INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Finding intelligible consonant-vowel sounds using high-quality articulatory synthesis

Daniel R. van Niekerk[1], *Anqi Xu*[1], *Branislav Gerazov*[2], *Paul K. Krug*[3], *Peter Birkholz*[3], *Yi Xu*[1]

[1]Department of Speech, Hearing and Phonetic Sciences, University College London, UK
[2]Faculty of Electrical Engineering and Information Technologies, UCMS, Skopje, RN Macedonia
[3]Institute of Acoustics and Speech Communication, Technische Universität Dresden, Germany
d.vniekerk@ucl.ac.uk

Abstract

In this study, a state-of-the-art articulatory speech synthesiser was used as the basis for simulating the exploration of CV sounds imitating speech stimuli. By adopting a relevant kinematic model and systematically reducing the search space of consonant articulatory targets, intelligible CV sounds can be found. Derivative-free optimisation strategies were evaluated to speed up the process of exploring articulatory space and the possibility of using automatic speech recognition as a means of evaluating intelligibility was explored.

Index Terms: computational phonetics, articulatory speech synthesis, early vocal learning.

1. Introduction

The task of obtaining articulatory trajectories from speech exemplars has been attempted with different motivations including (i) to model and understand the process of learning articulation to produce speech and (ii) to reproduce utterances using an articulatory synthesiser, i.e. copy synthesis. The former could serve as a basis for studies in speech and phonetic sciences [1] and early vocal learning [2]. The latter could have a direct impact on the development of speech technologies, especially where it is not possible or economical to use current state-of-the-art methods. For example, to bootstrap rapid development of real-world text-to-speech (TTS) synthesisers from limited or compromised data [3] or for data augmentation when building automatic speech recognition (ASR) systems [4, 5].

The recent development of the state-of-the-art articulatory synthesiser *VocalTractLab* [6, 7] has provided a compelling tool to investigate the discovery of articulatory targets necessary to produce intelligible and natural sounding speech. While previous studies have considered the task of modelling articulatory movements [8, 9, 10], the emphasis has been on modelling the articulatory to acoustic and inverse mappings. Consequently, the scope of these works has often been limited to analysis of articulatory or formant trajectories of vowels or artificial segment sequences, with limited evaluation of the generated speech.

In this work the focus is on imitative articulatory exploration as an important component of early vocal learning [8, 11]. The approach in [12] is followed in adopting a kinematic model for generating articulatory trajectories [13] and evaluating the outcomes in terms of intelligibility. The current paper extends that work by:

1. Evaluating derivative-free optimisation for speeding up the process of CV discovery compared to uniform sampling.
2. Comparing the articulatory parameter-tying configuration used in [12], motivated by the *syllable-synchronised sequential target approximation* hypothesis which asserts that articulatory dimensions not involved in implementing the consonant start moving towards the vowel target at the start of the syllable [14], with an unconstrained configuration.
3. Exploring the use of automatic speech recognition (ASR) as a means of estimating intelligibility.

In the following section a brief description of related work is presented and contrasted with the current work. Section 3 describes the experimental setup with results presented in Section 4. Finally, Section 5 contains a discussion with conclusions and proposals for future work.

2. Approach

Finding and refining articulatory movements to produce speech sounds imitatively is assumed to be an important stage of early vocal learning [8, 11] and is sometimes viewed as learning goal-directed sensori-motor control [8, 15]. Previous studies have often focused on the type of model and algorithms for learning the articulatory to acoustic and inverse mappings, including distal supervised learning [8, 9], reinforcement learning [16] and others [10] and sometimes assume that articulatory trajectories are the outputs of the learned model [9, 10].

Copy synthesis efforts typically focus on evaluating the generated speech against the target utterance [17]. The work by Gao et al. [17] adopts a kinematic model for articulator movements [13] and pragmatic constraints of parameters to reduce the computational demands of the optimisation process.

This paper considers the task of imitative articulatory exploration as a component of early vocal learning, differing in the following ways from the above studies: (i) intelligibility is the primary measure of success (ii) a simple kinematic model is adopted to produce articulatory trajectories [13] as in [17], and (iii) a phonetically motivated model of coarticulation is used; syllable-synchronised sequential target approximation is evaluated (point 2, Section 1).

3. Experimental setup

Two experiments were set up to simulate articulatory exploration with the goal of imitating CV onsets from a set of pre-recorded *acoustic templates* produced as complete words by a British male speaker (Table 1). In the first experiment, the input space was reduced using parameter tying described in Section 3.3. The second experiment compared different optimisation algorithms (Section 3.4) against uniform sampling of the input space. Results are presented in terms of recognition rates obtained from an online listening task and an ASR system de-

http://dx.doi.org/10.21437/Interspeech.2020-2545

Table 1: *Acoustic templates (CVC words).*

Vowel	/bV/	/dV/	/gV/
/iː/	bead	deed	
/ɪ/	bid	did	
/ɛ/	bed	dead	
/æ/	bad	dad	
/ɒ/	bod		god
/uː/	booed		good
/ʌ/	bud		

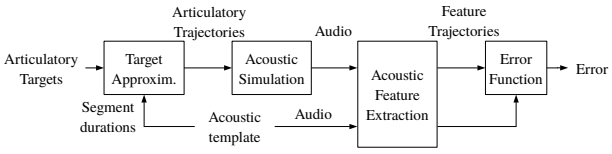

Figure 1: *Objective function implementation.*

scribed in sections 3.5 and 3.6. The implementation of the simulation is described in the following two sections.

3.1. Synthesis and objective function

During articulatory exploration samples are synthesised using the *VocalTractLab*[1] articulatory synthesiser. A single iteration of the simulation is shown in Figure 1. The *articulatory targets* are determined at each iteration by the optimisation algorithm and consist of one complete set of 26 parameters for each of the segments C and V. Given these two targets and the segment durations from the acoustic template, 24 *articulatory trajectories* are synthesised which drive the acoustic simulation by VocalTractLab. The vocal tract and glottis parameters as well as the two additional time-constants which determine the rate of target approximation are given with their ranges in Table 2 (these ranges are relevant to the *adult male speaker*, referred to as "JD2", with triangular glottis model defined in VocalTractLab). As can be seen, the tongue side elevation, lip minimum area and glottis parameters are kept constant during our simulations and all simulation runs are initialised with the neutral parameters which produce a schwa. As a result, the number of free parameters per segment is 15. The acoustic *feature trajectories* are 12-dimensional static Mel-frequency cepstral coefficients (including energy) extracted every 5 ms in a 10 ms Hamming window using *librosa*[2] [18]. The error function used is the mean squared error (MSE) assuming one-to-one frame alignment since the length corresponds to the durations in the template for both trajectories (this means that the scoring of the C segment is sensitive to the vocal tract time-constant τ_{vt} because of the effect on temporal alignment).

3.2. Articulatory exploration

The task of finding articulatory targets that best reproduce the acoustic template is defined as

$$\mathbf{x_o} = \underset{\mathbf{x}\in\mathbf{X}}{\operatorname{argmin}}\, f(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is a concatenation-vector of articulatory targets representing the C and V segments[3], f is the objective function

[1] `http://www.vocaltractlab.de/` (v2.3-beta)

[2] `https://github.com/librosa/librosa` (v0.7.2)

[3] This is 30-dimensional given the 15 free parameters from Table 2

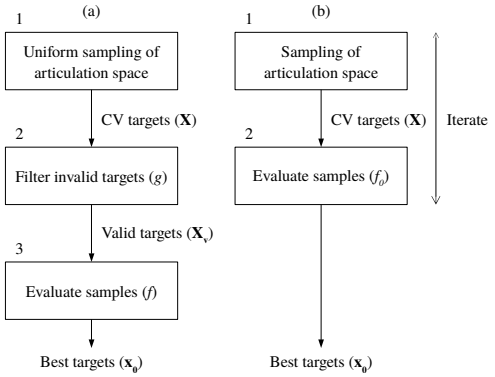

Figure 2: *Articulatory exploration: (a) Uniform sampling, and (b) sampling determined by optimisation algorithm.*

Table 2: *Target parameters with ranges used.*

Articulatory dimensions (d)		**Neutral**	**Range**	
Hyoid position (horz.)	HX	1.00	$[0.0, 1.0]$	cm
Hyoid position (vert.)	HY	-4.75	$[-6.0, -3.0]$	cm
Jaw position (horz.)	JX	0.00	$[-0.5, 0.0]$	cm
Jaw angle	JA	-2.00	$[-7.0, 0.0]$	deg.
Lip protrusion	LP	-0.07	$[-1.0, 1.0]$	cm
Lip distance	LD	0.95	$[-2.0, 4.0]$	cm
Velum shape	VS	0.00	$[0.0, 1.0]$	
Velic opening	VO	-0.10	$[-0.1, 1.0]$	cm^2
Tongue body (horz.)	TCX	-0.40	$[-3.0, 4.0]$	cm
Tongue body (vert.)	TCY	-1.46	$[-3.0, 1.0]$	cm
Tongue tip (horz.)	TTX	3.50	$[1.5, 5.5]$	cm
Tongue tip (vert.)	TTY	-1.00	$[-3.0, 2.5]$	cm
Tongue blade (horz.)	TBX	2.00	$[-3.0, 4.0]$	cm
Tongue blade (vert.)	TBY	0.50	$[-3.0, 5.0]$	cm
Tongue side elevation 1	$TS1$	0.00	0.00	cm
Tongue side elevation 2	$TS2$	0.00	0.00	cm
Tongue side elevation 3	$TS3$	0.00	0.00	cm
Lip minimum area	LMA	-0.05	-0.05	cm^2
Fundamental frequency	$F0_{gl}$	120.00	120.00	Hz
Sub-glottal pressure	SP_{gl}	8000.00	8000.00	dPa
Lower rest displacement	LD_{gl}	0.01	0.01	cm
Upper rest displacement	UD_{gl}	0.01	0.01	cm
Arytenoid area	AA_{gl}	0.00	0.00	cm^2
Aspiration strength	AS_{gl}	-40.00	-40.00	dB
Vocal tract time-constant	τ_{vt}	0.015	$[0.005, 0.039]$	s^{-1}
Glottis time-constant	τ_{gl}	0.015	0.015	s^{-1}

described in 3.1 and $\mathbf{X}$ is the set of vectors evaluated during the exploration process. Two distinct processes are compared (implementations shown in Figure 2):

1. Uniform sampling of the input space (as in [12]).
2. Derivative-free or zeroth-order optimisation algorithms.

In both cases we apply a simple validity test, $g : \mathbf{X} \to \{0, 1\}$, which asserts whether the V target (if achieved) would result in a relatively open vocal tract.[4] This avoids the computationally expensive process of synthesising and comparing the complete sample where possible. For uniform sampling, the set of samples does not depend on the output of previous iterations and invalid targets are simply filtered out in advance. For the optimisation algorithms $\mathbf{X}$ depends on the objective function and the specific algorithm; in this case the validity test was incorpo-

[4] This is implemented by thresholding the magnitude of the volume velocity transfer function.

Table 3: *Target parameters used for tied onsets (definitions in Table 2).*

Consonant	Control parameters
/b/	$\{JX, JA, LD, \tau_{vt}\}$
/d/	$\{JX, JA, TTX, TTY, TBX, TBY, \tau_{vt}\}$
/g/	$\{JX, JA, TCY, \tau_{vt}\}$

rated into the objective function as follows:

$$f_0(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{where } g(\mathbf{x}) = 1 \\ c \gg \max\limits_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}), & \text{otherwise} \end{cases}$$

3.3. Parameter tying

The effect of systematically reducing the search space for the consonant targets was investigated. Firstly, as done in [12], a specific subset of the articulatory parameters (the consonant *control parameters* listed in Table 3) was selected for optimisation, depending on the consonant identity (this constitutes a form of prior knowledge). The remaining undefined parameter values of the consonant target are copied from the vowel target, i.e. the parameters are tied to the vowel. This is a direct implementation of the assumption of *syllable-synchronised sequential target approximation*. Secondly, the jaw angle (JA) parameter was further limited (to the range $[-7.0, -2.0]$) to prevent closing the mouth for /d/ and /g/. The number of free parameters for the *tied-onset* configuration is thus 19 or 22 for the bilabial and velar or alveolar CVs respectively compared to 30 without parameter tying (*free-onset*).

3.4. Optimisation algorithms

The following two optimisation algorithms were investigated: (i) A model-based algorithm (referred to as *forest* in Section 4) which uses a regression model of the objective function to select the next point for evaluation [19]. The *scikit-optimize* package[5] was used to construct an extra-trees regressor [20] initialised with the neutral parameters and a uniform sampling of 10% of the maximum number of iterations. The selection of the next sample (acquisition function) was determined by the minimisation of the lower confidence bound (LCB) of the model. (ii) The controlled random search (*CRS*) with local mutation algorithm [21] which starts with a random "population" of points and evolve them using an algorithm similar to the Nelder-Mead method [22] as implemented in the *NLopt* package[6] [23].

3.5. Listening test

The CV targets obtained from exploration were evaluated using a free-recognition listening test. Firstly, the set of template words (Table 1) were synthesised using VocalTractLab to append a /d/ coda. The randomised word samples were presented to 9 listeners in an online experiment.[7] Listeners were expected to type in the word played back through headphones or indicate if the sample was unintelligible after listening to it no more than 3 times. For the listening test, only samples from uniform sampling and CRS were included.

[5] `https://github.com/scikit-optimize` (v0.7.4)

[6] `https://github.com/stevengj/nlopt` (v2.6.2)

[7] Run on *Gorilla* during May 2020 (`https://gorilla.sc/`)

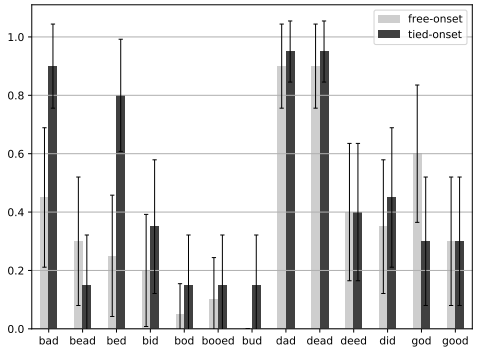

Figure 3: *Mean recognition rates using ASR with 95% conf. intervals.*

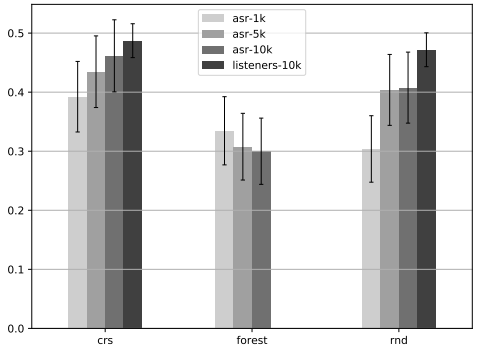

Figure 4: *Mean recognition rates (ASR and listeners) with 95% conf. intervals and exploration iteration limits: 1k, 5k and 10k.*

3.6. ASR-based evaluation

As a proxy for online listening tests, off-the-shelf ASR was investigated to estimate the intelligibility of the CV targets embedded in words as used in the listening test. For this purpose the *Google Speech-to-Text* service[8] was used. The latest version (*v1p1beta1*) of the synchronous recognition endpoint was called using the *Python* client[9] with default settings, i.e. the service automatically determines the appropriate back-end model to use based on the input. In addition to the 44.1 kHz audio samples, the set of words in Table 1 was submitted as "speech contexts". This adjusts the language model component in favour of this set of words and is considered best-practice for recognising short utterances.[10] A single request for the 10-best list was made to the service for each sample which was padded to ensure a minimum duration of 1.5s. However, the results in Section 4 only considered the 1-best output.

4. Results

4.1. Parameter tying

The CRS algorithm was used to compare CV outcomes with the reduced and full input space described in Section 3.3 over 20 exploration runs limited to 10k iterations. The mean recognition rates by the ASR system are presented in Figure 3, showing

[8] `https://cloud.google.com/speech-to-text` accessed during April and May 2020.

[9] `https://pypi.org/project/google-cloud-speech/` (v1.3.2)

[10] `https://cloud.google.com/speech-to-text/docs/best-practices`

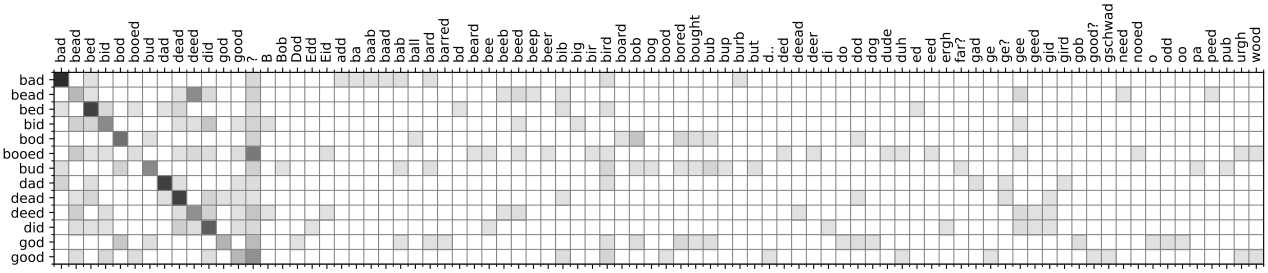

Figure 5: *Confusion matrix for online listeners (darker shades indicate higher values and white represents zero).*

significantly improved outcomes for 2 bilabial templates with tied onsets, with results for the remaining templates being similar. An analysis of the outputs confirm a higher number of misrecognitions due to consonant confusion for the free-onset configuration. This confirms that it is difficult to find the correct consonant articulation with the current simulation configuration in a limited number of iterations without prior knowledge of the consonant type.

4.2. Optimisation algorithms

With the tied-onset configuration, the outcomes using the different optimisation algorithms were compared with uniform sampling (*RND*). Figure 4 shows the overall recognition rates from the ASR system for a range of iteration limits and the listening test for 10k iterations (the listening test was limited to this setting and only compared RND and CRS). Firstly, CRS-based exploration resulted in significantly better results compared to uniform sampling at the 1k iteration limit. With 10k iterations the results from CRS and uniform sampling was similar both according to the ASR system and listeners. Secondly, overall recognition rates by listeners were not significantly different from rates with ASR (using Welch's unequal variances t-test: $p = 0.455$). However, more detailed analysis showed that recognition rates with ASR were significantly lower for "bod", "bud" and "did" and higher for "dad" and "dead". Reasons for lower rates may include lower language model weights for infrequent words ("bod" and "bud") and possibly hyper-articulation of "did" compared to what is typical in sentence context (function words are typically under-articulated [24]). For "dad" and "dead" higher confusion rates between /b/ and /d/ were found with listeners and "dead" was also often confused with "did".

The confusion matrix for online listeners (Figure 5) enables further analysis of the recognition rates over templates and possible sources of errors. The templates "bead", "booed" and "good" had the lowest recognition rates. The templates involving /uː/ were most frequently indicated as completely unintelligible and in the case of /buː/ the vowel often tends erroneously towards /i/. The case of "bead" is different, with the problem most frequently being consonant confusion with /d/ which seems to be part of a systematic problem of confusion between /b/ and /d/ when combined with /i, ɪ, ɛ, æ/. The fact that "god" is most often misrecognised due to the consonant and the parameter tying result (Figure 3) suggests that the tying configuration may need to be reconsidered in this case. Lastly, smaller systematic effects include /bæ/ tending toward /bɑ/ and some errors attributable to the coda.

5. Conclusion and discussion

This paper simulated the task of imitative articulatory exploration as a component of early vocal learning. Of particular relevance to this context are the results presented in terms of *intelligibility* and the fact that little prior information about articulatory targets are included in the process (only the consonant class used for parameter tying). The results are encouraging given that only global optimisation and a relatively simple objective function was used without attempting to maximise intelligibility. A summary of key findings are (using Welch's unequal variances t-test):

- The parameter tying configuration significantly improves success rates ($p = 0.033$), demonstrating its effectiveness in simulating coarticulation.
- The CRS algorithm is significantly more successful than uniform sampling when the number of iterations is limited ($p = 0.034$ at 1k).
- ASR success rates largely agree with results from our listening tests with a few exceptions.

As a result the approach and implementation presented here can serve as a technical and methodological basis for further work on early vocal learning, for example, to find samples for training articulatory to acoustic and inverse mappings without relying on predefined sets of articulatory targets.

Future work may include:

- Investigating and eliminating some of the systematic deficiencies pointed out in the results,
- Further testing the parameter tying configuration motivated by the *syllable-synchronised sequential target approximation* hypothesis in new scenarios (e.g. different consonant types and onset clusters), and
- Linking the task of articulatory exploration with other stages of early vocal learning, e.g. to learn constraints necessary for successful exploration and possibly to explicitly incorporate intelligibility into the simulation, potentially using ASR results as a feedback mechanism.

6. Acknowledgements

This work has been funded by the Leverhulme Trust Research Project Grant RPG-2019-241: "High quality simulation of early vocal learning".

7. References

[1] E. L. Saltzman and K. G. Munhall, "A Dynamical Approach to Gestural Patterning in Speech Production," *Ecological Psychology*, vol. 1, no. 4, pp. 333–382, Dec. 1989.

[2] P. K. Kuhl, "Early language acquisition: cracking the speech code," *Nature Reviews Neuroscience*, vol. 5, no. 11, pp. 831–843, Nov. 2004.

[3] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat, and R. Sproat, "TTS for Low Resource Languages: A Bangla Synthesizer," in *Proc. LREC*, Portorož, Slovenia, May 2016, pp. 2005–2010.

[4] A. Ragni, K. M. Knill, S. P. Rath, and M. J. Gales, "Data Augmentation for Low Resource Languages," in *Fifteenth Annual Conference of the International Speech Communication Association*. Singapore: ISCA, Sep. 2014, pp. 810–814.

[5] X. Cui, V. Goel, and B. Kingsbury, "Data Augmentation for Deep Neural Network Acoustic Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sep. 2015.

[6] P. Birkholz, *3D-Artikulatorische Sprachsynthese*. Berlin: Logos Verlag, 2005.

[7] ——, "Modeling Consonant-Vowel Coarticulation for Articulatory Speech Synthesis," *PLoS ONE*, vol. 8, no. 4, Apr. 2013.

[8] G. Bailly, "Learning to speak. Sensori-motor control of speech movements," *Speech Communication*, vol. 22, no. 2, pp. 251–267, Aug. 1997.

[9] I. S. Howard and M. A. Huckvale, "Training a Vocal Tract Synthesizer to Imitate Speech using Distal Supervised Learning," in *International Conference on Speech and Computer (SpeCom)*, Patras, Greece, 2005, pp. 159–162.

[10] A. K. Philippsen, R. F. Reinhart, and B. Wrede, "Learning how to speak: Imitation-based refinement of syllable production in an articulatory-acoustic model," in *International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, Genoa, Italy, Oct. 2014, pp. 195–200.

[11] P. K. Kuhl, R. R. Ramírez, A. Bosseler, J.-F. L. Lin, and T. Imada, "Infants' brain responses to speech suggest Analysis by Synthesis," *Proceedings of the National Academy of Sciences*, vol. 111, no. 31, pp. 11 238–11 245, 2014.

[12] A. Xu, P. Birkholz, and Y. Xu, "Coarticulation as synchronized dimension-specific sequential target approximation: An articulatory synthesis simulation," in *Proceedings of the International Congress of Phonetic Sciences (ICPhS)*, Melbourne, Australia, Aug. 2019, pp. 205–209.

[13] P. Birkholz, "Control of an Articulatory Speech Synthesizer Based on Dynamic Approximation of Spatial Articulatory Targets," in *Proc. Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 2865–2868.

[14] Y. Xu, "Syllable is a synchronization mechanism that makes human speech possible," *PsyArXiv*, Mar. 2020. [Online]. Available: https://osf.io/9v4hr

[15] B. Parrell, V. Ramanarayanan, S. Nagarajan, and J. Houde, "The FACTS model of speech motor control: Fusing state estimation and task-based control," *PLOS Computational Biology*, vol. 15, no. 9, p. e1007321, Sep. 2019.

[16] M. Murakami, B. Kröger, P. Birkholz, and J. Triesch, "Seeing [u] aids vocal learning: Babbling and imitation of vowels using a 3D vocal tract model, reinforcement learning, and reservoir computing," in *International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, Providence, Rhode Island, USA, Aug. 2015, pp. 208–213.

[17] Y. Gao, S. Stone, and P. Birkholz, "Articulatory Copy Synthesis Based on A Genetic Algorithm," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 3770–3774.

[18] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proc. Python in Science Conference (SciPy)*, Austin, Texas, USA, Jul. 2015, pp. 18–24.

[19] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential Model-Based Optimization for General Algorithm Configuration," in *Learning and Intelligent Optimization*. Berlin, Heidelberg: Springer, 2011, pp. 507–523.

[20] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr. 2006.

[21] P. Kaelo and M. M. Ali, "Some Variants of the Controlled Random Search Algorithm for Global Optimization," *Journal of Optimization Theory and Applications*, vol. 130, no. 2, pp. 253–264, Aug. 2006.

[22] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, Jan. 1965.

[23] S. G. Johnson, "The NLopt nonlinear-optimization package." [Online]. Available: http://github.com/stevengj/nlopt

[24] D. Jurafsky, A. Bell, E. Fosler-Lussiery, G. C., and R. W., "Reduction of English function words in Switchboard," in *Proceedings of the International Conference on Spoken Language Processing*, vol. 7, Sydney, Australia, 1998, pp. 3111–3114.