“All of the Rules of Jazz”: Stylistic Models and Algorithmic Creativity in Human-Computer Improvisation

Brian A. Miller

NOTE: The examples for the (text-only) PDF version of this item are available online at: https://www.mtosmt.org/issues/mto.20.26.3/mto.20.26.3.miller.php

KEYWORDS: improvisation, jazz, corpus studies, computer music, style, George Lewis

ABSTRACT: Though improvising computer systems are hardly new, jazz has recently become the focus of a number of novel computer music projects aimed at convincingly improvising alongside humans, with a particular focus on the use of machine learning to imitate human styles. The attempt to implement a sort of Turing test for jazz, and interest from organizations like DARPA in the results, raises important questions about the nature of improvisation and musical style, but also about the ways jazz comes popularly to stand for such broad concepts as “conversation” or “democracy.” This essay explores these questions by considering robots that play straight-ahead neoclassical jazz alongside George Lewis’s free-improvising Voyager system, reading the technical details of such projects in terms of the ways they theorize the recognition and production of style, but also in terms of the political implications of human-computer musicking in an age of algorithmic surveillance and big data.

Received May 2019

Volume 26, Number 3, September 2020
Copyright © 2020 Society for Music Theory

[0.1] In 2016, the Neukom Institute for Computational Science at Dartmouth College began hosting the “Turing Tests in Creative Arts,” a set of yearly competitions in music and literature intended to determine whether people can distinguish between human and algorithmic creativity. The musical categories for the most recent contest in 2018 include “Musical Style or Free Composition” and “Improvisation with a Human Performer.” In the former, computational systems have to generate music in the style of Charlie Parker given a lead sheet, Bach’s chorales given a soprano line, electroacoustic music given some source sound, or “free composition,” apparently also in a given style. The improvisation challenge tests a system’s musicality and interactivity with a human collaborator in either jazz or free composition. And while few can claim to have won prizes for passing it, the Turing test is often invoked by researchers developing algorithmic musical systems. The Flow Machines project at the Sony Computer Science Laboratories, led by François Pachet until his recent departure for Spotify, touts its Continuator (which uses a variable-order Markov model to improvise in the style of a human pianist by way of a call-and-response exchange) as having passed the Turing test, and press coverage often invokes the term when discussing a more
recent project from the same team aimed at generating pop songs (Jordan 2017). Similarly, computer scientist Donya Quick’s “Kulitta” system has garnered headlines like, “If There Was a Turing Test for Music Artificial Intelligence, ‘Kulitta’ Might Pass It” (Synthtopia 2015); other recent online articles have asked, “Can We Make a Musical Turing Test?” (Hornigold 2018) and answered, “A New AI Can Write Music as Well as a Human Composer” (Kaleagasi 2017).

[0.2] Many of these articles appear on sites with names like “SingularityHub” and “Futurism”; often, the Turing test is less a measure of actual computational achievement than a marker for a certain kind of popular techno-optimist (even if cynical) view of artificial intelligence and computation in general. Indeed, various scholars have argued that the test itself is widely misunderstood, owing not least to Turing himself, who begins the paper that introduces the “imitation game” with the provocation: “I propose to consider the question, ‘Can machines think?’” (1950, 433). But he immediately backtracks, arguing that the question as posed is untenable, and goes on to suggest the game itself as a “closely related” replacement. In the game, a man and a woman are located in one room, and an interrogator in another; the latter asks questions of both the man and the woman in order to identify which is which, where the woman answers truthfully and the man tries to cause the interrogator to choose incorrectly. The question now, rather than “Can machines think?” is “What will happen when a machine takes the part of [the man] in this game?” (Turing 1950, 434). Popular accounts of the test almost never account for two related aspects of the game, namely the inclusion of a gendered component and the doubled form of imitation involved, in which a computer imitates a man imitating a woman. Though the test is almost always understood—even in many scholarly accounts—as a matter of making a choice between “machine or human,” Turing gives no clear indication that the addition of the machine changes the interrogator’s options from “man or woman.” Thus the “imitation” in the game is not directly of human thought by mechanical means, but rather of human imitative abilities themselves—the imitation of imitation.

[0.3] Turing’s work on artificial intelligence is also inseparable from his codebreaking work for the British military during the Second World War, and a parallel conjuncture manifests itself today, perhaps surprisingly, in musical terms. Beginning in 2015, the United States Defense Advanced Research Projects Agency (DARPA) began funding a project called Musical Interactive Collaborative Agent (MUSICA). The project is part of DARPA’s Communicating with Computers program, which “aims to enable symmetric communication between people and computers in which machines are not merely receivers of instructions but collaborators, able to harness a full range of natural modes including language, gesture and facial or other expressions.” Apparently music is one such mode: the end goal of MUSICA is to produce a jazz-playing robot capable of performing convincingly with human collaborators. One of the project’s directors, Kelland Thomas, suggests that “jazz and improvisation in music represent a pinnacle of human intellectual and mental achievement” (quoted in Del Prado 2015; see Chella and Manzotti 2012 for a similar argument and an explicit proposal for a jazz Turing test). And while DARPA is famous for funding unorthodox, long-shot projects, the clear implication is that jazz improvisation is so paradigmatically representative of more general modes of human interaction that its technological replication would have some kind of military value going beyond its intellectual or aesthetic meaning. Though little detailed information on the project is publicly available, MUSICA is based in large part on machine learning techniques—advances in computational capabilities since Turing’s time that I will return to in some detail below. According to Thomas: “We’re going to build a database of musical transcription: every Miles Davis solo and every Louis Armstrong solo we’re going to hand-curate. We’re going to develop machine learning techniques to analyze these solos and find deeper relationships between the notes and the harmonies, and that will inform the system—that’ll be the knowledge base” (quoted in Thielman 2015). Though the broader claim—linking jazz to conversation in natural language and suggesting that modeling the former computationally is the best way to learn anything useful about the latter—evokes difficult questions about the relation between music and language, in its actual implementation MUSICA is more immediately concerned with questions of musical style. While the project’s few public statements never define jazz explicitly, it appears that what is at issue is a very specific, stereotypical view: small- to medium-sized jazz combos playing standards in a relatively conventional format; in other words, the neoclassical style associated with conservative institutions
like Jazz at Lincoln Center (see Chapman 2018). While the machine learning model is intended to capture the characteristic ways players like Armstrong and Davis form musical utterances, it is far from clear exactly how the system would reconcile such varied styles as Armstrong’s 1920s New Orleans sound and Davis’s “electric” work from the 1970s, or even to what extent the project recognizes such differences as relevant for musical interaction.

This article examines several different approaches to computational improvisation, all in the orbit of jazz but implementing two very different styles. While little information and no technical details about MUSICA are publicly available, another project, from the Robotic Musicianship Group at Georgia Tech’s Center for Music Technology, takes a similar approach to robotic jazz and has published a number of papers focused on the project’s technical aspects as well as many publicly available performance videos. This robot, named Shimon, plays the marimba alongside humans in a traditional jazz combo based on a conventional understanding of key, harmony, and form, but with a complex machine learning-based model for generating solos. I compare Shimon to a computer program called Impro-Visor (Gillick, Tang, and Keller 2010), which does not perform in real time but which generates solos in a similar style using a different corpus-based machine learning model, and I contrast both of these systems with George Lewis’s Voyager, a long-standing project that stems from Lewis’s work in free improvisation.

The juxtaposition has a dual focus: first, how do these computational approaches to improvisation handle the challenges of imitating human musical styles, and how is style itself theorized both implicitly and explicitly? In other words, how do the features and affordances of computation become musical in relation to such varied human improvisatory practices? Because all of these systems change frequently (for example, Voyager having been updated over the course of several decades, and Shimon having multiple modes of operation along with various upgrades), my account is not necessarily concerned with capturing any system’s exact functioning in any particular performance, nor am I interested in determining what the “best” computational implementation of jazz or free improvisation might be. Instead, for each system, I read the available technical details, however partial, for what they reveal about the theories of musical style embedded and embodied in the computational implementation. I also read these implicit theories with and against the creators’ own statements about style, and situate them within the broader frame of Leonard Meyer’s theory of musical style, which, I will argue, has its own computational entailments.

Second, what are the politics, broadly speaking, of the algorithmic choices revealed by this stylistic analysis? While DARPA’s interest in improvising machines provides one obvious point of reference for this question, I am also interested more generally in the growing role of algorithms in the organization of both research and social life, as machine learning and “Big Data” become increasingly ubiquitous. Notably, anthropologist Eitan Wilf (2013a, 2013b, 2014) has drawn on algorithmic musical improvisation to argue for a computationally informed rethinking of classic anthropological notions of culture itself, and I critique and extend his argument in light of my analysis of Shimon and my reading of Meyer. I also draw on media scholar John Cheney-Lippold’s (2011, 2017) theorization of “algorithmic identity” to argue that similar processes are at work in the algorithmic imitation of human musical styles and the production of data-driven categories and profiles in such “Big Data” practices as targeted advertising.

Finally, music theorists, though lacking the Department of Defense’s financial resources, have also developed considerable interest in the relation between human musical abilities and computation. Writing in support of computational approaches to music theory, David Temperley suggests that “creating a computational model that performs [a musical process] does not prove that humans perform it in the same way; but it satisfies one important requirement for such a model” (2007, 5–6). This notion might be understood as a particular kind of interpretation of the Turing test: one in which computers not only display convincing behaviors but actually produce those behaviors in ways that mirror or at least reflect meaningfully on human cognition. While I am not directly engaged with what might be called mainstream computational music theory here—consisting of practices of corpus studies and the often-associated study of music perception and cognition—my discussion should not be read as entirely detached from that area of inquiry,
especially given the corpus-driven nature of some of the projects considered here. As the fruits of corpus analysis, musical performances with computers might in fact be understood themselves as arguments about music psychology, and as arguments about the corpora on which they depend. (11) In any case, computational music theory and algorithmic improvisation are related by a broader set of algorithmic concerns that have not been properly adumbrated, and thus considerations of one will also have implications for the other. As media scholar Adrian Mackenzie puts it, “machine learning is said to transform the nature of knowledge. Might it transform the practice of critical thought?” (2017, 2).

1. Shimon

[1.1] The Robotic Musicianship Group at Georgia Tech’s Center for Music Technology has produced a number of robot musicians and musical prostheses, but the robot Shimon is the most complex, playing the marimba alongside human jazz musicians, a single cycloptic camera looming over the keyboard and bobbing in time with the music, swiveling to focus on its collaborators as the musical focus shifts; the project’s publications, in fact, stress its accomplishments in robotics more than its musical features. Video Example 1 provides a glimpse of the robot’s capabilities as of late 2018, showing two solos excerpted from a performance of Thelonius Monk’s “Blue Monk” as part of a robotic musicianship exhibition at the Museum Speelklok in Utrecht. Shimon’s underlying model is similar to MUSICA’s, mixing preprogrammed formal musical knowledge with machine learning. According to the director of the GTCMT and project leader Gil Weinberg, “We gave the robot chord progressions, we taught him all of the rules of jazz based on machine learning and rules” (UN Web TV 2018). Notably, if MUSICA seems destined for combat, Shimon has also done explicitly political work: the robot recently appeared at the United Nations (U.N.) in a program called “Jazz Democracy,” which presented a “forum/performance to showcase how jazz has become global and has embraced the broad democratic values that were first established in ancient Greece” (UN Web TV 2018). (12) Commenting on a video clip of Shimon trading solos with a human drummer, Weinberg celebrates the fact that “together they come up with something out of this back and forth, a little contentious but enough to create, maybe, democracy, peace and harmony.” This section examines Shimon’s musical workings in as much detail as is possible based on the group’s publications; I also introduce a particular reading of Leonard Meyer’s theory of musical style, which guides my discussions of style in the rest of the article.

Producing Computational Jazz

[1.2] Shimon’s jazz abilities are built on a machine learning model derived from a corpus of solo transcriptions; the specific tunes and performances involved are unspecified, though the authors mention that the robot is trained to play like John Coltrane and Thelonius Monk (Nikolaidis and Weinberg 2010, 716). As Example 1 shows, Shimon’s performance decisions are based on a mix of human input—what it hears from its collaborators—and these preexisting models, where performed notes also feed back as input, themselves “heard” in real time.

[1.3] Specifically, the robot implements a statistical technique called a hidden Markov model, or HMM, often used for tasks like translation or other kinds of language processing. In musical settings, HMMs have previously been used to model harmonic function (White and Quinn 2018), among other things, and that particular application presents an intuitive example for understanding the model’s basic principles. In general, a theory of harmonic function assumes that a segment of music may be described by a background state—say, tonic, predominant, or dominant—where the surface manifestation of that state can vary. Stereotypically, tonic may be expressed by I or VI (sometimes III), dominant by V and VII, and predominant by II and IV. Within each harmonic state, some chords are more likely to appear than others: I may be more likely than VI, V more likely than VII, and so on. Each harmonic function also has characteristic behaviors: tonic may move to either dominant or predominant, while dominant is by far more likely to move to tonic than to predominant, and so on. Similarly, an HMM consists of background states that emit surface tokens with varying frequencies. Example 2 presents an abstract HMM, where the states (labeled S) emit different tokens (labeled T). The states themselves change according to a separate set of probabilities, so that S1 moves to either S2 or S3 with equal likelihood but S2 is more likely to move to S3 than to S1, and S3 always moves to S1. To generate a set of surface tokens
using the model in Example 2, one would (1) begin in any background state; (2) choose a surface token at random based on the output, or “emission,” probabilities (the solid vertical arrows connecting the states to the tokens); (3) choose a new state at random based on the “transition” probabilities (the dotted lines connecting the states to each other); (4) return to step 2, and so on. Replace the state labels with harmonic functions T, P, and D and the tokens with Roman numerals I, VI, II, IV, V, and VII and the analogy with harmonic function is clear.

[1.4] The process for building (or training) an HMM from a dataset is too complex to examine in any detail here. Essentially, the training algorithm begins with random probabilities and tests how “wrong” the model is when tested against the actual data; the probabilities are adjusted iteratively until the model’s performance reaches a plateau. The end result is that any sequence of data from the relevant domain should be able to trace an appropriate path through the model, so that common events (like a I–IV–V progression) would be assigned a high probability and less common events (like, say, a deceptive cadence) would be assigned a lower probability relative to their frequency in the corpus.

[1.5] In Shimon’s model, the corpus is analyzed chord by chord, with every chord reduced to its harmonic function in the key of the song, and the transcribed pitches, now represented as scale degrees, are fed into separate HMMs for pitch and duration. The resulting models—now understood explicitly as “styles”—drive the robot’s real-time performance, which involves analyzing incoming notes for both harmonic and melodic content and matching the resulting patterns to sequences from the HMMs. Shimon determines musical key using Carol Krumhansl’s (1990) well-known pitch profile method, matching the frequencies of observed notes to existing scale degree profiles for major and minor keys. Harmony is determined by a similar method, with recently detected pitches being scored against prototypes for major, minor, and diminished chords (Nikolaidis and Weinberg 2010, 714). (The published materials mention only these chord qualities, though it seems likely, or maybe necessary, that various kinds of seventh chords are included as well.)

[1.6] Notes are generated in performance by retrieving a corpus-derived model for the chord thus determined and then searching for the sequence of recent notes; the model will output a new note based on the current state’s probability distribution for the given input sequence. Shimon uses a variable-order HMM (VOHMM), which allows the model to change how many previous events determine the probability of the next one. For example, a sequence C–D–E would prompt searches for E, D–E, and C–D–E, and for any one of these searches there could be multiple possible subsequent notes with different probabilities depending on their frequencies in the corpus. (C–D–E)–G might be most likely, followed by (C–D–E)–F; options like (C–D–E)–B♭ might be more likely than (C–D–E)–A if the chordal context is C7, or vice versa given Am7. In general, higher-order Markov models produce more predictable output by replicating longer strings, while lower-order ones are more “surprising.” Crucially, while such a model seems to imitate how humans might make decisions about similar processes in a general sense—choosing the next thing to play/say/do in terms of what has come before—a Markov model’s strict linearity and lack of long-term memory or forward planning make it less a model of human expression than a very particular, and very non-human, set of constraints on the kind of music that can be generated.

[1.7] Above the level of the specific corpus models, another notable aspect of Shimon’s style is its unwavering focus on individual harmonies as the background over which musical material is generated. The robot perceives individual pitches in real time, resolving those notes into harmonies as they match different harmonic prototypes. But the robot’s repertoire of licks is strictly walled off at the point of harmonic change; as the developers note, the model for each chord type is stored in a separate file, which means that no sequence is explicitly understood to accompany a succession of more than one chord, even though jazz solos are often conceptualized in relation to multi-chord sequences, like the standard II–V–I progression. That is not to suggest that the robot could not produce sequences that sound like conventional melodic patterns associated with II–V–I progressions, but such patterns would only appear in spite of the model’s lack of knowledge of chord-to-chord successions. This is a common desideratum for machine learning in general: a system without specific knowledge of real-world processes nevertheless produces them by force of
accumulated data. This is the case, for example, with neural networks that produce intelligible images of human faces without having been explicitly programmed to understand their component parts: eyes, mouths, noses, and so forth.

[1.8] Finally, one of Shimon’s most intriguing features allows its corpus-derived styles to be blended, both with each other and with the style of the human performing in real time, by way of a smartphone app with sliders with labels like “Coltrane,” “Monk,” and “Me” (Nikolaidis and Weinberg 2010, 716). On the technical level, such a blending would mean allowing the algorithm to choose new notes based on probabilities drawn from any of these individual models, where the choice among models is itself probabilistic. **Example 3** imagines some possible transition probabilities for the sequence F–A: if Shimon were set to play as 30% Monk and 70% Coltrane, the algorithm might first select one of the two players, then choose a note randomly using the weightings in the table. Other means of choosing may also be possible, as the developers never specify the exact mechanism for blending styles.

[1.9] But what could it really mean, either for a human or a machine, to play 30% like Monk and 70% like Coltrane? Shimon’s creators frame this ambiguity as a feature, suggesting that the strangeness of such a combined style represents something new, a kind of algorithmic creativity, such that a human might be inspired to hear new kinds of relations in the music. As the above discussion should make clear, however, the styles that are being blended are themselves not the “actual” styles of Monk or Coltrane, in the form of the habitual modes of action that those historical individuals employed to make music in their characteristic ways, but rather formal styles resulting from the constraints of the analytical algorithms interacting with the transcriptions that provide the model’s source material. That is not to say that Shimon could not potentially generate something that sounded to a human ear like a mix of Monk and Coltrane. But in general, the model, focused as it is on note-to-note successions and tied to individual harmonies rather than longer harmonic formulas, will never produce certain kinds of stylistic combinations: say, Monk’s characteristic sparse solo lines with Coltrane-style harmonic substitutions, or Coltrane’s “sheet of sound” passages with Monk’s percussive articulation.(16)

**Leonard Meyer’s Selectional Theory of Style**

[1.10] Both the general approach and the specific language used to describe style in the previous section is inspired by Leonard Meyer’s theory of musical style, represented in its most complete form in *Style and Music* (1989). Meyer’s theory unfolds from an initial definition: “Style is a replication of patterning, whether in human behavior or in the artifacts produced by human behavior, that results from a series of choices made within some set of constraints” (1989, 3). Later he describes how that definition would be put into practice: “It is the goal of style analysis to describe the patterning replicated in some group of works, to discover and formulate the rules and strategies that are the basis for such patterning, and to explain in the light of these constraints how the characteristics described are related to one another” (38).

[1.11] Though it aims much more broadly, Meyer’s theory is particularly well suited to describing algorithmic approaches to music, both conceptually and historically. For one thing, Meyer’s work has been extremely influential on more recent developments in corpus studies. But at a more basic level, even in its later form Meyer’s theory shows clear signs of his early engagement with information theory, dating to the 1950s. Information theory models the transmission of signals probabilistically, most prototypically in terms of a Markov chain in which the likelihood of any symbol appearing is based on the symbol or symbols that preceded it (Shannon and Weaver 1949). Whatever Meyer’s intentions in attempting to develop a far-reaching theory of stylistic constraints — psychological, historical, cultural—the basic idea behind his theory seems immanently algorithmic. To capture this sense of style as a branching tree of probabilistic choices, I refer to Meyer’s theory as “selectional,” mirroring language from Claude Shannon’s original account of information theory: “the actual message is one selected from a set of possible messages” (Shannon and Weaver 1949, 3, emphasis in original). Overwriting Meyer’s own term, “choice,” selection draws out an orientation already present in Meyer’s theory towards a computational understanding of style.
The hidden Markov model from the previous section provides a perfect example of selectional style: if the states represent harmonic functions and the tokens represent specific chords, the arrows describing the possible transitions and emissions constitute a set of constraints. Running the model over and over—making choices within the set of constraints—will produce a variety of musical “works” with statistical regularities but that are nonetheless not identical. And if one imagines a population of many such models, all with slightly different weightings, then the musical works produced by each model will bear its particular style, which could be represented directly by the diagram itself, presumably as a stand-in for its characteristic “sound.”

Shimon’s style can be understood in these same terms, but with a more sophisticated model trained on a corpus of jazz solos and relying on additional information, like harmony and meter, that is registered in real time as the robot performs. This is what I mean by a “formal style,” as opposed to the “actual” style of Thelonius Monk or John Coltrane. In short, what seems like merely a model of style can, especially in the context of algorithmic performance and improvisation, be more productively understood as a proper style in Meyer’s sense—as a system of constraints within which choices are made that actually produce music. It is worth noting that Shimon’s particular style is far from the only possible algorithmic approach to the jazz repertory that the robot is designed to play. To illustrate, I turn to another system developed by Jon Gillick, Kevin Tang, and Robert Keller, which I will refer to as Impro-Visor, as it is incorporated into the software of that name produced by Keller, though that program has other features not discussed here (see Gillick, Tang, and Keller 2010).

2. Impro-Visor

While Impro-Visor lacks Shimon’s ability to perform live with humans, it similarly produces jazz solos using a model built from a corpus of transcriptions, with two notable differences: notes are chosen not individually but in small motivic contours; and those groups are not made up of specific pitches but rather more abstract labels designating each pitch’s relationship with the governing harmony. For example, a given motive might be specified as (C L C H C), where C indicates a basic chord tone (root, 3rd, 5th, or 7th), L indicates a “color” tone drawn from the chord’s altered or extended notes (9th, #11, or b13), and H indicates what the authors call a “helpful” tone, which can be drawn from either C or L; another possible category, A, indicates a note that approaches a chord tone by semitone. These abstract melodic fragments can then be instantiated over any harmony, at which point the actual notes to be played are chosen at random from among the eligible pitches, with some consideration given to the intervals between consecutive notes. Example 4 shows one of the authors’ example melodies, repeated twice over different harmonies, which their full notation gives as (Δ -3 -4 C4 H8 H8 C4). The delta symbol (Δ) indicates what the authors call a “slope,” which is used to indicate unidirectional passages with multiple intervallic interpretations; the two numbers after the Δ give the smallest and largest allowed intervals. In this case, (Δ -3 -4) indicates that consecutive notes must always move downwards by either three or four semitones. Fully parsed, this sequence would be read as: begin with a chord tone that lasts for one quarter note (first C4); move downwards to a “helpful” note that is a major or minor third away (three or four semitones) for the duration of a single eighth note (first H8), then do so again (second H8); end by playing a chord tone, another third lower, for a quarter note (final C4). Note that both measures of Example 4 begin on a chord tone, but the first measure features the chord’s seventh while the second measure begins on the chord’s third. In the first measure, the “helpful” notes that follow include A and E as the D7 chord’s fifth and sharp ninth, respectively, and then D as a chord tone in G7.

Impro-Visor trains its model by analyzing a corpus made up of transcriptions of jazz solos, broken up into short fragments of preset length (a single measure in the authors’ examples). Using the harmonies given in the tune’s lead sheet, the solo’s fragments are transformed into abstract melodies. Notably, Impro-Visor’s abstract melodies are agnostic regarding changes of harmony. That is, the abstract melody from Example 4 could be implemented over a measure that consists of a single harmony just as easily as it can be applied to the demonstrated two-chord pattern. Once they have been extracted, the complete set of abstract melodies is then grouped
into clusters of “similar” melodies (see Example 5), where similarity is measured in terms of seven parameters:

1. number of notes in the abstract melody,
2. location of the first note that starts within the window,
3. total duration of rests,
4. average maximum slope of ascending or descending groups of notes,
5. whether the window starts on or off the beat,
6. order of the contour (how many times it changes direction), and
7. consonance. (Gillick, Tang, and Keller 2010, 60)(26)

[2.3] Gillick, Tang, and Keller (2010) treat the clusters themselves as states in a Markov model, returning to the full corpus of actual melodic fragments and classifying each by its cluster. They then calculate transition likelihoods among the clusters. This is analogous to the approach to harmonic states in the simple HMM of harmonic function that I described above: one abstract musical state (a harmonic function or cluster of abstract melodies, respectively) has propensities both to transition to other states and to emit more concrete musical objects (chords with specific pitches or abstract melodies). But the melodies emitted at this point are still abstract even after being selected from the background cluster; that is, they still have the form (C8 H8 L1, and so on) and must be converted into actual pitches by choosing randomly from the possible notes given the harmonic and intervallic context.

[2.4] Overall, this approach aims to capture the often-expressed intuition that improvising jazz players rely on a kind of vocabulary of licks from which a given solo is formed (22) Melody takes precedence over harmony, not in the actual pitches played, but at the level of contour. Indeed, where Markov models are most often used to describe musical structure in harmonic terms, Gillick, Tang, and Keller (2010) use such a model to generate melodic contours independent of any harmonic considerations, with the latter incorporated only at the final stage. This is also in contrast to Shimon’s approach, where the underlying harmonies enforce their restrictions before any other melodic decisions are made, because every possible harmony relative to the key has its own Markov model trained from the corpus, and those models themselves are selected based on the harmony that Shimon hears. What Impro-Visor’s example highlights is the contingency of stylistic constraints: computational models themselves constitute systems of constraints that can be mapped onto different musical parameters (e.g., harmony, melody, clusters of melodic contours) singularly or in combination. Moreover, they lend their particular shapes to the corpora on which they are trained; for an improvising robot, a corpus of Coltrane transcriptions does not meaningfully represent his style until it is refracted through an algorithm, whether Impro-Visor’s, Shimon’s, or one entirely different.

3. Stylistic Recognition and Interaction

[3.1] Having considered the ways Shimon and Impro-Visor produce jazz solos, I turn now to the interactive aspect of Shimon’s performance (since Impro-Visor cannot perform in real time), and particularly the process of stylistic recognition, which in improvisation is just as important as the production of style.

[3.2] In the broadest sense, Shimon is a generous listener. The robot can never hear anything as truly unstylistic, because even strings of notes that do not match an existing sequence from the training corpus are simply assimilated into the model in real time. And by making such adjustments to its model, Shimon is constantly modifying the interactional context of its performance. The meaning of a given sequence of notes—in Meyer’s sense, as a choice among many possibilities given some set of constraints—is not fixed, like a grammatical unit in a synchronic formal structure, but rather develops as the probabilities associated with the sequence change. Thus, the style that Shimon recognizes is malleable within the hard constraints imposed by
its perceptual abilities and built-in musical knowledge. What are the implications of an interaction, or “conversation,” with such an agent?

[3.3] It seems that something like this question is what led DARPA to take an interest in jazz, but the organization’s move to consider jazz as conversation is hardly unique. Rather, it is an appropriation of a metaphor common among performers, critics, and scholars, as ethnomusicological work on jazz has long pointed out (see Berliner 1994; Monson 1996). Ingrid Monson (1996) has extended the metaphor to consider the ways that theories from linguistic anthropology dealing with the coherence and processual nature of conversation can shed light on jazz improvisation. Particularly compelling is the concept of metapragmatics, developed by Michael Silverstein (1976; Lucy 1993), which describes the ways that linguistic utterances point not just to particular referents but also to themselves, reflexively establishing the very interactive context in which they take place. Monson is primarily interested in the rhythm section, whose “groove (or rhythmic feel),” she suggests, is “one of the most significant metapragmatic framing devices” for jazz improvisation (1996, 189). To put the concept in semiotic terms, metapragmatics describes a particular kind of indexicality that is pervasively and necessarily present in conversational interactions. A statement that I make can only become part of a conversation if someone else recognizes (correctly or not) that the statement was directed at them—and as a conversation develops, further statements not only point at (index) the interlocutors but also prior and future utterances. (28)

[3.4] Part of what makes Shimon a stylistic imitator, in the sense I previously associated with the Turing test, is that its actual conversational abilities are highly constrained. If the robot can never hear human playing as anything but input for its Markov models—that is, if no statement can ever truly fail to be understood—then there are no meaningful human-computer metapragmatics involved. Shimon could, in the simplest case, contribute to a conversational “context” of sorts by simply playing pre-determined material metronomically, so that a human has no trouble playing along in time. While it seems to go beyond this with its ability to follow a shifting tempo, to some extent much of what is involved in establishing its musical context comes down to the robot’s hardheadedness, its inability not to play along, combined with the human’s willingness to take the attempt at playing with a robot in good faith.

[3.5] Thus, the more interesting interactional questions have to do with Shimon’s stylistic imperatives, and especially its attempt at blending styles. Here, I return to the U.N.’s “Jazz Democracy Forum” event mentioned in the introduction to this article. In his presentation on Shimon, Weinberg’s focus is on the idea that improvising with robots will “inspire us and enhance us,” because machines will create “new ideas based on algorithms that humans will never come up with” (UN Web TV 2018). Thus, beyond merely asserting that two styles have been blended (by whatever mechanism), a successful performance would presumably involve an accompanying human recognizing stylistic blending as such and responding to it (perhaps as a result of “inspiration”) in his or her own playing. (29) For Shimon, then, the basic Turing test is reconfigured, perhaps paradoxically: in addition to imitating human capabilities, the robot must additionally go beyond the human’s limited ability to play in a single style.

[3.6] Lacking testimonials or more extensive recordings, it is difficult to tell exactly to what extent Shimon successfully communicates its blended styles, or influences its human collaborators by presenting them with new, inspiring improvisational options. If these projects were truly successful, we might expect such robots to be making appearances at jazz clubs or in classrooms, not just at science and technology festivals (or the U.N.). But what would be at stake in considering the possibility that Shimon and related projects are, if not unambiguously successful, then at least representative of a contemporary social situation, a form of life that is increasingly subject to a kind of algorithmic blending and mediation?

Culture as Style as Computation

[3.7] Writing about a similar robotic musicianship project, anthropologist Eitan Wilf (2013b, 719) argues that the algorithmic blending of musical styles provides a corrective to the very concept of culture that has been prominent in anthropological thought throughout the discipline’s history—
that is, the understanding of culture as a kind of style, consisting (in terms familiar to music theorists) of choices within a relatively static system of constraints, where each system is more or less coherent and distinct from those of other cultures. Wilf traces this conception of culture as style, which he associates with anthropologists ranging from Franz Boas and Alfred Kroeber to Pierre Bourdieu, among many others, to the notion of style in Western art, and specifically Romantic, organicist notions of style and creativity (Wilf 2013a; see also 2014, 2013b). For Wilf, machine learning algorithms shift the register on which style is produced; the humans designing a robot that can blend styles are engaged in “styling styles,” where this higher-order styling is itself stylized according to material and computational constraints along with the programmers’ own conceptualizations of style. And, importantly, if algorithms can successfully blend what were previously thought to be autonomous, holistic musical styles, then theories of culture must likewise be reconfigured so as to better understand cultural blending more generally—an imperative driven by the increasing influence of algorithms on culture, thanks to the likes of corporations like Google and Facebook. Style, and thus culture, turns out not to be so static or autonomous after all.

[3.8] Thus, just as I have argued that Meyer’s theory of musical style, with its basis in information-theoretic selectional processes, is particularly well-suited to engaging with computational approaches to music, Wilf argues that culture itself demands renewed theorization suited to its newly algorithmic forms. But in the sweep of his claims, and despite his attention to the mediated nature of statistically blended styles, Wilf takes the musical algorithms’ ultimate success for granted, moving too quickly past unresolved musical questions. Given the disciplinary difference involved, some of these issues are understandable, and my critique is not aimed to defeat Wilf’s larger project, with which I am sympathetic (and which is outside the scope of my own project). Rather, it is worth engaging closely with Wilf’s arguments precisely because they represent a serious engagement with the questions at issue in this article.

[3.9] Some of the most pressing issues come to the fore in the final section of Wilf 2013b, which includes responses from several other scholars along with a final response to the responses by Wilf. In one response, Nick Seaver and Tom Boellstorff note that different technologies encourage different notions of style. Certain player pianos, they note, could reproduce fine differences in timing and dynamics, so that “discourse around the player piano construed tempo and dynamic variation as the essence of style” (Seaver 2013, 735; see also Seaver 2011). But Wilf argues back that, “contrary to what [Seaver and Boellstorff’s] comments suggest, such pianos did not reproduce style. They were based in the principle of ‘mechanical fidelity’” (2013b, 737–8). He continues:

Such pianos reenacted specific performances recorded by great performers on a special recording piano. . . . The reenacting piano reenacted these performances by reproducing their qualia (e.g., the intensity of each note played by the performer during the recording). They thus reproduced tokens rather than types. It is the human listener who abstracted types (styles) by listening to these reenacted tokens and perceiving relations between relations between one token’s parts or between a number of tokens of the same type. (Wilf 2013b, 738)

[3.10] Wilf is perhaps right to point out an important distinction between style in composition and performance, at least for certain repertories—a distinction that, for example, Meyer’s theory acknowledges but is ill-equipped to deal with convincingly—but his understanding of style is in the end too limited to a narrow generative conception, in which pitch and rhythm are the only significant parameters. This limitation is deeply embedded in Wilf’s broader argument, because his claim is that modern computational technology is qualitatively different from earlier reproductive technologies specifically because the former reproduces types (or, in Meyer’s terms, systems of constraints themselves) while the latter merely reproduces tokens (individual instantiations of those systems).

[3.11] Wilf’s account further hinges on the idea that, beyond mere reproduction, the blending of style is significant because it represents a qualitatively different kind of stylistic creativity, where algorithms represent “techniques of innovation not within existing rules of transformation but rather through the reconfiguration and transformation of such rules” (Wilf 2013b, 731). This view
of innovation is remarkably close to Meyer’s, from *Style and Music* (Meyer 1989). In Meyer’s terms, innovation can involve either novel instantiations of existing compositional strategies, the development of new strategies altogether, or even a change in stylistic rules, which are the constraints that govern the very space of possible strategies—operating, to appropriate Wilf’s terms, at “a higher level of the reality of style” (Wilf 2013b, 721). But it seems, then, that we do not need algorithmically blended styles in order to conceive of fluid transformations of style by way of reconfigured rules. The history of music is full of self-aware experimentation, in which performers and composers consciously consider the nature and limits of musical language. Meyer’s own theory seems already to provide the view of style that Wilf needs, without worrying about machine learning algorithms at all.

[3.12] But this raises a larger problem, because Meyer’s theory itself is also clearly insufficient as the basis for a critique of the concept of culture that Wilf diagnoses, seemingly correctly, as a legacy of Romantic notions of style. This is the case specifically because Meyer’s theory takes the groupings of works to be subjected to style analysis as either self-evidently given, or based on the whims of the analyst. In an expansive list of possible units for stylistic analysis, he includes the oeuvre of a single composer, works by a group of composers from the same culture or time period, works of a particular genre, and the works of a whole civilization (e.g., the Western common practice, “the cultures of the South Pacific”) (38). Meyer’s theory thus replicates one of the primary features that Wilf wants to excise from anthropological notions of culture—namely, begging the question of the coherence of the archive, whether musical or cultural.

[3.13] This point prompts the return of a central question, both for Wilf and for this article’s investigation: What is really being blended when an algorithmic system mixes two styles? Here, Wilf’s account is particularly helpful because, while Shimon only blends the styles of jazz soloists (at least in the published materials), Wilf describes examples from his fieldwork of a robot that blends jazz and classical styles: in one instance, a human plays Beethoven for a robot trained to respond in a jazz style, and in another both styles are mixed in the robot’s internal settings, akin to Shimon’s ability to blend jazz styles (Wilf 2013b, 726). While I critiqued the idea that Shimon could meaningfully blend Coltrane and Monk above, the idea of blending classical and jazz styles in the same way puts further pressure on the question of what it means to imagine such a situation. Would one expect Beethovenian forms with jazz harmonies? Jazz tunes with Alberti bass accompaniment? There are plenty of examples of composers from the worlds of art music and jazz incorporating elements from seemingly foreign styles, but the kinds of algorithms discussed here would almost certainly produce nothing of the sort. Where Wilf imagines algorithms combining and transforming the rules associated with different styles to produce virtuosic new combinations, the algorithms available today seem only to be able to blend by flattening their disparate inputs into a single format. Rather than negotiating among the rules governing different styles, as understood explicitly or implicitly by composers and performers, these models instead subject all styles to a single rule or set of rules, whether a Markov model, neural network, or other machine learning technique.

[3.14] Media scholar John Cheney-Lippold (2017, 2011) describes a related process that is central to online advertising, among other practices associated with “Big Data.” For example, navigating to my own Google advertising preferences, I am informed that, in Google’s estimation, my algorithmic self is interested in “jazz” and “world music,” among a long list of other things. And while these “interests,” which (following Cheney-Lippold) I place in quotation marks to denote automated classifications, are accurate to a degree, they silo concepts and objects into categories over which I have no control and which my non-algorithmic self would dispute. Why should jazz be considered monolithically, as a single interest? And what about the widely recognized problems, political and conceptual, surrounding the term “world music”? Drawing on Deleuze, Cheney-Lippold notes that the resulting form of “algorithmic identity” splits the real-world individual into an array of “individuals” limited only by the kinds of discriminations available in the algorithmic system—in this case, the set of “interests” that Google recognizes (Cheney-Lippold 2011, 169; Deleuze 1992). And while both algorithmic and “real-world” identities are, to put it simplistically, socially constructed, there is a crucial difference in the site and degree of control over such constructions, and especially the ends to which they are put. Cheney-Lippold
frames “dividuation” as a matter of biopolitical control, highlighting the ways that algorithmic identities feed back into and help constitute real-world ones—as would be the case if the content that Google presents to me on the basis of my interest in “jazz” begins to affect my own understanding of jazz. I suggest that the formal styles of algorithmic performers can be understood in similar terms, so that the “jazz” that Shimon plays has its own algorithmic identity predicated not on genre labels and online search history but on Markov states derived from corpus data. Thus, returning to (and affirming) Wilf’s argument, what jazz robots afford is indeed a qualitatively new form of control over the “styles” derived from their corpora. But my reading of Meyer and Cheney-Lippold disputes the “level of reality of style,” in Wilf’s words, on which such systems operate and suggests a more critical view of algorithmic style.

As the robot’s engagement at the U.N. demonstrates, Shimon’s model of style and stylistic blending is predicated on the notion that jazz is inherently democratic, providing a model of conversation in which musical syntax furnishes an orderly set of controls over both the content and order of musical utterances among members of a group. Computation itself, by providing an information-theoretic infrastructure for robotic musicians, seems to reinforce the idea that interaction can be understood as a determinable, syntactically ordered set of procedures. But the above discussion has suggested that the relationship between machine learning and the musical “rules” governing these kinds of models is much more complicated than such a simple account allows. The rules coded into Shimon and Impro-Visor are related to the syntax of jazz, but in the end are more proximately the result of their various computational models, with all the decisions involved—including Shimon’s Markov model, as well as the limitation in both systems that confines melodic considerations to individual harmonies or measures. And while Shimon’s creators do not claim to model either jazz or human interaction perfectly, they do claim to leverage the specific qualities of computation and jazz, both as a matter of musical inspiration and as a means towards understanding human interaction more broadly. My contention has been that, for reasons involving both the lowest-level technical details and the highest-level musical ones, such a task is far more fraught than it seems.

On the other hand, and as I hope my discussion has also shown, the complexities of projects that mix computation, improvisation, and interaction make them rich theoretical sites for thinking through these issues; and as Wilf emphasizes, questions involving human interaction with complex computational systems are only becoming more pressing in the world at large. The final section of this article, then, turns to another kind of computer music project that refracts the issues encountered so far through a different improvisatory tradition and thus a different set of stylistic questions and their concomitant political stakes.

4. Voyager

George Lewis’s work in computer music takes explicit aim at some of the larger issues surrounding improvisation and human-computer interaction with which this article has been concerned. His long-running Voyager system is a real-time free-improvising partner, designed to lead, follow, or ignore its human collaborators and to develop musical ideas in complex ways. Lewis’s stake in the issue of computer improvisation is tied to a range of broader theoretical issues, as evinced in his role as co-editor of the Oxford Handbook of Critical Improvisation Studies (Lewis and Piekut 2016). In a response to a colloquy on the handbook published in Music Theory Online (Lewis 2013), he points to the ways improvisation impinges on issues of “ethics, social identity, order, and negotiation; computational creativity, telematics, and the virtual; time, consciousness, and complexity; indeterminacy and agency; migration and mobility; representation, cognition, and enactment; economic development; and . . . the histories of people and nations” (37). This staggering list reminds us that improvisation is not merely a musical phenomenon but rather a fundamental aspect of human activity.

In much of his scholarly work, Lewis also calls attention to a widespread failure to recognize Black experimental musical practices as such, especially where improvisation is involved. Rather than being understood as in dialogue with canonic experimental artists like John Cage, Black free improvisation is often understood merely as “jazz,” and earlier styles like bebop are reduced to popular music, despite the intertwined histories of these traditions and contrary to the claims of
many of the musicians involved (Lewis 2008, 1996). In addition to its implications for the relationship between Black musical practices and the more widely accepted canon of White experimental music, this argument should be understood to reflect on the distinction between the kinds of music played by Shimon as opposed to Voyager: while the latter, as an agent of free improvisation, is more recognizable experimentally for most listeners, the former relies for its musical material (i.e., the corpora of transcriptions on which its algorithms are trained) on artists who should also be understood historically as experimental—Monk, Parker, Davis, Coltrane—even if today they are often treated in conservative, neoclassical terms. The widespread understanding of mainstream jazz as conversational, democratic, and universal must be understood not as a feature of the music itself, but of the uses to which it is put, including in this case the kinds of computational models that are applied to it.

[4.3] My own stake in this conversation is not to define experimental music as such, but many of the same concerns that drive that question also bear on the issues raised in this article already: the ways computationally constrained styles are understood to embody certain kinds of grammar or syntax; the language- or conversation-like nature of jazz; and the relation between improvisation, style, and even the pragmatics of human-computer interaction. Because Lewis has theorized his own practice relatively extensively, and in so doing addressed some of these questions already, my discussion of his work will move in a slightly different direction, while aligning broadly with his own aims. In this section, I consider the specific workings of the free-improvising Voyager.

**Voyager 1: Improvising Ensembles**

[4.4] Lewis’s system does not encode a stereotypical musical grammar, like tonal harmonic syntax, and it lacks the internal representations for things like key and harmony that allow Shimon to recognize these kinds of musical objects. Its pitch interface consists of a simple MIDI parser combined with a “mid-level smoothing routine that uses this raw data to construct averages of pitch, velocity, probability of note activity and spacing between notes” (Lewis 2000a, 35; the rest of my general characterization of Voyager’s functionality is based on this essay). The system’s output consists of 64 monophonic MIDI voices, or “players,” which can be formed into different kinds of “ensembles” with different behaviors, themselves operating simultaneously. These ensembles are constantly being reconfigured, with a subroutine running every few seconds to determine how many players to include in the next ensemble, whether to reset all players at once, whether to reset only a single ensemble, or some other behavior. New ensembles are also assigned timbres, melodic algorithms, and pitch sets, potentially microtonal (though more recent performances take place on a Disklavier that features fewer options for some of these features than the programmable MIDI controllers featured in earlier iterations). Other parameters include “volume range, microtonal transposition, tactus (or ‘beat’), tempo, probability of playing a note, spacing between notes, interval-width range,” and others, including in particular the possibility of “imitating, directly opposing or ignoring the information coming from the [other] improvisors” (Lewis 2000a, 35). As the latter option implies, Voyager can perform entirely on its own if no human input is provided.

**Video Example 2** presents an excerpt from a 2016 performance at the Kennedy Center that features pianist Jason Moran improvising with Voyager (performing on/as a Disklavier). While it is impossible to read directly from the performance which algorithmic decisions lead to particular musical moments, some of the system’s basic features are easy to recognize: pausing to “listen,” the interjection of new ideas, and the imitation of features of the human’s playing, particularly when Moran makes sudden dynamic changes and Voyager follows suit.

[4.5] Indeed, when it comes to imitating human input, the ensemble architecture allows Voyager a wide range of possibilities. For example, an ensemble could be formed that imitates the music it just heard in terms of interval spacing but not rhythm, or another ensemble could repeat what it just heard but at a much lower volume; indeed, both could happen at once. If this kind of imitation is an imitation of style, then it takes place according to a very different kind of stylistic model than Shimon’s. The latter, as discussed above, hears everything as stylistic, incorporating every note it hears into the corpus-trained Markov model. Moreover, it works to assimilate notes into harmonies in real time, transforming raw input into useful knowledge about the state of its musical world, which in turn affects what action the system takes. Voyager similarly accepts all input, in the basic
sense that everything it hears is assumed to be a note with some definite duration and pitch, but
the indexical link between input and output is less hierarchical. Lewis does not specify the length
of Voyager’s memory, but the ensemble architecture means that even if the system stores input over
a relatively long period, such input is not incorporated into a probabilistic model like Shimon’s,
where recurring background harmonic states determine the immediate output, which itself
transforms the model associated with a given state. Instead, Voyager separates out the parameters
of its input without linking them to any background state, so that its musical ontology consists of
differentiated—and, in fact, entirely dissociated—statistical objects describing the kinds of features
listed above: pitch, velocity, probability of note activity, volume, and spacing between notes.

[4.6] The differences between the two models shine a light on the relation between Meyer’s
selectional, information-theoretic model of style and computation itself. The model at the heart of
modern digital computation, the Turing machine, is fundamentally understood to operate on the
basis of a set of underlying states that determine what kinds of inputs are accepted and how they
are processed. Turing’s theoretical model is borne out in actual computers, which, though more
complex, are limited to states built out of combinations of binary-valued transistors. If there is a
historical and conceptual link between the information-theoretic model of communication and
Meyer’s selectional account of style, the discrete state model behind modern computer hardware
puts this link into practice for algorithmically generated music; the result is that any system that
produces music computationally is necessarily selectional on some level. The typical move, as seen
in the discussion of hidden Markov models, is to map these computational states onto harmonies
or harmonic functions, a mapping that should be understood as both conceptual and “ontological”
on the level of the computer hardware.

[4.7] Voyager has no need for a traditional notion of harmonic function, but it is still a
computational system and thus cannot avoid the notion of “state” altogether. But where Shimon
leverages discrete computational states to overdetermine the robot’s response to conventionally
understood harmonic contexts, Voyager uses them to disaggregate the parameters of its input and
to recombine them in many different ways. Indeed, Lewis links this feature to Voyager’s particular
mode of stylistic recognition. In what we might consider a description of the system’s formal style,
as defined above, Lewis suggests that “through the accumulation and articulation of many small
details, an interactive, adaptive input structure that generates a sufficiently detailed representation
of its input can then produce a musical output perceptible by an improviser as analogous to
various states that were experienced during improvisation” (2000a, 36). Where for Shimon, the
accumulation of details is Markovian—characterized by a focus on short-term note-to-note
successions of pitch and rhythm—Voyager’s strategy is to reflect back fragmented aspects of the
human’s performance in one ensemble’s imitation of pitch spacing, another’s dynamics, perhaps
another’s combination of rhythmic and textural density. Lewis fittingly connects this strategy with
what he calls a “state-based” approach, as opposed to “motive-based”—a distinction inspired not
by Turing machines but rather by different ways humans organize their playing in free
improvisation. According to Lewis, “In this kind of [state-based] improvisation, the global
aggregation of sonic information, considered in a temporal sense, is privileged over moments of
linear development” (1999, 105).

Voyager 2: The Sound of Free Improvisation

[4.8] Over the course of decades of performances, Voyager seems generally to have been taken
seriously as a free-improvising agent. This is not to say that it is heard as if it were simply another
human on stage. Aaron Cohen, reviewing Lewis’s performance at the AACM’s 30th anniversary
festival in 1995, describes the piece as “a collage of seemingly random sounds—particularly bells
and crashes—that teleports jazz’s instinctive underpinnings into a more inflexible binary digital
system” (1996, 60). Zane East’s review of the eponymous 1993 record hears the program as
generally reactive: “Sustained trombone notes often cause explosive outbreaks from Voyager, while
more staccato playing evokes a similar response from the program. In any case, human performers
probably have the upper hand in playing with Voyager, because the program reacts to their
playing” (1995, 109). While the overall implication is that the human is in charge, East understands
that reactiveness as the basis for an interactive context at play in the recordings, noting that “the
performer need not necessarily be an expert on the program’s internal operation—if each plays and
listens simultaneously, everything will turn out all right” (110).

[4.9] Lewis frames his system’s interactional style in terms of the cumulative effect of Voyager’s
attentiveness to certain seemingly small details of a human performance, and the way these barely
noticed or unrelated features are indexically linked to broader categories like emotions: “In this
way, I believe, the emotional state of the improvisor may be mirrored in the computer partner,
even if the actual material played by the computer does not preserve pitch, duration, or
morphological structures found in the input” (1999, 106). The process is akin to the “dividuation”
that Cheney-Lippold attributes to the algorithms associated with Big Data, though here the object
is not a “human” (using Cheney-Lippold’s quotation marks to denote an algorithmically generated
identity category) made up of indices collected and stored over a relatively long time, but rather a
more local “emotional state.” In both cases, an object that stereotypically cannot be observed
directly—a human’s political leanings, material desires, or emotions—is indexed by signs that are
either seemingly unrelated or below the level of conscious recognition (e.g., web browsing habits,
distributions of note duration or loudness).

[4.10] Whatever the relation between a human performer’s emotional state and the statistical
details of his or her playing, Voyager’s way of dealing with these details is to reflect them back so
that the system’s performance is a kind of imitation. That means that “if an improvisor wishes to
hear the program play in a certain way, the most direct route is to actually play in that way. After
that, it’s up to the system to deal with what it finds. This can be viewed as ‘getting the system’s
attention’” (Lewis 1999, 104). But what kind of attention, exactly?

[4.11] Unlike Shimon, Voyager does not attend to its human partner(s) in the same way across the
duration of an interaction. While the system’s interface always takes in new input, that input may
be ignored; and more importantly, different ensembles attend to different features at different
times. The human performer is thus unstable as a component of Voyager’s formal style, which only
aggregates statistical information about its input over short and unpredictable periods. In short,
Voyager’s own internal ensemble model projects onto the humans to whom it listens. Different
semiotic agencies emerge and dissipate as individual ensembles constitute themselves on the basis
of different musical features, so that the pitches a human plays may have no interactive role in
what one ensemble is doing, while they may be almost entirely determinative of another’s
behavior. Thus, while interactions with Shimon seem to hinge on the mutual recognition of similar
capabilities in human and robot, a human improvising with Voyager is more likely to recognize a
plurality of musical capacities and approaches, where the indices that the system produces point
not only to certain strategies for the arrangement of musical materials but also to the source of
those materials in the human’s own playing.

[4.12] Furthermore, with Voyager, the kinds of propensities a human can assume on the computer’s
behalf are of a very different kind than those projected by Shimon. The latter, by responding to
harmonic contexts with vertically oriented, chord-focused melodic patterns, projects a propensity
to continue playing in that way. The interactive context transforms gradually over time as the
Markov models associated with specific harmonies are updated, so even if the human tries to
radically shift the context by playing extremely dense melodic lines or heavily altered harmonies,
Shimon will only respond when new styles of playing become reinforced strongly enough in the
computational model to become statistically likely as responses. For similar reasons, Shimon is also
unlikely to instigate sudden shifts in context.

architecture mean that it takes a more active role in transforming the interactive context. Rather
than deducing a governing background state from a human’s playing, Voyager partially dictates its
own state in the form of randomly emerging and reforming ensembles. Since the timing and
makeup of these ensembles is not determined in any way by human input (and the ensembles
themselves behave asynchronously), Voyager actually does a certain amount of its musical work
without attending to any external indices at all. Additionally, the ensembles themselves always
have the option of ignoring human input altogether. In many cases, though, the system modulates
the specific behavior of individual ensembles by incorporating aspects of the human performances
it hears. Lewis makes clear that this kind of willfulness is not meant to be musically neutral but instead reflects his own style—recalling that he considers Voyager to be first and foremost a composition—along with his own priorities in building free improvisation contexts: “part of the task of constructing Voyager consisted of providing the program with its ‘own sound’” (2000a, 37).

While it might seem paradoxical that a performer’s “own sound” could consist in its imitating other styles, this is an important aspect of all of these computer improvisation systems—Shimon, Voyager, and in a non-real-time context, Impro-Visor. At least in cases where improvisation is understood to involve some kind of dialogue (or bidirectional “transduction,” in Lewis’s terms), imitation becomes a primary index by which style is projected and recognized. Not just the fact of imitation tout court but rather the specifics: which musical features form the basis for the imitation, and what the recognition of imitation implies about the imitator’s capabilities and propensities. In other words, the extent and kind of imitation that one hears in Shimon or Voyager sets the terms of the musical exchange by directing attention to the musical processes themselves; an astute human performer will not only play along but will recognize what the computer has recognized in his or her own playing.

5. Conclusions

This article has explored the semiotic and improvisational capacities of a number of different computer music systems, organized loosely around conceptions of jazz, from traditional combo to free improvisation, and more specifically in terms of their efforts to imitate human styles of playing. I have argued that such systems evince style by way of two connected processes. First, computational formal styles represent musical objects in characteristic ways, themselves constrained by the selectional nature of computer hardware, that constrain the kinds of musical decisions they can make and indices they can produce. (Think of Shimon’s vertical, harmony-oriented style of soloing, or Voyager’s dividuating ensembles.) Second, computer improvisors recognize style in their human partners and respond accordingly. The stylistic appropriateness of the music so generated lies not only in its formal features—playing a succession of notes that sounds like something Coltrane once played—but in the way those features point to the improvisational context itself by responding to entailments of previous utterances and by modifying that context.

These are not just features of computer improvisation. After all, improvisation within a group of human musicians also requires that the notes be not only correct (or close enough) but also appropriate given what came before and capable of implying what should come after. Part of what makes computer improvisation an interesting object of study in itself is the fact that it attempts to forge an interactional context between such different agents, especially given the indexical richness of musical improvisation.

The stakes inherent in this attempt come to the fore in the questions these projects pose about the politics of relations between humans and computational systems. Lewis has often thematized this issue in his writings about computer music, noting that “interactivity has gradually become a metonym for information retrieval rather than dialogue, posing the danger of commodifying and ultimately reifying the encounter with technology,” especially as “corporate power assumes an important, even dominating role in conditioning our thinking about computers, art, image, and sound” (1995; quoted in Lewis 2000a, 36). Many of the features of Voyager’s ensemble model, of its “aesthetic of variation and difference” (36), are explicitly aimed at resisting this propensity of late capitalism. The connection to Cheney-Lippold’s account of algorithmic dividuation is thus complex: for Lewis, the fragmentation of the human player is framed as resistant to, rather than complicit with, the kinds of corporate interests behind today’s large-scale algorithmic infrastructures. Of course, Lewis first developed Voyager well before the advent of the current internet-driven economy, and in particular before the development of the kinds of sophisticated machine-learning techniques driving “Big Data” in its present form. Dividuation was not a significant feature of the discourse or practice of computation or artificial intelligence until a particular historical moment. Thus, while Voyager at one time looked radically different from contemporary forms of interaction between humans and computers, the distance is less
pronounced today. What differentiates Lewis’s system most clearly from today’s dividuating algorithms is its lack of memory—it will not build a statistical profile of its collaborators and thus lacks the musical equivalent of “targeted advertisements.” And while it seems even crude to even consider such a comparison given Voyager’s deep critical roots, it is imperative not to ignore the effects of changing technical conditions on both the aesthetics and politics of such a project.\(^{(47)}\)

[5.4] In that light, Shimon and the related projects modeling mainstream jazz provide more up-to-the-minute examples of the mutual implication of the technical and the theoretical, especially in Wilf’s reading. For Wilf, the blending afforded by algorithmic implementations of style produces an immanent critique of a body of social theory that is not up to the task of accounting for precisely the late-capitalist conditions that Cheney-Lippold describes. And unlike Voyager, Markov-based systems remember past inputs and thus build statistical profiles of the humans that interact with them (as well as of the corpora used to train them), and thus do, to a greater extent, reflect the present state of algorithmic practice and Big Data. I agree with Wilf that theory—either of culture or merely of style—must account for these blendings, which do indeed represent a qualitatively (and quantitatively) different approach to musical style. But, as I have argued, Wilf’s position can too easily be read to suggest that such algorithms are broadly successful, that stylistic blending can be taken at face value. In order to understand the political and theoretical implications of these robotic musicians, we have to attend even more closely to the ways their formal styles resist understanding in terms of “all of the rules of jazz” precisely because of their selectional, rule-based, computational nature. Wilf’s suggestion that we move from thinking of “style” to “styles of styling styles” is provocative, but we have to recognize that, in this computational setting, the latter formulation is not just a recursive nesting of the former with itself but also a transformation of the meaning of singular “style.”

Brian A. Miller  
Yale University Music Department  
469 College Street  
New Haven, CT 06520-8310  
brian.miller@yale.edu

---

**Works Cited**


Jurafsky, Daniel and James H. Martin. 2008. Speech and Language Processing, 2nd ed. Prentice Hall.


Seaver, Nick. 2011. “‘This Is Not a Copy’: Mechanical Fidelity and the Re-Enacting Piano.” *Differences* 22 (2–3): 54–73.


Discography


Footnotes

competitions tested algorithmic accompanists and DJs.

2. Ariza 2009 gives an overview with numerous examples, noting that the Turing test is almost always invoked in a form other than Turing’s original (discussed below). He also notes that published invocations of the test are, as a whole, disturbingly optimistic: “every executed musical [Turing test] reports machine success” (60).


4. Various aspects of the game seem underdetermined and have been debated, such as whether Turing cared whether the man is always the impersonator, and precisely how the various agents communicate. (It is generally presumed to be by typewritten notes or some kind of intermediary agent.) See Epstein, Roberts, and Beber 2009.

5. Ariza (2009, 50) does not discuss this distinction in his otherwise thorough overview of musical applications of the test. Interestingly, he does correctly point out that the original test explicitly prohibits requests for “practical demonstration,” leading him to argue that the only true musical Turing test would simply enact Turing’s original scenario, in the medium of natural language; one would ask the computer to speak intelligently about music (65).

6. See Ford, Glymour, and Hayes’s commentary on Turing’s original paper in Epstein, Roberts, and Beber 2009.

7. See the project’s website, http://www.musicaresearch.org. Earlier reporting on the project gives its name as “Musical Improvising Collaborative Agent,” but the website currently glosses the “I” as “Interactive.”


10. The systems considered here are representative but far from exhaustive of the full scope of algorithmic improvisers; my examples are intended to draw out a particular set of pressing concerns through close reading rather than to survey the field. Tatar 2019 present such a survey, categorizing and discussing seventy-eight “musical agents,” many of which improvise and many of which model style in some form. See also Banerji 2018b and Linson et al. 2015 for particularly notable approaches to algorithmic free improvisation, and Wilf 2013a and 2013b for another example of robotic mainstream jazz.

11. Ethnomusicologist Ritwik Banerji’s (2018a, 2018b) work leverages a related claim, treating algorithmic improvisation not only as music cognition but as ethnography. By observing and discussing performers’ interactions with a free-improvising agent named “Maxine,” he interrogates received notions of free improvisation’s egalitarian ethics and of ethnography’s methodological
commitments.
Return to text

12. See also an excerpt from Jazz at Lincoln Center’s mission statement: “We believe jazz is a metaphor for Democracy [sic].” Jazz at Lincoln Center website, https://www.jazz.org/about/ (accessed 27 February 2019).
Return to text

13. White and Quinn 2018 discuss one heuristic method for determining an appropriate number of states for a given problem, and their own analyses of the Bach chorales demonstrate solutions with three and thirteen states. They also discuss the algorithm for producing an HMM in more detail; see also Jurafsky and Martin 2008 for a more technical discussion in the context of natural language processing.
Return to text

14. To give a brief example, in Krumhansl’s model, if a passage contained many instances of all of the notes in C major, along with a few non-scale pitches like E and F, then the algorithm would still probably select C major as the most likely key. Krumhansl’s algorithm has been modified and improved in recent years, but Shimon seems to implement the original. See Krumhansl and Kessler 1982, and for a revision of the model, Temperley 1999.
Return to text

15. Shimon’s developers do not specify how the order is chosen. Usually VOHMMs change orders dynamically in order to improve the probability of the new output.
Return to text

16. This raises the issue, which I leave to the side for now, of the roles of different instruments and their affordances in this kind of stylistic mixing, given the odd combination of saxophone and piano in the training corpus, reproduced on marimba.
Return to text

17. See, for example, Gjerdingen 2014 and White 2014. David Huron’s work on music perception and cognition (most notably 2006) is complemented by his work on the Humdrum Toolkit, a set of tools for computational music analysis, https://www.humdrum.org (accessed 27 February 2019).
Return to text

Return to text

19. Here, I am glossing and modifying terms from art historian Whitney Davis (2011), who uses the clunkier terms “stylisticality” (instead of formal style) and “pure style.” The idea of pure style is intentionally problematic, and I introduce it here merely in contrast to the easily described formal styles of algorithms.
Return to text

Return to text

21. A passage that changes direction has to use multiple deltas, like (R8 L8 (Δ 3 5 H8) (Δ -2 -1 C4.)), which also includes a rest (R) and a dotted quarter note (4.)
Return to text

22. There are various rules for dealing with situations where previous note choices prevent an expression from being worked out correctly, e.g., if there were no chord tone within the allowed interval range. The exceptions generally err toward playing chord tones at the expense of the

23. While the interval constraints would have allowed the chord tone F# to be selected in place of the E in the first measure, in the second there would have been no other choice unless the unusual flat thirteenth were explicitly allowed to accompany a minor chord as a color tone—so ideas about what constitutes a chord or color tone are of course part of the system’s musical ontology.

24. Though the authors note that the window slide by varying amounts (so as to avoid only ever capturing melodic fragments from within the bar lines of a single measure), they do not discuss this parameter in any detail (59).

25. As one further example, the abstract melody (Δ 3 5 C C C C) could represent the sequence C–E–G–B over a C7 chord or C–F–G–B over the progression Fm7–C7.

26. The clusters are generated by an algorithm called k-means, which randomly chooses k data points as cluster centers, assigns the rest of the data points to the nearest cluster, calculates a new center point for each cluster, and then repeats the process until, like the HMM algorithm, further iterations produce little change in the model. Similarity (or nearness) is measured as Euclidean distance based on the seven parameters listed above. As with HMMs, the number of clusters is not automatically determined and must be chosen by the user; for Impro-Visor, the authors use the total number of melodies divided by ten. In the world of data science, k-means is often used as an exploratory technique, attempting to find structure in otherwise messy data without guaranteeing an optimal solution.

27. As Gillick, Tang, and Keller (2010, 57) put it, “it is known that this process [of jazz improvisation] is often informed by prior creation and practice of vocabulary ideas prior to performance.”

28. For an extensive semiotic treatment of this process, see Kockelman 2013, chapter 3.

29. In a Peircean idiom, Shimon thus implicitly suggests that style in improvisation, blended or not, is a semiotic object defined as much by the interpretants it gives rise to as by the signs that stand for it. See Kockelman 2013.

30. Meyer (1989, 104), for all his focus on compositional style, is quite clear that the performance of preexisting works is just as much a matter of style as composition. That said, Wilf’s conception of style largely reproduces Meyer’s focus on “primary” parameters, so-called because they are easily discretized and thus can be arranged syntactically, unlike “secondary” parameters like timbre, which only exist on a continuum (1989, 14ff).

31. To give an example familiar to many music scholars, a change of strategy might involve a new way of executing a formal structure, like using an IAC rather than a HC in the middle of a Classical parallel period, whereas a change in rules might involve the shifting of priorities within the tonal system itself. In the last section of the book, Meyer argues that the transition from Classicism to Romanticism exemplifies such a change. See Meyer 1989, chapters 4 and 6.

32. I do not mean to suggest that none of these groupings are stylistically meaningful but instead argue that Meyer’s ad hoc approach to stylistic grouping points out a lack of theorization of the concept of replication—perhaps the central term in his theory of style. If the same work
participates stylistically in groupings according to region, time period, and individual composer, are the same kinds of patterns being replicated in each case? What if the same musical material seems to play a role in different patterns at different scales or in different groupings? These are difficult questions that I sidestep in this article by focusing on computational imitation rather than replication in a broader sense; see Miller forthcoming for a more detailed examination of Meyer’s approach to these issues.

Return to text

33. Evaluating the actual musical output of such systems is difficult—I am aware of no recordings of any computer improvisers involving blendings of the kind Wilf describes. For Shimon’s part, of the dozens of videos available, most demonstrate the robot playing along with a jazz combo to various preprogrammed tunes. These examples make it difficult to evaluate the robot’s improvisational abilities or its style in actual performance; they generally involve some kind of introductory tempo-setting musical figure from the human keyboard player, after which Shimon joins in, generally holding to the beat and matching tempo changes, though these usually take a few beats to settle into place. One video demonstrating the specific kind of blending of styles described in the project’s publications begins with Shimon playing a short musical figure entered into a smartphone app; the robot repeats the figure in time with a backing drum track, and then begins eight measures of improvisation on that figure (see the link below). The phone’s screen shows two sliders, one on an axis between labels for Coltrane and Monk, with a marker set all the way at the latter end, and another simply labeled “Me,” initially set all the way up. As the human gradually lowers the “Me” slider and shifts the other from Monk to Coltrane, Shimon appears to play increasingly erratically, with bursts of notes in various registers occasionally making way for recognizable fragments of the original melodic figure. As the slider reaches Coltrane, Shimon plays faster, and just before returning to the original motive (at which point the video cuts off), a more or less clear scalar figure emerges in the marimba’s upper middle register. See in particular, “Shimon Zoozbeat monk coltrane short,” on YouTube channel Georgia Tech Center for Music Technology, https://youtu.be/FEpQwi0Pgw (accessed 28 January 2019).

Return to text

34. Like Shimon, the system that Wilf (2013b, 721) describes also uses some kind of Markov model for its stylistic imitation and blending, though more specific details are not available.

Return to text

35. Any Google user can view the categories that make up their algorithmic identity at https://adssettings.google.com/ (accessed 10 July 2020).

Return to text

36. Much work on the social effects of machine learning focuses on the ways algorithmic systems reinforce existing biases in data. See, for example, Noble 2018 and Bolukbasi et al. 2016. In his own response to Wilf 2013b, Adrian Mackenzie (2013, 733–4) attends to this issue, noting the irony in seeking inspiration from an algorithmic technique designed to capture the statistical status quo.

Return to text

37. While music scholars have tended to focus on a narrower understanding of improvisation, other disciplines have traditionally approached the concept in terms closer to those proposed by Lewis. Dumit et al. 2018, writing from the perspectives of anthropology and performance studies, note: “We’ve heard it joked that sociologists treat people as rule-followers, while anthropologists see people as constantly improvising in ways to make it look like they are following rules.”

Return to text

38. In a 2016 interview with Jason Moran, Lewis notes, “I’d never actually heard Voyager talked about as a jazz composition before, that was new to me.” “MacArthur Foundation: Jason Moran, George E. Lewis — Millenium Stage (October 2, 2016),” https://youtu.be/Mn3M2JLQ0ts (accessed 10 July 2020).

Return to text
39. Commercial recordings featuring *Voyager* include Lewis 1992 and Lewis 2000b. Note that, orthographically, I follow the conventions of the various systems’ creators: *Voyager* is treated as the title of a composition, where Shimon is the name of a robot.

Return to text

40. For an overview of Turing machines through a semiotic lens appropriate to the discussion here, see Kockelman 2017, 152ff.

Return to text

41. Friedrich Kittler (2014) takes this idea to the extreme, arguing that the binary, selectional structure of computer hardware fundamentally structures all computer programs, so that the apparent flexibility of software that presents the user with an interface of non-deterministic everyday language is an illusion.

Return to text

42. For a free-improvising agent with a different formal style, see Linson et al. 2015. Linson’s system, called Odessa, is based on the principles of a robotics architecture called Subsumption, which aims to navigate an environment without explicit representations of objects or spaces.

Return to text

43. Lewis (1999, 105) cites Evan Parker as a performer whose work (circa 1980) involves state-based interaction, and Anthony Braxton and the later Coltrane as primarily motive-based.

Return to text

44. Shimon, on the other hand, while largely controlled by underlying harmonic states, is ultimately meant to produce melodic, motivic imitations of the players in its corpus.

Return to text

45. Lewis makes an explicit link between the concepts of “sound” and “style”: “An improviser can ‘sound like’ another improvisor, as demonstrated by the numerous players who chose to ‘sound like’ Coltrane, Miles Davis, or Derek Bailey. The notion of ‘sound’ in improvised music may be seen as analogous to the Western compositional concept of ‘style,’ especially in a musically semiotic sense” (1999, 105).

Return to text

46. On modern computer languages as reflective of late capitalist modes of production, see Galloway 2013.

Return to text

47. More recently, Lewis has continued to strike a note of optimism about human-computer improvisation, albeit with a very different focus than my own. In the recent Oxford Handbook of Algorithmic Music, Lewis reflects on a suggestion from anthropologist Lucy Suchman, who takes “the boundaries between persons and machines to be discursively and materially enacted rather than naturally effected and to be available . . . for refiguring” (Suchman 2007, 12; quoted in Lewis 2018, 128). Lewis adds that, “At the moment at which musical improvisation with machines enacts this radical fluidity of identity, what we have is not a simulation of musical experience, but music making itself—a form of artificial life that produces nonartificial liveness.” (128).

Return to text

Copyright Statement

Copyright © 2020 by the Society for Music Theory. All rights reserved.

[1] Copyrights for individual items published in *Music Theory Online* (MTO) are held by their authors. Items appearing in MTO may be saved and stored in electronic or paper form, and may be shared among individuals for purposes of scholarly research or discussion, but may not be republished in any form, electronic or print, without prior, written permission from the author(s), and advance notification of the editors of MTO.
[2] Any redistributed form of items published in MTO must include the following information in a form appropriate to the medium in which the items are to appear:

This item appeared in Music Theory Online in [VOLUME #, ISSUE #] on [DAY/MONTH/YEAR]. It was authored by [FULL NAME, EMAIL ADDRESS], with whose written permission it is reprinted here.

[3] Libraries may archive issues of MTO in electronic or paper form for public access so long as each issue is stored in its entirety, and no access fee is charged. Exceptions to these requirements must be approved in writing by the editors of MTO, who will act in accordance with the decisions of the Society for Music Theory.

This document and all portions thereof are protected by U.S. and international copyright laws. Material contained herein may be copied and/or distributed for research purposes only.

Prepared by Andrew Eason, Editorial Assistant