UNIVERSITY OF
CAMBRIDGE
Department of Engineering

Extended VTS for noise-robust speech recognition

R. C. van Dalen
rcv25@cam.ac.uk

M. J. F. Gales
mjfg@eng.cam.ac.uk

Technical Report CUED/F-INFENG/TR.636

1 September 2009

Abstract

Model compensation is a standard way of improving the robustness of speech recognition systems to noise. A number of popular schemes are based on vector Taylor series (VTS) compensation, which uses a linear approximation to represent the influence of noise on the clean speech. To compensate the dynamic parameters, the *continuous time approximation* is often used. This approximation uses a point estimate of the gradient, which fails to take into account that dynamic coefficients are a function of a number of consecutive static coefficients. In this paper, the accuracy of dynamic parameter compensation is improved by representing the dynamic features as a linear transformation of a window of static features. A modified version of VTS compensation is applied to the distribution of the window of static features and, importantly, their correlations. These compensated distributions are then transformed to distributions over standard static and dynamic features. With this improved approximation, it is also possible to obtain full-covariance corrupted speech distributions. This addresses the correlation changes that occur in noise. The proposed scheme outperformed the standard VTS scheme by 10 % to 20 % relative on a range of tasks.

1 Introduction

Changes in background noise conditions can severely impact the performance of speech recognition systems. Standard approaches to address this problem are to use either feature enhancement or model compensation techniques. The latter have been found to yield good performance, particularly in conditions with low signal-to-noise ratios, and will be the focus of this paper.

The first stage in developing a noise compensation scheme is to express how the noise affects the clean speech. When cepstral-based coefficients are used, the *mismatch function* between clean and noise-corrupted speech is non-linear. This non-linearity makes computing the exact distribution of the noise-corrupted speech intractable. There are a range of approximations that can be used to estimate the model parameters given the mismatch function [2, 6]. A commonly used method that has yielded good results approximates the mismatch function with a first-order vector Taylor series (VTS) expansion [17, 2]. Using this VTS approximation it is straightforward to compensate the parameters for the static parameters based on MFCC features. However, in HMM-based speech recognition systems dynamic features, for example delta and delta-delta coefficients, are appended to the static features to form the feature vector. A number of approaches to compensate parameters for these dynamic features have been proposed in the literature [7, 6, 3]. The standard is to use the continuous time approximation [7]. The continuous time approximation makes the assumption is that the dynamic coefficients are the time derivatives of the statics. The form of compensation for the dynamic parameters is then closely related to the static parameters. The continuous time approximation allows a mismatch function to be defined for any form of dynamic parameters, both those based on linear regression and simple differences. If only simple differences are considered then it is possible to find compensation by storing extra clean speech statistics [6]. This should be more precise than the continuous time approximation approach, but is only applicable to simple difference based dynamic parameters. Another scheme that attempts to improve compensation by using additional statistics, but in the log-spectral domain, is described in [3]. However, as section 3.3 will show, this approach involves approximations that negate any potential improvements and basically yields the same form as the continuous time approximation. Though there are known limitations to the use of the continuous time approximation it is still the form used in the vast majority of model-based compensation schemes [2, 15, 12].

This paper proposes a new approach for compensating the dynamic parameters that is applicable to both linear-regression and simple-difference based dynamic features. The dynamic coefficients can be expressed as a linear transformation over a window of static feature coefficients. The mismatch function for the static features can be used for each element of the window. Once the distribution over this "extended" feature vector is known then by linearly transforming the parameters of the distribution over the extended feature vector, the distribution of the static and dynamic parameters can be found. In the same fashion as standard model-based compensation there are a range of schemes that can be used to combine the extended clean speech and noise distributions together to yield the extended corrupted speech distribution. In this work an extended version of VTS is introduced. This approach will be referred to as extended VTS (EVTS).

Improving the compensation of dynamic parameters also addresses a second problem. Standard model compensation methods diagonalise the covariance matrices for

the corrupted speech distributions. This is consistent with the form of the clean speech distribution, which allows robust parameter estimation and efficient decoding. However, the correlations between the features are expected to change due to variations in the background noise. For example, in the limit, when the noise masks the speech, the correlation pattern will be that of the noise. These effects could be modelled with full-covariance compensation. Though continuous time approximation compensation can be used to generate block-diagonal covariance matrices (one each for the static delta and delta-delta features), these have not been used for recognition. In this work it is shown that the reason for this, ignoring the computational costs, is that compensation with the continuous time approximation is not accurate enough. Full covariance matrix corrupted speech distribution compensation, is expected to be more sensitive to approximations in the dynamic parameter compensation than diagonal compensation. However, extended VTS should be more accurate than VTS with the continuous time approximation, thus it should enable effective full-covariance matrix compensation. Though the use of these full-covariance matrices during decoding is computationally expensive, approaches such as predictive linear transforms [5] can be used. This paper only concentrates on the theoretical aspects of improved covariance compensation, rather than the computational load.

This paper introduces extended VTS and discusses how it can be used to generate full-covariance compensation. The organisation of this paper is as follows. The next section surveys model compensation techniques. Section 3 introduces extended VTS. Section 4 discusses how to find the clean speech and noise statistics. Section 5 examines the accuracy of compensation with standard VTS and extended VTS. Section 6 discusses experimental results on AURORA 2, a noise-corrupted Resource Management task, and an in-car recorded corpus.

2 Model compensation

To compute the effect of the acoustic noise on the feature vectors of a speech recogniser, an expression for the mismatch between clean and corrupted speech is needed. In the time domain, the additive noise n and the convolutional noise h transform the clean speech x, resulting in noise-corrupted speech y. In the time domain this will have the form [1]

$$y = h \star x + n \tag{1}$$

where $\star$ denotes convolution. Many speech recognition systems are based on MFCC features, which are in the cepstral domain. For time t, the static MFCC noise-corrupted speech vector will be denoted as $\mathbf{y}_t^s$. Similarly, MFCC vectors of the other features will be written: $\mathbf{x}_t^s$ for the clean speech; $\mathbf{n}_t^s$ for the additive noise; and $\mathbf{h}_t^s$ for the convolutional noise. The mismatch function that relates the static corrupted speech with the sources is [2]

$$\begin{aligned}\mathbf{y}_t^s &= \mathbf{C}\log\left(\exp\left(\mathbf{C}^{-1}\left(\mathbf{x}_t^s + \mathbf{h}_t^s\right)\right) + \exp\left(\mathbf{C}^{-1}\mathbf{n}_t^s\right)\right) \\ &= \mathbf{x}_t^s + \mathbf{h}_t^s + \mathbf{C}\log\left(\mathbf{1} + \exp\left(\mathbf{C}^{-1}\left(\mathbf{n}_t^s - \mathbf{x}_t^s - \mathbf{h}_t^s\right)\right)\right) \\ &= \mathbf{f}\left(\mathbf{x}_t^s, \mathbf{n}_t^s, \mathbf{h}_t^s\right),\end{aligned} \tag{2}$$

where $\mathbf{1}$ is a vector of 1s, and $\log(\cdot)$ and $\exp(\cdot)$ indicate the element-wise logarithm and exponent, respectively. The superscript s will be used throughout this paper to denote the static coefficients and parameters.

It is standard practice in HMM-based speech recognition systems to augment the observation vector containing per-time slice (static) features, with dynamic features [4]. They represent the change of the static features over time. Both first- and second-order features ($\mathbf{y}_t^{\Delta}, \mathbf{y}_t^{\Delta^2}$ respectively) are normally used. Hence the observation feature vector becomes $\mathbf{y}_t = \left[\, \mathbf{y}_t^{s\mathsf{T}} \; \mathbf{y}_t^{\Delta\mathsf{T}} \; \mathbf{y}_t^{\Delta^2\mathsf{T}} \,\right]^{\mathsf{T}}$. For clarity of presentation only first-order, delta, coefficients $\mathbf{y}^{\Delta}$ will be considered, though the extension to delta-delta parameters or higher orders is simple. Dynamic coefficients are normally computed with linear regression as the slope of the statics. The first-order dynamics at time t are computed from a window $\pm w$ of static coefficients [23]:

$$\mathbf{y}_t^{\Delta} = \frac{\sum_{\tau=1}^{w} \tau(\mathbf{y}_{t+\tau}^{s} - \mathbf{y}_{t-\tau}^{s})}{2\sum_{\tau=1}^{w} \tau^2}. \tag{3}$$

Model compensation schemes combine clean speech and noise distributions using the relationship between the clean and corrupted speech, to yield the parameters for the noise-corrupted speech model. The clean speech distributions are based on the HMM trained on clean speech data, with Gaussian components $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. In this work, as in many others, the static convolutional (channel) noise is assumed constant $\mathbf{h}_t^s = \boldsymbol{\mu}_h^s$, and the additive noise is assumed Gaussian-distributed $\mathbf{n}_t \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. These assumption allow the noise model to be estimated in a maximum likelihood fashion on test data [15, 12] (section 2.3 will discuss this in detail). Also, since the noise is assumed independent and identically distributed, each clean speech Gaussian can be compensated separately. In this work each noise-corrupted speech component is also assumed Gaussian. Thus the parameters of this Gaussian $\mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ are[1]

$$\boldsymbol{\mu}_y = \mathcal{E}\{\mathbf{y}\}; \qquad \boldsymbol{\Sigma}_y = \mathcal{E}\big\{(\mathbf{y} - \boldsymbol{\mu}_y)(\mathbf{y} - \boldsymbol{\mu}_y)^{\mathsf{T}}\big\}. \tag{4}$$

where the expectations are over the distribution of a component of the clean speech model and the noise distribution. The speech and noise are combined using the mismatch function in (2). However, no closed forms for the expectations in (4) exist, so approximations must be used. The next sections briefly discuss two options, vector Taylor series (VTS) and data-driven parallel model combination (DPMC).

2.1 Vector Taylor series compensation

The mismatch function in (2) can be approximated with a first-order vector Taylor series (VTS) [17]. The expansion point is normally set to the means of the clean speech and the noise. In that case, the approximated mismatch function for the static parameters becomes (assuming the convolutional noise is constant $\mathbf{h}_t^s = \boldsymbol{\mu}_h^s$)

$$\mathbf{y}_{t,\mathrm{vts}}^{s} = \mathbf{f}(\boldsymbol{\mu}_x^s, \boldsymbol{\mu}_n^s, \boldsymbol{\mu}_h^s) + \mathbf{J}(\mathbf{x}_t^s - \boldsymbol{\mu}_x^s) + (\mathbf{I} - \mathbf{J})(\mathbf{n}_t^s - \boldsymbol{\mu}_n^s), \tag{5}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{J}$ is the Jacobian of the clean speech

$$\mathbf{J} = \left.\frac{\partial \mathbf{y}^s}{\partial \mathbf{x}^s}\right|_{\boldsymbol{\mu}_n^s, \boldsymbol{\mu}_x^s, \boldsymbol{\mu}_h^s}, \tag{6}$$

[1] The dependence on the component has been dropped from the notation used in this paper for clarity.

which is a full matrix. If the mean of the clean speech is much smaller than the mean of the noise, $\mathbf{J}$ will tend to $\mathbf{I}$. Conversely, under high noise conditions, $\mathbf{J}$ will tend to $\mathbf{0}$ and $(\mathbf{I}-\mathbf{J})$ in (5) will tend to $\mathbf{I}$.

When the vector Taylor series approximation in (5) is applied to model compensation, the corrupted static mean and covariance of the compensated component become [2]

$$\mu_y^s = \mathcal{E}\{y_{t,\text{vts}}^s\} = f(\mu_x^s, \mu_n^s, \mu_h^s); \tag{7a}$$

$$\begin{aligned}\Sigma_y^s &= \mathcal{E}\left\{(y_{t,\text{vts}}^s - \mu_y)(y_{t,\text{vts}}^s - \mu_y)^T\right\} \\ &= \mathcal{E}\left\{J(x_t^s - \mu_x^s)(x_t^s - \mu_x^s)^T J^T + (\mathbf{I}-\mathbf{J})(n_t^s - \mu_n^s)(n_t^s - \mu_n^s)^T(\mathbf{I}-\mathbf{J})^T\right\} \\ &= J\Sigma_x^s J^T + (\mathbf{I}-\mathbf{J})\Sigma_n^s(\mathbf{I}-\mathbf{J})^T.\end{aligned} \tag{7b}$$

To compensate the dynamic parameters, the continuous time approximation [7] is often used in conjunction with VTS. This approximation assumes that delta coefficients are derivatives of static coefficients with respect to time t, so that

$$y_t^\Delta \approx \left.\frac{\partial y^s}{\partial t}\right|_t; \qquad x_t^\Delta \approx \left.\frac{\partial x^s}{\partial t}\right|_t; \qquad n_t^\Delta \approx \left.\frac{\partial n^s}{\partial t}\right|_t. \tag{8}$$

Combining this approximation and the VTS approximation in (5), the dynamic coefficients become

$$y_t^\Delta \approx \left.\frac{\partial y^s}{\partial t}\right|_t = \left.\frac{\partial y^s}{\partial x^s}\,\frac{\partial x^s}{\partial t}\right|_t + \left.\frac{\partial y^s}{\partial n^s}\,\frac{\partial n^s}{\partial t}\right|_t \approx J x_t^\Delta + (\mathbf{I}-\mathbf{J})n_t^\Delta. \tag{9}$$

These mismatch functions can be used to yield the dynamic mean and covariance of the corrupted speech. Since the additive noise is assumed to be stateless, the expected value of its dynamic coefficients is zero. The compensated dynamic parameters are given by

$$\mu_y^\Delta = \mathcal{E}\left\{J x_t^\Delta + (\mathbf{I}-\mathbf{J})n_t^\Delta\right\} = J\mu_x^\Delta; \tag{10a}$$

$$\begin{aligned}\Sigma_y^\Delta &= \mathcal{E}\left\{\left(J(x_t^\Delta - \mu_x^\Delta) + (\mathbf{I}-\mathbf{J})n_t^\Delta\right)\left(J(x_t^\Delta - \mu_x^\Delta) + (\mathbf{I}-\mathbf{J})n_t^\Delta\right)^T\right\} \\ &= \mathcal{E}\left\{J(x_t^\Delta - \mu_x^\Delta)(x_t^\Delta - \mu_x^\Delta)^T J^T\right\} + \mathcal{E}\left\{(\mathbf{I}-\mathbf{J})n_t^\Delta n_t^{\Delta T}(\mathbf{I}-\mathbf{J})^T\right\} \\ &= J\Sigma_x^\Delta J^T + (\mathbf{I}-\mathbf{J})\Sigma_n^\Delta(\mathbf{I}-\mathbf{J})^T.\end{aligned} \tag{10b}$$

The noise dynamic mean (both for the delta and delta-delta parameters) are assumed to be zero in the above expressions. This is consistent with the i.i.d. assumptions behind the noise.

The compensated mean and covariance matrix are formed by concatenating the static and dynamic parameters:

$$\mu_y = \begin{bmatrix}\mu_y^s \\ \mu_y^\Delta\end{bmatrix}; \qquad \Sigma_y = \begin{bmatrix}\Sigma_y^s & 0 \\ 0 & \Sigma_y^\Delta\end{bmatrix}. \tag{11}$$

The resulting covariance matrix, Σ_y, will be block-diagonal in structure as the Jacobian matrix, $\mathbf{J}$, is full. Decoding with this block-diagonal structure has two problems. First, it is computationally expensive. Second, the continuous time approximation for the dynamic parameters does not yield accurate block-diagonal compensation (section 5.1.2

will discuss this in more detail). Therefore, when decoding the following, standard, form for the output probability is used

$$p(\mathbf{y}_t) = \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_y, \text{diag}(\boldsymbol{\Sigma}_y)), \tag{12}$$

where $\text{diag}(\cdot)$ denotes matrix diagonalisation.

2.2 Data-driven parallel model combination

The relationship between the corrupted speech, and the clean speech and the noise, is not linear. The first-order vector Taylor series expansion is therefore only an approximation. Data-driven parallel model combination (DPMC) is a Monte Carlo method for estimating the distribution of the corrupted speech [6]. The procedure it uses for the static coefficients is straightforward. From the distributions of the clean speech and the additive noise, K samples of the clean speech, $\mathbf{x}^{s(1)}, \ldots, \mathbf{x}^{s(K)}$, and noise, $\mathbf{n}^{s(1)}, \ldots, \mathbf{n}^{s(K)}$, are drawn. For each pair of samples $(\mathbf{x}^{s(k)}, \mathbf{n}^{s(k)})$, the mismatch function in (2) gives the corresponding corrupted speech sample $\mathbf{y}^{s(k)}$:

$$\mathbf{y}^{s(k)} = \mathbf{f}\left(\mathbf{x}_t^{s(k)}, \mathbf{n}_t^{s(k)}, \mathbf{h}^s\right). \tag{13}$$

The static parameters for the corrupted speech are then estimated with

$$\boldsymbol{\mu}_y^s = \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}^{s(k)}; \tag{14a}$$

$$\boldsymbol{\Sigma}_y^s = \left(\frac{1}{K}\sum_{k=1}^{K} \mathbf{y}^{s(k)}\left[\mathbf{y}^{s(k)}\right]^T\right) - \boldsymbol{\mu}_y^s \boldsymbol{\mu}_y^{s\,T}. \tag{14b}$$

This allows the static parameters to be compensated. In previous work on DPMC an alternative approach to the continuous time approximation for compensating the dynamic parameters was proposed [6]. This approach is only applicable when simple differences (linear regression using a window of one time instance left and one right) are used. By modelling the static coefficients from the previous time instance to the feature vector, $\mathbf{x}_{t-1}^s$, the dynamic coefficients for the noise-corrupted speech can be found using [2]

$$\mathbf{y}_t^{\Delta(k)} = \mathbf{f}\left(\mathbf{x}_t^{\Delta(k)} + \mathbf{x}_{t-1}^{s(k)}, \mathbf{n}_t^{\Delta(k)} + \mathbf{n}_{t-1}^{s(k)}, \mathbf{h}^s\right) - \mathbf{f}\left(\mathbf{x}_{t-1}^{s(k)}, \mathbf{n}_{t-1}^{s(k)}, \mathbf{h}^s\right). \tag{15}$$

However this form of approximation cannot be used for the linear-regression-base dynamic parameters.

In the limit as the number of samples goes to infinity, DPMC yields accurate Gaussian parameters for the noise-corrupted speech given the mismatch function and the speech and noise distributions, and could be viewed as an infinite-order VTS. However, as a large number of samples are necessary to train the noise-corrupted speech distributions, the computational cost is much greater than for VTS.

[2] Normalisation of dynamic parameters is ignored for clarity of presentation.

2.3 Noise estimation

The discussion so far has assumed that a distribution of the noise is known. In practice, however, this is seldom the case. The noise model must therefore be estimated. The model $\mathcal{M}_n = \{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, \boldsymbol{\mu}_h\}$ comprises the parameters of the additive noise, assumed Gaussian with $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, and the convolutional noise $\boldsymbol{\mu}_h$, which is assumed constant. The parameters are of the form

$$\boldsymbol{\mu}_n = \begin{bmatrix} \boldsymbol{\mu}_n^s \\ 0 \end{bmatrix}; \qquad \boldsymbol{\Sigma}_n = \begin{bmatrix} \operatorname{diag}(\boldsymbol{\Sigma}_n^s) & 0 \\ 0 & \operatorname{diag}(\boldsymbol{\Sigma}_n^\Delta) \end{bmatrix}; \qquad \boldsymbol{\mu}_h = \begin{bmatrix} \boldsymbol{\mu}_h^s \\ 0 \end{bmatrix}. \tag{16}$$

The expected value of the dynamic coefficients of the additive noise are zero because the noise model has no state changes. Since the convolutional noise is assumed constant, its dynamic parameters are also zero. Using data from the target environment, it is possible to find a noise estimate with expectation–maximisation that maximises the likelihood assuming a particular form of model-based compensation, for example VTS compensation [17, 14]. This iteratively updates the component-time posteriors (the expectation step) and the noise model (the maximisation step).

In the maximisation step, the static noise means can be updated at the same time using a fixed-point iteration [17]. The additive noise covariance, however, is more complex to estimate. It is possible to estimate it on the parts of the waveform known to contain noise without speech. Another options is to use gradient ascent to find an estimate for the additive noise variance for VTS with the continuous time approximation [14]. This needs to be alternated with the estimation of the noise mean.

Another option is to use Expectation-Maximisation (EM) to estimate the noise model parameters. Here the noise and corrupted speech are assumed to be jointly Gaussian. The noise distribution can be iteratively estimated using EM [11]. In [11] this approach was used with feature-based VTS. When model-based compensation schemes are used along with the diagonal corrupted speech distribution, the form in [11] cannot be used, unless additional approximations, such as diagonalising the cross-covariance of the noise and the corrupted speech are made. Thus in this work the approach described in [14] is used to estimate the noise model parameters.

The resulting noise model estimate maximises the likelihood of model compensation with VTS. Thus, the parameters do not necessarily correspond to the actual noise or to a consistent sequence of static observations.

3 Extended VTS

The continuous time approximation yields a simple approach to compensating the dynamic feature model parameters. However, it relies on an approximation which has been found to degrade performance for some noise conditions [12]. This section describes an alternative method for compensating the dynamic model parameters called "extended VTS". The key intuition is that the dynamic coefficients are a linear combination of consecutive static feature vectors. Thus, if the corrupted speech distribution over the consecutive static feature vectors can be estimated then the distribution for the dynamic coefficients can be found.

3.1 Model compensation with extended feature vectors

For simplicity, a window of ± 1 and only first-order dynamic coefficients will be considered. An *extended* feature vector $\mathbf{y}_t^e$, containing the static feature vectors in the surrounding window, is given by $\mathbf{y}_t^e = \left[\mathbf{y}_{t-1}^{s\,\mathsf{T}} \; \mathbf{y}_t^{s\,\mathsf{T}} \; \mathbf{y}_{t+1}^{s\,\mathsf{T}} \right]^\mathsf{T}$.[3] The transformation of the extended feature vector $\mathbf{y}_t^e$ to the standard feature vector with static and dynamic parameters $\mathbf{y}_t$ can be expressed as a linear projection $\mathbf{D}$:

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{y}_t^s \\ \mathbf{y}_t^\Delta \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\frac{\mathbf{I}}{2} & \mathbf{0} & \frac{\mathbf{I}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{t-1}^s \\ \mathbf{y}_t^s \\ \mathbf{y}_{t+1}^s \end{bmatrix} = \mathbf{D}\mathbf{y}_t^e. \tag{17}$$

The second row of $\mathbf{D}$ applies the transformation from a window of statics to yield the standard delta features. As $\mathbf{D}$ is a linear transformation, if the distribution of the extended corrupted speech $\mathbf{y}_t^e$ is assumed Gaussian with with mean $\boldsymbol{\mu}_y^e$ and covariance $\boldsymbol{\Sigma}_y^e$, the extended corrupted speech distribution can be transformed to a distribution over statics and dynamics with

$$\boldsymbol{\mu}_y = \mathcal{E}\{\mathbf{y}\} = \mathbf{D}\mathcal{E}\{\mathbf{y}^e\} = \mathbf{D}\boldsymbol{\mu}_y^e; \tag{18a}$$

$$\boldsymbol{\Sigma}_y = \mathcal{E}\left\{\mathbf{y}\mathbf{y}^\mathsf{T} - \boldsymbol{\mu}_y\boldsymbol{\mu}_y^\mathsf{T}\right\} = \mathbf{D}\mathcal{E}\left\{\mathbf{y}^e\mathbf{y}^{e\mathsf{T}} - \boldsymbol{\mu}_y^e\boldsymbol{\mu}_y^{e\mathsf{T}}\right\}\mathbf{D}^\mathsf{T} = \mathbf{D}\boldsymbol{\Sigma}_y^e\mathbf{D}^\mathsf{T}. \tag{18b}$$

It is interesting to look at the structure distribution of the extended feature vector, $\mathbf{y}^e$. The mean $\boldsymbol{\mu}_y^e$ of the concatenation of consecutive static feature vectors is simply a concatenation of static means at time offsets $-1, 0, +1$. For the corrupted speech, these will be written $\boldsymbol{\mu}_{y_{-1}}^s, \boldsymbol{\mu}_{y_0}^s, \boldsymbol{\mu}_{y_{+1}}^s$. The covariance $\boldsymbol{\Sigma}_y^e$ contains the covariance between statics at different time offsets. The covariance between offsets -1 and $+1$, for example, is written $\boldsymbol{\Sigma}_{y_{-1}y_{+1}}$. Thus, the full parameters of the extended distribution are

$$\boldsymbol{\mu}_y^e = \begin{bmatrix} \boldsymbol{\mu}_{y_{-1}}^s \\ \boldsymbol{\mu}_{y_0}^s \\ \boldsymbol{\mu}_{y_{+1}}^s \end{bmatrix}; \qquad \boldsymbol{\Sigma}_y^e = \begin{bmatrix} \boldsymbol{\Sigma}_{y_{-1}y_{-1}}^s & \boldsymbol{\Sigma}_{y_{-1}y_0}^s & \boldsymbol{\Sigma}_{y_{-1}y_{+1}}^s \\ \boldsymbol{\Sigma}_{y_0y_{-1}}^s & \boldsymbol{\Sigma}_{y_0y_0}^s & \boldsymbol{\Sigma}_{y_0y_{+1}}^s \\ \boldsymbol{\Sigma}_{y_{+1}y_{-1}}^s & \boldsymbol{\Sigma}_{y_{+1}y_0}^s & \boldsymbol{\Sigma}_{y_{+1}y_{+1}}^s \end{bmatrix}. \tag{19}$$

The standard parameters with statics and dynamics can be found from this extended distribution, For example $\boldsymbol{\Sigma}_y$ in (18b), substituting $\boldsymbol{\Sigma}_y^e$ from (19) and $\mathbf{D}$ from (17), can be expressed as

$$\boldsymbol{\Sigma}_y = \mathbf{D}\boldsymbol{\Sigma}_y^e\mathbf{D}^\mathsf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\frac{\mathbf{I}}{2} & \mathbf{0} & \frac{\mathbf{I}}{2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{y_{-1}y_{-1}}^s & \boldsymbol{\Sigma}_{y_{-1}y_0}^s & \boldsymbol{\Sigma}_{y_{-1}y_{+1}}^s \\ \boldsymbol{\Sigma}_{y_0y_{-1}}^s & \boldsymbol{\Sigma}_{y_0y_0}^s & \boldsymbol{\Sigma}_{y_0y_{+1}}^s \\ \boldsymbol{\Sigma}_{y_{+1}y_{-1}}^s & \boldsymbol{\Sigma}_{y_{+1}y_0}^s & \boldsymbol{\Sigma}_{y_{+1}y_{+1}}^s \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\frac{\mathbf{I}}{2} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{I}}{2} \end{bmatrix}. \tag{20}$$

The problem is to find the correct form for the extended distribution.

3.2 Extended distribution compensation

The procedure to find the distribution over the statics and dynamics of the corrupted speech outlined in the previous section requires parameters for the extended corrupted

[3] It is straightforward to extend this to handle both second-order dynamics and linear-regression coefficients over a larger window of $\pm w$, so that $\mathbf{y}_t^e = \left[\mathbf{y}_{t-w}^{s\,\mathsf{T}} \; \ldots \; \mathbf{y}_{t+w}^{s\,\mathsf{T}} \right]^\mathsf{T}$.

speech distribution in (19). Since the extended feature vector is a concatenation of static feature vectors, it is possible to use the static mismatch function for each time offset to yield an overall mismatch function for $\mathbf{y}^e$.

An extension to static compensation using VTS can be used to find the extended corrupted speech distribution. The first-order vector Taylor series approximation in (5) is applied to each time instance separately. Thus the expansion point for each time instance is given by the static means at the appropriate time offsets. These are obtained from the extended distributions distributions over the clean speech, $\mathbf{x}^e$, and noise, $\mathbf{n}^e$. Thus using the form of the VTS approximation in (5) per time instance:

$$\begin{bmatrix} \mathbf{y}^s_{t-1,\text{evts}} \\ \mathbf{y}^s_{t,\text{evts}} \\ \mathbf{y}^s_{t+1,\text{evts}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\boldsymbol{\mu}^s_{x_{-1}}, \boldsymbol{\mu}^s_{n_{-1}}, \boldsymbol{\mu}^s_{h_{-1}}\right) + \mathbf{J}_{-1}\left(\mathbf{x}^s_{t-1} - \boldsymbol{\mu}^s_{x_{-1}}\right) + (\mathbf{I} - \mathbf{J}_{-1})\left(\mathbf{n}^s_{t-1} - \boldsymbol{\mu}^s_{n_{-1}}\right) \\ \mathbf{f}\left(\boldsymbol{\mu}^s_{x_0}, \boldsymbol{\mu}^s_{n_0}, \boldsymbol{\mu}^s_{h_0}\right) + \mathbf{J}_0\left(\mathbf{x}^s_t - \boldsymbol{\mu}^s_{x_0}\right) + (\mathbf{I} - \mathbf{J}_0)\left(\mathbf{n}^s_t - \boldsymbol{\mu}^s_{n_0}\right) \\ \mathbf{f}\left(\boldsymbol{\mu}^s_{x_{+1}}, \boldsymbol{\mu}^s_{n_{+1}}, \boldsymbol{\mu}^s_{h_{+1}}\right) + \mathbf{J}_{+1}\left(\mathbf{x}^s_{t+1} - \boldsymbol{\mu}^s_{x_{+1}}\right) + (\mathbf{I} - \mathbf{J}_{+1})\left(\mathbf{n}^s_{t+1} - \boldsymbol{\mu}^s_{n_{+1}}\right) \end{bmatrix}, \tag{21}$$

where the offset-dependent Jacobians are given by

$$\mathbf{J}_{-1} = \left.\frac{\partial \mathbf{y}^s}{\partial \mathbf{x}^s}\right|_{\boldsymbol{\mu}^s_{n_{-1}}, \boldsymbol{\mu}^s_{x_{-1}}, \boldsymbol{\mu}^s_{h_{-1}}}; \quad \mathbf{J}_0 = \left.\frac{\partial \mathbf{y}^s}{\partial \mathbf{x}^s}\right|_{\boldsymbol{\mu}^s_{n_0}, \boldsymbol{\mu}^s_{x_0}, \boldsymbol{\mu}^s_{h_0}}; \quad \mathbf{J}_{+1} = \left.\frac{\partial \mathbf{y}^s}{\partial \mathbf{x}^s}\right|_{\boldsymbol{\mu}^s_{n_{+1}}, \boldsymbol{\mu}^s_{x_{+1}}, \boldsymbol{\mu}^s_{h_{+1}}}. \tag{22}$$

Note that $\mathbf{J}_0$ is equal to the Jacobian for standard VTS described in (6).

The expression for the corrupted speech in (21) requires distributions over extended feature vectors for the clean speech $\mathbf{x}^e$ (from training data) and noise $\mathbf{n}^e, \mathbf{h}^e$ (estimated). The forms of these distributions are analogous to those for the extended corrupted speech in (19):

$$\mathbf{x}^e = \begin{bmatrix} \mathbf{x}^s_{t-1} \\ \mathbf{x}^s_t \\ \mathbf{x}^s_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^s_{x_{-1}} \\ \boldsymbol{\mu}^s_{x_0} \\ \boldsymbol{\mu}^s_{x_{+1}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^s_{x_{-1}x_{-1}} & \boldsymbol{\Sigma}^s_{x_{-1}x_0} & \boldsymbol{\Sigma}^s_{x_{-1}x_{+1}} \\ \boldsymbol{\Sigma}^s_{x_0x_{-1}} & \boldsymbol{\Sigma}^s_{x_0x_0} & \boldsymbol{\Sigma}^s_{x_0x_{+1}} \\ \boldsymbol{\Sigma}^s_{x_{+1}x_{-1}} & \boldsymbol{\Sigma}^s_{x_{+1}x_0} & \boldsymbol{\Sigma}^s_{x_{+1}x_{+1}} \end{bmatrix}\right); \tag{23a}$$

$$\mathbf{n}^e = \begin{bmatrix} \mathbf{n}^s_{t-1} \\ \mathbf{n}^s_t \\ \mathbf{n}^s_{t+1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^s_{n_{-1}} \\ \boldsymbol{\mu}^s_{n_0} \\ \boldsymbol{\mu}^s_{n_{+1}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^s_{n_{-1}n_{-1}} & \boldsymbol{\Sigma}^s_{n_{-1}n_0} & \boldsymbol{\Sigma}^s_{n_{-1}n_{+1}} \\ \boldsymbol{\Sigma}^s_{n_0n_{-1}} & \boldsymbol{\Sigma}^s_{n_0n_0} & \boldsymbol{\Sigma}^s_{n_0n_{+1}} \\ \boldsymbol{\Sigma}^s_{n_{+1}n_{-1}} & \boldsymbol{\Sigma}^s_{n_{+1}n_0} & \boldsymbol{\Sigma}^s_{n_{+1}n_{+1}} \end{bmatrix}\right); \tag{23b}$$

$$\mathbf{h}^e = \begin{bmatrix} \mathbf{h}^s_{t-1} \\ \mathbf{h}^s_t \\ \mathbf{h}^s_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}^s_{h_{-1}} \\ \boldsymbol{\mu}^s_{h_0} \\ \boldsymbol{\mu}^s_{h_{+1}} \end{bmatrix}. \tag{23c}$$

Approaches and approximations for these clean speech and noise statistics will be discussed in section 4.1.

The parameters of the extended corrupted speech distribution in (19) can be found by computing expectations over the distributions in (23). The mean for time offset $+1$, for example, is given by

$$\boldsymbol{\mu}^s_{y_{+1}} = \mathcal{E}\left\{\mathbf{y}^s_{t+1,\text{evts}}\right\} = \mathbf{f}\left(\boldsymbol{\mu}^s_{x_{+1}}, \boldsymbol{\mu}^s_{n_{+1}}, \boldsymbol{\mu}^s_{h_{+1}}\right). \tag{24}$$

The covariance matrix $\boldsymbol{\Sigma}^e_y$ (shown in (19)) requires the correlations between all time offsets in the window to be computed. The covariance between offsets 0 and $+1$, for

example, is found by generalising (7b) to

$$\begin{aligned}
\boldsymbol{\Sigma}^{s}_{y_0 y_{+1}} &= \mathcal{E}\left\{(\mathbf{y}^{s}_{t,\text{evts}} - \boldsymbol{\mu}_{y_0})(\mathbf{y}^{s}_{t+1,\text{evts}} - \boldsymbol{\mu}_{y_{+1}})^{\mathsf{T}}\right\} \\
&= \mathcal{E}\Big\{ \left(\mathbf{J}_0(\mathbf{x}^{s}_{t} - \boldsymbol{\mu}^{s}_{x_0}) + (\mathbf{I} - \mathbf{J}_0)(\mathbf{n}^{s}_{t} - \boldsymbol{\mu}^{s}_{n_0})\right) \\
&\qquad \left(\mathbf{J}_{+1}(\mathbf{x}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{x_{+1}}) + (\mathbf{I} - \mathbf{J}_{+1})(\mathbf{n}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{n_{+1}})\right)^{\mathsf{T}} \Big\} \\
&= \mathcal{E}\Big\{ \mathbf{J}_0(\mathbf{x}^{s}_{t} - \boldsymbol{\mu}^{s}_{x_0})(\mathbf{x}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{x_{+1}})^{\mathsf{T}}\mathbf{J}^{\mathsf{T}}_{+1} \\
&\qquad + (\mathbf{I} - \mathbf{J}_0)(\mathbf{n}^{s}_{t} - \boldsymbol{\mu}^{s}_{n_0})(\mathbf{n}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{n_{+1}})^{\mathsf{T}}(\mathbf{I} - \mathbf{J}_{+1})^{\mathsf{T}} \Big\} \\
&= \mathbf{J}_0 \boldsymbol{\Sigma}^{s}_{x_0 x_{+1}} \mathbf{J}^{\mathsf{T}}_{+1} + (\mathbf{I} - \mathbf{J}_0)\boldsymbol{\Sigma}^{s}_{n_0 n_{+1}}(\mathbf{I} - \mathbf{J}_{+1})^{\mathsf{T}}. \qquad (25)
\end{aligned}$$

This is applied for each of the possible time offset blocks in $\boldsymbol{\Sigma}^{e}_{y}$.

3.3 Relationship between VTS and EVTS

It is interesting to examine the relationship between standard VTS and extended VTS described in the previous section. It is possible to describe the mismatch function for the dynamic coefficients of standard VTS, which uses the continuous time approximation for the dynamic coefficients, in terms of extended VTS.

The approximation that standard VTS uses for the static coefficients is exactly the same as the one extended VTS uses for the statics at the centre time offset. Therefore, the compensated static mean and covariance that standard VTS finds are the same as the ones extended VTS finds for time offset 0. However, dynamic parameter compensation with standard VTS uses the continuous time approximation. This uses the vector Taylor series expansion point of the static coefficients for all the dynamic coefficients. When the vector Taylor series expansion uses the same clean speech and noise means $\boldsymbol{\mu}^{s}_{x_0}, \boldsymbol{\mu}^{s}_{n_0}$ and the same Jacobian $\mathbf{J}_0$ for all time offsets in (21), it becomes

$$\begin{bmatrix} \mathbf{y}^{s}_{t-1,\text{vts}} \\ \mathbf{y}^{s}_{t,\text{vts}} \\ \mathbf{y}^{s}_{t+1,\text{vts}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\boldsymbol{\mu}^{s}_{x_0}, \boldsymbol{\mu}^{s}_{n_0}, \boldsymbol{\mu}^{s}_{h_0}\right) + \mathbf{J}_0\left(\mathbf{x}^{s}_{t-1} - \boldsymbol{\mu}^{s}_{x_0}\right) + (\mathbf{I} - \mathbf{J}_0)\left(\mathbf{n}^{s}_{t-1} - \boldsymbol{\mu}^{s}_{n_0}\right) \\ \mathbf{f}\left(\boldsymbol{\mu}^{s}_{x_0}, \boldsymbol{\mu}^{s}_{n_0}, \boldsymbol{\mu}^{s}_{h_0}\right) + \mathbf{J}_0\left(\mathbf{x}^{s}_{t} - \boldsymbol{\mu}^{s}_{x_0}\right) + (\mathbf{I} - \mathbf{J}_0)\left(\mathbf{n}^{s}_{t} - \boldsymbol{\mu}^{s}_{n_0}\right) \\ \mathbf{f}\left(\boldsymbol{\mu}^{s}_{x_0}, \boldsymbol{\mu}^{s}_{n_0}, \boldsymbol{\mu}^{s}_{h_0}\right) + \mathbf{J}_0\left(\mathbf{x}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{x_0}\right) + (\mathbf{I} - \mathbf{J}_0)\left(\mathbf{n}^{s}_{t+1} - \boldsymbol{\mu}^{s}_{n_0}\right) \end{bmatrix}. \qquad (26)$$

This approximation results in the following when substituted in the expression for computing dynamic coefficients in (3):

$$\begin{aligned}
\mathbf{y}^{\Delta}_{t} &= \frac{\sum_{\tau=1}^{w} \tau(\mathbf{y}^{s}_{t+\tau} - \mathbf{y}^{s}_{t-\tau})}{2\sum_{\tau=1}^{w}\tau^2} \\
&= \frac{\sum_{\tau=1}^{w} \tau\left(\mathbf{J}_0\mathbf{x}^{s}_{t+\tau} + (\mathbf{I} - \mathbf{J}_0)\mathbf{n}^{s}_{t+\tau} - \mathbf{J}_0\mathbf{x}^{s}_{t-\tau} - (\mathbf{I} - \mathbf{J}_0)\mathbf{n}^{s}_{t-\tau}\right)}{2\sum_{\tau=1}^{w}\tau^2} \\
&= \frac{\mathbf{J}_0\sum_{\tau=1}^{w}\tau\left(\mathbf{x}^{s}_{t+\tau} - \mathbf{x}^{s}_{t-\tau}\right) + (\mathbf{I} - \mathbf{J}_0)\sum_{\tau=1}^{w}\tau\left(\mathbf{n}^{s}_{t+\tau} - \mathbf{n}^{s}_{t-\tau}\right)}{2\sum_{\tau=1}^{w}\tau^2} \\
&= \mathbf{J}_0\mathbf{x}^{\Delta}_{t} + (\mathbf{I} - \mathbf{J}_0)\mathbf{n}^{\Delta}_{t}. \qquad (27)
\end{aligned}$$

This is exactly the same expression as the continuous time approximation when applied to VTS compensation (in (8)). Extended VTS compensation therefore becomes equivalent to standard VTS compensation when the expansion point is chosen equal for all

time offsets. Extended VTS performs the transformation from extended to standard parameters after compensation. This allows extended VTS to use a different vector Taylor series expansion point for every time offset to find more accurate compensation.

A scheme related to EVTS was described in [3]. This approach proposed a similar form of linear transformation of a window of static parameters to obtain the compensated dynamic parameters. However, there are a number of differences between the schemes. First, the scheme in [3] operated in the log-spectral domain, rather than the cepstral domain used for EVTS. Moreover, a large number of additional approximations are used in [3] to derive the VTS form. This includes ignoring correlations between time instances and using the same Jacobian between time instances. This approximation negates the advantages in accuracy of explicitly modelling distributions over a window of features over standard VTS.

3.4 Computational cost

Compensation with extended VTS is more computationally expensive than VTS with the continuous time approximation. This section examines the differences in detail. The computational complexity per component will be expressed in terms of the size of the static feature vector n (typically 13), the total width of the window $e = 4w + 1$ (typically 9), and the number of orders of statics and dynamics d^{Δ} (typically 3). Since the calculation of the covariance matrices dominates the computation time, the analysis will not explicitly consider the means.

	VTS		EVTS	
Statistics	diag.	block-d.	striped	full
Decoding	diag.	block-d.	diag.	full
Jacobians	n^3		en^3	
Compensation	$d^{\Delta}n^2$	$d^{\Delta}n^3$	e^2n^2	e^2n^3
Projection	—		$d^{\Delta}ne^2$	$d^{\Delta 2}n^2e^2$

Table 1 *Computational complexity $\mathcal{O}(\cdot)$ per component for compensation with VTS and with extended VTS, for diagonal blocks and for full blocks.*

Table 1 gives a comparison of the computational complexity for the three operations that can be distinguished in extended VTS compensation. The first one is computing the Jacobian of the mismatch function, which takes $\mathcal{O}\left(n^3\right)$. Standard VTS compensation uses one linearisation point per component, and therefore needs to compute the Jacobian only once. Extended VTS, however, uses a different linearisation point for all e time offset in the window, and computes a Jacobian for each of these.

Compensation of the covariance matrix is done one $n \times n$ block at a time. The expression for standard VTS is (repeated from (7b)):

$$\boldsymbol{\Sigma}_y^s = \mathbf{J}\boldsymbol{\Sigma}_x^s\mathbf{J}^T + (\mathbf{I} - \mathbf{J})\boldsymbol{\Sigma}_n^s(\mathbf{I} - \mathbf{J})^T. \tag{28}$$

The expression for extended VTS compensation has the same form (but different variables) for each block of the covariance matrix. It has time complexity $\mathcal{O}\left(n^2\right)$ if the blocks for the noise $\boldsymbol{\Sigma}_n^s$, the clean speech $\boldsymbol{\Sigma}_x^s$ and the corrupted speech $\boldsymbol{\Sigma}_y^s$ are diagonal. For standard VTS, this happens when the covariances for statistics and decoding

are all diagonal; for extended VTS, when covariances for statistics are striped, and for decoding are diagonal. When either the statistics or compensation uses full covariance matrices, then compensation takes $\mathcal{O}\left(n^3\right)$. For VTS, the d^Δ blocks along the diagonal are compensated; for extended VTS, for the $\frac{1}{2}e(e+1)$ blocks in the extended covariance matrix. The row labelled "Compensation" in table 1 summarises this.

Extended VTS projects the compensated distribution onto the standard feature space with statics and dynamics. Since the projection matrix $\mathbf{D}$ is striped, computing one entry of the resulting covariance matrix $\mathbf{\Sigma}_y^e = \mathbf{D}\mathbf{\Sigma}_y^e\mathbf{D}^T$ takes $\mathcal{O}\left(e^2\right)$. For diagonal-covariance decoding, $d^\Delta n$ entries need to be computed; for full-covariance decoding, ${d^\Delta}^2 n^2$.

Thus for full-covariance compensation, the computational complexity of EVTS is significantly higher than standard VTS. However, in practice per-Gaussian compensation is often too costly even when the standard version of VTS is used. Joint uncertainty decoding (JUD) [16] attempts to address this by computing compensation per base class rather than per Gaussian component. The choice for a number of base classes decides the trade-off between speed and accuracy. EVTS, and the other approximations discussed in the next section such as extended DPMC, can also be used within the JUD compensation framework. To apply these approaches to JUD statistics (as in section 4.1) and compensation are computed per base-class. Appendix A discusses this process in more detail.

Another important issue is the computational cost of decoding. If full-covariance compensation is found, joint uncertainty decoding still compensates for changes in the correlations by decoding with full covariance matrices. This is slow. Predictive linear transformations [5] can solve this issue by applying transformation to the feature vectors that eliminate the need to decode with full covariance matrices. Other predictive transforms, such as predictive semi-tied covariance matrices trained from extended VTS compensation, would also be possible. However, this paper concentrates on finding accurate compensation per component.

3.5 Alternative extended approximations

The previous section has discussed the use of VTS to determine the extended distribution. Rather than approximating the mismatch function to find the corrupted speech parameters analytically, it is also possible to use sampling approaches. This section discusses two sampling methods: extended DPMC (EDPMC), and the unscented transformation. Both draw sample pairs, $(\mathbf{x}^{e(k)}, \mathbf{n}^{e(k)})$, from the extended distributions of the clean speech and additive noise. Each sample has a weight $w^{(k)}$ associated with it. The total weight summed over all samples, w, is given by $w = \sum_k w^{(k)}$. The mismatch function $\mathbf{f}$ for the static cepstral parameters in (2) is applied to each time offset to yield an extended corrupted speech sample for $\mathbf{y}^{e(k)}$:

$$\mathbf{y}_t^{e(k)} = \begin{bmatrix} \mathbf{y}_{t-1}^{s(k)} \\ \mathbf{y}_t^{s(k)} \\ \mathbf{y}_{t+1}^{s(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{x}_{t-1}^{s(k)}, \mathbf{n}_{t-1}^{s(k)}, \mathbf{h}^s\right) \\ \mathbf{f}\left(\mathbf{x}_t^{s(k)}, \mathbf{n}_t^{s(k)}, \mathbf{h}^s\right) \\ \mathbf{f}\left(\mathbf{x}_{t+1}^{s(k)}, \mathbf{n}_{t+1}^{s(k)}, \mathbf{h}^s\right) \end{bmatrix}. \tag{29}$$

Maximum likelihood estimates of the extended corrupted speech parameters $\boldsymbol{\mu}_y^e$ and $\boldsymbol{\Sigma}_y^e$ can then be found based on these samples:

$$\boldsymbol{\mu}_y^e = \frac{1}{w}\sum_{k=1}^{K} w^{(k)}\mathbf{y}^{e(k)}; \tag{30a}$$

$$\boldsymbol{\Sigma}_y^e = \left(\frac{1}{w}\sum_{k=1}^{K} w^{(k)}\mathbf{y}^{e(k)}\left[\mathbf{y}^{e(k)}\right]^T\right) - \boldsymbol{\mu}_y^e\boldsymbol{\mu}_y^{eT}. \tag{30b}$$

The first method considered for choosing the samples is a Monte Carlo approach, extended DPMC. Extended DPMC, EDPMC, is based on DPMC as described in section 2.2, but it randomly draws samples $\mathbf{x}^{e(k)}$ and $\mathbf{n}^{e(k)}$, with constant weights, from the extended distributions of the clean speech and additive noise. This procedure is slow, but generates accurate compensation as the number of samples goes to infinity. This paper uses EDPMC as a reference compensation method. Additional results using EDPMC, along with its application to JUD and predictive transformations, are given in [20].

The second method of choosing the samples uses the *unscented transformation* [9]. This draws samples, called *sigma points*, deterministically. The samples and their weights are chosen according to certain properties, for example, so that their sample mean and covariance are equal to the distribution's. The standard scheme for choosing samples in [9] draws two symmetric samples around the mean for every dimension. The number of samples therefore scales linearly in the number of dimensions. The unscented transformation has been applied to feature enhancement for noise compensation earlier [19]. See appendix B for a discussion and preliminary results of model compensation with the unscented transformation.

4 Extended statistics

A practical issue when using EVTS is the form of the statistics for the clean speech and the noise. For standard VTS, the clean speech statistics are usually taken from the recogniser trained on clean speech and the noise model is usually estimated with maximum likelihood estimation, as discussed in section 2.3. In contrast, EVTS requires distributions over the extended clean speech and noise vectors. As these have more parameters than standard statistics, robustness and storage requirements need to be carefully considered.

4.1 Clean speech statistics

Model compensation schemes, such as VTS, use the Gaussian components from the uncompensated system as the clean speech distributions. For EVTS, however, distributions over the extended clean speech vector are required. For one extended clean speech Gaussian $\mathcal{N}(\boldsymbol{\mu}_x^e, \boldsymbol{\Sigma}_x^e)$, the parameters are those in (23a)

$$\boldsymbol{\mu}_x^e = \begin{bmatrix} \boldsymbol{\mu}_{x_{-1}}^s \\ \boldsymbol{\mu}_{x_0}^s \\ \boldsymbol{\mu}_{x_{+1}}^s \end{bmatrix}; \qquad \boldsymbol{\Sigma}_x^e = \begin{bmatrix} \boldsymbol{\Sigma}_{x_{-1}x_{-1}}^s & \boldsymbol{\Sigma}_{x_{-1}x_0}^s & \boldsymbol{\Sigma}_{x_{-1}x_{+1}}^s \\ \boldsymbol{\Sigma}_{x_0x_{-1}}^s & \boldsymbol{\Sigma}_{x_0x_0}^s & \boldsymbol{\Sigma}_{x_0x_{+1}}^s \\ \boldsymbol{\Sigma}_{x_{+1}x_{-1}}^s & \boldsymbol{\Sigma}_{x_{+1}x_0}^s & \boldsymbol{\Sigma}_{x_{+1}x_{+1}}^s \end{bmatrix}. \tag{31}$$

In common with standard model compensation schemes, when there is no noise the compensated system should be the original clean system. To ensure that this is the case single-pass retraining [6] should be used to obtain the extended clean speech distributions. Here the same posteriors (associated with the complete data set for EM) of the last standard clean speech training iteration (with static and dynamic parameters) are used to accumulate extended feature vectors around every time instance.

Another problem with using the extended statistics is ensuring robust estimation. The extended feature vectors contain more parameters than the standard static and dynamic ones. Hence, the estimates of their distributions will be less robust and take up more memory. If full covariance matrices for $\boldsymbol{\Sigma}_x^e$ are stored and used, both first- and second-order dynamic parameters use window widths of ± 2, and there are n static parameters, this requires estimating a $9n \times 9n$ covariance matrix for every component. This is memory-intensive and singular matrices and numerical accuracy problems can occur. One solution is to reduce the number of Gaussian components or states in the system. However, the precision of the speech model then decreases. Also, this makes it hard to compare the performance of compensation with extended VTS and standard VTS.

An alternative approach is to modify the structure of the covariance matrices, in the same fashion as diagonalising the standard clean speech covariance model. To maintain some level of inter-frame correlations, which may be useful for computing the dynamic parameters, each block is diagonalised. This yields the following structure

$$\boldsymbol{\Sigma}_x^e = \begin{bmatrix} \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{-1}x_{-1}}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{-1}x_0}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{-1}x_{+1}}\right) \\ \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_0x_{-1}}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_0x_0}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_0x_{+1}}\right) \\ \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{+1}x_{-1}}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{+1}x_0}\right) & \operatorname{diag}\left(\boldsymbol{\Sigma}^s_{x_{+1}x_{+1}}\right) \end{bmatrix}. \tag{32}$$

For each Gaussian component, the ith element of the static coefficients for a time instance is then assumed correlated with only itself and the ith element of other time instances. This causes $\boldsymbol{\Sigma}_x^e$ to have a striped structure with only $45n$ parameters rather than $9n(9n+1)/2$ for the full case. This type of covariance matrix will be called "striped". A useful attribute of this structure is that when there is no noise it will still yield the standard static and dynamic clean speech parameters.

4.2 Noise model estimation

A noise model with extended feature vectors is necessary to perform compensation with extended VTS. This noise model is of the form $\mathcal{M}_n^e = \{\boldsymbol{\mu}_n^e, \boldsymbol{\Sigma}_n^e, \boldsymbol{\mu}_h^e\}$, with parameters shown in (23). In this work, and the majority of other work, the noise model consists of a single Gaussian component. The distribution for each time offset therefore is by definition the same. This means that the extended means for the additive and convolutional noise simply repeat the static means. The structure of the extended covariance $\boldsymbol{\Sigma}_n^e$ is also known. Since the noise is assumed identically distributed for all time instances at the same distance, the correlation between time instances is always the same. Thus, all diagonals of the covariance matrix repeat the same entries. Let $\boldsymbol{\Sigma}^s_{n_0}, \boldsymbol{\Sigma}^s_{n_1}, \boldsymbol{\Sigma}^s_{n_2}$ indicate the cross-correlation between noise that is 0, 1, or 2 time instances apart. The extended noise model then has the following form:

$$\boldsymbol{\mu}_n^e = \begin{bmatrix} \boldsymbol{\mu}_n^s \\ \boldsymbol{\mu}_n^s \\ \boldsymbol{\mu}_n^s \end{bmatrix}; \qquad \boldsymbol{\Sigma}_n^e = \begin{bmatrix} \boldsymbol{\Sigma}^s_{n_0} & \boldsymbol{\Sigma}^{s\mathsf{T}}_{n_1} & \boldsymbol{\Sigma}^{s\mathsf{T}}_{n_2} \\ \boldsymbol{\Sigma}^s_{n_1} & \boldsymbol{\Sigma}^s_{n_0} & \boldsymbol{\Sigma}^{s\mathsf{T}}_{n_1} \\ \boldsymbol{\Sigma}^s_{n_2} & \boldsymbol{\Sigma}^s_{n_1} & \boldsymbol{\Sigma}^s_{n_0} \end{bmatrix}; \qquad \boldsymbol{\mu}_h^e = \begin{bmatrix} \boldsymbol{\mu}_h^s \\ \boldsymbol{\mu}_h^s \\ \boldsymbol{\mu}_h^s \end{bmatrix}. \tag{33}$$

In theory these noise parameters could be found using maximum likelihood estimation. However, this complicates the noise estimation process. It would be preferable to use the standard noise estimation schemes and map the parameters to the ones in the extended forms above. These "standard" noise parameters are

$$\mu_n = \begin{bmatrix} \mu_n^s \\ 0 \end{bmatrix}; \qquad \Sigma_n = \begin{bmatrix} \operatorname{diag}(\Sigma_n^s) & 0 \\ 0 & \operatorname{diag}(\Sigma_n^\Delta) \end{bmatrix}; \qquad \mu_h = \begin{bmatrix} \mu_h^s \\ 0 \end{bmatrix}. \tag{34}$$

The extended noise means are straightforward functions of the static means of the standard noise model (34). Similarly, $\Sigma_{n_0}^s$, the covariance between noise 0 time instances apart, is the static noise covariance Σ_n^s. Computing the off-diagonals of the extended covariance, however, is not as straightforward. The next subsections will discuss two forms of extended noise covariance from the standard noise covariance: the diagonal form, and a smooth reconstruction.

4.2.1 Diagonal extended noise covariance

A simple way of reconstructing the extended noise covariance from a standard noise model assumes that the noise is uncorrelated between time instances. This is done by setting the off-diagonal elements are set to zero, which yields

$$\Sigma_n^e = \begin{bmatrix} \Sigma_n^s & 0 & 0 \\ 0 & \Sigma_n^s & 0 \\ 0 & 0 & \Sigma_n^s \end{bmatrix}. \tag{35}$$

This only uses the static elements of the estimated noise covariance. For very low signal-to-noise ratio (SNR) conditions this form of extended noise distribution will not yield the standard noise distributions for the dynamic parameters.

4.2.2 Smooth reconstruction of the extended noise variance

Another option is to use the dynamic parameters of the noise model to find a reconstruction of the extended noise covariance from (34). A problem is that the mapping from the extended feature domain to statics and dynamics is straightforward, but the reverse mapping is under-specified. The standard feature vector $\mathbf{n}_t$ is related to one in the extended domain $\mathbf{n}_t^e$ (analogously to (17)):

$$\mathbf{n}_t = \begin{bmatrix} \mathbf{n}_t^s \\ \mathbf{n}_t^\Delta \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} & 0 \\ -\frac{\mathbf{I}}{2} & 0 & \frac{\mathbf{I}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{n}_{t-1}^s \\ \mathbf{n}_t^s \\ \mathbf{n}_{t+1}^s \end{bmatrix} = \mathbf{D}\mathbf{n}_t^e. \tag{36}$$

To reconstruct $\mathbf{n}_t^e$ from $\mathbf{n}_t$, extra constraints are necessary as $\mathbf{D}$ is not square, and therefore not invertible. These constraints should yield an extended feature vector that represents a plausible sequence of static feature vectors. The Moore-Penrose pseudo-inverse of $\mathbf{D}$ could be used. However, this would result in the $\mathbf{n}_t^e$ with the smallest norm. For the three-dimensional example used here, the reconstruction would set $\mathbf{n}_{t-1}^e = -\mathbf{n}_{t+1}^e$ without any reference to the value of the static coefficients $\mathbf{n}_t^s$. Thus, the Moore-Penrose pseudoinverse might lead to reconstructions with large changes in coefficients from one time to the next.

The need for smooth changes from time instance to time instance can be used as additional constraints. Thus the aim is to find a smooth reconstruction whilst satisfying the constraints to yield the standard static and dynamic distributions. To implement this constraint rows representing higher-frequency changes are added to $\mathbf{D}$ and zeros added to $\mathbf{n}_t$ to indicate their desired values. The extension of the projection matrix $\mathbf{D}$, $\mathbf{E}$, can then be made invertible. Thus

$$\begin{bmatrix} \mathbf{n}_t^s \\ \mathbf{n}_t^\Delta \\ 0 \end{bmatrix} = \mathbf{E}\mathbf{n}_t^e; \qquad \mathbf{E}^{-1} \begin{bmatrix} \mathbf{n}_t^s \\ \mathbf{n}_t^\Delta \\ 0 \end{bmatrix} = \mathbf{n}_t^e. \tag{37}$$

For the extra rows of $\mathbf{E}$, the corresponding rows from the discrete cosine transform (DCT) matrix are appropriate, since they indicate higher-order frequencies and are independent. The entries of a $N \times N$ DCT matrix $\mathbf{C}$ are given by

$$c_{ij} = \sqrt{\frac{2}{N}} \cos \frac{(2j-1)(i-1)\pi}{2N}. \tag{38}$$

The form of $\mathbf{E}$, $\mathbf{D}$ with DCT-derived blocks appended, is:

$$\mathbf{E} = \begin{bmatrix} 0 & \mathbf{I} & 0 \\ -\frac{\mathbf{I}}{2} & 0 & \frac{\mathbf{I}}{2} \\ c_{31}\mathbf{I} & c_{32}\mathbf{I} & c_{33}\mathbf{I} \end{bmatrix}. \tag{39}$$

Because the dynamic mean of the additive noise is zero, $\mathbf{E}^{-1}\boldsymbol{\mu}_n$ is equal to the extended mean in (33) (and similar for the convolutional noise). To reconstruct the extended covariance $\boldsymbol{\Sigma}_n^e$ from a standard noise model, the cross-covariance between statics and dynamics can be ignored, and the higher-order covariance terms set to zero. This results in the following expression:

$$\begin{bmatrix} \boldsymbol{\Sigma}_n^s & & \\ & \boldsymbol{\Sigma}_n^\Delta & \\ & & 0 \end{bmatrix} = \mathbf{E}\boldsymbol{\Sigma}_n^e\mathbf{E}^\mathsf{T} = \mathbf{E} \begin{bmatrix} \boldsymbol{\Sigma}_{n_0}^s & \boldsymbol{\Sigma}_{n_1}^{s\mathsf{T}} & \boldsymbol{\Sigma}_{n_2}^{s\mathsf{T}} \\ \boldsymbol{\Sigma}_{n_1}^s & \boldsymbol{\Sigma}_{n_0}^s & \boldsymbol{\Sigma}_{n_1}^{s\mathsf{T}} \\ \boldsymbol{\Sigma}_{n_2}^s & \boldsymbol{\Sigma}_{n_1}^s & \boldsymbol{\Sigma}_{n_0}^s \end{bmatrix} \mathbf{E}^\mathsf{T}, \tag{40}$$

which is a system with 3 sets of matrix equalities, which can be simply solved.[4] In this work, the estimated noise covariance matrix $\boldsymbol{\Sigma}_n$ is diagonal, so that $\boldsymbol{\Sigma}_{n_0}^s, \boldsymbol{\Sigma}_{n_1}^s, \boldsymbol{\Sigma}_{n_2}^s$ are also diagonal. This results in a striped matrix for $\boldsymbol{\Sigma}_n^e$.

4.2.3 Zeros in the noise variance estimate

An additional issue that can occur when estimating the noise model using maximum likelihood, is that noise variances estimates for some dimensions can become very small, or zero. Though this value may optimise the likelihood, it does not necessarily reflect the "true" noise variance. This can lead to the following problem in compensation.

One problem for the small noise variance estimates is that the clean speech "silence" models are never really estimated on silence. In practice even for clean speech there are always low levels of background noise. Thus the estimated noise is really only relative to this clean background level. At very high SNRs the noise may be at a similar level to

[4] This implicitly sets $\boldsymbol{\Sigma}_{n_0}^s$ to $\boldsymbol{\Sigma}_n^s$.

the clean "silence". This will cause very small noise variance values. Another problem results from the form of the covariance matrix compensation. For the static parameters this may be written as (repeated from (7b))

$$\boldsymbol{\Sigma}_y^s = \mathbf{J}\boldsymbol{\Sigma}_x^s\mathbf{J}^T + (\mathbf{I}-\mathbf{J})\boldsymbol{\Sigma}_n^s(\mathbf{I}-\mathbf{J})^T. \tag{41}$$

At low SNRs $\mathbf{J} \rightarrow \mathbf{0}$, so the corrupted distribution tends to the noise distribution. Conversely, at high SNRs $\mathbf{J} \rightarrow \mathbf{I}$ as the corrupted speech distribution tends to the clean speech distribution. The impact of this when estimating the noise covariance matrix $\boldsymbol{\Sigma}_n^s$ in high SNR conditions is that changes in the form of the noise covariance matrix have little impact on the final compensated distribution.

When VTS with the continuous time approximation, along with diagonal corrupted speech covariance matrices, is used during both noise estimation and recognition then the process is self-consistent. However if the noise estimates are used with EVTS to find full compensated covariance matrices this is not the case. This slight mismatch can cause problems. To address this issue a back-off strategy can be used. When the estimated noise variance has very low values rather than using full compensated covariance matrices, diagonal compensated variances can be used. This will occur at high SNRs, where the correlation changes compared to the clean speech conditions should be minimal. In this condition little gain is expected from full compensated covariance matrices.

An alternative approach to address this problem is to make the noise estimation and decoding consistent for EVTS. This is not investigated in this work. By using the same noise estimates for both VTS and EVTS, only differences in the compensation process are examined, rather than any differences in the noise estimation process. It should be emphasised that the results presented for EVTS may a slight underestimate of the possible performance if a fully integrated noise estimate was used. Integrated noise estimation with EVTS will be investigated in future work using for example the approach described in [11]. Here the joint distribution of the corrupted speech and extended noise is modelled using a Gaussian, and EM used to find the extended noise distribution.

5 Preliminary experiments

Preliminary experiments were run to assess the potential of various techniques discussed in precious section. For these experiments a noise corrupted version of the Resource Management task was used. The task and form of noise added will be discussed in greater detail in section 6.2. Noise was artificially added to the clean speech data. Hence it is possible to obtain the "correct" noise distribution, for both the standard and extended feature vector cases. This *known* noise situation allows both the accuracy of the compensation scheme to be compared to the ideal, single pass retrained, system [6], as well as the impact of varying the structure and approximations of the noise covariance matrix to be investigated.

5.1 Relationship to ideal compensation

It is usual to evaluation the performance of compensation methods by comparing word error rates. However, this does not allow a detailed assessment of which aspects of the compensation process are working well and which poorly. An alternative approach to

assessing the quality of model-based compensation schemes is to examine how close the compensated system is to the ideal single-pass retrained system. The Gaussian components in a single-pass-retrained system are estimated using noise-corrupted speech data and the component posteriors (associated with the complete data set) from the clean training data. This may be viewed as an ideal compensation scheme as the corrupted speech distributions are directly based on corrupting the clean speech data using noise and the static mismatch function. It is then possible to examine how closely the compensation Gaussians are to those in the single-pass retrained system. A useful comparison metric for this is the occupancy-weighted average of the component-for-component Kullback-Leibler (KL) divergence of the compensated system to the single-pass retrained system [6]. If $p^{(m)}$ is the mth Gaussian of the single-pass retrained system, and $q^{(m)}$ is the corresponding Gaussian of the compensated system, then this metric $\mathcal{D}$ is

$$\mathcal{D} = \sum_m \gamma^{(m)} \mathcal{KL}\left(p^{(m)} \| q^{(m)}\right) \Big/ \sum_m \gamma^{(m)}, \tag{42}$$

where $\gamma^{(m)}$ is the occupancy of component m in the last training iteration, for both the compensated and the single-pass retrained system.

A useful attribute of this form of metric is that, depending on the structure of the covariance matrices, it is possible to assess the compensation per coefficient or block of coefficients. When diagonal covariance matrices are used, each dimension may be considered separately. This allows the accuracy of the compensation scheme to be assessed for each dimension. Similarly, block-diagonal compensation can be examined per block of coefficients.

For these experiments the noise was scaled to yield an SNR of 20dB. As the noise is known it is possible to build extended noise models of any covariance structure. For these experiments full covariance matrices were used for the extended statistics in EVTS. For VTS the standard diagonal noise models were used.[5] To ensure robust extended clean speech statistics only single component distributions were used for the clean speech models (compared to the more standard six components in section 4.1).

5.1.1 Diagonal compensation

Normally compensated diagonal-covariance matrices are used in VTS. Thus it is interesting to initially examine this configuration. Using diagonal covariance matrices also allows each dimension to be assessed. Figure 1 contrasts the accuracy of an uncompensated system, and three forms of compensation: standard VTS, extended VTS, and, as an indicator of maximum possible performance, extended DPMC. The horizontal axis has the feature dimensions: 13 static MFCCs $\mathbf{y}^{s}$, 13 first-order dynamics $\mathbf{y}^{\Delta}$, and 13 second-order dynamics $\mathbf{y}^{\Delta^2}$. As expected, the uncompensated system is furthest away from the single-pass retrained system, and extended DPMC provides the most accurate compensation given the speech and noise models. The difference between standard VTS and extended VTS is interesting. By definition, both yield the same compensation for the statics. For the dynamics, however, the continuous time approximation does not consistently decrease the distance to the single-pass retrained system. Extended VTS, though not as accurate as extended DPMC, provides a substantial improvement over standard VTS.

[5] Similar trends were observed when striped noise statistics, consistent with diagonal standard noise models for VTS, were used for EVTS.

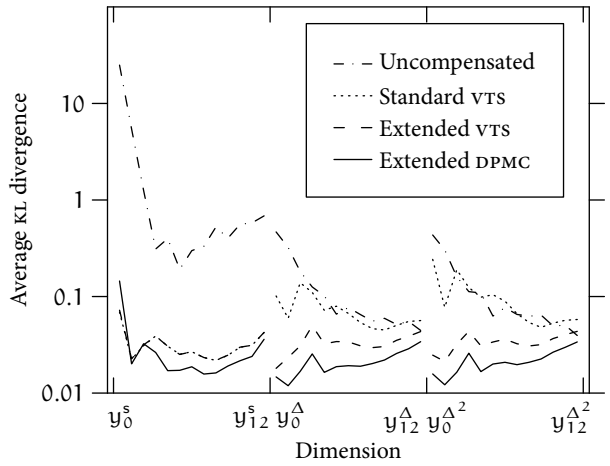

Figure 1 *Average Kullback-Leibler divergence between compensated systems and a single-pass retrained (ideal) system.*

5.1.2 Block-diagonal compensation

The previous section used diagonal covariance matrices. To compensate for changing correlations under noise more complex covariance matrix structures, such as full or block-diagonal, may be useful. Both VTS with the continuous time approximation, and EVTS, can also be used to generate block-diagonal covariance matrices for the output distributions. For this block diagonal case the corrupted speech covariance matrix will have the form (repeated from (11))

$$\boldsymbol{\Sigma}_y = \begin{bmatrix} \boldsymbol{\Sigma}_y^s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_y^\Delta \end{bmatrix}. \tag{43}$$

The KL divergence to a single-pass retrained system at the block level can then be used. This allows the compensation of each of the following blocks of features to be individually assessed: the statics, and first- and second-order dynamics. VTS compensation uses block-diagonal statistics for both the clean speech and noise models. For EVTS the extended statistics all have full covariance matrices.

Compensation	—	VTS	EVTS	EDPMC
$\mathbf{y}^s$	42.3	0.9	0.9	0.9
$\mathbf{y}^\Delta$	2.5	1.6	0.8	0.5
$\mathbf{y}^{\Delta^2}$	2.4	2.0	0.7	0.5

Table 2 *RM task: average KL divergence to a block-diagonal single-pass retrained system for VTS (continuous time), EVTS and DPMC at 20dB SNR.*

Table 2 shows the average KL divergence between a system compensated with block-diagonal VTS with the continuous time approximation and the block-diagonal SPR system. VTS finds compensated parameters close to the SPR system for the static features: the KL divergence goes from 42.3 to 0.9. However, the dynamic parameters are not compensated as accurately. Both the delta and delta-delta parameters are only slightly

closer to the SPR system than the uncompensated model set (2.4 to 2.0 for the delta-deltas). Similarly to diagonal compensation (see figure 1), with block-diagonal covariances standard VTS finds good compensation for the static parameters, but not for the deltas and delta-deltas.

EVTS has the same compensation as standard VTS for the statics. As in the diagonal-covariance case, however, for dynamic parameters compensation it is more accurate. It does yield a clear improvement over the uncompensated system (2.4 to 0.7 for the delta-deltas) and is close to EDPMC, which in the limit yields the best obtainable compensation.

5.2 Extended noise reconstruction

In practice labelled, sufficient, samples of the noise may not always be available to estimate the noise model. As discussed in section 4.2 when using the standard ML-estimated noise models there are two approaches to mapping the noise model parameters to the extended noise model parameters: one with a diagonal covariance matrix for the additive noise, and one with a smoothly reconstructed matrix. This section contrasts the performance of the two. The standard noise model parameters can either be derived from the actual noise data, the *known* case, or using ML-estimation, *estimated*. In this section the noise added to the RM task was scaled to yield 14dB SNR. Six Gaussian components per state clean speech models were used with striped covariance matrices.

	diag. $\boldsymbol{\Sigma}_y$		full $\boldsymbol{\Sigma}_y$	
$\boldsymbol{\Sigma}_n^e$	diag	smth	diag	smth
known	16.4	15.9	16.0	15.2
estimated	12.0	12.5	11.2	12.0

Table 3 *Resource Management task: word error rates for reconstructing an extended noise model at 14 dB SNR.*

The top row of table 3 compares diagonal and smoothed reconstructions of the extended noise when the standard noise model is estimated on the actual data. For diagonal-covariance and full-covariance compensation, smoothing results in 0.6 % and 0.8 % absolute improvement in the word error rate. However when the standard noise model parameters are estimated in an ML-fashion, the second line in table 3, the smoothing process degraded performance.

This degradation from the use of smoothing when using ML-estimated noise parameters is because of the nature of the ML-estimates. For the smoothing process it is assumed that there is some true underlying sequence of noise samples that yields the standard noise model parameters. This is guaranteed to be true for the known noise situation. However this is not necessarily the case for the estimated noise. The dynamics noise parameters are estimated using the continuous time approximation. There are no constraints that the estimates reflect a "true" sequence of noise samples. Thus by "improving" the extended noise covariance matrix using the smoothing approach, where relationships in the noise sample sequence are assumed, may not be helpful. The experiments in section 6 will therefore use the diagonal reconstruction for the extended noise distribution.

6 Experiments

The performance of extended VTS was examined on three tasks. Two used artificial noise, and are therefore useful to test the noise estimation and behaviour at various signal-to-noise ratios. AURORA 2 [8] is a well-known digit recognition task with artificial noise. The second task is the Resource Management [18] corpus artificially corrupted with NOISEX data [21]. The final task used real, in-car recorded data collected by Toshiba Research Europe. Results on this task can give insight in performance when other effects, such as the Lombard effect, may impact performance.

For all tasks, "clean" training data was used to train the speech models. 39-dimensional feature vectors were used: 12 MFCCs and the zeroth coefficient, augmented with deltas and delta-deltas. Unless indicated otherwise, the MFCCs were found with HTK [23] and the deltas and delta-deltas were computed over a window of 2 observations left and 2 right, making the total window width 9.

Unlike results in the previous section, for all tasks the noise models were estimated, finding the maximum likelihood noise model [14], as described in section 2.3, for compensating a clean system with VTS and the continuous time approximation. The initial noise model's Gaussian for the additive noise was the maximum-likelihood estimate from the first 20 and last 20 frames of the utterance, which were assumed to contain no speech. The initial convolutional noise estimate was $\mathbf{0}$. Given this initial noise estimate for an utterance, a recognition hypothesis was found. This was used to find component-time posteriors. Then, the noise means and the additive noise covariance were re-estimated. For EVTS the extended noise model was reconstructed from the standard noise model with diagonal covariance, as described in section 4.2.1.

6.1 AURORA task

AURORA 2 is a small vocabulary digit string recognition task [8]. Utterances are one to seven digits long and based on the TIDIGITS database with noise artificially added. The clean speech training data comprises 8440 utterances from 55 male and 55 female speakers. The test data is split into three sections. Test set A comprises 4 noise conditions: subway, babble, car and exhibition hall. Matched training data is available for these test conditions, but not used in this work. Test set B comprises 4 different noise conditions. For both test set A and B the noise was scaled and added to the waveforms. For the two noise conditions in test set C convolutional noise was also added. Each of the conditions has a test set of 1001 sentences with 52 male and 52 female speakers.

The feature vectors were extracted with the ETSI front-end [8]. The delta and delta-delta coefficients used 2 and 3 frames left and right, respectively, for a total window of 11 frames. The acoustic models were 16 emitting state whole word digit models, with 3 mixtures per state and silence. Since for this task the noise estimates did not contain zero elements in the variance, the back-off strategy for the noise estimate discussed in section 4.2.3 was not necessary.

Table 4 shows results for compensation with VTS with the continuous time approximation and EVTS. Both diagonal and block-diagonal forms of VTS were used. VTS with diagonal compensation (trained on diagonal speech statistics) is the standard method. Results for this are shown in the first three columns of table 4 and are treated as the baseline performance figures. VTS can also be used to produce block-diagonal covariance matrices. With diagonal-covariance clean speech statistics (not in the table), this

Scheme	VTS						EVTS		
Compensation	diagonal			block-diagonal			full		
SNR	A	B	C	A	B	C	A	B	C
00	28.2	26.2	25.9	24.3	22.9	23.6	23.5	24.3	22.6
05	10.5	9.3	9.9	8.2	7.9	8.2	7.1	7.2	6.9
10	4.3	3.9	4.4	3.3	3.2	3.4	2.5	2.4	2.8
15	2.2	2.2	2.3	1.9	1.8	1.8	1.2	1.2	1.4
20	1.6	1.4	1.6	1.3	1.2	1.2	0.8	0.7	1.0
Avg.	9.4	8.6	8.8	7.8	7.4	7.7	7.0	7.2	6.9

Table 4 *AURORA: diagonal compensation with standard VTS and full compensation with extended VTS.*

yielded no performance gain. However performance gains were obtained when using block-diagonal clean-speech models. The results for this are shown in the middle three columns of table 4. Compared to the standard diagonal VTS scheme, this gave, for example, relative reductions in word error rate of 15 % to 22 % at 5dB SNR.

The results for EVTS are shown in the last three columns of table 4. Here full covariance matrix extended clean speech models were used to produce compensated full covariance matrices for decoding. The improved compensation for dynamics causes extended VTS to perform better than to block-diagonal standard VTS in all but one noise conditions. At 5 dB again, relative improvements are an extra 3 % to 10 %.

The results presented here used the simple AURORA back-end. Large gains over the standard VTS approach (similar results for VTS are given in [12]) were obtained. Using the simple back-end recogniser, rather than one with more Gaussian components per state, has ensured that block-diagonal and full covariance matrix clean speech models can be robustly used. Though not always practical this indicates the possible gains from schemes such as EVTS on a standard task.

6.2 Resource Management task

The Resource Management task is a medium-vocabulary task, with a 1000-word vocabulary. A noise corrupted version of the this task was generated by adding Operations Room noise from the NOISEX-92 database scaled to yield SNRs of 20 dB and 14 dB. The training data contains 109 speakers reading 3990 sentences, 3.8 hours of data. State-clustered cross-word triphone models with 6 components per mixture were built using the HTK RM recipe. For this work the noise model was estimated per speaker. As in the AURORA 2 task, the noise covariance estimate did not contain any zero entries, so back-off as discussed in section 4.2.3 was not necessary. The number of samples per distribution for extended DPMC was set to 10 000 (performance did not improve with additional samples). All results are averaged over three of the four available test sets, Feb89, Oct89, and Feb91, a total of 30 test speakers and 900 utterances.

Table 5 shows contrasts between compensation with standard VTS and with extended feature vectors using either EVTS or EDPMC. The results in the first row are from the standard scheme, diagonal-covariance compensation with VTS. Block-diagonal compensation with standard VTS was also implemented and, as in the AURORA task, block-diagonal clean speech statistics were used. The results for this approach are in

Scheme	Σ_x	Σ_x^e	Decoding	20 dB	14 dB
VTS	diag	—	diag	6.8	13.7
	block	—	block	7.0	14.2
EVTS	—	striped	diag	6.2	12.0
			full	6.3	11.2
EDPMC	—	striped	diag	6.3	11.8
			full	6.0	11.2

Table 5 *Resource Management task: word error rates for standard VTS, extended VTS and extended DPMC.*

the second row. In contrast to the AURORA task, the use of the block-diagonal compensation with VTS degraded performance, for example 13.7 % to 14.2 % at 14 dB. This difference in performance between the tasks is felt to be because of the additional complexity of the RM task compared to AURORA.

For the extended systems EVTS and EDPMC, the extended clean speech statistics were striped (as discussed in section 4.1) for robustness. Compensation with EVTS (shown in the middle two rows of the table) yielded better performance than standard VTS for both diagonal and full compensated covariance matrices. For diagonal-covariance compensation, the relative improvement is around 10 % (6.8 to 6.2 %; 13.7 to 12.0 %) over standard VTS. Though at the higher SNR condition, 20 dB, full-covariance compensation did not make any difference in performance,[6] gains were observed at 14 dB SNR. At 14 dB full-covariance compensation produces an 11.2 % word error rate, which is a 20 % relative improvement from standard VTS, and 7% relative gain compared to the diagonal case.

In addition table 5 shows the performance of EDPMC, which in the limit can be viewed as the "optimal" compensation scheme. The results for this approach are shown in the bottom two rows of table 5. When compared with EDPMC, the first-order approximation in EVTS degrades performance by at most 0.3 % absolute. However, EVTS is significantly faster than EDPMC.

6.3 Toshiba in-car task

Experiments were also run on a task with real recorded noise: the Toshiba in-car database. This is a corpus collected by Toshiba Research Europe Limited's Cambridge Research Laboratory. It comprises a set of small/medium sized tasks with noisy speech collected in an office and in vehicles driving at various conditions. This work used three of the test sets containing digit sequences (phone numbers) recorded in a car with a microphone mounted on the rear-view mirror. The ENON set, which consists of 835 utterances, was recorded with the engine idle, and has a 35 dB average signal-to-noise ratio. The CITY set, which consists of 862 utterances, was recorded driving in the city, and has a 25 dB average signal-to-noise ratio. The HWY set, which consists of 887 utterances, was recorded on the highway, and has a 18 dB average signal-to-noise ratio. The clean speech models were trained on the Wall Street Journal corpus, based on the system described in [16], but the number of states was reduced to about 650, more appropriate for an embedded system. The acoustic models used were decision tree clustered

[6] The difference between diagonal and full EVTS, 6.249 % and 6.262 %, is only one word.

state, cross-word triphones, with three emitting states per HMM, twelve components per GMM and diagonal covariance matrices. The number of components was about 7800. For the EVTS scheme extended clean speech statistics were again striped for robustness. The language model was an open digit loop. In the noise estimation stage, the noise model was re-estimated twice on a new hypothesis.

Scheme	Decoding	ENON 35 dB	CITY 25 dB	HWY 18 dB
VTS	diag	1.2	2.5	3.2
EVTS	diag	1.1	2.4	2.8
	full	1.7	2.5	2.4
EVTS	back-off	1.1	2.2	2.4
% utterances		87 %	38 %	11 %

Table 6 *Extended VTS on the Toshiba in-car task.*

Table 6 shows results on the Toshiba task. The top row contains word error rates for the standard compensation method: VTS trained on diagonal speech statistics. The performance of EVTS using diagonal covariance matrices is shown in the second row. Again EVTS shows gains over VTS, especially at the lowest SNR condition, HWY. In the HWY condition about a 12% relative reduction in error rate was obtained.

Initially full-covariance matrix compensation with EVTS was evaluated without the use of the back-off scheme described in section 4.2.3. Using EVTS with full-covariance decoding yielded additional gains compared to diagonal compensation at low SNRS (2.8 % to 2.4 %). However the performance was degraded at higher SNR conditions, for example ENON where performance was degraded from 1.1 % to 1.7 %.

In contrast to the previous tasks, at high SNRS there were found to be zeros in the noise variance estimate. The back-off scheme, labelled "EVTS back-off" in table 6, was therefore used. Here diagonal covariance matrix compensation was used if any noise variance estimate fell below 0.05 times the variance floor used for clean speech model training (results were consistent over a range of values from 0.0 to 0.1). The bottom line in table 6 shows the percentage of utterances for each of the task where the system was backed off to diagonal covariance matrix compensation. As expected the percentage at high SNRS, 87%, was far higher than at lower SNRS, 11%. Using this back-off approach gave consistent gains over using either diagonal or full compensation EVTS alone. Note as the back-off is based on the ML-estimated noise variances it is fully automated. Compared to standard VTS, EVTS with back-off gave relative reductions of 25% in the HWY condition, 12% in CITY, and 8% in ENON.

7 Conclusion

A popular form of model-based compensation schemes to handle changes in background noise is VTS. In order to achieve the best performance it is necessary to compensate all the model parameters to reflect the impact of the background noise on the speech. Though it is easy to define a mismatch function for the static parameters, it is non-trivial to specify one for the dynamic parameters. To address this the continuous time approximation is often used. This paper has introduced an alternative, more accurate, approximation, extended VTS (EVTS). Here the distribution over dynamic param-

eters is computed based on a linear transformation of a window of static parameters. By using this more accurate dynamic parameter compensation scheme it is possible to use more complex covariance matrices in the compensated system. Computing full-covariance matrix compensation is more sensitive to approximations of the mismatch function. EVTS allows accurate full covariance matrices to be generated from the compensation process. This enables the acoustic models to better reflect any changes in the correlation in the feature space that result from the varying noise conditions.

The new method was tested on AURORA 2, a noise-corrupted Resource Management task, and a Toshiba in-car corpus. With a noise model estimated with maximum likelihood training for standard VTS, extended VTS obtained about 10 % relative reduction in error rate over standard VTS for diagonal compensation at higher signal-to-noise ratios, and about 20 % for full-covariance compensation at lower signal-to-noise ratios.

Though EVTS yields reductions in word error rate with full covariance matrices, there are a number of refinements that can be implemented. First to address the computational cost, JUD or predictive approaches can be used. The current implementation of EVTS uses noise model estimates based on VTS with the standard continuous time approximation. Improved performance may be obtained by using noise estimates based on EVTS. Also, canonical model parameters could be estimated with adaptive training, similar to [10]. Finally, improved efficient approximations, for example using unscented transformations, may also be used. In initial experiments this was found to yield gains with EVTS-style approaches. All these approaches will be examined in future work.

A Joint uncertainty decoding with extended VTS

Joint uncertainty decoding (JUD) [13] is a model compensation method that models the relation between the clean speech and the noise with a jointly Gaussian distribution per base class. If the joint distribution for a base class is

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_x & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_y \end{bmatrix} \right), \tag{44}$$

then the JUD compensation for component m in that base class is of the form

$$p(\mathbf{y}|m) = |\mathbf{A}| \mathcal{N}\left(\mathbf{A}\mathbf{y} + \mathbf{b}\,;\, \boldsymbol{\mu}_x^{(m)}\,,\, \boldsymbol{\Sigma}_x^{(m)} + \boldsymbol{\Sigma}_{\text{bias}} \right). \tag{45}$$

In this expression, $\mathbf{A}$, $\mathbf{b}$, and $\boldsymbol{\Sigma}_{\text{bias}}$ are functions of the joint distribution in (44).

The per-base class joint distribution can be computed by applying a model compensation method. $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ can be set to the overall Gaussian of the base class. The model compensation method can be used straightforwardly to find $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$. For VTS compensation, the expression is exactly (11). However, a model compensation method needs an addendum to find the cross-covariance $\boldsymbol{\Sigma}_{yx}$. ($\boldsymbol{\Sigma}_{xy} = \boldsymbol{\Sigma}_{yx}^T$, as the joint distribution's covariance matrix is symmetric.) For VTS, the addendum for computing the cross-covariance is given in [14, 22].

Computing the joint distribution in (44) for a base class with extended VTS starts by computing the joint distribution of the extended clean and corrupted speech

$$\begin{bmatrix} \mathbf{x}^e \\ \mathbf{y}^e \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x^e \\ \boldsymbol{\mu}_y^e \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_x^e & \boldsymbol{\Sigma}_{xy}^e \\ \boldsymbol{\Sigma}_{yx}^e & \boldsymbol{\Sigma}_y^e \end{bmatrix} \right). \tag{46}$$

A distribution for the extended clean speech for a base class $\mathbf{x}^e \sim \mathcal{N}(\boldsymbol{\mu}_x^e, \boldsymbol{\Sigma}_x^e)$ can be found from a model set over extended feature vectors. From that distribution, $\mathbf{y}^e \sim \mathcal{N}\left(\boldsymbol{\mu}_y^e, \boldsymbol{\Sigma}_y^e\right)$ can be found in the same way as normal extended VTS does, in (25). That leaves the cross-covariance $\boldsymbol{\Sigma}_{yx}^e$ to be computed. $\boldsymbol{\Sigma}_{yx}^e$ has a structure (analogous to (19)):

$$\boldsymbol{\Sigma}_{yx}^e = \begin{bmatrix} \boldsymbol{\Sigma}_{y_{-1}x_{-1}}^s & \boldsymbol{\Sigma}_{y_{-1}x_0}^s & \boldsymbol{\Sigma}_{y_{-1}x_{+1}}^s \\ \boldsymbol{\Sigma}_{y_0x_{-1}}^s & \boldsymbol{\Sigma}_{y_0x_0}^s & \boldsymbol{\Sigma}_{y_0x_{+1}}^s \\ \boldsymbol{\Sigma}_{y_{+1}x_{-1}}^s & \boldsymbol{\Sigma}_{y_{+1}x_0}^s & \boldsymbol{\Sigma}_{y_{+1}x_{+1}}^s \end{bmatrix}. \tag{47}$$

The blocks of this can each be found analogously to (25). For example, noting that the clean speech and the noise are assumed independent,

$$\begin{aligned} \boldsymbol{\Sigma}_{y_0x_{+1}}^s &= \mathcal{E}\left\{(\mathbf{y}_{t,\text{vts}}^s - \boldsymbol{\mu}_{y_0})(\mathbf{x}_{t+1}^s - \boldsymbol{\mu}_{x_{+1}})^T\right\} \\ &= \mathcal{E}\left\{\left(\mathbf{J}_0(\mathbf{x}_t^s - \boldsymbol{\mu}_{x_0}^s) + (\mathbf{I} - \mathbf{J}_0)(\mathbf{n}_t^s - \boldsymbol{\mu}_{n_0}^s)\right)\left(\mathbf{x}_{t+1}^s - \boldsymbol{\mu}_{x_{+1}}^s\right)^T\right\} \\ &= \mathcal{E}\left\{\mathbf{J}_0(\mathbf{x}_t^s - \boldsymbol{\mu}_{x_0}^s)(\mathbf{x}_{t+1}^s - \boldsymbol{\mu}_{x_{+1}}^s)^T\right\} \\ &= \mathbf{J}_0\boldsymbol{\Sigma}_{x_0x_{+1}}^s. \end{aligned} \tag{48}$$

These form the blocks of the cross-covariance $\boldsymbol{\Sigma}_{yx}^e$, which completes the estimation the extended joint distribution (46).

As an aside, the extended joint distribution can also be generated with extended DPMC [20]. The natural extension of extended DPMC to joint uncertainty decoding estimates the complete joint distribution and implicitly includes the cross-covariance. Extended DPMC (see section 3.5) draws a sample $\mathbf{x}^{e(k)}$ from the extended clean speech distribution and one $\mathbf{n}^{e(k)}$ from the extended additive noise distribution, and combines them to produce a corrupted speech sample $\mathbf{y}^{e(k)}$. Maximum likelihood estimation then finds the compensated distribution from the corrupted speech samples $\mathbf{y}^{e(k)}$ only. The straightforward extension to finding the joint distribution is to train it on joint samples $\begin{bmatrix} \mathbf{x}^{e(k)} \\ \mathbf{y}^{e(k)} \end{bmatrix}$. The extended joint distribution that results implicitly includes the cross-correlation.

The next step is to transform the joint distribution over extended feature vectors, whether generated with extended VTS or extended DPMC, into the standard domain. A joint feature vector of the extended clean speech and the extended noise-corrupted speech can be converted to a feature vector with statics and dynamics with

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}^e \\ \mathbf{y}^e \end{bmatrix}. \tag{49}$$

Therefore, the same transformation can be applied to the distribution in (46):

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{D}\boldsymbol{\mu}_x^e \\ \mathbf{D}\boldsymbol{\mu}_y^e \end{bmatrix}, \begin{bmatrix} \mathbf{D}\boldsymbol{\Sigma}_x^e\mathbf{D}^T & \mathbf{D}\boldsymbol{\Sigma}_{xy}^e\mathbf{D}^T \\ \mathbf{D}\boldsymbol{\Sigma}_{yx}^e\mathbf{D}^T & \mathbf{D}\boldsymbol{\Sigma}_y^e\mathbf{D}^T \end{bmatrix}\right), \tag{50}$$

so that the parameters of the standard-domain joint distribution in (44) are

$$\boldsymbol{\mu}_x = \mathbf{D}\boldsymbol{\mu}_x^e; \qquad \boldsymbol{\mu}_y = \mathbf{D}\boldsymbol{\mu}_y^e; \tag{51a}$$

$$\boldsymbol{\Sigma}_x = \mathbf{D}\boldsymbol{\Sigma}_x^e\mathbf{D}^T; \qquad \boldsymbol{\Sigma}_y = \mathbf{D}\boldsymbol{\Sigma}_y^e\mathbf{D}^T; \qquad \boldsymbol{\Sigma}_{yx} = \mathbf{D}\boldsymbol{\Sigma}_{yx}^e\mathbf{D}^T. \tag{51b}$$

Given these parameters per base class, decoding uses the same form as standard joint uncertainty decoding, in (45).

B Unscented transformation

Approximations like the vector Taylor series one or DPMC are necessary for noise compensation because the exact parameters of the clean speech distribution are found by:

$$p(\mathbf{y}) = \int\int \mathbf{f}(\mathbf{x}, \mathbf{n}, \mathbf{h})\, p(\mathbf{n})\, p(\mathbf{x}) d\mathbf{n} d\mathbf{x}. \tag{52}$$

Since the mismatch function $\mathbf{f}$ is non-linear, the distribution of the noise-corrupted speech is non-Gaussian. VTS linearises the mismatch function so that the corrupted speech becomes Gaussian. It is also possible to approximate distributions $p(\mathbf{n})$ and $p(\mathbf{x})$ in (52). The *unscented transformation* [9] uses the intuition that "it is easier to approximate a probability distribution than it is to approximate an arbitrary non-linear function or transformation".

This method chooses a set of samples, called *sigma points*, that represent the distribution of the inputs to the function. The samples are chosen to exhibit certain properties, such as having the correct sample mean and covariance of the distribution. They are usually points on a covariance contour. The function is applied to each of these points, and the statistics of the outputs give an estimate of the parameters of the corrupted speech distribution.

This is similar to Monte Carlo methods such as DPMC (see section 2.2), but there are several differences. First, the choice of the sigma points is deterministic. Second, the weights applied to points are not necessarily in the interval $[0, 1]$. Negative weights are possible, and indeed occur in may cases. Third, the number of samples by definition grows linearly in the number of dimensions.

The statistics of the outputs are weighted versions of DPMC's in (14). If $w^{(k)}$ is the weight of the kth sample, let the total weight be $w = \sum_k w^{(k)}$. Calling the input sample $\mathbf{z}^{(k)}$, the corresponding output is

$$\mathbf{o}^{(k)} = \mathbf{f}\left(\mathbf{z}^{(k)}\right). \tag{53}$$

The mean and the covariance of the output become

$$\boldsymbol{\mu}_o = \frac{1}{w} \sum_k w^{(k)} \mathbf{o}^{(k)}; \tag{54a}$$

$$\boldsymbol{\Sigma}_o = \left(\frac{1}{w} \sum_k w^{(k)} \mathbf{o}^{(k)} [\mathbf{o}^{(k)}]^T \right) - \boldsymbol{\mu}_o \boldsymbol{\mu}_o^T. \tag{54b}$$

One option for selecting samples that has the same sample mean and variance as a Gaussian $\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ is to generate samples on a covariance contour. If $\mathbf{z}$ has d dimensions, generate d pairs of symmetric samples

$$\mathbf{z}^{(k)} = \boldsymbol{\mu}_z + \left(\sqrt{d\boldsymbol{\Sigma}_z}\right)_i^T; \tag{55a}$$

$$\mathbf{z}^{(d+k)} = \boldsymbol{\mu}_z - \left(\sqrt{d\boldsymbol{\Sigma}_z}\right)_i^T, \tag{55b}$$

where $\sqrt{\boldsymbol{\Sigma}_z}$ is the matrix squared root of $\boldsymbol{\Sigma}_z$ (for example, the Choleski decomposition), and $(\cdot)_i^T$ denotes the transpose of the ith row. The weights $w^{(k)} = 1$ of these samples are all set equal. It is also possible to augment this set with the actual mean

and adjust the other samples and the weights. For example, setting the weight of the mean to $w^{(0)} = 2d - \frac{2}{3}d^2$ matches some of the fourth-order moments [9]. Note that for more than 3 dimensions this weight is negative.

The unscented transformation has been applied to noise-robustness in speech recognition before [19]. This was done at the front-end. The joint distribution was found with unscented transformation, with the source vector composed of static coefficients of the clean speech and the additive noise:

$$z = \begin{bmatrix} x^s \\ n^s \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x^s \\ \mu_n^s \end{bmatrix}, \begin{bmatrix} \Sigma_x^s & 0 \\ 0 & \Sigma_n^s \end{bmatrix} \right). \tag{56}$$

Here, $\mathcal{N}(\mu_x^s, \Sigma_x^s)$ gives the clean speech distribution for the front-end component, and $\mathcal{N}(\mu_n^s, \Sigma_n^s)$ the distribution of the additive noise. The unscented transformation generated transformed samples to estimate a component of joint distribution over the clean speech x^s and the corrupted speech y^s. Each feature vector y^s was then transformed into a minimum mean square error estimate of the clean speech. Only then were dynamic coefficients added to the feature vector.

To apply the unscented transformation in the model space, the dynamic parameters need to be compensated explicitly. This can be done with extended feature vectors, by estimating an extended distribution and converting it to a distribution of statics and dynamics. This is similar to extended DPMC, but the samples are drawn deterministically. The source vector is composed of extended feature vectors of the clean speech and the additive noise:

$$z = \begin{bmatrix} x^e \\ n^e \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x^e \\ \mu_n^e \end{bmatrix}, \begin{bmatrix} \Sigma_x^e & 0 \\ 0 & \Sigma_n^e \end{bmatrix} \right). \tag{57}$$

Through the unscented transformation, an extended noise-corrupted speech distribution is then acquired, which is converted to standard-domain parameters with (18).

B.1 Preliminary results

Preliminary experiments using the unscented transformation on extended feature vectors were performed on the Resource Management task discussed in section 6.2. The system used one component per state to guarantee robustness. The clean speech statistics had full covariance. The noise model was trained on the known noise and had a full-covariance additive noise model.

The sigma points for the unscented transformation were chosen as in (55). Adding the mean to capture fourth-order moments was also tried. However, because of the high dimensionality of twice 117 (13 MFCCs and 9 frames in a window) for the sources, the weight for the mean became negative and large, and the other sample points were on a very tight covariance contour. This led to invertibility problems with the resulting corrupted speech matrix. Therefore, only points on the covariance contour were used, as in (55).

Table 7 contains word error rates comparing the extended versions of VTS, DPMC, and unscented transformation. When decoding uses diagonal covariances, the full Monte Carlo method, extended DPMC, performs best. The unscented transformation attains performance similar to extended VTS, and 0.4 % absolute worse than extended DPMC. However, when full covariances are used in decoding, compensation with the

Scheme	Known 20 dB Decoding diag	Known 20 dB Decoding full	Known 14 dB Decoding diag	Known 14 dB Decoding full
eVTS	12.6 %	11.3 %	21.8 %	18.8 %
eDPMC	12.4 %	10.0 %	21.4 %	18.4 %
eUT	12.8 %	9.4 %	21.8 %	16.7 %

Table 7 *Resource Management: word error rates for compensation with the unscented transformation with extended feature vectors compared with eVTS and eDPMC.*

unscented transformation outperforms extended DPMC by 0.6 % and 1.7 % absolute. The same trends are visible on results (not shown in the table) on a 6-component system. This is surprising, because extended DPMC generates 10 000 (albeit random) samples, wheres the unscented transformation generates $13 \times 9 \times 2 \times 2 = 468$. It is conjectured that the unscented transformation is biased towards compensation that allows for more discrimination. However, more research is needed.

Bibliography

[1] Alex Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. PhD thesis, Carnegie Mellon University, 1990.

[2] Alex Acero, Li Deng, Trausti Kristjansson, and Jerry Zhang. HMM adaptation using vector Taylor series for noisy speech recognition. In *Proceedings of ICSLP*, volume 3, pages 229–232, 2000.

[3] Ángel de la Torre, Dominique Fohr, and Jean-Paul Haton. Statistical adaptation of acoustic models to noise conditions for robust speech recognition. In *Proceedings of ICSLP*, pages 1437–1440, 2002.

[4] Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59, 1986.

[5] M. J. F. Gales and R. C. van Dalen. Predictive linear transforms for noise robust speech recognition. In *Proceedings of ASRU*, pages 59–64, 2007.

[6] Mark J. F. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 1995.

[7] R. A. Gopinath, M. J. F. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny. Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task. In *ARPA Workshop on Spoken Language System Technology*, pages 127–130, 1995.

[8] Hans-Günter Hirsch and David Pearce. The AURORA experimental framework for the performance evaluation of speech recognition systems under noise conditions. In *Proceedings of ASR*, pages 181–188, 2000.

[9] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[10] O. Kalinli, M.L.Seltzer, and A.Acero. Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition. In *Proceedings of ICASSP*, pages 3825–3828, April 2009.

[11] Do Yeong Kim, Chong Kwan Un, and Nam Soo Kim. Speech recognition in noisy environments using first-order vector Taylor series. *Speech Communication*, 24:39–49, 1998.

[12] Jinyu Li, Li Deng, Dong Yu, Yifan Gong, and Alex Acero. High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series. In *Proceedings of ASRU*, pages 65–70, 2007.

[13] H. Liao and M. J. F. Gales. Uncertainty decoding for noise robust speech recognition. In *Proceedings of Interspeech*, 2005.

[14] H. Liao and M. J. F. Gales. Joint uncertainty decoding for robust large vocabulary speech recognition. Technical Report CUED/F-INFENG/TR.552, Cambridge University Engineering Department, November 2006.

[15] H. Liao and M. J. F. Gales. Adaptive training with joint uncertainty decoding for robust recognition of noisy data. In *Proceedings of ICASSP*, volume IV, pages 389–392, 2007.

[16] Hank Liao. *Uncertainty Decoding for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 2007.

[17] Pedro J. Moreno. *Speech Recognition in Noisy Environments*. PhD thesis, Carnegie Mellon University, 1996.

[18] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett. The DARPA 1000-word resource management database for continuous speech recognition. In *Proceedings of ICASSP*, volume 1, pages 651–654, 1988.

[19] Yusuke Shinohara and Masami Akamine. Bayesian feature enhancement using a mixture of unscented transformations for uncertainty decoding of noisy speech. In *Proceedings of ICASSP*, pages 4569–4572, 2009.

[20] R. C. van Dalen and M. J. F. Gales. Covariance modelling for noise robust speech recognition. In *Proceedings of Interspeech*, pages 2000–2003, 2008.

[21] A. Varga and H. J. M. Steeneken. Assessment for automatic speech recognition II: NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251, 1993.

[22] Haitian Xu, Luca Rigazio, and David Kryze. Vector Taylor series based joint uncertainty decoding. In *Proceedings of Interspeech*, pages 1125–1128, 2006.

[23] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. The HTK book (for HTK version 3.4), 2006.