

**International Conference on New Interfaces for Musical Expression**

# **Algorithmic Power Ballads**

**Shelly Knotts<sup>1</sup>**

**<sup>1</sup>Durham University**

**Published on:** May 31, 2021

**License:** [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

## ABSTRACT

Algorithmic Power Ballads is a performance for Saxophone and autonomous improviser, with an optional third performer who can use the web interface to hand-write note sequences, and adjust synthesis parameters. The performance system explores shifting power dynamics between acoustic, algorithmic and autonomous performers through modifying the amount of control and agency they have over the sound over the duration of the performance. A higher-level algorithm how strongly the machine listening algorithms, which analyse the saxophone input, influence the rhythmic and melodic patterns generated by the system. The autonomous improviser is trained on power ballad melodies prior to the performance and in lieu of influence from the saxophonist and live coder strays towards melodic phrases from this musical style. The piece is written in javascript and WebAudio API and uses MMLL a browser-based machine listening library.

## Author Keywords

Machine Learning, Machine Listening, Improvisation

## CCS Concepts

- **Applied computing** → **Sound and music computing**; Performing arts;
- **Information systems** → *Music retrieval*;

## Introduction

Algorithmic Power Ballads (APB) is a performance for Saxophone and autonomous improviser, with an optional third performer who can use the web interface to hand-write note sequences, and adjust synthesis parameters. The performance system is written in Webaudio API and explores shifting power dynamics between acoustic, algorithmic and autonomous performers through modifying the amount of control and agency they have over the sound over the duration of the performance.

While the saxophonist improvises freely with the synthesis output, machine listening and learning algorithms are used to generate editable note sequences and audio that the other performer can interact with.

A higher-level algorithm controls how strongly each input influences the melodic patterns generated by the system. The autonomous improviser is trained on power

ballad melodies prior to the performance and in lieu of influence from the saxophonist and electronic music performer strays towards melodic phrases from this musical style.

## Power Ballads vs. Algorithmic Music

Power Ballads were chosen as the musical data for the autonomous improviser as a challenge to the convention that AI generated music is formulaic and stale in comparison to human music. The live coder (trained in academic and algorithmic music making) and the autonomous improviser (trained on Power Ballads) work with the same sound material during the performance in an algorithmic music battle to sway the music towards the immediate and emotional or the cerebral. The saxophone was chosen as the acoustic instrument in the trio as saxophone solos often feature at the peak of 1980s power ballads (Whitney Houston's cover of I Will Always Love You, for example), yet the instrument has a long history in improvisation and less formulaic music forms such as jazz and free improvisation.

In the performance the autonomous improviser tries to push the human performers towards making more 'emotional' music than tends to be present in algorithmic music scenes. Whilst the quantifiable characteristics of human produced music are easily reproducible (likely chord sequencers, melodic, and rhythmic patterns, etc) with sufficiently complicated algorithms, replicating emotional content is rather more challenging given theories of psychology still struggle to fully explain human emotion. [\[1\]](#)

Discussion around affective algorithms often work on the basis that human emotions can be separated into distinct parameters and quantified and therefore computed (see [\[2\]](#) for example). Often valence-arousal models are used for computing emotion, due to the ease of working in two-dimensions, though this model has been found by psychologists to over simplify human emotion [\[3\]](#). While algorithmic music making is able to produce performances indistinguishable from human performance [\[4\]](#), producing novel emotion in AI systems is still a challenge [\[5\]](#). In APB we stay clear of attempts to quantify human emotion and produce musical responses that could be categorised with psychological models of emotion and instead use human generated music with culturally proven emotional content to train an autonomous improviser to generate emotional musical content.

Metzler [\[6\]](#) describes Power Ballads as highly formulaic songs that combine sentimentality and uplift through a structure in constant escalation. Power Ballads are typically cross genre, with the 'host' genre using the ballad song form with emotional

lyrics as a way to access expressive aspects of music making within genres where this isn't a typical feature (such as rock). The emotions expressed in Power Ballads are typically vague meaning listeners can draw whatever they want from Power Ballads' 'large, indiscriminate and immediate' emotional content. This easy access to emotion partly explains the prominence of the Power Ballad in popular culture.

Algorithmic music is typically considered to be non-emotional, and in particular AI generated music has in recent year's sparked various existential crises within the music industry. With less and less direct human manipulation of parameters, artists and producers have questioned the artistry and legitimacy of such systems. To circumvent this crisis of emotionality in algorithmic music and to challenge the idea of human-generated music being inherently more emotional, in APB we strive to piggy-back the Power Ballad's formulaic emotional power, training our algorithm to produce "emotional" algorithmic music, in direct opposition to human musicians who typically play in more experimental genres.

During the performance the algorithm is exposed to experimental inputs through machine listening algorithms tracking audio features of the saxophone player and typed sequences from the electronic music performer. The three parties are in a battle to push the music towards one extreme of emotion or the other, without succumbing to the influence of each other. This is confounded by a higher level algorithm that modifies the influence of the saxophone (and machine listening algorithms), algorithmic performer and data-driven autonomous improviser over the course of the performance, perhaps letting particular characteristics dominate in the ebb and flow of the performance.

In a broader sense, the work also explores the quantification and appropriation of music. We performatively ask if turning power ballads into data and having the autonomous computer performer write its own versions of them really has the same cultural weight as a human produced power ballad. APB also asks what we lose in this process when the music is removed from its cultural context, and reduced to numbers to be interpreted by an algorithm. Finally, are the data-driven musical interpretations of emotion of the autonomous performer strong enough to illicit emotional responses from collaborators and audience members, and to sway the most cerebral of human performers towards immediacy and uplift?

## System Design

APB is written in the WebAudio API and utilises JavaScript libraries for musical machine listening and machine learning in the browser. The work is part of the AHRC funded MIMIC project, on machine learning and machine listening for creative projects.

The system design of APB consists of several main components which are unpacked in detail below: listeners, which track the parameters of the saxophone input in real time; an RNN, which periodically generates melodic sequences from the given input data; a synthesis engine which uses various parameters to play back the generated sequence; and a browser interface with various info, input and control panels.

### Listeners

APB is written in the WebAudio API and utilises JavaScript libraries for musical machine listening and machine learning in the browser. The work is part of the AHRC funded MIMIC project, on machine learning and machine listening for creative projects.

The system design of APB consists of several main components which are unpacked in detail below: listeners, which track the parameters of the saxophone input in real time; an RNN, which periodically generates melodic sequences from the given input data; a synthesis engine which uses various parameters to play back the generated sequence; and a browser interface with various info, input and control panels.

## ABSTRACT

Algorithmic Power Ballads is a performance for Saxophone and autonomous improviser, with an optional third performer who can use the web interface to hand-write note sequences, and adjust synthesis parameters. The performance system explores shifting power dynamics between acoustic, algorithmic and autonomous performers through modifying the amount of control and agency they have over the sound over the duration of the performance. A higher-level algorithm how strongly the machine listening algorithms, which analyse the saxophone input, influence the rhythmic and melodic patterns generated by the system. The autonomous improviser is trained on power ballad melodies prior to the performance and in lieu of influence from the saxophonist and live coder strays towards melodic phrases from this musical style. The piece is written in javascript and WebAudio API and uses MMLL a browser-based machine listening library.

## Author Keywords

Machine Learning, Machine Listening, Improvisation

## CCS Concepts

- **Applied computing** → **Sound and music computing**; Performing arts;
- **Information systems** → *Music retrieval*;

## Introduction

Algorithmic Power Ballads (APB) is a performance for Saxophone and autonomous improviser, with an optional third performer who can use the web interface to hand-write note sequences, and adjust synthesis parameters. The performance system is written in Webaudio API and explores shifting power dynamics between acoustic, algorithmic and autonomous performers through modifying the amount of control and agency they have over the sound over the duration of the performance.

While the saxophonist improvises freely with the synthesis output, machine listening and learning algorithms are used to generate editable note sequences and audio that the other performer can interact with.

A higher-level algorithm controls how strongly each input influences the melodic patterns generated by the system. The autonomous improviser is trained on power ballad melodies prior to the performance and in lieu of influence from the saxophonist and electronic music performer strays towards melodic phrases from this musical style.

## Power Ballads vs. Algorithmic Music

Power Ballads were chosen as the musical data for the autonomous improviser as a challenge to the convention that AI generated music is formulaic and stale in comparison to human music. The live coder (trained in academic and algorithmic music making) and the autonomous improviser (trained on Power Ballads) work with the same sound material during the performance in an algorithmic music battle to sway the music towards the immediate and emotional or the cerebral. The saxophone was chosen as the acoustic instrument in the trio as saxophone solos often feature at the peak of 1980s power ballads (Whitney Houston's cover of I Will Always Love You, for example), yet the instrument has a long history in improvisation and less formulaic music forms such as jazz and free improvisation.

In the performance the autonomous improviser tries to push the human performers towards making more 'emotional' music than tends to be present in algorithmic music

scenes. Whilst the quantifiable characteristics of human produced music are easily reproducible (likely chord sequencers, melodic, and rhythmic patterns, etc) with sufficiently complicated algorithms, replicating emotional content is rather more challenging given theories of psychology still struggle to fully explain human emotion. [1]

Discussion around affective algorithms often work on the basis that human emotions can be separated into distinct parameters and quantified and therefore computed (see [2] for example). Often valence-arousal models are used for computing emotion, due to the ease of working in two-dimensions, though this model has been found by psychologists to over simplify human emotion [3]. While algorithmic music making is able to produce performances indistinguishable from human performance [4], producing novel emotion in AI systems is still a challenge [5]. In APB we stay clear of attempts to quantify human emotion and produce musical responses that could be categorised with psychological models of emotion and instead use human generated music with culturally proven emotional content to train an autonomous improviser to generate emotional musical content.

Metzler [6] describes Power Ballads as highly formulaic songs that combine sentimentality and uplift through a structure in constant escalation. Power Ballads are typically cross genre, with the 'host' genre using the ballad song form with emotional lyrics as a way to access expressive aspects of music making within genres where this isn't a typical feature (such as rock). The emotions expressed in Power Ballads are typically vague meaning listeners can draw whatever they want from Power Ballads' 'large, indiscriminate and immediate' emotional content. This easy access to emotion partly explains the prominence of the Power Ballad in popular culture.

Algorithmic music is typically considered to be non-emotional, and in particular AI generated music has in recent year's sparked various existential crises within the music industry. With less and less direct human manipulation of parameters, artists and producers have questioned the artistry and legitimacy of such systems. To circumvent this crisis of emotionality in algorithmic music and to challenge the idea of human-generated music being inherently more emotional, in APB we strive to piggy-back the Power Ballad's formulaic emotional power, training our algorithm to produce "emotional" algorithmic music, in direct opposition to human musicians who typically play in more experimental genres.

During the performance the algorithm is exposed to experimental inputs through machine listening algorithms tracking audio features of the saxophone player and

typed sequences from the electronic music performer. The three parties are in a battle to push the music towards one extreme of emotion or the other, without succumbing to the influence of each other. This is confounded by a higher level algorithm that modifies the influence of the saxophone (and machine listening algorithms), algorithmic performer and data-driven autonomous improviser over the course of the performance, perhaps letting particular characteristics dominate in the ebb and flow of the performance.

In a broader sense, the work also explores the quantification and appropriation of music. We performatively ask if turning power ballads into data and having the autonomous computer performer write its own versions of them really has the same cultural weight as a human produced power ballad. APB also asks what we lose in this process when the music is removed from its cultural context, and reduced to numbers to be interpreted by an algorithm. Finally, are the data-driven musical interpretations of emotion of the autonomous performer strong enough to illicit emotional responses from collaborators and audience members, and to sway the most cerebral of human performers towards immediacy and uplift?

## System Design

APB is written in the WebAudio API and utilises JavaScript libraries for musical machine listening and machine learning in the browser. The work is part of the AHRC funded MIMIC project (<https://mimicproject.com/>), on machine learning and machine listening for creative projects.

The system design of APB consists of several main components which are unpacked in detail below: listeners, which track the parameters of the saxophone input in real time; an RNN, which periodically generates melodic sequences from the given input data; a synthesis engine which uses various parameters to play back the generated sequence; and a browser interface with various info, input and control panels.

## Listeners

The MMLL Library [7] provides higher level listeners for computer music including onset detectors, pitch detectors and various spectral trackers. All listening objects can run live or as feature extractors. In APB, both are used – a feature tracker is run on audio files of Power Ballads before the performance, and live trackers are used on the saxophone audio input.



During the performance, the saxophone's pitch and quantized inter-onset-interval values are tracked in real time, and chunks of Power Ballad are returned from the database that have a related pitch profile. These appear as editable text strings in the interface – one each for melodic and rhythmic values.

Spectral features (spectral centroid, spectral percentile, sensory dissonance, RMS and loudness) are recorded and averaged over 10 second blocks. The averaged spectral feature data is mapped to various parameters of the synth which plays back the RNN generated melodic sequences. The mapped parameters include LFO frequency, filter and amplitude modulation.

## **RNN**

At periodic intervals – either according to a timer, or a user activated evaluation -- the last 8 pitch and IOI values from both saxophone and power ballads are concatenated according to the current power balance values and fed into the google Magenta Music RNN model, which generates a new sequence of related material which is played in a loop by a synthesizer generated in Maximilian library with the current synthesis parameters.

The “power ballance” function curtails the amount of data from each input that is fed to the MusicRNN. For example if the Saxophone power value is 30% and the Power Ballad value is 70%, approximately 30% of the Saxophone's last 8 notes and 70% of the Power Ballad's last 8 notes will be concatenated together and used as the input to the RNN. The power balance is recalculated after each evaluation of the RNN, either choosing new random values or shifting by up to 5%.

## **Synthesis**

The audio is synthesised using the maximilian.js library which provides a subtractive synthesiser with mappable parameters, that can receive and play back melodic sequences.

The spectral feature data extracted from the saxophone is mapped to the synthesis parameters, but Maximilian also includes a GUI for direct manipulation of the synth, as such the synth is playable by both the saxophonist and third performer (if present).

## **Structure**

The evaluate function can be run manually by the performer, or is auto-evaluated by the system every 10-30 seconds. This feeds the data into the Music RNN, generates a

new melodic sequence, and sends new mapping parameters to the synth.

If the spectral feature data of the saxophone is “more noisy” in the previous 10 second block, than the average for the duration of the performance direct playback power ballads via the synth is triggered for 10-30 seconds with various looping parameters.

## Interface

The interface consists of a web page, currently hosted at <https://apb.beakfm.com/>. The web page setup enables significant accessibility to working with musicians, as technical proficiency is less of an obstacle, and facilitates working with collaborators who may not be in the same room. This is of benefit over traditional setups with e.g. supercollider, max patches where software install, and cross compatibility issues often slow down engagement, or make long distance collaboration less possible without significant technical support.

At present the interface requires the user to carry out a number of manual setup operations consisting of selecting the midi files of the particular power ballads to be used in the piece, and then initialising the web audio setup by enabling the microphone, and then starting the synthesiser. In future this process should be streamlined to allow a 1-step initialization of the work.

Some text boxes follow which show the most recently recorded MIDI pitch, IOI, and tempo values of the saxophone input, and a selected chunk of the power ballad with similar pitch profiles to the current saxophone input. On evaluation, the values shown in these boxes will be fed into the Music RNN.

The “power ballance” [sic] shows the proportion of the values in the text info boxes from each input which is fed into the Music RNN. This has two modes – one which moves the sliders up to 5% each time the Music RNN is run, and another which randomises the values. It is set to randomize as default.

On each evaluation a new Power Ballad is chosen as the input for the next round into the Music RNN. The ballad which was last fed into the Music RNN is shown in the “current ballad” box and the next to feed in is shown in the “next ballad” box.

The evaluate button triggers the data being fed into the Music RNN according to the given parameters, and to generate a new melodic sequence.

Maximilian generates a synthesizer GUI, which provides sliders for each of the synth parameters. These are set automatically on each evaluation but can also be modified

manually by the performer.

## Testing

The interface has been tested with two saxophonists, one local to the author – who was able to run the software while the saxophonist improvised, and another physical location to the author.

In the first instance, the testing proceeded in the usual way with the author controlling the setup and interface elements, where necessary, and debugging any arising technical problems.

In the second instance, the author shared a web link with the saxophonist. Although initial technical problems occurred due to hardcoded file paths which could be expected from a first test on a new machine, the saxophonist was then able, with some direction, to run the software without assistance. The technical issues could be rectified from afar and new files uploaded to the web page for by the author. This was then accessible to the Saxophonist on a hard reload of the web browser.

After a short discussion about how to use the interface, the saxophonist jammed with the interface before a second discussion when she gave some feedback. She then recorded a number of improvisations with the interfaces which were sent to the author for further evaluation.

On initial evaluation, compared to running live electronics works using locally installed softwares, the process of collaboration was extremely smooth. Web-based interfaces offer the advantage of avoiding compatibility and install issues, making custom algorithmic interfaces more accessible to performers without high level technical skills. This is also of great benefit in the present moment where being in the same room as collaborators is not always possible.

The saxophonist reported that she found the interface musically engaging and would like to play more with it in the future.

## Demo Video

A short demo of the piece which shows a saxophonist interacting with the interface can be viewed at the following link: <https://vimeo.com/507045005>

## Future Work

It is envisaged that the text boxes that currently allow a hand written input of pitch and rhythmic values into the system will evolve into a more fully functional system to allow algorithmic sequence generation and modification. This would allow the algorithmic performer to not only input values, but also to code the modification of sequences over time.

The ability to dynamically create presets during a performance will also be added, allowing the performer to 'save' particular sequences and associations, allowing more structural control.

It is also intended to make the interface more visually appealing, and to change the layout fit on one screen to avoid the need to scroll.

## Conclusions

The paper describes, Algorithmic Power Ballads, a performance system written in web audio which combines libraries for machine listening, machine learning and sound synthesis in a web browser-based interface for a saxophonist and optional algorithmic musician. The web browser offers significant advantages when working with musicians, particularly at a distance as compatibility and technical setup issues are negligible and can be fixed from a distance. The work explores the quantification of power ballads as part of an improvised performance, asking whether AI can produce 'emotional' music using formulaic styles, with the same weight as culturally situated music forms.

## Citations

1. Barrett, L. F. (2017). Categories and their role in the science of emotion. *Psychological Inquiry*, 28(1), 20-26. [↗](#)
2. Wallis, I., Ingalls, T., & Campana, E. (2008). Computer-Generating emotional music: The design of an affective music algorithm. *Proceedings - 11th International Conference on Digital Audio Effects, DAFx 2008*, 7-12. [↗](#)
3. Posner, J., Russell, J. A., & Peterson, B. S. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and

psychopathology. *Development and Psychopathology*, 17(3), 715–734.

<https://doi.org/10.1017/S0954579405050340>

4. Schubert, E., Canazza, S., De Poli, Giovanni, & Rodà, A. (2017). Algorithms can Mimic Human Piano Performance: The Deep Blues of Music. *Journal of New Music Research*, 46(2), 175–186. <https://doi.org/10.1080/09298215.2016.1264976>

5. Crowder, J. A., & Friess, S. (2012). Artificial psychology: The psychology of AI. *3rd International Multi-Conference on Complexity, Informatics and Cybernetics, IMCIC 2012 - Proceedings*, (November), 86–90.

6. Metzger, D. (2012). The power ballad. *Popular Music*, 31(3), 437–459.

<https://doi.org/10.1017/S0261143012000347>

7. Collins, N., & Knotts, S. (2019). A Javascript Musical Machine Listening Library. In *ICMC* (Vol. 19, p. 5).