Interactive Musical Partner: A Demonstration of Musical Personality Settings for Influencing the Behavior of an Interactive Musical Generation System

Jeffrey Albert
Loyola University New Orleans
College of Music and Fine Arts
6363 St. Charles Ave.
New Orleans, LA 70118
jvalbert@loyno.edu

Abstract

The Interactive Musical Partner (IMP) is software designed for use in duo improvisations, with one human improviser and one instance of IMP, focusing on a freely improvised duo aesthetic. IMP has Musical Personality Settings (MPS) that can be set prior to performance, and these MPS guide the way IMP responds to musical input. The MPS also govern the probability of particular outcomes from IMP's creative algorithms. The IMP uses audio data feature extraction methods to listen to the human partner, and react to, or ignore, the humans musical input, based on the current MPS. This demonstration shows how the MPS interface with IMP's generative algorithm.

Introduction

The Interactive Musical Partner (IMP) is software designed for use in duo improvisations, with one human improviser and one instance of IMP, focusing on a freely improvised duo aesthetic. My goal in the creation of the IMP was to make a software system that could live up to all three parts of its name. It would have to be interactive, meaning that there is a two-way flow of information between the human performer and IMP. It would have to be musical, meaning that the results are musically rewarding both the performer and to listeners. And finally, it should be a partner, meaning that it is equal parts leader and follower, not always simply accompanying the improvising human, and at the same time, not always requiring the human to musically accommodate its output.

It is important to understand the aesthetic space that a work of art, or piece of software, strives to inhabit, especially if openness is at the heart of that space. IMP strives to function in the aesthetic lineage of the freely improvised duo. David Borgo describes the music, "often dubbed 'free improvisation'" as tending to, "devalue the two dimensions that have traditionally dominated music representation - quantized pitch and metered durations - in favor of the microsubtleties of timbral and temporal modification."(Borgo 2005) This is an accurate description of the musical priorities of IMP, and informs the way IMP deals with the ways it hears, remembers, and creates musical content with the fewest possible levels of abstraction. Whenever possible pitches are dealt with in terms of frequency and durations in terms of milliseconds. By avoiding thinking in terms of note names and note vales, IMP can more easily navigate the spaces outside of tonality and tempo.

A number of interactive musical systems have preceded IMP. Jean-Claude Risset's *Duo for Piano* is composed music that uses computer manipulated MIDI information to control an interactive part played by the computer on the same piano(Risset and Duyne 1996). Richard Teitelbaum's *Concerto Grosso for Human Concertino and Robotic Ripieno* also uses MIDI input from the human participants, although some of it is generated by acoustic sound run through pitch to MIDI converters, and similarly the computer output is to machine playable pianos(Teitelbaum 2006). Both of these pieces use the musical input from the human participants to control or influence the computer's output.

Another class of interactive software learns from the human's input and creates a style based on what the human plays. Francois Pachet's Continuator(Pachet 2003), and the OMax software developed at IRCAM(Assayag, Bloch, and Chemillier 2006) both build their playing style based on the input of the human partner, or a preloaded corpus. Pachet sees his software as continuing the phrase of the human, while OMax is often described as co-improvising with the human.

The common thread through these systems is that they are all dependent upon human input. A system that interacts with human input, but does not depend upon it, is the *Voyager* system by George E. Lewis. Lewis describes *Voyager*s structure: "as multiple parallel streams of music generation, emanating from both the computers and the humans a non-hierarchical, improvisational, subject-subject model of discourse, rather than a stimulus/response setup."(Lewis 2000) In *Voyager*, the computer and human are equal in terms of agency; either one can make music without the other.

IMP is philosophically most similar to *Voyager*, in that IMP is not dependent upon a human partner, it simply interacts with one. IMP could play music on its own. Unlike *Voyager*, IMP has but one voice. With IMP's single voice syn-

thesis, a performance with IMP might superficially sound more like OMax or the Continuator, but its philosophy of interactivity and structure are much more like *Voyager*. Unlike all of these other systems, IMP listens to audio data, deals with input from the human and generational algorithmic data in the least abstracted way possible, usually just frequency, amplitude, and length of event in milliseconds.

IMP was programmed in the Max 6 Programming environment, using externals by Tristan Jehan and Tap.Tools, and uses The Wekinator for machine learning implementation. IMP consists of: a synthesis module, a Musical Personality Settings (MPS) module, a frequency decider, a duration decider, a global variation module, a listener module, and a number of smaller decider modules. I will use the term creative algorithm to refer to the aspects of the duration decider and frequency decider that control IMP's autonomous output.

Since this article is of limited length, and focused on a demonstration of the Musical Personality Settings, the other aspects of the system will only be briefly described. Deeper explanations of the function of other parts of the system and their underlying philosophies can be found in my thesis(Albert 2013).

Synthesis Method

The synthesis module is IMP's voice. This is the section of the software that makes the sounds, and manages the messages sent from the other modules. The frequency decider, duration decider, global variation module, and listener module are all sub-patches of the synthesis module.

The synthesis module uses frequency modulation (FM) synthesis to generate its sounds(Chowning 1985). The choice of FM synthesis as IMP's voice was as much an aesthetic decision as a technical one. While it was tempting to try to design IMP with a more organic voice, in part to try to make the human performer forget that IMP was not in fact human, I ultimately decided that giving IMP a voice that would remind the performer that IMP was not human was a better path.

IMP also employs a second order FM synthesis, meaning that there is a second modulating oscillator, which modulates the product of the first order oscillators. Variations of the combination of these two FM pairs are how IMP controls timbral variety. The primary FM pair is usually set to a simple harmonicity ratio; 2 is the default. A more complex ratio on the second FM pair allows for IMP to make its sound more complex and strident by adjusting the amount of modulation from the second modulating oscillator. The modulation depth of this second order pair is controlled by a gain, which is controlled by the timbral noise analysis module. This means that at times the second order modulator is completely muted and does not affect the sound. This mechanism will be discussed in greater detail in the section on Machine Learning and Timbral Interaction.

The data sent to the synthesis module consists of a frequency and a duration for each event; amplitude and timbre are determined in other processes. There are two primary modules in the creative algorithm: the frequency decider, and the duration decider. Each functions very similarly, but their processes do not affect each other. The frequency decisions are made independently of the duration decisions, and vice versa. The frequency and duration deciders each choose between musical data generated by IMP, independently of the human input, and musical data that was introduced by the human.

Musical Personality Settings

One of the original goals of this research was to design a system with variable sets of behavioral characteristics, or musical personalities. This is implemented in IMP through the Musical Personality Settings (MPS), which are seven separate parameters that influence various aspects of IMP's behavior. The parameters are: Density of Events, Length of Events, Rhythmic Regularity, Frequency Listenerness, Duration Listenerness, Melodicness, and Variation.

These parameters were selected so that various aspects of IMP's behavior could be controlled independently. The MPS are used to weight decisions made in the generative algorithms, which generate the actual data used to create the sounds (frequency, duration, etc). The MPS affect musical output by affecting the ways in which the generative algorithm makes its decisions, but the data sent from the MPS to the generative modules is simply integers that plug into weighted decision making modules. Melodicness is the only MPS parameter that determines actual musical data, in that it controls the set(s) of pitches from which the generative algorithm will chose.

The Density of Events parameter controls the weighting of the `density_decider` 's decision algorithm, which decides whether an event will make sound or not. The higher this parameter is set the higher the sound to silence ratio will be. This parameter is also influenced by what is heard from the human, once an episode begins.

The Length of Events and Rhythmic Regularity parameters work together to control IMP's tempo and sense of pulse. I use these terms (tempo and sense of pulse) loosely in this context, since there is no abstraction of meter present, but there can be a sense of IMP playing faster or slower, and in more or less regular event lengths. The Rhythmic Regularity parameter controls a pool from which duration proportions are chosen in the creative algorithm, and the Length of Events parameter controls a factor that controls the speed at which these proportions are realized.

Listenerness is a term I have coined to describe the two parameters that control IMP's responsiveness to human input. The lower the listenerness value, the more independently IMP will behave, and the higher the value, the more IMP will derive its output from what it has heard from the human. There are two listenerness settings; one for frequency and one for duration. Frequency Listenerness controls the weighting of the frequency decider mechanisms and influences whether IMP's pitch output is derived from its creative algorithm or from the pool of pitches it remembers hearing from the human. Duration Listenerness controls the weighting of the duration decider and similarly influences IMP's output in terms of duration of events.

The Melodicness parameter sets a set of pitches from which the creative algorithm chooses when IMP is generat-

ing content on its own. As the value moves from low to high the pool of available pitches moves from pentatonic sets, through major scales, melodic minor (ascending) scales, diminished scales, whole tone scales, and finally to a fully chromatic set of pitches.

The final MPS parameter is Variation. This parameter weights the decisions made by the global variation module, which controls a mechanism that causes variation of the other MPS parameters. The most often the parameters will change is once per second, and the least often they will change is every 100 seconds, with the largest possible jump on any MPS scale being 10 units, on a 128 unit scale. This keeps IMP's output from seeming static in content, but helps avoid seemingly random huge shifts in musical space as well.

Machine Learning and Timbral Interaction

The Wekinator is a real-time machine learning application by Rebecca Fiebrink(Fiebrink 2011). While IMP is playing, the Wekinator is running as a separate application on the same computer. IMPs listener module sends loudness, brightness, and noisiness data to the Wekinator via OSC. The Wekinator runs these three streams of data through a neural network that outputs a single value between 0 and 127, which is sent back to IMP via OSC where it controls the timbral elements of the synthesis module.

The Wekinator must first be trained by playing tones into the feature extractor (which is part of the listener module), and assigning a value between 0 and 127 to each sound played in. This is usually done with 0 being the most pure tone, and 127 being the noisiest tone. However, if one wanted IMP to respond differently in the timbral domain, one could train the Wekinator differently. When IMP gets a 0 from the Wekinator, IMP plays its most pure tone, and a 127 gives its noisiest tone, with the varying degrees in between. With that knowledge, the Wekinator could be trained for any given input to make pure tones or noisy tones, as long as that input is associated with that value in the training stage. Once this training has been done, an .arff file can be saved and loaded at a later time. This also allows for performances with different timbral training data by simply loading a different .arff file into the Wekinator.

The value generated by the Wekinator is tied to the gain on the second order modulation oscillator in the synthesis module. This means that when the human is playing pure tones, the second order modulation is turned off. As the humans sounds get noisier, the second order modulation depth is increased and IMPs tone gets more strident. After a certain threshold, the harmonicity ratio on the first order modulation begins to change to a non-harmonic ratio as well, which can get quite crunchy. This direct relationship between the timbre of the human input and the timbre of IMP is the way I prefer to play with IMP, but it is entirely dependent on how the Wekinator is trained. Different training data can produce very different results.

Demonstration

The demonstration includes the author playing trombone with IMP, while varying each of the MPS in somewhat extreme ways to make the effect of each setting clear. The demo concludes with a short performance by the author with IMP.

Performance video is available here: http://www.youtube.com/watch?v=Yul5FTf7pJc

Acknowledgements

I would like to thank Dr. Stephen David Beck and Dr. Jesse Allison for their advice and guidance. IMP could not have come to be without the input of the musicians who helped me test the software: Janna Saslaw, Mark McGrain, Ray Moore, Rick Trolsen, and Brad Walker. Thank you.

References

Albert, J. V. 2013. *Interactive Musical Partner: A System for Human/Computer Duo Improvisations*. Dissertation, Louisiana State University, Baton Rouge, LA.

Assayag, G.; Bloch, G.; and Chemillier, M. 2006. OMAX-OFON.

Borgo, D. 2005. *Sync or Swarm: Improvising Music in a Complex Age*. New York: Continuum International Publishing Group.

Chowning, J. 1985. The synthesis of complex audio spectra by means of frequncey modulation. In Roads, C., and Strawn, J., eds., *Foundations of Computer Music*. Cambridge, MA: The MIT Press. 6 – 29.

Fiebrink, R. 2011. *Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. Dissertation, Princeton University, Princeton, NJ.

Jehan, T., and Schoner, B. 2001. An audio-driven, spectral analysis-based, perceptual synthesis engine.

Lewis, G. E. 2000. Too many notes: Computers, complexity and culture in "Voyager". *Leonardo Music Journal* 10:33–39.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32(3):333–341.

Risset, J.-C., and Duyne, S. V. 1996. Real-time performance interaction with a computer-controlled acoustic piano. *Computer Music Journal* 20(1):62–75. ArticleType: research-article / Full publication date: Spring, 1996 / Copyright 1996 The MIT Press.

Teitelbaum, R. 2006. Improvisation, computers, and the unconscious mind. *Contemporary Music Review* 25:497–508. 5/6.

Appendix: The Software Archive

The IMP software package is archived at http://research.jeffalbert.com/imp/. The most recent version is available, along with any older variants, links to related publications, and available audio and video of performances with IMP.