot.

ChordEase remaps input notes so that their corresponding output notes fit the chord progression of a predetermined song. Figure 1 shows the song in the upper right corner. The song consists of chord symbols, which are associated with scales, modes, and chord voicings by a user-customizable chord dictionary. Songs can be created and edited within the user interface or via a text editor, or imported from Impro-Visor [10].

Regardless of how the song modulates or what scales it uses, it can be played as if it were confined to a single major key. The input key is configurable, though herein we assume the key of C. In the default case, white keys map to scale tones, whereas black keys map to either non-scale tones or duplicate scale tones. The user synchronizes with the song by following a built-in metronome via headphones. Due to its efficient multi-threaded design, a single instance of ChordEase can support many users simultaneously with minimal latency. For optimal throughput on a multi-core CPU, each instrument should have its own MIDI interface. A typical multi-user hardware configuration is shown in Figure 2.

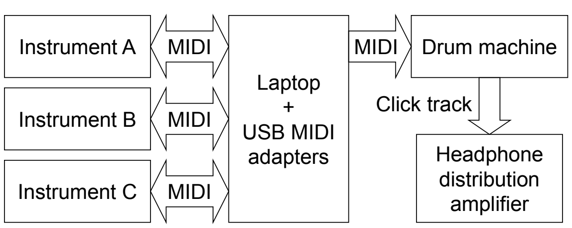

Figure 2: A multi-user ChordEase setup.

4. MAPPING FUNCTIONS

ChordEase supports three mapping functions, each of which facilitates a specific musical role. Lead is intended for soloing, Bass for bass playing, and Comp for chordal accompaniment.

The Lead function transposes input notes in position, by adding accidentals as needed. The input notes are modified as little as possible in order to fit the chord progression. This function is useful for soloing because unlike transposition by shifting, it avoids pitch leaps at chord change boundaries, and thereby preserves the input’s melodic shape across chord changes.

The Bass function transposes input notes by shifting, so that each note of the input diatonic scale has a fixed harmonic role. C is always the root of the current chord, E is always the third, G is always the fifth, etc. This facilitates bass playing because chord tones, particularly the root, remain stationary on the instrument and can be located trivially. The Bass function also supports an approach mode that lets any chord be approached as if it were C. Approach mode shifts the current chord scale so that C maps to the root of the target chord, or as close as possible. Unlike the Lead function, the Bass function may produce pitch leaps at chord boundaries, as shown in Table 1.

Table 1: Comparison of Lead and Bass functions.

Input		C4	D4	E4	F4	G4	A4	B4
Am7	Lead	C4	D4	E4	F#4	G4	A4	B4
	Bass	A3	B3	C4	D4	E4	F#4	G4
Fm7	Lead	C4	D4	Eb4	F4	G4	Ab4	Bb4
	Bass	F3	G3	Ab3	Bb3	C4	D4	Eb4

In the Comp function, a single input note triggers a chord, with user control over voicings, variations, and arpeggiation. The algorithm resembles the one proposed by R. Dias et al. [11] in that higher notes produce higher chords. Specifically, the input note’s pitch determines the bottom note of a window within which the output chord is voiced. The chord is inverted as needed to keep it within the specified window. Voicing options include close, drop 2, drop 3, etc. Two customizable variations are defined for each supported chord type. The first variation uses a conventional spelling, whereas the second uses an altered spelling. The output automatically alternates between the two chord variations. The arpeggio speed and order are dynamically variable, facilitating emulation of strumming and finger-picking.

5. ADDITIONAL FEATURES

Additional features essential to ChordEase are explained below. Other useful but non-essential features include recording of the output, synchronization with external sequencers, and remote control of all parameters via MIDI.

In order to support multiple users simultaneously, ChordEase permits the creation of any number of parts, each having its own independent state. A part is a data structure that specifies how to map input notes to output notes, and includes many settings such as MIDI port and channel selections, which mapping function to use, and mapping function parameters. A single user can also play multiple parts at once, either by splitting an instrument into zones that control different parts, or by layering parts together. Melodic lines can be orchestrated by layering Lead and Comp.

In an ensemble, one part may intentionally change chords early, relative to other parts. This is harmonic anticipation, and ChordEase allows it to be specified differently for each part. Positive anticipation creates a time window before each chord change during which the part’s input notes are mapped to the subsequent chord’s scale. Negative harmonic anticipation is also permitted, in which case the part lags behind the changes.

If a note is sustained over a chord change, the note may become invalid, i.e. it may not be a member of the new chord’s scale. ChordEase can optionally correct held notes, by substituting notes from the new chord’s scale. The Lead and Bass functions substitute the nearest scale tone, whereas the Comp function substitutes the nearest chord tone. If a Comp chord is held over multiple chord changes, cumulative corrections progressively alter the original chord. Harmonic anticipation can prevent held note corrections, because notes struck within the harmonic anticipation window are mapped using the anticipated chord's scale, and are thus axiomatically valid when the chord arrives.

ChordEase includes a harmonizer that automatically generates a harmony line for any part. The user specifies a generic interval, which is then altered as needed to fit the chord changes. The harmonizer also supports static harmony, implemented as follows: the previous harmony note is compared to each subsequent melody note, and as long as the interval between them stays within a specified range, the previous harmony note is reused. Multi-part harmony is also achievable by layering parts that have different harmony settings.

6. NON-DIATONIC INPUT

In its usual configuration, ChordEase maps the white keys to chord scale tones. But what happens if a black key is struck? The result depends on which non-diatonic rule is in effect, as shown in Figure 3 and explained below.

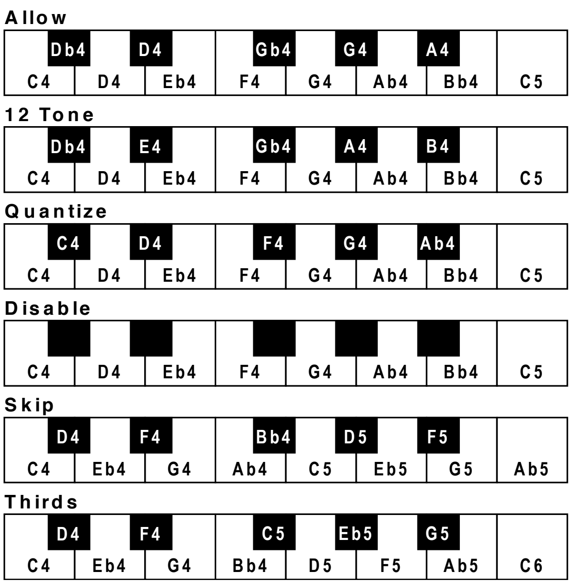

Figure 3: Non-diatonic rules for Lead over F Dorian.

The default rule is Allow, which maps each black key to the scale tone of the white key above it, minus a half step. The result is either a non-scale tone or a duplicate note, depending on the interval between the adjacent scale tones.

The Twelve-Tone rule is similar to Allow, but eliminates the duplicate notes. The entire chromatic scale is made available, but at the cost of presenting the notes slightly out of order. The black keys map to the five unique non-scale tones.

The Quantize rule rounds to the nearest scale tone, so that all black keys generate duplicate notes. Only scale tones are accessible, making the instrument accident-proof and more suitable for gestural playing. The Disable rule similarly restricts the user to scale tones, by disabling the black keys.

The Skip rule compresses the diatonic scale onto the chromatic scale, so that all keys are active and output unique scale tones. This is useful when inputting notes via a continuous controller, theremin, or similar device, because it ensures that the scale tones are linearly mapped to controller coordinates, i.e. any two adjacent scale tones are always separated by the same increment of physical motion. Interval distances are greatly reduced, e.g. playing a fifth yields an octave. The mapping is aligned so that middle C (C4) retains its usual pitch, but all other notes are shifted.

Finally, the Thirds rule maps input notes to output notes in such a way that successive white keys form the cycle of thirds. This makes it much easier to play arpeggios. Each black key outputs the scale tone that the adjacent white keys step over. Interval distances are again greatly reduced: each input octave spans two output octaves.

7. CONCLUSION

We sought to facilitate the performance of improvisational non-modal music, particularly jazz, by delegating music theory calculations to a computer. We implemented several alternative interfaces for ordinary MIDI instruments, each optimized for a different musical role. These interfaces create new possibilities for musical expression, and make it easy to play parts that would otherwise be demanding or impossible. On the other hand they make common tones harder to locate. It might be helpful to indicate common tones visually. We plan to develop additional mapping functions in the future, including melodic vocabulary and palettes of relative scale tones. We hope that ChordEase proves to be a useful platform for further experimentation as well as pedagogy, and speculate that it could be of particular interest to people with disabilities.

8. ACKNOWLEDGMENTS

Thanks to Kevin Roche, Robert Kimberk, Elisabeth Shaw, Josh Rosenstock, V.J. Manzo, Greg Waltzer, Richard Clements, Douglas Ruuska, Todd Winkler, and the NIME reviewers for their comments and suggestions.

9. REFERENCES

[1] Z. C. Kinter. MIDI re-mapping process for utilizing proper music theory when playing a keyboard. US8642875. 2014.

[2] J. Hotz. MIDI musical translator. US5099738. 1992.

[3] G. Johnson. Dynamic chord interval and quality modification keyboard, chord board CX10. US5440071. 1995.

[4] W. H. Sim. System to enable the use of white keys of musical keyboards for scales. US20040231500. 2004.

[5] K. Ueki et al. Assistive apparatus, method and computer program for playing music. US20040112203. 2004.

[6] J. Haury. La pianotechnie ou notage des partitions musicales pour une interpretation immediate sur le metapiano. In *Proc. of the 14th Journees d'Informatique Musicale*. 2009.

[7] J. Buschert. Musician Maker: Play expressive music without practice. In *Proc. of New Interfaces for Musical Expression*, 2012.

[8] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *Proc. of New Interfaces for Musical Expression*, 2002, 1-6.

[9] J. Malloch, D. Birnbaum, E. Sinyor, and M. M. Wanderley. Towards a new conceptual framework for digital musical instruments. In *Proc. of the 9th International Conference on Digital Audio Effects*, 2006, 49-52.

[10] R. Keller, A. Schofield, A. Toman-Yih, Z. Merritt, and J. Elliott. Automating the explanation of jazz chord progressions using idiomatic analysis. *Computer Music Journal*, 37 (4), 2013, 54-69.

[11] R. Dias, C. Guedes, and T. Marques. A computer-mediated Interface for Jazz Piano Comping. In *Proc. of ICMC/SMC*, 2014, 558-564.

10. Appendices

The ChordEase screen shot, documentation and source code are available, along with demo and instructional videos.