A NOVEL SELF-ORGANISING SPEECH PRODUCTION SYSTEM USING PSEUDO-ARTICULATORS

C. S. Blackburn and S. J. Young
Cambridge University Engineering Department (CUED), England
email: csb@eng.cam.ac.uk

ABSTRACT

A novel articulatory speech production system which is stochastically trained from a pre-specified initialisation state is presented. The target positions for a set of pseudo-articulators and the mapping from these to output speech spectral vectors are jointly optimised using linearised Kalman filtering and an assembly of neural networks. The techniques used to initialise and train the system are described, and preliminary results when synthesising speech are demonstrated.

INTRODUCTION

Articulatory speech synthesisers model human speech dynamics and hence theoretically can produce very high quality speech waveforms with explicit time-domain modelling of co-articulation [8, 12, 15]. Two major problems confronting such systems are:

- Specification of the sequence of articulator positions or vocal tract area functions corresponding to a given text.
- Provision of an accurate model of the human vocal tract.

The former is frequently achieved using an "inverse" model to map parametrised speech, usually in the form of spectral vectors, into articulator positions or vocal tract areas and hence determine target positions for the phonemes to be synthesised. We use a Kelly-Lochbaum synthesiser [6, 12] to generate a codebook of (articulator vector, spectral vector) pairs [13] which is inverted using dynamic programming (DP) incorporating geometrical constraints on the articulator trajectories, as shown in figure 1.

The inverse mapping is non-unique, so dissimilar articulator positions may result in similar acoustic outputs [2, 7], hence attempts to model the inverse transformation using acoustic error alone [1, 10] are likely to produce discontinuous articulatory output. A continuity constraint should therefore be applied to such trajectories, which may be implicit as in inverse filtering techniques [16], or explicitly imposed via a restriction to critically damped second order transitions [14] or the minimisation of geometrical distances [13, 17].

In addition, the non-linearity of the inverse mapping combined with its non-uniqueness can result in non-convex target regions in articulator space [4], so gradient-based algorithms which average over a number of training vectors, whether a single neural network [1, 10, 17], Jacobian computation [5] or unconstrained optimisation [7], may converge to an average which does not lie within the target class, resulting in an incorrect inverse model. This problem can be avoided either by subdividing the input space into regions in which the non-linear mapping is unique [11], or by jointly optimising an (inverse, forward) model pair to restrict the inverse model to a particular solution [3].

In our system the use of codebook look-up guarantees that a particular inverse solution is chosen at each point in time, and the DP search incorporates both acoustic and geometric constraints to ensure continuity.

The second problem, that of determining an accurate vocal tract model, is approached in our system by relaxing the constraint that the system exactly mimic human physiology. Instead, we use "pseudo-articulators" which fulfil roles similar to those of human articulators but whose positions are stochastically estimated from the training data. The initial articulator trajectory estimates obtained from the DP algorithm are iteratively re-estimated using linearised Kalman filtering and an assembly of neural networks which map from articulator positions to output speech.

SYSTEM INITIALISATION

System initialisation is shown in figure 1. Vocal tract area functions are determined from a set of five pseudo-articulators as in [9]. Four of these, roughly specifying tongue position, are sampled at regular intervals to give 6321 basic vocal tract shapes. A logarithmic quantisation is then applied to eliminate very similar shapes; since our aim in initialisation is to determine a set of articulator trajectories, time domain quantisation

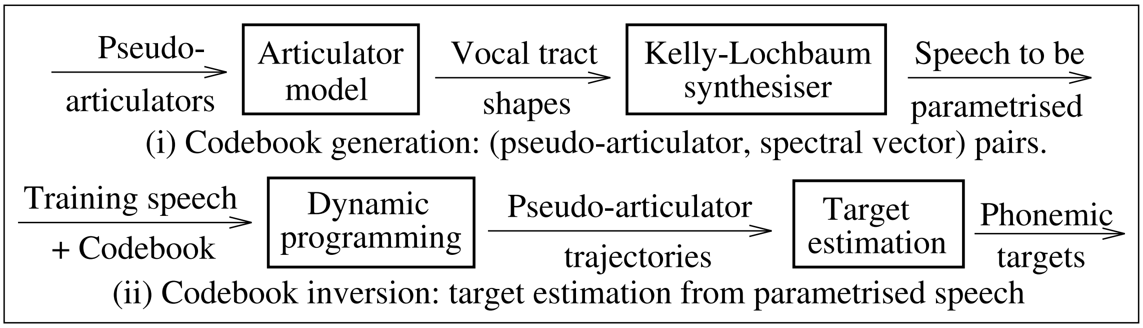

Figure 1: System initialisation.

is preferable to that in the frequency domain as used elsewhere [13].

Quantised lip opening is then added as a fifth parameter giving 27651 pseudo-articulator vectors which are used to generate a corresponding set of 10-section vocal tract area functions. These are interpolated in the logarithmic domain and re-sampled to yield an appropriate number of area sections for use in the Kelly-Lochbaum synthesiser, which treats the vocal tract as a variable number of fixed cross-sectional area tubes and incorporates separate oral and nasal tracts, as well as modelling transmission loss. A sampling frequency of 16kHz corresponding to area sections of length $\approx$1.1cm was chosen, and both 15 and 16-section resampled area functions were used, giving a total of 55302 basic shapes.

Fricative waveforms are created from shapes with a constriction of less than 0.3cm^2 using a random noise source at the constricted point which is correlated with the voiced excitation, if any. Nasals are generated from the parallel combination of a variable oral tract and a fixed nasal tract, for three values of velum opening. In all, 31848 voiced and unvoiced fricatives and 15126 nasals were included, in addition to 55302 purely voiced waveforms. In each case the speech waveforms were parametrised by the CUED HTK recogniser to give one 12-dimensional liftered cepstral vector per 10msec of speech. Finally, 212 cepstral vectors representing "silence" or background noise in the training speech were added to give a total codebook size of 102488 vectors.

Codebook inversion

The training speech comprised 600 sentences of one adult male from the speaker-dependent training portion of the Defence Advanced Research Projects Agency (DARPA) Resource Management corpus. This speech was also coded into 12-dimensional cepstral vectors, and dynamic programming was used to find a path through the lattice of possible articulator trajectories.

At each step of the DP algorithm, both the acoustical mismatch between the parametrised training speech vector and the codebook acoustic vectors *and* the geometrical mismatch between successive articulatory vectors are combined into a weighted score when evaluating paths. To reduce the computational load, a sub-optimal search was used in which only the 500 codebook vectors with the best acoustic match were considered at each step.

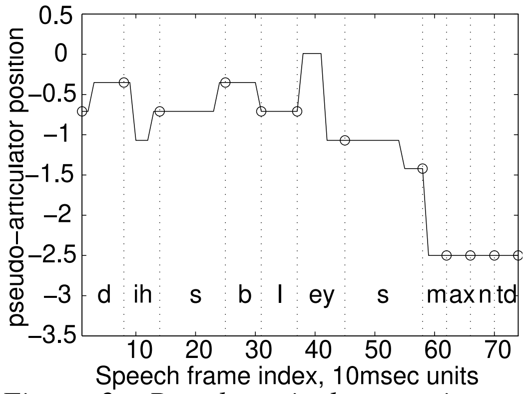

Figure 2: Pseudo-articulator trajectory for "displacement".

Pseudo-articulator trajectories such as that in figure 2 were generated in this way for all 600 sentences. This figure shows the trajectory of one pseudo-articulator during the word "displacement", where phoneme boundaries taken from the HTK-produced label file are marked as vertical lines. The phone labels are taken from the DARPA transcription of the speech, and in most cases the pseudo-articulator is steady between boundaries.

Target statistics are thus determined from the values of the articulators at the midpoint of each occurrence of each phoneme to give initial target means and covariance matrices for each of the five basic articulators for the 47 phonemes in the data set.

TRAINING

A separate neural network is used to learn the mapping from the pseudo-articulator trajectories of each phoneme to output speech. The trajectories are piecewise linear interpolations of the phoneme target means, constrained to pass through the average of two adjacent target means at the phonemic boundary. The training set output vectors were 24-dimensional mel-scaled log spectral coefficients; while this is a less efficient representation than the cepstral coefficients used previously, their use results in a more easily learned non-linear function.

The purpose of the neural networks is to approximate this mapping from articulatory to acoustic space, so that the linearised Jacobian matrix can be used to re-estimate the phonemic targets; hence their performance and architecture are not crucial to the training process. We trained feed-forward multi-layer perceptrons with 12 inputs, 30 hidden units, 24 outputs and sigmoid non-linearities at the hidden units using resilient back-propagation (rprop) for 1000 batch update epochs, giving mean errors in estimated spectral coefficients of around 10%.

The global error covariance matrix for each network mapping is estimated from its performance on an unseen test set, and the Jacobian matrix is found by extending the usual error back-propagation formulae to evaluate the derivative of each output with respect to each input:

$$\frac{\partial y_k}{\partial y_i} = \sum_j (w_{ij} w_{jk} y_j (1 - y_j))$$

where y_i, y_j, y_k are the outputs of nodes in the input, hidden and output layers respectively and w_{ij}, w_{jk} are the input-hidden and hidden-output weights respectively.

If the initial estimate of a phoneme's articulatory target mean vector is denoted $\hat{\mathrm{x}}$, with associated initial covariance matrix $\hat{P}$, and if the neural mapping is denoted $h()$ with Jacobian matrix H at the target estimate, output z and output error covariance matrix R, the target vector can be re-estimated using linearised Kalman filtering as:

$$\mathrm{x} = \hat{\mathrm{x}} + \hat{P} H^T (H \hat{P} H^T + R)^{-1} (\mathrm{z} - h(\hat{\mathrm{x}}))$$

This gives a re-estimated target vector for each occurrence of each phoneme, from which new target mean and covariance statistics are computed. Updated pseudo-articulator trajectories are then derived and the networks re-trained. This process is iterated until the optimum set of phoneme targets is obtained, from which speech is synthesised.

RESULTS

Figure 3 shows original and synthetic smoothed 24-dimensional mel-scaled filter bank vectors for the phrase "clear windows". The phoneme alignment produced by HTK has resulted in small timing errors at phoneme boundary positions, however the gross spectral characteristics of the two plots correlate well.

Formant transitions are generally well defined, although the co-articulation from the stop /d/ to the following vowel /ow/ in "windows" has been missed by the synthesiser. The use of a separate neural network for each phoneme results in some discontinuities at phoneme boundaries, for example immediately preceding the final fricated /z/ in "windows", however the formants themselves are well-defined across boundaries, and high-frequency frication is successfully modelled.

Future work

The system is still under development, and many features have yet to be implemented. In particular, improved co-articulation modelling could be provided via the explicit modification of the target means according to their context. Since we have statistics for target means *and* variances for each phoneme, this should permit statistically-based co-articulation effects to be modelled.

In addition, the use of pseudo-articulators which are not constrained to human physiology provides the possibility of adding additional articulators during the training phase, thus potentially increasing the amount of information available to the neural mappings.

Finally, a method for smoothly combining the outputs of the neural networks across phoneme boundaries should reduce errors due to discontinuities.

CONCLUSIONS

This paper has presented a novel pseudo-articulatory speech production model, which is initialised by generating a codebook of (acoustic vector, spectral vector) pairs using a conventional Kelly-Lochbaum articulatory synthesiser which is inverted using sub-optimal dynamic programming search combining acoustic and geometric cost functions. The means and covariance matrices of

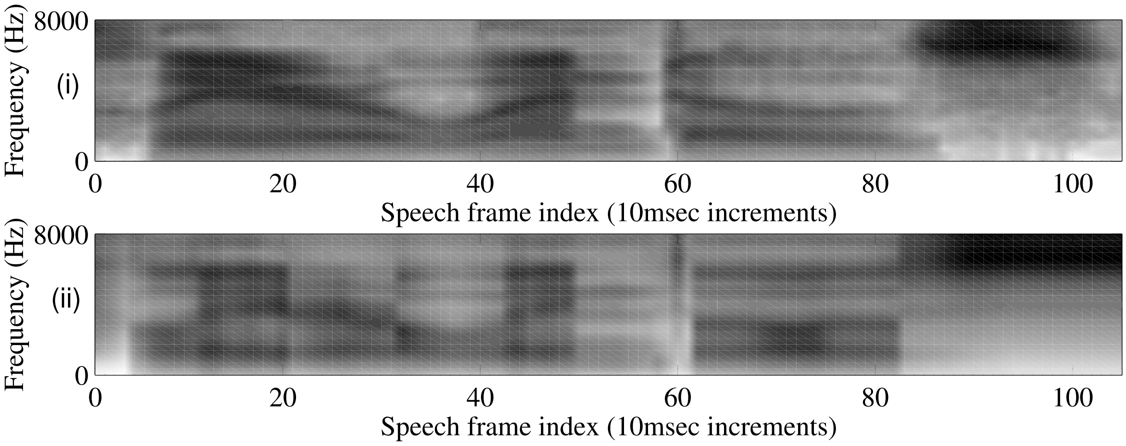

Figure 3: (i) original and (ii) synthesised filter bank output for phrase "clear windows".

the articulator targets for each phoneme are then estimated over 600 sentences of one speaker, and articulator trajectories corresponding to the training speech are constructed using constrained piecewise linear interpolation between the means.

An individual neural network is then trained to learn the mapping from articulators to parametrised speech vectors for each of 47 phonemes, and the target means are re-estimated using these mappings and a linearised Kalman filter. This process is iterated to find the optimum set of target means from which output speech is synthesised.

While articulatory synthesisers still do not produce speech comparable to that of the best rule-based synthesisers, we have attempted to show that the inability to exactly model the human speech production mechanism need not limit their viability, and have demonstrated a preliminary stochastically-trained system which yields promising results.

REFERENCES

[1] B. S. Atal. "Neural networks for estimating articulatory positions from speech". *J. Acoust. Soc. Am.*, 86:S67, 1989.

[2] B. S. Atal et al. "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique". *J. Acoust. Soc. Am.*, 63(5):1535–1555, May 1978.

[3] M. George, P. Jospa, and A. Soquet. "Articulatory trajectories generated by the control of the vocal tract by a neural network". In *Proc. Int. Conf. Sp. Lang. Proc.*, volume 2, pages 583–586, 1994.

[4] M. I. Jordan and D. E. Rumelhart. "Forward models: Supervised learning with a distal teacher". *Cog. Sc.*, 16:307–354, 1992.

[5] P. Jospa and A. Soquet. "The acoustic-articulatory mapping and the variational method". In *Proc. Int. Conf. Sp. Lang. Proc.*, volume 2, pages 595–598, 1994.

[6] J. L. Kelly Jr. and C. Lochbaum. "Speech synthesis". In *Sp. Comm. Sem.*, Stockholm, 1962.

[7] S. E. Levinson and C. E. Schmidt. "Adaptive computation of articulatory parameters from the speech signal". *J. Acoust. Soc. Am.*, 74(4):1145–1154, 1983.

[8] P. Mermelstein. "Articulatory model for the study of speech production". *J. Acoust. Soc. Am.*, 53(4):1070–1082, 1973.

[9] P. Meyer, R. Wilhelms, and H. W. Strube. "A quasiarticulatory speech synthesizer for German language running in real time". *J. Acoust. Soc. Am.*, 86(2):523–539, 1989.

[10] G. Papcun et al. "Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data". *J. Acoust. Soc. Am.*, 92(2 Part 1):688–700, Aug. 1992.

[11] M. G. Rahim et al. "On the use of neural networks in articulatory speech synthesis". *J. Acoust. Soc. Am.*, 93(2):1109–1121, Feb. 1993.

[12] P. Rubin, T. Baer, and P. Mermelstein. "An articulatory synthesizer for perceptual research". *J. Acoust. Soc. Am.*, 70(2):321–328, Aug. 1981.

[13] J. Schroeter and M. M. Sondhi. "Techniques for Estimating Vocal-Tract Shapes from the Speech Signal". *IEEE Trans. Sp. Aud. Proc.*, 2(1):133–150, Jan. 1994.

[14] K. Shirai and T. Kobayashi. "Estimating articulatory motion from speech wave". *Sp. Comm.*, 5(2):159–170, June 1986.

[15] M. M. Sondhi and J. Schroeter. "A hybrid time-frequency domain articulatory speech synthesizer". *IEEE Trans. Acoust. Sp. Sig. Proc.*, ASSP-35(7):955–967, July 1987.

[16] H. Wakita. "Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms". *IEEE Trans. Aud. Electroacoust.*, AU-21(5):417–427, Oct. 1973.

[17] J. Zacks and T. R. Thomas. "A new neural network for articulatory speech recognition and its application to vowel identification". *Comp. Sp. Lang.*, 8:189–209, 1994.