

# **Computers Composing Music: An Artistic Utilization of Hidden Markov Models for Music Composition**

By Lee Frankel-Goldwater

Department of Computer Science, University of Rochester

Spring 2005

## Abstract:

Natural systems are the source of inspiration for the human tendency to pursue creative endeavors. Music composition is a language for human expression and can therefore be utilized in conveying the expressive capabilities of other systems. Using a Hidden Markov Model (HMM) learning system, a computer can be taught to create music that is coherent and aesthetically sufficient given the correct tools. The tools selected for this project include: twenty-two years of sun spot data as the natural system from which to creatively draw; a compositional framework for structure, pitch, dynamics, and rhythm to facilitate a human understanding of the system's expressiveness; the jMusic<sup>1</sup>, open source, music composition software; and an HMM learning system<sup>2</sup> with implementations of the Forward-Backward, Viterbi, and Baum-Welch algorithms. In composing a final piece of music the attempt was made to impose as few creative restrictions on the system as possible. Through these tools every aspect of the composition's generation can be repeated. In this way the robust analytical capabilities of the system are displayed via the piece and its generative procedures, thereby displaying an artificial intelligence's potential for music composition and perhaps larger creative projects.

---

<sup>1</sup> jMusic, <http://jmusic.ci.qut.edu.au/>

<sup>2</sup> Tapas Kanungo, <http://www.cfar.umd.edu/~kanungo/software/software.html>

### Artistic Motivations:

The artistic intentions behind this project are as follows:

- Nature has discernable patterns within its manifestations
- Patterns are interpreted on a basis of past experience
- Music composition is driven by a need to express relationships

A clear empirical observation is that natural systems exhibit certain repetitive and cyclic properties. Human beings interact with their environment on a constant basis and one of the greatest skills we exhibit is the ability to recognize and learn from the patterns that occur within that environment. What is particularly interesting is that we seem to be able to observe a pattern within nature and then apply the observations about that series of relationships to other, seemingly unrelated tasks.

Engineering, mathematics, psychology, and any science; martial arts, painting, music composition, and any artistic endeavor appear to be exhibitions of the human capacity to interpret nature. Inquiries into the basis of this ability are at the heart of some of the greatest pursuits of philosophical study. What is already apparent however is that humans utilize past experience to interpret newly recognized patterns and to build on their base of knowledge.

It is also interesting to observe that part of the human condition is the creative urge. The need to express ones thoughts in some medium, whether it is through engineering or music composition, is a universal trait. One artistic assumption that was made during this project is that this human creative urge stems from a natural desire to express our interpretations of the patterns and relationships that we have recognized in nature. This

does not appear to be an inappropriate leap in logic because the fuel for creativity must come from nature in some capacity. Without learned concepts and patterns we would have no basis for creation.

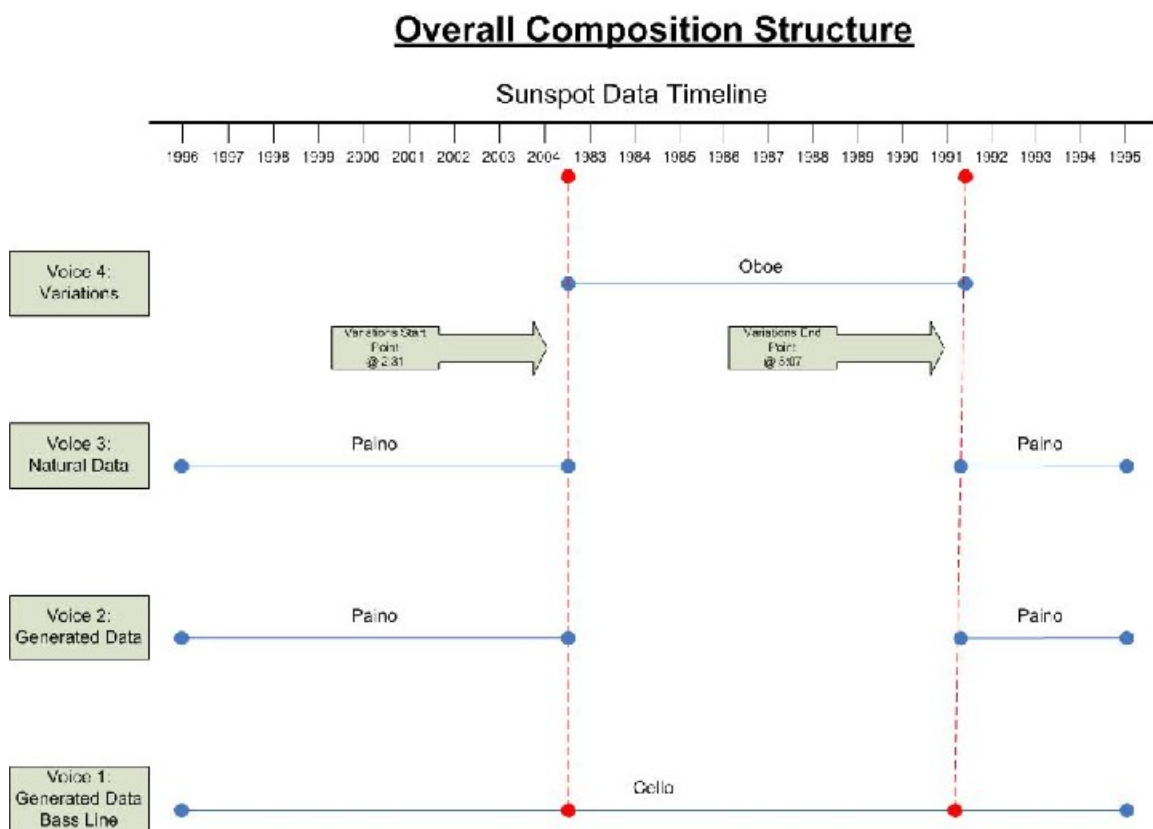
Therefore, it seems possible to imbue an artificial system with the tools necessary to execute a similar process of creation through the medium of music composition; utilizing the proper learning tools (Hidden Markov Models) and with a person to guide the creation and transmission of the product to listeners (jMusic and the composer).

#### Tools:

Two main software resources were used in the course of this project. The first is a Hidden Markov Model (HMM) software implementation of the Forward-Backward, Viterbi, and Baum-Welch algorithms. This is a somewhat ideal tool for the creation of HMM's that can be manipulated and reused to generate data from ambiguous observation sets. It can be easily installed on any UNIX system and generates output to the command prompt that can be piped into whatever file is desired. There are two main executables of note in the package. The first is called 'esthmm' and is the HMM generator. It takes for input a number of states, a number of symbols, and an observation filename which all correspond to the necessities for generating an HMM. It also takes for input an optional random number seed. This guarantees generation repeatability. The second tool is called 'genseq' which generates sequences from an HMM. This also allows a seed for repeatability reasons.

The second tool is jMusic, an open source, java based, algorithmic composition software. As the author accurately describes, "jMusic is a project designed to provide

composers and software developers with a library of compositional and audio processing tools. It provides a solid framework for computer-assisted composition in Java, and is also used for generative music, instrument building, interactive performance, and music analysis.”<sup>3</sup> This is an extremely useful set of tools that are more robust than was necessary for this project. Yet, it was still intuitive to use and had every resource required. The structure is built around a note object that gets added to successively larger collections of notes within other object structures. The primary use for jMusic in this project was used to manipulate the data compositionally once the process with the HMM tools was complete.



*Diagram A: Compositional Structure Flow chart*

<sup>3</sup> jMusic, <http://jmusic.ci.qut.edu.au/>

### Compositional Structure:

The structure of this piece was designed to make evident to the listener the pattern recognizing and the expressive capabilities of the HMM system. The flowchart in Diagram A is a visual example of the structure. It is a four voice piece built around the data generated by the HMM system overlaid with the original observation data. This data is a 22 year collection of sunspot data from 1983 to 2004 gathered from the World Sunspot Index.<sup>4</sup> Sunspots are a series of dark blotches that occur on the surface of the sun and are studied because of the magnetic phenomena surrounding them. Sunspot data was chosen because the spots seem to have a cyclic quality across a 22 year period. Note that the piece was rotated so that it begins in 1996 for aesthetic reasons.

Beginning in 1992 measurement techniques became more advanced and so there is an increased amount of data starting in this year. Voices 2 and 3 utilize an extra two columns of data provided in the years 1992 through 2004 for the purpose of varied rhythms and dynamics. The pitch material for all of the voices of the piece comes from the first column of data with runs from 1983 through 2004 (See Diagram B below).

Voice 3 utilizes the original sunspot data as a source for pitch, rhythm, and dynamic material; voice 2 utilizes HMM generated data to determine the pitch, rhythm, and dynamics; the bass line, voice 1 uses HMM data for the pitch material, but the rhythm and dynamics are static. This choice was made so that the listener has some sense of meter because the piece has specific meter. The variations voice, voice 4, is built over the section of the piece where the material for voices 2 and 3 is incomplete because of the year. While this voice is built off of the same pitch material as the line below it, it utilizes a more staggered structure that is apparent to the listener.

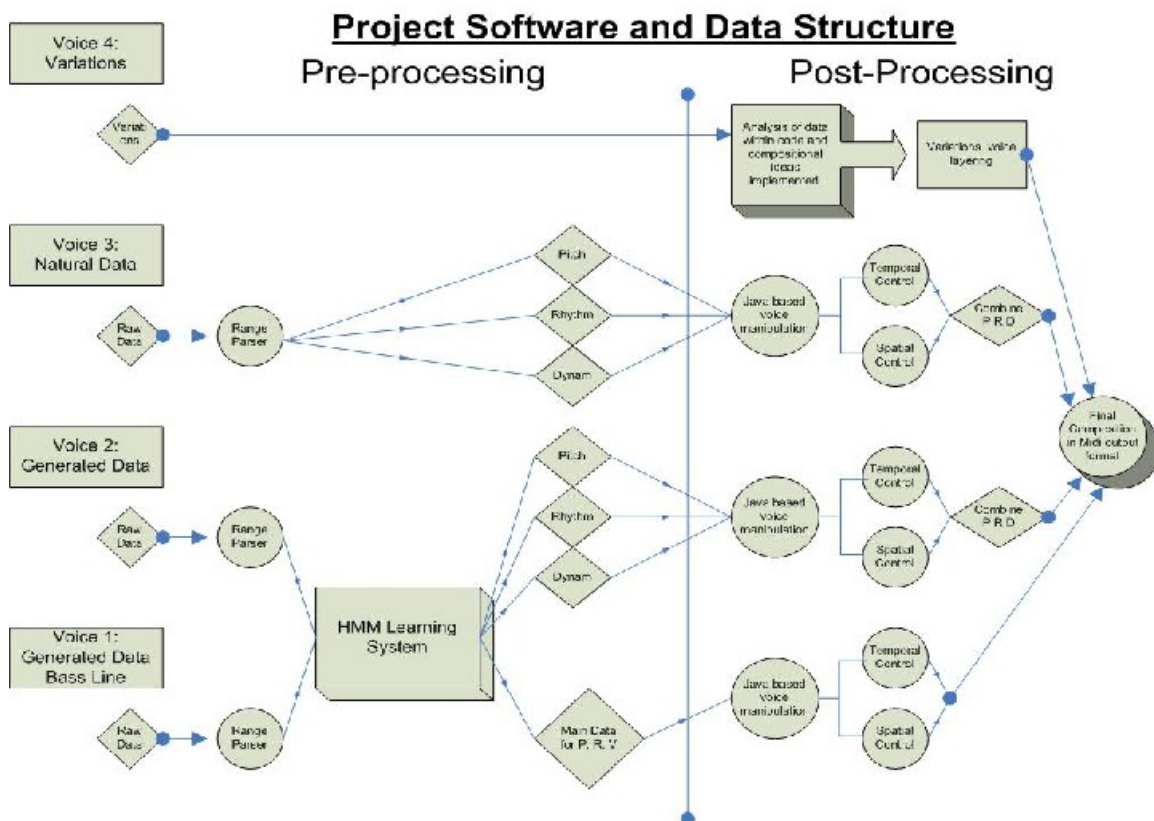
---

<sup>4</sup> SIDC World Sunspot Index, <http://sidc.oma.be/index.php3>

The tempo for the final form of the piece is 5000 beats per minute to facilitate a reasonable overall length across 22 years of data. The scale that was chosen to filter the pitch data through for the final form is pentatonic. It is believed that this scale allows the listener to further appreciate the aesthetic qualities of the piece. The chromatic version however facilitates a better sense of the cyclic motion naturally found within the data.

Most of the filtering was done linearly. A change in the data by one notch would affect the pitch, rhythm, or dynamics a single notch where appropriate. All layers of data were kept lined up by day so that for example the pitch data in voice 4 for April 3<sup>rd</sup>, 1995 will temporally line up with the same pitch data in the base line.

In these ways delineated above, the hypothesis is that the listener should be able to hear patterns found by the HMM system both melodically and harmonically.



*Diagram B: Software Design Flow Chart*

### Program Structure:

The program structure is well defined by diagram B shown above. A set of raw data from the sun spots is first parsed into the proper ranges for use in the music. Sunspot data had a range within a given column as high as 246 through 0. This is too great to be applied to the Western pitch system. The 'Range Parser' tool specified in the diagram changes these ranges in a balanced way to protect the cyclic structure of the data, while allowing for a reasonable range to work with compositionally.

Relative to the appropriate voice, the parsed data is then fed through the HMM learning system, saved, and then generation sequences were made based on the HMM's interpretation of the natural data. There was quite a rigorous stage of data manipulation in this section of the project that could have been more easily facilitated by an entirely Java based system.

The proper columns of data were then formatted for use in Java and passed to the proper voice controllers as specified in the previous section. Each voice was generated independently of the others. Temporal and spatial relationships were checked so that the patterns within data would remain in tact and the artistic intent maintained. In this stage of the post-processing the variations voice was compiled and collected with the other voices in preparation for exporting.

jMusic facilitates a convenient conversion of musical data into standard Midi format. Thus a completed piece of music is created. It is important to note for artistic and scientific purposes that all aspects of this project, including all generations and combinations, are repeatable using the methods specified in this paper and the seed value

0605. This number was chosen because June 5<sup>th</sup> is the composer's birthday and 1983, the starting year of the spot data is his birth year.

### Conclusions:

The creation of this piece was a pleasure to facilitate. Next time, I would like to work on building more co-dependence between the voices and further elaboration on the variations section. As the bibliography will show, not much information is to be found on HMM composition. Most of the work with HMM's is in the areas of musical transcription and recognition. As this project shows, the potential for an HMM composition system is quite viable and applications for composition are available for expansion. The composer can choose any level of utility for the system; from thorough piece creation as with this project, to simply using it as a source for compositional material.

It is clear to the authors that an AI system has the potential to be highly expressive given a robust enough base of experience and a more advanced ability to communicate with human beings.

### Acknowledgements:

To my advisor Chris Brown for his leadership and valuable lessons in AI and group work.

To Josh Mailman for his compositional advise.

To Mark Bocco for allowing me to use the Electrical Engineering music facilities.

To the various other University of Rochester and internet resources that gave me assistance.

And of course, to my Mother... just because.



## Annotated Bibliography

### *Music and Artificial Intelligence (1993) by Chris Dobrian*

Intelligent musical behavior, whether in cognition, performance, or composition, usually involves use of more than one process simultaneously or sequentially. Good examples of how to turn input into interesting musical variations.

### *An Introduction to Music and Artificial Intelligence by Eduardo Reck Miranda*

It is debatable whether musicians want to believe in the possibility of an almighty musical machine. Musicians will keep pushing the definition of musicality away from automatism for the same reasons that scientists keep redefining intelligence. Nevertheless, AI is helping musicians to better operate the technology available for music making and to formulate new theories of music.

### *Research in music and artificial intelligence by Curtis Roads*

Although the boundaries of AI remain elusive, computers can now perform musical tasks that were formerly associated exclusively with naturally intelligent musicians. After a historical note, this paper sermonizes on the need for AI techniques in four areas of musical research: composition, performance, music theory, and digital sound processing. The next part surveys recent work involving AI and music. The discussion concentrates on applications in the four areas of research just mentioned. The final part examines how AI techniques of planning and learning could be used to expand the knowledge base and enrich the behavior of musically intelligent systems.

### *The Age of Intelligent Machines: Artificial Intelligence and Musical Composition by Charles Ames*

Gives a basic overview of the compositional techniques used by AI composers to create music. The general outlook is a proposal of creativity and its relationship to AI music. It is published on the Kerzweil AI site and provides links to other useful resources.

### *BoB: an Interactive Improvisational Music Companion by Belinda Thom*

The goal of this research is to build an agent that learns from a player and provides accompaniment. It is also stated that it wants to create an agent that is fun to play with, to build a believable improvisation music companion. The agent has interesting implications on the values of a positive human appeal to the output. Is that what the goal of AI music composition is?

### *GenJam in Transition: from Genetic Jammer to Generative Jammer by John A. Biles*

This paper considers GenJam as a generative art system. Generative art produces “unique and non-repeatable events” that express a designer’s generating idea. The

designer's generating idea defines a *species* of events, represented in a genetic code. In music, these events could be individual notes, melodic phrases, and even entire pieces. In GenJam the events are four-measure phrases, or "licks" in the jazz vernacular.

*Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness by John A. Biles*

This paper focuses on a successful attempt to eliminate the fitness bottleneck in GenJam, an interactive genetic algorithm that performs jazz improvisation, by eliminating the need for fitness. This was accomplished by seeding an initial population of melodic ideas with phrases selected from a database of published jazz licks, and employing an intelligent crossover operator to breed child licks that tend to preserve the musicality of their parents. After a brief overview of the changes made to GenJam's architecture, the paper describes the mapping of licks to measure and phrase individuals in GenJam Normal Form. The intelligent crossover operator on phrases is then described, along with a discussion of measure mutations performed during a solo to insure that repeated measures are altered in interesting ways. The paper concludes with a discussion of the implications of removing fitness from a genetic algorithm and whether the result still qualifies as a genetic algorithm.

*GenJam: An Interactive Genetic Algorithm Jazz Improviser by John A. Biles*

The subject of this article is a software sideman and featured soloist in the Al Biles Virtual Quintet. It is able to improvise solos in real time on arbitrary tunes, and it can trade fours or eights interactively with a human soloist. This article briefly describes how GenJam works, how it carries on conversations with human soloists, and what it is like for the only human member of the Virtual Quintet to play gigs with it.

*An Introduction to Hidden Markov Models by Rabner & Juang*

The basic theory of Markov chains has been known to mathematicians and engineers for close to 80 years, but it is only in the past decade that it has been applied explicitly to problems in speech processing. One of the major reasons why speech models, based on Markov chains, have not been developed until recently was the lack of a method for optimizing the parameters of the Markov model to match observed signal patterns. Such a method was proposed in the late 1960's and was immediately applied to speech processing in several research institutions. Continued refinements in the theory and implementation of Markov modeling techniques have greatly enhanced the method, leading to a wide, range of applications of these models. It is the purpose of this tutorial paper to give an introduction to, the theory of Markov models, and to illustrate how they have been applied to problems in speech recognition.

*Computer Generated Music Composition by Chong (John) Yu*

A computer composition engine has been designed in an attempt to capture basic music composition and improvisation knowledge in a variety of styles. The output is based

solely on user-controlled parameters and low level rules embedded within the generation engine. Although the generator itself is platform independent, current versions exist for both Windows and Java, using MIDI and sound file output respectively.

*Grammar Based Music Composition by Jon McCormack*

Using non-deterministic grammars with context sensitivity allows the simulation of Nth-order Markov models with a more economical representation than transition matrices and greater flexibility than previous composition models based on finite state automata or Petri nets. Using symbols in the grammar to represent relationships between notes, (rather than absolute notes) in combination with a hierarchical grammar representation, permits the emergence of complex music compositions from a relatively simple grammars.

*Creating Musical Compositions with JAVA by Michael Newton*

Java provides support for Musical Instrument Digital Interface (MIDI) programming with the Java Sound API. This study investigates the facilities that Java provides for MIDI programmers and looks at using other Java packages such as Java Swing for interface design and implementing user interaction. Java's suitability for building MIDI musical compositions will be demonstrated by producing a Simple Music Composer Application allowing basic facilities for music construction, playback using MIDI synthesis and the saving of musical compositions. The investigation highlights design issues that are involved with interface design and MIDI programming. These include constructing musical notation, printing a musical score to paper, conforming to MIDI standards to save and load Standard MIDI Files and producing musical notation from a given MIDI file. The results of the development work and the research behind the Simple Music Composer is presented together with the specification, design and implementation details in this report.

*jMusic: Music Composition in Java by Andrew Sorensen and Andrew Brown*

jMusic is a project designed to provide composers and software developers with a library of compositional and audio processing tools. It provides a solid framework for computer-assisted composition in Java, and is also used for generative music, instrument building, interactive performance, and music analysis. jMusic supports musicians with its familiar music data structure based upon note/sound events, and provides methods for organizing, manipulating and analyzing that musical data. jMusic scores can be rendered as MIDI or audio files for storage and later processing or playback in real-time. jMusic can read and write MIDI files, audio files, XML files, and its own .jm files; there is real-time support for JavaSound, QuickTime and MIDIShare. jMusic is designed to be extendible, encouraging you to build upon its functionality by programming in Java to create your own musical compositions, tools, and instruments. In a spirit of mutual collaboration, jMusic is provided free and is an open source project. jMusic is 100% Java and works on Windows, Mac OS, Linux, BSD, Solaris, or any other platform with Java support.