TARGETOPTIMIZER 2.0: ENHANCED ESTIMATION OF ARTICULATORY TARGETS

Paul Konstantin Krug, Simon Stone, Alexander Wilbrandt, Peter Birkholz

Technische Universität Dresden
paul_konstantin.krug@tu-dresden.de

Abstract: The TARGET-APPROXIMATION-MODEL (TAM) is a syllable-based model to represent articulatory trajectories and pitch contours by a compact set of target parameters. In this work, we extended the TAM by an additional degree of freedom, namely the position of the target boundaries. This feature was integrated in the open-source software TARGETOPTIMIZER (TO), which is now officially updated to version 2.0 (TO2). The TO2 transfers articulatory trajectories into TAM parameters using a joint optimization with regularization. We demonstrate that the boundary optimization significantly reduces the modelling error compared to the case without boundary optimization. As an additional feature, the TO2 allows to initialize the boundaries uniformly across an utterance. Therefore, a prior determination of the syllable boundaries is no longer a necessary requirement in order to find the optimal targets. The TO2 is written in C++ and available for download as an open-source software under the GNU General Public License.

1 Introduction

The TARGET-APPROXIMATION-MODEL (TAM) [1, 2] is a well-known intonation model that does not only allow the generation of very natural pitch contours, but can also be used to produce realistic trajectories of articulators, which is especially useful for articulatory speech synthesis [3]. To create highly realistic pitch or articulatory trajectories, natural acoustic or articulatory speech data can be used as a reference. However, in order to transfer pitch or articulatory measurements into the TAM representation, the parameters of the model must be adapted to the measured data. The current state-of-the-art method to perform such a model fit is provided by the open-source software TARGETOPTIMIZER 1.0 (TO1) [4]. This program jointly optimizes all targets of a given utterance, while using a regularization mechanism to prevent the algorithm from preferring unnatural target parameters. Despite the fact that the TO1 can produce natural sounding pitch contours, we found that its performance is suboptimal under certain circumstances. It was observed that the used (fixed) target boundaries, that are given by the syllable boundaries of the utterance, do not correspond to the optimal target boundary positions (see Figure 1), which is a restriction of the TAM itself. Further, due to the simplistic nature of the TO1, the input and output (I/O) options are quite limited, which can make huge batch processing operations difficult. Last but not least, the graphical user interface (GUI) provides only limited information and options to the user, which is not ideal as well.

In this work, we present the official update of this software[1], the TARGETOPTIMIZER 2.0 (TO2). In this version, we extended the TAM by an additional degree of freedom, namely the

[1]Software available under:
https://github.com/TUD-STKS/TargetOptimizer
https://vocaltractlab.de/index.php?page=targetoptimizer-download

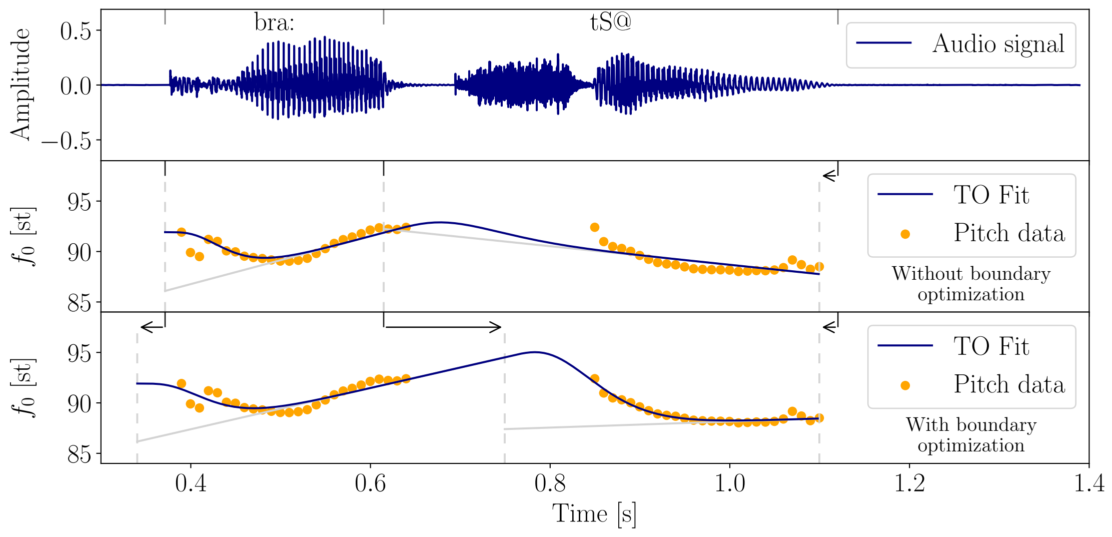

Figure 1 – The top plot shows the digital waveform of the recorded German utterance "*Bratsche*" (English: *viola*, SAMPA: /bra:tS@/). The middle and bottom plots show the corresponding measured pitch data points and the TO2 fit without and with boundary optimization, respectively. The vertical, dashed lines indicate the target boundaries and the regions in between the dashed lines represent the targets. Each target's slope is drawn as a solid line within the corresponding target region. The small vertical solid lines indicate the annotated syllable boundaries. The optimal boundaries are shifted with respect to the initial boundaries by up to 140 ms. The last boundary is always shifted to the last pitch point, since the region behind the last point has no meaning to the fit.

position of the target boundaries, which are jointly optimized together with the other parameters. New features include enhanced I/O routines that make the software much more flexible and robust against undefined behaviour induced by wrong file types etc., as well as a new GUI based on the WXWIDGETS C++ library and extended command line functionality.

2 Methods

2.1 Estimation of target boundaries

Within the TO2 framework, the process of target optimisation takes place in a similar way as described in [4]. However, the parameter set (m, b, τ), consisting of the slope, offset and time constant of each articulatory target, respectively, is extended by a fourth parameter δ_B (referred to as *boundary delta*). In contrast to the other parameters, which are optimized as absolute values, the boundary delta is a relative value between -1 and 1. At each step of the BOBYQA [5] optimization procedure, each target's onset is changed relative to the initial position of the corresponding boundary. Considering a number of N articulatory targets, the initial target onset positions are given by T_n (in ms), where $(n = 1, \ldots, N)$. The positions of the shifted onsets T_n^{shift} are calculated the following way:

$$T_n^{\text{shift}} = \begin{cases} T_n + \delta_B^n \cdot 100\,\text{ms}, & \text{if } \delta_B < 0 \text{ and } n = 1, \\ T_n + \delta_B^n \cdot |T_n - T_{n-1}|, & \text{if } \delta_B < 0 \text{ and } N \geq n > 1, \\ T_n + \delta_B^n \cdot |T_{n+1} - T_n|, & \text{if } \delta_B \geq 0 \text{ and } N \geq n > 1. \end{cases} \quad (1)$$

Evidently, the boundaries are shifted with respect to the duration of the preceding (succeeding) target if the boundary delta is negative (positive). Therefore, target boundaries can not shift beyond the initial onset boundaries of adjacent targets. However, shifted targets might move beyond a preceding shifted target. If this happens, boundaries are reordered to be monotonically increasing in time, so that the individual targets are always well defined and never overlap. Since there is no preceding target in the case of $n = 1$, the maximum negative time shift of the first boundary was defined as $-100\,\text{ms}$. Even though there should be no trajectory data before the first initial boundary, a shift of the first boundary towards an earlier time instant can be beneficial due to the initialization and the impulse response of the TAM filter. On the other hand, if the time shift of the first boundary is positive, it might move past the first data point. To prevent this, the first boundary is set to the time of the first data point in such a case. The offset of the last target N (boundary number $N+1$) does not get optimized. It is always set equal to the time point of the last data point, since the region beyond the last data point has no influence on the fit.

Boundary initialization

The maximum possible boundary shift $[-\delta_{B,\max}, +\delta_{B,\max}]$ is controlled by the user. If $\delta_{B,\max} < 1$, the boundary optimization is limited to a respectively smaller region around the initial boundaries and setting it to zero restores the TO1 behaviour (no boundary optimization). If boundary optimization is activated and $\delta_{B,\max} = 1$, the optimal position of all target boundaries can in principle be estimated from any given initial configuration of boundaries. Therefore, an initialization by prior syllable annotation is no longer a necessary requirement as it was in TO1. Instead, the TO2 provides the possibility to initialize the boundaries at evenly spaced positions between the first and last data point.

Early stopping

As described in [4], the BOBYQA algorithm does not necessarily find the global minimum of the objective function in a single run. Consequently, the optimization procedure must be executed multiple times, each time initialized with a different set of parameters drawn at random from the defined search space. In the following, such multiple runs are referred to as *random iterations*. The random iterations are independent from each other and the run that minimizes the objective function is selected as the optimization result. In contrast to the TO1, the number of random iterations is a tunable parameter in TO2 that can be adjusted by the user. This is important because the boundary optimization added another dimension to the search space and the computational complexity of the problem increased. Therefore, runs with boundary optimization require a lot more random iterations to converge to the optimum, compared to runs without boundary optimization. On the other hand, additional random iterations become unnecessary once the optimum is found. In order to reduce the computational cost, a method referred to as *early stopping* has been developed to terminate the search prematurely as soon as a certain convergence criterion is met. The procedure of early stopping is shown in Algorithm 1. Here, f_{tmp} denotes the temporary minimum of the TO objective function of each respective random iteration and f_{min} denotes the global optimum across the multiple iterations. If early stopping is used, a temporary minimum f_{tmp} must be smaller than the current minimum minus a percentage of the current minimum (determined by the hyper parameter ε that can be set by the user) in order to be counted as the new (temporary) global minimum f_{min}. If the global minimum is updated, a *convergence* counter (that is increased by one after every iteration) is reset to zero. Once the convergence counter reaches a certain value, referred to as *patience*, the search is terminated. The patience is another hyper-parameter that can be adjusted by the user. However, in order to provide useful default values for the patience and ε, the convergence behaviour of the boundary optimization was studied and evaluated as described in Section 2.2 and 3.2.

Algorithm 1: Early stopping

```
$f_{\min} \leftarrow$ 1e6;
$convergence \leftarrow 0$;
$search\_finished \leftarrow$ false;
for $random\_iterations$ do
    if not $search\_finished$ then
        $f_{\text{tmp}} \leftarrow$ BOBYQA();
        if $f_{\text{tmp}} < f_{\min} \cdot (1 - \varepsilon)$ then
            $convergence \leftarrow 0$;
            $f_{\text{tmp}} \leftarrow f_{\min}$;
        end
        $convergence \leftarrow convergence + 1$;
        if $convergence \geq patience$ then
            $search\_finished \leftarrow$ true;
        end
    end
end
```

2.2 Hyper-parameter tuning and performance tests

Speech corpus

To evaluate the performance of the TO2 we optimized the TAM parameters to model the pitch contours of 1562 German words[2], that were uttered by a professional female speaker. The corpus is similar to the one used in [4], however, duplicate entries (multiple recordings of the same utterance) were excluded. The distribution of the number of syllables per utterance is shown as a histogram in Figure 2.

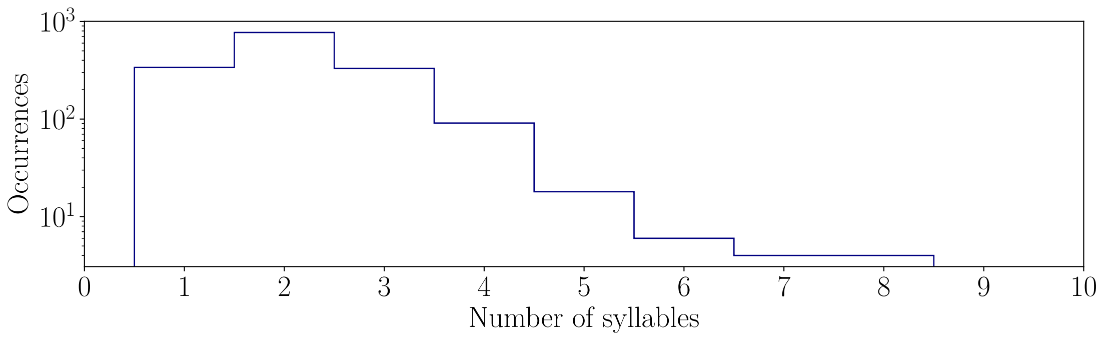

Figure 2 – The composition of the used speech data set as a function of the number of syllables.

The TO2 expects TextGrid and PitchTier files as input file formats for syllable annotations and pitch data, respectively. Therefore, the syllable boundaries of all words were manually annotated using the software PRAAT [6]. Subsequently, the annotations were exported as TextGrid files and the pitch contour of each word was extracted and saved as a PitchTier file. The pitch contours were checked individually for inaccuracies and manually corrected when necessary.

[2]WAV files available under: https://www.degruyter.com/view/product/19839

Hyper-parameters

The BOBYQA optimization algorithm has two important hyper-parameters: the maximum number of cost evaluations $N_f^{\max}$ (objective function evaluations per single random iteration), and the final trust region radius ρ_{E}. In TO2, both parameters can be adjusted by the user. Nevertheless, since these parameters can have a huge impact on the performance and the computational costs, good default values were estimated via a grid search. Since this tuning of hyper-parameters was time-consuming from a computational point of view, it was not performed on the whole speech corpus, but only on a small subset of 8 utterances. In this subset all syllable numbers were represented once. The parameter $N_f^{\max}$ was modified in ten equal steps from 10^5 to 10^6 and ρ_{E} was varied in four steps by four orders of magnitude from 10^{-6} to 10^{-3}. Each combination of utterances and parameters was evaluated, giving a total of 320 TO2 runs with boundary optimization. The number of random iterations per run was set to 1000 and early stopping was disabled.

Performance tests

In order to evaluate the performance gain from the boundary optimization, the TO2 performance was compared to the TO1 performance. Thereby, the target optimization process was run three times (without boundary optimization; with boundary optimization and uniform initialization; with boundary optimization and the syllable annotation-based initialization) for the whole speech corpus of 1562 words. The number of random iterations was set to 1000 and early stopping was disabled. The optimization hyper-parameters $N_f^{\max}$ and ρ_{E} were set to 10^5 and 10^{-3}, respectively. The search space and regularization space parameters were kept at their default values as defined in [4], except for the mean time constant τ, which was set to 20 ms, as this seems to be a more natural value. In addition to all the information about the optimization process, the computation time for each optimization was measured.

3 Results

3.1 Hyper-parameter optimization

From the results of the hyper-parameter optimization it could be concluded that neither the variation of $N_f^{\max}$ nor the variation of ρ_{E} had an impact on the found minimum of the objective function. This was observed for all the tested utterances. Since smaller values of ρ_{E}, had a negative impact on the computational performance and no observable effect on the optimization minimum, $\rho_{\mathrm{E}} = 10^{-3}$ was selected as the default value. The fact that the variation of $N_f^{\max}$, had no effect on the computational or optimization performance suggests that 10^5 random iterations are more than enough in order to obtain a stable minimum. Therefore, $N_f^{\max} = 10^5$ was chosen as the default value.

3.2 Performance tests

Comparison with TO1

In Figure 3 the TO2 performance in terms of $f_{\min}$, the minimum of the cost function optimized by the TO, the root-mean-square error (RMSE) between the fit and the measured pitch, and the respective linear correlation coefficient, is shown relative to the TO1 performance. It is observed that the boundary optimization yields a significant reduction of $f_{\min}$ and RMSE, as well as a significant increase of the correlation coefficient with respect to the performance without

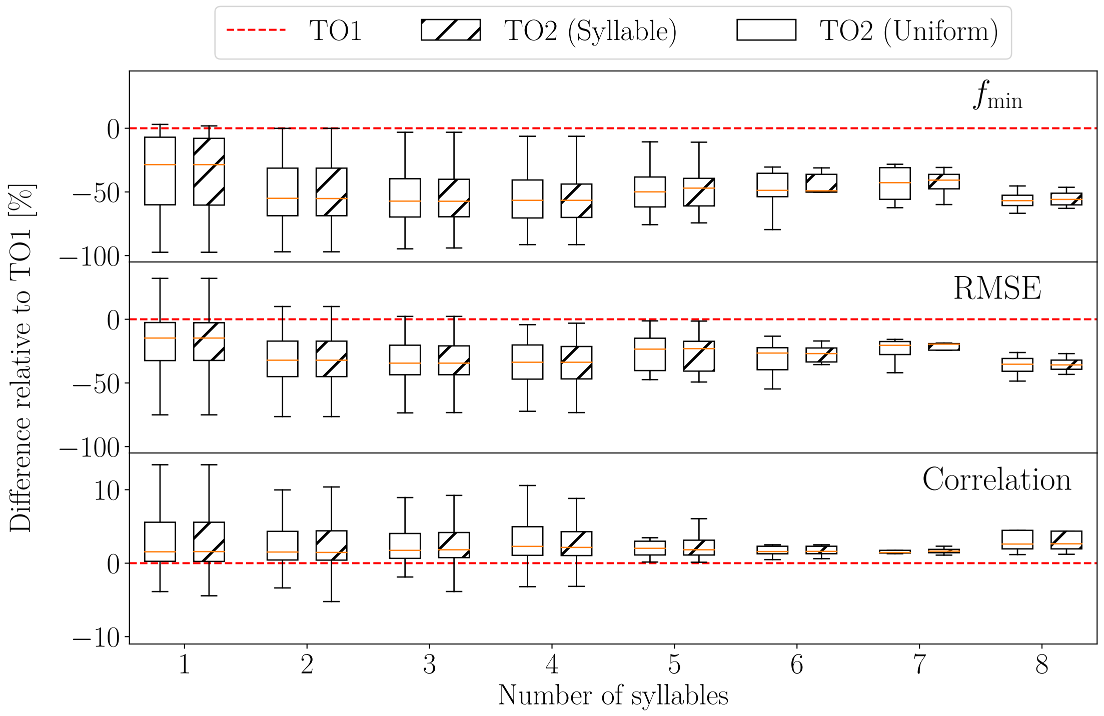

Figure 3 – The TO2 performance (top: the minimum of the cost function optimized by the TO, middle: the RMSE between the fit and the measured pitch, bottom: the respective linear correlation coefficient) is shown relative to the TO1 performance, as a box plot. The median of each distribution is indicated by the solid line within each box. Outliers are not shown.

boundary optimization. The minimum of the objective function is reduced for almost all tested utterances, except for some rare instances of one-syllable utterances. However, in such cases only the first boundary can be modified and only in a limited way. Therefore, the performance is strongly affected by the TAM-filter initialization. Nevertheless, even in case of one-syllable words, the $f_{\min}$ is greatly reduced for the large majority of utterances compared to the case without boundary optimization. For utterances with two or more syllables, the impact of the boundary optimization becomes even more significant. Averaged over the 1181 words with two or more syllables and both syllable (hatched boxes) and uniform initialization (solid boxes), the medians for $f_{\min}$, RMSE and the correlation are $-55.9^{+1.6}_{-1.2}\,\%$, $-32.9^{+1.3}_{-1.4}\,\%$ and $1.6^{+0.2}_{-0.1}\,\%$, respectively. The performance of the syllable-based and uniform initialization is very similar, suggesting that the same optimum is found for both types of initialization in most cases.

Computational performance and early stopping

Figure 4 shows the mean computation time needed for the optimization as a function of the number of syllables. The curves for TO1 and TO2 both show a non-linear, monotonic increase. However, in the case of TO2, the increase is much stronger due to the higher computational complexity caused by the boundary optimization. For words with eight syllables, the runs with boundary optimization take more than four times as long as the runs without boundary optimization. It is also observed that, on average, runs with uniform boundary initialization require systematically less computation time than runs with syllable-initialized boundaries.

In order to find meaningful default values for the early stopping parameters, the maximum patience was calculated for different values of ε. The maximum patience is defined as the longest interval of random iterations (which do not contain the global minimum of the optimization at the same time) during which the f_{tmp} does not fall below the current $f_{\min}$ within a

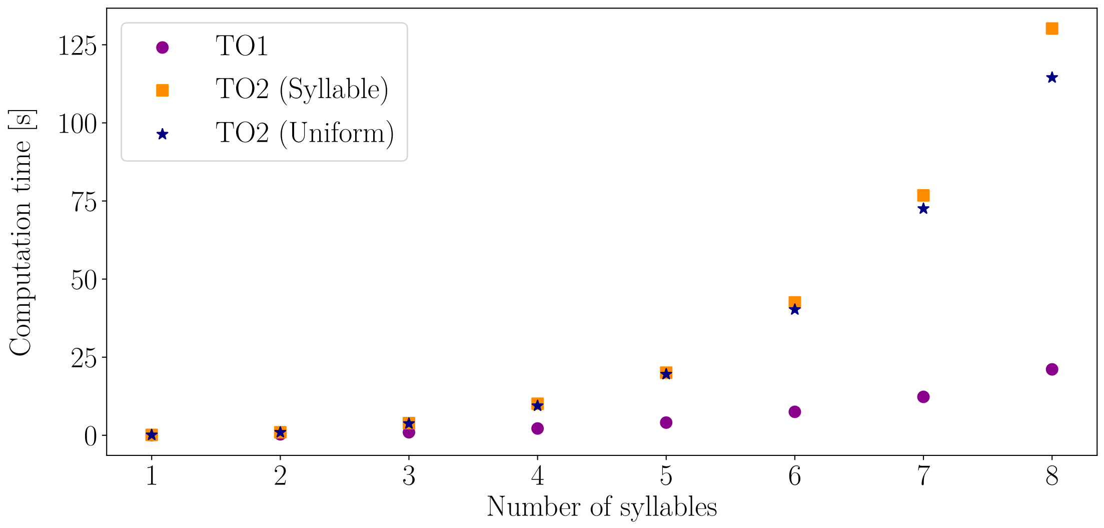

Figure 4 – The computation time is shown as a function of the number of syllables. The performance without boundary optimization is visualized as circles. Results from runs with boundary optimization are visualized as squares (syllable initialization) and stars (uniform initialization), respectively.

certain margin determined by ε. In Figure 5, the maximum patience is shown as a function of the number of syllables for three different values of ε. The results show that if a difference of 10 % with respect to the global minimum is accepted, useful default values for the patience are given by the relation $15 \cdot x$, whereby x is the number of (syllable) targets. For this approximation, the results for more than 6 syllables were disregarded because they were too infrequent to calculate reliable statistics.

4 Discussion

We conclude that the optimization of target boundaries improves TAM-based fits significantly. Furthermore, we demonstrated that the performance of syllable and uniformly initialized optimization runs is very similar. The data even suggest that the uniform initialization slightly outperformed the syllable initialization on average, yielding a lower computation time, lower patience and slightly better performance when it comes to f_{min}, RMSE and correlation. Nevertheless, much more examples of words with more than four syllables would be needed in order to investigate whether there is really a significant difference in performance or not.

The additional TAM dimension has interesting implications. Targets can no longer be interpreted as syllables but have become an abstract quantity of the model. In future work, it should be examined whether and according to which regularities the optimal boundaries differ from syllable boundaries. Various use-cases of the boundary optimization are conceivable, for example the determination of the number of targets, the study of target onset-synchronization and the study of articulatory trajectories that can also in theory be handled by the optimization procedure implemented in TO2. Future minor updates will therefore add the option to import arbitrary point contours as the target contour instead of just PitchTier files. Finally, in future updates of the software, the huge increase of the computational expense from the boundary optimization should be addressed by implementing a more efficient search procedure.

Acknowledgments

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

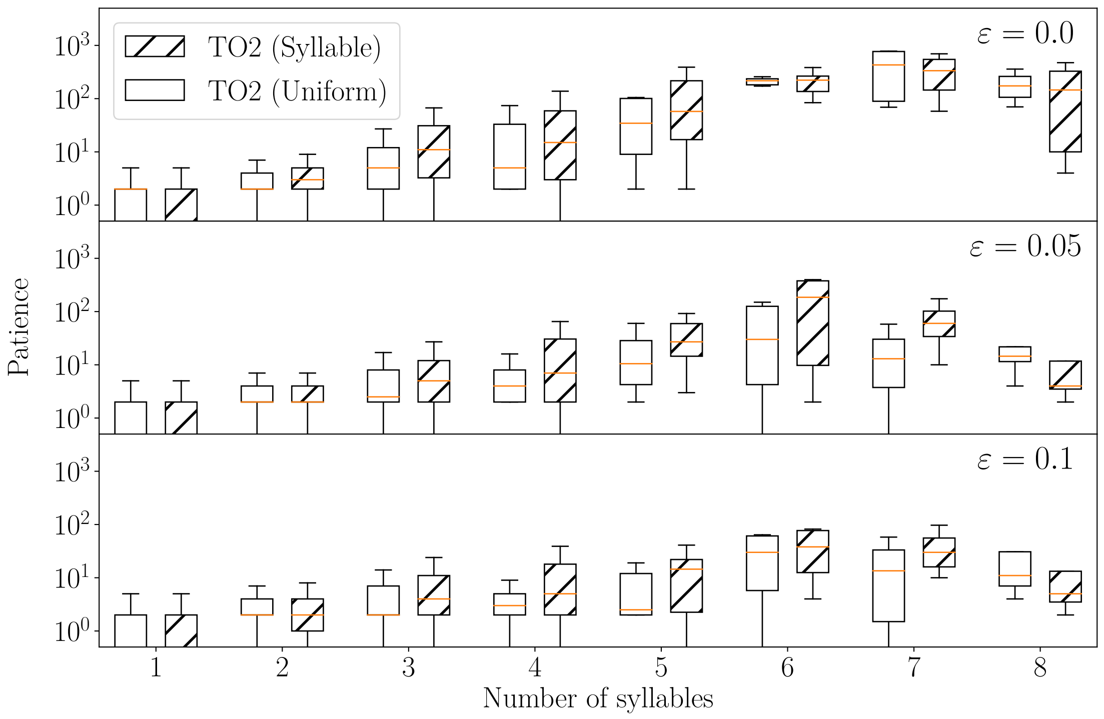

Figure 5 – The maximum patience is shown as a function of the number of syllables for TO2 optimizations with syllable initialization (hatched boxes) and uniform initialization (solid boxes). Top: $\varepsilon = 0.0$. Middle: $\varepsilon = 0.05$. Bottom: $\varepsilon = 0.1$.

Centre for Information Services and High Performance Computing TU Dresden for providing its facilities for high throughput calculations.

References

[1] XU, Y. and Q. E. WANG: *Pitch targets and their realization: Evidence from Mandarin Chinese*. *Speech Commun.*, 33(4), pp. 319–337, 2001.

[2] PROM-ON, S., Y. XU, and B. THIPAKORN: *Modeling tone and intonation in Mandarin and English as a process of target approximation*. *J. Acoust. Soc. Am.*, 125(1), pp. 405–424, 2009.

[3] BIRKHOLZ, P.: *Modeling consonant-vowel coarticulation for articulatory speech synthesis*. *PloS ONE*, 8(4), p. e60603, 2013.

[4] BIRKHOLZ, P., P. SCHMAGER, and Y. XU: *Estimation of pitch targets from speech signals by joint regularized optimization*. In *Proc. of the 26th European Signal Processing Conference (EUSIPCO)*, pp. 2075–2079. 2018.

[5] POWELL, M. J.: *The BOBYQA algorithm for bound constrained optimization without derivatives*. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pp. 26–46, 2009.

[6] BOERSMA, P. and D. WEENICK: *Praat: Doing phonetics by computer (version 4.3.14) [computer program]*. 2005. URL `http://www.praat.org`. (Last viewed January 20, 2021).