Speech Recognition using Linear Dynamic Models

Joe Frankel, *Member, IEEE* and Simon King, *Member, IEEE*

***Abstract*— The majority of automatic speech recognition (ASR) systems rely on hidden Markov models, in which Gaussian mixtures model the output distributions associated with sub-phone states. This approach, whilst successful, models consecutive feature vectors (augmented to include derivative information) as statistically independent. Furthermore, spatial correlations present in speech parameters are frequently ignored through the use of diagonal covariance matrices. This paper continues the work of Digalakis and others who proposed instead a first-order linear state-space model which has the capacity to model underlying dynamics, and furthermore give a model of spatial correlations. This paper examines the assumptions made in applying such a model and shows that the addition of a hidden dynamic state leads to increases in accuracy over otherwise equivalent static models. We also propose a time-asynchronous decoding strategy suited to recognition with segment models. We describe implementation of decoding for linear dynamic models and present TIMIT phone recognition results.**

***Index Terms*— LDM, ASR, Stack decoding**

I. Introduction

THE work described in this paper is motivated by the following belief: a model which reflects the characteristics of speech production will ultimately lead to improvements in automatic speech recognition. The articulators move slowly and continuously along highly constrained trajectories, each one capable of a limited set of gestures which are organized in an overlapping, asynchronous fashion. Feature extraction on the resulting acoustic signal produces a piecewise smooth, spatially correlated set of parameters in which neighbouring feature vectors are highly correlated, and dependencies can spread over many frames. An acoustic model should reflect these properties. A number of authors have proposed that this may be approached by modelling speech at the segment rather than frame level, where segment refers to a sub-word unit such as a phone or syllable. A review is given in [1].

This work investigates acoustic modelling using a form of linear state-space model, aiming to enhance speech recognition through the addition of a hidden dynamic representation. State-space models make a distinction between the underlying properties of the system and the parameterization. Allowing the hidden state to be continuous across model boundaries offers the potential to model longer range dependencies, loosening the assumption of inter-segmental independence.

Digalakis' original application of linear dynamic models (LDM) to ASR [2] used a smoothed Gauss-Markov form, though linear dimensionality reduction can form an integral part of these models. We investigate the effect of subspace modelling and form of noise covariance on phone-class discrimination.

Evaluation of new acoustic models for ASR frequently relies on rescoring of hidden Markov model (HMM) lattices. Whilst convenient, rescoring experiments are prone to errors introduced by the models used to generate the lattice. Decoding with segment models can be computationally expensive: unlike frame-level models, it is not always possible to share likelihood calculations for the observations of proposed segments with differing start and end times. We suggest that a stack decoder with A^* search offers an efficient means of jointly searching for the most likely model sequence and segmentation without resorting to rescoring.

II. Linear dynamic models

The LDM (or Kalman filter model) is a generative model with a time-varying multivariate unimodal Gaussian output distribution. The LDM is from the family of linear Gaussian models [3], [4], and is specified by the following pair of equations:

$$\mathbf{y}_t = H\mathbf{x}_t + \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C) \tag{1}$$

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t \qquad \boldsymbol{\eta}_t \sim N(\mathbf{w}, D) \tag{2}$$

and an initial state distribution $\mathbf{x}_1 \sim N(\boldsymbol{\pi}, \Lambda)$. We use $\mathbf{y}_t$ and $\mathbf{x}_t$ to denote p- and q-dimensional observation and state vectors respectively. The state is described by a multivariate Gaussian distribution, and propagation is governed by a first-order autoregressive process with $q \times q$ evolution matrix F and the addition of Gaussian noise $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$. A linear projection via $p \times q$ dimensional H links the observation and state processes, along with the addition of more Gaussian noise $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$, which is assumed uncorrelated to the state noise $\boldsymbol{\eta}_t$. By setting the state to have lower dimensionality than the observations, H is used to encode linear dimensionality reduction. In this way, a distinction is made between the parameterization and the number of degrees of freedom required to describe the underlying spatial and temporal characteristics. A description of the properties and types of trajectories which the LDM generates can be found in [1].

The remainder of this section is arranged as follows: we outline inference, parameter estimation and likelihood calculation in Section II-A, then discuss constraints which affect modelling in Section II-B, efficient computation in Section II-C, and consider the internal structure of the model in II-D. Sections II-E and II-F look at how LDMs may be applied to speech data, and the assumptions which are made in doing so.

A. Inference, parameter estimation and evaluation

The Kalman filter [5] and Rauch-Tung-Striebel (RTS) smoother [6] are used to infer state information given an N-length observation sequence $\mathcal{Y} = \mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and a

Manuscript received September 2004. This work was supported by EPSRC grant GR/S21281/01

Joe Frankel and Simon King are both with the Centre for Speech Technology Research, University of Edinburgh.

set of model parameters Θ. Filtering provides an estimate of the state distribution at time t given all the observations up to and including that time, $p(\mathbf{x}_t|\mathbf{y}_1^t, \Theta)$, and smoothing gives a corresponding complete-data estimate $p(\mathbf{x}_t|\mathbf{y}_1^N, \Theta)$. We use $\hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{x}}_{t|N}$ to denote the filtered and smoothed state means respectively, with $\Sigma_{t|t}$ and $\Sigma_{t|N}$ denoting the corresponding covariances.

Kalman filtering is a recursive process which alternates between making predictions of the state mean and covariance, $\hat{\mathbf{x}}_{t|t-1}$ and $\Sigma_{t|t-1}$, given a set of model parameters and the filtered statistics from the previous time $\hat{\mathbf{x}}_{t-1|t-1}$ and $\Sigma_{t-1|t-1}$, and then updating these to arrive at $\hat{\mathbf{x}}_{t|t}$ and $\Sigma_{t|t}$ given newly observed $\mathbf{y}_t$. The update is made in such a way as to minimize the filtered state covariance $\Sigma_{t|t}$. Applying the RTS smoother yields estimates which are the optimal linear combination of one forward and one backward filter so as to minimize $\Sigma_{t|N}$. The Kalman filter and RTS smoother equations are given in the Appendix.

Parameter estimation uses the expectation maximization (EM) algorithm [7], which iterates toward a generalized maximum likelihood solution by alternately computing complete-data state expectations using the current parameter set, and then updating parameters based on these estimates. Taking an LDM and multiplying one dimension of the state by some factor whilst dividing the corresponding column of H by the same gives distributions over the observations identical to those of the original. Despite the lack of unique parameter estimates and the inherent degeneracy [3], EM training for LDMs is stable in practice and converges quickly, though it is sensitive to initial parameter estimates [1].

Classification and recognition require calculation of the likelihood of a given model generating a section of speech data. The Kalman filter recursions as given in the Appendix include calculation of the prediction error $\mathbf{e}_t$ and associated covariance $\Sigma_{\mathbf{e}_t}$:

$$\mathbf{e}_t = \mathbf{y}_t - H\hat{\mathbf{x}}_{t|t-1} - \mathbf{v} \tag{3}$$

$$\Sigma_{\mathbf{e}_t} = H\Sigma_{t|t-1}H^T + C \tag{4}$$

where $\hat{\mathbf{x}}_{t|t-1}$ and $\Sigma_{t|t-1}$ are the predicted state mean and covariance respectively. With errors assumed uncorrelated and Gaussian, the log-likelihood of an N-frame observed sequence $\mathbf{y}_1^N$ given an LDM with parameter set Θ is calculated as:

$$\log p(\mathbf{y}_1^N|\Theta) = -\frac{1}{2}\sum_{t=1}^{N}\left\{\log|\Sigma_{\mathbf{e}_t}| + \mathbf{e}_t^T\,\Sigma_{\mathbf{e}_t}^{-1}\,\mathbf{e}_t\right\} - \mathcal{K} \tag{5}$$

The normalization term $\mathcal{K} = \frac{Np}{2}\log(2\pi)$ can be omitted when comparing multiple models on a single given section of data.

B. Constraints

Constraints on the LDM parameters can be used to alter the properties of the model. The state noise covariance can be set to the identity or a diagonal matrix with no loss in generality [3]. With a diagonal observation noise covariance C, the output distribution is approximated by a projection of a lower dimensioned state via the observation matrix H. This gives a model with significantly fewer parameters than one with a fully specified noise covariance matrix, though represents a loss in generality.

Setting H to be the identity matrix removes subspace modelling, and gives the smoothed Gauss-Markov form used in [2]. A state of dimension zero, or equating $H = 0$, gives a Gaussian classifier, as all modelling is through the observation noise, $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$. Alternatively, a factor analyzer model [3], [4] sets $F = 0$, the observation noise C to be diagonal, and gives an LDM without state dynamics in which subspace modelling is used to give a reduced-parameter approximation to a full-covariance Gaussian.

One constraint is always enforced during training: F is set to be a decaying mapping, i.e. $|F| < 1$. If $|F| > 1$ were allowed, the state evolution could give a model of exponential growth. Such behaviour may not be apparent over small numbers of frames, whilst still introducing an element of numerical instability. To constrain $|F| < 1$, the singular value decomposition (SVD) is employed immediately after the the re-estimation step as given in [1, Equation 4.27]. The SVD provides a pair of orthonormal bases U and V, and a diagonal matrix of singular values S such that $F = USV^T$. Given that U and V are orthonormal, $|U| = |V| = 1$, and hence $|F| = |S|$. Letting s_{ii} denote element i, i of S, we set $s_{ii}^{'} = \min(s_{ii}, 1-\kappa)$ for $i = \{1, \ldots, q\}$. In this work we used $\kappa = 0.005$, with the result that $|S^{'}| < 1$. By re-computing $F = US^{'}V^T$, the bases of F are preserved whilst forcing the transform along them to be decaying.

C. Efficient implementation

An examination of the relevant filter and smoother recursions reveals that the none of the computations for the 2^{nd} order statistics at time t involve the newly observed value $\mathbf{y}_t$. During the forward pass, the predicted $\Sigma_{t|t-1}$ and posterior $\Sigma_{t|\,t}$ state covariance along with the cross-covariance $\Sigma_{t,\,t-1|t}$, Kalman gain K_t, and error covariance $\Sigma_{\mathbf{e}\,t}$ will then be identical for any pair of observation sequences $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_1}\}$ and $\{\mathbf{y'}_1, \ldots, \mathbf{y'}_{N_2}\}$ for $t \leq N_1, N_2$.

The situation is slightly different for the smoothing pass, though the above also applies to A_t, the backward analogue of the Kalman gain, which is calculated using the filtered parameters $\Sigma_{t-1|t-1}$ and $\Sigma_{t|\,t-1}$. However, the smoothed state covariances are dependent on N, and so $\Sigma_{t-1|N}$ and $\Sigma_{t,t-1|N}$ are identical for any pair of observation sequences $\{\mathbf{y}_1, \ldots, \mathbf{y}_{N_1}\}$ and $\{\mathbf{y'}_1, \ldots, \mathbf{y'}_{N_2}\}$ for which $N_1 = N_2$. These observations lead to implementational strategies in which state covariances and the correction factors K_t and A_t can be calculated, cached, and reused. The matrix operations which are used to compute these quantities form the bulk of the computation of implementing LDMs and so considerable speed-ups can be found by employing such a strategy.

Table I shows estimation and classification speeds for a set of 61 LDMs with observations of 12 MFCCs and energy, and run on a 3.0GHz Pentium P4 processor. Caching computations leads to 6 and 15-fold speed increases on training and classification respectively. For comparison, speeds are also given for estimation and classification using full covariance Gaussian models.

model	task	speed ($\times$ real time)
Gaussian	estimation	0.00002
LDM	1 iteration EM, no caching	0.0006
LDM	1 iteration EM, caching	0.0001
Gaussian	classification	0.06
LDM	classification, no caching	2
LDM	classification, caching	0.01

TABLE I

ESTIMATION AND CLASSIFICATION SPEEDS FOR LDMS AND FULL COVARIANCE GAUSSIANS WITH 13-DIMENSIONAL OBSERVATIONS.

D. A non-traditional view of LDM modelling

An r^{th}-order vector autoregressive (AR) model describing an N-length sequence of p-dimensional random variables $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ can be written as:

$$\mathbf{z}_t = \sum_{i=1}^{r} A_i \mathbf{z}_{t-i} + \boldsymbol{\eta}_t \tag{6}$$

where the A_is are $p \times p$ matrices and $\boldsymbol{\eta}_t$ is additive Gaussian noise given by $\boldsymbol{\eta}_t \sim N(\mathbf{w}, D)$. We introduce the notation:

$$\mathbf{Z}_{t-r+1}^{t} = \left[\mathbf{z}_t^T, \mathbf{z}_{t-1}^T, \ldots, \mathbf{z}_{t-r+1}^T\right]^T \tag{7}$$

$$A_r^p = \begin{bmatrix} A_1 & A_2 & \cdots & & A_r \\ I_p & 0_p & \cdots & & 0_p \\ 0_p & I_p & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0_p & \cdots & 0_p & I_p & 0_p \end{bmatrix} \tag{8}$$

$$\boldsymbol{\eta}_t^p = \left[\boldsymbol{\eta}_t^T, \mathbf{0}_p^T, \ldots, \mathbf{0}_p^T\right]^T \tag{9}$$

where I_p represents a $p \times p$ identity matrix, 0_p a $p \times p$ matrix of zeros, and $\mathbf{0}_p$ a vector of zeros length p. Then letting $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ be a set of p-dimensional observations which we wish to model with the relationship $\mathbf{y}_t = \mathbf{z}_t + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim N(\mathbf{v}, C)$, we can write:

$$\mathbf{y}_t = \begin{bmatrix} I_p & 0_p & \cdots & 0_p \end{bmatrix} \mathbf{Z}_{t-r+1}^{t} + \boldsymbol{\epsilon}_t \tag{10}$$

$$\mathbf{Z}_{t-r+1}^{t} = A_r^p \mathbf{Z}_{t-r}^{t-1} + \boldsymbol{\eta}_t^p \tag{11}$$

Note that 11 is the autoregressive process of Equation 6. By setting the covariance C of $\boldsymbol{\epsilon}_t$ to be zero, the model remains a vector autoregressive process with $\mathcal{Y}$ simply a displaced version of $\mathcal{Z}$. However, the addition of observation noise through $\boldsymbol{\epsilon}_t$ renders $\mathcal{Z}$ a hidden variable, and makes the model described by Equations 10 and 11 a constrained form of LDM. The evolution matrix in Equation 11 holds the original r-order autoregressive process and acts as a shift operator for each component of the stacked state vector $\mathbf{Z}_{t-r+1}^{t}$. Writing the LDM in this form shows how the state can 'remember' past values, here noisy versions of the observations.

With $\mathcal{Z}$ of equal dimension to the observations $\mathcal{Y}$, the state is of dimension rp. Letting $\mathcal{Z}$ be a d-dimensional vector, and with the B_is representing $p \times d$ matrices, Equations 10 and 11 can be written as:

$$\mathbf{y}_t = \begin{bmatrix} B_1 & B_2 & \cdots & B_r \end{bmatrix} \mathbf{Z}_{t-r+1}^{t} + \boldsymbol{\epsilon}_t \tag{12}$$

$$\mathbf{Z}_{t-r+1}^{t} = A_r^d \mathbf{Z}_{t-r}^{t-1} + \boldsymbol{\eta}_t^d \tag{13}$$

Note that the model of equations 10 and 11 can be found by setting $d = p$, $B_1 = I_d$ and $B_i = 0_d$ for $i = \{2, \ldots, r\}$. The matrices B_i can be used to provide linear dimensionality reduction, and so allow the autoregressive model just as many degrees of freedom as required to model any underlying dynamics. Specifying B_1 but setting the remaining B_is to be zero matrices ensures that y_t has a dependence only on z_t. In this case, the observations are modelled as a corrupted-by-noise version of a lower-dimensional order r autoregressive process. Further specifying B_i for $i = \{2, \ldots, r\}$ gives y_t a dependence also on z_{t-i}.

In practice, estimation for LDMs is largely unconstrained. The state vector is not explicitly divided into separate components, as rd-dimensional $\mathbf{Z}_{t-r+1}^{t}$ is replaced by q-dimensional $\mathbf{x}_t$. Neither the state evolution or observation matrices are forced to place zeros as shown in Equations 10 and 11. However, writing the model in this fashion serves to show the structure which may be contained by the LDM. The addition of observation noise sets the LDM apart from an AR model by making the autoregressive component a hidden process. When combined with dimensionality reduction via the observation process, the effect is to obscure the order of the modelling in the state.

E. The LDM as a model for speech recognition

A simple manner in which to use LDMs for ASR is to train a single model for each phone class in the inventory of a given corpus. This approach is taken in the majority of experiments presented in this paper. This will be referred to as the *LDM-of-phone* formulation, and makes the following assumptions:

- the correlation between consecutive frames within segments is constant.
- segments are not duration-normalized. Therefore, a short instance of a phone is assumed to posses the dynamic characteristics of a portion of a longer example.
- speech parameters can be modelled by a multivariate unimodal Gaussian distribution subject to systematic mean and covariance modification throughout a given segment.

The LDM incorporates the idea of speech being modelled in a domain other than the observations, which are seen as noisy transforms of an underlying process. The internal variables of the hidden state reflect some of the known properties of speech production, where articulators move relatively slowly along constrained trajectories. Depending on the implementation, the state may be reset at the start of each phone or each sentence. A degree of coarticulation modelling is implicit during regions where the state is continuous, as the distribution of the state at time t affects its distribution at time $t + \tau$.

A linear mapping between state and observation processes dictates that points close in state space are also close in observation space. Therefore, trajectories which are continuous in state space are also continuous in observation space. If the hidden state is seen as having articulator-like characteristics, such a constraint is not universally appropriate as small changes in articulatory configuration can sometimes lead to radical changes in the acoustics. Experiments reported in [1] and [2] suggest that whilst linear models give poor descriptions of the dependencies *between* phone segments, behaviour *within* phones can be accounted for by a linear predictor. This is reflected in the LDM-of-phone formulation: within phone models, the output distribution evolves in a linear, continuous fashion. Discontinuities and non-linearities can be incorporated at phone boundaries where resetting the state and switching the observation process parameters H, $\mathbf{v}$ and C results in a sudden shift in acoustic space. By passing state statistics across model boundaries (as discussed below in Section II-F.1), the state process can remain continuous through such shifts.

In our work, the state has the function of giving a compact and dynamic representation of the observed parameters. A number of studies, such as [8]–[12], have attempted to incorporate the relationship between articulation and acoustics through the use of state-space models with non-linear state-observation mappings. In [11], [12], the state is set to model the pole locations of the vocal tract by initializing training using vocal-tract-resonances (VTR), though learning accurate non-linear projections from VTR to acoustic domains proved problematic. In [13], [14], a mixture of linear models is proposed with which to approximate a non-linear relationship whilst retaining many of the useful properties of linear models.

F. Extensions to LDM-of-phone modelling

1) State-passed: A state process which is continuous both within and between phone segments, as found in [9]–[15], represents a step toward the goal of an acoustic model which reflects the properties of speech production. Passing state information across model boundaries offers a degree of contextual modelling, and furthermore gives the possibility of modelling longer range dependencies than contained within phone segments.

We use the terms state-passed and state-reset to differentiate implementations where state statistics are passed across or reset at model boundaries. At the start of each new segment, the prediction of the state distribution, $\mathbf{x}_{t|t-1}$, is required to initialize filtering as described in the Appendix. In the state-reset case, the LDM's learned initial parameters are used, so that $\mathbf{x}_{t|t-1} \sim N(\boldsymbol{\pi}, \Lambda)$. In the state-passed case, predictions are calculated using the posterior state distribution at the preceding time and the current model parameters, so that $\mathbf{x}_{t|t-1} \sim N(F\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{w},\ F\Sigma_{t-1|t-1}F^T + D)$.

Training with a fully continuous state and known segmentation requires a simple modification of the state-reset case as above. However, exact state-passed classification would lead to an exponential increase in computation. Therefore, an approximation is made by introducing pruning at phone boundaries which, whilst not strictly admissible, is believed to be a reasonable approximation and substantially improves efficiency. An alternative approach is to reset the state covariance, but not mean, at boundaries. Some information will still be carried from one phone to the next, but efficient computation can be maintained by pre-computing or caching the 2^{nd} order filter statistics as discussed in Section II-C.

The spectrograms in Figures 1(a), 1(b) and 1(c) give visual evidence of the potential benefits of a state which is continuous across model boundaries. The first spectrogram in the figure shows the original MFCCs corresponding to the utterance 'Do atypical farmers grow oats?'. A Mel-warped frequency scale is used, with regions of high and low energy shown by areas of light and dark shading respectively. The second shows the predicted state mean $\hat{\mathbf{x}}_{t|t-1}$ calculated during the forward filtering pass, projected into the observation space via Equation 2. The time-aligned phone labels determine the parameters used during filtering within each segment. The third spectrogram is derived in the same way as the second, though in this case the state statistics have been passed across phone boundaries during filtering.

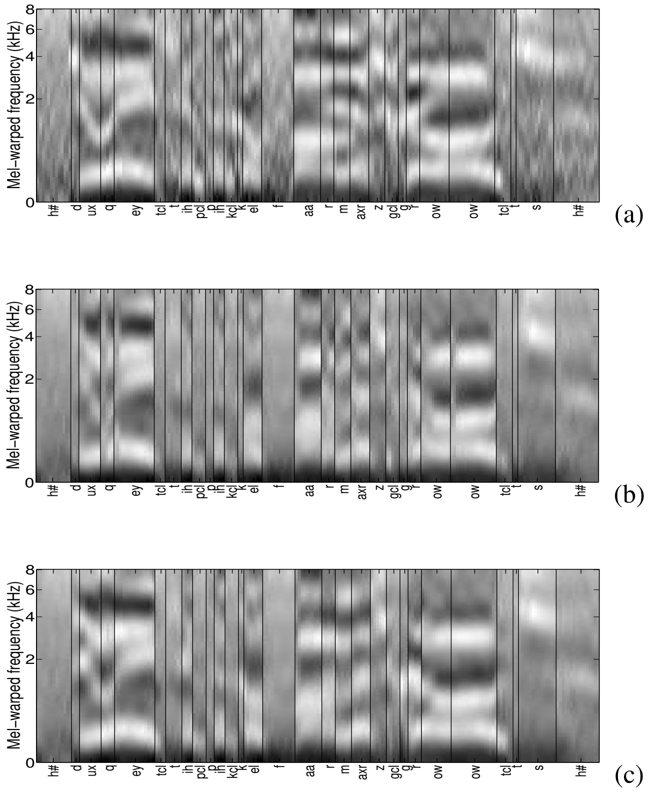

Fig. 1. Spectrograms generated from observed MFCCs, also from state-reset and state-passed LDM predictions for the utterance 'Do atypical farmers grow oats?'

Comparing Figures 1(a) and 1(b), it is apparent that the state-reset LDMs follow many of spectral characteristics of the acoustic signal. However, spectral transitions are subject to strong boundary effects as each new model takes a few frames to find an appropriate location in state-space. The spectrogram in Figure 1(c) demonstrates how a fully continuous state reduces these effects. For example, the discontinuities in the transition of the first formant through the phones `[ux q ey]` early in the utterance (Figure 1(b)) are removed when the state is passed across segment boundaries.

2) Multiple regime models: The multiple regime (MR) formulation splits each phone into a number of regions, each of which is modelled by a separate LDM. Following [2][1], a deterministic mapping dependent on segment duration dictates the sequence of sub-models which are used to generate each phone. The assumptions described in Section II-E then apply within sub-phone regions. The state can be passed or reset between regions as described above.

An MR approach was not taken initially in this work as using deterministic, hand-chosen mappings to partition segments is suboptimal. If such an internal structure is to be used, it should be described by some discrete, hidden random variable, and the transition network learned probabilistically. Furthermore, subdividing segments risks losing their 'segmental' nature. The intention is to model longer sections of speech in which linguistic events occur. Partitioning phone-length segments will produce regions consisting of only a few frames. Modelling may then tend toward the HMM where models describe short, stationary regions of the speech parameters within which there is little requirement for a model of dynamics. Lastly, there are a number of design choices in the LDM-of-phone which warrant investigation prior to modification in this way. However, these models do provide an interesting extension to the LDM-of-phone formulation and warrant investigation as precursors to switching models [16].

III. Classification experiments

A. Data and method

All experimental work uses the TIMIT corpus [17] and follows the standard train/test division, omitting the sa sentences which are the same for each speaker. Validation data comprising 480 sentences was set aside from the train set as in [1]. Both MFCC and PLP features were derived from the acoustic signal, calculated within 25ms windows at a frame rate of 10ms. All experiments use context-independent models.

For classification, a number of EM iterations are performed to estimate parameters using the training data minus the validation set, and the models stored. Classification accuracy on the validation set is used to determine how many iterations the models should be trained for, and to choose a bigram language model scaling factor. Models are then retrained using the combined training and validation data, and the final classification accuracy is for the full test set. The allowable confusions introduced in [18] are used to collapse the 61 phones down to 39 for final evaluation.

Where results are reported as representing statistically significant differences, a paired t-test has been used. The 1344 TIMIT test utterances are split into 24 subsets, accuracy computed within each and a paired t-test performed on the results. A significance level of $p < 0.001$ is used to determine if differences are consistent across the test set.

Recalling that model likelihood is calculated according to Equation 5, initial experiments showed that the state's contribution to the error covariance $\Sigma_{\mathbf{e}_t}$ was detrimental to classification accuracy. The state covariance is normally reset to a value learned during training at the start of each segment, and converges during the first few filter recursions. It was suspected that the resulting fluctuations in the likelihoods computed during segment-initial frames would have most effect on the overall likelihood of shorter phone segments. Figure 2 shows the number of correctly classified tokens with a feature set of MFCCs and energy from 480 TIMIT validation sentences, broken down by segment length. The dashed line shows classification accuracy with $\mathbf{e}_t$ normalized by $\Sigma_{\mathbf{e}_t}$, and the solid line shows the results of the same task where $\Sigma_{\mathbf{e}_t} = C + H\Sigma_{t|t-1}H^T$ has been replaced with $\Sigma'_{\mathbf{e}_t} = C$. For segments over 11 frames, the correct form of likelihood calculation gives a slightly higher accuracy. However, for shorter segments, a modified $\Sigma_{\mathbf{e}_t}$ gives markedly higher classification accuracy. These results are for the 61 phone TIMIT set, prior to the addition of language model, and correspond to overall accuracies of 40.1% and 46.7% using the correct and modified likelihood calculations respectively.

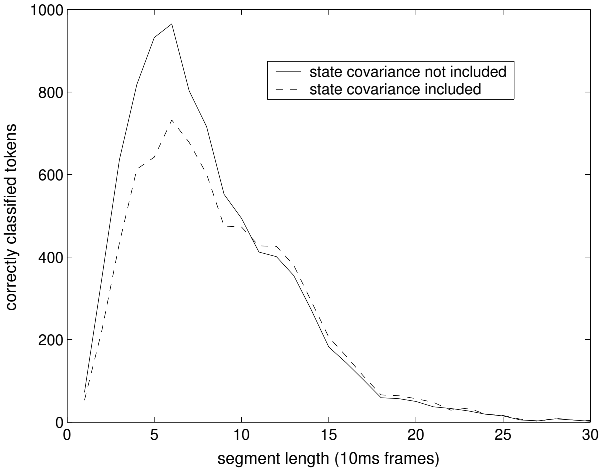

Fig. 2. Phone classification by segment length for 480 TIMIT sentences. The dashed line shows accuracy using the correct form of likelihood calculation, and the solid line accuracy where likelihoods are computed replacing $\Sigma_{\mathbf{e}_t} = C + H\Sigma_{t|t-1}H^T$ with $\Sigma'_{\mathbf{e}_t} = C$.

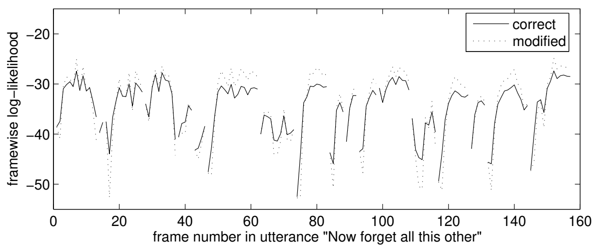

Fig. 3. Framewise likelihood computed using both correct and modified form for the utterance 'Now forget all this other'.

Figure 3 shows framewise log-likelihoods across the utterance 'Now forget all this other' computed using both forms, with the time-aligned phone labels used to determine which model is used within each segment. The figure suggests that in fact, the greatest fluctuations at the start of new segments are found where the modified form is used. The plot of Figure 4 shows the framewise likelihoods averaged over the 61 models for each of the 18 phone segments in the same utterance as in Figure 3. A single standard deviation either side of the mean

[1]This was originally referred to as correlation invariance (CI), renamed here to avoid confusion with the term 'context-independent'.

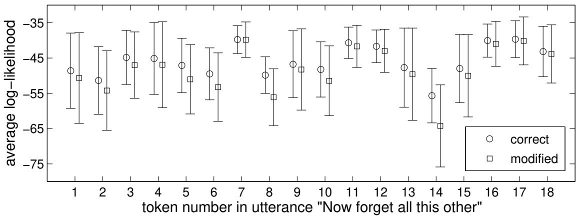

Fig. 4. Mean and single standard deviations of likelihood under all 61 models for each of the 18 tokens in the utterance 'Now forget all this other', computed using both modified and correct form.

is also given. We see from this plot that the average likelihood is consistently higher, and has lower spread, where the correct form has been used.

likelihood computed over	likelihood form: correct	likelihood form: modified
true model only	-34.5	**-34.1**
all models	**-46.3**	-48.2

TABLE II

COMPARISON OF AVERAGE FRAMEWISE LIKELIHOOD COMPUTED USING CORRECT AND MODIFIED FORMS, COMPUTED FOR TRUE MODEL ONLY OR AVERAGED OVER ALL MODELS.

Table II compares true-model and average likelihoods over the full 480 validation utterances. Where the true model according to the labelling is used, the modified form gives a slightly higher framewise likelihood, -34.1 compared with -34.5. However, when averaged over all models, the correct form yields higher likelihood, -46.3 compared to -48.2.

In summary, the modified form yields lower average likelihood with greater spread when all models are considered, though yields higher likelihood when computed according to the true model parameters. The modified form, which was shown above to give improved phone-class discrimination, will be used for the experiments reported in this paper unless otherwise stated.

B. LDM-of-phone results

Table III shows the classification accuracy for LDM-of-phone and static models. The LDM state dimension (shown in parentheses) is chosen based on exhaustive search according to classification accuracy on the validation set. A full set of validation results are given in [1]. The static model is a full covariance Gaussian, which gives a single-state monophone HMM with a unimodal full covariance output distribution. The rationale for choosing such a baseline is to isolate the contribution of the dynamic state. Section IV-C below compares LDM performance with classical HMM baselines in a recognition setting. Table III shows that for each feature set, LDMs give higher accuracy than the static model. These differences are statistically significant, which shows that the dynamic state yields a modest yet consistent performance improvement.

Section II-B above gave a number of variants on a fully specified LDM. The LDM includes three Gaussian covariances: initial state Λ, state noise D, and observation noise

model	PLP, energy	$+\delta$	$+\delta+\delta\delta$
static	66.3%	70.1%	71.3%
LDM	67.8% (10)	71.0% (9)	72.2% (13)
model	MFCC, energy	$+\delta$	$+\delta+\delta\delta$
static	66.4%	70.2%	71.3%
LDM	67.4% (12)	71.3% (12)	72.3% (9)

TABLE III

CLASSIFICATION ACCURACIES FOR LDM AND STATIC MODELS. STATE DIMENSIONS ARE GIVEN IN PARENTHESES.

C. Given the similar performances found using MFCC and PLP features, we choose to explore a number of modelling possibilities using only MFCCs. Table IV gives classification accuracies where some or all of these are constrained to be diagonal rather than full, for both H estimated from the data and set to the identity matrix. These results show that the state covariances D, Λ can be set to be diagonal with minimal impact on accuracy, though full observation noise covariance C is required for best performance. In all cases but one (base features, diagonal D, Λ), including subspace modelling through H leads to accuracy increases over the smoothed Gauss-Markov realization as used in [2].

MFCC classification accuracy				
diagonal	$H=I_p$	MFCC, energy	$+\delta$	$+\delta+\delta\delta$
C, D, Λ	$\surd$	63.7%	67.2%	67.3%
C, D, Λ	$\times$	64.4%	68.6%	68.8%
C	$\surd$	63.8%	67.8%	68.0%
C	$\times$	64.9%	68.4%	69.0%
D, Λ	$\surd$	67.1%	70.5%	71.7%
D, Λ	$\times$	67.0%	71.0%	72.0%
–	$\surd$	67.2%	70.6%	71.7%
–	$\times$	67.4%	71.3%	72.3%

TABLE IV

CLASSIFICATION ACCURACIES WITH STATE, OBSERVATION OR ALL COVARIANCES CONSTRAINED TO BE DIAGONAL, WITH H SPECIFIED OR SET TO THE IDENTITY MATRIX.

C. Multiple regime models

Models of fricatives, silence and oral stop closures are modelled with a single region as the speech signal is considered to be approximately stationary during these sounds. Two regimes corresponding to 'coming in' and 'going out' are used for nasal stops, semivowels and glides, and for affricates which consist of the combination of a stop and a fricative. Vowels, which are subject to strong contextual variation, are split into 3 regimes modelling 'onset', 'steady state' and 'offset'. [19] describes oral stop releases as consisting of 3 distinct regions: a transient, frication at the point of articulation and finally aspiration. Oral stop releases are accordingly split into 3 regimes. All segments are split equally into their chosen number of regions except vowels which are apportioned in the ratio 3:2:3.

Table V shows the classification results for MR static and LDM models with both PLP and MFCC features. An MR static model corresponds to a particular form of HMM in which the state transitions are deterministic given segmental duration, and the output distribution is a unimodal full

PLP classification accuracy			
model	PLP, energy	+ δ	+ δ + $\delta\delta$
LDM-of-phone	67.8%	71.0%	72.2%
MR static	68.6%	73.2%	74.2%
state-reset MR LDM	68.9%	73.5%	74.4%
state-passed MR LDM	70.2%	73.6%	74.5%
MFCC classification accuracy			
model	MFCC, energy	+ δ	+ δ + $\delta\delta$
LDM-of-phone	67.4%	71.3%	72.3%
MR static	68.6%	73.3%	74.3%
state-reset MR LDM	67.9%	73.3%	74.3%
state-passed MR LDM	69.5%	73.7%	74.5%

TABLE V

TIMIT CLASSIFICATION USING MR STATIC AND LDM MODELS WITH RESULTS GIVEN FOR BOTH STATE-RESET AND STATE-PASSED MR LDMS.

covariance Gaussian. These are included to determine if the dynamic portion of the model still contributes under this implementation. The MR static models outperform LDM-of-phone models, though the (dynamic) state-passed MR LDMs give the highest accuracies for all feature sets. However, it is only where no δ or $\delta\delta$s are included that the MR LDMs give a statistically significant increase in accuracy over the static models, suggesting that the benefit of a dynamic state is reduced when segments are divided in this way.

D. State-passed

The classification results of this section are for the TIMIT core (rather than full) test set with an MFCC parameterization, and the language models are the backed-off bigrams as used in the recognition experiments of Section IV. Otherwise, the classification procedure remains as described above. Standard state-reset classification with these different language models and test set using MFCCs and energy gives an accuracy of 67.4%, identical to the result presented in Table III, and provides a baseline for the following experiments. The LDMs were initialized identically to those used in producing the baseline result and trained from scratch with both state means and covariances passed over segment boundaries.

implementation: training	implementation: testing	classification accuracy	speed ($\times$ real time)
state reset	state reset	67.4%	0.25
state passed	state reset	66.0%	0.25
state passed	state passed	67.0%	75
state passed	state passed, correct likelihood	66.7%	75

TABLE VI

STATE-PASSED MFCC AND ENERGY CLASSIFICATION ACCURACIES.

The state-passed results of Table VI were found with both the state mean and covariance passed across phone boundaries, which results in decoding at around 75 times slower than real-time on a 2.4GHz Pentium P4 processor, which compares to 4 times faster than real time for the state-reset models.

These results show that the highest accuracy of 67.4% is given by the baseline result where the state is reset at the start

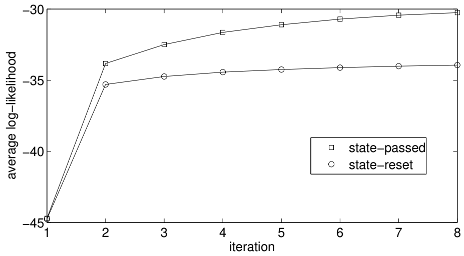

Fig. 5. Framewise likelihood during state-passed and -reset training.

of each new segment in both training and testing. State-passed training followed by state-reset testing results in an accuracy of 66.0%, though using these same models and passing the state between segments during testing gives the improved result of 67.0%. The latter is close to the baseline, and suggests that a mismatch between training and testing causes a reduction in performance.

The state-passed formulation yields higher likelihoods during training than state-reset models, as shown in Figure 5. Similarly, the model fit on unseen data is improved, with average framewise validation likelihoods of -32.6 and -34.1 for the state-passed and state-reset respectively. This behaviour is shown pictorially in Figure 6, which presents state-passed and state-reset framewise likelihoods through the utterance 'Now forget all this other'. The sudden decreases in the state-reset likelihood correspond to segment boundaries. The

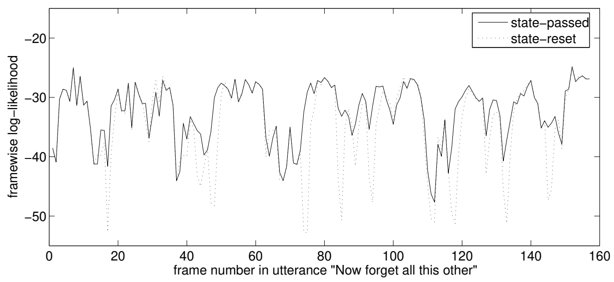

Fig. 6. Framewise state-passed and state-reset likelihood for the utterance 'Now forget all this other'.

results of Table VI show that in this case, improving the generative model and increasing model likelihood does not lead to improved discrimination.

Section III-A above showed that a modified likelihood calculation gave higher classification accuracies for shorter phones. With the state continuous across entire utterances, there may be an advantage by re-including the contribution of state covariance in normalizing the prediction errors. The last result of Table VI shows that in fact this causes a slight reduction in performance, giving an accuracy of 66.7%. We find similar behaviour to that described in Section III-A above, with the modified form giving higher likelihood when evaluated using the true model, -32.6 compared to -33.4.

IV. CONTINUOUS SPEECH RECOGNITION

Letting $\mathcal{Y} = \mathbf{y}_1^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ denote an N-length observation sequence, and $\mathcal{W} = w_1^{j_N} = \{w_1, \ldots, w_{j_N}\}$ denote a corresponding word sequence, decoding can be defined as finding the maximum a posteriori (MAP) probability of words $\mathcal{W}$ given observations $\mathcal{Y}$:

$$\mathcal{W}^* = \underset{\mathcal{W}}{\operatorname{argmax}} P(\mathcal{W}|\mathcal{Y}) \tag{14}$$

Bayes rule is used to decompose Equation 14 in terms of a sequence of sub-word models accounting for the full observation sequence $\mathcal{M} = m_1^{k_N} = \{m_1, \ldots, m_{k_N}\}$, and then the Viterbi approximation [20] is applied to give:

$$\mathcal{W}^* \simeq \underset{\mathcal{W}}{\operatorname{argmax}} \left\{ P(\mathcal{W}) \max_{\mathcal{M}} p(\mathcal{Y}|\mathcal{M}) P(\mathcal{M}|\mathcal{W}) \right\} \tag{15}$$

By searching for $\mathcal{W}^*$ whilst taking the maximum likelihood model sequence rather than summing over all possibilities gives a significant increase in computational efficiency.

Decoding for ASR can be defined in terms of the search ordering, with a common strategy being time-synchronous forward dynamic programming (Viterbi decoding [21]), where all hypotheses at a given time are evaluated before the search proceeds to the next time. An alternative approach, time-asynchronous $A*$ search, is considered in this work.

A. A^* search

During best-first search, such as A^* stack decoding, the search order is determined by an evaluation function, h^*. At each cycle, the current most promising partial hypothesis is chosen for extension. For an N-length observation sequence $\mathbf{y}_1^N$, we define the evaluation function h_t^* for a hypothesis with a path ending at time $t \leq N$ as being composed of two parts:

$$h_t^* = f_t + g_t^* \tag{16}$$

The first is the detailed match f_t, and contains the likelihood of observations $\mathbf{y}_1^t$ under the acoustic, language and lexical models:

$$f_t = p(\mathbf{y}_1^t | m_1^{k_t}) \, P(m_1^{k_t} | w_1^{j_t}) \, P(w_1^{j_t}) \tag{17}$$

where $w_1^{j_t} = \{w_1, \ldots, w_{j_t}\}$ and $m_1^{k_t} = \{m_1, \ldots, m_{k_t}\}$ represent the hypothesized sequence of words and sub-word models respectively. The second is the lookahead function g_t^*, which holds an estimated likelihood cost to account for the remainder of the observations $\mathbf{y}_{t+1}^N$. Using an evaluation function composed of detailed match and lookahead function is key to time-asynchronous search, as it allows the comparison of hypotheses of differing lengths. Such a search strategy is admissible as long as g_t^* gives an *upper bound* on the acoustic likelihood [22].

The efficiency or otherwise of an A^* search is largely determined by g_t^*. Whilst the estimate of the remaining likelihood must be optimistic, over-estimates can lead to a vastly increased search space. Exact computation of the remaining cost (which would involve summation over all possible word sequences) is usually considered impractical and approximations are made using heuristic approaches [23], [24].

B. Decoding for linear dynamic models

Pre-compiling a transition network according to the language, lexical and acoustic models [21] is a natural approach for decoding with HMMs since the models are discrete and finite-state right down to state level. By contrast, LDMs give models of variable-length segments, and the continuous-valued state means that the Viterbi criterion, integral to efficient time-synchronous search, is never admissible on a frame-wise basis (though dependent on implementation, may be at the ends of phones or words). LDMs also require increased computation over frame-based models: with $p(\mathbf{y}_t^{t+\tau}|m_{k_t}^{k_{t+\tau}})$ already calculated, extending acoustic matching by a single frame is straightforward. Since

$$p(\mathbf{y}_t^{t+\tau+1}|m_{k_t}^{k_{t+\tau+1}}) = p(\mathbf{e}_{t+\tau+1}|m_{k+\tau+1}) \, p(\mathbf{y}_t^{t+\tau}|m_{k_t}^{k_{t+\tau}}) \tag{18}$$

all that is required is a further forward Kalman recursion. However, $p(\mathbf{y}_{t-1}^{t+\tau}|m_{k_t-1}^{k_{t+\tau}})$ cannot be calculated in such an efficient manner. The state's initial value influences the subsequent forward filtered state statistics, and hence any likelihood computation. Therefore, a separate Kalman filter must be run to compute the model likelihoods for each candidate start time.

In [11], a time-synchronous strategy was proposed for decoding a non-linear state-space model, in which a stack structure maintains a set of candidate paths for each phone node at each time. When inserting a hypothesis onto a stack, the Viterbi approximation is made on paths which are close together in state space. Another approach to decoding with a non-linear state-space model is given by [15], in which the continuous hidden space is discretized to validate a time-synchronous search.

In the current study, we propose that a time-asynchronous strategy is well suited to decoding with continuous state models: with no requirement that the Viterbi criterion be applied at a frame level, the decoder is flexible to the choice of acoustic model, and dependent on the accuracy of lookahead, such search can be efficient in only exploring likely paths. This approach also has the advantage that, unlike Viterbi decoding, the language model is not used to generate each new hypothesis. Decoupling the language model and hypothesis generation in this way means that the decoder can be designed in a modular fashion, with the only restriction on the language model being that it must be able to assign probabilities to initial portions of sentences consisting of whole words.

1) Implementation of the core acoustic matching: In practice, the detailed match of Equation 17 is computed as a weighted sum of log probabilities with the addition of a word insertion penalty, as in [21], and a log-Gaussian phone duration distribution estimated on the training set [1]. For each hypothesis which is popped, decoding involves a depth-first walk over a tree-shaped lexicon as described in [25]. Acoustic matching takes place in a grid structure with time increasing down the y-axis and a column for each phone model to be added. Hypotheses are extended by whole words, one phone at a time, with an optional silence added at the start of each new word. Phones are added as follows: for each candidate start time, a Kalman filter is run to compute the acoustic likelihoods for a range of end times. These are combined with the other elements of the detailed match and the previous path likelihood, then entered in the appropriate rows of the following column. If the state is reset between phone models, the Viterbi approximation is applied where multiple paths meet.

2) Computing the lookahead function g_t^*: The decoding experiments presented below consist of phone recognition of isolated sentences. For every utterance to be decoded, a Kalman filter is run across the full observation sequence for each of the 61 TIMIT phone models. The frame-wise likelihoods under each model are ranked, then an average taken across the top n. These averages are then summed so as to produce a reverse accumulation of framewise likelihood. The experiments reported in this work use $n = 1$ which provides a practical upper bound on the remaining likelihood, though ignoring language model and durational constraints means that the lookahead is over-estimated.

3) Pruning: Beam pruning, which is dependent on calculated likelihood f_t rather than lookahead g_t^*, is implemented both in the grid and on the stack, with $\Delta^{(grid)}$ and $\Delta^{(stack)}$ denoting the grid and stack beam widths respectively. As each word is added to the most recently popped hypothesis, an upper bound $\Psi_t^{(grid)}$ is maintained on the likelihoods in the grid. Any paths for which $f_t < \Psi_t^{(grid)} - \Delta^{(grid)}$ are discarded. Similarly a stack upper bound $\Psi_t^{(stack)}$ is maintained and paths for which $f_t > \Psi_t^{(stack)} - \Delta^{(stack)}$ are removed.

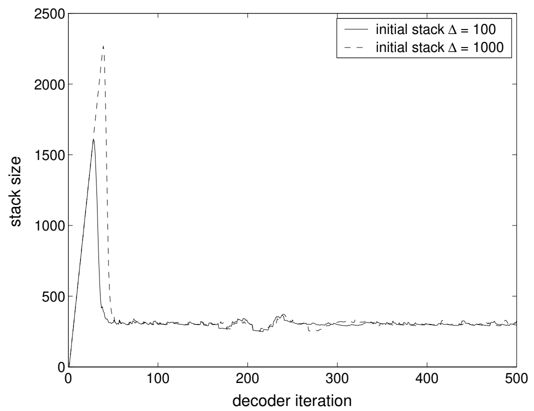

Fig. 7. The adaptive pruning adjusts the stack beam width $\Delta^{(stack)}$ at each iteration to maintain a roughly constant number of stack items. This figure shows the first 500 cycles of the decoder for large and small initial $\Delta^{(stack)}$ and a target stack size of 300.

In practice, finding suitable values of $\Delta^{(stack)}$ proved problematic: tight thresholds could result in pruning away all hypotheses, whilst larger values of $\Delta^{(stack)}$ resulted in a stack which grew to a size which significantly increased decoding time. An adaptive pruning scheme was developed in which a target stack size is chosen and at each iteration, the stack beam width is updated dependent on the current stack size. Relation 19 gives the factor by which the stack beam width $\Delta^{(stack)}$ is adjusted:

$$\Delta^{(stack)\prime} = \left(1 - \alpha \log \frac{\text{stack size}}{\text{target stack size}}\right) \Delta^{(stack)} \quad (19)$$

The tuning parameter α dictates how rapidly the beam width can change. A value of $\alpha = 0.1$ was found to be suitable. Figure 7 illustrates the adaptive pruning scheme maintaining a stack of 300 partial hypotheses during decoding. Through 1000 decoder cycles, the stack size increases initially, but is soon capped and then remains fairly constant.

We make the following observations of the effect of pruning on the experiments reported below: the number of partial hypotheses kept on the stack has a significant effect on the speed at which the decoder runs. The local beam width $\Delta^{(grid)}$ affects accuracy but has little effect on time to decode each utterance. For smaller stack sizes, pruning in the grid can be advantageous to recognition performance, as the highest accuracies do not correspond to the largest grid beam widths. Such pruning has the effect of removing unlikely hypotheses at the first possible opportunity. In phone recognition experiments, pruning in the grid is found to make little difference to decoding speed, however the local beam width $\Delta^{(grid)}$ may have a more significant effect on the decoder speed for word recognition in which multiple phone models are evaluated in the grid.

4) Efficient implementation: Pre-computation of state statistics was discussed in Section II-C, and can be used during recognition with correspondingly significant savings. Since the state is reset between phones, computation can be further reduced by caching acoustic likelihoods.

C. Experimental results

The LDM-of-phone recognition experiments use the model sets which produced the classification results of Table III. The various scaling factors and word insertion penalty were chosen on the validation set. A number of HMM baseline results have been prepared using HTK [21], with models trained, validated and tested on identical data and language model to that used in the LDM experiments. The HMMs were initialized with uniform segmentation and Viterbi training, then Baum-Welch to convergence with fixed segment label times followed by full embedded training.

All results are given on the NIST core test set, and use the same levels of pruning as applied during validation. Decoding uses a set of 61 models, though in reporting results the phone set is collapsed down to 39.

phone	PLP, energy	$+\delta$	$+\delta+\delta\delta$
% correct	58.7%	62.9%	62.0%
% accuracy	55.2%	58.5%	58.5%
phone	MFCC, energy	$+\delta$	$+\delta+\delta\delta$
% correct	54.0%	60.0%	63.9%
% accuracy	51.1%	57.2%	60.3%

TABLE VII

TIMIT NIST CORE TEST-SET LDM RECOGNITION ACCURACIES.

LDM recognition results are given in Table VII and show that MFCCs with energy, δs and $\delta\delta$s appended give overall highest accuracy of 60.3%. The majority of the confusions are between vowels, with phones commonly misclassified as [`ix`] [`ax`] or [`ao`]. Also, errors appear in making voicing decisions, with [`b, d`] being frequently recognized as their voiceless counterparts [`p, t`].

HMM states	HMM mixtures	HMM covariance	accuracy	params	speed (× real time)
1	1	diagonal	51.4%	4.8K	0.045
3	1	diagonal	58.9%	14.5K	0.041
1	1	full	58.1%	50.0K	0.064
3	1	full	65.6%	150.1K	0.14
1	20	diagonal	64.2%	96.4K	0.14
3	20	diagonal	69.4%	289.1K	0.36
1	2	full	60.3%	100.0K	0.11

LDM	accuracy	params	speed (× real time)
	60.3%	82.8K	26

TABLE VIII

TIMIT NIST CORE TEST-SET LDM AND HMM RECOGNITION ACCURACIES FOR MFCCS WITH ENERGY, δS AND $\delta\delta$S. NUMBERS OF FREE PARAMETERS AND DECODING SPEEDS ARE ALSO GIVEN.

As previously, we take a full covariance Gaussian as the static model baseline, which equates to a single state monophone HMM with unimodal full covariance Gaussian output distribution. The recognition accuracy of 58.1% is given in row 3 of Table VIII, and represents a statistically significant reduction on the LDM accuracy of 60.3%. A number of other HMM results are given in Table VIII, along with numbers of free parameters and decoding speeds. The classical TIMIT 3-state HMM baseline gives an accuracy of 69.4%, substantially higher than found for LDMs, and uses over 3 times more free parameters.

V. DISCUSSION

LDMs have been proposed for ASR under a variety of implementations [2], [13], [14], [16]. This work has examined the core assumptions made in using such a model, along with the associated implementational issues, and demonstrated that the addition of a hidden dynamic state leads to improved accuracy. Relative error reductions of 3.5% and 5.5% were found using LDMs compared to otherwise equivalent static models on TIMIT phone classification and recognition tasks. However, in light of the extra computation, these gains do not make a strong case for adoption of these models.

One possible conclusion which may be drawn is that a first order linear state process is inappropriate for modelling of speech parameters. This was the finding of [16], though this study did not make additions which were found to be beneficial, such as full noise covariances and a modified likelihood calculation. Alternatively, the true benefits of the state process might be found with an alternative implementation.

Given that the state is used to model underlying dynamics from segments which are subject to variation both between and within speakers, the state-observation mapping should be tuned to minimize these effects and produce consistent underlying behaviour. One possibility is to employ a non-linear mapping between state and observations. The linear Gaussian assumptions made by the Kalman filter do not hold in this case, and so [11], [12] apply an extended Kalman filter (EKF), in which the non-linearity is approximated by constructing locally linear state and observation equations. This assumes that the errors on truncating a Taylor series to first order will be negligible, which in practice may not be valid. The problems inherent in the EKF may also be associated with the practical difficulty in training a non-linear state-observation model as discussed in [13], and an alternative filtering approach, such as proposed by [26], may prove rewarding.

Alternatively, [13], [14] propose a switching observation process designed to approximate a non-linear mapping, whilst retaining many of the useful properties of a linear Gaussian models. The maximum number of mixture components used in the observation processes was 4, which is small compared to the number of components which may be employed in an HMM-based system. Increasing the number of components may be beneficial. Another possible approach for reducing the effects of inter-speaker variability is through adaptation of the observation process using a form of maximum likelihood linear regression (MLLR) [27], which could be implemented whilst retaining the linear-Gaussian properties of the LDM.

The use of context-dependent models has become standard in HMM systems, and may also be applied to LDMs. As with HMMs, parameter tying will be required to alleviate problems of data sparsity, though the LDM offers a multitude of ways in which this may be implemented as models may share some or all parameters. For example, models within the same triphone cluster might share observation but not state parameters, or in the case of a switching observation process as discussed above, models might share noise models and differ in their observation matrices H.

Results in Section III-B showed that a full covariance observation noise model gave an accuracy increase over diagonal models. The increase in computation is marginal, as the Kalman filter recursions yield full prediction error covariance matrices, though the number of free parameters is increased by $\frac{1}{2}p(p-1)$ per Gaussian, where p is the observation dimension. Modelling the precision (inverse covariance) matrix as in [28], leads to a factorization which separates a full covariance matrix into rotation and magnitude components. This approach facilitates learning of covariance matrices which are between diagonal and full, and also offers flexible parameter tying schemes where a covariance matrices share a common rotation component, but have unique magnitudes. Both of these may prove useful in ensuring robust estimation whilst increasing the number of models, whether through the introduction of context-dependent or mixture models.

The findings of Section III-D were that passing state statistics across segment boundaries led to decreases in classification accuracy. However, the success of such an approach might depend on occasional resetting of the state as there is a great deal of variation in the nature of the transitions between segments. In some cases, these will be highly non-linear, such as found between the closure and release portions of plosives. At other times, the segmental boundaries are less well-defined, such as in the transition between a vowel and a nasal stop. It may be that resetting the state for the first of these examples would act as a regularizer for the state covariances, but allowing passing of the state in the second would enhance modelling. Building an understanding of the manner in which this choice interacts with the ability to discriminate phone classes would be non-trivial, though desirable given the intuitive appeal of such a model for ASR.

ACKNOWLEDGEMENTS

The authors would like to thank Korin Richmond for numerous constructive discussions, John Bridle and Chris Williams for their comments on this work when it was in the form of a PhD thesis, also the anonymous reviewers for their constructive feedback.

APPENDIX
INFERENCE

The Kalman filter equations are as below, and initialized by setting $\hat{\mathbf{x}}_{1|0}$ and $\Sigma_{1|0}$ to the initial state mean and covariance.

$$\begin{aligned}
\hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + K_t \mathbf{e}_t \\
\hat{\mathbf{x}}_{t|t-1} &= F\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{w} \\
\mathbf{e}_t &= \mathbf{y}_t - \hat{\mathbf{y}}_t \;=\; \mathbf{y}_t - \mathbf{v} - H\hat{\mathbf{x}}_{t|t-1} \\
K_t &= \Sigma_{t|t-1} H^T \Sigma_{\mathbf{e}_t}^{-1} \\
\Sigma_{\mathbf{e}_t} &= H\Sigma_{t|t-1}H^T + C \\
\Sigma_{t|t} &= \Sigma_{t|t-1} - K_t \Sigma_{\mathbf{e}_t} K_t^T \\
\Sigma_{t,t-1|t} &= (I - K_t H) F \Sigma_{t-1,t-1} \\
\Sigma_{t|t-1} &= F\Sigma_{t-1|t-1}F^T + D
\end{aligned}$$

A backward pass with the RTS smoother gives complete-data estimates:

$$\begin{aligned}
\hat{\mathbf{x}}_{t-1|N} &= \hat{\mathbf{x}}_{t-1|t-1} + A_t(\hat{\mathbf{x}}_{t|N} - \hat{\mathbf{x}}_{t|t-1}) \\
\Sigma_{t-1|N} &= \Sigma_{t-1|t-1} + A_t(\Sigma_{t|N} - \Sigma_{t|t-1})A_t^T \\
A_t &= \Sigma_{t-1|t-1} F^T \Sigma_{t|t-1}^{-1} \\
\Sigma_{t,t-1|N} &= \Sigma_{t,t-1|t} + (\Sigma_{t|N} - \Sigma_{t|t})\Sigma_{t|t}^{-1}\Sigma_{t,t-1|t}
\end{aligned}$$

REFERENCES

[1] J. Frankel, "Linear dynamic models for automatic speech recognition," Ph.D. dissertation, The Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK, 2003.
[2] V. Digalakis, "Segment-based stochastic models of spectral dynamics for continuous speech recognition," Ph.D. dissertation, Boston University Graduate School, Boston, MA, 1992.
[3] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models." *Neural Computation*, vol. 11, no. 2, 1999.
[4] A. Rosti and M. Gales, "Generalised linear Gaussian models," Cambridge University Engineering, Tech. Rep. CUED/F-INFENG/TR.420, 2001.
[5] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–44, March 1960.
[6] H. E. Rauch, "Solutions to the linear smoothing problem." *IEEE Transactions on Automatic Control*, vol. 8, pp. 371–372, 1963.
[7] V. Digalakis, J. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 4, pp. 431–442, October 1993.
[8] K. Iso, "Speech recognition using dynamical model of speech production," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Minneapolis, MN, 1993, pp. 283–286.
[9] J. Bridle, L. Deng, J. Picone, H. Richards, J. Ma, T. Kamm, M. Schuster, S. Pike, and R. Reagan, "An investigation of segmental hidden dynamic models of speech coarticulation for automatic speech recognition," Workshop on Language Engineering, Center for Language and Speech Processing at Johns Hopkins University, Tech. Rep., 1998.
[10] J. Picone, S. Pike, R. Regan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, and M. Schuster, "Initial evaluation of hidden dynamic models on conversational speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Phoenix, AZ, 1999, pp. 109–112.
[11] J. Ma and L. Deng, "A path-stack algorithm for optimizing dynamic regimes in a statistical hidden dynamic model of speech," *Computer Speech and Language*, vol. 14, no. 2, pp. 101–114, 2000.
[12] L. Deng and J. Ma, "Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics," *The Journal of the Acoustical Society of America*, vol. 108, no. 6, pp. 3036–3048, December 2000.
[13] J. Ma and L. Deng, "Target-directed mixture linear dynamic models for spontaneous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 1, pp. 47–58, January 2004.
[14] J. Ma and L. Deng, "A mixed-level switching dynamic system for continuous speech recognition," *Computer Speech and Language*, vol. 18, pp. 49–65, 2004.
[15] F. Seide, J. Zhou, and L. Deng, "Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM — MAP decoding and evaluation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Hong Kong, China, 2003, pp. 748–751.
[16] A.-V. I. Rosti, "Linear Gaussian models for speech recognition," Ph.D. dissertation, Cambridge University Engineering Department, Cambridge, UK, 2004.
[17] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: design and analysis of the acoustic-phonetic corpus." in *Proc. Speech Recognition Workshop*, Palo Alto, CA., February 1986, pp. 100–109.
[18] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, November 1989.
[19] K. Stevens, *Acoustic Phonetics*. Cambridge, Mass.: The MIT Press, 1998.
[20] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm." *IEEE Transactions on Information Processing*, vol. 13, pp. 260–269, 1967.
[21] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2)*, Cambridge University Engineering Department, Cambridge, UK, 2002.
[22] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. MacGraw-Hill (New York NY), 1971.
[23] D. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model." in *Proc. ICASSP*, vol. 1, San Francisco, 1992., pp. 25–28.
[24] P. Kenny, R. Hollan, V. Gupta, M. Lennig, P. Mermelstein, and D. O'Shaughnessy, "A*-admissible heuristics for rapid lexical access." *IEEE Transactions on Speech and Audio Processing.*, vol. 1, no. 1, pp. 49–58, January 1993.
[25] S. Renals and M. Hochberg, "Decoder technology for connectionist large vocabulary speech recognition," Dept. of Computer Science, University of Sheffield. Dept. of Computer Science, Tech. Rep. +CS-95-17, 1995.
[26] S. Julier and J. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Dept. of Engineering Science, University of Oxford, Tech. Rep., 1996.
[27] M. Gales and P. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer, Speech and Language*, vol. 10, pp. 249–264, 1996.
[28] J. Bilmes, "Factored sparse inverse covariance matrices," in *Proc. ICASSP*, 2000.

Joe Frankel (M'05) graduated with first class honours in Mathematics and Statistics from Edinburgh University in 1998. A background in probabilistic modelling paved the way for a PhD place at Centre for Speech Technology Research (CSTR). By the time he had completed his PhD in summer 2003, he had gained a strong interest in the application of machine learning techniques to automatic speech recognition. He is currently a research fellow at CSTR.

Simon King (M'95) has been involved in speech technology since 1992, and has been at the Centre for Speech Technology Research (CSTR) since 1993. His interests include speech synthesis, with involvement in the Festival TTS system, as well as model-based articulatory synthesis, information extraction and singing synthesis. He has recently been awarded an advanced fellowship by the EPSRC to pursue research into novel acoustic models for automatic speech recognition.