Autoregressive Models for Statistical Parametric Speech Synthesis

Matt Shannon, *Student Member, IEEE,* Heiga Zen, *Member, IEEE,* and William Byrne, *Senior Member, IEEE*

***Abstract*—We propose using the autoregressive hidden Markov model (HMM) for speech synthesis. The autoregressive HMM uses the same model for parameter estimation and synthesis in a consistent way, in contrast to the standard approach to statistical parametric speech synthesis. It supports easy and efficient parameter estimation using expectation maximization, in contrast to the trajectory HMM. At the same time its similarities to the standard approach allow use of established high quality synthesis algorithms such as speech parameter generation considering global variance. The autoregressive HMM also supports a speech parameter generation algorithm not available for the standard approach or the trajectory HMM and which has particular advantages in the domain of real-time, low latency synthesis. We show how to do efficient parameter estimation and synthesis with the autoregressive HMM and look at some of the similarities and differences between the standard approach, the trajectory HMM and the autoregressive HMM. We compare the three approaches in subjective and objective evaluations. We also systematically investigate which choices of parameters such as autoregressive order and number of states are optimal for the autoregressive HMM.**

***Index Terms*—Acoustic modeling, autoregressive hidden Markov model, autoregressive processes, hidden Markov models (HMMs), speech, statistical parametric speech synthesis.**

I. Introduction

IT has been shown that it is possible to synthesize natural sounding speech with hidden Markov models (HMMs) and the quality of the best HMM-based statistical parametric speech synthesis systems now rivals the best unit selection synthesis systems [1]. A breakthrough that helped make this possible was realizing how to use dynamic feature information during synthesis, by respecting the constraints between static and dynamic features [2].

However the established approach to HMM-based speech synthesis is inconsistent in the enforcement of these constraints [3]. During synthesis we take the constraints between static and dynamic features into account, whereas during parameter estimation we assume the static and dynamic feature sequences are independent.

This is a recognized problem and has been addressed previously. Zen *et al.* showed how a *trajectory HMM* could be employed so that the same model is used for parameter estimation and synthesis [3]. Synthesis quality improved as a result. However parameter estimation for the trajectory HMM is more complicated than for the standard HMM, requiring gradient-based parameter estimation, and exact expectation maximization training is intractable. The challenge remains to find a model which can easily and consistently be used for both parameter estimation and synthesis.

In this paper we propose using the *autoregressive HMM* [4]–[7] for speech synthesis. The autoregressive HMM relaxes the traditional HMM conditional independence assumption, allowing state output distributions which depend on past output as well as the current state. In this way the autoregressive HMM explicitly models some of the dynamics of speech and introduces the continuity and context dependence needed for good quality synthesis. This approach is also flexible, providing a simple way to turn a sequence modeling problem into a finite-dimensional regression problem.

Autoregressive HMMs have been used before for speech recognition [4]–[6], [8], but have not been extensively investigated for speech synthesis.[1] A basic formulation of the autoregressive HMM for statistical parametric speech synthesis showing how to do expectation maximization-based parameter estimation and parameter generation considering global variance was given in [11]. Details of how to do decision tree clustering for the autoregressive HMM were given in [12]. As mentioned above the autoregressive HMM and trajectory HMM both remedy an inconsistency present in the standard approach, and the effect of this inconsistency was investigated in [13]. Quillen investigated the problem of stability of autoregressive coefficients for speech synthesis using the autoregressive HMM, and in addition found that using alignments derived from a speech recognition system to estimate autoregressive HMM parameters led to an improvement in an objective metric compared with embedded re-estimation from a flat start [14].

The present paper builds on previous work by directly comparing the standard approach to the autoregressive HMM with autoregressive decision tree clustering, comparing the autoregressive HMM to the trajectory HMM in subjective evaluations, extensively investigating how to set parameters such as autoregressive order and number of states for the autoregressive

Manuscript received June 15, 2012; revised September 06, 2012; accepted October 22, 2012. Date of publication November 15, 2012. This work was supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 213845 (EMIME) and in part by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chung-Hsien Wu.

M. Shannon and W. Byrne are with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: sms46@eng.cam.ac.uk).

H. Zen was with Cambridge Research Laboratory, Toshiba Research Europe, Cambridge CB4 0GZ, U.K. He is now with Google, London SW1W 9TQ, U.K.

[1]For the autoregressive HMM considered here the observations are acoustic feature vectors. This is distinct from the *hidden filter HMM* [9], [10] for which the observations are waveform samples.

HMM, and presenting theoretical comparisons between the autoregressive HMM and the standard approach and trajectory HMM.

The algorithms described in this paper for parameter estimation and synthesis using the autoregressive HMM have been implemented in an open source extension [15] to the *HMM-based speech synthesis system (HTS)* [16].

The remainder of this paper is laid out as follows. In Section II we review the standard approach to statistical parametric speech synthesis. In Section III we specify the autoregressive HMM model and show how to do efficient parameter estimation, decision tree clustering and synthesis. In Section IV we look at some of the similarities and differences between the standard HMM synthesis framework, the trajectory HMM and the autoregressive HMM. In Section V we report results of experiments comparing the autoregressive HMM with the two other models, and investigating the appropriate choice of model structure parameters for the autoregressive HMM. Finally in Section VI we discuss our experimental results and give conclusions.

II. Background

A. Statistical Parametric Speech Synthesis

Speech synthesis aims to synthesize speech from text. In a typical statistical parametric speech synthesis system the text is represented as a sequence of *labels* $l = l_{1:J}$ of length J and the speech audio is represented as a sequence of *acoustic feature vectors* (or *speech parameters*) $c = c_{1:T}$ of length T encoding spectral, fundamental frequency and aperiodicity information [1]. The parameters (ν, λ) of a parametric statistical model $P(c \,|\, l, \nu, \lambda)$ are learned from data, and this model is used to synthesize audio for previously unseen text. This approach is referred to as *parametric* since the feature vectors are used as speech parameters for a vocoder which converts between the feature vector sequence and audio.

The generative model $P(c \,|\, l, \nu, \lambda)$ is broken down into two separate components: a *state transition model* $P(\theta \,|\, l, \nu)$ which probabilistically generates a *hidden state sequence* $\theta = \theta_{1:T}$ given a label sequence, and an *acoustic model* $P(c \,|\, \theta, \lambda)$ which probabilistically generates a feature vector sequence given this state sequence. Typically the hidden state consists of the current label, the index of the current label within the label sequence, the current *sub-label* (or *state*), and the number of frames remaining[2] in the current sub-label.

The state transition model has Markovian form $P(\theta \,|\, l, \nu) = \prod_t P(\theta_t \,|\, \theta_{t-1}, l, \nu)$. This paper concerns the form of the acoustic model $P(c \,|\, \theta, \lambda)$.

B. The Standard HMM Synthesis Framework

A simple form of acoustic model is $P(c \,|\, \theta, \lambda) = \prod_t P(c_t \,|\, \theta_t, \lambda)$, which assumes the feature vectors (c_t) are conditionally independent given the state sequence. Together the Markovian state transition model and this simple acoustic model form a *hidden Markov model (HMM)* $P(c, \theta \,|\, l, \nu, \lambda)$. The corresponding graphical model is shown in Fig. 1.

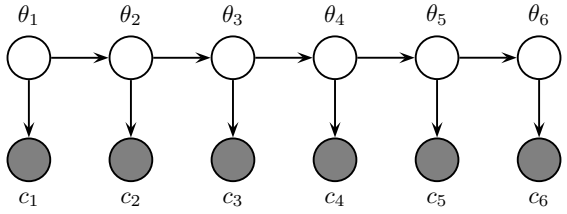


Fig. 1. Graphical model for a conventional HMM. Here $\theta = \theta_{1:6}$ is the state sequence and $c = c_{1:6}$ is the feature vector sequence. The dependence on the label sequence l and parameters (ν, λ) is not shown. Note that this is *not* the model used during training in the standard HMM synthesis framework, which augments the static feature vector sequence with dynamic features.

This simple acoustic model may be extended to take into account correlations between frames. This is conventionally done by augmenting the (static) feature vector sequence c with *dynamic features* to obtain an *observation vector sequence* $o = o_{1:T}$ and then using an HMM to model o. During synthesis we consider only those observation sequences o that arise from augmenting some static feature vector sequence c. This procedure allows the information encoded in the dynamic feature parameters to be used during speech parameter generation. We refer to the combination of modeling o with an HMM during training but restricting to c during synthesis as the *standard HMM synthesis framework*.

This approach is inconsistent since the model used during training is different to the model used during synthesis. Alternatively the model used during training can be viewed as a model defined over static features c only, in which case it correctly enforces the constraints between static and dynamic features but is *unnormalized*, i.e. the probability of the set of all sequences of static features is not one [3].

III. Autoregressive HMM

The autoregressive HMM [4]–[7], [11] uses an acoustic model of the form

$$P(c \,|\, \theta, \lambda) = \prod_t P(c_t \,|\, c_{t-K:t-1}, \theta_t, \lambda) \tag{1}$$

where $K \in \mathbb{N}$ is referred to as the *order* or *depth* of the model. Together the Markovian state transition model and this acoustic model form an autoregressive HMM $P(c, \theta \,|\, l, \nu, \lambda)$. A graphical model for the case $K = 2$ is shown in Fig. 2. Note that the conventional (static feature vector-only) HMM as shown in Fig. 1 is an autoregressive HMM with depth 0. The autoregressive HMM models the correlations between frames by explicitly encoding the dependence of the feature vector at time t on the feature vectors in the recent past. This approach provides a simple way to turn a sequence modeling problem into a finite-dimensional regression problem. It is flexible since in principle almost any regression model can be used for $P(c_t \,|\, c_{t-K:t-1}, \theta_t, \lambda)$.

In this investigation we use a very simple form of regression model where the dependence on $c_{t-K:t-1}$ is linear-Gaussian

[2]When using explicit duration models [17], [18] to model the duration of each sub-label, the hidden state is augmented with the number of frames remaining in the current sub-label and the transition structure is updated accordingly. This converts the state transition model from semi-Markovian to Markovian and allows efficient inference [19], [20].

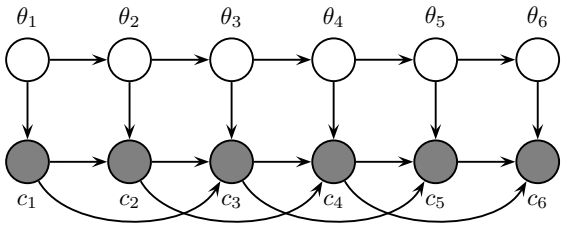


Fig. 2. Graphical model for an autoregressive HMM of depth 2. Here $\theta = \theta_{1:6}$ is the state sequence and $c = c_{1:6}$ is the feature vector sequence. The dependence on the label sequence l and parameters (ν, λ) is not shown.

and the dependence on θ_t is given by a decision tree:

$$P(c_t \,|\, c_{t-K:t-1}, \theta_t, \lambda) = \prod_i P(c_t^i \,|\, c_{t-K:t-1}^i, \theta_t, \lambda) \tag{2}$$

$$P(c_t^i \,|\, c_{t-K:t-1}^i, \theta_t, \lambda) = \mathcal{N}\left(c_t^i; m_q^i(c_{t-K:t-1}^i), (\sigma^2)_q^i\right) \tag{3}$$

$$m_q^i(v) \triangleq \sum_{d=1}^{D} a_q^{id} f^{id}(v) \tag{4}$$

where i indexes components of the feature vector, q is the leaf associated with state θ_t, and each $f^{id} : \mathbb{R}^K \to \mathbb{R}$ is a *basis function* that computes a real-valued summary of the recent past output $c_{t-K:t-1}^i$. The parameters λ of the autoregressive HMM are the *autoregressive coefficients* $(a_q^{id} : q, i, d)$ and the variance parameters $((\sigma^2)_q^i : q, i)$. We use $D \triangleq K + 1$ basis functions of the form $f^{id}(v) \triangleq v^d$ (the dth component of v) for $1 \le d < D$ and $f^{iD}(v) \triangleq 1$, so the mean $m_q^i(c_{t-K:t-1}^i)$ is a state-dependent linear combination of the recent past output plus a bias. The initial context $c_{-(K-1):0}$ is taken to be zero in this investigation. Specifying this initial context is necessary to define $P(c_t \,|\, c_{t-K:t-1}, \theta_t, \lambda)$ for $t \le K$.

We refer to the sequence of a component of the feature vector over time $c^i = c_{1:T}^i$ as a *trajectory*. We have taken the basis functions f^{id} to be functions of the recent past output in the same component i. This is consistent with the common assumption when modeling speech that the trajectories (c^i) for different components of the feature vector are independent given the state sequence θ. However it causes no problem if the basis functions depend on all recent past output $c_{t-K:t-1}$, or even on the present output up to the given component $c_t^{1:i-1}$.

A. Parameter Estimation

The autoregressive HMM permits efficient parameter estimation using expectation maximization [6], [11]. Here we summarize the re-estimation formulae used to compute updated parameter values given the *state occupancies* $\gamma_q(t)$ obtained using the Forward-Backward algorithm [21].

We define accumulators

$$\gamma_q \triangleq \sum_t \gamma_q(t) \tag{5a}$$

$$s_q^i \triangleq \sum_t \gamma_q(t) c_t^i c_t^i \tag{5b}$$

$$r_q^{id} \triangleq \sum_t \gamma_q(t) f^{id}(c_{t-K:t-1}^i) c_t^i \tag{5c}$$

$$R_q^{ide} \triangleq \sum_t \gamma_q(t) f^{id}(c_{t-K:t-1}^i) f^{ie}(c_{t-K:t-1}^i) \tag{5d}$$

where q ranges over decision tree leaves, i ranges over feature vector components and $1 \le d, e \le D$.

The re-estimation formulae giving the updated parameter values $(\widehat{a}_q^{id}, (\widehat{\sigma}^2)_q^i)$ are

$$\sum_{e=1}^{D} R_q^{ide} \widehat{a}_q^{ie} = r_q^{id} \tag{6}$$

$$(\widehat{\sigma}^2)_q^i = \frac{1}{\gamma_q}\left(s_q^i - \sum_{d=1}^{D} \widehat{a}_q^{id} r_q^{id}\right) \tag{7}$$

where q ranges over decision tree leaves, i ranges over feature vector components and $1 \le d \le D$. Note that computing the $(\widehat{a}_q^{id})$ using (6) involves storing and inverting a $D \times D$ matrix for each q and i. A typical depth 3 model has $D = 4$.

The value of the expectation maximization auxiliary function at its maximum $(\widehat{a}_q^{id}, (\widehat{\sigma}^2)_q^i)$ is

$$-\frac{1}{2}T\left(\log 2\pi + 1\right) - \frac{1}{2}\sum_q \gamma_q \sum_i \log(\widehat{\sigma}^2)_q^i. \tag{8}$$

B. Decision Tree Clustering

The standard approach to *decision tree clustering* [22] is modified for the autoregressive HMM [12]. As in the standard approach each leaf is recursively split using the question that maximizes the change in auxiliary function value, unless the maximum achievable change is less than a *clustering threshold* ξ in which case we do not split that leaf.

The accumulators (5) for an arbitrary leaf may be obtained by summing the corresponding state-level accumulators in the usual way. Thus we can use (7) to compute $(\widehat{\sigma}^2)_q^i$ and so compute the change in auxiliary function value (8) for a hypothesized split.

For the autoregressive HMM the updated parameter values $(\widehat{a}_q^{id}, (\widehat{\sigma}^2)_q^i)$ together with state occupancies (γ_q) are *not* in general sufficient to recover the accumulators (5) [12]. This means we must pass the decision tree clustering algorithm the accumulators themselves, and not just the re-estimated parameter values together with occupancies as for the standard HMM synthesis framework.

The *minimum description length (MDL)* criterion [23] allows automated setting of the clustering threshold ξ for the standard HMM synthesis framework. It sets

$$\xi = \frac{1}{2}\rho k \log N \tag{9}$$

where k is the number of free parameters per leaf, N is the total occupancy of the root node, and ρ is a heuristic scaling factor which should theoretically be 1. We refer to ρ as the *MDL tuning factor*. Previous experiments indicated that $\rho = 1$ is not appropriate for the autoregressive HMM, and suggested using $\rho = 0.3$ instead [12].

C. Synthesis

As discussed in Section IV-E below, the autoregressive HMM with linear basis functions uses a similar form of Gaussian distribution $P(c^i \mid \theta, \lambda)$ to that effectively used by the standard HMM synthesis framework. This common structure makes it possible to use both the standard speech parameter generation algorithm (case 1 in [2]) and speech parameter generation considering global variance [24] with the autoregressive HMM simply by passing the relevant Gaussian parameters into these standard synthesis algorithms.[3]

In fact there is a way to compute the mean trajectory for the autoregressive HMM (with linear basis functions) that is even simpler than the standard speech parameter generation algorithm. The mean functions $m_q^i(c_{t-K:t-1}^i)$ in (4) are affine-linear, and expectation is a linear operator. Therefore the mean vector sequence $\mu^i \triangleq \mathbb{E}[c^i \mid \theta, \lambda]$ can be computed efficiently by a simple one-pass forward recursion over time:

$$\mu_t^i = m_{\theta_t}^i(\mu_{t-K:t-1}^i). \tag{10}$$

We refer to this as the *autoregressive speech parameter generation algorithm*. A minor memory saving when using this algorithm is to discard the variance parameters $((\sigma^2)_q^i)$, which do not appear in (10). As we will see in Section IV-C this algorithm has particular advantages in the case of real-time, low latency synthesis.

IV. Comparison

In this section we look at some of the similarities and differences between the the standard HMM synthesis framework, the trajectory HMM and the autoregressive HMM.

A. Consistency

The autoregressive HMM and trajectory HMM are both consistent—they use the same normalized (probabilities sum to 1) model during training and synthesis. As discussed in Section II-B the standard HMM synthesis framework is inconsistent, with an unnormalized model effectively used during training. One of the consequences of the lack of normalization present during training in the standard approach is that it greatly underestimates predictive variance [13].

B. Efficiency of Parameter Estimation

Parameter estimation for the autoregressive HMM and the standard HMM synthesis framework is more efficient than for the trajectory HMM.

For the simplest form of training assuming a fixed state sequence, the autoregressive HMM and standard HMM synthesis framework have separable closed form solutions for the maximum likelihood parameters. In contrast the trajectory HMM does not have a closed form solution for the variance parameters, and requires a gradient descent scheme to optimize these [3]. The mean parameters do have a closed form solution, but it is not separable over the parameters for different states and involves solving a potentially large set of linear equations.

For the autoregressive HMM and standard HMM synthesis framework the distribution $P(c, \theta \mid l, \nu, \lambda)$ factorizes over time with respect to the state sequence θ, which allows the Viterbi and Forward-Backward algorithms to be used. In contrast the trajectory HMM must resort to an approximate delayed decision Viterbi decoder for alignment.

The above two points mean that the autoregressive HMM and standard HMM synthesis framework both support efficient re-estimation using expectation maximization and efficient decision tree clustering whereas the trajectory HMM does not.

It should be noted that the autoregressive HMM can be less efficient during training than the standard HMM synthesis framework if very large depths are used. Accumulation requires $O(D^2)$ memory and the M-step of re-estimation requires $O(D^3)$ time, where $D = K+1$ is the number of basis functions, in contrast to the standard HMM which requires $O(D)$ memory and time, where D is the number of windows. However for the typical depths used in this paper of $K = 2$ or $K = 3$ this effect is not substantial.

C. Low Latency Synthesis

As discussed in Section III-C the autoregressive HMM supports a speech parameter generation algorithm not available for the standard approach or the trajectory HMM. This autoregressive speech parameter generation algorithm has particular advantages in the case of real-time, low latency synthesis.

The standard speech parameter generation algorithm (case 1 in [2]) involves a Cholesky decomposition and requires $O(T)$ time to compute the first frame. This means latency can potentially be high, and both latency and memory usage are not predictable at design time since utterances vary in length. In practice a time-recursive version [25] of the speech parameter generation algorithm is often used in real-time synthesis systems and other applications that would otherwise use the standard speech parameter generation algorithm and which require low latency [25]–[27]. This time-recursive algorithm is approximate and slower but has predictable latency, memory and CPU requirements.

In contrast the autoregressive speech parameter generation algorithm above requires only $O(1)$ time to compute the first frame, and so is exact, low latency, and has predictably small memory and CPU requirements.[4]

D. Stability

The autoregressive HMM suffers from a potential pathology. We refer to a set of autoregressive coefficients $(a_q^{id} : d)$ as *stable* if the autoregressive linear filter given by the same

[3]It might be thought that for speech parameter generation considering global variance the autoregressive HMM and the trajectory HMM would require a different weighting factor than the standard HMM synthesis framework since the latter underestimates predictive variance [13] and so penalizes trajectories away from the mean more harshly than the autoregressive HMM and the trajectory HMM. In practice this has been found not to be necessary since in all three cases the extra term in the global variance cost function essentially acts as a hard constraint to set the global variance of the synthesized utterance to the mean of the global variance pdf.

[4]Note that latency is still high when using speech parameter generation considering global variance with the autoregressive HMM. In general for low latency synthesis post-filtering [28], [29] is more practical than using speech parameter generation considering global variance.

coefficients is bounded input, bounded output (BIBO)-stable [30]. If the autoregressive coefficients for a state are unstable and the duration of the state during synthesis is much longer than the typical duration of that state during training, then it is possible for the mean trajectory to diverge outside the range of values which are plausible for that feature vector component.

In principle (6) could be replaced with an equation that estimated the maximum likelihood solution given the constraint that the re-estimated coefficients are stable. Alternatively Quillen has suggested two heuristic schemes to ensure the re-estimated autoregressive coefficients are stable [14].

In this paper we make no effort to ensure estimated coefficients are stable. Divergent trajectories are unlikely to occur with standard synthesis algorithms and duration models since typical durations of states during synthesis are similar to those during training. Indeed in practice we have only ever observed divergent trajectories occasionally when using unusual alignments and poorly trained systems, even though there are many states with unstable coefficients in a typical trained system. Allowing states with unstable autoregressive coefficients may even be beneficial to synthesis quality since this provides a slightly richer model class.

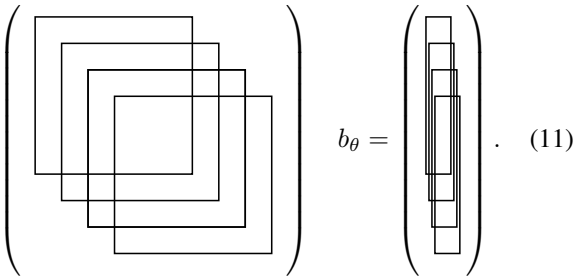
We suspect that this pathology is either impossible or much less likely to occur for the standard HMM synthesis framework and trajectory HMM with conventional windows.

E. Decomposition Into Local Contributions

There is a strong similarity in the form of the distribution $P(c^i \,|\, \theta, \lambda)$ used by the standard HMM synthesis framework, the trajectory HMM and the autoregressive HMM with linear basis functions. In all three cases $P(c^i \,|\, \theta, \lambda)$ is (proportional to) a multidimensional Gaussian with band diagonal precision matrix [3], [21].

Furthermore there is a strong similarity in the way the parameters of this Gaussian depend on the state sequence θ. For all three models the precision matrix P_θ and *b-value* b_θ, which is related to the mean trajectory μ_θ by $P_\theta \mu_\theta = b_\theta$, may be decomposed as a sum of overlapping local contributions, where successive local contributions are functions of the state at successive times e.g. $\theta_1, \theta_2, \theta_3, \theta_4$ [21]. Schematically

$$P_\theta = \left(\quad \right) \qquad b_\theta = \left(\quad \right) . \tag{11}$$



The difference between the models is in the form of the local contributions. For the standard HMM synthesis framework and the trajectory HMM each local contribution to the precision matrix P_θ is a state-dependent sum of the outer product of a fixed set of vectors, whereas for the autoregressive HMM each local contribution to P_θ is the outer product of a state-dependent sum of a fixed set of vectors [21].

F. The Trajectory HMM as a Generalized Autoregressive HMM

Any trajectory HMM can be viewed as an instance of a generalized form of autoregressive HMM [31]. For the trajectory HMM with conventional windows (± 1 frame)

$$P(c \,|\, \theta, \lambda) = \prod_t P(c_t \,|\, c_{t-2:t-1}, \theta_{t-1:T}, \lambda) \tag{12}$$

where the dependence on $c_{t-2:t-1}$ is linear-Gaussian. The corresponding equation for the autoregressive HMM is (1). This shows that the trajectory HMM may be rewritten in the form of a linear-Gaussian autoregressive model, but where the parameters of the linear-Gaussian distribution at each time depend not only on the current state, but also on the remaining duration of the current state, the duration of the next state, etc. This conceptual viewpoint is sometimes useful when comparing the behavior of the trajectory HMM and the autoregressive HMM.

V. Experiments

We performed two sets of experiments to investigate the autoregressive HMM for speech synthesis. Firstly we compared the autoregressive HMM to the standard HMM synthesis framework and to the trajectory HMM in both subjective and objective evaluations. Secondly we investigated the possible choices for structure parameters such as depth and number of states for the autoregressive HMM using objective evaluations.

A. Experimental Metrics

We use naturalness as judged by human opinion scores as the metric for the subjective evaluation [32]. For the objective evaluations we use two metrics.

Test set log probability (TSLP) is the log probability the model assigns to an unseen test set

$$\text{TSLP}((l, c^{\text{nat}}), \lambda) \triangleq \log P(c^{\text{nat}} \,|\, l, \lambda) \tag{13}$$

where (l, c^{nat}) is an unseen pair of label sequence and natural feature vector sequence. We quote TSLP values per frame.

Mel cepstral distortion (MCD) [33] is a measure of the difference between a synthesized mel cepstral sequence and the corresponding natural mel cepstral sequence. We use a form of MCD based on dynamic time warping. Full details are given in Section B. To compute the MCD score for an unseen pair (l, c^{nat}) we take c^{synth} to be the feature vector sequence output by the standard speech parameter generation algorithm for the label sequence l.

Test set log probability and mel cepstral distortion provide complementary views of a model. TSLP is a natural measure of how well a model predicts unseen frames, and achieving a high TSLP requires a model to have both accurate mean trajectories and accurate trajectory covariances. It also allows us to detect over-fitting. MCD provides useful information about the accuracy of the mean trajectories independent of the trajectory covariances.

B. Experimental Systems

We built a standard HMM synthesis system, a trajectory HMM system, and several autoregressive systems.

The systems were trained on the CMU ARCTIC corpus [34] for the single speaker 'slt' (approximately 1 hour), with 50 held-out utterances. The original waveforms had a sampling frequency of 16 000 Hz. The spectral portion of the feature vector consisted of 40-dimensional mel cepstra (mcep) [35] with frequency warping factor $\alpha = 0.42$, the fundamental frequency portion of the feature vector consisted of $\log F_0$, and the aperiodicity portion of the feature vector consisted of 5-band aperiodicity [36]. We used STRAIGHT vocoding [37]. A frame shift of 5 ms was used, and F_0 was estimated using STRAIGHT (min 80 Hz, max 350 Hz).

The standard and autoregressive systems were built using HTS 2.1 [16]. The similarity in parameter estimation and synthesis methods between the autoregressive HMM and standard HMM synthesis framework allowed us to implement the autoregressive HMM relatively easily using HTS, though there are some important adjustments required such as passing the decision tree clustering algorithm accumulators rather than re-estimated parameters as discussed in Section III-B.

All systems used a 5-state (by default) left-to-right topology for modeling at the phone level, with Gaussian explicit duration models for each state used during both parameter estimation and synthesis [18]. We made a minor modification to HTS to ensure explicit duration distributions are properly normalized wherever they are used, though the M-step re-estimation equations were not modified. In the standard version of HTS these distributions are not fully normalized due to the fact a Gaussian pdf is being used for a random variable with range the positive integers.

For the autoregressive systems the spectral and aperiodicity portions of the feature vector were modeled using the autoregressive HMM, with a depth of 3 and an MDL tuning factor of 0.3 by default. For the standard system these portions of the feature vector were modeled using the standard HMM with the conventional three windows [16], a single Gaussian with diagonal covariance per state, and an MDL tuning factor of 1.0. For all systems the F_0 portion of the feature vector was modeled using standard multi-space distributions [38] with the conventional three windows [16] and an MDL tuning factor of 1.0. This means that even the autoregressive systems suffer from some inconsistency between training and synthesis since the F_0 portion of the feature vector is still modeled using the inconsistent standard approach.

It is possible to model F_0 using the autoregressive HMM. Perhaps the most natural approach would be to use a continuous F_0 model such as that used by Yu [39], though more complicated approaches would also fit naturally within the autoregressive framework. Investigation of these further departures from the standard approach is left for future work.

The training regime for the standard and autoregressive systems was adapted from the HTS speaker dependent training demo [16], with monophone initialization based on initial phone-level alignments derived from a monophone speech recognition-style system followed by monophone embedded re-estimation, decision tree clustering, embedded re-estimation, another round of decision tree clustering, and further embedded re-estimation.

TABLE I
SYSTEMS USED TO COMPARE THE AUTOREGRESSIVE HMM TO EXISTING MODELS

system	description
N	natural speech
S	standard HMM synthesis framework
SB	system S with artificial 3× variance boost
T	trajectory HMM
A	autoregressive HMM (standard structure parameters)
AM	autoregressive HMM (modified structure parameters)

The trajectory HMM system took the trained standard system as a starting point, and re-estimated the spectral leaf parameters based on a fixed alignment.

The synthesized trajectories for all systems were produced using speech parameter generation considering global variance [24]. The extensions to HTS and the HTS demo we used to implement the autoregressive HMM are released open source and are available for download [15].

C. Experiment 1—Comparison to Existing Models

To evaluate the autoregressive HMM for synthesis we compared the baseline standard HMM system, the trajectory HMM system and two autoregressive systems using both subjective and objective metrics. The systems under comparison are shown in Table I. System A is a "conventional" autoregressive system (5 states, depth 3, MDL tuning factor of 0.3) which has structure parameters which give good TSLP. System AM is an autoregressive system with structure parameters tuned to have good MCD (5 states, depth 2, MDL tuning factor of 0.18). System SB is system S with a uniform variance boost (see below for details).

The subjective evaluation was conducted with systems N, S, T, A and AM following a methodology similar to that used for the Blizzard Challenge [32]. The listening test consisted of 10 sections of 5 utterances each. For all sections listeners were asked to rate the *naturalness* of each utterance on a scale of 1 to 5 inclusive. Prompts were the 50 held-out utterances in a fixed order. Listeners were allotted to one of 5 groups, and the ordering of the systems for each group was determined with a balanced Latin square design. The listening test was presented via an interactive website over two weeks.

For the objective evaluation we computed test set log probability and MCD on the 50 held-out utterances. For this experiment only (comparison to existing models) we computed the TSLP as $\log P(c^{0:39} \mid \theta^*, \lambda)$ where $c^{0:39}$ is the spectral portion of the feature vector sequence and θ^* is the median alignment (Section A) computed using all portions of the feature vector sequence c. Alignments based on system A were used to evaluate system A, system AM to evaluate system AM, and system S to evaluate systems S, SB and T. A fixed state sequence was used because the true test set log probability, which is obtained by marginalizing $P(c, \theta \mid \lambda)$ over θ, is difficult to compute for the trajectory HMM and the standard HMM synthesis framework. Note that for the standard HMM synthesis framework the test

TABLE II
COMPARISON OF THE AUTOREGRESSIVE HMM TO EXISTING MODELS

system	mcep leaves	opinion score		approx mcep-only TSLP (nats)	MCD (dB)
		mean	median		
N	-	4.7	5	-	-
S	812	2.4	2	29.3	5.6
SB	812	-	-	46.9	5.6
T	812	2.6	3	47.6	5.5
A	771	2.1	2	47.9	5.9
AM	2879	2.4	2	47.6	5.6

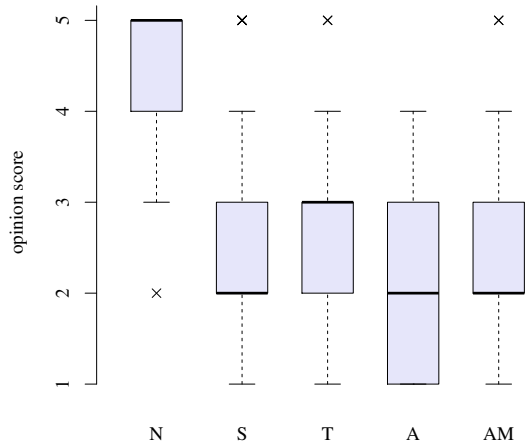


Fig. 3. Box plot showing results of the subjective evaluation.

set log probabilities we compute are for the model effectively used during synthesis, i.e. the trajectory HMM with the same parameters, not the model used during training.

In total 36 native English speakers (various dialects) completed the evaluation. Table II shows a summary of the results. Fig. 3 is an opinion score box plot [40], and a matrix of statistically significant differences between the various systems is shown in Table III. Fig. 4 shows a *complementary cumulative plot* of these results, which displays more information than

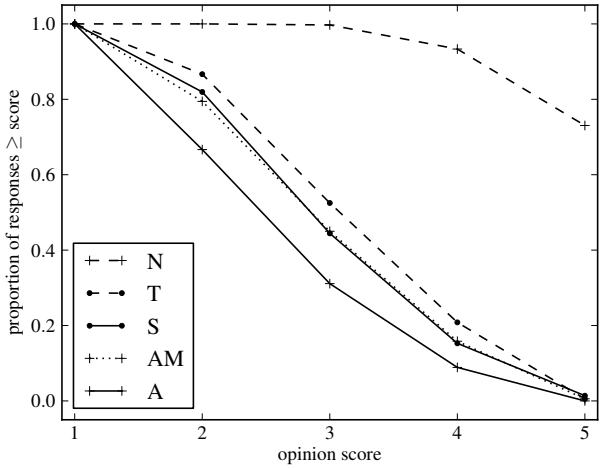


Fig. 4. Complementary cumulative plot showing results of the subjective evaluation in more detail. For an opinion score s, the ordinate gives the proportion of participant responses that were s or greater. For any given opinion score larger ordinate values are better.

TABLE III
PAIRWISE COMPARISONS OF SIGNIFICANT DIFFERENCES BETWEEN NATURALNESS USING BONFERRONI-CORRECTED MANN-WHITNEY U TESTS (■ INDICATES A SIGNIFICANT DIFFERENCE AT 1%)

	N	S	T	A	AM
N		■	■	■	■
S	■		□	■	□
T	■	□		■	□
A	■	■	■		■
AM	■	□	□	■	

the box plot. We can see that the modified autoregressive system (AM) has extremely similar performance to the standard HMM synthesis framework (S). The trajectory system (T) does slightly better than these two systems, and the default autoregressive system (A) does noticeably worse than these two systems. The statistical significance test has S, T and AM possibly identical in performance with A statistically different from the other three. The results also show that MCD was more useful than TSLP as a surrogate for human judgment when selecting model structure parameters for the autoregressive system.

The objective results are presented in Table II. To provide some intuitive calibration of the TSLP and MCD scales, the reader may be interested to know that using full context instead of monophone models results in a typical improvement of roughly $+0.4$ to $+0.6$ nats for approximate mcep-only TSLP, and roughly -1.1 dB to -1.3 dB for MCD.

We can see that the trajectory system (T) and the modified autoregressive system (AM) are comparable in terms of test set log probability. The default autoregressive system (A) does quite a bit better than any of the other systems. The standard system (S) has extremely low test set log probability, due to the fact it systematically underestimates predictive variance [13]. For interest we also computed the test set log probability of the standard system with a multiplier of 3 applied to the covariance of each trajectory (equivalently, a multiplier of 3 applied to every variance parameter in the system).[5] This results in a system (SB) that no longer systematically underestimates predictive variance and has a much greater test set log probability. However there is still a large gap between the variance-boosted standard system (SB) and the normalized models. These results suggest that the autoregressive HMM performs favorably compared to existing models as a probabilistic model of speech.

The MCD results are qualitatively similar to the subjective listening test results. The modified autoregressive system (AM) and the standard HMM synthesis framework (S) have very similar MCD, with the trajectory HMM system (T) very slightly better and the default autoregressive system (A) noticeably worse. Thus the trajectory HMM appears to provide the best model of the mean trajectory. These results suggest that the autoregressive HMM inherently provides a slightly

[5]The value of 3 is close to optimal for all mcep components, in the sense of maximizing test set log probability amongst the family of all possible uniform variance boosts. In preliminary experiments we observed that the optimal uniform variance boost for standard systems is often roughly the number of windows (here 3).

poorer model of the mean trajectory than the standard HMM synthesis framework, but that MCD performance on the level of the standard approach can be obtained with the autoregressive HMM by using more leaves (system AM).

It should be noted that the MCD results appear to depend strongly on the precise form of MCD used. In preliminary experiments with forms of MCD using a fixed alignment rather than dynamic time warping, we found that for some methods of computing the alignment systems S and AM were similar in MCD score, but for other methods of computing the alignment system S was noticeably better than system AM.

D. Experiment 2—Model Structure Investigation

Our second set of experiments investigated the possible choices for model structure parameters such as number of states, depth and MDL tuning factor for the autoregressive HMM. The customary values used for the standard HMM synthesis framework may not be optimal for the autoregressive HMM, and some parameters such as depth have no direct analog in the standard framework. Investigating these choices involves evaluating an extensive set of systems, and so we chose to measure objective performance only.

Using the systems A (5 states, depth 3, MDL tuning factor 0.3) and AM (5 states, depth 2, MDL tuning factor 0.18) as starting points, we varied the model structure parameter under investigation. All other aspects of the systems were as specified in Section V-B.

Ideally for each number of states and depth considered we would choose the optimal MDL tuning factor. However just choosing the MDL tuning factor which achieves the best score on the test set would involve substantial re-use of the test set, and conducting a full 3-dimensional search with a held-out validation set or using cross validation would be computationally intensive. Therefore we only used MDL tuning factors of 0.18 and 0.3, except for the depth 0 case where an MDL tuning factor of 1.0 was suspected on theoretical grounds (the depth 0 autoregressive HMM is just a conventional HMM) and experiments confirmed this was better.

We also report the test set log probability and MCD of the monophone system below, since this is not sensitive to the choice of MDL tuning factor.

1) Depth: We trained autoregressive systems of various depths. The depth was varied for the spectral portion of the feature vector only. We found that local maxima during training were a problem, with the *training set* log likelihood for depth 5 monophone models lower than for depth 4 monophone models in spite of the fact depth 4 models are a special case of depth 5 models. Thus we decided to train more carefully, starting with depth 0 models and gradually increasing the depth of the model with several iterations of embedded re-estimation in between.

The results are shown in Fig. 5. The optimal test set log probability is at depth 3 with system A, and depths 3, 4 and 5 are all close to optimal. We can see that increasing the depth gives decent improvements in TSLP up to depth 3, and thereafter results in minor degradation. The optimal MCD is at depth 1 with system AM, and depths 1, 2 and 3 are all close to optimal. We can see that increasing the depth gives a decent

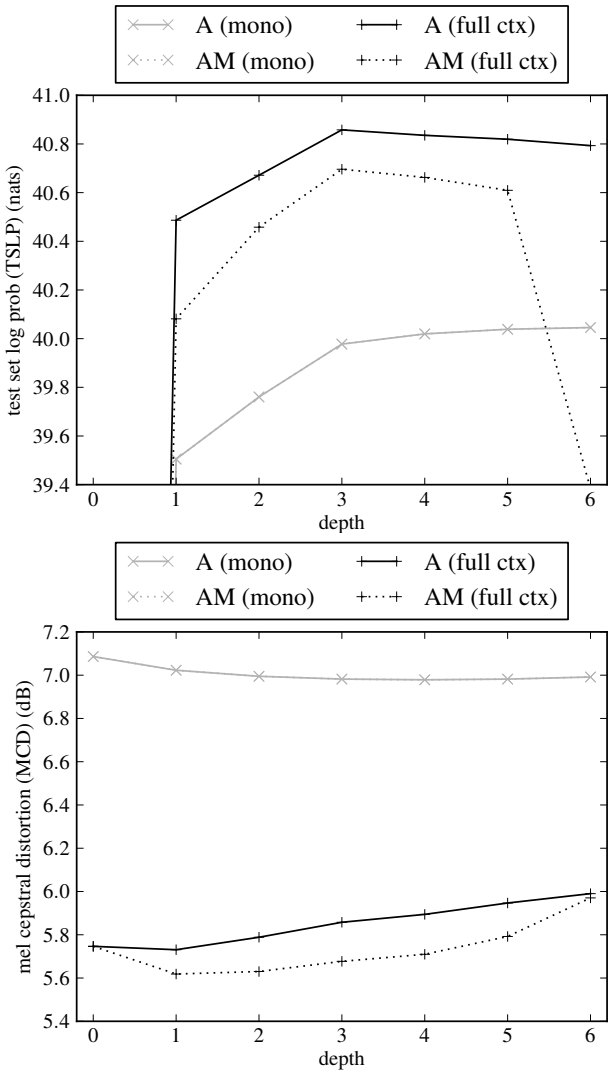


Fig. 5. How depth affects (top) test set log probability and (bottom) mel cepstral distortion. The monophone versions of system A and system AM differ only in depth and so have identical results in this figure.

improvement in MCD up to depth 1, and thereafter results in minor degradation.

The gradual training procedure used here was not used in the other experiments. Without gradual training depth 3 system A is still optimal in terms of TSLP, but depth 2 system AM is now optimal in terms of MCD and -0.10 dB better than depth 1. For the depth 3 system A and depth 2 system AM used elsewhere the difference made by gradual training was minimal (TSLP within 0.04 nats and MCD within 0.02 dB).

We therefore suggest 2 or 3 is the most appropriate choice of depth for the autoregressive HMM.

2) Number of States: We trained autoregressive systems with various numbers of states. The results are shown in Fig. 6. We can see a clear peak in TSLP at the conventional value of 5 states for system A, and 5 and 6 states are both close to optimal. The optimal MCD is at 5 states with system AM, and 5, 6 and 7 states are all close to optimal.

The convention of using 5 states inherited from the standard HMM synthesis framework is thus appropriate for the autore-

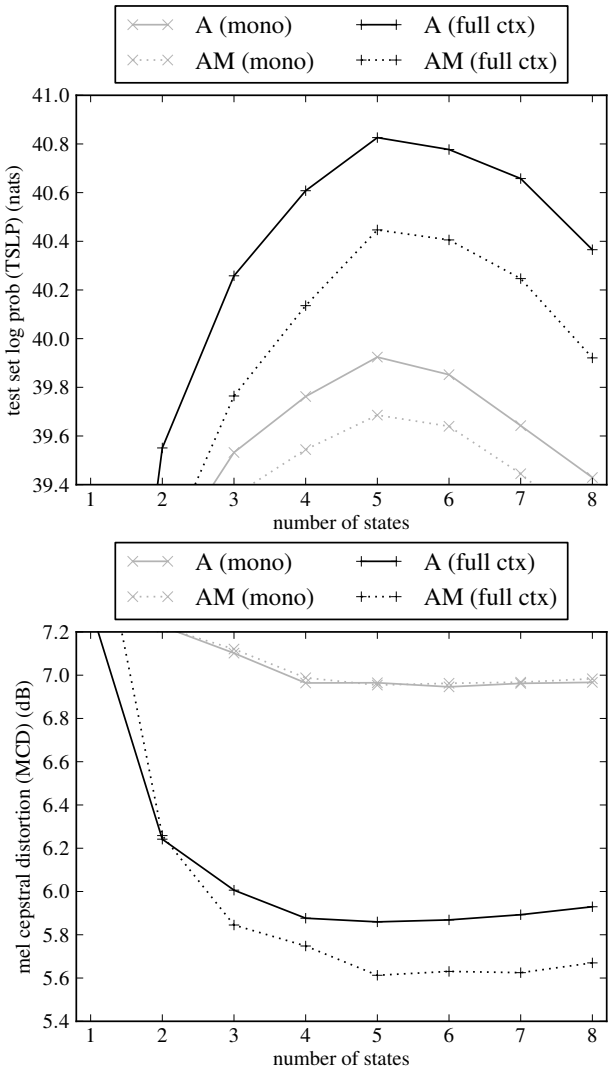


Fig. 6. How number of states affects (top) test set log probability and (bottom) mel cepstral distortion.

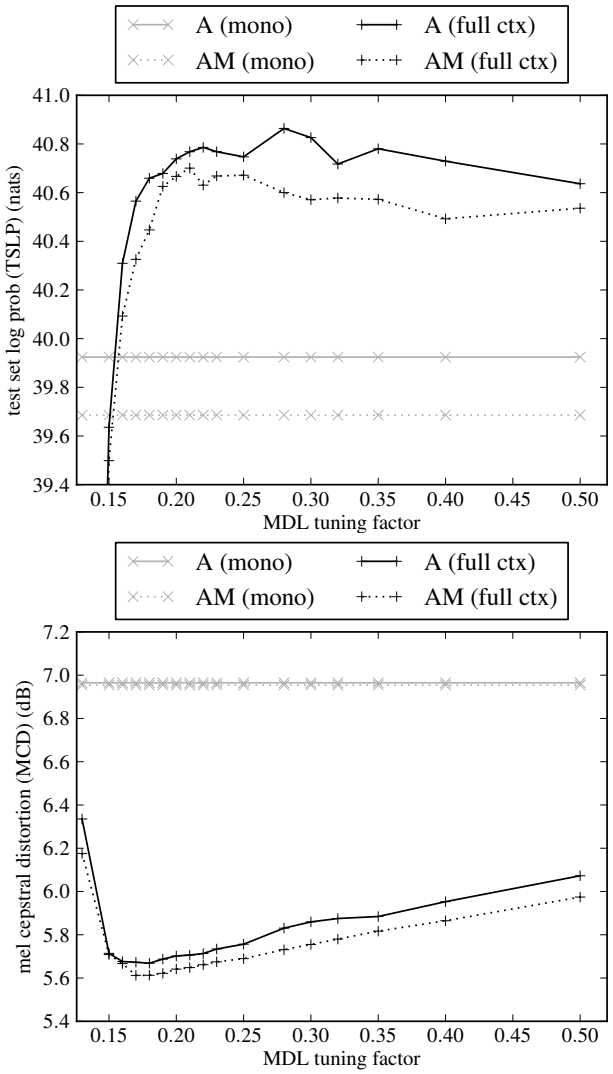


Fig. 7. How MDL tuning factor affects (top) test set log probability and (bottom) mel cepstral distortion.

gressive HMM.

We noticed an effect where the *training set* log likelihood was lower for the monophone models with 6 to 8 states than with 5 states, which at first seems surprising. However instead of being a local maxima effect as in the case of depth, preliminary investigations suggest that this is mainly caused by the minimum duration restriction. Since each state is restricted to a minimum duration of 1 frame, a 5-state model has a minimum phone duration of 5 frames and an 8-state model has a minimum phone duration of 8 frames. This warrants further investigation.

3) MDL Tuning Factor: We trained autoregressive systems with various MDL tuning factors used during decision tree clustering. The MDL tuning factor was varied for the spectral portion of the feature vector only. The results are shown in Fig. 7. For the TSLP of system A we can see that there is no clear peak, with any MDL tuning factor from 0.20 to 0.35, corresponding to a total of roughly 600 to 1600 mcep leaves, being close to optimal. For the MCD of system AM we see a narrower range of good MDL tuning factors, with any value from 0.17 to 0.20, corresponding to a total of roughly 2000 to 3500 mcep leaves, being close to optimal. It should be noted that the number of leaves increases very rapidly as we lower the MDL tuning factor from 0.18—system AM has around 1000 mcep leaves at an MDL tuning factor of 0.3, 2900 leaves at 0.18, 6400 leaves at 0.15, and 16 000 leaves at 0.13.

We therefore suggest a value between 0.18 and 0.3 is the most appropriate choice of MDL tuning factor for the autoregressive HMM.

VI. Discussion

We have seen that the form of autoregressive HMM with linear basis functions explored here appears to be capable of matching but not bettering the naturalness of the standard HMM synthesis framework. The consistency of the autoregressive HMM means that it does not grossly underestimate predictive variance in the same way as the standard HMM synthesis framework [13], and even once this flaw is corrected (system SB above) the autoregressive HMM has a better model of trajectory covariance as evidenced by a much greater test set log probability. The autoregressive HMM seems to provide

an inherently slightly poorer model of the mean trajectory than the standard HMM synthesis framework, but this can be compensated for by using more leaves.

Compared to the standard HMM synthesis framework, the trajectory HMM has slightly better mean trajectories, much better trajectory covariances, and a higher naturalness score. Compared to the autoregressive HMM, the trajectory HMM has better mean trajectory modeling, but appears to have slightly worse trajectory covariance modeling.

We have also seen that for the autoregressive HMM we obtain higher naturalness on this corpus using a depth of 2 and an MDL tuning factor of 0.18, which roughly corresponds to optimizing MCD, rather than a depth of 3 and an MDL tuning factor of 0.3, which roughly corresponds to optimizing TSLP.

It is interesting to consider reasons the autoregressive HMM might model the mean trajectory worse than the trajectory HMM. One natural candidate is the fact that the trajectory HMM inherently incorporates a notion of lookahead (see Section IV-F), and so the mean trajectory starts to smoothly transition to a value suitable for the next state while still in the current state. In contrast the autoregressive HMM must encode such information by judicious use of right-context questions in the decision tree. This warrants further investigation.

It is worth noting that while the autoregressive HMM appears to provide the best trajectory covariances, none of the models go very far towards capturing the true trajectory covariance structure present in speech, as evidenced by the fact that sampled trajectories from all three models sound bad [13].

Appendix A
Median Alignments

Here we briefly describe *median alignments*. When a left-to-right topology is used for modeling at the label and sub-label levels as is common, each label sequence l defines a sequence of (label, sub-label) pairs. At each time t the marginal posterior $P(\theta_t \mid l, c, \nu, \lambda)$ defines a distribution over the index of the current (label, sub-label) pair within this sequence. The median alignment θ^* is obtained by at each time t choosing θ_t^* to be median of this distribution. Computationally median alignments are easily obtained using the Forward-Backward algorithm. Median alignments have nicer theoretical properties with respect to marginalization than Viterbi alignment, though in practice there is often little difference between the two.[6]

Appendix B
DTW-Based Mel Cepstral Distortion

We use the following form of MCD based on dynamic time warping:

$$\mathrm{MCD}(c, \tilde{c}) \triangleq \frac{k}{T(c)} \min_{\pi \in \Pi} \sum_{(s,t) \in \pi} \left(\sum_{i=1}^{39} \left(c_s^i - \tilde{c}_t^i \right)^2 \right)^{0.5} \quad (14)$$

[6]It might be thought that median alignments may have pathologies such as later labels appearing before earlier labels. This would not necessarily be a problem for our application if it did occur. However for the form of models we use, the median alignments are in fact guaranteed to correspond to valid state sequences. In particular they are always left-to-right, and satisfy the constraint that each sub-label must last for at least 1 frame.

where $k \triangleq \sqrt{2} \cdot 10 / \log 10$, c and $\tilde{c}$ are feature vector sequences, $c^{0:39}$ is the spectral portion of a feature vector sequence c, $T(c)$ is the number of frames in a feature vector sequence c, $\pi \subset \mathbb{N} \times \mathbb{N}$ is a relation between frames in the natural and the synthesized feature vector sequences, and Π is the set of admissible relations. A relation $\pi \subset \mathbb{N} \times \mathbb{N}$ is *admissible* if there is a sequence $((s_p, t_p))_{p=1}^{P}$ such that $(s_1, t_1) = (1, 1)$, $(s_P, t_P) = (T^{\text{nat}}, T^{\text{synth}})$, $(s_{p+1} - s_p, t_{p+1} - t_p)$ is either (0, 1), (1, 0) or (1, 1) for $p = 1, \ldots, P-1$, and the sets $\{(s_p, t_p)\}$ and π are equal. The minimum over admissible relations is computed using dynamic time warping. We compute the MCD between a natural feature vector sequence c^{nat} and a synthesized feature vector sequence c^{synth} as $\mathrm{MCD}(c^{\text{nat}}, c^{\text{synth}})$.

Acknowledgment

The authors are very grateful to the organizers of the Blizzard Challenge for providing the scripts used to conduct the subjective evaluation.

References

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.
[2] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP 2000*, 2000, pp. 1315–1318.
[3] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech and Language*, vol. 21, no. 1, pp. 153–173, 2007.
[4] C. Wellekens, "Explicit time correlation in hidden Markov models for speech recognition," in *Proc. ICASSP 1987*, vol. 12, 1987, pp. 384–386.
[5] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 220–225, 1990.
[6] P. C. Woodland, "Hidden Markov models using vector linear prediction and discriminative output distributions," in *Proc. ICASSP 1992*, 1992, pp. 509–512.
[7] J. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical foundations of speech and language processing*, M. Johnson, S. P. Khudanpur, M. Ostendorf, and R. Rosenfeld, Eds. Springer-Verlag, 2004.
[8] K. K. Chin and P. C. Woodland, "Maximum mutual information training of hidden Markov models with vector linear predictors," in *Proc. Interspeech 2002*, 2002, pp. 997–1000.
[9] A. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. ICASSP 1982*, vol. 7, 1982, pp. 1291–1294.
[10] B. H. Juang and L. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 6, pp. 1404–1413, 1985.
[11] M. Shannon and W. Byrne, "Autoregressive HMMs for speech synthesis," in *Proc. Interspeech 2009*, 2009, pp. 400–403.
[12] ——, "Autoregressive clustering for HMM speech synthesis," in *Proc. Interspeech 2010*, 2010, pp. 829–832.
[13] M. Shannon, H. Zen, and W. Byrne, "The effect of using normalized models in statistical speech synthesis," in *Proc. Interspeech 2011*, 2011, pp. 121–124.
[14] C. Quillen, "Autoregressive HMM speech synthesis," in *Proc. ICASSP 2012*, 2012, pp. 4021–4024.
[15] EMIME consortium, "Tools," http://www.emime.org/participate/tools, accessed 21 March 2012.
[16] HTS working group, "HMM-based speech synthesis system (HTS)," http://hts.sp.nitech.ac.jp/, accessed 21 March 2012.
[17] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," in *Proc. ICSLP 1998*, 1998.
[18] H. Zen, K. Tokuda, T. Masuko, T. Kobayasih, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[19] S. Z. Yu and H. Kobayashi, "An efficient forward-backward algorithm for an explicit-duration hidden Markov model," *IEEE Signal Process. Lett.*, vol. 10, no. 1, pp. 11–14, 2003.

[20] H. Zen, "Implementing an HSMM-based speech synthesis system using an efficient forward-backward algorithm," Nagoya Institute of Technology, Technical Report TR-SP-0001, 2007.

[21] M. Shannon and W. Byrne, "A formulation of the autoregressive HMM for speech synthesis," Department of Engineering, University of Cambridge, UK, Technical Report CUED/F-INFENG/TR.629, 2009, http://mi.eng.cam.ac.uk/~sms46/papers/shannon2009fah.pdf.

[22] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Human Language Technology Workshop*, 1994, pp. 307–312.

[23] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn. (E)*, vol. 21, no. 2, pp. 79–86, 2000.

[24] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[25] K. Koishida, K. Tokuda, T. Masuko, and T. Kobayashi, "Vector quantization of speech spectral parameters using statistics of static and dynamic features," *IEICE Trans. Inf. Syst.*, vol. E84-D, no. 10, pp. 1427–1434, 2001.

[26] T. Muramatsu, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory," in *Proc. Interspeech 2008*, 2008, pp. 1076–1079.

[27] W. Han, L. Wang, F. Soong, and B. Yuan, "Improved minimum converted trajectory error training for real-time speech-to-lips conversion," in *Proc. ICASSP 2012*, 2012, pp. 4513–4516.

[28] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE Trans. Inf. Syst. (Japanese edition)*, vol. J87-D-II, no. 8, pp. 1565–1571, 2004.

[29] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method," in *Proc. Blizzard Challenge Workshop 2006*, 2006.

[30] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, 1st ed. Prentice Hall, 1975, p. 15.

[31] M. Shannon and W. Byrne, "Viewing the trajectory HMM as a generalized autoregressive HMM," Department of Engineering, University of Cambridge, UK, Technical Report CUED/F-INFENG/TR.677, 2012, http://mi.eng.cam.ac.uk/~sms46/papers/shannon2012viewing.pdf.

[32] A. W. Black and K. Tokuda, "The Blizzard Challenge 2005: Evaluating corpus-based speech synthesis on common datasets," in *Proc. Interspeech 2005*, 2005, pp. 77–80.

[33] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, 1993, pp. 125–128.

[34] J. Kominek and A. W. Black, "The CMU ARCTIC databases for speech synthesis," Carnegie Mellon University, Technical Report CMU-LTI-03-177, 2003.

[35] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP 1992*, 1992, pp. 137–140.

[36] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 325–333, 2007.

[37] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveign, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[38] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Trans. Inf. Syst.*, vol. E85-D, no. 3, pp. 455–464, 2002.

[39] K. Yu, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Trans. Audio Speech Language Process.*, vol. 19, no. 5, pp. 1071–1079, 2011.

[40] R. A. J. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the Blizzard Challenge 2007 listening test results," in *Proc. Blizzard Challenge Workshop 2007*, 2007.

Matt Shannon (S'12) received the B.A. (Hons) degree in mathematics in 2005, M.Math. degree in mathematics in 2006, and M.Phil. degree in computer, speech, text and internet technology in 2008 from the University of Cambridge, Cambridge, U.K.

He did an internship in speech synthesis at Google, U.K. in 2011. Currently he is a Ph.D. student in the Department of Engineering, University of Cambridge, Cambridge, U.K. His research interests include statistical speech synthesis and speech recognition.

Heiga Zen (M'10) received the A.E. degree from the Suzuka National College of Technology, Japan, in 1999, and the B.E., M.E., and Ph.D. degrees from the Nagoya Institute of Technology, Japan, in 2001, 2003, and 2006, respectively. From 2004 to 2005, 2006 to 2008, and 2008 to 2011, he worked at the IBM T. J. Watson Research Center, NY, Nagoya Institute of Technology, and Toshiba Research Europe, U.K., respectively. Currently, he is a Research Scientist at Google, U.K. His research interests include speech recognition and synthesis.

Dr. Zen was awarded a 2006 ASJ Awaya Award, a 2008 ASJ Itakura Award, a 2008 TAF TELECOM System Technology Award, a 2008 IEICE Information and Systems Society Best Paper Award, and a 2009 IPSJ Yamashita SIG Research Award. He is a member of the ASJ and IPSJ, and has been a member of the SLTC since 2012.

William Byrne (M'82–SM'07) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1993.

He is a Reader in Information Engineering in the Department of Engineering, University of Cambridge, Cambridge, U.K. After the Ph.D. degree, he joined The Johns Hopkins University Center for Language and Speech Processing, Baltimore, MD, as an Associate Research Scientist and then as Research Associate Professor. He has worked with several speech and language technology companies, including Entropic Research Laboratory and Voice Signal Technology, and he is currently a Senior Research Scientist with SDL plc. His current research interests are in speech recognition, speech synthesis, and statistical machine translation.

Dr. Byrne is a Fellow of Clare College, Cambridge. He was general co-Chair for the 2003 IEEE Automatic Speech Recognition and Understanding Workshop, a member of the Speech Technical Committee from 2004 through 2006, and an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING from 2006 to 2008.