

OPTIMIZATION TECHNIQUES FOR A PHYSICAL MODEL OF HUMAN VOCALISATION

Mateo Cámara

Information Processing & Telecomm. Center
Universidad Politécnica de Madrid
Madrid, Spain
mateo.camara@upm.es

Zhiyuan Xu

Centre for Digital Music
Queen Mary University of London
London, UK
zhiyuan.xu@qmul.ac.uk

Yisu Zong

Centre for Digital Music
Queen Mary University of London
London, UK
y.zong@qmul.ac.uk

José Luis Blanco

Information Processing & Telecomm. Center
Universidad Politécnica de Madrid
Madrid, Spain
jl.blanco@upm.es

Joshua D. Reiss

Centre for Digital Music
Queen Mary University of London
London, UK
joshua.reiss@qmul.ac.uk

ABSTRACT

We present a non-supervised approach to optimize and evaluate the synthesis of non-speech audio effects from a speech production model. We use the Pink Trombone synthesizer as a case study of a simplified production model of the vocal tract to target non-speech human audio signals –yawnings. We selected and optimized the control parameters of the synthesizer to minimize the difference between real and generated audio. We validated the most common optimization techniques reported in the literature and a specifically designed neural network. We evaluated several popular quality metrics as error functions. These include both objective quality metrics and subjective-equivalent metrics. We compared the results in terms of total error and computational demand. Results show that genetic and swarm optimizers outperform least squares algorithms at the cost of executing slower and that specific combinations of optimizers and audio representations offer significantly different results. The proposed methodology could be used in benchmarking other physical models and audio types.

1. INTRODUCTION

Articulatory synthesis provides a unique opportunity to delve into the mechanics of speech production [1, 2]. Unlike black box models, physical models achieve an *interpretable representation* of the inner characteristics of the vocal tract. This allows for a deeper understanding of the processes involved in speech production. They also provide precise control of the speech’s articulatory, resonance, and phonatory characteristics, such as the position of the tongue, lips, existing constrictions, or nose size; as well as informed control of model parameters. This makes natural-sounding synthetic speech samples less prone to artifacts than other synthesis models. Furthermore, they may produce any type of human sound coming out of the mouth and the nose. Those include sounds that are not words, such as sighs, laughs, yawns, and so on.

These *non-speech sounds* are becoming increasingly important in today’s audiovisual productions and digital interactions.

From the sound effects in movies and videogames to the soundscapes in podcasts and audiobooks [3, 4, 5], the ability to generate these sounds has become a critical aspect to produce realistic performances. Analyzing the ability of models to construct these types of sounds is crucial to understand the limits and limitations of models [6], as well as the underlying complexities of producing naturally sounding audio samples. Answering those questions opens up new possibilities for sound and user-experience designers, video-game developers, and audio production professionals looking for new and innovative ways to create high-quality, realistic, and engaging sound experiences.

Physical models for speech synthesis pose challenges that are extensively reported in the literature. They often include many parameters that are difficult to configure simultaneously to achieve high-quality sounds. Their combined optimization can be demanding, computationally expensive, time-consuming, and challenging to implement in real time. These complications explain the need to improve and optimize the synthesizer.

Our research focuses on articulatory parameters from a black-box point of view. We optimize synthesizers without paying specific attention to what each parameter represents to maximize objective similarity by minimizing the difference between a target signal and the synthesized signal. This ensures superior generalization capabilities for the proposed method and valuable results for other contexts.

In this contribution, we look at the physical model known as the *Pink Trombone (PT)*¹. This is a simplified version of the vocal tract that uses a small set of fundamental parameters to control the shape and movements of the articulators during speech production [7]. We fixed its articulatory bounds to focus on sounds that a human could physically produce, and used these to optimize the PT and compare its result with human audio samples.

We conduct a case study using synthetic, sustained, and yawning sounds to understand its capabilities and limitations. We test different black-box strategies to predict the synthesizer control parameters, including well-known *optimization techniques* and *Deep Neural Networks*, trained on a set of PT synthetic samples. For experimentation, we use sound files generated by the PT, as well as audio clips downloaded from the *Freesound* platform [8].

Experiments shall lay the foundations for studies on articulatory and production models with multiple parameters and different

Copyright: © 2023 Mateo Cámara et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author and source are credited.

¹<https://dood.al/pinktrombone/>

audio types. The dataset, test sounds, and algorithms are available online². Our goals for this contribution are the following:

- *Determine if PT parameters can be accurately predicted exclusively from acoustic features.* We optimize synthesizer control variables from audio samples as a black-box.
- *Determine best optimization technique for articulatory variables.* We evaluate how different optimizers perform in front of increasingly challenging sounds.
- *Determine the error metric that yields a more satisfactory outcome.* We benchmark different techniques for standard error metrics and acoustic parameterizations of audio files.

The remainder of this paper is organized as follows. Sec. 2 expands on the optimization techniques and the parameterizations reported in the literature. Sec. 3 describes the experiments covered to meet our objectives. Sec. 4 analyzes and discusses the results obtained, and Sec. 5 concludes the paper.

2. BACKGROUND

For sound-matching optimization, one may focus on the control parameters of the synthesizer, the input acoustic features extracted from the audio, and the process that leads to optimization. All these aspects provide insights into the methodologies and objectives of various optimization studies in synthesizers. Fig. 1 depicts the overall schematic of the optimization process.

2.1. Optimization Methods

Considering the complexity of sound synthesizers, there is a need for reliable optimization techniques. Numerous optimization methods have been investigated in terms of optimizing parameters for physical models or traditional synthesizers. The related work can be organized into two main categories:

Search-based Methods: these have been widely applied to physical models in audio synthesis due to their ability to handle non-differentiable, non-linear, and non-convex optimization problems. They are universal and regard the synthesizer as a black-box model, focusing solely on parameter space optimization. Standard ways include the use of Evolutionary Algorithms (EA), including Evolution Strategies [9], Genetic Algorithm (GA) [10], or Particle Swarm Optimization (PSO) [11]. Other methods, including Hill Climber [12], Levenberg–Marquardt Algorithm [13] and Nelder-Mead Method [14] are also considered.

Model-based Methods: machine learning (ML) models have become mainstream for synthesizer parameter estimation in recent years. They learn the mapping between the latter and audio features directly from data. In [15], authors used a strided Convolutional Neural Network (CNN) to predict the parameters of a subtractive synthesizer. Recent work proposed differentiable digital signal processing (DDSP) [16] and integrated an additive synthesizer with a filtered noise synthesizer into the end-to-end deep learning framework. These allow direct gradient descent optimization. DDSP is now widely utilized for parameter estimation [17], despite its need for precise reproduction of the target synthesizer in a differentiable manner, which poses difficulties.

²<https://slash-trombone.github.io/>

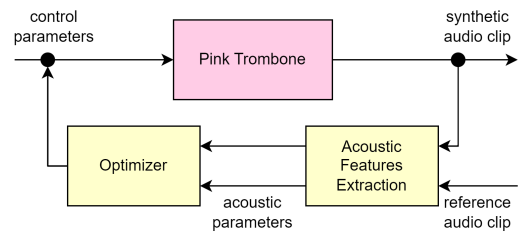


Figure 1: Schematic on the optimization process.

Each approach has its own benefits and limitations, leading to ongoing discussions. In [12], authors compared sound-matching performance on a VST synthesizer using two search strategies and three neural network methods. Results indicated that search methods are limited by their computational cost, and modeling methods are restricted to the inductive bias of model structure and data availability. We tested these limitations for the PT, including simple speech and non-speech vocalisations to evaluate the performance of the optimized model parameters to reproduce sounds.

2.2. Parameter Selection

The control parameters of the synthesizer and the input parameters for the optimizer largely depend on the synthesis technique and the desired outcomes, in accordance with Fig. 1. We focus on the PT control parameters –see Table 1. Physical models may alternatively use local constrictions to describe the configuration required for the vocal tract to produce a certain sound. The PT can actually operate on those as well. Nonetheless, we are interested in the primary ones.

Furthermore, inputs to the optimizer must represent the acoustic content of the audio samples so that the model may produce accurate outputs. For this task, we shall look at acoustic features.

2.3. Acoustic Features Extraction

Various acoustic features have been used in synthesizer optimization studies to evaluate and quantify the quality of the synthesized sounds to guide the optimization process. Spectral features are the most common, but finding the best metric with good perceptual consistency is still an open question [18]. In [19] authors focus on the spectral norm error, [10] used spectral norm plus spectral centroid error extracted from short-time Fourier transform (STFT) for each frame, [9] used relative spectral error, which is computed by summing normalized differences between frequency components extracted from two spectra, [20] combined the least squared error of the STFT of two sounds plus the perceptual error applying a narrow band masking curve. On error computation, [12] used Euclidean distance of Mel-Frequency Cepstral Coefficients (MFCCs), [16] used a deep representation extracted from MFCCs, and multiscale spectral loss plus perceptual loss, [15] compared the following features as the Deep Neural Networks (DNN) input: a set of spectral features [21], STFT, and deep representation extracted by a CNN from the raw signal. Results showed that STFT and deep representations seem more representative than handcrafted features. Our aim now is to identify suitable ones for the PT parameters optimization.

3. EXPERIMENTATION

We designed our experiments to focus on three specific characteristics that are relevant to the acoustic-to-articulatory inversion: the *optimizer* used to predict control variables, the *audio representation* to compute the difference between original and synthesized audio, and the *signal complexity*.

3.1. Pink Trombone Fundamentals

The PT is a simple vocal tract model that can be interacted with through a web interface. It is a Kelly-Lochbaum (KL) type model whose technical details can be found in [22]. In our black-box case study, we focused on the number of parameters to be used and their bounds. Table 1 summarizes these, which correspond to physical attributes of the vocal tract. We aimed to decouple the meaning of these parameters from their human meaningfulness for our method to be useful to any synthesizer.

Table 1: Pink Trombone parameters and their bounds.

Pink Trombone Parameters	Lower bound	Upper bound
Pitch (Hz)	75	330
Voiceness	0	1
Tongue Index	14	27
Tongue Diameter (cm)	1.55	3
Lips Diameter (cm)	0.6	1.2
Constriction index	12	42
Constriction Diameter (cm)	0.6	1.2
Throat Constriction (cm)	0.5	1.0

3.2. Signal complexity

Signal complexity refers to the challenges we pose to the optimizers to predict the exact parameters. In that sense, we consider three independent characteristics of the signal. First, the *origin of the audio file*: audio generated by a speech synthesizer or a person. Second, *variations over time*: sustained notes or dynamic audio (such as a yawn). Third, *number of variables to optimize*: related to the characteristics of the synthesizer. Hereafter we enumerate all experiments conducted from the least to the most complex.

- PT generated sounds for which:
 - One of the control parameters is unknown.
 - All of the control parameters are unknown.
 - Gaussian white noise is added. This evaluates the robustness of optimizers dealing with non-ideal signals.
 - Control parameters vary over time.
- Audio clips containing:
 - Sustained vowel sounds.
 - Yawnings.

3.3. Audio Representation and Quality Assessment

3.3.1. Representations focusing on spectral difference

To minimize the difference between the target and reconstructed sound, we focused on the spectral features of the audio signals. The following list includes all the transformations evaluated:

- *STFT*. A window size of 1024 samples with a 2x overlap STFT was taken.
- *Multiscale spectrogram*. The window sizes were {64, 128, 256, 512, 1024}, with a 75% overlap.
- *MEL-spectrogram*. We used 128 filters in the MEL bank up to a maximum frequency of 8 KHz.
- *MFCCs*. We took 20 cepstral coefficients from the MEL-spectrograms.

Computations were performed in Python 3.9, using the Au-raLoss library [23]. The Mean Absolute Error (MAE) between the input and reconstructed audio was computed as the error function.

3.3.2. Perceptual metrics

In addition to MAE, we also computed a set of perceptual quality and intelligibility metrics. These metrics were not used as error functions in the optimization process. The findings may be representative of the perceptual similarity between sounds. However, we encourage readers to listen to the results we posted online. The following full reference metrics were analyzed:

- PESQ: Perceptual Evaluation of Speech Qlt. [24].
- PEAQ: Perceptual Evaluation of Audio Qlt. [25].
- ViSQOL: Visually-Inspired Speech Qlt. Obj. Listener [26].
- STOI: Short-Time Objective Intelligibility [27].

3.4. Selected Optimizers

We used *optimization algorithms* and a *CNN* to predict the control parameters of the synthesizer. We fed the algorithms with the MAE between the original and the synthesized signal. Hereafter we briefly introduce the selected optimization algorithms, which we have evaluated in terms of computational cost and reconstruction error.

Genetic Algorithm (GA): is an optimization technique inspired by natural selection and genetics [28]. The candidate solutions are defined by a set of genes. In every generation (loop over all candidates), the genes are able to randomly change (mutation), combine with other candidates (crossover), and be selected (optimization) to search for optimal solutions in the solution space. The fitness function seeks to minimize differences in the input/output signals.

We used 32 bits to define the genes, a crossover rate of 0.9, a mutation rate of 0.03, and a population of 10 individuals.

Particle Swarm Optimization (PSO): is a nature-inspired meta-heuristic optimization technique that simulates the social behavior of swarms [29]. PSO operates by iteratively adjusting the position of particles within the search space based on their individual and global best experiences, converging towards the optimal solution. In our case, we set acceleration parameters $c_1 = 0.5$, $c_2 = 0.3$ (trust in itself, trust in its neighbors), and inertia weight $w = 0.9$, with 10 particle population.

Trust Region reFlective (TRF): The Trust Region reFlective [30] algorithm is a computational technique for solving least squares optimization problems. It employs a model-based method, seeking to minimize a function by iteratively creating simplified models of

the objective function within certain trusted regions. The term “reflective” refers to the method’s way of handling boundaries and constraints: if a proposed step hits a boundary, it is reflected in the feasible region.

Nelder-Mead Method (NM): also known as the downhill simplex method [31], is a multidimensional optimization technique well suited for non-linear problems. The algorithm starts with an initial simplex, a set of $n+1$ points in an n -dimensional space. The algorithm iteratively updates the position of the simplex by reflecting, expanding, contracting, or shrinking it, based on the values of the function being optimized at the vertices of the simplex.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES): is a stochastic optimization algorithm that uses information about the distribution of the samples generated by the algorithm to guide the search for the optimal solution [32]. The algorithm starts with an initial guess for the solution and then generates a set of samples around this point. The distribution of these samples is then updated based on the fitness of the samples, with higher-fitness samples being more likely to be selected. As the algorithm progresses, outcomes increasingly concentrate around the optimal solution.

Neural network prediction (NN): In addition to the optimization techniques, we tested the capability of neural networks to predict the control variables based on the acoustic features. We designed a CNN that admits spectrograms as input and outputs the control variables. To train it we collected a database of 400,000 different PT clips. We trained four different networks, each admitting as input for each audio representation mentioned in subsection 3.3.1. The network has been coded with Pytorch 1.7.1, with 2 convolutional layers, ReLU as the activation function, 0.0001 as the learning rate, ADAM optimizer, and following the 60/20/20 data splitting strategy between training, validation, and test.

3.5. Materials

Attending to the scope of this contribution, the assessment of the performance achieved by the different techniques, audio representations, and optimizers in predicting the parameters of the physical model for sound-matching required two sets of audio files:

- Synthetic audio samples, generated at 48 kHz sampling rate and 1 s long. To generate these, we used the Programmable version of the PT³ modified to be a Node.js server. We generated 80 audio clips with random control parameters.
- Audio samples downloaded from Freesound containing utterances from different speakers. We focused on sustained vowels (5 clips) and yawnings (8 clips). The vowels are one second long and the yawnings are three seconds on average. All files were recorded at a 48 kHz sampling rate to match the same conditions as the synthetic audio.

4. RESULTS

In this section, we present the results of the experiments aimed at predicting control parameters for PT. We sought to fix the same conditions for all optimizers to ensure a fair comparison. Some considerations apply to all experiments:

³<https://github.com/zakaton/Pink-Trombone>

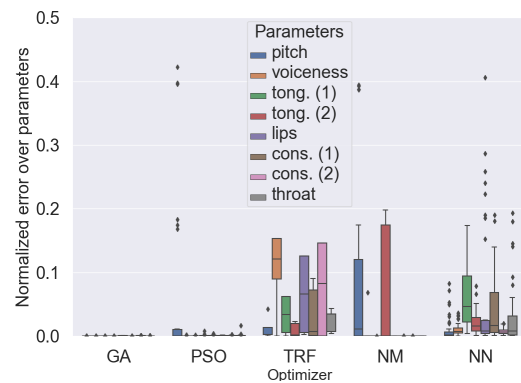


Figure 2: Optimizer performance over one control parameter. X-axis includes the optimizers. Y-axis represents the normalized error. Each bar is a control parameter.

- Error values are normalized with respect to the maximum and minimum values that each parameter can take.
- The random seed was fixed to randomize the PT control parameters in each experiment, such that the optimizers face the same initial conditions in all cases.
- Each experiment was repeated 20 times. Initial conditions and target values were randomized.
- All optimizers had the same stop criterion: reach an error of less than 0.0001 in the metric or stop to improve the relative error with respect to the previous 20 loops.

4.1. Optimization of PT-generated sounds

Hereafter we present the results of the different tests that were conducted using PT synthetic audio clips as inputs.

4.1.1. Optimization of one control parameter

In this experiment, we fixed all control parameters except for one. We predicted the unknown value. This set of experiments does not include the CMA-ES algorithm because its particular design does not support single-parameter prediction. The results are shown in Figure 2, for the different optimizers and PT control parameters.

Results demonstrate the effectiveness of GA and PSO in accurately predicting the control parameters. No outliers were observed in the genetic algorithm’s performance. The NM algorithm successfully reached the absolute minimum for most parameters. However, it struggled to achieve the same for the pitch and one tongue-related parameter. A closer examination of these parameters revealed that their error functions contained multiple local minima. Since the performance of the NM is heavily influenced by its initial conditions, it makes it prone to getting stuck in them.

Despite not always arriving at the optimal values, TRF and NN converge rapidly to the minimum. Once it is trained, NN takes less than a second to reach the minima. TRF algorithm takes 5 seconds on average, which is four times faster to optimize than PSO and NM, and 100 times faster than GA.

In the same line, Figure 3 illustrates the error associated with each audio representation. All audio representations are suitable for optimizing individual control parameters. However, no error is observed in the multiresolution. This makes sense, since it is an extension of the STFT that better represents the spectral characteristics of the signal.

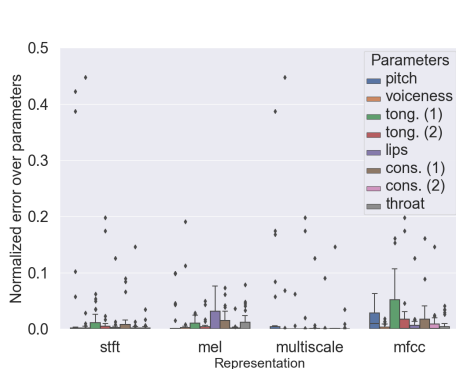


Figure 3: Audio representation performance over one control parameter. X-axis represents each audio representation. Y-axis represents the normalized error. Each bar is a control parameter.

4.1.2. All control parameters

The single-parameter experiments validate that articulatory parameters can be predicted from sound representations alone. However, this scenario is too simplified to clarify which optimizer is more accurate. This can be done by increasing the complexity of the experiment, seeking to predict all control parameters at once. The prediction results for each parameter are shown in Figure 4.

Experiments focusing on predicting all parameters demonstrate the superior performance of GA, CMA-ES, and PSO compared to other methods. In this experiment set, the eight-dimensional search area makes the optimization more challenging. The TRF and NM algorithms yielded unsatisfactory results, deeming them unsuitable for tackling the problem. As the number of potential solutions grows exponentially with increasing dimensions, only the most robust methods can find an optimal solution. Genetic algorithms and PSO can outperform least squares minimization or the downhill simplex method because they are more robust in handling complex search spaces, non-convex functions, and intricate relationships between variables.

On the other hand, observing how the different audio representations behave in this scenario is interesting. They are shown in Figure 5. We can observe that no representation performs significantly better than the rest, not even the multiresolution represen-

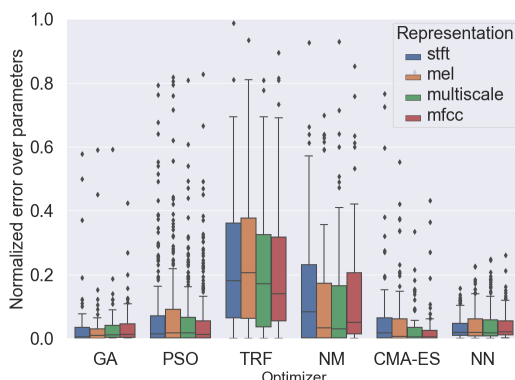


Figure 4: Optimizer performance when all parameters are predicted at the same time. X-axis includes all optimizers. Y-axis represents the normalized error. Each bar corresponds to an audio representation.

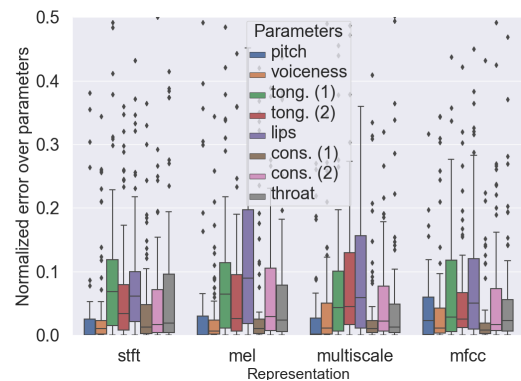


Figure 5: Audio representation performance while predicting all parameters at a time. X-axis covers the representations. Y-axis represents the normalized error. Each bar corresponds to a control parameter.

tation. However, finding a higher error is not necessarily a serious problem when reconstructing the signal. Most control parameters have local minima very close to the global minimum. This means that different PT configurations can produce almost the same reconstruction. This does not apply to the pitch parameter, which is one of the critical parameters in quality and, as can be seen, the MFCCs do not make it easy to reach its minimum.

In fact, Figure 6 illustrates precisely these phenomena. It shows the MAE of the original and reconstructed signal. It is observed that regardless of the optimizer used when the search space is large, the MFCCs do not achieve satisfactory results. Thus, this experiment indicates that the best prediction of the control parameters can be made with GA or the PSO using the MEL scale or Multiresolution spectrograms.

In addition, Figure 7 shows the computational costs of each optimizer. It shows that the NN is the fastest once trained, while PSO is the fastest of the suitable optimization techniques.

4.1.3. All control parameters which vary over time

The next level of complexity we tested was optimizing parameters that varied over time. To achieve this, we created an interpolator that generated intermediate values between two temporal spaces

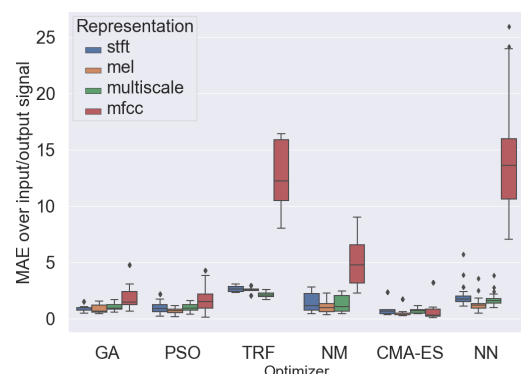


Figure 6: Absolute performance of the optimizers and representations. X-axis includes all optimizers. Y-axis represents the MAE of the target and reconstructed audio file. Each bar is the audio representation.

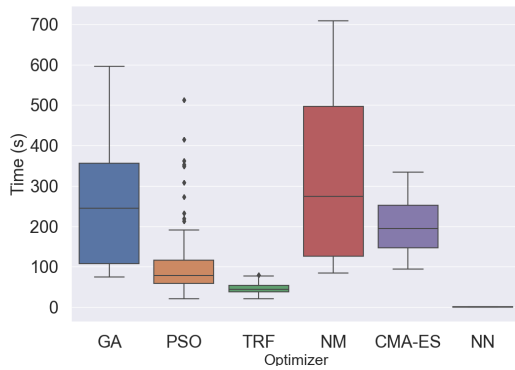


Figure 7: Computational cost for the different optimizers –in X -axis. Y-axis represents the time to converge, in seconds.

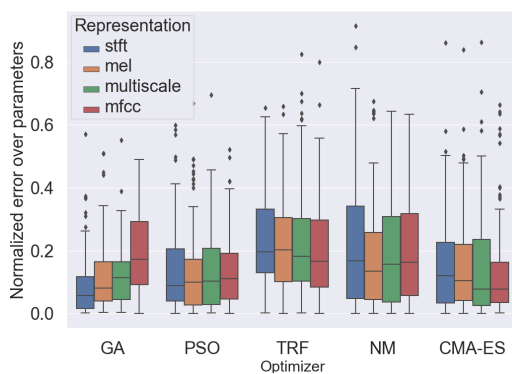


Figure 8: Performance of optimizers and representations when the signal varies over time. X-axis includes all audio optimizers. Y-axis represents the normalized error between the target and reconstructed audio. Each bar corresponds to an audio representation.

defined by two sets of articulatory parameters in the PT. As shown in Figure 8, none of the optimizers were able to achieve a satisfactory result when optimizing a time-variant set of parameters. The only optimizer that achieved a result closer to zero was the GA, using the STFT. The tendency is that as more parameters are added to optimize, the search space becomes more complex and therefore very difficult to reach the absolute minimum. It is important to note that this method is not suitable for a neural network. It would be necessary to train new networks depending on the number of parameters to be predicted.

To achieve a more satisfactory, general solution, it was decided that the best strategy for optimizing signals that vary over time is to segment the signal into small windows and optimize them as if they were a static signals. We tested different window sizes and found out a 100 milliseconds length performed optimally. These windows can then be connected using a Savitzky-Golay filter [33], which smooths out the result. The optimization results of these tests suggest insights equivalent to predicting a non-variant set of parameters. That is why this technique is the preferred choice for optimizing sounds created by humans.

4.1.4. All control parameters in noise

In these experiments, different amounts of Gaussian white noise were added to the original signal. We sought to predict the articu-

latory parameters that defined the signal. As shown in Figure 9, the optimizers performance deteriorates as more noise is added. Additionally, we observed that the optimizers do not start to exhibit exponentially growing errors until the signal-to-noise ratio reaches 20 decibels. All optimizers act similarly, with the exception of the NN, which does not tolerate noise at its input.

From these experiments, we can conclude that it is possible to optimize signals that are not perfectly generated by a synthesizer but may come from any source, such as a recording from a public database. This finding is significant because it suggests that our approach can be applied in real-world scenarios where the input signals will likely not be perfectly recorded.

4.2. Optimization of real audio files

Results of the tests with real sounds can be seen in Table 2. The results for each perceptual metric are shown for the best-performing combination of the optimizer-representation pair. The columns detail the optimizers and the color shows the best audio representation for each case. We used different perceptual metrics to measure how similar the sounds generated by the synthesizer were to human-generated ones. We also include how the perceptual metrics behaved when predicting PT samples. These set up a benchmark to compare the upper limit that could be reached. Still, we encourage readers to visit our website, where we have published these audio files, and evaluate the quality themselves.

As shown in the table, CMA-ES and GA achieved superior results compared to other optimizers in terms of perceptual similarity in most of the cases. It is important to note that the MOS (Mean Opinion Score) equivalent results can still be considered low compared to the scale, as the sounds are synthesized by a certain vocal tract that may not correspond to the vocal tracts of the people who generated the original sound. Therefore, we do not claim that we can produce an exactly identical sound but an equivalent one.

For all types of signal, we used the strategy of dividing the signal into small windows and smoothing them out into full-length signals. Systematically, PT-generated sounds are predicted with better scores than human-generated sounds. Furthermore, in many of the cases we found that yawns are perceptually recognized as more similar than sustained vowels. This is because the timbre in the sustained vowel has a much greater influence than in the yawn. The PT has vocal characteristics that do not match those of the

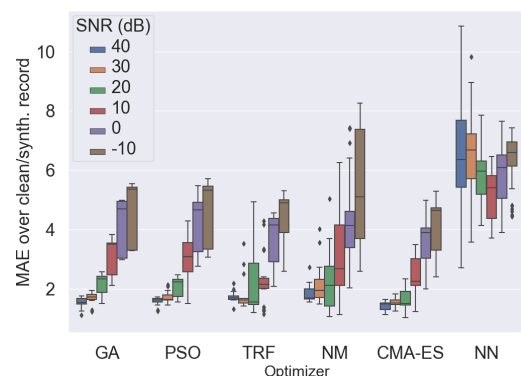


Figure 9: Performance of the optimizers when Gaussian White Noise was applied at the input. X-axis includes all audio optimizers. Y-axis represents the MAE between the target and the reconstructed audio file. Each bar is a different Signal-to-Noise ration.

Table 2: Perceptual equivalent metrics of the real sounds. Type "PT" stands for "Pink Trombone" generated, "VW" for sustained "Vowel", and "Y" for "Yawn". All metrics are in MOS scale (from 1 to 5) except STOI (from 0 to 1). We used a color code to indicate the best-performing set of acoustic parameters per audio type and optimizer.

		GA	PSO	TRF	NM	CMA-ES	NN	Best	Result	Legend
PESQ	PT	2.2 ± 0.9	2.1 ± 1.2	1.8 ± 1.0	2.2 ± 1.0	2.6 ± 0.9	1.8 ± 0.9	CMA-ES	2.6 ± 0.9	mel
	VW	1.8 ± 0.8	1.5 ± 0.4	1.6 ± 0.6	1.8 ± 0.7	1.5 ± 0.5	1.8 ± 1.2	GA	1.8 ± 0.8	mfcc
	Y	1.5 ± 0.4	1.3 ± 0.3	1.4 ± 0.2	1.3 ± 0.1	1.3 ± 0.2	1.5 ± 0.7	GA	1.5 ± 0.4	multiscale
PEAQ	PT	3.0 ± 0.6	3.2 ± 0.8	2.6 ± 0.6	3.0 ± 0.9	3.5 ± 0.7	3.0 ± 0.8	CMA-ES	3.5 ± 0.7	stft
	VW	2.8 ± 0.7	2.2 ± 0.1	2.6 ± 0.7	2.5 ± 0.7	2.5 ± 0.6	2.6 ± 0.7	GA	2.8 ± 0.7	
	Y	2.5 ± 0.8	3.0 ± 1.1	2.7 ± 0.7	2.6 ± 0.7	2.7 ± 1.0	2.8 ± 1.0	PSO	3.0 ± 1.1	
ViSQOL	PT	3.1 ± 0.9	3.4 ± 0.8	1.7 ± 0.5	3.3 ± 1.2	4.3 ± 0.7	3.0 ± 0.6	CMA-ES	4.3 ± 0.7	
	VW	1.9 ± 0.5	2.0 ± 0.2	2 ± 0.3	1.9 ± 0.3	2.1 ± 0.4	1.8 ± 0.4	CMA-ES	2.1 ± 0.4	
	Y	2.1 ± 0.1	2.1 ± 0.1	2.3 ± 0.3	2.3 ± 0.4	2.1 ± 0.1	1.9 ± 0.2	TRF	2.3 ± 0.3	
STOI	PT	0.3 ± 0.2	0.4 ± 0.3	0.1 ± 0.1	0.5 ± 0.4	0.5 ± 0.3	0.2 ± 0.1	CMA-ES	0.5 ± 0.3	
	VW	0.1 ± 0.1	0.1 ± 0.0	0.1 ± 0.0	0.1 ± 0.1	0.1 ± 0.0	0.1 ± 0.0	CMA-ES	0.1 ± 0.0	
	Y	0.3 ± 0.1	0.3 ± 0.1	0.3 ± 0.1	0.3 ± 0.1	0.3 ± 0.1	0.1 ± 0.1	-	0.3 ± 0.1	

person who recorded the sounds. For this reason, it is more difficult to recreate a perfectly harmonic voice like the vowel than a noisy sound like the yawn. This does not imply that our optimizers are malfunctioning, as the goal is to create comparable sounds, not exactly the same. The STOI metric in this regard is very representative of this situation, giving the yawn almost equal score to the PT-generated values, while the vowel is perceived as different.

These experiments also yield two interesting insights. First, one may identify optimizer-representation combinations that perform better than others. In particular, multiscale representation works well for yawns, while for sustained vowels STFT representation can do the job. None performed well using MFCC. Thus, one may need to take into account the type of signal to get good results from the optimizer. Second, there is consistency among the perceptual metrics. Those experiments that are more challenging consistently score worse than simpler ones.

5. CONCLUSION

Optimization techniques effectively predict the parameters of the Pink Trombone to produce human-like vocalisations. The selected algorithms delivered tuned control parameters while operating on different acoustic features and metrics. The resulting audio samples match the selected input sounds regarding the absolute error and perpetual equivalent metrics. A similar trend was observed on sustained vowels and yawnings; as well as under additive noise conditions. Nonetheless, the lower performance levels for the collected audio samples compared to the synthetic inputs, in absolute error and according to the perceptual-equivalent metrics, may be influenced by the limited ability of the Pink Trombone to match sounds out of its standard tract setting.

We comprehensively evaluated some of the most commonly used optimization algorithms in a black-box approach, predicting their control parameters to synthesize non-speech sounds. We tested different audio representations and conducted experiments in different scenarios, ranging from simple single-parameter predictions to complex, time-varying parameters or non-speech human-made sounds. Our results show that the Evolution Strategies (GA and CMA-ES) and Particle Swarm Optimization with multiresolution representation are the most effective for predicting control parameters with minimum error (MAE < 1%) and high quality (ViSQOL 4.3 for PT, PEAQ 3.0 for yawns). Also, PSO achieved the best performance vs. computational cost ratio.

According to our results, GA and PSO algorithms were superior to the other optimization methods in most cases. The NM algorithm struggled with local minima, and the TRF algorithm, although fast, could not optimize the parameters satisfactorily. The NN could predict the control parameters when the input was a sound generated by the PT. However, it failed when confronted with real sounds or noisy inputs. NN strategy can be useful for certain scenarios since it is also very fast once trained, but it can hardly reach the generalization of the GA.

Regarding audio representations, our experiments demonstrate that all those that we tested, including MFCC, STFT, MEL, and multiresolution decomposition, are suitable for optimizing individual control parameters. However, the MFCC representation showed poorer pitch prediction capability than other representations. This is consistent with what is expected from a cepstral representation according to the literature.

Perceptual metrics validate that the optimizers are able to faithfully predict audio samples generated by the synthesizer itself. Taking this benchmark, we can observe that real sounds do not reach such a high performance. The conclusion is that our synthesizer cannot achieve certain characteristics of real voices in the given conditions (e.g. vocal tract size). Nevertheless, comparable sounds have been achieved, which is the goal of our research.

Future research should explore new techniques that enhance the prediction of time-varying signals. Further analysis is needed to investigate the parameter's flow, whether it aligns with the typical structures of a human vocal tract or the chosen optimization strategy. Additionally, hierarchical optimizations can improve the neural network's performance. Predictions should be conducted in two stages by initially narrowing the bounds and then fine-tuning.

Finally, future work should use this framework to benchmark other solutions to the problem, including alternative optimization methods, acoustic features, metrics, and subjective tests.

6. ACKNOWLEDGMENTS

The authors would like to thank David Südholt for the valuable discussions, support and help. Activities described in this contribution were partially funded by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 101003750, the Ministry of Economy and Competitiveness of Spain under grant PID2021-128469OB-I00, and the UPM Research Programme, *Programa Propio de I+D+I 2022*.

7. REFERENCES

- [1] Christopher T Kello and David C Plaut, “A neural network model of the articulatory-acoustic forward mapping trained on recordings of articulatory parameters,” *The Journal of the Acoustical Society of America*, vol. 116, no. 4, pp. 2354–2364, 2004.
- [2] Sandesh Aryal and Ricardo Gutierrez-Osuna, “Data driven articulatory synthesis with deep neural networks,” *Computer Speech & Language*, vol. 36, pp. 260–273, 2016.
- [3] M Mehdi Afsar et al., “Generating diverse realistic laughter for interactive art,” *arXiv preprint arXiv:2111.03146*, 2021.
- [4] Chin-Cheng Hsu, “Synthesizing personalized non-speech vocalization from discrete speech representations,” *arXiv preprint arXiv:2206.12662*, 2022.
- [5] Andrey Anikin, “Soundgen: an open-source tool for synthesizing nonverbal vocalizations,” *Behavior research methods*, vol. 51, no. 2, pp. 778–792, 2019.
- [6] Aaron van den Oord et al., “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [7] Brad H Story, “A parametric model of the vocal tract area function for vowel and consonant simulation,” *The Journal of the Acoustical Society of America*, vol. 117, no. 5, pp. 3231–3254, 2005.
- [8] “Freesound technical demo,” in *ACM International Conference on Multimedia (MM’13)*, Barcelona, Spain, oct. 2013, ACM, pp. 411–412.
- [9] Thomas J Mitchell and David P Creasey, “Evolutionary sound matching: A test methodology and comparative study,” in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. IEEE, 2007, pp. 229–234.
- [10] Yuyo Lai et al., “Automated optimization of parameters for fm sound synthesis with genetic algorithms,” in *International Workshop on Computer Music and Audio Technology*. Cite-seer, 2006, p. 205.
- [11] Jorge Zúñiga and Joshua D Reiss, “Realistic procedural sound synthesis of bird song using particle swarm optimization,” in *Audio Engineering Society Convention 147*, 2019.
- [12] Matthew John Yee-King et al., “Automatic programming of vst sound synthesizers using deep networks and other techniques,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, 2018.
- [13] G Pyz et al., “Lithuanian speech synthesis by computer using additive synthesis,” *Elektronika ir Elektrotechnika*, vol. 18, no. 8, pp. 77–80, 2012.
- [14] MT Vakil Baghmisheh et al., “Frequency modulation sound parameter identification using novel hybrid evolutionary algorithms,” in *2008 International Symposium on Telecommunications*. IEEE, 2008, pp. 67–72.
- [15] Oren Barkan and David Tsiris, “Deep synthesizer parameter estimation,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 3887–3891.
- [16] Jesse Engel et al., “Ddsp: Differentiable digital signal processing,” *arXiv preprint arXiv:2001.04643*, 2020.
- [17] Naotake Masuda and Daisuke Saito, “Synthesizer sound matching with differentiable dsp,” in *ISMIR*, 2021, pp. 428–434.
- [18] David Moffat and Joshua D Reiss, “Perceptual evaluation of synthesized sound effects,” *ACM Transactions on Applied Perception (TAP)*, vol. 15, no. 2, pp. 1–19, 2018.
- [19] Daniel M Munoz et al., “Opposition-based shuffled pso with passive congregation applied to fm matching synthesis,” in *IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2775–2781.
- [20] Janne Riionheimo and Vesa Välimäki, “Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2003, pp. 1–15, 2003.
- [21] Katsutoshi Itoyama and Hiroshi G Okuno, “Parameter estimation of virtual musical instrument synthesizers,” in *ICMC*, 2014.
- [22] John Kelly, “Speech synthesis,” *Proc. 4th Int. Congr. Acoustics, 1962*, pp. 1–4, 1962.
- [23] Christian J. Steinmetz and Joshua D. Reiss, “auraloss: Audio focused loss functions in PyTorch,” in *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.
- [24] Antony W Rix et al., “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, 2001, vol. 2, pp. 749–752.
- [25] Thilo Thiede et al., “Peaq - the itu standard for objective measurement of perceived audio quality,” *Journal of the Audio Engineering Society*, vol. 48, no. 1/2, pp. 3–29, 2000.
- [26] Michael Chinen et al., “Visqol v3: An open source production ready objective speech and audio metric,” in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.
- [27] Norman R French and John C Steinberg, “Factors governing the intelligibility of speech sounds,” *The journal of the Acoustical society of America*, vol. 19, no. 1, pp. 90–119, 1947.
- [28] John H Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.
- [29] James Kennedy and Russell Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*. IEEE, 1995, vol. 4, pp. 1942–1948.
- [30] Mary Ann Branch, Thomas F Coleman, and Yuying Li, “A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems,” *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 1–23, 1999.
- [31] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 01 1965.
- [32] Nikolaus Hansen and Andreas Ostermeier, “Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation,” 06 1996, pp. 312 – 317.
- [33] Abraham. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.