NOISY SPEECH SEGMENTATION USING NON-LINEAR OBSERVATION SWITCHING STATE SPACE MODEL AND UNSCENTED KALMAN FILTERING

Pamornpol Jinachitra

Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
pj97@ccrma.stanford.edu

ABSTRACT

A reliable speech segmentation in noisy environments is desirable for segment-based speech enhancement and efficient coding. Switching state space model with hidden dynamics has been shown to lend itself naturally to the speech segmentation problem. However, when noise is present, the distorted observation features lead to a poor recognition and segmentation performance. In this paper, the Unscented Kalman Filtering (UKF) is used during inference to compensate nonlinearly for the effect of noise on the observed features in the log-frequency domain. The proposed algorithms resulted in a much improved segmentation performance in a variety of noises.

1. INTRODUCTION

Switching state-space model (SSM) has been used to model speech for recognition and segmentation applications [1] [2]. While conventional HMM can only model a sequence of discrete states, which usually correspond to some speech units, SSM can also model a smooth continuous hidden dynamics, reflecting the speech production process [3]. This results in a compact representation of speech with a good physically-related constraint which allows for deviation caused by co-articulation and hence a more robust decoding.

While SSM has also been applied to noisy speech for speech enhancement, both for listening [4] and as an enhanced front-end feature extraction to a speech recognizer [5], none has been reported for segmentation where phonetically meaningful units are desired. On the other hand, SSM used in such speech segmentation, such as in [1], has not attempted to address the problem when noise is present. In this paper, we focus on the use of SSM for speech segmentation under noisy circumstances. Although, the optimal data-driven units, obtainable through learning, have shown a better performance than phone units in speech recognition tasks [2], some applications, such as the segment-based speech enhancement and the phoneme-based coding, require accurate phone segmentation.

The clean speech model presented here is an extension from [1] to a mixture of Gaussians model, similarly to [2] and [3]. Using a mixture model can lead to a significant improvement over a single Gaussian model. One of the difficulties involving the use of SSM is the intractable exact inference. In this paper, the Generalized Pseudo-Bayesian of order one and two (GPB1 and GPB2) are presented as the way to do approximate inferences. An additive noise in time affects most frequency-domain features nonlinearly. For log-frequency type of features, an approximate nonlinear observation model can be obtained. To deal with nonlinearity in the observation model, the Unscented Kalman Filter (UKF) is employed during the inference step.

In the remainder of this paper, we describe in more details the SSM mixture model employed, the learning and then the approximate inference to decode the phoneme class sequence from a noisy observation. We finish by presenting the results of segmentation among various choices of parameters, algorithms and background noises.

P.Jinachitra is sponsored by Toyota ITC, Palo Alto, USA.

2. SWITCHING STATE SPACE OF MIXTURE MODELS

A clean speech utterance can be modeled using the following equations of standard SSM (see [1] for notation description)

$$\mathbf{x}_t = \mathbf{A}_m(S_t) \cdot \mathbf{x}_{t-1} + \mathbf{v}_t(S_t) \quad (1)$$

$$\mathbf{y}_t = \mathbf{C}_m(S_t) \cdot \mathbf{x}_t + \mathbf{D}_m(S_t) + \mathbf{w}_t(S_t) \quad (2)$$

$$\mathbf{v}(S_t) \sim \mathcal{N}(0, \mathbf{Q}_m(S_t)), \mathbf{w}(S_t) \sim \mathcal{N}(0, \mathbf{R}_m(S_t))$$

$$T(i,j) = Pr(S_t = i | S_{t-1} = j), i, j = 1, \ldots, |S| \quad (3)$$

$$Pr(S_0 = i) = \pi_i, i = 1, \ldots, |S| \quad (4)$$

The hidden dynamic model at each time instant is determined by the discrete state at that instant, S_t. The discrete states represent the phone classes while the continuous states reflect the slowly varying articulatory-related parameters. The number of phone classes is $|S|$, each consisted of $m = 1, \ldots, M$ dynamic systems, i.e., within each class, from one step to the next, there are M possible trajectories and hence M output Gaussians, modeling output distributions which may be non-Gaussian or multimodal. The mixture distribution is mixed through the prior probability weights, $\alpha_m = Pr(m|S)$, which will be obtained through training.

2.1. Learning

The transition matrix and the prior probability in (3) and (4) are both learned by counting from labeled frames of clean speeches in the training set. Given the discrete state sequence, the continuous state space parameters can be learned through the Expectation-Maximization (EM) algorithm as follows:
E step : For each m, find the sufficient statistics

$$\hat{\mathbf{x}}_{t|T_k} = E(\mathbf{x}_t | \mathbf{y}_{1:T_k}) \quad (5)$$

$$V_{t|T_k} = \text{cov}(\mathbf{x}_t \mathbf{x}_t' | \mathbf{y}_{1:T_k}) \quad (6)$$

$$V_{t,t-1|T_k} = \text{cov}(\mathbf{x}_t \mathbf{x}'_{t-1} | \mathbf{y}_{1:T_k}) \quad (7)$$

$$< \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t|T} >= V_{t,t|T_k} + \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t|T} \quad (8)$$

$$< \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t-1|T} >= V_{t,t-1|T_k} + \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t-1|T} \quad (9)$$

The statistics as shown of each component m are obtained by Kalman smoothing each phone segment, k, and collecting over all K_s phone class segments. To avoid having to collapse M distributions at every time-step, we assume that each component propagate separately during the phone segment. Since each phone segment is now represented by a mixture of components, at the end of each segment, the smoothed states are combined to form a single Gaussian state through moment matching for use in the following phone segment. If the segment is too short, having only one frame, only filtering is used to arrive at the required statistics from that segment.

M step : Denoting $\mathbf{y}_t - D_m$ by $\bar{\mathbf{y}}$ where m is omitted, subjecting to context, ML estimates of the system parameters can be found from

$$\mathbf{A}_m = \left[\sum_{k=1}^{K_s} \sum_{t=2}^{T_k} \omega_t(m) < \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t-1|T} >_m \right] \cdot \left[\sum_{k=1}^{K_s} \sum_{t=2}^{T_k} \omega_t(m) < \hat{\mathbf{x}}_{t-1|T} \hat{\mathbf{x}}'_{t-1|T} >_m \right]^{-1} \quad (10)$$

$$\mathbf{C}_m = \left[\sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m) \bar{\mathbf{y}}_{k,t} \hat{\mathbf{x}}'_{t-1|T,m} \right] \cdot \left[\sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m) < \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t|T} >_m \right]^{-1} \quad (11)$$

$$\mathbf{D}_m = \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m) \cdot \mathbf{y}_t / \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m) \quad (12)$$

$$\mathbf{Q}_m = \frac{\sum_{k=1}^{K_s} \sum_{t=2}^{T_k} < \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t|T} >_m - A_m < \hat{\mathbf{x}}_{t|T} \hat{\mathbf{x}}'_{t-1|T} >_m}{\sum_{k=1}^{K_s} \sum_{t=2}^{T_k} \omega_t(m)} \quad (13)$$

$$\mathbf{R}_m = \frac{\sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \left[\omega_t(m) (\bar{\mathbf{y}}_{k,t} \bar{\mathbf{y}}'_{k,t} - \mathbf{C}_m \hat{\mathbf{x}}_{t|T,m} \bar{\mathbf{y}}'_{k,t}) \right]}{\sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m)} \quad (14)$$

$$\bar{\hat{\mathbf{x}}}_{1,m} = \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_1(m) \hat{\mathbf{x}}_{1,m} / \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_1(m) \quad (15)$$

$$\bar{\hat{V}}_{1,m} = \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_1(m) (\mathbf{x}_{1,m} - \bar{\hat{\mathbf{x}}}_{1,m}) (\hat{\mathbf{x}}_{1,m} - \bar{\hat{\mathbf{x}}}_{1,m})' / \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_1(m) \quad (16)$$

$$\alpha_m = \sum_{k=1}^{K_s} \sum_{t=1}^{T_k} \omega_t(m) / \sum_{k=1}^{K_s} T_k \quad (17)$$

where $\bar{\hat{\mathbf{x}}}_1$ and $\bar{\hat{V}}_1$ are initial state and covariance estimates at time $t = 1$ respectively. As usual, to avoid scaling ambiguity between C_m and Q_m, C_m is constrained to have unit columns. Also,

$$\begin{aligned} \omega_t(m) &= p(m_t | \mathbf{y}_t, S_t, \Theta_t) \\ &= \frac{p(\mathbf{y}_t | \mathbf{x}_t, m, \Theta) \cdot p(m_t | S_t, \Theta_t)}{\sum_{m'=1}^{M} p(\mathbf{y}_t | \mathbf{x}_t, m', \Theta) \cdot p(m'_t | S_t, \Theta_t)} \\ &= \frac{L(m) \cdot \alpha_m}{\sum_{m'=1}^{M} L(m') \cdot \alpha_{m'}} \end{aligned} \quad (18)$$

where Θ_t represents all system parameters at time t for that iteration. L is the likelihood obtained from filtering or smoothing. Note that each m is conditional on the state S so where an index m is shown, it refers to the component m within a class S and hence S is omitted for cleaner presentation.

The parameters $\mathbf{C}_m$, $\mathbf{R}_m$ and α_m were initialized using probabilistic PCA for mixture distributions [6]. $\mathbf{A}$ and $\mathbf{Q}$ are initialized using linear regression from the projected hidden states. $\mathbf{Q}$ and $\mathbf{R}$ are forced to be diagonal for numerical stability during learning.

2.2. Inference

The ultimate goal is to find the globally smoothed posterior probability sequence, $Pr(S_{1:T} | \mathbf{Y}_{1:T})$, given the observations. This is accomplished by using the likelihood calculated from the filtering and smoothing and the trained discrete state transition matrix, as in normal GPB1 or GPB2 algorithm. The difference in this work from others is the use of mixture model in GPB1 and GPB2 as well as the noise compensation in inference using UKF and a nonlinear noisy observation model. The classification of each frame is taken to be the class with the highest smoothed posterior (MAP solution). Phone classifications through smoothing result in smooth and robust segmentation with few spurious decisions. Smoothing also gives a sharper response and hence more accurate segmentation than filtering.

2.2.1. Nonlinear Noisy Observation Model

When the observed features are based in the log-frequency domain, for example, the MFCC, the effect of noise can be approximately expressed by [5]

$$Z(k) \approx F \cdot \log(10^{F^\dagger \cdot Y(k)} + 10^{F^\dagger \cdot N(k)}) \quad (19)$$

where $Z(k)$, $Y(k)$ and $N(k)$ are the k^{th} MFCC of the observation, the clean speech and the additive noise, respectively, as calculated from power spectral output of a Mel filter bank. F is the DCT matrix used to convert a log power spectrum to a cepstrum, whereas $F^\dagger$ is its right-inverse matrix such that $F \cdot F^\dagger = I$. When only log-Mel filter bank (LMFB) outputs are used, F and $F^\dagger$ are omitted. The expression for log-Mel filter bank is precise only if the clean speech and the noise are in-phase, hence the approximation [7].

2.2.2. GPB1 and GPB2 Inference Using UKF

In addition to the set of equation (1)-(4), which models a clean speech, we now have another layer of the generative model for the noisy speech observation $Z(t)$ as given by equation (19). As mentioned earlier, we employed GPB1 and GPB2, with an incorporation of the UKF, as approximate inference methods.

The basic operation of the GPB(r) algorithm is to approximate $|S|^t$ Gaussian components at time t by $|S|^{(r-1)}$ Gaussians using moment matching. The resulted Gaussian after collapsing through moment matching is optimal in the Kullback-Leibler sense and the error can be shown to be bounded despite the approximations. In the forward-pass, at each time step, the current state is filtered using each of the class's component systems. For GPB1, the collapsing is done at each step, keeping only one state estimates, while for GPB2, we keep $|S|$ state estimates, one for each possible discrete state. See [8] for more details on GPB1 and GPB2 on SSM.

UKF is a method of state inference in nonlinear dynamical system [9]. Instead of linearizing the nonlinearity as in the extended Kalman filtering (EKF), UKF updates the states by passing a deterministically sampled set of points that characterize the current state's distribution through the dynamic system and approximate the filtered distribution from them. The accuracy is generally up to second order, which is better than EKF while demanding similar computational load. In fact, our experiments using EKF for state update have been unsuccessful, without regularization, partially due to numerical instability of the Jacobian mainly caused by the exponential in the observation model.

For GPB1, UKF can be directly applied in the filtering step, giving filtered hidden state and covariance estimates [9]. Smoothing is not allowed in GPB1, however, but we can approximate it by simply repeating the filtering process but in the time-reverse manner. The smoothed posterior is calculated from the filtered posteriors and the transition matrix, backward in time. On the other hand, GPB2 allows for smoothing but this requires an estimation of the cross-covariance between adjacent time steps which is non-standard for the UKF. It can be shown that the filtered cross-covariance, $V_{t,t-1|t}$, can be expressed as

$$V_{t,t-1|t} = A_t V_{t,t|t} - K_t E[(\mathbf{z}_t - \hat{\mathbf{z}}_t^-)(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}^-)] \quad (20)$$

$$E[(\mathbf{z}_t - \hat{\mathbf{z}}_t^-)(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}^-)] \approx \sum_i \mathcal{W}_i^{(c)} [(\mathcal{Z}_t)_i - \mathbf{z}_t^-][(\mathcal{X}_{t-1})_i - \mathbf{x}_{t-1}^-]^T \quad (21)$$

where $\mathcal{W}_i^{(c)}$ is the conventional Unscented Transform weight for the i^{th} sample point, $(\mathcal{Z}_t)_i$ is the unscented transform output point from its corresponding "Sigma point", $(\mathcal{X}_t)_i$. $\mathbf{z}_t^-$ is the predicted observation and $\mathbf{x}_{t-1}^-$ is the predicted state from the previous time-step. The last bracket in (21) can be stored directly for each previous time-step as the difference. Due to the linearity of the hidden continuous state dynamic model, the smoothing is exactly the same as in conventional Kalman smoothing. For details of other steps in UKF, see [9]. In this work, the noise feature is also assumed to be deterministic in equation (19), being characterized only by its mean for simplicity.

Although initial continuous states and covariances are not generally important, it was found that using initial state and covariances estimates from training (equation (15) and (16)), consistently gives better accuracy, especially at low SNR, than estimating by reverse projection from the first frame observation, albeit only for a few decimals of a percentage point.

At each step in both forward- and backward-pass, multiple Gaussians resulting from different component propagation need to be collapsed in order to keep the number of distributions finite. Experimentally, collapsing components within the same class before collapsing with others give the same performance as simultaneously collapsing all components. Only the latter is used for the reported results, nevertheless. Also, the combined likelihood of class S at time-frame t is calculated from

$$L(S) = \sum_m \alpha_m \, L(m) \quad (22)$$

3. EXPERIMENTS AND RESULTS

For comparison with [1], we use the same training set of ten female speakers in DR1 of the TRAIN data in TIMIT and the rest of four female speakers for testing. The phone classes are : 1) vowels, semi-vowels and glides, 2) nasals, 3) fricatives and stop releases, and 4) stop closures and pauses. The features are extracted frame-by-frame using a frame length of 20 ms with 10 ms overlap and Hamming window applied. The dimension of the features is kept at 10 for both LMFB and MFCC while the hidden continuous state dimension is 2. Note that, however, MFCC is derived from 40 Mel-filter bank before getting reduced to 10 via the DCT.

%	AV	GPB2	GPB2(2)	GPB2(4)	GPB2(6)
LMFB	64.5	76.9	78.6	80.4	81.9
MFCC	N/A	81.2	82.7	82.8	82.9

Table 1. Clean speech phone classification accuracy in % of original approximate Viterbi (AV), the single Gaussian GPB2 and GPB2 with mixture model where $M = 2, 4$ and 6 (in parentheses), using LMFB and MFCC as features.

%	Vowels	Nasals	Fric.	Sil.
Vowels	92	4	4	0
Nasals	37	59	3	1
Fric.	14	2	77	6
Sil.	3	5	18	73

Table 2. The confusion matrix in % using GPB2(6) with MFCC as features on clean speech test set

Table 1 shows the classification accuracy results of the clean speech test set where the numbers shown are the percentages of frames correctly classified, compared to frame's time-majority ground truth labels. Note that this measure will favor an algorithm which does well with the first phone class due to more number of frames encountered being vowels. This should be considered as reasonable for real world applications rather than an unfair bias. Both GPB1 and GPB2 outperform the approximate Viterbi algorithm used in [1], which keeps only the maximum likelihood node path in each forward-pass step. GPB2 performs slightly better than GPB1 and the performance increases with the number of mixture components, at the expense of more computation. In fact, approximate Viterbi fails dramatically using MFCC. Note that, however, in [1], approximate Viterbi was used more successfully, as confirmed by our own experiment, with line spectral frequency features, which should be more linear than both LMFB and MFCC, but has no closed-form expression in noise. Nevertheless, using a mixture model is encouraging. For a complete picture, Table 2 also shows the results by class from the best total accuracy achieved from using GPB2-UKF(6), where 6 is the number of mixture components, with MFCC as features.

For noisy speech segmentation, the noise feature means, $N(k)$, are assumed to be stationary and are estimated from some silence frames at the start of the file. Table 3 shows the results using GPB1 and GPB2 with mixture model and UKF noise compensation, averaged over SNR$= 0, 5, \ldots, 20$ dB.

The results in Table 3 show that the proposed scheme can greatly improve overall frame recognition rate for both GPB1 and GPB2. GPB2 outperforms GPB1in all cases and all SNR's (not shown in details due to space limit). While the two-mixture model consistently outperforms the single one at all SNR's, the benefits of having more than two mixture components is less certain, especially at low SNR when it comes to the noisy segmentation. This is probably due to more confusion under approximated compensation and estimation or perhaps under-training due to more

%	white		car		babble	
GPB2	49.3	38.4	49.6	32.4	46.5	37.9
Feat. Sub.+GPB2	57.0	60.8	53.7	55.1	54.3	55.8
Denoise+GPB2	56.0	54.8	57.2	51.6	55.5	55.1
GPB1-UKF	64.6	65.4	65.3	66.9	63.6	66.1
GPB2-UKF	62.8	67.2	66.5	68.4	64.9	67.8
GPB2-UKF(2)	71.1	69.4	67.4	68.5	64.9	64.9
GPB2-UKF(4)	70.5	66.8	68.7	69.1	64.7	62.3
GPB2-UKF(6)	70.4	63.3	67.2	60.4	64.9	57.0

Table 3. Noisy speech phone classification accuracy in % using basic single Gaussian GPB2, noise feature subtraction and basic GPB2, front-end denoising and basic GPB2, GPB1-UKF and GPB2-UKF with mixture model where $M = 2, 4$ and 6, in various types of noise. The first and second column of each noise has LMFB and MFCC as features respectively

%	Vowels	Nasals	Fric.	Sil.
Vowels	89	8	1	2
Nasals	35	54	7	4
Fric.	14	11	45	30
Sil.	3	8	30	59

Table 4. The confusion matrix using GPB-UKF(4) and LMFB as features in a car noise environment, SNR = 10 dB

parameters. For comparison, simply estimating clean speech features solving (19) before using basic GPB2 improves only moderately and so does applying front-end denoising on the corrupted speeches, using the MMSE log-spectral estimator by Ephraim and Malah [10]. The results from relatively non-stationary babble noise still show a comparable improvement. An example of by-class accuracy result is shown in Table 4 for car noise with SNR = 10 dB. As expected, despite the compensation, the fricative class is misclassified as silence while other classes are greatly improved. Especially in the case of white noise, most of the salient feature at high frequency of the class three member is buried in noise, making it difficult to get recognized. It should be mentioned that without noise compensation, most phones are mistaken as the fricative class for white noise, as the vowel class for the car noise and as nasals for babble noise, depending on the characteristics of each noise.

Figure 1 shows an example utterance in car noise and its classification/segmentation results using GPB2-UKF(4) with MFCC as features. Note that fricatives such as /s/ is hardly evident in noise with a rather low sampling rate of 8 kHz, yet can still be detected. Using LMFB instead can only detect the fricatives such as /sh/ and /ch/. Further improvements in the choice of features, the sampling rate increase, other forms of approximate inference and noise tracking, may be pursued as future works.

4. CONCLUSIONS

Algorithms for robust noisy speech segmentation into phoneme classes have been presented. They employ a mixture model of the SSM as the speech and observation model, with the UKF as a means to compensate for nonlinear observation model caused by the noise on the observed features. The results have shown significant improvements over the uncompensated algorithms especially under stationary noises and two mixture components.

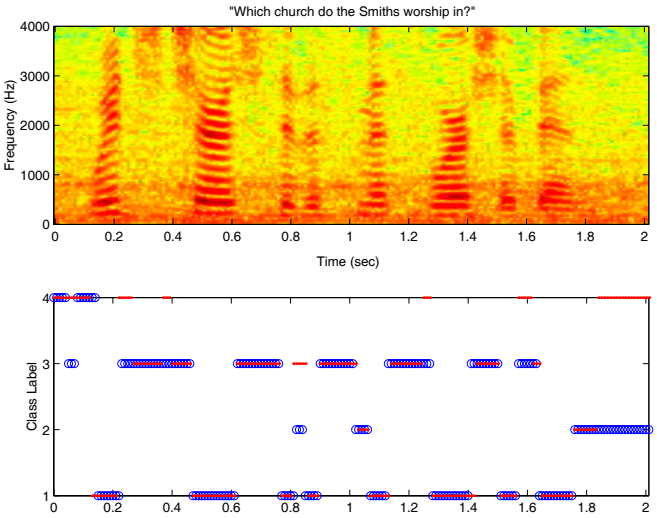

Fig. 1. A spectrogram plot of an utterance in 10 dB car noise (top) and the classification results (blue/circles) using GPB2-UKF(4) along with ground truth labeling (red/dots) (bottom)

5. REFERENCES

[1] Y. Zheng and M. Hasegawa-Johnson, "Acoustic segmentation using switching state Kalman filter," in *ICASSP'2003*, 2003, vol. 1.

[2] J. L. Zhou, F. Seide, and L. Deng, "Coarticulation modeling by embedding a target-directed hidden trajectory model into HMM - model and training," in *ICASSP*, 2003.

[3] J .Z. Ma and L. Deng, "Target-directed mixture dynamic models for spontaneous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 1, pp. 47–58, 2004.

[4] J. Deng, M. Bouchard, and T. Yeap, "Speech enhancement using a switching Kalman filter with perceptual post-filter," in *ICASSP*, 2005.

[5] J. Droppo and A. Acero, "Noise robust speech recognition with a switching linear dynamic model," in *ICASSP*, 2004, pp. 953–956.

[6] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[7] J. Droppo, L. Deng, and A. Acero, "A comparison of three non-linear observation models for noisy speech features," in *Proc. Eurospeech*, September 2003, pp. 681–684.

[8] K. P. Murphy, "Switching Kalman filter," Tech. Rep., Compaq Cambridge Res. Lab, Cambridge, MA, 1998.

[9] E. A. Wan and R. van der Merwe, *Kalman Filtering and Neural Networks*, chapter The Unscented Kalman Filter, Wiley Publishing, 2001.

[10] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. ASSP-33, pp. 443–445, April 1985.