CAMBRIDGE UNIVERSITY

ENGINEERING DEPARTMENT

SWITCHING LINEAR DYNAMICAL SYSTEMS FOR SPEECH RECOGNITION

A-V.I. Rosti & M.J.F. Gales

CUED/F-INFENG/TR.461

December 12, 2003

Cambridge University Engineering Department
Trumpington Street
Cambridge. CB2 1PZ
UK

E-mail: {avir2, mjfg}@eng.cam.ac.uk

Abstract

This paper describes the application of Rao-Blackwellised Gibbs sampling (RBGS) to speech recognition using switching linear dynamical systems (SLDSs) as the acoustic model. The SLDS is a hybrid of standard hidden Markov models (HMMs) and linear dynamical systems. It is an extension of the stochastic segment model (SSM) where segments are assumed independent. SLDSs explicitly take into account the strong co-articulation present in speech using a Gauss-Markov process in a low dimensional, latent, state space. Unfortunately, inference in SLDS is intractable unless the discrete state sequence is known. RBGS is one approach that may be applied for both improved training and decoding for this form of intractable model. The theory of SLDS and RBGS is described, along with an efficient proposal distribution. The performance of the SLDS and SSM using RBGS for training and inference is evaluated on the ARPA Resource Management task.

1 Introduction

Currently the most popular acoustic model for speech recognition is the hidden Markov model (HMM). However, HMMs are based on a series of assumptions some of which are known to be poor for speech signals. In particular successive speech frames are assumed to be conditionally independent given the state that generated them. To overcome this limitation, segment models [19] have been proposed. These model whole segments of frames rather than individual frames. One example is the stochastic segment model (SSM) [7]. This uses a standard linear dynamical system (LDS) to model the sequence of observations within a segment.

The LDS belongs to the subset of state space models called linear Gaussian models [24]. In the LDS the dynamics are modelled by a linear first-order Gauss-Markov process in some low dimensional state space. This continuous state space allows more information about the time evolution to be obtained than the discrete state space used in HMMs. The observed feature vector is a noise corrupted linear transformation of this state vector. An alternative argument in favour of LDSs might be based on a state space representation of the moving articulators to observation mapping. However, this argument is rather weak since both the articulator movement and articulator to acoustic observation mappings are nonlinear [3]. The parameters of a standard LDS can be optimised using the expectation maximisation (EM) algorithm [6]. Traditionally the expectation step is carried out using the Kalman filter [14] and Rauch-Tung-Striebel (RTS) [22] smoother consecutively.

For the stochastic segment model, segments are assumed to be independent. The state vectors are thus initialised at the segment boundaries using the initial state vector distribution in the LDS. This is a poor assumption for speech due to the co-articulation between the modelling units. The switching linear dynamical systems (SLDSs) are proposed as more appropriate acoustic models. In SLDS the posterior distribution of the state vector is propagated over the segment boundaries. Unfortunately, exact inference for SLDS is intractable, as the likelihood at any time depends on the entire discrete state sequence. Therefore, parameter optimisation using the standard EM algorithm, and inference using the Viterbi algorithm, is not feasible. Some approximate methods for inference have previously been proposed in the machine learning literature. These include an approximate Viterbi algorithm [20], a generalised pseudo Bayesian algorithm [1, 18], an expectation maximisation algorithm [12] and structured variational approximations [21]. Recently in the speech literature approximate decoding schemes for similar state space models have also been investigated. The interacting multiple model approximation in [1] was investigated for both inference and training in [15] and a related state space quantised version in [30]. Unfortunately, these deterministic approximations cannot easily be modified to employ mixture models in the noise processes.

The approximate inference scheme examined in this paper is based on Markov chain Monte Carlo methods [23]. Rather than modifying the model structure, or removing dependencies in the state history, a sampling approach is adopted. Furthermore for efficiency, rather than sampling from the joint discrete and continuous state space, algorithm based on Rao-Blackwellisation is used [5, 8]. Here discrete samples are drawn from the *proposal distribution* for the discrete state space. The continuous state space statistics are then computed using the standard methods given the estimated discrete state. The main problem is to obtain an efficient form for the proposal distribution. For the SLDS the complexity of the proposal distribution can be shown to be $O(T)$ per iteration (sequence of samples). Also, handling non-Gaussian noise processes, such as Gaussian mixture models, may easily be included in the Gibbs sampling algorithm.

Rao-Blackwellised Gibbs sampling has previously been applied for low dimensional regression problem in [5] and tracking of moving target in [8]. Here, the state and observation space dimensionalities range from 2 to 4 and the number of discrete states is 3 at most. In this paper the RBGS and methods to apply it in speech recognition are presented. In speech recognition, the dimensionalities of the state and observation space typically range between 13 and 39, and the number of discrete states in the thousands. Thus, the size of the state space the samples are drawn from is dramatically larger than in the previous applications. Also the dynamic range of the continuous space statistics is much larger. Due to the multi-modal feature vector distributions Gaussian mixture models are used as the observation noise process. In addition to the inference algorithm a parameter optimisation scheme based on maximum likelihood state sequence is proposed.

This paper is organised as follows. The following section introduces the models in a generative state space model framework. The inference and training methods are also discussed. Section 3 presents the approximate inference and training schemes based on Rao-Blackwellised Gibbs sampling. Also, the application for speech recognition is discussed. Experiments using ARPA Resource Management are reviewed in Section 4. Finally, Section 5 concludes the paper.

1.1 Notation

In this paper, bold capital letters are used to denote matrices, e.g. $\boldsymbol{A}$, bold letters refer to vectors, e.g. $\boldsymbol{a}$, and plain letters represent scalars, e.g. c. All vectors are column vectors unless otherwise stated. Prime is used to denote the transpose of a matrix or a vector, e.g. $\boldsymbol{A}', \boldsymbol{a}'$. The determinant of a matrix is denoted by $|\boldsymbol{A}|$. Gaussian distributed vectors, e.g. $\boldsymbol{x}$ with mean vector, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, are denoted by $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The likelihood of a vector $\boldsymbol{z}$ being generated by the above Gaussian; i.e., the Gaussian evaluated at the point $\boldsymbol{z}$, is represented as $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The lower case letter p is used to represent a continuous distribution, whereas a capital letter P is used to denote a probability mass function of a discrete variable. The probability that a discrete random variable, q, equals j is denoted by $P(q = j)$. If the discrete random variable, such as q_t, explicitly has as a value and it is not practical to enumerate all the events, the probability of the event is denoted by $P(q_t)$.

2 Switching Linear Dynamical Systems

The models presented in this paper can be viewed as general state space models with N_s hidden discrete Markov states. In speech recognition applications the discrete state normally represents a phone, or context dependent phone. A hidden k-dimensional state vector, $\boldsymbol{x}_t$, is generated by the state evolution process. This continuous state vector can be viewed as an intermediate time evolving representation of the observation vectors. Every time instant, a p-dimensional observation vector, $\boldsymbol{o}_t$, is generated by a linear observation process. Various forms of state to observation representations are possible [24]. For all the models in this paper, the observation process is based on factor analysis.

2.1 Generative Models

The simplest state evolution process is a discrete state dependent vector of Gaussian distributed noise. This model is called the factor analysed HMM (FAHMM) [25]. Instead of generating the observation vectors, the underlying HMM generates vectors of latent variables for the factor analysis observation process. The generative model of FAHMM can be written as follows

$$\boxed{\begin{aligned} q_t &\sim P(q_t|q_{t-1}) \\ \boldsymbol{x}_t &= \boldsymbol{w}_{q_t} \\ \boldsymbol{o}_t &= \boldsymbol{C}_{q_t}\boldsymbol{x}_t + \boldsymbol{v}_{q_t} \end{aligned}}$$

where the discrete state sequence, $Q = \{q_1, q_2, \dots, q_T\}$, is defined by the initial state probability, $\pi_j = P(q_1 = j)$, and a set of transition probabilities, $a_{ij} = P(q_t = j|q_{t-1} = i)$. The state noise, $\boldsymbol{w}_j$, and the observation noise, $\boldsymbol{v}_j$, are distributed according to Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}_j^{(x)}, \boldsymbol{\Sigma}_j^{(x)})$ and $\mathcal{N}(\boldsymbol{\mu}_j^{(o)}, \boldsymbol{\Sigma}_j^{(o)})$, respectively. The observation matrices, $\boldsymbol{C}_j$, depend on the discrete state but any parameter in FAHMM can be arbitrarily tied. For the FAHMM it is also possible to use Gaussian mixture models (GMMs) for the state and observation noise sources. The state evolution process can be described as piece-wise constant. The observation process defines a time evolving factor analysis model where a state vector is a collection of factors, an observation matrix is the factor loading matrix and an observation noise is the factor analysis error term.

In SLDS the state vectors evolve according to a first-order Gauss-Markov process. The generative model for a SLDS can be expressed as

$$\boxed{\begin{aligned} q_t &\sim P(q_t|q_{t-1}) \\ \boldsymbol{x}_t &= \boldsymbol{A}_{q_t}\boldsymbol{x}_{t-1} + \boldsymbol{w}_{q_t} \\ \boldsymbol{o}_t &= \boldsymbol{C}_{q_t}\boldsymbol{x}_t + \boldsymbol{v}_{q_t} \end{aligned}}$$

where both the state transition matrices, $\boldsymbol{A}_j$, and the observation matrices, $\boldsymbol{C}_j$, are chosen by the discrete state, $q_t = j$, and the state evolution and observation noises are Gaussian distributed as in FAHMM. The initial continuous state is also Gaussian distributed, $\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}_{q_1}, \boldsymbol{\Sigma}^{(i)}_{q_1})$. It is also possible to use GMMs for all the noise distributions. However, it will render the inference even more complicated as will be discussed later.

In addition to the above generative models, the conditional independence assumptions made in the models can be illustrated by dynamic Bayesian networks (DBNs) in Figure 1. In both models, the new discrete state, q_{t+1}, is conditionally independent on the history of the discrete states given the state at time t. For the FAHMM on the left hand side, both the current observation vectors, $\boldsymbol{o}_t$, and the continuous state vectors, $\boldsymbol{x}_t$, are conditionally independent on the history of all the other variables given the current discrete state, q_t. For the SLDS on the right hand side, the new continuous state vector, $\boldsymbol{x}_{t+1}$, is conditionally independent on the history of all the variables given the new discrete state, q_{t+1}, and the continuous state vector at time t. The current observation vector, $\boldsymbol{o}_t$, is conditionally independent on the history given both the current discrete and continuous state.

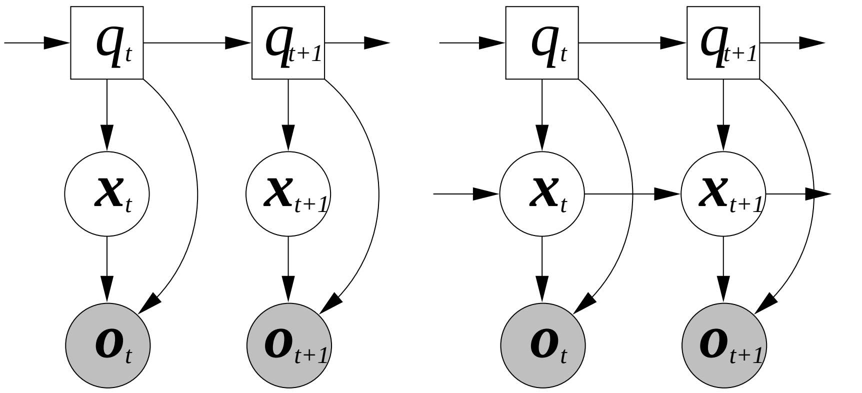

Figure 1: DBNs representing a factor analysed HMM and a SLDS. Square nodes represent discrete variables and round nodes continuous variables. Variables not connected by the directed arcs are assumed to be conditionally independent.

The state evolution process in the SSM is a compromise between the FAHMM and SLDS. Instead of propagating the continuous state vector over the segment boundaries, it is reset according to the initial state distribution when the discrete state switches. The generative model of SSM can be expressed as

$$\boxed{\begin{aligned} & & q_t &\sim P(q_t|q_{t-1}) \\ q_t \neq q_{t-1}: & & \boldsymbol{x}_t &\sim \mathcal{N}(\boldsymbol{\mu}^{(i)}_{q_t}, \boldsymbol{\Sigma}^{(i)}_{q_t}) \\ q_t = q_{t-1}: & & \boldsymbol{x}_t &= \boldsymbol{A}_{q_t}\boldsymbol{x}_{t-1} + \boldsymbol{w}_{q_t} \\ & & \boldsymbol{o}_t &= \boldsymbol{C}_{q_t}\boldsymbol{x}_t + \boldsymbol{v}_{q_t} \end{aligned}}$$

The resetting of the continuous state cannot be graphically expressed in a DBN without introducing auxiliary switching parameters and is therefore omitted. SSMs have previously been applied for phoneme recognition using the TIMIT corpus in [7].

The difference between SLDS and SSM can be seen looking at the continuous state posteriors in Figure 2. The posterior means are very similar apart from a few transients at the segment boundaries, marked by dotted vertical lines. However, the posterior variances differ significantly. Since the posterior covariance matrices are determined by a series of matrix multiplications, the variances reach a steady state in the middle of long segments as the covariance matrices tend toward the eigenvectors of the state transition matrices. The propagation of the continuous state posterior in SLDS smooths the segment boundaries whereas the resetting of the continuous state results in large transients in SSM.

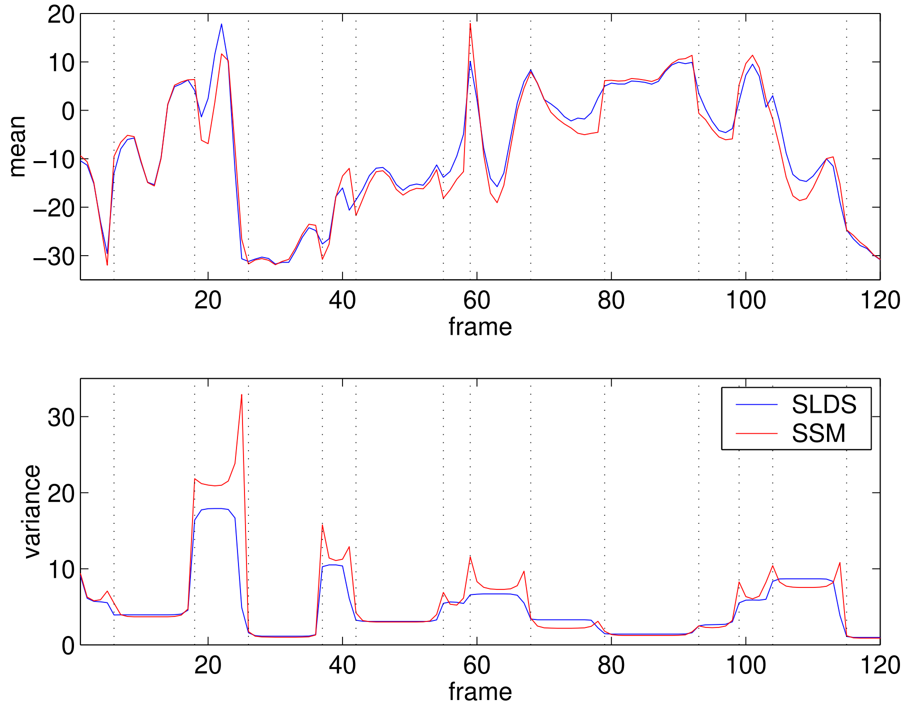

Figure 2: The posterior mean and variance of the first state vector element for an utterance.

2.2 Inference and Training

For the FAHMM the inference is simple due to the conditional independence assumption. Both the standard Viterbi and forward-backward algorithms for the HMMs can be easily implemented for FAHMMs in $O(T)$ by modifying the likelihood calculations [25]. The parameter optimisation can be carried out using the EM algorithm [6]. Due to the additional level of hidden parameters, an efficient two level algorithm may be used [25]. The inference for the SSM is more complicated. The position in the continuous state space depends on the number of frames spent in the current segment. However, the standard optimisation methods are feasible [19], but at a computational cost of $O(T^2)$.

For the SLDS the inference is intractable. Since the current position in the continuous state space depends on the entire history of the discrete states, marginalisation becomes prohibitive. Exact computation of the observation likelihood or the posterior likelihood of the hidden variables given the observation sequence has to be carried out over $O(N_s^T)$ paths. However, given the discrete state sequence and the mixture component indicator sequences when GMMs are used, SLDS becomes tractable and the traditional Kalman filtering and RTS smoothing algorithms can be used for inference and EM algorithm for optimising the model parameters. The intractable inference also renders any standard decoding algorithm inadmissible. Instead of full decoding, evaluating the performance of a SLDS system may be done if the segmentations of a number of hypotheses were known. The segmentations for the training and N-best rescoring may be obtained from a tractable system such as the FAHMM.

For statistical models, inference usually requires estimating the predicted, filtered or smoothed statistics of an unknown variable. For the LDS, the predicted statistics are $\boldsymbol{x}_{t+1|t} = E\{\boldsymbol{x}_{t+1}|\boldsymbol{o}_{1:t}\}$ and $\boldsymbol{\Sigma}_{t+1|t} = E\{\boldsymbol{x}_{t+1}\boldsymbol{x}'_{t+1}|\boldsymbol{o}_{1:t}\}$ where $\boldsymbol{o}_{1:t}$ denotes a partial observation sequence up to time t. The filtered estimates are defined as $\boldsymbol{x}_{t|t} = E\{\boldsymbol{x}_t|\boldsymbol{o}_{1:t}\}$ and $\boldsymbol{\Sigma}_{t|t} = E\{\boldsymbol{x}_t\boldsymbol{x}'_t|\boldsymbol{o}_{1:t}\}$. To evaluate these estimates the standard Kalman filter recursion [14] can be written as follows

$$\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t\big(\boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t + \boldsymbol{\Sigma}^{(o)}_t\big)^{-1}\boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1} \tag{1}$$

$$\boldsymbol{\Sigma}_{t+1|t} = \boldsymbol{A}_{t+1}\boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1} + \boldsymbol{\Sigma}^{(x)}_{t+1} \tag{2}$$

where $\boldsymbol{\Sigma}_{1|0} = \boldsymbol{\Sigma}^{(i)}_1$ and the mean vectors are

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t\big(\boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t + \boldsymbol{\Sigma}^{(o)}_t\big)^{-1}(\boldsymbol{o}_t - \boldsymbol{C}_t\boldsymbol{x}_{t|t-1} - \boldsymbol{\mu}^{(o)}_t) \tag{3}$$

$$\boldsymbol{x}_{t+1|t} = \boldsymbol{A}_{t+1}\boldsymbol{x}_{t|t} + \boldsymbol{\mu}^{(x)}_{t+1} \tag{4}$$

and $\boldsymbol{x}_{1|0} = \boldsymbol{\mu}^{(i)}_1$. This is also known as the covariance form of the Kalman filter [13]. The distribution defined by $p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t-1}) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$ is called the predicted distribution of the state vector. The distribution $p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_{t|t}, \boldsymbol{\Sigma}_{t|t})$ is called the filtered distribution. An observation, $\boldsymbol{o}_t$, given the history of past observations, $\boldsymbol{o}_{1:t-1}$, is distributed according to the following Gaussian

$$p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{C}_t\boldsymbol{x}_{t|t-1} + \boldsymbol{\mu}^{(o)}_t, \boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t + \boldsymbol{\Sigma}^{(o)}_t) \tag{5}$$

The standard derivation of the filtering and smoothing algorithms has been based on minimum mean square estimation and the orthogonality principle [13]. Alternatively, the derivation may be done completely using properties of conditional Gaussian distributions and matrix algebra [24]. In case of the SLDS, these statistics are only conditionally Gaussian given the discrete state sequence Q and the mixture component indicator sequences $\Omega^{(x)}$ and $\Omega^{(o)}$.

Traditionally, the statistics of the smoothed distribution, $p(\boldsymbol{x}_t|\boldsymbol{O}) = \mathcal{N}(\boldsymbol{x}_t; \hat{\boldsymbol{x}}_t, \hat{\boldsymbol{\Sigma}}_t)$, are obtained using the Rauch-Tung-Striebel (RTS) smoother [22]. The RTS smoothing algorithm requires the above Kalman filter statistics be known. The recursion can be written as follows

$$\hat{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{-1}_{t+1|t}\big(\hat{\boldsymbol{\Sigma}}_{t+1} - \boldsymbol{\Sigma}_{t+1|t}\big)\boldsymbol{\Sigma}^{-1}_{t+1|t}\boldsymbol{A}_{t+1}\boldsymbol{\Sigma}_{t|t} \tag{6}$$

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{-1}_{t+1|t}(\hat{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_{t+1|t}) \tag{7}$$

Alternatively, the smoother statistics may be estimated using the information form algorithms [13]. The derivation of the information form algorithms for models which include the noise mean vectors is presented Appendix B. The noise mean vectors need to be explicit to enable Gaussian mixture model noise sources. In the information form, the forward and backward passes may be run independently, and the smoother estimates are obtained by combining the statistics from both passes. The information form statistics will be used later in this paper.

The EM algorithm for the standard LDS consists of estimating the smoother statistics as above and updating the model parameters using these statistics. The observation process parameters are updated as follows

$$\hat{\boldsymbol{C}} = \Big(\sum_{t=1}^{T}\boldsymbol{o}_t\hat{\boldsymbol{x}}_t' - \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{o}_t\sum_{t=1}^{T}\hat{\boldsymbol{x}}_t'\Big)\Big(\sum_{t=1}^{T}\hat{\boldsymbol{R}}_t - \frac{1}{T}\sum_{t=1}^{T}\hat{\boldsymbol{x}}_t\sum_{t=1}^{T}\hat{\boldsymbol{x}}_t'\Big)^{-1} \tag{8}$$

$$\hat{\boldsymbol{\mu}}^{(o)} = \frac{1}{T}\sum_{t=1}^{T}\big(\boldsymbol{o}_t - \hat{\boldsymbol{C}}\hat{\boldsymbol{x}}_t\big) \tag{9}$$

$$\hat{\boldsymbol{\Sigma}}^{(o)} = \frac{1}{T}\sum_{t=1}^{T}\Big(\boldsymbol{o}_t\boldsymbol{o}_t' - \Big[\,\hat{\boldsymbol{C}}\ \hat{\boldsymbol{\mu}}^{(o)}\Big]\big[\,\boldsymbol{o}_t\hat{\boldsymbol{x}}_t'\ \boldsymbol{o}_t\,\big]'\Big) \tag{10}$$

and the state evolution parameters are updated as follows

$$\hat{\boldsymbol{A}} = \Big(\sum_{t=2}^{T}\hat{\boldsymbol{R}}_{t-1,t} - \frac{1}{T-1}\sum_{t=2}^{T}\hat{\boldsymbol{x}}_t\sum_{t=2}^{T}\hat{\boldsymbol{x}}_{t-1}'\Big)\Big(\sum_{t=2}^{T}\hat{\boldsymbol{R}}_{t-1} - \frac{1}{T-1}\sum_{t=2}^{T}\hat{\boldsymbol{x}}_{t-1}\sum_{t=2}^{T}\hat{\boldsymbol{x}}_{t-1}'\Big)^{-1} \tag{11}$$

$$\hat{\boldsymbol{\mu}}^{(x)} = \frac{1}{T-1}\sum_{t=2}^{T}\big(\hat{\boldsymbol{x}}_t - \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_{t-1}\big) \tag{12}$$

$$\hat{\boldsymbol{\Sigma}}^{(x)} = \frac{1}{T-1}\sum_{t=2}^{T}\Big(\hat{\boldsymbol{R}}_t - \Big[\,\hat{\boldsymbol{A}}\ \hat{\boldsymbol{\mu}}^{(x)}\Big]\big[\,\hat{\boldsymbol{R}}_{t-1,t}\ \hat{\boldsymbol{x}}_t\,\big]'\Big) \tag{13}$$

$$\hat{\boldsymbol{\mu}}^{(i)} = \hat{\boldsymbol{x}}_1 \tag{14}$$

$$\hat{\boldsymbol{\Sigma}}^{(i)} = \hat{\boldsymbol{R}}_1 - \hat{\boldsymbol{\mu}}^{(i)}\hat{\boldsymbol{\mu}}^{(i)\prime} \tag{15}$$

which requires the covariance matrix of a joint posterior of two successive state vectors be known. This distribution, $p(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}|\boldsymbol{O})$, is also a Gaussian and its covariance matrix can be written as

$$\hat{\boldsymbol{\Sigma}}_{t,t+1} = \hat{\boldsymbol{\Sigma}}_{t+1}\boldsymbol{\Sigma}_{t+1|t}^{-1}\boldsymbol{A}_{t+1}\boldsymbol{\Sigma}_{t|t} \tag{16}$$

The posterior mean vectors may also be used in estimating the true trajectory of the modelled signal. For all the state space models presented above, the estimated trajectory is obtained by $\hat{\boldsymbol{o}}_t = \boldsymbol{C}_t E\{\boldsymbol{x}_t|\boldsymbol{O}, Q\} + \boldsymbol{\mu}^{(o)}$. The discrete state sequence may be obtained using a system based on tractable model such as FAHMM. The true and the estimated trajectories are shown in Figure 3, where a three emitting states per phone FAHMM was used to obtain the discrete state sequence. Single state per phone SLDS and SSM systems were used to compare the trajectories against the FAHMM for which the state alignment within a model was inferred using Viterbi algorithm. Despite the different continuous state propagation assumptions in SLDS and SSM, the estimated trajectories are very close which could be predicted from the posteriors in Figure 2. However, the difference compared to the FAHMM trajectory is quite remarkable, especially at 'ow' and 'ao', and has been used to argue in favour of the LDSs against HMMs [7, 10] for speech recognition.

2.3 Approximate Inference and Learning

The approximate inference and learning algorithms used in the literature can be first categorised into deterministic and stochastic methods. The deterministic algorithms include

- **Generalised pseudo Bayesian algorithm** of order r (GPB$^{(r)}$) approximates N_s^t mixture components at time t by a Gaussian mixture with r components using moment matching [1, 18]. Usually orders $r = 1$ and $r = 2$ have been used. The moment matching can be shown to be optimal in Kullback-Leibler sense and it has been shown that the error is bounded [4] despite they accumulate over time. This kind of collapsing technique can be used for both filtering and smoothing.

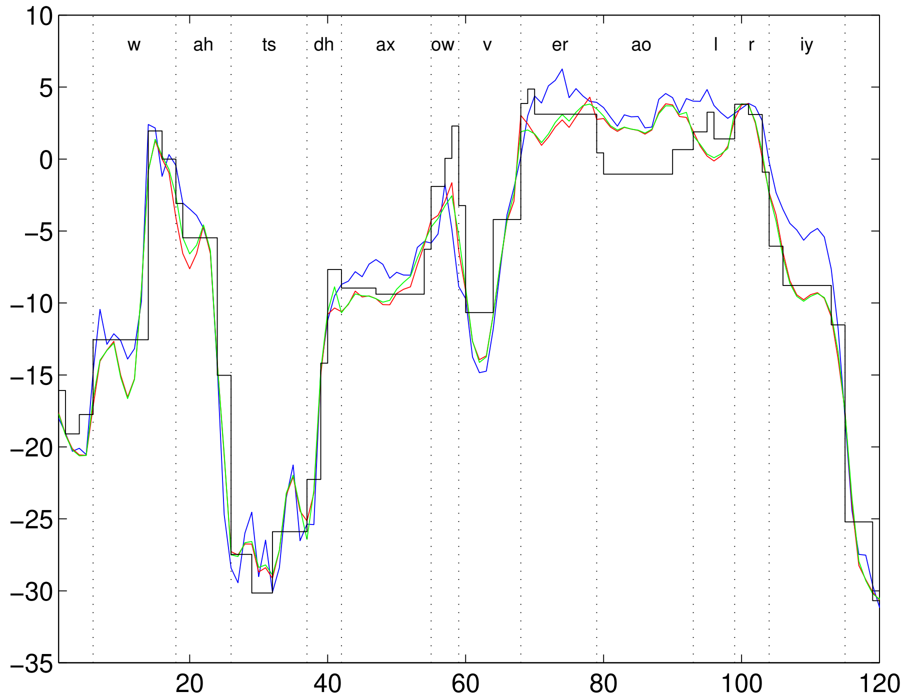

Figure 3: True and estimated trajectories of the first mel-frequency cepstral coefficient.

- **Expectation Propagation** [17] is an iterative algorithm related to GPB$^{(1)}$ where the resulting mixture distribution at each time instant is approximated by a single Gaussian. The expectation propagation algorithm allows the estimates to be refined iteratively. The first two moments at any time instant can be made arbitrarily close to the moments of the true mixture distribution [12].

- **Approximate Viterbi algorithm** keeps only the path with the highest log-likelihood active [20]. Unfortunately, since the observation likelihood depends on the entire history of the discrete state sequence a true Viterbi algorithm [27] is not admissible. The approximate Viterbi algorithm can only be justified if one can argue that the corresponding positions in the state space are the same for different discrete state sequences.

- **Structured variational approximation** [26] exploits the tractable substructures in the SLDS by decoupling the discrete state evolution from the standard LDS. Instead, a tractable variational distribution is used as approximate HMM output distributions and discrete state posteriors. The variational approximation consists of alternating between standard forward-backward algorithm for HMM and standard inference for LDS. The HMM output distribution values are estimated from the sufficient statistics obtained in the LDS inference and the time varying LDS parameters are obtained using the discrete state posteriors from the HMM inference [21].

In all the above algorithms the model parameters are updated according to standard equations presented in Section 2.2 once the posterior distributions have been estimated. Also, all the algorithms are based on modifying the distributions or removing some dependencies in the model. None of them can be guaranteed to converge even in the limit. Employing GMM noise sourced introduces even more approximation errors or is impossible. In contrast the stochastic algorithms do not modify the model structure in any way and they converge in the limit.

The stochastic algorithms in the literature are all based on Gibbs sampling which is one of the Markov chain Monte Carlo (MCMC) methods [23]. The algorithm in [5] uses explicit definition of backward state space model and it is the first algorithm that implements the proposal distribution in $O(T)$ operations. The Gibbs sampling algorithm described in the following section is very similar to the one in [8]. They both take advantage of the backward information filtering which does not require explicit computation of the backward state space model although implicit assumption of invertible state evolution matrices is made [13]. Also, the covariance matrices are assumed to be diagonal and positive definite in this paper. The mean vectors of all the noise sources have been included as well to make the extension to GMMs possible.

3 Monte Carlo Methods

Monte Carlo methods are used to approximate expectations of functions under distributions which cannot be analytically solved. Two problems arise in this simulation based approach. First, how to draw samples from a given probability distribution. Second, how to approximate expectations under these distributions. Given N samples, $\{x^{(1)}, \ldots, x^{(N)}\}$ from a distribution $p(x)$, it can be approximated with the following empirical point-mass function

$$\hat{p}_N(x) = \frac{1}{N}\sum_{n=1}^{N}\delta(x - x^{(n)}) \tag{17}$$

where $\delta(x)$ denotes the Dirac delta function. Consequently, the expectations (integrals), $I(f)$, of functions, $f(x)$, under the distribution can be approximated with tractable sums

$$\hat{I}_N(f) = \frac{1}{N}\sum_{n=1}^{N} f(x^{(n)}) \underset{N\to\infty}{\longrightarrow} I(f) = \int f(x)p(x)dx, \tag{18}$$

which are unbiased and by the strong law of large numbers will converge almost surely as N tends to infinity [23].

The first problem, drawing samples from a given probability distribution is more complicated. It is straightforward only if the distribution, $p(x)$, is of a standard form, for example Gaussian or discrete. Otherwise, Monte Carlo methods including rejection sampling, importance sampling or Markov chain Monte Carlo (MCMC) algorithms have to be used.

3.1 Gibbs Sampling

Gibbs sampling is an instance of Markov chain Monte Carlo (MCMC) method for sampling from distributions over at least two dimensions [23]. It is assumed that whilst $p(\boldsymbol{x})$ is too complex to draw samples from directly, its conditional distributions, $p(x_j|x_{-j}^{(n)})$ where $x_{-j}^{(n)} = \{x_1^{(n)}, \ldots, x_{j-1}^{(n)}, x_{j+1}^{(n-1)}, \ldots, x_N^{(n-1)}\}$, can be used as proposal distributions. The superscript refers to the nth iteration. For many graphical models these conditional distributions are easy to sample from. In general, a single iteration of Gibbs sampling

$$x_j^{(n)} \sim p(x_j|x_{-j}^{(n)}). \tag{19}$$

for all j. After N iterations the final estimates are computed in common with all Monte Carlo methods

$$\hat{p}_N(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}^{(n)}), \tag{20}$$

which converge toward its invariant distribution, $p(\boldsymbol{x})$, if the Markov chain is finite state irreducible and aperiodic [9]. Gibbs sampling explores the state space by a random walk steered by the conditional distributions. It may be very slow to converge if the state space is large. Sometimes the structure of the model allows efficient sampling by separating tractable substructures and thus enables drawing samples in a lower dimensional state space.

Rao-Blackwellisation [23] can be used to increase the efficiency of Gibbs sampling for SLDS. Instead of drawing samples directly from the joint posterior of the discrete and continuous states, $p(q_t, \boldsymbol{x}_t|\boldsymbol{O}, q_{-t}, \boldsymbol{x}_{-t})$, samples are drawn from the posterior of the discrete state, $P(q_t|\boldsymbol{O}, q_{-t})$. Standard inference for the posterior of the continuous state vectors, $p(\boldsymbol{x}_t|\boldsymbol{O}, \boldsymbol{Q}^{(n)})$, can be used given the estimated discrete state sequence, $\boldsymbol{Q}^{(n)}$.

The Rao-Blackwellised Gibbs sampling (RBGS) for SLDS [8] can be summarised as the following algorithm

1. initialise $\{q_1^{(1)}, \dots, q_T^{(1)}\}$
2. for iteration $n > 1$
   - draw samples $q_t^{(n)} \sim P(q_t|\boldsymbol{O}, q_{-t}^{(n)})$ for all $t \in [1, T]$
   - estimate statistics $\hat{\boldsymbol{x}}_t^{(n)} = E\{\boldsymbol{x}_t|\boldsymbol{O}, \boldsymbol{Q}^{(n)}\}$ and $\hat{\boldsymbol{R}}_t^{(n)} = E\{\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{O}, \boldsymbol{Q}^{(n)}\}$.

Once all N iterations are finished, the final estimates can be approximated as follows

$$\gamma_j(t) \approx \frac{1}{N}\sum_{n=1}^{N} \delta(j - q_t^{(n)}) \tag{21}$$

$$\hat{\boldsymbol{x}}_t \approx \frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{x}}_t^{(n)} \tag{22}$$

$$\hat{\boldsymbol{R}}_t \approx \frac{1}{N}\sum_{n=1}^{N} \hat{\boldsymbol{R}}_t^{(n)} \tag{23}$$

which converge almost surely [8] toward the true posterior statistics $\gamma_j(t) = P(q_t = j|\boldsymbol{O})$, $\hat{\boldsymbol{x}}_t = E\{\boldsymbol{x}_t|\boldsymbol{O}\}$ and $\hat{\boldsymbol{R}}_t = E\{\boldsymbol{x}_t\boldsymbol{x}_t'|\boldsymbol{O}\}$.

3.2 Proposal Distribution for RBGS

A simple solution to obtain $P(q_t|\boldsymbol{O}, q_{-t})$ would require running standard Kalman filter for all the N_s discrete states for every time instance per iteration since $P(q_t = j|\boldsymbol{O}, q_{-t}) \propto p(\boldsymbol{O}|q_t = j, q_{-t})p(q_t = j|q_{-t})$. Unfortunately, this solution has a complexity of $O(T^2)$ since the observation distributions are dependent on the entire discrete state history. More efficient algorithm can be derived using the following result [8]

$$\begin{aligned} &P(q_t|\boldsymbol{O}, q_{-t}) \propto \\ &\quad P(q_{t+1}|q_t)P(q_t|q_{t-1})p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, q_{1:t}) \int p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t})d\boldsymbol{x}_t \end{aligned} \tag{24}$$

where the term immediately before the integral, $p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, q_{1:t})$, and the second term inside the integral, $p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t})$, are given by the standard Kalman filter described in Appendix B as follows

$$p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, q_{1:t}) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{C}_t\boldsymbol{x}_{t|t-1} + \boldsymbol{\mu}_t^{(o)}, \boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}_t' + \boldsymbol{\Sigma}_t^{(o)}) \tag{25}$$

$$p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t}) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_{t|t}, \boldsymbol{\Sigma}_{t|t}) \tag{26}$$

Defining parameters $\boldsymbol{C}_{t|t+1}^{(f)}$, $\boldsymbol{\mu}_{t|t+1}^{(f)}$ and $\boldsymbol{\Sigma}_{t|t+1}^{(f)}$ for the distribution $p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})$ as in Appendix B.2, the integral in Equation 24 can be expressed as the following Gaussian

$$\begin{aligned} &\int p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t})d\boldsymbol{x}_t = \\ &\quad \mathcal{N}(\boldsymbol{o}_{t+1:T}; \boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{x}_{t|t} + \boldsymbol{\mu}_{t|t+1}^{(f)}, \boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{\Sigma}_{t|t}\boldsymbol{C}_{t|t+1}'^{(f)} + \boldsymbol{\Sigma}_{t|t+1}^{(f)}) \\ &\propto \left|\boldsymbol{\Sigma}_{t|t}\boldsymbol{P}_{t|t+1}^{-1} + \boldsymbol{I}\right|^{-\frac{1}{2}} \exp\Big\{\boldsymbol{x}_{t|t}'\boldsymbol{P}_{t|t+1}^{-1}\boldsymbol{m}_{t|t+1} - \frac{1}{2}\boldsymbol{x}_{t|t}'\boldsymbol{P}_{t|t+1}^{-1}\boldsymbol{x}_{t|t} \\ &\quad + \frac{1}{2}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})'\boldsymbol{P}_{t|t+1}^{-1}\big(\boldsymbol{P}_{t|t+1}^{-1} + \boldsymbol{\Sigma}_{t|t}^{-1}\big)^{-1}\boldsymbol{P}_{t|t+1}^{-1}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})\Big\} \end{aligned} \tag{27}$$

where $\boldsymbol{P}^{-1}_{t|t+1}$ and $\boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{m}_{t|t+1}$ are obtained using the following backward information filter derived in Appendix B.2

$$\boldsymbol{P}^{-1}_{t|t} = \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_t\boldsymbol{C}_t + \boldsymbol{P}^{-1}_{t|t+1} \tag{28}$$

$$\boldsymbol{P}^{-1}_{t-1|t} = \boldsymbol{A}'_t\big(\boldsymbol{P}^{-1}_{t|t}\boldsymbol{\Sigma}^{(x)}_t + \boldsymbol{I}\big)^{-1}\boldsymbol{P}^{-1}_{t|t}\boldsymbol{A}_t \tag{29}$$

where $\boldsymbol{P}^{-1}_{T|T+1} = \boldsymbol{0}$ and

$$\boldsymbol{P}^{-1}_{t|t}\boldsymbol{m}_{t|t} = \boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{m}_{t|t+1} + \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_t(\boldsymbol{o}_t - \boldsymbol{\mu}^{(o)}_t) \tag{30}$$

$$\boldsymbol{P}^{-1}_{t-1|t}\boldsymbol{m}_{t-1|t} = \boldsymbol{A}'_t\big(\boldsymbol{P}^{-1}_{t|t}\boldsymbol{\Sigma}^{(x)}_t + \boldsymbol{I}\big)^{-1}\boldsymbol{P}^{-1}_{t|t}(\boldsymbol{m}_{t|t} - \boldsymbol{\mu}^{(x)}_t) \tag{31}$$

where $\boldsymbol{P}^{-1}_{T|T+1}\boldsymbol{m}_{T|T+1} = \boldsymbol{0}$. These algorithms differ from the ones presented in [8] by including the mean vectors which allows an extension to Gaussian mixture model noise sources.

Finally, the proposal distribution, $P(q_t|\boldsymbol{O}, q_{-t})$, to draw the samples from in Rao-Blackwellised Gibbs sampling for SLDS can be expressed as

$$\begin{aligned}
&P(q_t|\boldsymbol{O}, q_{-t}) \propto \\
&\quad P(q_{t+1}|q_t)P(q_t|q_{t-1})\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{C}_t\boldsymbol{x}_{t|t-1} + \boldsymbol{\mu}^{(o)}_t, \boldsymbol{C}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{C}'_t + \boldsymbol{\Sigma}^{(o)}_t)\big|\boldsymbol{\Sigma}_{t|t}\boldsymbol{P}^{-1}_{t|t+1} + \boldsymbol{I}\big|^{-\frac{1}{2}} \\
&\quad \times \exp\Big\{\boldsymbol{x}'_{t|t}\boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{m}_{t|t+1} - \frac{1}{2}\boldsymbol{x}'_{t|t}\boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{x}_{t|t} \\
&\quad + \frac{1}{2}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})'\boldsymbol{P}^{-1}_{t|t+1}\big(\boldsymbol{P}^{-1}_{t|t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{P}^{-1}_{t|t+1}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})\Big\}
\end{aligned} \tag{32}$$

A detailed derivation can be found in Appendix C.

As discussed in Section 2.1, Gaussian mixture models may be used as the noise sources for the SLDS. They can be easily incorporated into the Gibbs sampling framework. In the initialisation, the mixture component indicator sequences must also be initialised. The proposal distribution in Equation 32 has to be modified by multiplying it by the mixture component priors. For example, if GMM is used as the observation noise distribution the proposal distribution has the form

$$P(q_t, \omega^o_t|\boldsymbol{O}, q_{-t}, \omega^o_{-t}) = P(\omega^o_t)P(q_t|\boldsymbol{O}, q_{-t}) \tag{33}$$

If there are $M^{(o)}$ observation noise components and N_s possible states at a time instant t, the sample has to be drawn from $M^{(o)}N_s$ alternatives. Also, the forward and backward information filter statistics have to be computed along the fixed mixture component indicator sequences.

3.3 RBGS in Speech Recognition

The convergence of the Gibbs sampling depends heavily on the initialisation and the size of the state space where the samples are drawn from. The number of discrete states in a speech recognition system is typically in the thousands. For the initialisation a factor analysed HMM (FAHMM) [25] may be used. In the experiments later in this paper, a single discrete state per phone is used. Very similar FAHMM can be trained by tying all the FAHMM parameters except the state space mean vectors at the model level. The only difference between the FAHMM and SLDS is in the state evolution process. The FAHMM is based on piece-wise constant state evolution whereas SLDS is smooth and linear. The training data can thus be aligned using the FAHMM to obtain a reasonable initial segmentation for the SLDS training. The state space may be further reduced by taking the transition restrictions into account.

Since the transcriptions for the training data are given in the standard supervised training schemes, the alignments from the Gibbs sampling iterations have to satisfy the restrictions imposed by the transcriptions. The restrictions can be summarised as follows.

- An utterance has to start in the first label in the transcription;

- The correct order of the labels in the transcription has to be retained at all times;
- An utterance has to finish in the last label in the transcription.

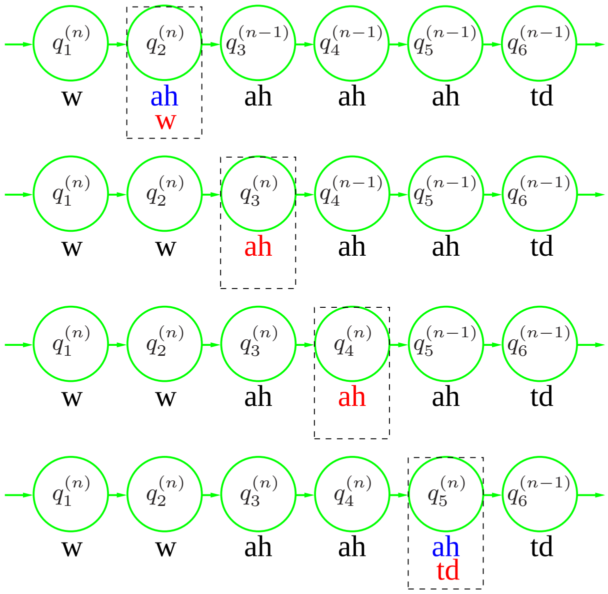

Figure 4: Example iteration of Gibbs sampling for utterance 'what'.

An example iteration for an utterance 'what' is shown in Figure 4. The utterance consists of three phones 'w', 'ah' and 'td'. No samples have to be drawn for the first and the last state since the transcription starts in the label 'w' and finishes in the label 'td'. The sampling iteration moves to the right and at time $t = 2$ the sample has to be drawn from two alternatives, 'w' and 'ah'. Since the state boundaries can move to the left only by a single time instant, similar restriction is imposed to right movements. Therefore at time $t = 3$ no samples have to be drawn. At time $t = 4$, no samples can be drawn to retain the correct order of the labels in the transcription. Finally, the boundary between 'ah' and 'td' must be contested at time $t = 5$.

3.4 Parameter Optimisation

The Gibbs sampling above explores different segmentations of the training data and given the initialisation is satisfactory, only a few samples have to be drawn to get reasonable estimates for the state posteriors. The standard expectation maximisation algorithm [6] can be generalised by using Monte Carlo approximation in the E step. In the case where multiple samples are drawn, this is known as the Monte Carlo EM (MCEM) [28]. The auxiliary function of SLDS for the EM algorithm can be expressed as

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \sum_{\forall Q} \int p(\boldsymbol{X}, Q|\boldsymbol{O}, \mathcal{M}) \log p(\boldsymbol{O}, \boldsymbol{X}, Q|\hat{\mathcal{M}}) d\boldsymbol{X} \tag{34}$$

Using the standard techniques to maximise the log-likelihood of the data the new discrete state parameters are given by

$$\hat{\pi}_j = \frac{\gamma_j(1)}{\sum_{i=1}^{N_s} \gamma_i(1)} \tag{35}$$

$$\hat{a}_{ij} = \frac{\sum_{t=2}^{T} \xi_{ij}(t)}{\sum_{t=2}^{T} \gamma_i(t-1)} \tag{36}$$

where $\xi_{ij}(t) = P(q_{t-1} = i, q_t = j|\boldsymbol{O})$ must be replaced by the counts of the transitions from state i to j and $\gamma_j(t)$ as defined in Equation 21.

The new linear dynamical system parameters are updated according to the standard formulae in Equations 8-15 where the sufficient statistics are accumulated along the fixed discrete state and possible mixture component indicator sequences. The update formulae in Equations 8-15 are based on the assumption of the posteriors being single component Gaussians. For the SLDS this is not true since there should be N_s^T components at every time instance in an utterance and the convergence of MCEM cannot be guaranteed.

An alternative approach to parameter update is to use a single most likely discrete state sequence (MLSS) found during RBGS and update the LDS parameters along this path. Given the discrete state sequence the continuous state posteriors are distributed according to a single Gaussian and the standard LDS parameter update formulae may be used. In MLSS training, the discrete state posteriors, $\gamma_j(t)$, in Equations 8-15 are replaced by a zero-one function that picks the statistics from the along the most likely path.

4 Experiments

In this section, experiments using the ARPA Resource Management Corpus are presented. The aim of these experiments is to make a fair comparison between a tractable system such as the FAHMM and the SLDS for which approximate methods have to be used. The word error rates for the state-of-the-art three state systems are quoted just for reference. The more complex systems were evaluated with simpler configurations. The SLDS and SSM systems were also compared using the same approximate training and evaluation schemes. Even though exact inference is possible for the SSM, it cannot be extended to include Gaussian mixture models in the noise sources. Since the same approximate training and inference schemes apply for SMMs, it may be argued that the comparisons presented below are fair.

Following the HTK RM Recipe [29], it is easy to build a state-of-the-art system based on HMMs. The baseline systems for the experiments below were built starting from a context independent single mixture Gaussian HMM system with three emitting states and diagonal covariance matrices. The initial monophone models were cloned to produce a triphone system for which decision tree clustering was applied to make the final model sets. The clustering was done in both model (single state systems) and state level (three state systems) to produce two different systems.

Two observation vector configurations based on Mel-frequency cepstral coefficients (MFCCs) were used. In the first configuration, the vectors comprised 13 MFCCs with the c_0 replaced by the energy of the frame. This is referred as the $p = 13$ dimensional system. In the second configuration, the delta and delta-delta parameters were added in the observation vectors. This configuration is referred as the $p = 39$ dimensional system.

The training data set of the RM task consists of 3990 utterances. For the evaluation, 1200 test utterances (feb89, oct89, feb91, sep92), `test`, and a randomly selected 300 utterance subset of the training data, `train`, were used. For some of the more time consuming experiments only the `feb89` set was used. A simple word-pair grammar was used as the language model in all the experiments.

4.1 Single State Systems

A single state SLDS was used to make a fair comparison between the baseline FAHMM and the SLDS systems. The baseline FAHMM was based on the HMM system with three emitting states. All the FAHMM parameters apart from the continuous state space mean vectors were tied on the model level. The SLDS system on the other hand had a single state per model. By tying this way, it is possible to see how the different state evolution assumptions, piece-wise constant and linear continuous, perform on the same task with approximately equal complexity. The baseline FAHMM system was used to align the training data and produce the aligned 100-best lists. Single and two observation mixture component systems were used. The observation process parameters of SLDS and SSM systems were initialised based on the baseline FAHMM. Both systems used a single set of LDS parameters per triphone. The initial continuous state vector distribution was initialised to the parameters of the first emitting state of the alignment FAHMM. The state evolution noise mean vectors were set to zeroes and the variances equal to the initial state variances. The state transition matrices, $\boldsymbol{A}_j$, were all initialised to identity matrices.

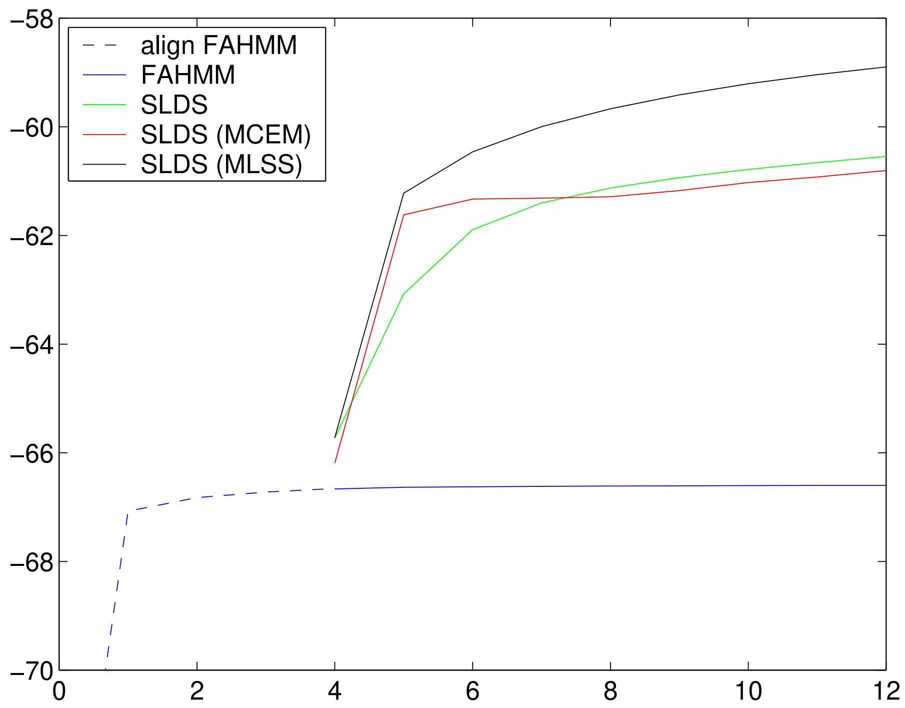

Figure 5: Average log-likelihood of the training data against the number of iterations.

The model aligned training data was used to train the SLDS and SSM systems with the EM algorithm. For the FAHMM, the Baum-Welch algorithm [2] was used to infer the state alignment keeping the model alignment fixed. This way the algorithm could find the optimal discrete state alignments within the fixed segments. The average log-likelihoods of the training data against the number of iterations are shown in Figure 5. The first four iterations correspond to the baseline FAHMM training with full Baum-Welch algorithm and the last nine iterations correspond to the model aligned training. For the SLDS with fixed training alignment (SLDS) the log-likelihood slowly increased. Using MCEM the log-likelihood always increased, though not as smoothly, and yielded a lower final log-likelihood than the fixed alignment training. As discussed in Section 3, the MCEM is not guaranteed to increase the log-likelihood. The state posteriors for this data were highly non-Gaussian. In initial experiments MCEM gave significantly worse performance than other forms of training and was not investigated further. The MLSS training log-likelihoods with 5 iterations of Gibbs sampling are also shown in Figure 5. MLSS training clearly finds alignments with higher log-likelihood than using the fixed alignments. The maximum log-likelihood value was found during the first 5 iterations with these model and data sets. Higher numbers up to 1000 iterations for individual utterances were tested with no significant improvement.

Task	HMM	FAHMM	13-dim FAHMM		39-dim FAHMM	
	$M = 6$	$M = 5$	$M = 1$	$M = 2$	$M = 1$	$M = 2$
`test`	3.99	3.67	19.39	17.60	8.85	7.28
`train`	1.73	1.85	6.37	4.37	1.28	0.60

Table 1: Full decoding results for the state-of-the-art HMM and FAHMM systems and for the 13 and 39 dimensional baseline FAHMMs.

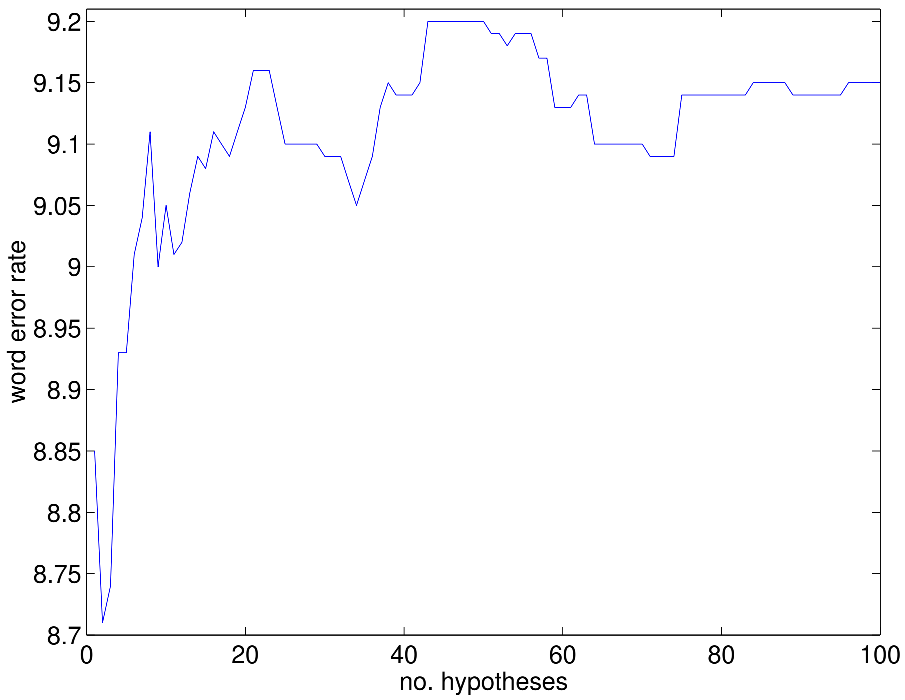

Figure 6: Word error rate for the `test` data against the number of hypotheses for SLDS with fixed aligned N-best rescoring.

The full decoding word error rates for the 13 and 39-dimensional baseline FAHMMs are shown in Table 1. As a reference, word error rates for state clustered multiple component HMM and FAHMM systems are given for the standard 39-dimensional front-end. Due to the non-standard model clustering, tying schemes and single component noise distributions, the baseline FAHMM results are far from the best achievable. These baseline FAHMM systems were used to produce the 100-best lists for the rescoring experiments below. To give an idea of the range of the word error rates that may be obtained by rescoring the 100-best lists, the “oracle” (best) and “idiot” (worst) error rates are shown in Table 2. The number of hypotheses in an N-best rescoring scheme should also be large enough to minimise the cross system effects due to the different modelling assumptions. The word error rates for the `test` data against the number of hypotheses for SLDS with fixed aligned N-best rescoring is shown in Figure 6. The error rates varied significantly up to about 20 hypotheses and 50-best rescoring seems to give a bit worse results compared to the

100-best. However, the SLDS and FAHMM state evolution assumptions are different and it can always be argued that a full decoding with SLDS might produce a transcription different to any of the hypotheses produced by the FAHMM.

Task	13-dim FAHMM		39-dim FAHMM	
	$M = 1$	$M = 2$	$M = 1$	$M = 2$
`test`	5.28 - 60.31	4.59 - 59.14	1.13 - 59.47	0.73 - 57.02
`train`	0.19 - 44.20	0.11 - 42.61	0.0 - 44.88	0.0 - 42.73

Table 2: The "oracle - idiot" word error rates for the 13 and 39 dimensional baseline FAHMMs. These give the limits for the word error rates that may be obtained rescoring the corresponding 100-best lists.

The SSM and SLDS systems trained with the fixed alignments and MLSS with 5 iterations were evaluated by rescoring the aligned 100-best lists. The rescoring results are shown in Table 3. The 13 dimensional systems trained with the fixed alignments seem to yield better performance than the baseline FAHMM. This could be a result of the better temporal correlation modelling in the SSM and SLDS compared to the FAHMM when no delta or delta-delta parameters were used. However, the improvement over the baseline is not very significant since it is still far off the state-of-the-art systems. The 39 dimensional single component systems with fixed alignment training show some signs of over training since the `train` set performance is better than the baseline. This does not seem to be the case in the two component systems where the `test` set performance is even lower than the single component one. Surprisingly, the performance is much worse when the MLSS training has been applied even though the average log-likelihoods suggest the training is more efficient. Especially, for the 13 dimensional two component system the performance of the systems trained with MLSS is plainly bad. However, it is well known that the models producing higher log-likelihoods for the seen data do not necessarily perform better on recognising unseen data.

Task	p	fixed SSM		fixed SLDS		MLSS SSM		MLSS SLDS	
		$M = 1$	$M = 2$	$M = 1$	$M = 2$	$M = 1$	$M = 2$	$M = 1$	$M = 2$
`test`	13	17.52	17.23	16.36	16.67	17.86	23.44	16.21	19.51
`train`		6.37	5.95	4.94	4.56	7.01	11.79	5.12	6.48
`test`	39	9.07	9.66	9.16	9.48	12.96	12.18	11.68	15.18
`train`		1.06	0.98	1.09	1.06	2.86	2.15	2.03	3.17

Table 3: Fixed alignment 100-best rescoring word error rates for the model clustered systems trained with fixed alignments, and MLSS with 5 Gibbs sampling iterations.

The rescoring using Gibbs sampling was done using the `feb89` evaluation data set because the iteration of all the 100 hypotheses for even 5 times was very time consuming. The number of iterations was chosen again based on the finding that the highest log-likelihood value was obtained during the first 5 iterations. To illustrate this, the average of the maximum likelihoods for a set of 100 test utterances against the number of Gibbs sampling iterations is shown in Figure 7. The rescoring results using Gibbs sampling are shown in Table 4. The baseline word error rates were 17.57 and 15.70 for the 13 dimensional single and two component systems, and 6.40 and 5.35 for the 39 dimensional ones. Again the performance is disappointing compared to the baseline.

4.2 Three State Systems

Finally, three state systems based on state clustered triphone HMMs and 39 dimensional front-end were built. In these experiments, the FAHMM system was close to the state-of-the-art systems. Only fewer observation mixture components were used. Due to the increased number of states in the systems, more Gibbs sampling iterations had to be carried out to find the maximum likelihood

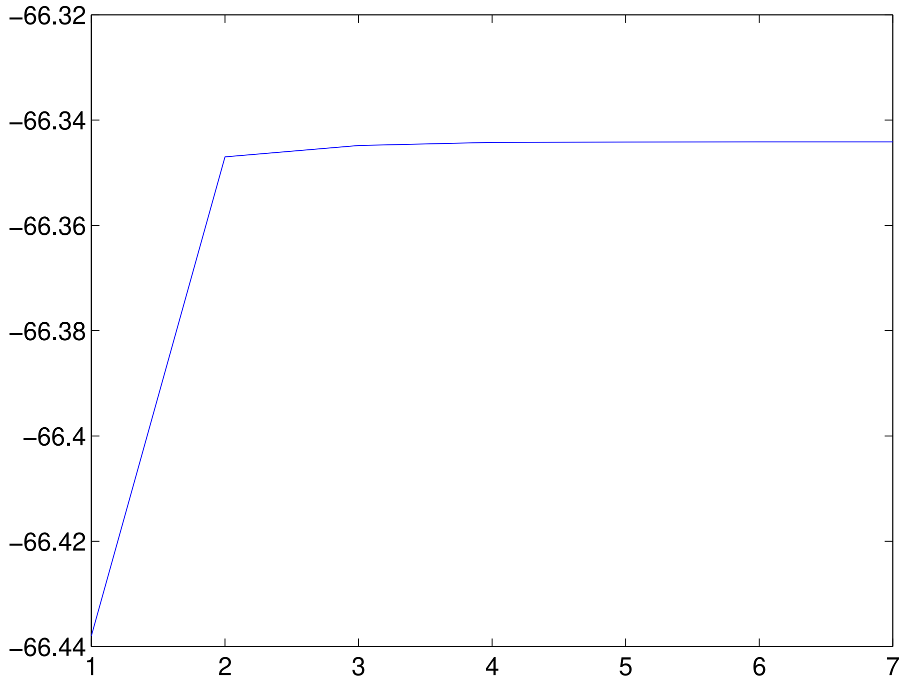

Figure 7: Average of maximum log-likelihood for a set of 100 test utterances with 100 hypotheses against the number of Gibbs sampling iterations. The highest log-likelihoods are obtained within the first 5 iterations.

p	fixed SSM		fixed SLDS		MLSS SSM		MLSS SLDS	
	$M=1$	$M=2$	$M=1$	$M=2$	$M=1$	$M=2$	$M=1$	$M=2$
13	17.42	16.91	15.31	15.38	17.96	17.81	15.31	16.83
39	7.81	8.24	8.06	8.28	10.62	8.59	8.55	9.02

Table 4: 100-best rescoring results in the `feb89` set for the model clustered systems using 5 Gibbs sampling iterations per hypothesis.

state sequence. Due to time constraints MLSS training was not performed. The rescoring results for the SLDS and SSM systems using fixed alignments and Gibbs sampling with 10 iterations are shown in Table 5. For the SSM, Gibbs sampling did not find any better discrete state alignments and the results are equal to the fixed alignment results. The more discrete states there are in a SSM system the closer it becomes to a FAHMM system. This close relationship between the SSM and FAHMM systems may explain that no better alignments were found. Unfortunately, the SLDS performance is again worse than the baseline performance especially for the two component case and when Gibbs sampling was used. The strictly linear first-order state evolution process seems to be inappropriate for speech recognition.

M	FAHMM	SLDS		SSM	
		$N=0$	$N=10$	$N=0$	$N=10$
1	3.67	3.67	3.98	4.49	4.49
2	2.97	3.36	3.44	4.30	4.30

Table 5: 100-best rescoring results in the `feb89` set for the state clustered systems and fixed alignment training. Oracle: 0.12 ($M=1$), 0.08 ($M=2$). Idiot: 51.78 ($M=1$), 51.43 ($M=2$).

5 Conclusions

This paper has introduced a new method to train and evaluate switching linear dynamical systems when used as an acoustic model for speech recognition. The new scheme is based on MCMC simulation of the discrete state space and takes advantage of the tractable sub-structures in the model. Various implementation and efficiency issues for applying Rao-Blackwellised Gibbs sampling to speech recognition have been described.

The performance of the SLDS and FAHMM were compared. RBGS was successfully applied to SLDS for both training and decoding, in terms of increasing log-likelihoods. However, the rescoring results were disappointing. The error rates were typically worse than the baseline FAHMM that was used to generate the N-best lists. Furthermore the performance became worse as "better" state alignments were used. Only the fixed alignment trained models showed any performance gain over the highly simplified alignment FAHMM. This error rate is still significantly worse than a standard HMM, or FAHMM. This happens despite the RBGS is guaranteed to converge in the limit and higher log-likelihoods were obtained in both training and evaluation.

Despite the structure that should provide better model for both spatial and temporal correlation, it appears that the linear first-order state evolution assumption is inappropriate for speech recognition. The systems based on HMMs which have a piece-wise constant state evolution can model a wider range of signals, even non-linear ones and seem to be more suitable for speech signals. The MCMC methods can be applied to non-linear state space models as well but the proposal mechanisms for those cannot be implemented as efficiently as for the SLDS. Currently, no practical algorithms exist for non-linear state space models when applied for speech recognition.

6 Acknowledgements

A-V.I. Rosti is funded by an EPSRC studentship and Tampere Graduate School in Information Science and Engineering. He received additional support from Jenny and Antti Wihuri Foundation. This work made use of equipment kindly supplied by IBM under an SUR award.

References

[1] Y. Bar-Shalom and X-R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, 1993.

[2] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966.

[3] C.S. Blackburn. *Articulatory Methods for Speech Production and Recognition*. PhD thesis, University of Cambridge, 1996.

[4] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings Conference on Uncertainty in Artificial Intelligence*, pages 33–42, 1998.

[5] C.K. Carter and R. Kohn. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83:589–601, 1996.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[7] V. Digalakis. *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*. PhD thesis, Boston University, 1992.

[8] A. Doucet and C. Andrieu. Iterative algorithms for state estimation of jump Markov linear systems. *IEEE Transactions on Sigmal Processing*, 49(6):1216–1227, 2001.

[9] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[10] J. Frankel, K. Richmond, S. King, and P. Taylor. An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces. In *Proceedings International Conference on Speech and Language Processing*, 2000.

[11] D.A. Harville. *Matrix Algebra from a Statistician's Perspective*. Springer, 1997.

[12] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesien networks. In *Proceedings Conference on Uncertainty in Artificial Intelligence*, pages 216–223, 2002.

[13] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[14] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineering, Series D, Journal of Basic Engineering*, 82:35–45, 1960.

[15] J.Z. Ma and L. Deng. Efficient decoding strategy for conversational speech recognition using state-space models for vocal-tract-resonance dynamics. In *Proceedings European Conference on Speech Communication and Technology*, pages 603–606, 2001.

[16] D.Q. Mayne. A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92, 1966.

[17] T.P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[18] K. Murphy. Learning switching Kalman filter models. Technical Report 98-10, Compaq Cambridge Research Lab., 1998. Available at `http://www.cs.berkeley.edu/~murphyk/publ.html`.

[19] M. Ostendorf, V. Digalakis, and O. Kimball. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, 1996.

[20] V. Pavlović, J.M. Rehg, T-J. Cham, and K.P. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proceedings International Conference on Computer Vision*, pages 94–101, 1999.

[21] V. Pavlović, J.M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Proceedings Neural Information Processing Systems*, pages 981–987, 2000.

[22] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *American Institute of Aeronautics and Astronautics Journal*, 3(8):1445–1450, 1965.

[23] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

[24] A-V.I. Rosti and M.J.F. Gales. Generalised linear Gaussian models. Technical Report CUED/F-INFENG/TR.420, Cambridge University Engineering Department, 2001. Available via anonymous ftp from `ftp://svr-ftp.eng.cam.ac.uk/pub/reports/`.

[25] A-V.I. Rosti and M.J.F. Gales. Factor analysed hidden Markov models for speech recognition. *Computer Speech and Language*, 2004. To appear.

[26] L. Saul and M.I. Jordan. Exploiting tractable substructures in intractable networks. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 486–492. The MIT Press, 1996.

[27] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, 1967.

[28] G.C.G. Wei and M.A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.

[29] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.0).* Cambridge University, 2000.

[30] J-L. Zhou, F. Seide, and L. Deng. Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM – model and training. In *Proceedings International Conference on Acoustics, Speech and Signal Processing*, pages 744–747, 2003.

A Useful Results from Matrix Algebra

An inverse of a p by p matrix of the form $\boldsymbol{STU}+\boldsymbol{R}$ has the following decomposition [11]

$$(\boldsymbol{STU}+\boldsymbol{R})^{-1} = \boldsymbol{R}^{-1} - \boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1})^{-1}\boldsymbol{U}\boldsymbol{R}^{-1} \tag{37}$$

This is often called the matrix inversion lemma. It can be often applied in linear estimation problems [13]. Especially, a matrix of the form $\boldsymbol{TU}(\boldsymbol{STU}+\boldsymbol{R})^{-1}$ is often seen in minimum mean square estimation. The following identity for this matrix can be derived using the matrix inversion lemma

$$\begin{aligned}
\boldsymbol{TU}(\boldsymbol{STU}+\boldsymbol{R})^{-1} &= \boldsymbol{TU}\big(\boldsymbol{R}^{-1} - \boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1})^{-1}\boldsymbol{U}\boldsymbol{R}^{-1}\big) \\
&= \boldsymbol{T}\big((\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1}) - \boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}\big)(\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1})^{-1}\boldsymbol{U}\boldsymbol{R}^{-1} \\
&= (\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1})^{-1}\boldsymbol{U}\boldsymbol{R}^{-1}
\end{aligned} \tag{38}$$

A determinant of a p by p matrix of the form $\boldsymbol{STU}+\boldsymbol{R}$ has the following decomposition [11]

$$\big|\boldsymbol{STU}+\boldsymbol{R}\big| = \big|\boldsymbol{R}\big|\big|\boldsymbol{T}\big|\big|\boldsymbol{U}\boldsymbol{R}^{-1}\boldsymbol{S}+\boldsymbol{T}^{-1}\big| \tag{39}$$

B Information Forms of Kalman Filter

The standard covariance form of the Kalman filter and smoother were given in Section 2.2. In this appendix the information forms are presented. In the information form the forward and backward passes can be run independently, and the smoother estimates can be obtained by combining the two in common with the forward backward algorithm for HMMs [29].

B.1 Forward Information Filter

An alternative derivation of the Kalman filter using the information form is presented here. The derivation requires a backward LDS to be introduced. Assuming invertible state evolution matrices, the backward LDS can be written as

$$\boldsymbol{x}_{t-1} = \boldsymbol{A}_t^{-1}\boldsymbol{x}_t - \boldsymbol{A}_t^{-1}\boldsymbol{w}_t \tag{40}$$

$$\boldsymbol{o}_t = \boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{v}_t \tag{41}$$

where the state evolution noise $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are the same independent noise variables as in the forward LDS. The observations from time 1 to t, $\boldsymbol{o}_{1:t}$, can be expressed in terms of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ as

$$\boldsymbol{o}_{1:t} = \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{x}_t + \boldsymbol{v}^{(b)}_{t|t} \tag{42}$$

$$\boldsymbol{o}_{1:t} = \boldsymbol{C}^{(b)}_{t+1|t}\boldsymbol{x}_{t+1} + \boldsymbol{v}^{(b)}_{t+1|t} \tag{43}$$

where the observation matrices, $\boldsymbol{C}^{(b)}_{t|t}$ and $\boldsymbol{C}^{(b)}_{t+1|t}$, are constructed as follows

$$\boldsymbol{C}^{(b)}_{t|t} = \begin{bmatrix} \boldsymbol{C}^{(b)}_{t|t-1} \\ \boldsymbol{C}_t \end{bmatrix} \tag{44}$$

$$\boldsymbol{C}^{(b)}_{t+1|t} = \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1} \tag{45}$$

where $\boldsymbol{C}^{(b)}_{0|1} = [\quad]$; i.e., an empty matrix, and the observation noises, $\boldsymbol{v}^{(b)}_{t|t}$ and $\boldsymbol{v}^{(b)}_{t+1|t}$, are constructed as follows

$$\boldsymbol{v}^{(b)}_{t|t} = \begin{bmatrix} \boldsymbol{v}^{(b)}_{t|t-1} \\ \boldsymbol{v}_t \end{bmatrix} \tag{46}$$

$$\boldsymbol{v}^{(b)}_{t+1|t} = \boldsymbol{v}^{(b)}_{t|t} - \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{w}_{t+1} \tag{47}$$

where $\boldsymbol{v}^{(b)}_{0|1} = [\ \ ]$. Since the observation and state evolution noises, $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$, are assumed independent, the statistics of $\boldsymbol{v}^{(b)}_{t|t}$ and $\boldsymbol{v}^{(b)}_{t+1|t}$ can be expressed as

$$\boldsymbol{\mu}^{(b)}_{t|t} = \begin{bmatrix} \boldsymbol{\mu}^{(b)}_{t|t-1} \\ \boldsymbol{\mu}^{(o)}_{t} \end{bmatrix} \tag{48}$$

$$\boldsymbol{\Sigma}^{(b)}_{t|t} = \begin{bmatrix} \boldsymbol{\Sigma}^{(b)}_{t|t-1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{(o)}_{t} \end{bmatrix} \tag{49}$$

$$\boldsymbol{\mu}^{(b)}_{t+1|t} = \boldsymbol{\mu}^{(b)}_{t|t} - \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\mu}^{(x)}_{t+1} \tag{50}$$

$$\boldsymbol{\Sigma}^{(b)}_{t+1|t} = \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\Sigma}^{(x)}_{t+1}(\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t} + \boldsymbol{\Sigma}^{(b)}_{t|t} \tag{51}$$

where $\boldsymbol{\mu}^{(b)}_{0|1} = [\ \ ]$ and $\boldsymbol{\Sigma}^{(b)}_{0|1} = [\ \ ]$.

The Fisher estimators [13] of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ given the observation sequence $\boldsymbol{o}_{1:t}$ define the recursions for the forward information filter and predictor covariance matrices, $\boldsymbol{\Sigma}^{-1}_{t|t}$ and $\boldsymbol{\Sigma}^{-1}_{t+1|t}$, as follows

$$\begin{aligned}
\boldsymbol{\Sigma}^{-1}_{t|t} &= \boldsymbol{C}^{(b)\prime}_{t|t}\boldsymbol{\Sigma}^{(b)-1}_{t|t}\boldsymbol{C}^{(b)}_{t|t} \\
&= \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_{t}\boldsymbol{C}_t + \boldsymbol{\Sigma}^{-1}_{t|t-1} \qquad (52)\\
\boldsymbol{\Sigma}^{-1}_{t+1|t} &= \boldsymbol{C}^{(b)\prime}_{t+1|t}\boldsymbol{\Sigma}^{(b)-1}_{t+1|t}\boldsymbol{C}^{(b)}_{t+1|t} \\
&= (\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t}\big(\boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\Sigma}^{(x)}_{t+1}(\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t} + \boldsymbol{\Sigma}^{(b)}_{t|t}\big)^{-1}\boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1} \\
&= \boldsymbol{\Sigma}^{(x)-1}_{t+1} - \boldsymbol{\Sigma}^{(x)-1}_{t+1}\big((\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t}\boldsymbol{\Sigma}^{(b)-1}_{t|t}\boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1} + \boldsymbol{\Sigma}^{(x)-1}_{t+1}\big)^{-1}\boldsymbol{\Sigma}^{(x)-1}_{t+1} \\
&= \boldsymbol{\Sigma}^{(x)-1}_{t+1} - \boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1}\big(\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{(x)-1}_{t+1} \qquad (53)
\end{aligned}$$

where $\boldsymbol{\Sigma}^{-1}_{1|0} = \boldsymbol{\Sigma}^{(i)-1}_{1}$ and the matrix inversion lemma in Equation 37 is used between the first and second line. The forward information filter and predictor mean vectors are

$$\begin{aligned}
\boldsymbol{\Sigma}^{-1}_{t|t}\boldsymbol{x}_{t|t} &= \boldsymbol{C}^{(b)\prime}_{t|t}\boldsymbol{\Sigma}^{(b)-1}_{t|t}(\boldsymbol{o}_{1:t} - \boldsymbol{\mu}^{(b)}_{t|t}) \\
&= \boldsymbol{\Sigma}^{-1}_{t|t-1}\boldsymbol{x}_{t|t-1} + \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_{t}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(o)}_{t}) \qquad (54)\\
\boldsymbol{\Sigma}^{-1}_{t+1|t}\boldsymbol{x}_{t+1|t} &= \boldsymbol{C}^{(b)\prime}_{t+1|t}\boldsymbol{\Sigma}^{(b)-1}_{t+1|t}(\boldsymbol{o}_{1:t} - \boldsymbol{\mu}^{(b)}_{t+1|t}) \\
&= (\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t}\big(\boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\Sigma}^{(x)}_{t+1}(\boldsymbol{A}^{-1}_{t+1})'\boldsymbol{C}^{(b)\prime}_{t|t} + \boldsymbol{\Sigma}^{(b)}_{t|t}\big)^{-1}(\boldsymbol{o}_{1:t} + \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\mu}^{(x)}_{t+1} - \boldsymbol{\mu}^{(o)}_{t}) \\
&= \boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1}\big(\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{C}^{(b)\prime}_{t|t}\boldsymbol{\Sigma}^{(b)-1}_{t|t}(\boldsymbol{o}_{1:t} + \boldsymbol{C}^{(b)}_{t|t}\boldsymbol{A}^{-1}_{t+1}\boldsymbol{\mu}^{(x)}_{t+1} - \boldsymbol{\mu}^{(o)}_{t}) \\
&= \boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1}\big(\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}^{(x)-1}_{t+1}\boldsymbol{A}_{t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{\Sigma}^{-1}_{t|t}\boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}^{-1}_{t+1|t}\boldsymbol{\mu}^{(x)}_{t+1} \qquad (55)
\end{aligned}$$

where $\boldsymbol{\Sigma}^{-1}_{1|0}\boldsymbol{x}_{1|0} = \boldsymbol{\Sigma}^{(i)-1}_{1}\boldsymbol{\mu}^{(i)}_{1}$ and the matrix identity in Equation 38 is used between the second and third line.

The information form of the Kalman filter can also be derived by applying the matrix inversion lemma in Equation 37 for the covariance form recursion. Analogously, the covariance form of the Kalman filter can be derived from the information form above.

B.2 Backward Information Filter

The derivation of backward information filter follows the same way as for the forward information filter above. The observations from time t to time T, $\boldsymbol{o}_{t:T}$, can be expressed in terms of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ as

$$\boldsymbol{o}_{t:T} = \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{x}_t + \boldsymbol{v}^{(f)}_{t|t} \tag{56}$$

$$\boldsymbol{o}_{t:T} = \boldsymbol{C}^{(f)}_{t-1|t}\boldsymbol{x}_{t-1} + \boldsymbol{v}^{(f)}_{t-1|t} \tag{57}$$

where the observation matrices, $\boldsymbol{C}^{(f)}_{t|t}$ and $\boldsymbol{C}^{(f)}_{t-1|t}$, are constructed as follows

$$\boldsymbol{C}^{(f)}_{t|t} = \begin{bmatrix} \boldsymbol{C}_t \\ \boldsymbol{C}^{(f)}_{t|t+1} \end{bmatrix} \tag{58}$$

$$\boldsymbol{C}^{(f)}_{t-1|t} = \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{A}_t \tag{59}$$

where $\boldsymbol{C}^{(f)}_{T|T+1} = [\quad]$; i.e., an empty matrix, and the observation noises, $\boldsymbol{v}^{(f)}_{t|t}$ and $\boldsymbol{v}^{(f)}_{t-1|t}$, are constructed as follows

$$\boldsymbol{v}^{(f)}_{t|t} = \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{v}^{(f)}_{t|t+1} \end{bmatrix} \tag{60}$$

$$\boldsymbol{v}^{(f)}_{t-1|t} = \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{w}_t + \boldsymbol{v}^{(f)}_{t|t} \tag{61}$$

where $\boldsymbol{v}^{(f)}_{T|T+1} = [\quad]$. Since the observation and state evolution noises, $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$, are assumed independent, the statistics of $\boldsymbol{v}^{(f)}_{t|t}$ and $\boldsymbol{v}^{(f)}_{t-1|t}$ can be expressed as

$$\boldsymbol{\mu}^{(f)}_{t|t} = \begin{bmatrix} \boldsymbol{\mu}^{(o)}_t \\ \boldsymbol{\mu}^{(f)}_{t|t+1} \end{bmatrix} \tag{62}$$

$$\boldsymbol{\Sigma}^{(f)}_{t|t} = \begin{bmatrix} \boldsymbol{\Sigma}^{(o)}_t & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}^{(f)}_{t|t+1} \end{bmatrix} \tag{63}$$

$$\boldsymbol{\mu}^{(f)}_{t-1|t} = \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\mu}^{(x)}_t + \boldsymbol{\mu}^{(f)}_{t|t} \tag{64}$$

$$\boldsymbol{\Sigma}^{(f)}_{t-1|t} = \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\Sigma}^{(x)}_t\boldsymbol{C}^{(f)\prime}_{t|t} + \boldsymbol{\Sigma}^{(f)}_{t|t} \tag{65}$$

where $\boldsymbol{\mu}^{(f)}_{T|T+1} = [\quad]$ and $\boldsymbol{\Sigma}^{(f)}_{T|T+1} = [\quad]$.

The Fisher estimators [13] of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ given the observation sequence $\boldsymbol{o}_{t:T}$ define the recursions for the backward information filter and predictor covariance matrices, $\boldsymbol{P}^{-1}_{t|t}$ and $\boldsymbol{P}^{-1}_{t-1|t}$, [16] as follows

$$\begin{aligned} \boldsymbol{P}^{-1}_{t|t} &= \boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}\boldsymbol{C}^{(f)}_{t|t} \\ &= \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_t\boldsymbol{C}_t + \boldsymbol{P}^{-1}_{t|t+1} \end{aligned} \tag{66}$$

$$\begin{aligned} \boldsymbol{P}^{-1}_{t-1|t} &= \boldsymbol{C}^{(f)\prime}_{t-1|t}\boldsymbol{\Sigma}^{(f)-1}_{t-1|t}\boldsymbol{C}^{(f)}_{t-1|t} \\ &= \boldsymbol{A}'_t\boldsymbol{C}^{(f)\prime}_{t|t}\big(\boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\Sigma}^{(x)}_t\boldsymbol{C}^{(f)\prime}_{t|t} + \boldsymbol{\Sigma}^{(f)}_{t|t}\big)^{-1}\boldsymbol{C}^{(f)}_{t|t}\boldsymbol{A}_t \\ &= \boldsymbol{A}'_t\boldsymbol{\Sigma}^{(x)-1}_t\big(\boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}\boldsymbol{C}^{(f)}_{t|t} + \boldsymbol{\Sigma}^{(x)-1}_t\big)^{-1}\boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}\boldsymbol{C}^{(f)}_{t|t}\boldsymbol{A}_t \\ &= \boldsymbol{A}'_t\big(\boldsymbol{P}^{-1}_{t|t}\boldsymbol{\Sigma}^{(x)}_t + \boldsymbol{I}\big)^{-1}\boldsymbol{P}^{-1}_{t|t}\boldsymbol{A}_t \end{aligned} \tag{67}$$

where $\boldsymbol{P}^{-1}_{T|T+1} = \boldsymbol{0}$ and the matrix identity in Equation 38 is used between the second and third line. The backward information filter and predictor mean vectors are

$$\begin{aligned} \boldsymbol{P}^{-1}_{t|t}\boldsymbol{m}_{t|t} &= \boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}(\boldsymbol{o}_{t:T} - \boldsymbol{\mu}^{(f)}_{t|t}) \\ &= \boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{m}_{t|t+1} + \boldsymbol{C}'_t\boldsymbol{\Sigma}^{(o)-1}_t(\boldsymbol{o}_t - \boldsymbol{\mu}^{(o)}_t) \end{aligned} \tag{68}$$

$$\begin{aligned} \boldsymbol{P}^{-1}_{t-1|t}\boldsymbol{m}_{t-1|t} &= \boldsymbol{C}^{(f)\prime}_{t-1|t}\boldsymbol{\Sigma}^{(f)-1}_{t-1|t}(\boldsymbol{o}_{t:T} - \boldsymbol{\mu}^{(f)}_{t-1|t}) \\ &= \boldsymbol{A}'_t\boldsymbol{C}^{(f)\prime}_{t|t}\big(\boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\Sigma}^{(x)}_t\boldsymbol{C}^{(f)\prime}_{t|t} + \boldsymbol{\Sigma}^{(f)}_{t|t}\big)^{-1}(\boldsymbol{o}_{t:T} - \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\mu}^{(x)}_t - \boldsymbol{\mu}^{(f)}_{t|t}) \\ &= \boldsymbol{A}'_t\boldsymbol{\Sigma}^{(x)-1}_t\big(\boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}\boldsymbol{C}^{(f)}_{t|t} + \boldsymbol{\Sigma}^{(x)-1}_t\big)^{-1}\boldsymbol{C}^{(f)\prime}_{t|t}\boldsymbol{\Sigma}^{(f)-1}_{t|t}(\boldsymbol{o}_{t:T} - \boldsymbol{C}^{(f)}_{t|t}\boldsymbol{\mu}^{(x)}_t - \boldsymbol{\mu}^{(f)}_{t|t}) \\ &= \boldsymbol{A}'_t\big(\boldsymbol{P}^{-1}_{t|t}\boldsymbol{\Sigma}^{(x)}_t + \boldsymbol{I}\big)^{-1}\boldsymbol{P}^{-1}_{t|t}(\boldsymbol{m}_{t|t} - \boldsymbol{\mu}^{(x)}_t) \end{aligned} \tag{69}$$

where $\boldsymbol{P}^{-1}_{T|T+1}\boldsymbol{m}_{T|T+1} = \boldsymbol{0}$.

B.3 Two Filter Formulae for Kalman Smoothing

Sometimes it is beneficial to estimate the smoothed statistics in two independent sweeps. Since a covariance form of backward Kalman filter in general would require the knowledge of the state noise statistics from time 1 to t [13], the information form has to be used for the backward sweep. In terms of the current state vector, $\boldsymbol{x}_t$, the observation sequence, $\boldsymbol{O}$, can be factored as follows

$$\underbrace{\begin{bmatrix} \boldsymbol{o}_{1:t-1} \\ \boldsymbol{o}_{t:T} \end{bmatrix}}_{\boldsymbol{O}} = \underbrace{\begin{bmatrix} \boldsymbol{C}^{(b)}_{t|t-1} \\ \boldsymbol{C}^{(f)}_{t|t} \end{bmatrix}}_{\boldsymbol{C}^{(s)}_t} \boldsymbol{x}_t + \underbrace{\begin{bmatrix} \boldsymbol{v}^{(b)}_{t|t-1} \\ \boldsymbol{v}^{(f)}_{t|t} \end{bmatrix}}_{\boldsymbol{v}^{(s)}_t} \tag{70}$$

where the parameters $\boldsymbol{C}^{(b)}_{t|t-1}$, $\boldsymbol{v}^{(b)}_{t|t-1}$, $\boldsymbol{C}^{(f)}_{t|t}$ and $\boldsymbol{v}^{(f)}_{t|t}$ are defined as in Appendices B.1 and B.2. It should be noted that the observation noises $\boldsymbol{v}^{(b)}_{t|t-1}$ and $\boldsymbol{v}^{(f)}_{t|t}$ are independent. The fisher estimator [13] of the state vector, $\boldsymbol{x}_t$, given the observation sequence, $\boldsymbol{O}$, can be expressed in terms of the parameters $\boldsymbol{C}^{(s)}_t$ and $\boldsymbol{\Sigma}^{(s)}_t$ as follows

$$\hat{\boldsymbol{\Sigma}}_t^{-1} = \boldsymbol{C}^{(s)\prime}_t \boldsymbol{\Sigma}^{(s)-1}_t \boldsymbol{C}^{(s)}_t = \boldsymbol{C}^{(b)\prime}_{t|t-1} \boldsymbol{\Sigma}^{(b)-1}_{t|t-1} \boldsymbol{C}^{(b)}_{t|t-1} + \boldsymbol{C}^{(f)\prime}_{t|t} \boldsymbol{\Sigma}^{(f)-1}_{t|t} \boldsymbol{C}^{(f)}_{t|t} = \boldsymbol{\Sigma}^{-1}_{t|t-1} + \boldsymbol{P}^{-1}_{t|t} \tag{71}$$

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_t^{-1} \hat{\boldsymbol{x}}_t &= \boldsymbol{C}^{(s)\prime}_t \boldsymbol{\Sigma}^{(s)-1}_t (\boldsymbol{O} - \boldsymbol{\mu}^{(s)}_t) \\ &= \boldsymbol{C}^{(b)\prime}_{t|t-1} \boldsymbol{\Sigma}^{(b)-1}_{t|t-1} (\boldsymbol{o}_{1:t-1} - \boldsymbol{\mu}^{(b)}_{t|t-1}) + \boldsymbol{C}^{(f)\prime}_{t|t} \boldsymbol{\Sigma}^{(f)-1}_{t|t} (\boldsymbol{o}_{t:T} - \boldsymbol{\mu}^{(f)}_{t|t}) \\ &= \boldsymbol{\Sigma}^{-1}_{t|t-1} \boldsymbol{x}_{t|t-1} + \boldsymbol{P}^{-1}_{t|t} \boldsymbol{m}_{t|t} \end{aligned} \tag{72}$$

The smoothed statistics can be expressed as

$$\hat{\boldsymbol{\Sigma}}_t = (\boldsymbol{\Sigma}^{-1}_{t|t-1} + \boldsymbol{P}^{-1}_{t|t})^{-1} \tag{73}$$

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{\Sigma}}_t (\boldsymbol{\Sigma}^{-1}_{t|t-1} \boldsymbol{x}_{t|t-1} + \boldsymbol{P}^{-1}_{t|t} \boldsymbol{m}_{t|t}) \tag{74}$$

Analogously, the smoothed statistics can be derived using the forward filtered estimates, $\boldsymbol{\Sigma}_{t|t}$, and backward predicted estimates, $\boldsymbol{P}_{t|t+1}$, as follows

$$\hat{\boldsymbol{\Sigma}}_t = (\boldsymbol{\Sigma}^{-1}_{t|t} + \boldsymbol{P}^{-1}_{t|t+1})^{-1} \tag{75}$$

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{\Sigma}}_t (\boldsymbol{\Sigma}^{-1}_{t|t} \boldsymbol{x}_{t|t} + \boldsymbol{P}^{-1}_{t|t+1} \boldsymbol{m}_{t|t+1}) \tag{76}$$

The equivalence of the RTS and the two filter smoother can be verified using the principle of mathematical induction. Initially, it holds for the covariance matrix that

$$\hat{\boldsymbol{\Sigma}}_T = \boldsymbol{\Sigma}_{T|T} = (\boldsymbol{\Sigma}^{-1}_{T|T} + \boldsymbol{P}^{-1}_{T|T+1})^{-1} \tag{77}$$

because $\boldsymbol{P}^{-1}_{T|T+1} = \boldsymbol{0}$. Assuming that $\hat{\boldsymbol{\Sigma}}_{t+1} = (\boldsymbol{\Sigma}^{-1}_{t+1|t} + \boldsymbol{P}^{-1}_{t+1|t+1})^{-1}$, the RTS smoother covariances can be converted into the two filter smoother covariances as follows

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_t &= \boldsymbol{\Sigma}_{t|t} + \boldsymbol{\Sigma}_{t|t} \boldsymbol{A}'_{t+1} \boldsymbol{\Sigma}^{-1}_{t+1|t} \big(\hat{\boldsymbol{\Sigma}}_{t+1} - \boldsymbol{\Sigma}_{t+1|t}\big) \boldsymbol{\Sigma}^{-1}_{t+1|t} \boldsymbol{A}_{t+1} \boldsymbol{\Sigma}_{t|t} \\ &= \boldsymbol{\Sigma}_{t|t} + \boldsymbol{\Sigma}_{t|t} \boldsymbol{A}'_{t+1} \boldsymbol{\Sigma}^{-1}_{t+1|t} \big((\boldsymbol{\Sigma}^{-1}_{t+1|t} + \boldsymbol{P}^{-1}_{t+1|t+1})^{-1} - \boldsymbol{\Sigma}_{t+1|t}\big) \boldsymbol{\Sigma}^{-1}_{t+1|t} \boldsymbol{A}_{t+1} \boldsymbol{\Sigma}_{t|t} \\ &= \boldsymbol{\Sigma}_{t|t} + \boldsymbol{\Sigma}_{t|t} \boldsymbol{A}'_{t+1} \big((\boldsymbol{\Sigma}_{t+1|t} + \boldsymbol{\Sigma}_{t+1|t} \boldsymbol{P}^{-1}_{t+1|t+1} \boldsymbol{\Sigma}_{t+1|t})^{-1} - \boldsymbol{\Sigma}^{-1}_{t+1|t}\big) \boldsymbol{A}_{t+1} \boldsymbol{\Sigma}_{t|t} \\ &= \boldsymbol{\Sigma}_{t|t} - \boldsymbol{\Sigma}_{t|t} \boldsymbol{A}'_{t+1} \big(\boldsymbol{\Sigma}_{t+1|t} + \boldsymbol{P}_{t+1|t+1}\big)^{-1} \boldsymbol{A}_{t+1} \boldsymbol{\Sigma}_{t|t} \\ &= \boldsymbol{\Sigma}_{t|t} - \boldsymbol{\Sigma}_{t|t} \big(\boldsymbol{\Sigma}_{t|t} + \boldsymbol{P}_{t|t+1}\big)^{-1} \boldsymbol{\Sigma}_{t|t} \\ &= \big(\boldsymbol{\Sigma}^{-1}_{t|t} + \boldsymbol{P}^{-1}_{t|t+1}\big)^{-1} \end{aligned} \tag{78}$$

where the matrix inversion lemma in Equation 37 is used between the third and fourth line, and between the fifth and sixth line. Also, the identities $\boldsymbol{\Sigma}_{t+1|t} = \boldsymbol{A}_{t+1} \boldsymbol{\Sigma}_{t|t} \boldsymbol{A}'_{t+1} + \boldsymbol{\Sigma}^{(x)}_{t+1}$ and $\boldsymbol{P}_{t+1|t+1} = \boldsymbol{A}_{t+1} \boldsymbol{P}_{t|t+1} \boldsymbol{A}'_{t+1} - \boldsymbol{\Sigma}^{(x)}_{t+1}$ are used between lines four and five.

The RTS smoother mean vectors can be converted into their two filter equivalents the same way. Initially, it holds that

$$\hat{\boldsymbol{x}}_T = \boldsymbol{x}_{T|T} = \boldsymbol{\Sigma}_{T|T}(\boldsymbol{\Sigma}_{T|T}^{-1}\boldsymbol{x}_{T|T} + \boldsymbol{P}_{T|T+1}^{-1}\boldsymbol{m}_{T|T+1}) \tag{79}$$

because $\boldsymbol{P}_{T|T+1}^{-1}\boldsymbol{m}_{T|T+1} = \boldsymbol{0}$. Assuming $\hat{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{\Sigma}}_{t+1}(\boldsymbol{\Sigma}_{t+1|t}^{-1}\boldsymbol{x}_{t+1|t} + \boldsymbol{P}_{t+1|t+1}^{-1}\boldsymbol{m}_{t+1|t+1})$ and knowing that $\hat{\boldsymbol{\Sigma}}_{t+1} = (\boldsymbol{\Sigma}_{t+1|t}^{-1} + \boldsymbol{P}_{t+1|t+1}^{-1})^{-1}$, the RTS smoother mean vectors can be written as

$$\begin{aligned}
\hat{\boldsymbol{x}}_t &= \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}_{t+1|t}^{-1}(\hat{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_{t+1|t}) \\
&= \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}_{t+1|t}^{-1}\big(\hat{\boldsymbol{\Sigma}}_{t+1}(\boldsymbol{\Sigma}_{t+1|t}^{-1}\boldsymbol{x}_{t+1|t} + \boldsymbol{P}_{t+1|t+1}^{-1}\boldsymbol{m}_{t+1|t+1}) - \boldsymbol{x}_{t+1|t}\big) \\
&= \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\boldsymbol{\Sigma}_{t+1|t}^{-1}\big((\boldsymbol{\Sigma}_{t+1|t}^{-1} + \boldsymbol{P}_{t+1|t+1}^{-1})^{-1}(\boldsymbol{\Sigma}_{t+1|t}^{-1}\boldsymbol{x}_{t+1|t} + \boldsymbol{P}_{t+1|t+1}^{-1}\boldsymbol{m}_{t+1|t+1}) - \boldsymbol{x}_{t+1|t}\big) \\
&= \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\boldsymbol{A}'_{t+1}\big(\boldsymbol{\Sigma}_{t+1|t} + \boldsymbol{P}_{t+1|t+1}\big)^{-1}(\boldsymbol{m}_{t+1|t+1} - \boldsymbol{x}_{t+1|t}) \\
&= \boldsymbol{x}_{t|t} + \boldsymbol{\Sigma}_{t|t}\big(\boldsymbol{\Sigma}_{t|t} + \boldsymbol{P}_{t|t+1}\big)^{-1}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t}) \\
&= \hat{\boldsymbol{\Sigma}}_t(\boldsymbol{\Sigma}_{t|t}^{-1}\boldsymbol{x}_{t|t} + \boldsymbol{P}_{t|t+1}^{-1}\boldsymbol{m}_{t|t+1})
\end{aligned} \tag{80}$$

where the identities $\boldsymbol{x}_{t+1|t} = \boldsymbol{A}_{t+1}\boldsymbol{x}_{t|t} + \boldsymbol{\mu}_{t+1}^{(x)}$ and $\boldsymbol{m}_{t+1|t+1} = \boldsymbol{A}_{t+1}\boldsymbol{m}_{t|t+1} + \boldsymbol{\mu}_{t+1}^{(x)}$ are used between the fourth and fifth lines. All the other matrix manipulations are similar to the covariance derivations above.

C Proposal Distribution for Gibbs Sampling

The proposal distribution used in Rao-Blackwellised Gibbs sampling for switching linear dynamical systems can be expressed as

$$\begin{aligned}
&P(q_t|\boldsymbol{O}, q_{-t}) \propto \\
&\quad P(q_{t+1}|q_t)P(q_t|q_{t-1})p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, q_{1:t}) \int p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t})d\boldsymbol{x}_t
\end{aligned} \tag{81}$$

Using the parameters $\boldsymbol{C}_{t|t+1}^{(f)}$, $\boldsymbol{\mu}_{t|t+1}^{(f)}$ and $\boldsymbol{\Sigma}_{t|t+1}^{(f)}$ for the distribution $p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})$ as in Appendix B.2, the above integral can be written as

$$\begin{aligned}
&\int p(\boldsymbol{o}_{t+1:T}|\boldsymbol{x}_t, q_{t+1:T})p(\boldsymbol{x}_t|\boldsymbol{o}_{1:t}, q_{1:t})d\boldsymbol{x}_t = \\
&\quad \int \mathcal{N}(\boldsymbol{o}_{t+1:T}; \boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{x}_t + \boldsymbol{\mu}_{t|t+1}^{(f)}, \boldsymbol{\Sigma}_{t|t+1}^{(f)})\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_{t|t}, \boldsymbol{\Sigma}_{t|t})d\boldsymbol{x}_t \\
&= \mathcal{N}(\boldsymbol{o}_{t+1:T}; \boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{x}_{t|t} + \boldsymbol{\mu}_{t|t+1}^{(f)}, \boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{\Sigma}_{t|t}\boldsymbol{C}_{t|t+1}^{(f)\prime} + \boldsymbol{\Sigma}_{t|t+1}^{(f)})
\end{aligned} \tag{82}$$

Using the backward information filter variables from Appendix B.2 and the identity for determinants in Equation 39, the determinant of the covariance matrix in the above Gaussian can be expressed as

$$\begin{aligned}
\big|\boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{\Sigma}_{t|t}\boldsymbol{C}_{t|t+1}^{(f)\prime} + \boldsymbol{\Sigma}_{t|t+1}^{(f)}\big| &= \big|\boldsymbol{\Sigma}_{t|t+1}^{(f)}\big|\big|\boldsymbol{\Sigma}_{t|t}\big|\big|\boldsymbol{C}_{t|t+1}^{(f)\prime}\boldsymbol{\Sigma}_{t|t+1}^{(f)-1}\boldsymbol{C}_{t|t+1}^{(f)} + \boldsymbol{\Sigma}_{t|t}^{-1}\big| \\
&= \big|\boldsymbol{\Sigma}_{t|t+1}^{(f)}\big|\big|\boldsymbol{\Sigma}_{t|t}\boldsymbol{P}_{t|t+1}^{-1} + \boldsymbol{I}\big|
\end{aligned} \tag{83}$$

where $\big|\boldsymbol{\Sigma}_{t|t+1}^{(f)}\big|$ does not depend on q_t.

Using the matrix inversion lemma in Equation 37, the inverse covariance matrix of the Gaussian in Equation 82 can be written as

$$\begin{aligned}
&(\boldsymbol{C}_{t|t+1}^{(f)}\boldsymbol{\Sigma}_{t|t}\boldsymbol{C}_{t|t+1}^{(f)\prime} + \boldsymbol{\Sigma}_{t|t+1}^{(f)})^{-1} = \\
&\quad \boldsymbol{\Sigma}_{t|t+1}^{(f)-1} - \boldsymbol{\Sigma}_{t|t+1}^{(f)-1}\boldsymbol{C}_{t|t+1}^{(f)}\big(\boldsymbol{C}_{t|t+1}^{(f)\prime}\boldsymbol{\Sigma}_{t|t+1}^{(f)-1}\boldsymbol{C}_{t|t+1}^{(f)} + \boldsymbol{\Sigma}_{t|t}^{-1}\big)^{-1}\boldsymbol{C}_{t|t+1}^{(f)\prime}\boldsymbol{\Sigma}_{t|t+1}^{(f)-1} \\
&= \boldsymbol{\Sigma}_{t|t+1}^{(f)-1} - \boldsymbol{\Sigma}_{t|t+1}^{(f)-1}\boldsymbol{C}_{t|t+1}^{(f)}\big(\boldsymbol{P}_{t|t+1}^{-1} + \boldsymbol{\Sigma}_{t|t}^{-1}\big)^{-1}\boldsymbol{C}_{t|t+1}^{(f)\prime}\boldsymbol{\Sigma}_{t|t+1}^{(f)-1}
\end{aligned} \tag{84}$$

Writing the Gaussian as $Z\exp(-\frac{1}{2}\gamma)$ where Z is a constant. The term γ can be expressed as

$$\begin{aligned}
\gamma &= (\boldsymbol{o}_{t+1:T} - \boldsymbol{C}^{(f)}_{t|t+1}\boldsymbol{x}_{t|t} - \boldsymbol{\mu}^{(f)}_{t|t+1})'\big(\boldsymbol{\Sigma}^{(f)-1}_{t|t+1} - \boldsymbol{\Sigma}^{(f)-1}_{t|t+1}\boldsymbol{C}^{(f)}_{t|t+1}\big(\boldsymbol{P}^{-1}_{t|t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{C}^{(f)\prime}_{t|t+1}\boldsymbol{\Sigma}^{(f)-1}_{t|t+1}\big) \\
&\quad \times(\boldsymbol{o}_{t+1:T} - \boldsymbol{C}^{(f)}_{t|t+1}\boldsymbol{x}_{t|t} - \boldsymbol{\mu}^{(f)}_{t|t+1}) \\
&= (\boldsymbol{o}_{t+1:T} - \boldsymbol{\mu}^{(f)}_{t|t+1})'\boldsymbol{\Sigma}^{(f)-1}_{t|t+1}(\boldsymbol{o}_{t+1:T} - \boldsymbol{\mu}^{(f)}_{t|t+1}) + \boldsymbol{x}'_{t|t}\boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{x}_{t|t} - 2\boldsymbol{x}'_{t|t}\boldsymbol{P}^{-1}_{t|t+1}\boldsymbol{m}_{t|t+1} \\
&\quad -(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})'\boldsymbol{P}^{-1}_{t|t+1}\big(\boldsymbol{P}^{-1}_{t|t+1} + \boldsymbol{\Sigma}^{-1}_{t|t}\big)^{-1}\boldsymbol{P}^{-1}_{t|t+1}(\boldsymbol{m}_{t|t+1} - \boldsymbol{x}_{t|t})
\end{aligned} \tag{85}$$

where $(\boldsymbol{o}_{t+1:T} - \boldsymbol{\mu}^{(f)}_{t|t+1})'\boldsymbol{\Sigma}^{(f)-1}_{t|t+1}(\boldsymbol{o}_{t+1:T} - \boldsymbol{\mu}^{(f)}_{t|t+1})$ does not depend on q_t. Discarding all terms independent on q_t, Equation 27 results.