Musical Expressions: Synthesizing Music from Facial Expressions

Sameeksha Katoch, Anik Jha
School of ECEE, Arizona State University

***Abstract*—The aim of the project is to build an experiential system to create emotional awareness, of a given person, by exploiting his/her facial expressions. The system primarily focusses on the expressions generated by the movement of key facial components namely mouth, eyes and eyebrows. The expressions targeted in the project include scream, smile and surprise. A distinguished audio feedback is generated based on the class of expressions to create the awareness.**

***Index Terms*—Face Expressions, Additive Synthesizer, FaceOSC, Machine Learning**

I. Introduction

Sound theory as well as design principles individually have had a significant impact in enabling the understanding of human reactions under different testing conditions. However, this project intends to explore the possibility of their amalgamation to further enhance their utility through an audiovisual tool. Importance of facial expressions in person-to-person interaction is one of the strongest motivation for development of this system. Apart from this, facial expressions are an important aspect of many dance forms like mime, kathakali etc. Their importance cannot be mentioned enough to bring life-like aura in sculptures and paintings.

While appreciating the possibility of modifying/substituting the existing visual notion or perceptions with the new sonic signatures for certain expressions, the project also explores an alternative communication means using sonification. It is analogous to creating a new dictionary where expressions are the words and the meaning is the sonic output which can alternatively be used as a feedback depending on the application involved.

The system performs online sound generation based on the facial muscle movement. The feature vector taken into account is utilized to train a machine learning model which classifies facial expressions into three different categories i.e. scream, surprise and smile and uses these classes to produce complex sound timbres in real time.

The paper discusses the development of the tool, motivation for developing the system and its potential applications.

II. Related Work

There has been an ample amount of research in studying the face expressions. However, there are only a certain number of papers that talk about a sonification system using the facial input. [1], [2], [3] and [4] explore the idea of synthesizing music using eye and mouth movements. [1] poses the solution for the noise and control issues that surface when using eye movements for generating music. Rather than completely removing the noise due to eye tremors the musicians tend to make specific compositions using that noise. [4] extract shape parameters from the mouth opening and output these as MIDI control changes. A set of action units is used to describe the facial movement in [5]. In [6] pitch of a sine wave can be set dynamically based on features computed from the data. Funk et. al. [7] associate facial movements with sound synthesis in a topographically specific fashion.

III. Experimental Setup and Methodology

The project is divided into 2 phases. Both the phases use FaceOSC to extract facial features through the movement of facial muscles. In the first phase, simple thresholding is used to define the class of expressions derived from the extracted features. Whereas, in the second phase a machine learning algorithm is used to classify the facial expressions. Post classification, based on the class definition, the expressions are mapped to the basic music generation tool in Max/MSP. The following section discusses the details of the execution of these phases.

A. Extracting Facial Features

FaceOSC is a tool developed for face based interaction. It tracks a face and send its pose and gesture data over OSC. The ”pose” and ”gesture” toggle can be used to send the pose and gestural stats. The tool gives an option to select required features out of 12 different possibilities. Based on the definition of the required classes of the facial expressions the following 5 features are selected:

- mouth width: /gesture/mouth/width
- mouth height: /gesture/mouth/height
- left eyebrow height: /gesture/eyebrow/left
- left eye openness: /gesture/eye/left
- right eye openness: /gesture/eye/right

Face OSC GUI maps the whole face into a mesh like structure which maintains the relative measurements between faces of different sizes by marking the distance between the nose tip and the upper lip. The absence of the mesh like structure is evident in case the face is not being detected.

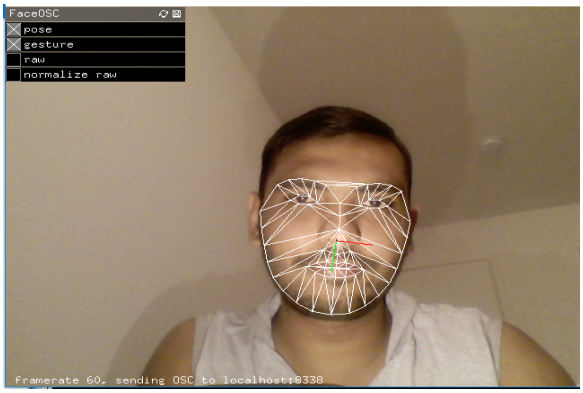

Fig. 1: Detection of Face using Face OSC

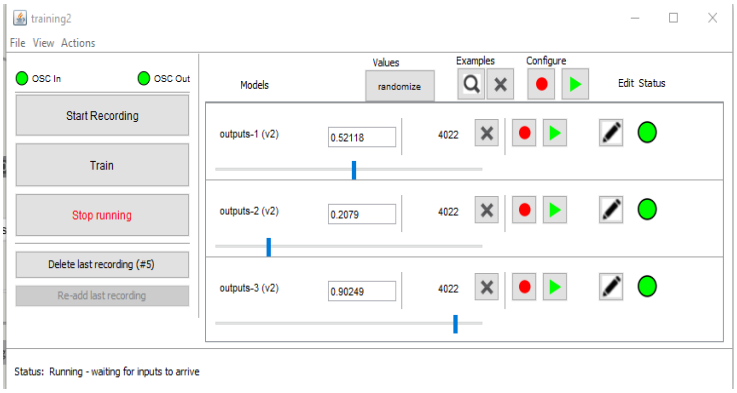

Fig. 2: Testing in Wekinator

B. Project Phase I

1) Feature Mapping:

- A menu to choose from sine, saw, triangle and rectangle oscillator
- A function object to control the envelope of the signal.
- kslider object to give the midi output based on the facial input.

For the initial mapping the conditional if statements are used to create separate ranges for each of the three expressions. The input from the expressions was given to a kslider object, in Max/MSP, which produces a MIDI output.

To generate a sonic output for each expression an additive synthesizer is developed. The first oscillator is designed to facilitate tone change in amplitude over time using function object. The frequency component is then controlled by the pitch obtained from the k-slider object. Furthermore, complex timbres, that are life-like, are produced using multiple oscillators. A menu is created to choose between different kind of oscillators including sine, saw, rectangle and triangle oscillators. This gives the tool a flexibility to choose different combinations of oscillators for different features depending on the type of sound that is desired. For example, for a facial expression like smiling one would prefer an aesthetically pleasing sound unlike screaming. This feature helps to map happy expressions to pleasant sounds or grave sounds for distress expressions.

Octaves for each oscillator are controlled by multiplying the pitch output frequency with a given factor. Since each pitch output from k-slider is multiplied with some factor, all rational values are chosen in order to maintain harmonicity i.e. choose integer multiples of the fundamental frequency. Using multiples factor values it is possible to explore the harmonic and inharmonic relationships between the oscillators. This feature gives the flexibility to explore more change in the pitch of the sound depending on the expressions. Moreover, if these factors are vey close to integer multiples which are whole numbers it helps to experience richer tone due to the phase relationships. Preset object aids to choose different factor, different order of oscillators and different envelope shape for each expression which in-turn enables to create a clear distinction between the expressions sonically. The setdomain object is used to control the duration or rather length of the note.

C. Project Phase II

1) Feature Mapping:

- Wekinator to train the algorithm to classify between three expressions depending on the facial input.
- A menu to choose from sine, saw, triangle and rectangle oscillator
- A function object to control the envelope of the signal.
- makenote object to give the midi output based on the facial input.

2) Wekinator: Wekinator is a tool that serves as a platform to enable the user to classify the real-time inputs, to a given system, using a specified set of machine learning algorithms. In the Wekinator GUI, five inputs are selected, which are the facial features obtained from the faceOSC. Based on these features combination of three outputs are generated. These outputs are used to classify four classes of expressions, which includes SCREAM, SMILE, SURPRISE and NEUTRAL STATE.

The classification of expressions is achieved through Neural network module of the Wekinator. To train the network, around 1000 samples for each expression are used.Following is an example of a typical neural network, designed for five inputs and four outputs. The output type is real valued and continuous limited between 0 and 1.

The makenote object is used instead of the kslider object to synthesize the MIDI nodes. Unlike in phase two, where conditional statements were utilized to provide the input to kslider, in this phase a trained machine learning model is used to provide the input to the makenote object. Each of the three parameters of the makenote object i.e. pitch, velocity and duration, are controlled by the three outputs from the trained model. The output of the makenote object is provided to the

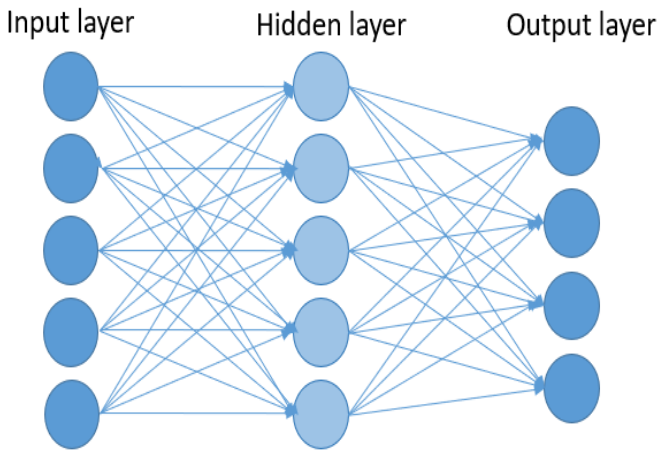

Fig. 3: Sample 5 input 4 output neural network with one hidden layer

same additive synthesizer part of the patch used in second phase. It was observed that for lower pitch values the change in amplitude as note goes across the oscillators was evident which was not the case for higher pitch values which gave very cohesive complex tones.

Providing the 3 outputs from Wekinator to SimpleFM object was also explored. Each parameter was chosen to control a parameter of SimpleFM object i.e. modulation index, harmonicity and carrier frequency. Depending on each parameter value the changes in the sound quality was studied. However, after a number of iterations, it was observed that using multiple oscillators which generates complex timbres is a better option for producing an aesthetically pleasing output.

IV. Applications

The system can be suited for a wide range of applications. A few of them are discussed below:

- Performances where facial expressions are widely used. (Kathakali and Mime)
- Information kiosks (depending on the mood of the customer it can learnt beforehand that a presence help operator is required rather than an automated system.)
- Meditation (to hear sounds depending on persons mood)
- To make the impact of VR games more powerful.
- Can be used as an alternative communication method for visually impaired people.

V. Conclusion and Future Work

The system developed in this project is currently able to synthesize distinctive set of notes using the features extracted from the set of predefined facial expressions by mapping them to the pitch, velocity and duration parameters of the makenote object in MAX/MSP. It also enables the user to enhance the amplitude envelope of the notes to make it aesthetically more pleasing. Owing to the high accuracy of the current model allows the possibility to include remaining facial features to enhance the resolution of the synthesized notes. Simultaneously, a codebook for different expressions can be created. This can further be used as a feedback to the system to enhance the accuracy of the model.This would also allow the system to recognize the facial expressions based on the music synthesized. Subjective feedback can be taken from a varied group of users to understand how easy or difficult it is utilize the system in performance settings. Depending on that personalized mappings can be created for different set of users i.e. forest ambience sounds for nature enthusiasts, different piano sounds for traditional piano lovers etc. The system can be tested in terms of comfortability, learning and recreation.

VI. Individual Contributions

Anik Jha - Extraction of Face Features and Make the notes distinctive based on Facial Expressions i.e. desired mapping.

Sameeksha Katoch - Training the system to Classify Facial Expressions. Building an Additive Synthesizer to generate complex tones from MIDI notes.

References

[1] A.J. Hornof, T. Rogers, and T. Halverson, EyeMusic: Performing live music and multimedia compositions with eye movements, NIME 2007: Conference on New Interfaces for Musical Expression. In proceedings on pp. 299-300.

[2] J. Kim, G. Schiemer, and T. Narushima, Coulog: Playing with Eye Movements, NIME 2007, New York City, USA.

[3] G. C. de Silva, T. Smyth, M. J. Lyons, A Novel Face-tracking Mouth Controller and its Application to Interacting with Bioacoustic Models, NIME 2004.

[4] M. J. Lyons, M. Haehnel, Designing, Playing, and Performing with a Vision-based Mouth Interface, NIME 03, Montreal, Canada.

[5] P. Ekman and W. Friesen. Facial Action Coding System: Investigators Guide. Consulting Psychologists Press, 1978.

[6] Y. Visell and J.R. Cooperstock, Modeling and Continuous Sonification of Affordances for GestureBased Interfaces, 13th Intl. Conf. on Auditory Display, Montreal, Canada, June 2007.

[7] Funk, Mathias, Kazuhiro Kuwabara, and Michael J. Lyons. "Sonification of facial actions for musical expression." Proceedings of the 2005 conference on New interfaces for musical expression. National University of Singapore, 2005.

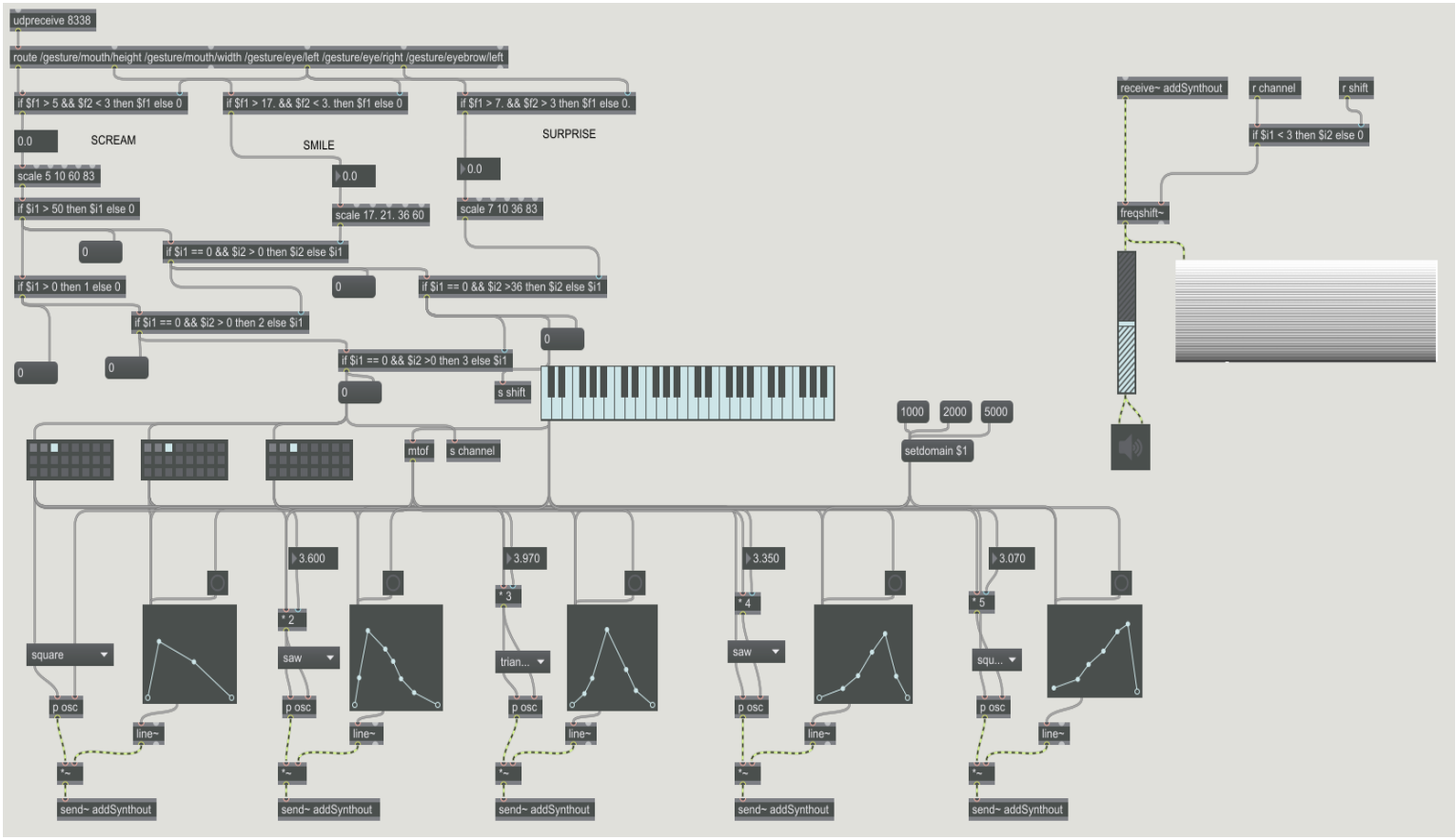

Fig. 4: Thresholding features using loops and controlling feature mapping using additive synthesizer

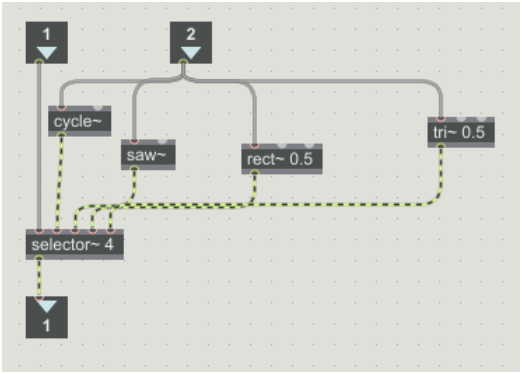

Fig. 5: Multiple oscillators encapsulated

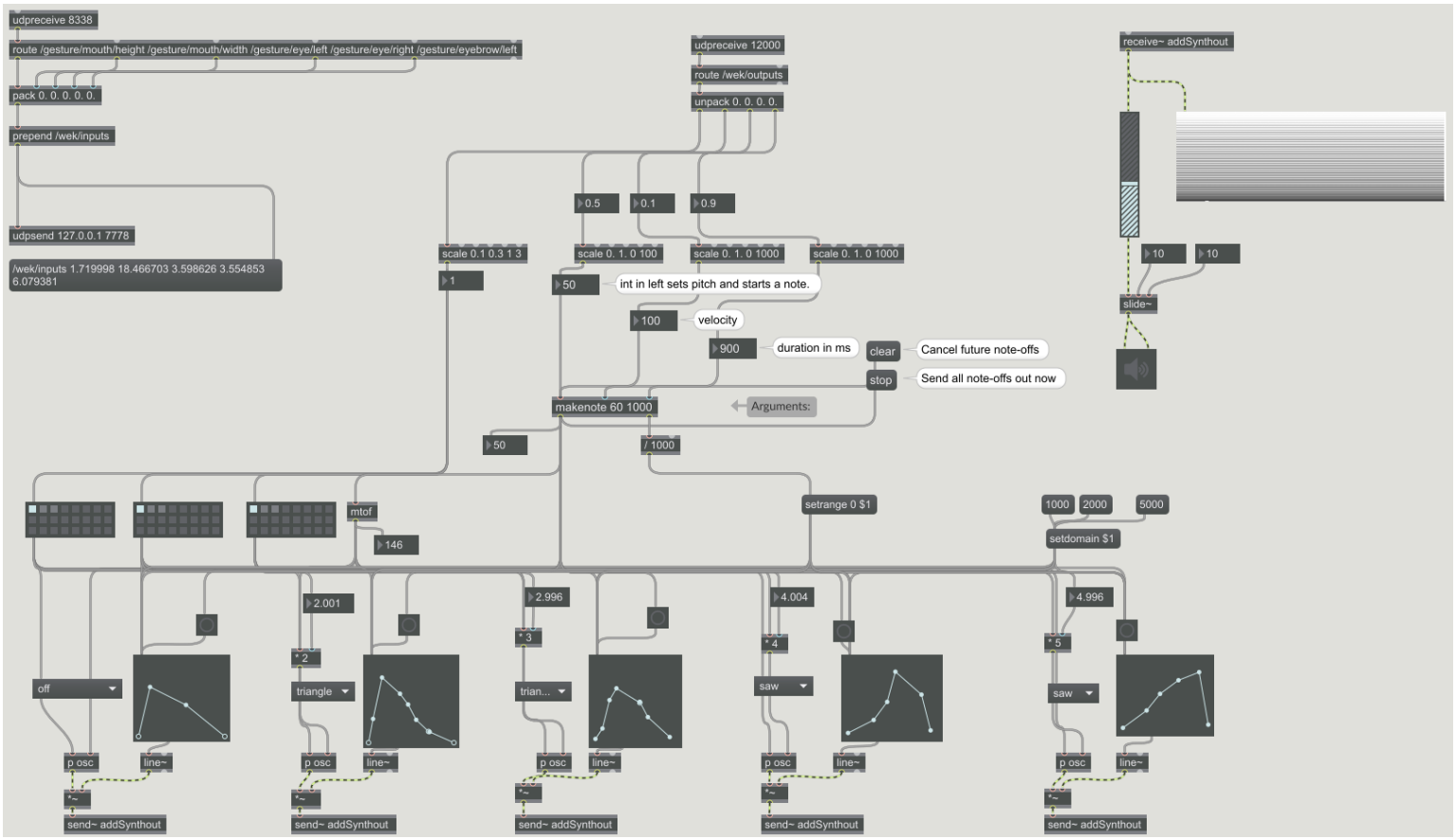

Fig. 6: Learning features using Wekinator controlling feature mapping using additive synthesizer

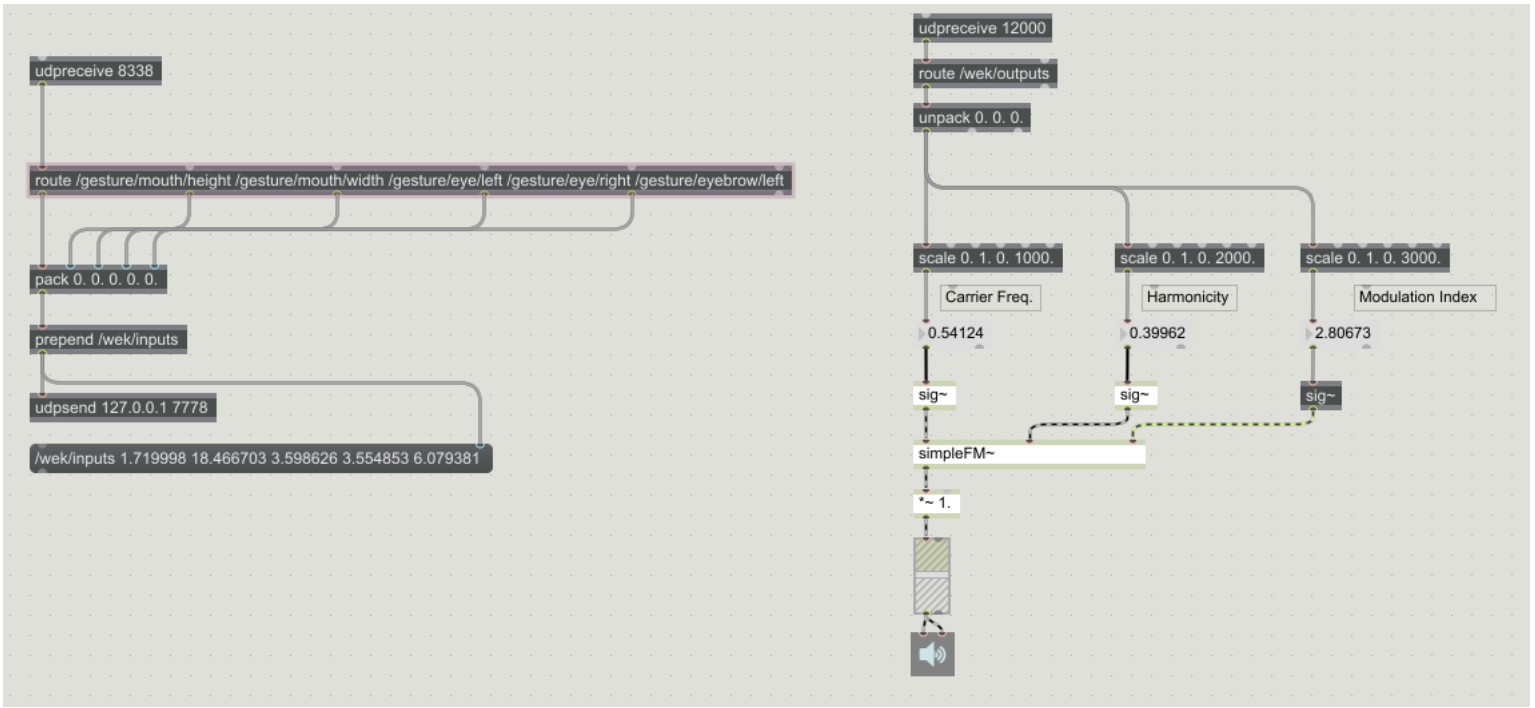

Fig. 7: Controlling feature mapping using simple FM