Algorithmic Music Composition and Interactivity in the MIRAG: Musically Intelligent Robotic Algorithmic Guitar.

Chester Udell A.B.D.
University of Florida
2911 sw 13th st
Gainesville, FL 32608
cudell@ufl.edu

Dr. Antonio Arroyo
University of Florida
Machine Intelligence Lab (MIL)
MAEB 321, UF
Gainesville, FL 32611
arroyo@mil.ufl.edu

Dr. Eric Schwartz
University of Florida
Machine Intelligence Lab (MIL)
MAEB 321, UF
Gainesville, FL 32611
ems37uf@gmail.com

ABSTRACT

The MIRAG (Musically Intelligent Robotic Algorithmic Guitar) is a "musically autonomous" robot encompassing an eclectic variety of disciplines including: 17th century counterpoint, Brazilian folk guitar, algorithmic music composition, proximal human-robot interactivity, and color tracking. Inspired by Pat Metheney's *guitarbot* built for his Orchestration project, the MIRAG consists of 3 monocords (platforms on which individual strings and actuators are mounted) that play note events. These note events are generated by a central brain processor using rules of 17th century counterpoint (step-by-step rules that determine how to construct music note-for-note that composers like J.S. Bach would have been familiar with). The "brain" microprocessor takes in environmental data like human proximity, ambient light, and the color of the human observer's shirt (red, green, or blue) to influence musical parameters as it composes a musical improvisation in real-time and sends note event commands to "reflex" microprocessors that drive the actuators and strings.

Keywords

Algorithmic music, musical intelligence, autonomy, proximal interactivity, counterpoint, color tracking.

1. INTRODUCTION

The MIRAG (Musically Intelligent Robotic Algorithmic Guitar) is a musically autonomous robot encompassing an eclectic variety of disciplines including 17th century counterpoint, Brazilian folk guitar, algorithmic music composition, proximal human-robot interactivity, and color tracking. Inspired by Pat Metheney's *guitarbot* built by LEMUR[1] for his Orchestration project, the MIRAG consists of 3 monocords (platforms on which individual strings and actuators are mounted) that play note events. A central brain processor uses rules of 17th century counterpoint to generate note events. Counterpoint is a discipline that defines step-by-step rules that determine how to construct music note-for-note. Composers like J.S. Bach and Palestrina were extremely familiar with these rules.[2] The brain also takes in environmental data like human proximity, ambient light, and the color of a human observer's shirt (red, green, or blue) to influence musical parameters as it composes an improvisation in real-time and sends note event commands to "reflex" microprocessors to drive the actuators and strings.

Look Ma… No Score! Most "orchestrions" (i.e. machines designed to play music autonomously) are pre-programmed with musical scores (a stored list of note events to execute over time that comprises a "song"). In other words, music that is played by these machines to a great extent is already *predetermined*. What is notable about the MIRAG is that it has no pre-programmed score and no preconceived musical events or sequences, thus all of the sounding events are improvised or generated on-the-fly based on a sophisticated set of algorithms that literally compose music in real time.

2. Integrated System

2.1 Brain Processor

A single brain processor receives all of the incoming sensor data and also contains:

- Vectors with musical modes and scales
- Algorithms to generate 3-voice counterpoint
- Timers dedicated to control tempo and subdivision
- A wireless RF communication protocol to query a nearby computer for color data.

In between beat/note executions, the brain generates new note messages based on the counterpoint algorithm and the state of the environmental sensor data. The beats themselves trigger an interrupt service routine. The ISR assembles the generated notes into data packets and sends them via SPI to the reflex processors.

The peripheral requirements and the author's previous experience lead toward the selection of Texas Instrument's MSP 430 F2272[3] to serve as the brain and reflex microprocessors.

2.2 Sensor Suite

The sensors used in this system detect human proximity and ambient light and also facilitates RF communication. Maxbotix Sonar, iSight camera, Photo resistor, and Nordic transceivers comprise the overall sensor system. The table below lists the sensor suite's function:

Table 1. Sensor Suite

Sensor	Assignment
Maxbotix LV-EZ3[4]	Detect People Proximity: trigger color track wireless command
Photo-resistor	Ambient light: Tempo
Bump Sensors	Calibration Sequence: detect when nodes arrive at fret 0.
Nordic nRFL01+[5] wireless transceivers	Facilitate wireless communication between MIRAG and computer
iSight camera	[on computer] color tracking

2.3 Actuation Platforms

The actuation platforms look very similar to the neck of a guitar with a nut and bridge to suspend 2 strings tuned either in octaves or unison. The distance between the nut and bridge (known as the "saddle") is 580 mm in total length. The platforms also each have a Stepper motor, a servo to pluck the string back and fourth, a servo to dampen the string, a pick-up for amplification. The microprocessor used to supply the actuation signals and the stepper motor for each platform is located on the circuit board directly underneath the brain processor. An MSP430 F2272 microprocessor interfaced with a stepper motor driver via GPIO drives the stepper motor. A logical pin sets direction and the rising edge of a pulse initiates 1 step. Another logical pin sets the direction up or down. One bump sensor is placed near the nut behind "Fret 0" so that the reflex can find the origin point automatically during its self-calibrate startup routine.

The stepper motor drives a belt that runs parallel to the string. A dowel, guided by a track, is attached to the belt and rests with just enough force under the string to act as a "node." There appears to be undue slack in the belt, but this is desirable for mechanical reasons due to picking. If the tension in the belt is too great, this elevates the string too high for the pick to be effective. The position of this node will determine the pitch. The 12th root of 2 principle was used to determine the location of each respective semi-tone on the string.

Servo 1 is controlled via the Timer_A PWM outputs. A guitar pick is attached to the servo and each "strum" is executed by moving the servo back and fourth.

Servo 2 serves as a dampening mechanism controlled via the same Timer_A PWM outputs, but on a separate Capture-Compare register. This is positioned at the bottom of the string and will dampen the string – if desired – as the stepper motor moves the node mechanism to change pitch. This will prevent unwanted "glissando" from one pitch to the next when the node is in motion. The dampen command can be "ignored" if glissando is a desired effect.

An incoming music actuation packet looks like [command, pitch]. Upon receipt, Servo 2 dampens the string unless it should be ignored and immediately moves the stepper motor to target position (i.e. calculate its current position via signed arithmetic to determine how many steps away target position is) MSB is tested and determine the GPIO of the stepper motors logical direction pin. Then it will invert all other bits, add 1 to determine distance if MSB==1. Once in position, Servo 2 will un-dampen (if dampen > 0) and then servo 1 will pluck the string. The strings themselves come from the Brazilian folk guitar type: *Viola Caipira*. They were chosen for their bright and unique sound quality. In the Spring 2011 semester, the author collaborated with Paul Pino to construct new platforms made of Plexiglas using a laser cutter.

2.4 Remote Computer

A remote computer can be set up oriented towards an approaching human observer equipped with a Nordic RF transceiver, iSight camera, and color tracking software. The RF transceiver receives commands from the MIRAG and transmits color data back to the brain for use. Additional to the color data is also the X/Y position of the color's centroid. This may be mapped in the future to other musical parameters. The color tracking software was developed by the author in a program called MaxMSP/Jitter. To adhere to the autonomous nature of the MIRAG, this unit is not required for the MIRAG to operate in default mode, but is essential for the MIRAG to operate at full interactive capacity. In other words, if the remote computer unit shuts down or the RF connection is severed, the MIRAG continues to improvise music using default settings until the connection is reestablished.

2.5 Amplification

The sound of the strings had to be amplified, as the platforms do not generate a loud enough sound themselves. To do this, rare earth magnets were attached to the top of 33mH inductors directly underneath the vibrating string, which connect to an audio jack. Plugging this into a standard guitar amplifier results in an effective method to pick-up and amplify of the strings. Amplification was the least critical part of the system during the development process, but in hindsight: amplification significantly increased the tone quality of the instrument and reduced the unwanted mechanical noise.

3. Intelligence

3.1 Music Algorithm

The onboard musical intelligence will be on the same MSP430 F2272 microprocessor that the sensors are connected to. The sensor data will be used as parameters to map onto musical parameters out of which the microprocessor will generate music actuation message packets to send over SPI to the Actuation or "Reflex" processors.

The general program flowchart of the brain is detailed at the end of this section. The counterpoint Algorithms are fairly complex and use a weighted probability paradigm that would be too complicated to outline in flowchart form. The more voices that are implemented (MIRAG currently used 3), the significantly more complex the counterpoint protocol must become to accommodate the various possible states each voice can be in to account for the types of harmonic combinations.

Scale/Mode: Scales are pre-defined in vectors in memory (arrays). Scales can be broken down into "semitones away from tonic" such that a major scale will be:
int Array_Major[7] = [0,2,4,5,7,9,11]; // Do, Re, Mi, Fa, Sol, La, Ti (do). The position of these numbers then becomes the "Scale Degree" number. With this abstraction, the counterpoint algorithm only needs to generate scale degree numbers alone. These are latter mapped onto the actual scale and Tonic, but because these can be selected independently of the counterpoint system – it is possible to change these parameters on-the-fly.
The actual pitches are generated when these numbers are added to the tonic note. Middle C on the piano is a MIDI note of 60. As you can see, a database of several of these scales/modes can be a parameter that I plan to map to the color of a persons shirt. Color of a shirt determines "color" of the scale.

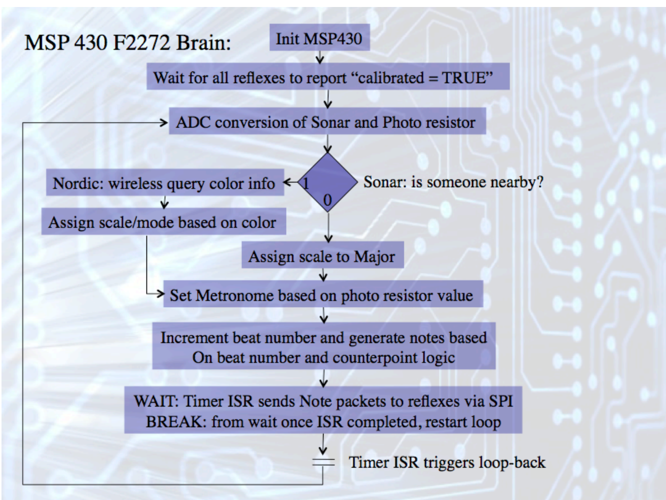
Harmonic structures: Tonal harmony is based on the triad. However, there is no need for harmonic array types to be defined. Principles of counterpoint were employed centuries before Tonal Harmonic theory (see Palestrina and Corelli). Counterpoint operates on intervals above the bass and terms of consonance versus dissonance. Our tonal-exposed ears perceive this as tonal triadic harmony even though the algorithm accounts for no such thing in principle!

Rhythmic structures: For now, the rhythms of the MIRAG is extremely static: the lower 2 voices play in half notes and the top voice plays twice as often (quarter notes). What can be easily implemented in the future on top of the 14th C. Mensuration principles for meter is just to simply (probabilistically) filter out whether or not a beat should play for each given note onset.

3.2 Anticipation

Just like a human performer who calculates ahead to determine what one will do in the near future, the MIRAG will calculate the notes it is going to play before it sends the message to instantiate the actual note. This solves significant timing issues. For example, you are playing the piano and you recognize you will soon need to move your hand from the bottom keys to the top key far away. Right after your low note is done, you immediately move to the next one before it is time to actually play it to ensure you are at the right pitch on time. This same principle holds for MIRAGE.

3.3 Flowchart



4. Behaviors

Wandering:

When no person is in proximity of the sonar, MIRAG generates counterpoint in the major mode.

Someone nearby:

When the sonar detects a subject as nearby, MIRAG will immediately request information about the color of the person's shirt from the iSight attached to the computer via Nordic RF Transceiver. It will use this information to determine the nearest color class of the shirt which will determine overall global musical parameters.

Blue: This is mapped to the 5-note pentatonic scale and is often associated with the orient and eastern music. This is the only mode where counterpoint principles are very loosely followed and omits phrase structure (it does not pause after so many notes and does not cadence).

Green: This is mapped to the Dorian mode, contrapuntal principles and phrase structure are normal.

Red: Mapped to the Hungarian minor scale – also known as the Gypsy scale.

Others colors will be mapped in the future and the protocol will be expanded to incorporate other musical style parameters.

5. ACKNOWLEDGMENTS

Special thanks to Paul Pino (Plexiglas design), Karl Gugel, Antonio Arroyo, Eric Schwartz, Mike Braddock, James Paul Sain. TA's: Tim Martin, Ryan Stephens, Thomas Vermeer, Devin Hughes

Monique Udell, my wife (for putting up with loosing both her husband and the dining room table for the IMDL semester).

6. REFERENCES

[1] "LEMUR: Purveyors of Fine Musical Robots Since 2000." [Online]. Available: http://lemurbots.org/. [Accessed: 11-Apr-2011].

[2] H. (Harold O. Owen, *Modal and Tonal Counterpoint: From Josquin to Stravinsky*, 1st ed. Schirmer, 1992.

[3] Texas Instruments, "MSP430x2xx datasheet." SLAU 144e-2008.

[4] inc Maxbotix, "LV-MaxSomar-EZ3-Datasheet." Maxbotix inc., Jan-2007.

[5] S. Nordic, "nRF24L01+ Datasheet." Nordic Semiconductor, Mar-2008.