Gesture-Timbre Space: Multidimensional Feature Mapping Using Machine Learning & Concatenative Synthesis

Michael Zbyszyński, Balandino Di Donato, and Atau Tanaka *

Embodied Audiovisual Interaction Group
Goldsmiths, University of London
New Cross, London, SE14 6NW, UK
m.zbyszynski@gold.ac.uk, b.didonato@gold.ac.uk, a.tanaka@gold.ac.uk

Abstract. This paper presents a method for mapping embodied gesture, acquired with electromyography and motion sensing, to a corpus of small sound units, organised by derived timbral features using concatenative synthesis. Gestures and sounds can be associated directly using individual units and static poses, or by using a sound tracing method that leverages our intuitive associations between sound and embodied movement. We propose a method for augmenting corporal density to enable expressive variation on the original gesture-timbre space.

1 Introduction

Corpus-based concatenative synthesis (CBCS) is a compelling means to create new sonic timbres based on navigating a timbral feature space. In its use of atomic source units that are analysed, CBCS is an extension of granular synthesis that harnesses the power of music information retrieval and the timbral descriptors it generates. The actual sound to be played is specified by a target and features associated with that target. In speech synthesis, the target is text. In audio resynthesis and "mosaicing" applications, the target can be another sound. In digital musical instrument (DMI) performance, the target may be sensor data or some representation of performer action, or gesture. The target may be of the same or different modality than the corpus, and it may have the same or different feature dimensionality.

CBCS performance systems until now have, on the whole, been implemented using dimensionality reduction. A subset of corporal features are projected into a low dimension space, typically Cartesian, and performance input is constrained to these dimensions. The dimensionality reduction acts as funnel that does not provide access to the complete feature space of the corpus and may forsake the richness of performance input.

* The research leading to these results has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant agreement No. 789825)

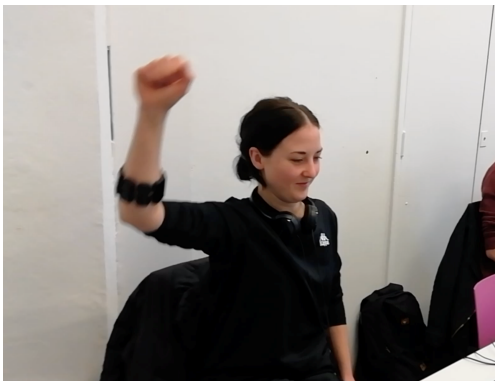

Fig. 1. A performer, wearing a sensor armband and engaging in a sound tracing study.

Our work builds upon a previous sound tracing study, where participants designed gestures to articulate time-varying sound using simple granular synthesis (Fig. 1). While sound tracing usually studies evoked gestural response [3], we extended traditional sound tracing to enable gesture-sound reproduction, and trained machine learning models to enable exploratory gestural performance by articulating expressive variations on the original sound. Participants were interested in exploring new timbres, but were limited by the provided corpus. The regression algorithm allowed them to scrub to different granular parameters in the stimulus sound, but not to a broader heterogeneous corpus. The study pointed out the potential for a more robust synthesis outlet for the gesture input regression model. Could we provide a corpus with sounds not in the original sound tracing stimulus? Could a regression model be harnessed to carry out feature mapping from the input domain (gesture) to the output domain (sound)?

We propose a system for exploring and performing with a multidimensional audio space using multimodal gesture sensing as the input. The input takes features extracted from electromyographic (EMG) and inertial sensors, and uses machine learning through regression modelling to create a contiguous gesture and motion space. EMG sensors on the forearm have demonstrated potential for expressive, multidimensional musical control, capturing small voltage variations associated with motions of the hand and fingers. The output target is generated via CBCS[11, 13], a technique which creates longer sounds by combining shorter sounds, called "units." A corpus of sounds is segmented into units which are catalogued by auditory features. Units can be recalled by query with a vector of those features. Our system allows musicians to quickly create an association between points and trajectories in a gesture feature space and units in a timbral feature space. The spaces can be explored and augmented together, interactively.

The paper is structured as follows. We first review related work on concatenative synthesis in performance. We then describe the proposed system, its archi-

tecture and technical implementation. Section 3 presents sound design strategies that address questions of corporal density to enable expressive performance, and the user workflow to associate gesture and CBCS sound via regression. In the discussion we provide a critical assessment of this approach and point out perspectives for future work before concluding.

2 Related Work

Aucouturier and Pachet [1] used concatenative sound synthesis to generate new musical pieces by recomposing segments of pre-existent pieces. They developed a constraint-satisfaction algorithm for controlling high-level properties like energy or continuity of the new track. They presented an example where a musician controls the system via MIDI, demonstrating an audio engine suitable for building real-time, interactive audio systems.

Stowell and Plumbley's [15] work focused on building associations between two differently distributed, unlabelled sets of timbre data. They succeeded in the implementation of a regression technique which learns the relations between a corpus of audio grains and input control data. In evaluating their system, they observed that such an approach provides a robust way of building trajectories between grains, and mapping these trajectories to input control parameters.

Schwarz et al. [14] used CataRT controlled through a 2D GUI in live performance taking five different approaches: (i) re-arranging the corpus in a different order than the original one, (ii) interaction with self-recorded live sound, (iii) composing by navigation of the corpus, (iv) cross-selection and interpolation between sound corpora, and (v) corpus-based orchestration by descriptor organisation. After performing in these five modes they concluded that CataRT empowers musicians to produce rich and complex sounds while maintaining precision in the gestural control of synthesis parameters. It presents itself as a blank canvas, without imposing upon the composer/performer any precise sonority.

In a later work [12], Schwarz et al. extended interaction modes and controllers (2D or 3D positional control, audio input) and concluded by stating the need for machine learning approaches in order to allow the user to explore a corpus by the use of XYZ-type input devices. They present gestural control of CataRT as an expressive and playful tool for improvised performance.

Savary et al. [9, 10] created *Dirty Tangible Interfaces*, a typology of user interfaces that favour the production of very rich and complex sounds using CataRT. Interfaces can be constantly evolved, irreversibly, by different performers at the same time. The interface is composed of a black box containing a camera and LED to illuminate a glass positioned above the camera, where users can position solid and liquid materials. Material topologies are detected by the camera, where a grey scale gradient is then converted into a depth map. This map is the projected onto a 3D reduction of the corpus space to trigger different grains.

Another example of gestural control of concatenative synthesis is the artistic project *Luna Park* by G. Beller [2]. He uses one accelerometer on top of each hand to estimate momentum variation, hit energy, and absolute position of the

hands. Two piezoelectric microphones responded to percussive patterns played in different zones of his body (one near the left hip and the other one near the right shoulder). Sensor data were then mapped to audio engine parameters to synthesise and interact with another performers recorded speech.

3 Methodology

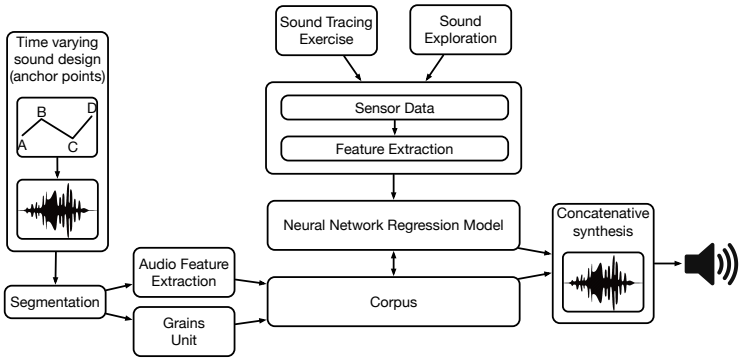

Fig. 2. System architecture diagram.

3.1 Implementation

We use a commercial sensor device[1], an armband worn on the forearm which packages eight electromyography (EMG) muscle sensors and an inertial measurement unit (IMU) for gross movement and orientation sensing, and transmits them over Bluetooth to a computer. We have also verified our approach with other biosensor packages, such as Plux's BITalino[2].

The software system is implemented in Cycling '74's Max[3]. We use the `myo`[4] object to capture raw EMG output of the sensors along with orientation quaternions from the on-board IMU to generate a multimodal feature vector representing the orientation, motion, and muscular state of the performer's forearm. Quaternions (x, y, z, and w: calculated by the device from accelerometer, gyroscope, and magnetometer data) give orientation, and we take the first-order difference between the current quaternion frame and the previous frame (x_d, y_d,

[1] `https://developerblog.myo.com/`
[2] `https://bitalino.com/`
[3] `https://cycling74.com/`
[4] `https://github.com/JulesFrancoise/myo-for-max`

z_d, w_d) to represent the current motion of the forearm. This is an important feature because hand gestures can be performed ballistically or in a more static fashion, causing different patterns of muscular activation even though the results are visually similar.

Raw EMG signals are intrinsically noisy, and we do not include them in our feature vector. Instead, we use a Bayesian[8] filter to probabilistically predict the amplitude envelope for each electrode in the armband. The sum of all eight amplitude envelopes is also included in the input feature vector, along with a new feature we have developed called "vector sum." Vector sum (Fig. 3) is a representation of the fact that the forearm muscles are situated around the arm in such a way that they can oppose or reinforce the action of other muscles. To calculate the vector sum, we model each electrode as representing a vector pointing away from the centre of a circle, evenly spaced every 45 degrees. The direction for each electrode vector does not change and the magnitude is proportional to the amplitude calculated by the Bayesian filter. The eight vectors are summed, and the resulting vector is related to the overall direction of force represented by all of the electrodes. When compared to the sum of all electrodes, the vector sum can distinguish gestures where muscles are opposing one another isometrically. This is an important feature, since joint movement might be minimal in such gestures but the subjective perception of effort is quite high. The vector sum is reported as a pair of Cartesian coordinates, which are better suited to regression than polar coordinates because they do not wrap around at zero degrees. See table 3.1 for a lists of the full gestural and timbral feature vectors. Where relevant, we took the average (μ) and standard deviation (σ) of each timbral feature over the whole audio unit.

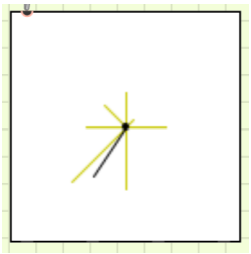

Fig. 3. An example vector sum, in black, drawn with the individual EMG vectors in yellow.

We implement machine learning in Max using an external object called `rapidmax`[7]. This object implements basic machine learning algorithms, such as multilayer perceptrons, k-nearest neighbour, and dynamic time warping, to allow Max users to quickly employ machine learning for regression or classification tasks. It is a Max wrapper around RapidLib [18], a C++ and JavaScript

library for creative, interactive machine learning applications in the style of Wekinator[4]. Specifically, we use a multilayer perceptron (MLP) neural network with one hidden layer to create models that perform regression based on user-provided training examples. This particular implementation uses a linear activation function on the output layer, allowing for model outputs that go beyond the numerical range of the provided examples, creating a larger and potentially more interesting generative space for aesthetic exploration. Training examples are created by associating inputs—gesture feature vectors—with outputs: vectors of timbral features. Performers can record example interactions, associating positions and gestures with sounds, to build an exploratory and performative gesture-timbre space.

Table 1. Input and output feature vectors for regression models

Input features(gesture)	Output features (timbre)
x	Duration
y	Frequency μ
z	Frequency σ
w	Energy μ
x_d	Energy σ
y_d	Periodicity μ
z_d	Periodicity σ
w_d	AC1 μ
EMG_1	AC1 σ
EMG_2	Loudness μ
EMG_3	Loudness σ
EMG_4	Centroid μ
EMG_5	Centroid σ
EMG_6	Spread μ
EMG_7	Spread σ
EMG_8	Skewness μ
EMG_{sum}	Skewness σ
$vectorSum_x$	Kurtosis μ
$vectorSum_y$	Kurtosis σ

The audio engine is implemented using MuBu[5] CataRT Max objects. We use the `mubu.process` object for segmentation and auditory feature analysis, `mubu.knn` for retrieval of the closest matching unit to a given set of auditory features, and the `mubu.concat` object for synthesising the unit once recalled in our workflow (see Section 3.3).

When a sound file is imported into MuBu, it is automatically segmented into units, either of a fixed length or determined by an onset detection algorithm (Fig. 4). A vector of auditory features (enumerated in table 3.1) is derived for each unit. These vectors of auditory features are associated with sensor feature

[5] `https://forumnet.ircam.fr/product/mubu-en/`

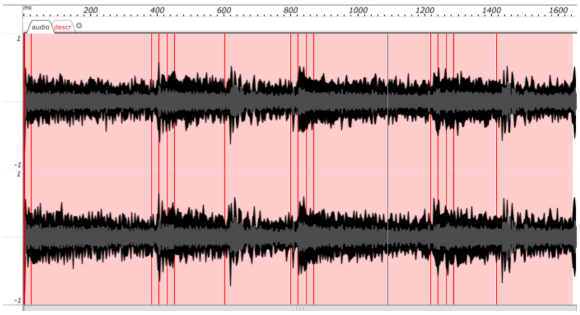

Fig. 4. A sound file imported into a MuBu buffer, using the onseg algorithm. Unit boundaries are shown as vertical, red lines. These lines would be equally spaced in chop mode. This display can be opened from the main gui window.

vectors to train a neural network, and roughly represent a high-dimensional timbral similarity space.

During playback, the amplitude and panning of the output is controlled by the "Amplitude Panner" (Fig. 5, upper right panel). The EMG sensors are divided into two groups and their amplitude envelopes are summed. The sum of each group is used to control the overall amplitude of the audio output in the left and right channels, respectively. When there is no muscular activation, both channels have near zero gain, giving the performer a natural method to make the instrument silent when they are not putting any energy into the system.

3.2 Sound & Gesture Design

In our previous study [16], we proposed four different approaches to designing gesture-timbre interaction based on a sound tracing exercise. In this system we revisit two of those approaches using our concatenative audio engine. Sound tracing is an exercise where a sound is given as a stimulus to study evoked gestural response [3]. Sound tracing has been used as a starting point for techniques of "mapping-by-demonstration" [5].

For that exercise, we used a general purpose software synthesizer, SCP by Manuel Poletti, controlled a breakpoint envelope-based playback system. We chose to design sounds that transition between four fixed anchor points with fixed synthesis parameters, primarily using SCP's granular synthesis engine. Envelopes interpolate between these fixed points. The temporal evolution of sound is captured as different states in the breakpoint editor whose envelopes run during playback. Any of the parameters can be assigned to breakpoint envelopes to be controlled during playback.

Users were asked to design a gesture that matched a pre-designed sound, and to train a regression model by associating data from that gesture with

the parameters of the sound. This created an exploratory space for performing variations on the source sound.

Our current work extends that activity with the use of CBCS. We go beyond the regression-based control of parametric synthesis from the previous study to create a mapping from gesture features to timbral features. This enables the user to perform the sound's corpus in real time, using variations on the original sound tracing gesture to articulate new sounds.

With this technique, we encounter potential problems of sparsity of the corpus feature space. There is no guarantee that there will be a unit that is closely related to the timbral features generated by the neural network in response to a given set of target gesture features. In order to address this, we added a step in our sound design method to fill the corpus with sound related to the original sound stimulus, but that had a wider range of timbral features. To do so, we went back to the original SCP sound authoring patch and replaced the source sample with a series of other sound samples. We then played the synthesis envelopes to generate time-varying sounds that followed the pitch/amplitude/parametric contour of the original stimulus. These timbral variants were recorded as separate audio files, imported into CataRT, and analysed. In this way, the corpus was enriched in a way that was directly related to the sound design of the original stimulus but had a greater diversity of timbral features, creating potential for more expressive variation in performance.

3.3 Workflow

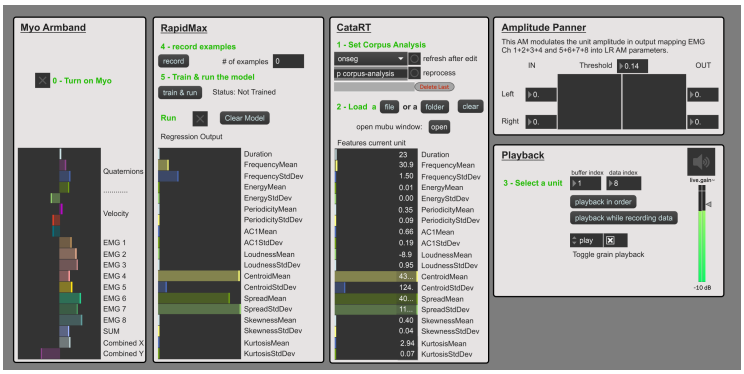

Fig. 5. Graphical-User Interface.

Fig. 5 shows the controls to interact with our system. At the beginning of a session, a performer activates the sensor armband using the toggle in the first column. The feature vector derived from the EMG/IMU sensors is displayed in

the same panel, allowing the performer to verify that the sensors are working as expected. They next select, in the CataRT column, the type of segmentation and analysis that will be performed on sound files imported into a corpus, choosing between onset-based segmentation and "chop," which divides the sound into equal-sized units. Parameters for these are available in a subpanel. The performer then imports individual sound files or folders of sounds into the corpus. Importing is cumulative; the whole corpus can be cleared with the clear button. It is also possible to re-analyse the current corpus using new parameters.

Once the sensors are activated and a corpus has been imported, segmented, and analysed, the gesture-timbre mapping process can begin. Performers may listen to individual units by selecting the buffer index (which file the unit is from) and data index (which unit in a file). The analysed timbral features associated with the selected unit are displayed by the multislider in the same column. In order to associate gestural features with the selected timbral features, they press the *record* button in the RapidMax column, which automatically captures 500ms of sensor data associated with the selected unit data.

Another way of interacting with audio units is to play an entire file from the first to the last unit. This enables the sound tracing workflow. Performers can listen to the whole sound and design the appropriate gesture to accompany that sound. Once that gesture has been designed, they can click on the *playback while recording data* button and then perform their gesture synchronously with sound playback. As each unit passes in order, the associated timbral data will be recorded in conjunction with the gestural data at that moment. This corresponds to the "whole gesture regression" mode from our previous work.

Once a set of potentially interesting example data has been recorded, users train the neural network by clicking the *train & run* button. When training has finished, the system enters run mode. At this point, incoming gestural data is sent to the trained regression model. This model outputs a vector of target timbral features that is sent to a k-nearest neighbours algorithm implemented in MuBu. That model outputs the unit buffer and data indices that most closely match the targeted timbral features. `mubu.concat` receives the indices and plays the requested unit. In this way, we have coupled the MLP model of gestural and timbral features to a k-NN search of timbrally defined units the corpus. This process allows musicians to explore the gesture-timbre space, and perform with it in real time.

4 Discussion

In lieu of a formal evaluation, in this section we reflect on the strengths and weaknesses of the system with a critical assessment of its affordances based on testing by the authors.

Concatenative synthesis is described in terms of a target that one tries to synthesise by navigating a corpus. In an audio-audio mosaicing task, the "target" is an example sound that one is trying to resynthesize with the corpus. In cases using interfaces for live controllers [9, 12], the mapping between gesture

and sound has, until now, taken place in a reduced dimension space. Two or three features are selected as pertinent and projected onto a graphical Cartesian representation. Our system does not require dimensionality reduction, and the number of input dimensions does not need to match the number output dimensions. This creates the disadvantage, however, of not being able to visualise the feature space. Schwarz in [12] finds seeing a reduced projection of the feature space convenient, but prefers to perform without it.

Schwarz describes exploratory performance as a DMI application of CBCS that distinguishes it from the more deterministic applications of speech synthesis or audio mosaicing [12]. He provides the example of improvised music where the performer uses an input device to explore a corpus, sometimes one that is being filled during the performance by live sampling another instrumentalist. This creates an element of surprise for the performer. Here we sought to create a system that enables timbral exploration, but that would be reproducible, and useful in compositional contexts where both sound and associated gesture can be designed *a priori*. In using our system, we were able to perform the sound tracing gesture to reconstruct the original sound. This shows that the generation of time-varying sound sources from our parametric synthesis programme were faithfully reproduced by CataRT in this playing mode.

Difficulties arose when we created gesture variations where the regression model started to "look for" units in the feature space that simply were not there, raising the problem of corpus sparseness. The sound design strategy to generate variants effectively filled the feature space. It was important that the feature space was filled not just with any sound, but with sound relevant to the original for which the gesture had been authored. By generating variants using different sound samples but that followed the same broad sonic morphology, we populate the corpus with units that were musically connected to the original but that were timbrally (and in terms of features) distinct. This creates a kind of hybrid between a homogeneous and heterogeneous corpus. It is heterogeneous in the diversity of sound at the performer's disposal, but remains musically coherent and homogeneous with the original sound/gesture design. This allowed expressive variation on a composed sound tracing gesture.

In order to support expressive performance we need to create gesture-timbre spaces that maximise sonic diversity. When a performer navigates through gesture space, the outcome is more rich and expressive if a diverse range of individual units are activated. The nuances of gesture become sonically meaningful if that gestural trajectory has a fine-grained sonic result. This is not always the result of the proposed workflow, especially in the case where the user chooses individual units and associates them with specific gestural input. It is, again, a problem of sparseness; the distribution of units in a high-dimensional space is not usually even. In a typical corpus, there will be areas with large clusters of units and other units that are relatively isolated. When musicians choose individual units to use for gesture mapping, there is a tendency to choose the units that have the most character. These units are often outliers. In a good outcome, outlier units represent the edges of the timbral space of the corpus. In this case, a regression

between units on different edges of the space will activate a wide range of intermediate units. However, it is also possible that a gestural path between two interesting units does not pass near any other units in the corpus. In that case, the resulting space performs more like a classifier, allowing the performer to play one unit or another without any transitional material between them. It would be helpful to give users an idea about where the potentially interesting parts of a corpus lie. It might also be possible to automatically present performers with units that represent extreme points of the timbral space, or areas where gesture mapping might yield interesting results.

Future work to address varying sparseness and density of the corpus feature space maybe be in dynamic focus on areas more likely to have sound. Schwarz in [12] uses Delaunay triangulation to evenly redistribute the three dimensional projection of the corpus in his tablet based performance interface. This operation would be more difficult in a higher dimensional space. One recent development in the CataRT community has been the exploration of using self-organising maps to create a more even distribution of features in the data space [6].

Another potentially interesting way to use multidimensional gesture-timbre mapping is to generate feature mapping using one corpus of sounds, and then either augment that corpus or change to an entirely different corpus – moving the timbral trajectory of a gesture space into a new set of sounds. This can be fruitful when units in the new corpus intersect with the existing gesture space, but it is difficult to give users an idea about whether or not that will be the case. One idea we are exploring is "transposing" a trajectory in timbral feature space so that it intersects with the highest number of units in a new corpus. This could be accomplished using machine learning techniques, such as dynamic time warping, to calculate the "cost" of different ways to match a specific trajectory to a given set of units, and find the optimal transposition. We know from Wessel's seminal work on timbre spaces [17] that transposition in a low dimensional timbre space is perceptually relevant. Automatically generating these transpositions, or suggesting multiple possible transpositions, has the potential to generate novel musical phrases that are perceptually connected to the original training inputs.

5 Conclusions

We have presented a system that combines regression-based machine learning with corpus-based concatenative synthesis. We extend a previous study where a sound tracing workflow was used to design gestures to articulate time varying sounds. Gesture input from EMG and IMU sensors generated multidimensional targets and are associated with specific points in a high-dimensional timbral feature space in order to train a neural network. Using this workflow, we were able to reproduce original sound tracings. By populating the corpus with related, but timbrally diverse grains, we increased the corporal density to enable expressive variation on the original gesture. This workflow demonstrates the use of real-time, interactive machine learning with CataRT and creates a multidimensional feature mapping linking gesture to sound synthesis.

References

1. J.-J. Aucouturier and F. Pachet, *Jamming with Plunderphonics: Interactive concatenative synthesis of music*, Journal of New Music Research, 35 (2006), pp. 35–50.
2. G. Beller, *Gestural control of real time speech synthesis in lunapark*, in Proceedings of Sound Music Computing Conference, SMC, Padova, Italy, 2011.
3. B. Caramiaux, F. Bevilacqua, and N. Schnell, *Towards a gesture-sound cross-modal analysis*, in Gesture in Embodied Communication and Human-Computer Interaction, Berlin, Heidelberg, 2010, pp. 158–170.
4. R. Fiebrink and P. R. Cook, *The wekinator: a system for real-time, interactive machine learning in music*, in Proceedings of The International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, 2010.
5. J. Françoise, *Motion-sound mapping by demonstration*, PhD thesis, UPMC, 2015.
6. J. Margraf, *Masters thesis*, TU Berlin, (2019).
7. S. T. Parke-Wolfe, H. Scurto, and R. Fiebrink, *Sound control: Supporting custom musical interface design for children with disabilities*, in Proceedings of the International Conference on New Interfaces for Musical Expression, NIME'19, Porto Alegre, Brazil, 2019.
8. T. D. Sanger, *Bayesian filtering of myoelectric signals*, Journal of neurophysiology, 97 (2007), pp. 1839–1845.
9. M. Savary, D. Schwarz, and D. Pellerin, *Dirti —dirty tangible interfaces*, in Proceedings of the International Conference on New Interfaces for Musical Expression, NIME'12, Ann Arbor, Michigan, 2012.
10. M. Savary, D. Schwarz, D. Pellerin, F. Massin, C. Jacquemin, and R. Cahen, *Dirty tangible interfaces: Expressive control of computers with true grit*, in CHI '13 Extended Abstracts on Human Factors in Computing Systems, CHI EA '13, Paris, France, 2013, ACM, pp. 2991–2994.
11. D. Schwarz, *Concatenative sound synthesis: The early years*, Journal of New Music Research, 35 (2006), pp. 3–22.
12. D. Schwarz, *The sound space as musical instrument: Playing corpus-based concatenative synthesis*, in Proceedings of the International Conference on New Interfaces for Musical Expression, NIME'12, Ann Arbor, Michigan, 2012.
13. D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, *Real-Time Corpus-Based Concatenative Synthesis with CataRT*, in 9th International Conference on Digital Audio Effects, DAFx 19, Montreal, Canada, 2006, pp. 279–282.
14. D. Schwarz, R. Cahen, and S. Britton, *Principles and applications of interactive corpus-based concatenative synthesis*, in Journes d'Informatique Musicale, JIM, Albi, France, 2008.
15. D. Stowell and M. D. Pumbley, *Timbre remapping through a regression-tree trchnique*, in Proceedings of the Sound Music Computing Confetence, SMC, 2010.
16. A. Tanaka, B. Di Donato, and M. Zbyszyński, *Designing gestures for continuous sonic interaction*, in Proceedings of the International Conference on New Interfaces for Musical Expression, NIME'19, Porto Alegre, Brazil, 2019.
17. D. L. Wessel, *Timbre space as a musical control structure*, Computer music journal, (1979), pp. 45–52.
18. M. Zbyszyński, M. Grierson, and M. Yee-King, *Rapid prototyping of new instruments with codecircle*, in Proceedings of the International Conference on New Interfaces for Musical Expression, Copenhagen, Denmark, 2017, pp. 227–230.