Final Project Report
Chester Udell 12/7/2010
MIRAG: Musically Intelligent Robotic Algorithmic Guitar
IMDL: EEL 5666
Instructors: Dr. A. Antonio Arroyo, Dr. Eric M. Schwartz
TAs: Tim Martin, Ryan Stephens, Thomas Vermeer, Devin Hughes

## Table of Contents

# Special Thanks:

Karl Gugel, Antonio Arroyo, Eric Schwartz
Mike Braddock
TA's: Tim Martin, Ryan Stephens, Thomas Vermeer, Devin Hughes
Monique Udell (for putting up with loosing both her husband and the dining room table for a semester)
James Paul Sain and Paul Koonce

# Abstract:


## Original Conception . . .

My aim for this project is to construct a *"musically autonomous"* robot with the following components:

A) This robot will play 4 or 5 separate mandolin-like guitar strings – *monochords* – each on its own platform. I will construct the sounding mechanisms myself to create a new instrument body entirely. Specifically, the scale length and string-type will be of a Brazilian folk guitar called the *Viola Caipira*.

B) Sensors: it will sense its environment by means of sonar (people proximity) and photo resistors (ambient light). It will have, for its special sensor, an Apple iSight camera that is wired to a computer for color and motion matrix analysis, which is then broadcasted via Nordic RF transceivers to the robot. On startup, each monochord will self-calibrate and notify the *Brain* processor when to proceed.

C) Intelligence: The robot will take the above sensor data and use this to inform parameters that will effect musical improvisations. For instance, color will affect parameters for scale-type, temperament (how in tune is it?), subdivision density, and other stylistic parameters. A processor is dedicated solely to this task and will compile packets of Note-on/off information to send to the actuator processors I call the "reflexes."

D) Actuation: The music packets will be sent via SPI to the reflex processors that will handle control of the servos and stepper motor drivers. For each string: 1 servo will pick the string back and fourth, 1 servo with a felt tip will dampen the string, a stepper motor will drive a belt up and down parallel to the string. On the belt will be clamped a small dowel or rod that will 'node' the string in that given position – controlling the pitch of the string. This scheme will make for very interesting musical control as not only are discrete pitches possible, but also sliding from one pitch to another (glissando), oscillating around a given pitch (vibrato), and playing notes not in a tempered scale (alternative-non-western tunings).

Given this system, through human interaction it will be possible to predict the behavior of this robot under specific circumstances, but all music heard is generated on-the-fly and not predetermined. It is part *sonification* of its environmental data and part musical improvisation.

See Diagram attached to back.

## Currently . . .

I overestimated the amount that could be accomplished in a semester's time so: only 3 monochords could be implemented instead of the 4 or 5. Also, the music algorithms are able to generate music in only one time signature (4/4) and generates counterpoint at the 2nd species level. Future work will be done to get the counterpoint up to the 4th species, add the 4th voice, vary up the rhythm, and add learning algorithms.

# Introduction:

I wanted to synthesize my current skill-sets including: music composition, music technology, algorithmic (i.e. rule-guided) music generation, human/computer interactivity, artificial "musical" intelligence, and microprocessors into a cohesive project: a Musically Intelligent Robotic Algorithmic Guitar (MIRAG).

I am familiar with the work of The League of Electronic Musical Urban Robots and their involvement with Pat Metheney's *Orchestration* project. This project extends the performative capacities of this renowned Jazz artist through the use of machines dedicated to actuating musical instruments. These machines receive MIDI messages from Metheney's guitar during live performance. MIDI messages inform what notes the machines will play for live musical accompaniment on stage. My aim for this project is to use one of the machines, the *GuitarBot*, as a conceptual model and modify it to have its own onboard musical intelligence and environmental sensors.

The "GuitarBot" by Joshua Fried consists of 4 platforms, each with a guitar string and actuators, and 1 platform as a controller mechanism. A youtube of this machine in action is here: http://www.youtube.com/watch?v=HxLTJmmyqn4
One of the more challenging and time-consuming aspects of this project is constructing the actual platforms on which the strings and actuators are mounted, as each platform needed to be an acoustic musical instrument in itself. Currently, 3 platforms have been implemented and a 4$^{th}$ has all of the necessary hardware assembled. In the future, my aim is to realize the full 4 or 5 strings on the robot. Mike Braddock was generous enough to let me come in and use his facilities to elegantly construct my platforms.

Musical behaviors can be broken down into a combination of variable musical parameters such as: tempo, density, temperament, mode/scale/key, rhythm, instrumentation, etc. The *Music Genome Project* made popular by *Pandora Radio* is a tangible and arguably successful manifestation of this premise: that musical style can be broken down into component parameters. My short-term aim for the musical intelligence of MIRAG was to program into it some basic building blocks of musical information out of which it can algorithmically generate musical improvisations based on environmental sensor data.

Despite only the 4/4 time signature being implemented, my ultimate goal is to use the principles of: 14$^{th}$ C. *mensuraion* to generate ever-changing rhythmic patterns and metrics. Contrapuntal rules from the 17$^{th}$ C. – 19$^{th}$ C. from the Harold Owen *Modal and Tomal Counterpoint* book are employed to determine chord structure, voice leading, and harmony. A sophisticated system of weighted probability arrays are continually modified during performance, so that at any given point in time there are several paths the brain can take in generating the next set of pitches. Coupled with 21$^{st}$ C. technology – this may turn out to be a truly eclectic robot.

**Where this robot breaks the mould:** Most "orchestrions" (i.e. machines designed to play music *autonomously*) are pre-programmed with musical scores (lists of note events). In other words, music that is played by these machines to a great extent is already predetermined even though there may be interactive qualities between the machine and a person. This model is ideal for most performance situations (this excludes Jazz) where particular sequences of musical events need to be carefully constructed and predictable. The MIRAG has no pre-programmed score, thus all of the sounding events are *improvised* or generated on-the-fly based on a sophisticated set of algorithms and incoming sensor data.

# Integrated System:

The system I devised consists of:

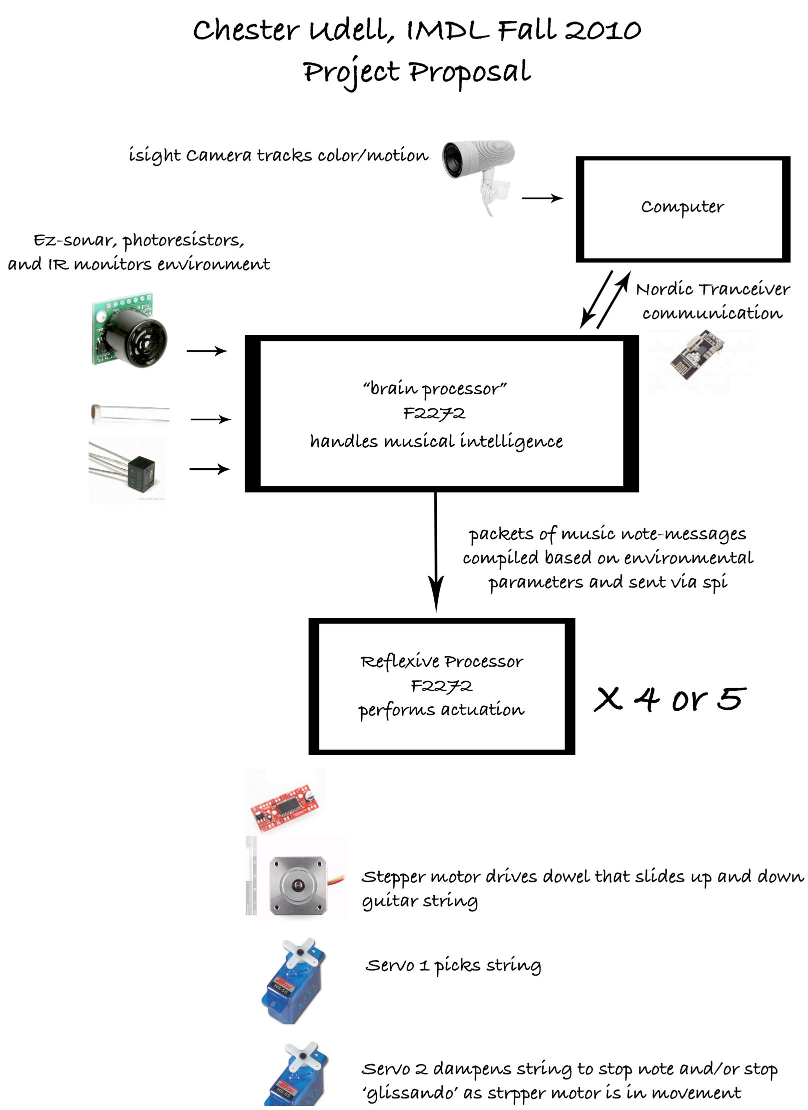
A) 1 brain processor that process all of the incoming sensor data and also contains: vectors with musical modes, and scales, algorithms to generate 3 and 4-voice counterpoint, and also timers dedicated to control tempo and subdivision, and a wireless RF communication protocol to query an nearby computer for color data.

B) 4 Actuation platforms each with a microprocessor that will receive note packets transmitted from the brain processor via SPI. Each platform will be similar to each other in construction and layout. A different pitched string will be the only general difference between them. Stepper motors will move a dowel that nodes the string up and down using a belt system. Servos will pluck and dampen the string. The Strings used will be from the Brazilian folk guitar type: Viola Caipira.

C) An Apple iSight camera will be attached to a computer that will analyze a matrix to determine motion and color. These messages will then be sent wirelessly using Nordic RF transceivers to the Brain processor.

D) The sound of the strings had to be amplified as the platforms do not have their own resonant bodies as a result of cost and time. 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.

Figure 1: Illustration of Integrated System.



# Actuation Platforms:

The actuation platforms look very similar to the neck of a guitar with a nut and bridge to suspend 2 strings tuned in octaves. The distance between the nut and bridge (also known as the "saddle") is 580 mm in total length. The platforms also each have a Stepper motor, 1 Servo to pluck the string back and fourth, 1 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 this 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.

1 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.

A 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 the "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 *$12^{th}$ root of 2* principle was used to determine the length of each respective semi-tone.

Servo 1 will be interfaced to the same microprocessor and will be controlled via its Timer_A PWM outputs. A guitar pick will be attached to the servo and each "strum" will be 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 [pitch, duration]. 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 will be 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.

There will be a Duration variable implemented in the future to determine how long the string will remain sustained until dampened.

# Sensors:

Figure 2: Sensor Suite.

| Sensor Combination | Musical Parameter |
|---|---|
| iSight camera + Nordic Transceiver | |
| • General Color | Scale/Mode (i.e. “color”) – Also determines some “style” characteristics. |
| • Physical position L/R | Not yet mapped, but perhaps could be mapped to ‘note density’ and ‘transposition’ latter |
| EZ-Sonar: Proximity | Is someone nearby? True = query computer for color info and map to scale. False = play in Major mode, no query. |
| Photo-resistor (ambient light) | Tempo: rate of beat generation |
| Bump Sensors | Each monochord self-calibrates on startup and tells the Brain when to proceed. |

The sensors used determine human presence (proximity), Color of a subjects clothing (the shirt), motion, and ambient light. This provides MIRAG with and “awareness: of a combination of ambience from the environment (that may or may not be manipulated by human observers) and human presence. Another MSP430 F2272 will handle the ADC processing as well as generating musical messages based on its internal intelligence (discussed in the next section).

The MAX EZ Sonar sensor interfaces directly with an ADC pin and determines whether or not someone is standing in front of it (i.e. paying attention to it) and how close the person is. This will help in determining its musical behavior: if someone is within “proximity” it will begin prompting a computer to send information about the color of the subject’s shirt and position via a Nordic RF Transceiver.

An Apple iSight is connected to a computer and a video processing software called *Jitter* will be used to parse out the appropriate information. It will only transmit information to the MIRAG via Nordic transceiver if and only if it is requested. This is acting as my special sensor for the project.

A Photo-resistor will be attached to an ADC pin and will measure ambient light levels of the space. This way, there are some determinate-yet-aleatoric (pseudo-random) variables out of which the MIRAG will create variety in its musical decisions (it will play differently in a dark room, on stage, outside, etc). this is mapped directly to Tempo for now and several more photo resistors may be used in the future to determine other musical parameters.

# Intelligence:

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 $14^{th}$ C. Mensuration principles for meter is just to simply (probabilistically) filter out whether or not a beat should play for each given note onset.

**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.

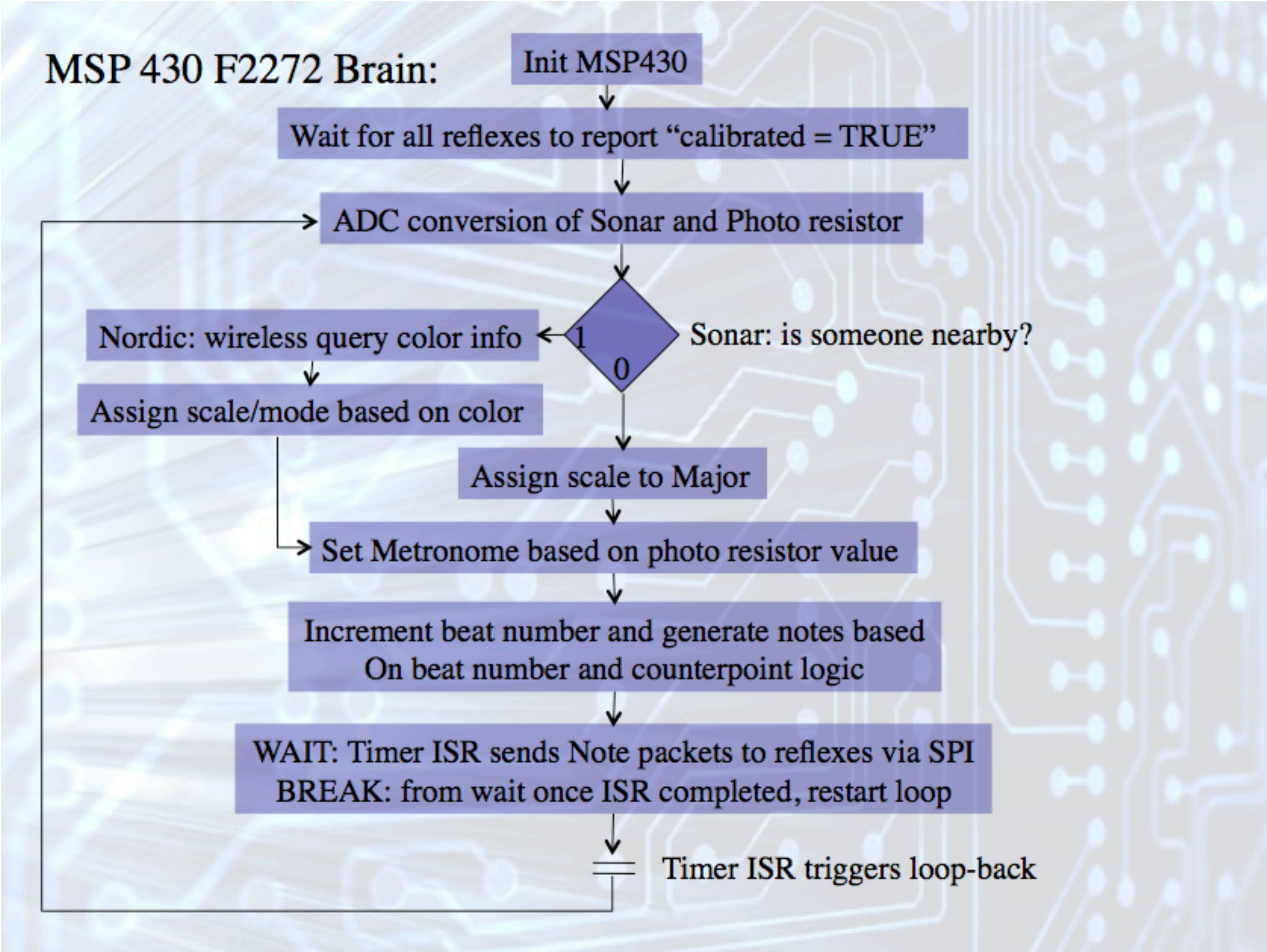



Figure 3: Brain Flowchart.

# Musical 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.