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 synthesizers 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 thereby 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 [15] 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 articulator error alone [11, 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 of our axes 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 specified 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; this procedure in initialisation is to determine a set of articulator trajectories, time domain quantisation 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 ≈1.1cm was chosen, and both 15 and 16-section re-sampled 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 constriction 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 parameterised by the CUED HTK recogniser to give one 12-dimensional lifter cepstral vector per 10ms cepstral units. Finally, 212 cepstral vectors representing "silence" or background noise in the training speech were added to give a total codebook size of 102488 vectors.

Figure 1: System initialisation.

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

(a) Codebook generation: (pseudo-articulator, spectral vector) pairs.

(b) Codebook inversion: target estimation from parametrised speech.
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 linearized 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 derivatives of each output with respect to each input:

\[ \frac{\partial E}{\partial y_i} = \sum_j (w_j y_j u_j (1 - y_j)) \]

where \( y_i, y_j, y_k \) are the outputs of nodes in the input, hidden and output layers respectively and \( w_j, w_k \) are the input-hidden and hidden-output weights respectively.

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

\[ \dot{x} = \dot{P}^T (HPH^T + R)^{-1} (z - h(\tilde{z})) \]

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 window". The phoneme alignment produced by HTK has resulted in small timing errors at phoneme boundary positions, however the overall 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 synthesizer. The use of a separate neural network for each phoneme results in some discontinuities at phoneme boundaries, for example immediately preceding the final fricative /s/ 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 achieved 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 output of the two 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 (auditory) vocal tract vectors 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