Switching Linear Dynamical Systems for Noise Robust Speech Recognition

Bertrand Mesot & David Barber

IDIAP–RR 06–08

April 2007

to appear in
IEEE Transactions on Audio, Speech and Language Processing

IDIAP Research Report 06–08

Switching Linear Dynamical Systems for Noise Robust Speech Recognition

Bertrand Mesot & David Barber

April 2007

to appear in
IEEE Transactions on Audio, Speech and Language Processing

Abstract. Real world applications such as hands-free dialling in cars may have to deal with potentially very noisy environments. Existing state-of-the-art solutions to this problem use feature-based HMMs, with a preprocessing stage to clean the noisy signal. However, the effect that raw signal noise has on the induced HMM features is poorly understood, and limits the performance of the HMM system. An alternative to feature-based HMMs is to model the raw signal, which has the potential advantage that including an explicit noise model is straightforward. Here we jointly model the dynamics of both the raw speech signal *and* the noise, using a Switching Linear Dynamical System (SLDS). The new model was tested on isolated digit utterances corrupted by Gaussian noise. Contrary to the Autoregressive HMM and its derivatives, which provides a model of uncorrupted raw speech, the SLDS is comparatively noise robust and also significantly outperforms a state-of-the-art feature-based HMM. The computational complexity of the SLDS scales exponentially with the length of the time series. To counter this we use Expectation Correction which provides a stable and accurate linear-time approximation for this important class of models, aiding their further application in acoustic modelling.

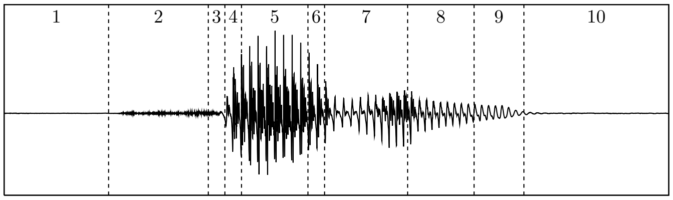


Figure 1: Example of a state segmentation given by a SAR-HMM on the digit "seven" taken from the TI-DIGITS database. The switch state is shown on top of each segment.

1 Introduction

Current state-of-the-art automatic speech recognition (ASR) systems use the framework of feature-based *Hidden Markov Models* (HMMs) [23]. Whilst successful under controlled conditions, this standard approach is often particularly fragile in the presence of noise [17]. This important issue is commonly addressed by a preprocessing step which attempts to remove noise, see for example [17], [7], [12], [25] and [1]. The explicit influence of the noise on the features extracted is poorly understood and hence incorporating noise models directly into standard feature-based HMM approach would be difficult. An alternative strategy is to model the raw acoustic signal directly which has the potential advantage that the noise may also be explicitly modelled.

The early work of Poritz [22] and the more recent *Switching Autoregressive HMM* (SAR-HMM) introduced by Ephraim and Roberts [11] have shown that, for isolated digit recognition in clean conditions, modelling the raw speech signal directly can be a reasonable alternative to feature-based HMMs. The basic idea behind the SAR-HMM is to model the speech signal as an autoregressive (AR) process. The intrinsic non-stationarity of the speech signal is dealt with by switching between a finite set of AR models (with different parameters), see Figure 1. Whilst the SAR-HMM has comparable to state-of-the-art performance on clean speech, this degrades rapidly under noisy conditions. A possible explanation for this undesirable behaviour is that the AR process is defined on the (potentially noisy) observed signal directly; since the model forms predictions on the basis of past observations, the recognition accuracy of the SAR-HMM drops significantly if the speech signal is corrupted with noise.

To deal with noise, without having to train a new model, we extend the SAR-HMM to include an explicit noise process whereby the observed signal is viewed as a corrupted version of a clean *hidden* signal. This approach naturally leads to a *Switching Linear Dynamical System* (SLDS) [3] which represents the signal as a piecewise linear hidden variable model. This approach enhances noise robustness since the switching AR process is defined on a hidden clean counterpart of the noisy signal. Here we will make the simple assumption of independent Gaussian noise, although the method may be extended to include more complex noise processes.

Contrary to the SAR-HMM, where inferring the posterior of the hidden variables can be carried out using a standard forward-backward algorithm, inference is formally intractable in the SLDS [3], scaling exponentially with the length of the speech utterance. Arguably, this has been the fundamental reason why the powerful class of SLDS models has found relatively little support amongst the automatic speech recognition (ASR) community. Two well-known methods for performing approximate inference in the SLDS are *Expectation Propagation* (EP) [20] and *Generalised Pseudo Bayes* (GPB) [3, 21]. They both suffer from limitations which can be relaxed in the case of the SLDS—see [4] for a detailed explanation. To overcome limitations in existing approximate inference procedures, we recently introduced the *Expectation Correction* (EC) algorithm [4] which provides a stable, accurate approximation and scales well to large applications such as ASR.

Previous applications of the SLDS to ASR (see, for example, [10, 26]), have modelled the feature vectors, and not the raw signal directly. However, work in acoustic modelling [8] suggests that, provided the difficulties of performing inference and learning can be addressed, the SLDS is a potentially powerful tool for modelling the raw acoustic dynamics. In the following section, we rephrase the SAR-HMM and discuss our 'correction' approach for performing inference. This model will subsequently be extended to a noise-robust version by construction of a suitable SLDS. We then compare the SAR-HMM, SLDS and a state-of-the-art feature based noise reduction method for recognising isolated digits from the TI-DIGITS database [18]. Our contribution, which consists of making a *joint* model of both the *raw* speech and noise signals offers improved noise robustness against the other methods. Furthermore, the formal computational limitations of exactly implementing the SLDS are well-addressed using Expectation Correction.

2 The SAR-HMM

One of the simplest models of a continuous time series is an AR process. However, due to the intrinsic non-stationarity of the speech signal, using a fixed set of AR parameters for the whole signal is too restrictive. The SAR-HMM [11] therefore introduces a discrete switch variable s_t, for each time t, which can be in one of S different states, each corresponding to a particular setting of the AR parameters. The switch state is assumed Markovian with transition probability $p(s_t \,|\, s_{t-1})$. Given a particular switch state s_t, the model assumes that the observed sample v_t at time t is a linear combination of the R preceding observations plus a Gaussian distributed innovation $\eta(s_t)$:

$$v_t = -\sum_{r=1}^{R} c_r(s_t)\, v_{t-r} + \eta_t \quad \text{with} \quad \eta_t \sim \mathcal{N}(0, \sigma^2_{s_t}) \tag{1}$$

where $\eta \sim \mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distributed random variable with mean μ and (co)variance σ^2. Probabilistically, this may be written as[1]

$$p(v_t \,|\, v_{t-R:t-1}, s_t) \propto \exp\left\{ -\frac{1}{2\sigma^2_{s_t}} \left(v_t + \textstyle\sum_r c_r(s_t)\, v_{t-r} \right)^2 \right\}$$

The role of the innovation η_t is to model variations in the speech signal from pure autoregression, and does *not* model a separate independent additive noise process. In cases where the signal is indeed inherently noisy, the predictions of the SAR-HMM would depend directly on previous noisy observations, limiting the suitability of the SAR-HMM in noisy environments. Nevertheless, the SAR-HMM serves as a baseline raw-signal model, which we extend to include an explicit noise model in Section 3.

For a sequence of samples $v_{1:T}$ of length T, the SAR-HMM defines the joint distribution

$$p(s_{1:T}, v_{1:T}) = p(v_1 \,|\, s_1)\, p(s_1) \prod_{t=2}^{T} p(v_t \,|\, v_{t-R:t-1}, s_t)\, p(s_t \,|\, s_{t-1}) \tag{2}$$

This is a form of Dynamical Bayesian Network (DBN) [14], whose structure is given in Figure 2. The initial part of the series lacks sufficient observations, and hence we define

$$p(v_t \,|\, v_{t-R:t-1}, s_t) \equiv p(v_t \,|\, v_{1:t-1}, s_t) \quad \text{when} \quad 1 < t \le R.$$

To ensure that switching between the different AR models is not too rapid, the model is constrained to stay an integer multiple of K time steps in the same state. To achieve this, the transition probability in Equation (2) is modified as

$$\begin{cases} p(s_t \,|\, s_{t-1}) & \text{if } t = 0 \mod K \\ 1 & \text{if } t > 0 \mod K \text{ and } s_t = s_{t-1} \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

[1] The notation $z_{t_1:t_2}$ refers to the sequence $z_{t_1}, \ldots, z_{t_2}$.

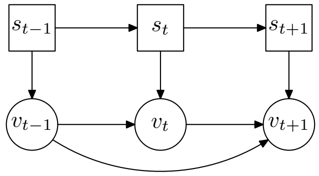


Figure 2: Dynamic Bayesian network representation of the SAR-HMM; s_t represents the discrete hidden switch variable and v_t is the observed value of the sample at time t.

In practice, the transition probability $p(s_t \,|\, s_{t-1})$ is usually defined such that the switch state number cannot decrease[2]. With such a setting, the signal is split into a fixed sequence of segments of variable length, each modelled by a separate AR process (see Figure 1). Despite the apparent complexity of the SAR-HMM, the model remains a specially constrained version of an HMM, for which inference is computationally straightforward, scaling linearly with T [23].

2.1 Gain Adaptation

To cope with variations of the energy contour between utterances and across speakers, the variances σ_s^2 of the innovation need to be adapted to each utterance. This procedure, known has *Gain Adaptation*, is a key component of the SAR-HMM and considerably improves recognition accuracy. The state variance σ_s^2 is thus replaced by the segment-state variance σ_{ns}^2—we refer here to the segmentation induced by the modified transition probability (3)—which maximises the likelihood of the observed signal. For segment n and state s, we desire to find the variance which maximises the segment log-likelihood

$$\log p_{ns}(v_{t_n:t_n+K}) = \sum_{t=t_n}^{t_n+K} \log p(v_t \,|\, v_{t-R:t-1}, s_t)\big|_{s_t=s}$$

which is achieved by setting

$$\sigma_{ns}^2 = \frac{1}{K} \sum_{t=t_n}^{t_n+K} \left(v_t + \textstyle\sum_r c_r(s_t) v_{t-r}\right)^2 \tag{4}$$

where t_n is the time point at which the segment n begins[3].

2.2 Inference and Learning

Following [11], we evaluate the performance of the SAR-HMM on an isolated digit recognition task from the TI-DIGITS database [18]. This is achieved by training a separate SAR-HMM for each of the eleven digits (0–9 and "oh") using the EM algorithm [9] on a set of training utterances. Recognition is then performed by associating the utterance to the digit whose model has the highest likelihood.

For a single sequence, given the current setting of the SAR-HMM parameters ϑ, the M-step of EM maximises the expected complete log-likelihood[4] [9]

$$\Big\langle \log p\big(s_{1:T}, v_{1:T} \,|\, \hat{\vartheta}\big) \Big\rangle_{p(s_{1:T} \,|\, v_{1:T}, \vartheta)} \tag{5}$$

with respect to the new parameter setting $\hat{\vartheta}$. The updated AR coefficients $\hat{c}_i(s)$ are given by[5]

$$\left[\begin{array}{cccc} \hat{c}_1(s) & \hat{c}_2(s) & \ldots & \hat{c}_r(s) \end{array}\right]^{\mathsf{T}} = -B^{-1}(s)\, d(s)$$

[2] This is often called a *left-right transition matrix* in the HMM jargon

[3] At the end of the waveform, the final segment will generally consist of fewer than K time points, and the upper bound of the sum in Equation 4 needs to be modified accordingly.

[4] $\langle\cdot\rangle_p$ denotes the average with respect to the distribution p.

[5] $^{\mathsf{T}}$ denotes the matrix transpose.

where $B(s)$ and $d(s)$ are given by

$$\begin{aligned}
[B(s)]_{ij} &= \sum_t v_{t-i}\, v_{t-j}\, p(s_t \,|\, v_{1:T})\big|_{s_t=s} \\
[d(s)]_j &= \sum_t v_t\, v_{t-j}\, p(s_t \,|\, v_{1:T})\big|_{s_t=s}.
\end{aligned}$$

For multiple sequences, the above are summed over all sequences. In order to calculate the expected complete log-likelihood (5), we need to infer the marginal posterior distributions $p(s_t \,|\, v_{1:T})$. Inference in chain-structured distributions, such as the HMM, is generally achieved by the forward-backward algorithm [6]. In the SAR-HMM however, the standard backward pass is more complicated because of the forward dependencies between the observations (Figure 2). We therefore consider a different scheme based on a correction smoother [24] where the backward pass calculates directly the posterior $p(s_t \,|\, v_{1:T})$ by *correcting* the result of the forward pass. This method forms the basis of our Expectation Correction method used for the more complex SLDS, and hence serves as a useful introduction.

2.2.1 Forward Pass

The goal of the forward pass is to calculate, for each time step t, the 'filtered' posterior $p(s_t \,|\, v_{1:t})$ which contains all the information coming from the past. By using the structure of the distribution (2), if the previous filtered posterior $p(s_{t-1} \,|\, v_{1:t-1})$ is known, then the current posterior can be found by recursion:

$$\begin{aligned}
p(s_t \,|\, v_{1:t}) &\propto p(s_t, v_t | v_{1:t-1}) \\
&= \sum_{s_{t-1}} p(v_t \,|\, v_{t-R:t-1}, s_t)\, p(s_t \,|\, s_{t-1})\, p(s_{t-1} \,|\, v_{1:t-1}). \qquad (6)
\end{aligned}$$

Starting with the initial posterior $p(s_1 \,|\, v_1) \propto p(v_1 \,|\, s_1)\, p(s_1)$, the filtered posterior at each time step can then be found by applying Equation 6 iteratively.

2.2.2 Backward Pass

The goal of the backward pass is to calculate, for each time step t, the smoothed posterior $p(s_t \,|\, v_{1:T})$. A recursion for $p(s_t \,|\, v_{1:T})$ in terms of $p(s_{t+1} \,|\, v_{1:T})$ can be derived by considering an equation similar to that used by the *Rauch-Tung-Striebel* (RTS) correction smoother for the *Kalman Filter* [24]:

$$\begin{aligned}
p(s_t \,|\, v_{1:T}) &= \sum_{s_{t+1}} p(s_t \,|\, s_{t+1}, v_{1:T})\, p(s_{t+1} \,|\, v_{1:T}) \\
&= \sum_{s_{t+1}} p(s_t \,|\, s_{t+1}, v_{1:t})\, p(s_{t+1} \,|\, v_{1:T}) \qquad (7) \\
&\propto \sum_{s_{t+1}} p(s_{t+1} \,|\, s_t)\, p(s_t \,|\, v_{1:t})\, p(s_{t+1} \,|\, v_{1:T}) \qquad (8)
\end{aligned}$$

where the second and third terms in (8) are the filtered posterior at time t and the smoothed posterior at time $t+1$ respectively. In (7), we used the fact that, in chain-structured graphs like the one of Figure 2, s_t is independent of any future information $v_{t+1:T}$ if the state of s_{t+1} is known. The iteration (8) is initialised with the last filtered posterior $p(s_T \,|\, v_{1:T})$. Furthermore, since the transition distribution $p(s_t \,|\, s_{t-1})$ is replaced by (3), the smoothed posterior remains the same over a segment and therefore only needs to be computed at segment boundaries.

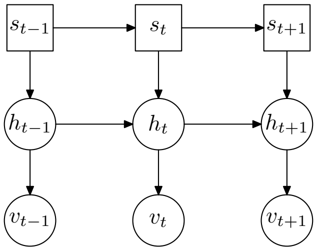


Figure 3: Dynamic Bayesian network representing the AR-SLDS: s_t represents the discrete hidden switch variable, h_t the continuous hidden clean signal and v_t is the observed value of the sample at time t.

2.2.3 Likelihood

The likelihood of an observed sequence $v_{1:T}$ can be calculated using the recursion

$$\begin{aligned} p(v_{1:t}) &= p(v_{1:t-1})\, p(v_t \,|\, v_{1:t-1}) \\ &= p(v_{1:t-1}) \sum_{s_t} p(s_t, v_t \,|\, v_{1:t-1}) \end{aligned} \tag{9}$$

where $p(s_t, v_t \,|\, v_{1:t-1})$ is given by the right hand side (rhs) of Equation 6 and $p(v_{1:t-1})$ is the previous partial likelihood. The recursion (9) is initialised with $p(v_1) = \sum_{s_1} p(v_1 \,|\, s_1)\, p(s_1)$.

3 The AR-SLDS

The SAR-HMM is a useful model of clean raw-speech, but is fragile in the presence of noise. To overcome some of the limitations of the SAR-HMM, we introduce the AR-SLDS which considers the observed speech sample v_t as a noisy version of a clean hidden sample. The clean one-dimensional signal is obtained from the projection of a higher dimensional vector h_t, whose dynamics follows a stochastic linear recursion, parameterised by s_t[6]:

$$h_t = A(s_t)h_{t-1} + \eta_t^{\mathcal{H}} \quad \text{with} \quad \eta_t^{\mathcal{H}} \sim \mathcal{N}\big(0, \Sigma_{\mathcal{H}}(s_t)\big) \tag{10}$$

Here $A(s_t)$ is the transition matrix which characterises the dynamics of the hidden variable, under state s_t. The 'innovation' (or hidden noise) $\eta_t^{\mathcal{H}}$ models variations from pure linear state dynamics. Equation 10 defines a continuous hidden transition distribution $p(h_t \,|\, h_{t-1}, s_t)$ proportional to

$$\exp\left\{-\frac{1}{2}\big(h_t - A(s_t)h_{t-1}\big)^{\mathsf{T}} \Sigma_{\mathcal{H}}^{-1}(s_t)\big(h_t - A(s_t)h_{t-1}\big)\right\}.$$

The observation is given by projecting the vector h_t to a scalar v_t:

$$v_t = Bh_t + \eta_t^{\mathcal{V}} \quad \text{with} \quad \eta_t^{\mathcal{V}} \sim \mathcal{N}(0, \sigma_{\mathcal{V}}^2). \tag{11}$$

Here the noise $\eta_t^{\mathcal{V}}$ models independent additive Gaussian white noise on the clean signal Bh_t. Unlike the innovation η_t in the SAR-HMM (Equation 1), $\eta_t^{\mathcal{V}}$ models noise on the signal, independently of the dynamics of the clean signal. Equation 11 corresponds to the Gaussian distribution

$$p(v_t \,|\, h_t) \propto \exp\left\{-\frac{1}{2\sigma_{\mathcal{V}}^2}\big(v_t - Bh_t\big)^2\right\}.$$

[6] In order to keep the notation simple, we use $\mathcal{H}$ and $\mathcal{V}$ to indicate if the variable/parameter is associated with a hidden or visible variable respectively.

For a sequence of samples $v_{1:T}$ of length T, the AR-SLDS defines the joint distribution

$$p(h_{1:T}, s_{1:T}, v_{1:T}) = p(v_1 \mid h_1)\, p(h_1 \mid s_1)\, p(s_1) \prod_{t=2}^{T} p(v_t \mid h_t)\, p(h_t \mid h_{t-1}, s_t)\, p(s_t \mid s_{t-1}) \qquad (12)$$

where $p(s_1)$ and $p(h_1 \mid s_1)$ are prior distributions over the discrete and continuous variables and $p(s_t \mid s_{t-1})$ is the state transition distribution. The model forms a first order Markovian dynamics on the hidden space, whose graphical structure is depicted in Figure 3. The SLDS models both the dynamics of a clean underlying signal, plus independent additive noise. It is this joint signal plus noise modelling which we hope will bring benefit over the simpler SAR-HMM.

The model presented so far is generic and it would be interesting to see what potential performance it has on ASR. However, to demonstrate possible improvement in noise robustness using the SLDS over the SAR-HMM, we construct a specific SLDS which, when $\sigma_\mathcal{V}^2 \equiv 0$, mimics the SAR-HMM. To do this we set $A(s_t)$ to be an $R \times R$ matrix where the first row contains the AR coefficients $-c_r(s_t)$ and the rest is a shifted down identity matrix. For example, in the case of a third order AR process, we would have

$$A(s_t) = \begin{bmatrix} -c_1(s_t) & -c_2(s_t) & -c_3(s_t) \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The $1 \times R$ projection matrix B extracts the first component of h_t:

$$B = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}.$$

Since the SAR-HMM innovation only influences the most recent observation, the hidden covariance matrix $\Sigma_\mathcal{H}(s)$ must be set such that all elements are zero, except the top-left most which is set to σ_s^2, i.e., the innovation variance used in Equation 1. In this way, the model exactly mimics the SAR-HMM if $\sigma_\mathcal{V}^2 \equiv 0$. For $\sigma_\mathcal{V}^2 > 0$, the model is effectively an SAR-HMM model of clean speech, *plus* a model of additive Gaussian white noise, and should thus provide a level of noise robustness. As for the SAR-HMM, we use the likelihood of an utterance to perform classification. Since Gain Adaptation is a key component of a successful SAR-HMM system, we include this also for the SLDS, as explained below.

3.1 Gain Adaptation

We perform gain adaptation in the AR-SLDS by adjusting the innovation covariance $\Sigma_\mathcal{H}(s)$ to each utterance. Following the same approach as for the SAR-HMM, we replace the hidden covariance $\Sigma_\mathcal{H}(s)$ by the segment-state hidden covariance matrix $\Sigma_\mathcal{H}^{ns}$ that maximises the likelihood of the observed sequence. An explicit formula for $\Sigma_\mathcal{H}^{ns}$ cannot be obtained; we use EM instead to estimate that quantity. Given the current estimate of $\Sigma_\mathcal{H}^{ns}$, the M-step of EM maximises the expected complete log-likelihood

$$\left\langle \log p\big(s_{1:T}, h_{1:T}, v_{1:T} \mid \hat{\Sigma}_\mathcal{H}^{ns}\big) \right\rangle_{p(s_{1:T}, h_{1:T} \mid v_{1:T}, \Sigma_\mathcal{H}^{ns})}$$

with respect to the new parameters $\hat{\Sigma}_\mathcal{H}^{ns}$. This yields the update

$$\hat{\Sigma}_\mathcal{H}^{ns} = \frac{1}{K} \sum_{t=t_n}^{t_n+K} \left\langle \big(h_t - A(s)h_{t-1}\big)\big(h_t - A(s)h_{t-1}\big)^\mathsf{T} \right\rangle \qquad (13)$$

where the average is taken with respect to the posterior $p(h_{t-1}, h_t \mid s_t, v_{1:T}, \Sigma_\mathcal{H}^{ns})|_{s_t=s}$. The posterior distribution required by Equations 13 is obtained by inference on the distribution (12), conditioned on $v_{1:T}$. Contrary to the SAR-HMM, where the posterior distribution $p(s_t \mid v_{1:T})$ can be computed exactly, inferring $p(s_t, h_t \mid v_{1:T})$ in a SLDS is $\mathcal{O}(S^T)$, and therefore requires approximations [3]. To address this we recently developed the *Expectation Correction* algorithm [4] which is a generic algorithm for approximate inference in SLDSs, as briefly described in the next section.

3.2 Inference using Expectation Correction

The EC algorithm approximates the smoothed posteriors $p(s_t, h_t \,|\, v_{1:T})$ in two steps: the forward pass first finds the filtered posteriors $p(s_t, h_t \,|\, v_{1:t})$ and the backward pass corrects the filtered estimate to form the smoothed posterior $p(s_t, h_t \,|\, v_{1:T})$. Both passes are linear in T, compared with the $\mathcal{O}(S^T)$ complexity of exact inference.

Without loss of generality, we may represent the filtered and smoothed posteriors as a product of a continuous and a discrete distribution:

$$\begin{aligned} p(s_t, h_t \,|\, v_{1:t}) &= p(h_t \,|\, s_t, v_{1:t})\, p(s_t \,|\, v_{1:t}) \\ p(s_t, h_t \,|\, v_{1:T}) &= p(h_t \,|\, s_t, v_{1:T})\, p(s_t \,|\, v_{1:T}). \end{aligned}$$

Space here is too limited to provide more than a cursory explanation of the algorithm and the reader is referred to [4] for further details. The procedure presented below is not specific to the AR-SLDS, whose particular structure allows some computational savings, as described in Appendix A.

3.2.1 Forward Pass

If we denote $x_t = \{h_t, s_t\}$ as the hidden variables, the generic form of the forward recursion is

$$p(x_t \,|\, v_{1:t}) \propto p(v_t \,|\, x_t) \int_{x_{t-1}} p(x_t \,|\, x_{t-1})\, p(x_{t-1} \,|\, v_{1:t-1})$$

Using the structure of the DBN of Figure 3, the rhs can be expanded as

$$\sum_{s_{t-1}} p(s_t \,|\, s_{t-1})\, p(s_{t-1} \,|\, v_{1:t-1})\, p(v_t \,|\, h_t) \int_{h_{t-1}} p(h_t \,|\, h_{t-1}, s_t)\, p(h_{t-1} \,|\, s_{t-1}, v_{1:t-1}).$$

The filtered posterior at time t is therefore a mixture of Gaussians of the form

$$\sum_{s_{t-1}} p(h_t \,|\, s_{t-1}, s_t, v_{1:t})\, p(s_{t-1}, s_t \,|\, v_{1:t}) \tag{14}$$

with

$$\begin{aligned} p(s_{t-1}, s_t \,|\, v_{1:t}) &= p(v_t \,|\, s_t, v_{1:t-1})\, p(s_{t-1} \,|\, v_{1:t-1}) && (15) \\ p(h_t \,|\, s_{t-1}, s_t, v_{1:t}) &= p(v_t \,|\, h_t) \left\langle p(h_t \,|\, h_{t-1}, s_t) \right\rangle && (16) \end{aligned}$$

where the average is taken with respect to the filtered posterior $p(h_{t-1} \,|\, s_{t-1}, v_{1:t-1})$ and $p(v_t \,|\, s_t, v_{1:t-1})$ is obtained by integrating the rhs of Equation 16 over h_t. The recursion is initialised with

$$p(h_1 \,|\, s_1, v_1) \propto p(v_1 \,|\, h_1)\, p(h_1 \,|\, s_1) \quad \text{and} \quad p(s_1 \,|\, v_1) \propto \int_{h_1} p(v_1 \,|\, h_1)\, p(h_1|s_1)\, p(s_1)$$

where $p(s_1)$ is the discrete prior, and $p(h_1 \,|\, s_1)$ is the indexed Gaussian prior on the continuous hidden state. The number of mixture components required to represent the filtered posterior exactly is multiplied by S at each time step and thus grows exponentially with t. A simple remedy is to collapse the mixture obtained at each time step to a mixture with fewer components. This corresponds to the so-called *Gaussian Sum Approximation* [2] which is a form of *Assumed Density Filtering* [20]. For the experiments presented in this paper we simply collapsed the mixture to a single Gaussian, which proved sufficiently accurate.

3.2.2 Backward Pass

The backward recursion is similar to that used in the RTS method. It has the generic form

$$p(x_t \,|\, v_{1:T}) = \int_{x_{t+1}} p(x_t \,|\, x_{t+1}, v_{1:t})\, p(x_{t+1} \,|\, v_{1:T}).$$

Note that the first factor is independent of the future observations $v_{t+1:T}$ because it is conditioned on x_{t+1}. Expanding the rhs yields

$$p(h_t, s_t \,|\, v_{1:T}) = \sum_{s_{t+1}} p(s_{t+1} \,|\, v_{1:T}) \left\langle p(h_t, s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t}) \right\rangle \tag{17}$$

where the average is taken with respect to the smoothed posterior $p(h_{t+1} \,|\, s_{t+1}, v_{1:T})$. Without loss of generality the average in Equation 17 can be written as

$$\left\langle p(h_t \,|\, h_{t+1}, s_t, s_{t+1}, v_{1:t})\, p(s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t}) \right\rangle.$$

This is difficult to evaluate because of the dependency between h_{t+1} and s_{t+1}. In EC, the average is approximated by

$$\left\langle p(h_t \,|\, h_{t+1}, s_t, s_{t+1}, v_{1:t}) \right\rangle \left\langle p(s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t}) \right\rangle. \tag{18}$$

The first factor corresponds to the continuous part of the smoothed posterior distribution. Its form is the same as in the RTS method and can be evaluated exactly by conditioning on h_{t+1} the joint distribution

$$p(h_t, h_{t+1} \,|\, s_t, s_{t+1}, v_{1:t}) = p(h_{t+1} \,|\, h_t, s_{t+1})\, p(h_t \,|\, s_t, v_{1:t}) \tag{19}$$

which can be obtained by forward propagation.

The second factor in (18) is still difficult to evaluate exactly. The simplest approach within EC is to approximate it by

$$p(s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t})\big|_{h_{t+1} = \langle h_{t+1} \,|\, s_{t+1}, v_{1:T} \rangle} \tag{20}$$

where $\langle h_{t+1} \,|\, s_{t+1}, v_{1:T} \rangle$ is the mean of h_{t+1} with respect to the smoothed posterior $p(h_{t+1} \,|\, s_{t+1}, v_{1:T})$. More sophisticated approximation schemes may be applied, but practically the proposed one has proven to be accurate enough for the application considered here. Note also that this approximation is less severe than that used in the GPB backward pass, where

$$\left\langle p(s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t}) \right\rangle \approx p(s_t \,|\, s_{t+1}, v_{1:t}).$$

This approximation, proposed by Kim [15, 16], depends only on the filtered posterior and does not include any information coming from the continuous variable h_{t+1}. Finally, Expression 20 can be evaluated by considering

$$p(s_t, h_{t+1} \,|\, s_{t+1}, v_{1:t}) \propto p(h_{t+1} \,|\, s_t, s_{t+1}, v_{1:t})\, p(s_{t+1} \,|\, s_t)\, p(s_t \,|\, v_{1:t})$$

where $p(h_{t+1} \,|\, s_t, s_{t+1}, v_{1:t})$ is obtained by marginalising Equation 19 over h_t.

In summary, the smoothed posterior, as given by Equation 17, is a mixture of Gaussians of the form

$$\sum_{s_{t+1}} p(h_t \,|\, s_t, s_{t+1}, v_{1:T})\, p(s_t, s_{t+1} \,|\, v_{1:T}). \tag{21}$$

In its most generic form, EC approximates each term by

$$\begin{aligned} p(s_t, s_{t+1} \,|\, v_{1:T}) &\approx p(s_{t+1} \,|\, v_{1:T}) \left\langle p(s_t \,|\, h_{t+1}, s_{t+1}, v_{1:t}) \right\rangle \\ p(h_t \,|\, s_t, s_{t+1}, v_{1:T}) &\approx \left\langle p(h_t \,|\, h_{t+1}, s_t, s_{t+1}, v_{1:t}) \right\rangle \end{aligned}$$

where the average is taken with respect to the smoothed posterior $p(h_{t+1} \,|\, s_{t+1}, v_{1:T})$. As in the forward pass, the number of mixture components is multiplied by S at each iteration. In EC, we therefore collapse the mixture (21) to a mixture with fewer components. For the experiments presented in this paper, collapsing to a single Gaussian proved to be sufficient.

4 Training & Evaluation

4.1 The SAR-HMM

Following [11], we trained a separate SAR-HMM model for each of the eleven digits (0–9 and "oh") from the TI-DIGITS database [18]. The training set for each digit was composed of 110 single digit utterances downsampled to 8 kHz, each one pronounced by a male speaker. Each SAR-HMM was composed of ten states with a left-right transition matrix. Each state was associated with a 10th-order AR process and the model was constrained to stay an integer multiple of $K = 140$ time steps (0.0175 seconds) in the same state. The number of parameters to be trained was therefore: 10 AR coefficients per state and 9 transition probabilities. This makes a total of 109 free parameters, without counting the innovation variance which is implicitly obtained from each utterance.

For each training utterance the adapted gain σ^2_{ns} associated to each pair of segment and state was computed according to Equation 4. After this procedure had been carried out for each utterance of the training set, the parameters of the model—i.e, the transition matrix and the AR coefficients of each state—were updated and a new iteration took place if:

$$\frac{\sum_m \big(\log p(v^m_{1:T_m} \,|\, \hat{\vartheta}) - \log p(v^m_{1:T_m} \,|\, \vartheta)\big)}{\sum_m \log p(v^m_{1:T_m} \,|\, \hat{\vartheta})} > 10^{-7}$$

where $v^m_{1:T_m}$ represents the m-th utterance and ϑ and $\hat{\vartheta}$ the old and new sets of SAR-HMM parameters, respectively. The model was then evaluated on a test set composed of 112 utterances of each of the eleven digits, each pronounced by a different male speaker from that used in the training set. For each test utterance and for each model, the adapted gain σ^2_{ns} was computed and used to evaluate the likelihood of the sequence given the model. The recognition was then performed by selecting the model with the highest likelihood.

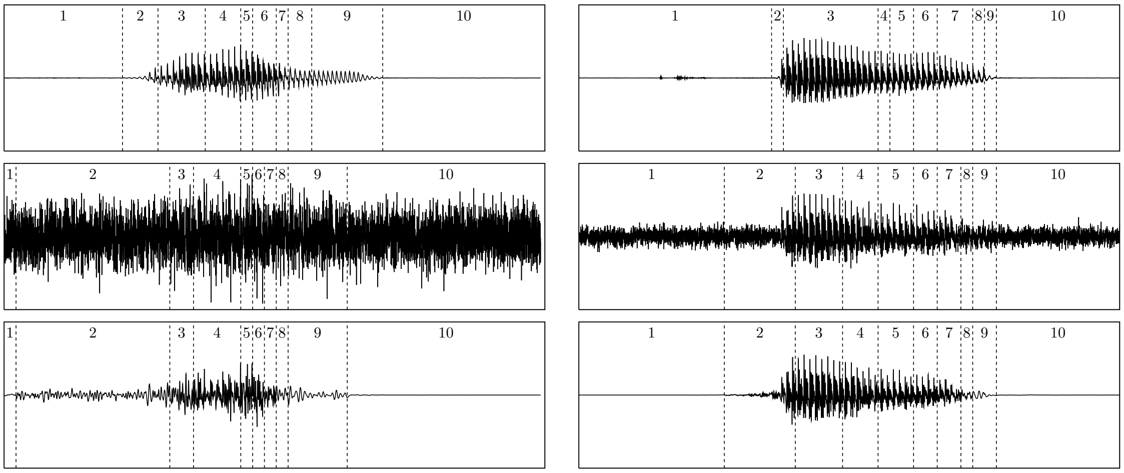
4.2 The AR-SLDS

The AR-SLDS was not trained directly. Instead its parameters were simply set to the same value as in the corresponding trained SAR-HMM, i.e., the AR coefficients $c_r(s)$ are copied into the first row of the matrix $A(s)$ and the same state transition distribution $p(s_t \,|\, s_{t-1})$ is used. The model was then tested on the same test set as used for the SAR-HMM. The innovation covariance was iteratively adapted using Equations 13 until the relative likelihood difference between two consecutive iterations was less than 10^{-7}. To seed recursion 13, an initial estimate of $\Sigma^{ns}_{\mathcal{H}}$ was obtained from the training set by using the SAR-HMM maximum likelihood estimate of σ^2_s for each state s

$$\sigma^2_s = \frac{\sum_{m,t} p(s_t = s \,|\, v^m_{1:T_m})\big(v_t + \sum_r c_r(s) v_{t-r}\big)^2}{\sum_{m,t} p(s_t = s \,|\, v^m_{1:T_m})}.$$

The initial $\Sigma^{ns}_{\mathcal{H}}$ was defined to have all elements equal to zero except for the top-left element which was set to σ^2_s, thus disregarding the segment number n. Compared to the SAR-HMM, the AR-SLDS has one additional parameter, the noise variance $\sigma^2_{\mathcal{V}}$.

The complexity of the EC forward and backward pass is $\mathcal{O}(SR^2T)$ and $\mathcal{O}(SR^3T)$ respectively. The backward pass is slightly more complex because, at each segment boundary, three matrix inversions are required and the complexity of the matrix multiplications cannot be reduced as in the forward pass. The total number of bytes required by the forward pass is $4 \cdot S\big(T + TR + \lfloor T/K \rfloor R^2\big)$. The total number of bytes used by the backward pass is $4 \cdot S\big(\lfloor T/K \rfloor + 2\,TR + T(2R)^2\big)$. As an example, the total amount of space required to evaluate a sequence of 7000 samples with a ten state 10th-order AR-SLDS is about 300 MB and the time required to evaluate all of the 1232 digits of the test set with *only one model* is around 3 days on a 3.2 GHz Pentium 4 machine.



"one" at SNR 0.7 dB

"five" at SNR 10.6 dB

Figure 4: Two examples of signal reconstruction using the AR-SLDS; (top) original clean signal taken from the TI-DIGITS database, (middle) noisy signal, i.e., clean signal artificially corrupted by Gaussian noise, (bottom) reconstructed clean signal. The dashed lines and the numbers show the most-likely state segmentation.

5 Examples of Signal Reconstruction

In order to demonstrate the noise robustness capabilities of the AR-SLDS, we plotted in Figure 4: (top) the original raw speech signal taken from the TI-DIGITS database and down-sampled to 8 kHz, (middle) its artificial Gaussian white noise corrupted version and (bottom) the corresponding reconstructed clean speech signal of the AR-SLDS. The latter is obtained by taking, for each time step, the mean $\langle h_t \rangle$ of the smoothed posterior $p(h_t \,|\, s'_t, v_{1:T})$ where the state segmentation is given by $s'_t = \arg\max_{s_t} p(s_t \,|\, v_{1:T})$. The clean reconstructed sample at time t is then given by $\langle h_t \rangle$. Figure 4 also shows, for each signal, the corresponding state segmentation given by the AR-SLDS.

In Figure 4 both noise-corrupted signals are correctly recognised by our SLDS procedure. This is encouraging since when the SNR is close to 0 dB, the shape of the original clean speech signal has almost disappeared and any de-noising method which does not consider the dynamics of the clean signal will most likely fail. In this example, the reconstructed signal is reminiscent of a digit "one"; for the higher SNR level on the right of Figure 4 the reconstruction is much closer to the original clean signal. The noisy "one" shown on the left side of Figure 4 has a likelihood of 2.0 when evaluated with the AR-SLDS corresponding to "one" and a likelihood of 1.9995 with the model corresponding to "oh". This demonstrates that, under extremely noisy conditions, an accurate approximation of the likelihood is important, since many digit models are likely to have generated such a noisy example. In both examples shown in Figure 4, the models stay in the second state for too long; this problem arises because the dynamics of the initial section of the speech signal is difficult to distinguish from silence. The performance could therefore be improved by explicitly modelling the state duration [8].

6 Results

For the TI-DIGITS database, we compared the noise robustness of the SAR-HMM against the AR-SLDS and a state-of-the-art de-noising method using a frequency domain feature-based HMM. Each test utterance was corrupted with additive noise independently sampled from a Gaussian with zero mean and covariance σ^2.

Noise Variance	SNR (dB)	HMM	SAR-HMM	AR-SLDS
0	26.5	100.0%	97.0%	96.8%
10^{-7}	26.3	100.0%	79.8%	96.8%
10^{-6}	25.1	*90.9%	56.7%	96.4%
10^{-5}	19.7	86.4%	22.2%	94.8%
10^{-4}	10.6	59.1%	9.7%	84.0%
10^{-3}	0.7	9.1%	9.1%	61.2%

Table 1: Comparison of the recognition accuracy of three models when the test utterances are corrupted by various levels of Gaussian noise. *This performance is worse than without unsupervised spectral subtraction, which gives 95.5%.

The features for the HMM were computed using *Unsupervised Spectral Subtraction* (USS) [17], thereby providing filtered features to the HMM recogniser. The setup used for the feature-based HMM was the same as that used to obtain the baseline performance on the AURORA task [13], namely 18 states, left-right transition matrix, a mixture of three Gaussians per state and 39 MFCC features, including first and second temporal derivatives as well as energy. The number of parameter to be trained was therefore: $18 \times 3 \times 39 = 2106$ mean values (one for each MFCC feature), 2106 variances (one for each MFCC feature) and 17 transition probabilities. This makes a total of 4229 parameters. For the SAR-HMM and the AR-SLDS, no prior filtering was applied. Each AR-SLDS digit model was explicitly tested with a range of noise variances and recognition was performed by picking the model with the highest likelihood.

Table 1 shows the recognition accuracies obtained by the different models for various levels of noise. As expected, the performance of the SAR-HMM rapidly decreases with noise. Thanks to USS, the feature-based HMM is able to maintain a recognition accuracy above 90% as long as the SNR is higher than 20 dB, below which the noise is too strong to be filtered out accurately without considering the dynamics of the clean signal. In contrast, the AR-SLDS has a recognition accuracy of 61.2% with a SNR close to 0 dB, whilst the performance of the other two methods is equivalent to random guessing (9.1%).

If all possible noise effects can be enumerated a priori – for example, if it is known that noise of variance either σ_1^2 or σ_2^2 is added to the signal – then an alternative would be to train the SAR-HMM on noisy versions of the clean utterances. Whilst of limited practical value[7], we carried out such an experiment as a comparative method of improving noise robustness in the SAR-HMM. To our surprise, the performance of the SAR-HMM trained with the same level of noise as that for which it was tested on, gave better results than the AR-SLDS initialised with the SAR-HMM trained on clean. As can be seen in Table 2, at SNR 0 dB, the accuracy is more than 10% higher than that of the AR-SLDS and significantly better otherwise. A possible explanation is that adding stationary Gaussian noise on the samples has a regularising effect which prevents overfitting. Since noise makes the signal less predictable, the AR coefficients obtained after training on noisy utterances are therefore more conservative than those obtained on clean, and tend to model the part of the signal which is the more stable. This explanation is plausible since the performance with a noise variance of 10^{-6} is actually better than with 10^{-7} and on clean speech. A Bayesian alternative to the SAR-HMM [19] may therefore be worthwhile considering.

7 Discussion and Conclusion

Our main goal was to investigate how much improvement we could expect by embedding a SAR-HMM *trained on clean signals* into a SLDS, and to present the underlying theoretical aspects. We

[7] In more practical scenarios the noise distributions are generally not stationary, nor even from a finite fixed set of possible distributions.

Noise Variance	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
Accuracy	97.2	97.4	96.3	92.53	74.2

Table 2: Recognition accuracy of the SAR-HMM trained and tested with the same noise variance.

concentrated on stationary noise, but non-stationary sources of noise may also be modelled, such as found in the AURORA [13] database. For example, we could model an observed noisy signal as the superposition of a clean hidden signal, modelled by an AR-SLDS, and noise, modelled by a *Bayesian Kalman Filter* [5]. The Bayesian Kalman filter is particularly useful in this situation because it allows the parameters of the noise model to be adapted automatically. This is important since the noise dynamics in practice would not be known a priori.

In summary, modelling signals as an explicit combination of speech and noise signals may be a viable route to noise reduction in speech recognition. Whilst we concentrated here on isolated digit recognition, it would be interesting to extend the approach to filter more generic speech units embedded in noisy signals, which then could be used for example as a preprocessing step in standard speech recognition models.

8 Code Availability

The code as well as the complete setup that we used during the preparation of this paper are available at the following address: `http://www.idiap.ch/~bmesot/arslds`.

A EC applied to the AR-SLDS

EC as described in Section 3.2 does not take into account the particular structure of the AR-SLDS, in particular the segmentation implied by the modified discrete transition probability (3). Inside a segment, the state does not change, and the sum over s_{t-1} in Equation 14 therefore disappears and one is left with the simpler expression

$$p(s_t, h_t \,|\, v_{1:t}) \propto \int_{h_{t-1}} p(v_t, s_t, h_t, s_{t-1}, h_{t-1} \,|\, v_{1:t-1})$$

which corresponds to a *Kalman Filter*. A nice property of the Kalman filter is that the variance of h_t does not depend on the observations $v_{1:t}$ and quickly converges to a fixed value [3]. This is useful in practice since the filtered covariance matrices used during the backward pass, which otherwise must be stored for each time-step, can be replaced by their segment converged approximations. Furthermore, conditioning (19) on h_{t+1} defines a reversal of the dynamics which requires a matrix inversion (see [4] for details). However, the inversion depends only on the filtered covariance matrix and on s_t and s_{t+1}. Using the same filtered covariance matrix over a whole segment, and the left-right structure of the transition matrix, this can be reduced to $3S$ inversions per segment.

B SNR Computation

The SNRs shown in Table 1 have been computed using the following formula:

$$10 \log_{10} \left(\frac{\sigma_s^2}{\sigma_n^2 + \sigma^2} \right)$$

where σ_s^2, σ_n^2 and σ^2 are the variance of the clean speech signal, clean noise and additional noise respectively. σ_s^2 was computed by retaining, for each utterance of the test set, only the samples whose energy was higher than 10% of the maximal energy. Those samples were assumed to belong to the speech signal. The remaining were used for computing σ_n^2.

Acknowledgement

We would like to thank Guillaume Lathoud for helpful comments on the manuscript.

References

[1] Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced frond-end feature extraction algorithm; compression algorithms. ETSI standard doc. ES 202 050 V1.1.1, October 2002.

[2] D. L. Alspach and H. W. Sorensen. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.

[3] Y. Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Norwood, MA, 1998.

[4] D. Barber. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, 7:2515–2540, November 2006.

[5] D. Barber and S. Chiappa. Unified inference for variational Bayesian linear Gaussian state-space models. In *Proceedings of NIPS 2006*, volume 20. To appear.

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

[7] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transaction on Acoustic, Speech and Signal Processing*, 27(2):113–120, April 1979.

[8] A. T. Cemgil, B. Kappen, and D. Barber. A generative model for music transcription. *IEEE Transactions on Speech and Audio Processing*, 2005. Accepted for future publication.

[9] A. P. Dempster. Maximum-likelihood from incomplete data. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[10] J. Droppo and A. Acero. Noise robust speech recognition with a switching linear dynamical model. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, May 2004.

[11] Y. Ephraim and W. J. J. Roberts. Revisiting autoregressive hidden Markov modeling of speech signals. *IEEE Signal Processing Letters*, 12(2):166–169, February 2005.

[12] H. Hermansky. RASTA processing of speech. *IEEE Transaction on Speech and Audio Processing*, 2(4):578–589, October 1994.

[13] H.-G. Hirsch and D. Pearce. The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ASR-2000*, pages 181–188, 2000.

[14] M.I. Jordan, editor. *Learning in Graphical Models*. MIT press, 1998.

[15] C.-J. Kim. Dynamic linear models with markov-switching. *Journal of Econometrics*, 60(1–2):1–22, 1994.

[16] C.-J. Kim and C. R. Nelson. *State-Space Models with Regime Switching*. MIT Press, 1999.

[17] G. Lathoud, M. Magimai-Doss, B. Mesot, and H. Bourlard. Unsupervised spectral subtraction for noise-robust ASR. In *Proceedings of ASRU 2005*, pages 189–194, November 2005.

[18] R.G. Leonard. A database for speaker independent digit recognition. In *Proceedings of ICASSP84*, volume 3, 1984.

[19] B. Mesot and D. Barber. A Bayesian alternative to gain adaptation in autoregressive hidden Markov models. In *Proceedings of ICASSP 2007*, April 2007. To appear.

[20] T. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, January 2001.

[21] K. Murphy. Learning switching Kalman filter models. Technical Report 98-10, Compaq Cambridge Research Lab, 1998.

[22] A. B. Poritz. Linear predictive hidden Markov models and the speech signal. In *Proceedings of the IEEE International Confererence on Acoustics, Speech, and Signal Processing*, volume 7, pages 1291–1294, May 1982.

[23] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), 1989.

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

[25] C. Ris and S. Dupont. Assessing local noise level estimation methods: Application to noise robust ASR. *Speech Communication*, 34(1–2):141–158, April 2001.

[26] A.-V. I. Rosti. *Linear Gaussian Models for Speech Recognition.* PhD thesis, University of Cambridge, EN, May 2004.