# OVERCOMING THE VECTOR TAYLOR SERIES APPROXIMATION IN SPEECH FEATURE ENHANCEMENT — A PARTICLE FILTER APPROACH

*Friedrich Faubel and Matthias Wölfel*

Institut für theoretische Informatik, Universität Karlsruhe (TH)
Am Fasanengarten 5, 76131 Karlsruhe, Germany
ffaubel@gmail.com, wolfel@ira.uka.de

## ABSTRACT

We present a simple, fast and previously unreported noise compensation method for *particle filter* (PF) based speech feature enhancement, which outperforms the vector Taylor series noise compensation method used by current PF approaches in terms of speed as well as word error rate. Furthermore, we devise a fast acceptance test that overcomes the particle decimation problem associated with PFs for speech feature enhancement, which makes the particle filter approach computationally more efficient.

***Index Terms***— Speech feature enhancement, particle filter, vector Taylor series, statistical inference, automatic speech recognition

## 1. INTRODUCTION

Since its appearance in 1996 Moreno's *Vector Taylor Series* (VTS) approach [1] has kind of become todays "industry standard" for non-stationary noise compensation in speech feature enhancement. Originally, the noise distribution — assumed to be Gaussian — was estimated with a modified *expectation maximization* (EM) algorithm on a segment of corrupted speech, which was later extended to sequential estimation of non-stationary noises. The merit for this is probably due to Kim and his inspiring sequential EM [2] and *interacting multiple model* (IMM) [3] approaches, which have motivated a lot of work in this direction and eventually led to the application of particle filtering to the problem [4].

We show how VTS-based noise compensation, as used in current PF approaches [4, 5, 6], can be derived without a Taylor series approximation. This is achieved by reformulating speech feature enhancement as a tracking problem, which leads to a statistical inference approach where the VTS formula is obtained by introduction of a hidden variable. Not introducing this hidden variable yields a straight-forward noise compensation method, that is not only computationally much less intensive[1] but also outperformed the speech recognition results of VTS noise compensation in experiments.

This paper is organized as follows. In section 2 we briefly restate Raj et al.'s approach [4]. Section 3 derives the novel noise compensation technique as well as the VTS noise compensation method. In section 4 we devise a fast acceptance test that overcomes the particle decimation problem associated with PFs for speech feature enhancement. Sections 5 and 6 present experimental results and our conclusions.

[1] As mentioned in [7] VTS noise compensation in practice dominates the computational cost of the particle filter.

## 2. PARTICLE FILTER BASED SPEECH FEATURE ENHANCEMENT

In [4] the evolution of (log Mel) noise spectra is modeled as a 1st-order autoregressive process

$$n_t = A \cdot n_{t-1} + \varepsilon_t$$

where $A$ is the transition matrix that is learned for a specific type of noise and $n_t$ denotes the noise spectrum at time $t$. The $\varepsilon_t$ terms are considered to be i.i.d. zero mean Gaussian, i.e. $\varepsilon_t \sim \mathcal{N}(0, \Sigma_{noise})$, where the covariance matrix $\Sigma_{noise}$ is assumed to be diagonal. So the noise transition probability $p(n_{t+1}|n_t)$ can be written

$$p(n_{t+1}|n_t) = \mathcal{N}(n_{t+1}; A \cdot n_t, \Sigma_{noise}) \tag{1}$$

Modeling the distribution $p_x$ of clean speech (log Mel) spectra $x_t$ as a mixture of $K$ Gaussians $\mathcal{N}(\mu_k, \Sigma_k)$ with mixture weights $c_k$ and using the relationship[2]

$$x_t = y_t + \log(\underline{1} - e^{n_t - y_t}) \tag{2}$$

between corrupted speech spectra $y_t$, $n_t$ and $x_t$ (all in the log Mel domain), the likelihood $l(n_t^{(j)}; y_t) = p(y_t|n_t^{(j)})$ of a noise hypothesis $n_t^{(j)}$ can be evaluated as

$$p(y_t|n_t^{(j)}) = \frac{p_x(y_t + \log(\underline{1} - e^{n_t^{(j)} - y_t})}{\prod_{i=1}^{d} \left| 1 - e^{\hat{n}_{t,i}^{(j)} - y_{t,i}} \right|} \tag{3}$$

because of the fundamental transformation law of probability. If, however, $n_t^{(j)}$ exceeds $y_t$ in just one spectral bin – say the ith – then

$$n_{t,i}^{(j)} \geq y_{t,i} \Rightarrow e^{n_{t,i}^{(j)}} \geq e^{y_{t,i}} \Rightarrow e^{n_{t,i}^{(j)} - y_{t,i}} \geq 1$$

and $\log(\underline{1} - e^{n_t^{(j)} - y_t})$ can't be calculated. That is the result of considering noise and speech power spectra (not in the log domain) to be strictly additive (see [6] or [8] for more detail). Hence it is impossible[3] that $\|n_{t,i}^{(f)}\|^2 > \|y_{t,i}^{(f)}\|^2$, which is translated to probability by setting $p(y_t|n_t) := 0$. Thus, the particle filter for speech feature enhancement can be outlined as follows:

1. **Sampling** — At time zero ($t = 0$) noise hypotheses (particles) $n_0^{(j)}$ ($j = 1, ..., N$) are drawn from the prior noise density $p(n_0)$. If $t$ is bigger than zero, $n_t^{(j)}$ is sampled from the noise transition probability $p(n_t|\bar{n}_{t-1}^{(j)})$ (equation (1)) for $j = 1, ..., N$.

[2] log and $e$ are applied componentwise and $\underline{1} = (1, \ldots, 1)$

[3] The speech power spectrum $\|x_t^{(f)}\|^2$ is always positive. Superscript $(f)$ denotes the Fourier domain and $x_t$ is $\log(\|x_t^{(f)}\|^2)$.

2. **Calculating the normalized importance weights** — The importance weight (likelihood) of each noise hypothesis $n_t^{(j)}$ is evaluated according to equation (3) if $n_{t,i}^{(j)} < y_{t,i}$ for $i = 1, \ldots, d$. Otherwise $p(y_t|n_t^{(j)})$ is set to zero. The normalized importance weights are calculated as

$$\tilde{\omega}_t^{(j)} = \frac{p(y_t|n_t^{(j)})}{\sum_{m=1}^{N} p(y_t|n_t^{(m)})}$$

3. **Inferring clean speech** — Clean speech is inferred by using the *weighted empirical density*

$$\tilde{p}(n_t|y_{1:t}) = \sum_{j=1}^{N} \tilde{\omega}_t^{(j)} \delta_{n_t^{(j)}}(n_t) \tag{4}$$

— a discrete Monte Carlo representation of the continuous filtering density $p(n_t|y_{1:t})$ (see [8]) — to approximate $\mathrm{E}[x_t|y_{1:t}]$. The details will be derived in section 3.

4. **Importance resampling** — The normalized importance weights are used to resample among the noise hypotheses $n_t^{(j)}$ $(j = 1, ..., N)$. This can be regarded as a pruning step where likely hypotheses are multiplied, unlikely ones are removed from the population.

These Steps are repeated with $t \mapsto (t+1)$ until all time-frames are processed.

## 3. INFERRING CLEAN SPEECH

Speech feature enhancement can be formulated as to track the clean speech spectrum $x_t$ with the observation history $y_{1:t} = \{y_1, \ldots, y_t\}$ using the probabilistic relationship $p(x_t|y_{1:t})$. As stated by Julier and Uhlmann [9] the MMSE solution to such a tracking problem consists in finding the conditional mean $\mathrm{E}[x_{1:t}|y_{1:t}]$. Assuming that $(X_t)_{t\in\mathbb{N}}$ is a Markov process and that the current observation is only dependent on the current state facilitates sequential calculation of the conditional mean (a proof can be found in [8]):

$$\mathrm{E}[x_t|y_{1:t}] = \int x_t \cdot p(x_t|y_{1:t}) dx_t \tag{5}$$

The noise can be introduced as a hidden variable since $p(x_t|y_{1:t})$ can be calculated as marginal density of $p(x_t, n_t|y_{1:t})$:

$$p(x_t|y_{1:t}) = \int p(x_t, n_t|y_{1:t}) dn_t$$

Further, using $p(x_t, n_t|y_{1:t}) = p(x_t|y_{1:t}, n_t) \cdot p(n_t|y_{1:t})$ and changing the order of integration we obtain

$$\mathrm{E}[x_t|y_{1:t}] = \int \underbrace{\int x_t \cdot p(x_t|y_{1:t}, n_t) dx_t}_{=:h_t(n_t)} p(n_t|y_{1:t}) dn_t \tag{6}$$

This is equivalent to calculating $E_{p(n_t|y_{1:t})}[h_t(n_t)|y_{1:t}]$. Hence the weighted empirical density (4) provided by the PF can be used to approximate (6) by Monte Carlo integration (see [10] or [8] on this topic):

$$\mathrm{E}[x_t|y_{1:t}] \approx \sum_{j=1}^{N} h_t(n_t^{(j)}) \tilde{\omega}_t^{(j)} \tag{7}$$

We still need to evaluate $h_t(n_t^{(j)})$, the integral defined in (6). That's where the VTS approach departs from the straight-forward statistical inference approach.

### 3.1. The Straight-Forward Approach

The *straight-forward approach* (SFA) uses the relationship between $x_t$, $n_t$ and $y_t$ from equation (2). This makes the probability density $p(x_t|y_{1:t}, n_t)$ deterministic, since $x_t$ is completely determined if $y_t$ and $n_t$ are given:

$$p(x_t|y_{1:t}, n_t) = \delta_{y_t + log(\underline{1} - e^{n_t - y_t})}(x_t)$$

where $\delta_{(\cdot)}$ denotes a translated Dirac delta function. Substitution of $p(x_t|y_{1:t}, n_t)$ in $h_t(n_t) = \int x_t \cdot p(x_t|y_{1:t}, n_t) dx_t$ yields

$$\begin{aligned} h_t^{(1)}(n_t) &= \int x_t \cdot \delta_{y_t + log(\underline{1} - e^{n_t - y_t})}(x_t) dx_t \\ &= y_t + log(\underline{1} - e^{n_t - y_t}) \end{aligned} \tag{8}$$

This can be regarded as spectral subtraction in the logarithmic domain (for one noise hypothesis).

### 3.2. The Vector Taylor Series Approach

The approach proposed by Raj et al. [4] is to use Moreno's VTS method [1] which approximates $log(\underline{1} + e^{n_t - x_t})$ by its 0th order[4] Taylor series expansion around the kth Gaussian's mean $\mu_k$ of the clean speech distribution. For the case of the PF, where the noise variance is implicitly contained in the different noise hypotheses of the weighted empirical density, the VTS noise compensation scheme can be derived directly, i.e. without VTS approximation. The following statistical derivation shows which assumptions (they are normally hidden in the VTS approximation) have to be made:

First of all, the number of a specific Gaussian of the clean speech distribution can be introduced as a hidden variable $k$, since $p(x_t|y_{1:t}, n_t)$ can be represented as the marginal density

$$p(x_t|y_{1:t}, n_t) = \sum_{k=1}^{K} p(x_t, k|y_{1:t}, n_t)$$

Further, using the equality $p(x_t, k|y_{1:t}, n_t) = p(k|y_{1:t}, n_t) \cdot p(x_t|k, y_{1:t}, n_t)$, $h_t(n_t)$ can be written

$$h_t(n_t) = \sum_{k=1}^{K} p(k|y_{1:t}, n_t) \int x_t \cdot p(x_t|k, y_{1:t}, n_t) dx_t \tag{9}$$

where the sum over $k$ was pulled out of the integral. Now, the noise can be considered to shift the means of the clean speech distribution $p_x$ in the spectral domain. The effect of $n_t$ to the $kth$ Gaussian in the log spectral domain is

$$e^{\mu'_k} = e^{\mu_k} + e^{n_t}$$

Solving for $\mu'_k$ yields

$$\mu'_k = \mu_k + \underbrace{log(1 + e^{n_t - \mu_k})}_{=:\Delta_{\mu_k, n_t}} \tag{10}$$

Instead of shifting the mean, we can conversely shift the corrupted spectrum $y_t$ in the opposite direction[5] to obtain the clean speech spectrum

$$x_t = y_t - \Delta_{\mu_k, n_t} \tag{11}$$

[4] Higher order approximations were also examined in [1].

[5] Note, that this does not change the probability, i.e. $\mathcal{N}(y_t; \mu'_k, \Sigma_k) = \mathcal{N}(x_t; \mu_k, \Sigma_k)$.

This deterministic relationship yields $p(x_t|k, y_{1:t}, n_t) = \delta_{y_t - \Delta_{\mu_k, n_t}}(x_t)$ and hence

$$\begin{aligned} h_t^{(2)}(n_t) &= \sum_{k=1}^{K} p(k|y_{1:t}, n_t) \int x \cdot \delta_{y_t - \Delta_{\mu_k, n_t}}(x_t) dx_t \\ &= \sum_{k=1}^{K} p(k|y_{1:t}, n_t) \left(y_t - \Delta_{\mu_k, n_t}\right) \\ &= y_t - \sum_{k=1}^{K} p(k|y_{1:t}, n_t) \Delta_{\mu_k, n_t} \end{aligned} \quad (12)$$

Comparing this equation with equation (19) of [4] shows that the VTS approach approximates $p(k|y_{1:t}, n_t)$ by $p(k|y_t, n_t)$.This is equivalent to assuming that $k$ — the number of the Gaussian in the Gaussian mixture $p_x$ that "produced" the speech spectrum $x_t$ — is independent of the preceding corrupted speech spectra, if $y_t$ and $n_t$ are known. Another assumption implicitly made by Raj et al. is that $k$ is independent of the current noise spectrum, i.e. $p(k|n_t) = p(k) = c_k$, which is perfectly true for additive noise, not however for noise spectra that contain reverberations of speech as in far distance data. Assuming that, $p(k|y_{1:t}, n_t)$ can be calculated as

$$\begin{aligned} p(k|y_t, n_t) &= \frac{p(y_t|n_t, k) \cdot p(k|n_t)}{p(y_t|n_t)} \\ &= \frac{c_k \cdot p(y_t|n_t, k)}{\sum_{k=1}^{K} c_k \cdot p(y_t|n_t, k)} \end{aligned}$$

## 4. GETTING THE FILTER TO WORK

In practice there are some issues that prevent the particle filter from working well. The major issue is that noise hypotheses are not allowed to exceed the observed, contaminated spectrum. Assigning a zero weight in this case comes with the side-effect that overestimations of the actual noise as well as cancellation due to relative phase differences between noise and speech lead to a decimation of the particle population up to its complete annihilation if all weights are zero. This problem has previously been described by Haeb-Umbach and Schmalenstroeer [6]. In [7] we described, how dropouts — the case where all particles have zero likelihood — can be handled by reinitializing the particle filter with the noise distribution. Here we show how to tackle this problem by reducing the number of dropouts through a fast acceptance test.

### 4.1. A Fast Acceptance Test

The best solution to handling sample attrition and dropouts is of course be not to let them happen. In fact, the number of dropouts can be reduced by increasing the number of samples ($N$), which however greatly increases the computational time. We propose to use a *fast acceptance test* (FAT) that virtually boosts the number of particles up to $(N \cdot B)$ if necessary. The acceptance test works as follows: when drawing noise hypotheses for time $t$ in the sampling step of the particle filter, the drawn sample $n_t^{(j)}$ is rejected if $n_{t,i}^{(j)} < y_{t,i}\ \forall i$ is not satisfied. In case of rejection another particle $n_{t-1}^{(s)}$ is selected by drawing $s \in \{1, \ldots, N\}$ and $n_t^{(j)}$ is sampled from $p(n_t|n_{t-1}^{(s)})$. This is repeated until the noise hypothesis is accepted or a certain number $B$ of iterations has passed.

```
for j = 1 to N do
    l = 0
    s = j
    accept = false
    while (l < B) and (accept == false)
        sample n_t^(H) from p(n_t | n̄_{t-1}^(s))
        if (n_{t,i}^(H) < y_{t,i} ∀i)
            accept = true
        else
            randomly select s ∈ {1, ..., N}
            l = l + 1
    end while
    n_t^(j) = n_t^(H)
end for
```

Algorithm 4.1: Fast Acceptance Test

The advantage of this approach is that the number of samples stays constant while particle shortages due to the decimation problem can be overcome. The worst case computational time is limited by $B$, but in practice it is much lower then using $(N \cdot B)$ particles. Randomly selecting another particle to sample from in case of rejection makes sure that the predictive density $p(n_t|n_{t-1})$ is not changed. Dropouts still occur, however less often.

## 5. EXPERIMENTS

In order to evaluate the performance of the proposed PF enhancements under realistic conditions we chose approximately 45 minutes of lecture speech. As a speech recognition engine we used the *Janus Recognition Toolkit* (JRTk) with the same setup as described in [7]. Furthermore, we used warped *minimum variance distortionless response* (MVDR) cepstral coefficients [11], since they have been shown to outperform Mel frequency cepstral coefficients [7]. The acoustic training material (approximately 100 hours) used for the experiments reported here was taken from the ICSI, NIST, and CMU meeting corpora, as well as the *Translanguage English Database* (TED) corpus resulting in 3,500 context dependent codebooks with up to 64 Gaussians with diagonal covariances each. The 3-gram language model contained approximately 23,000 words with a perplexity of 125.

In a first experiment we artificially added highly dynamic noise with a broad variety of sounds coming from a truck, slamming rubbish containers, distant voices, and shouts [12] to evaluate the fast acceptance test's capability to reduce particle decimation and dropouts. We have selected a *signal to noise ratio* (SNR) of 0dB, since the decimation problem is typically more severe for lower SNRs. Clean speech inference was performed with the SFA. Table 1 shows that the dropout rate is significantly reduced by the fast acceptance test. While it decreases with the number of particles without FAT, it seems to vary around $1.4\%$ if FAT is used. The *mean square error reduction* (MSER) per frame increases with the number of particles in both cases, but it is much higher with the fast acceptance test — more than $6.5\%$ absolute. The *word error rate* (WER) is shown for an unadapted and an adapted pass, which uses *maximum likelihood linear regression* (MLLR). While there seems to be some gain on the unadapted pass, this is much more unclear for the adapted pass. In the following experiments we used 100 particles and the fast acceptance test.

Table 2 gives a comparison with VTS (with FAT). While there is practically no difference in MSER (compare to table 1), the WERs differ greatly for the two approaches. An interesting result is that the improvement in WER of the unadapted pass only marginally

| #particles | FAT | % dropout | MSER | WER unadp. | WER adapted |
|---|---|---|---|---|---|
| 100 | no | 6.17 | 12.52 | 55.1 | 39.6 |
| 200 | no | 5.69 | 13.99 | 55.0 | 39.3 |
| 400 | no | 4.62 | 15.66 | 55.5 | 40.8 |
| 800 | no | 4.40 | 16.97 | 54.2 | 39.6 |
| 100 | yes | 1.48 | 19.57 | 53.3 | 39.9 |
| 200 | yes | 1.54 | 20.85 | 53.9 | 38.9 |
| 400 | yes | 1.21 | 22.37 | 54.6 | 40.0 |
| 800 | yes | 1.50 | 23.52 | 54.1 | 39.8 |

**Table 1**. *Mean square error reduction* (MSER) and *word error rates* (WER)s for different numbers of particles with SFA noise compensation with and without the *fast acceptance test* (FAT).

diminishes on the adapted pass. The baseline (without particle filter) is 60.2% for the unadapted and 42.4% for the adapted pass, respectively. On the more significiant, adapted pass, the WER for VTS noise compensation is not significantly better than the baseline, while the WER for the SFA is more than 2% (5% relative) lower.

| #particles | SFE | MSER | WER unadp. | WER adapted |
|---|---|---|---|---|
| 100 | VTS | 19.54 | 55.3 | 41.3 |
| 200 | VTS | 20.87 | 55.5 | 41.3 |
| 400 | VTS | 22.30 | 55.8 | 42.1 |
| 800 | VTS | 23.47 | 56.0 | 42.6 |

**Table 2**. *Mean square error reduction* (MSER) and *word error rates* (WER)s for *vector Taylor series* (VTS) noise compensation with fast acceptance test

We performed some more experiments (Table 3) with different SNRs for the highly dynamic noise of the previous experiments (noise type 1) as well as for a much less dynamic (almost stationary) noise — the humming of a hydroelectric power plant [13] (noise type 2). The SFA approach is constantly better than VTS noise compensation. Comparison with the speech recognition results without *speech feature enhancement* (SFE) shows that those gains are not insignificant. The WER is much lower for the stationary noise than for the non-

| | | WER | | | |
|---|---|---|---|---|---|
| | | Noise Type 1 | | Noise Type 2 | |
| SNR | SFE | unadp. | adapted | unadp. | adapted |
| clean | none | 31.0 | 25.4 | 31.0 | 25.4 |
| 10dB | none | 39.4 | 29.2 | 33.9 | 26.6 |
| 5dB | none | 48.1 | 33.8 | 34.8 | 28.3 |
| 0dB | none | 60.2 | 42.4 | 40.5 | 32.9 |
| 10dB | VTS | 36.9 | 28.6 | 32.6 | 26.9 |
| 5dB | VTS | 43.7 | 32.6 | 34.7 | 28.9 |
| 0dB | VTS | 55.3 | 41.3 | 40.0 | 31.7 |
| 10dB | SFA | 36.3 | 28.1 | 31.2 | 26.2 |
| 5dB | SFA | 43.1 | 31.9 | 34.2 | 27.5 |
| 0dB | SFA | 53.3 | 39.9 | 39.9 | 31.5 |

**Table 3**. *Word error rates* (WER)s for 100 particles, fast acceptance test, dynamic and static noise types

stationary one. This might be due to *cepstral mean normalization* (CMN) and *cepstral variance normalization* (CVN), or because the stationary noise's mechanical character does not have such a serious impact on our features: *linear discriminant analysis* (LDA) coefficients of 15 successive mel frequency cepstral coefficients (7 to the left, 7 to the right).

## 6. CONCLUSIONS

We feel it is worth to replace VTS with SFA noise compensation when it comes to particle filter approaches. It is faster and easier to implement, while performing somewhat better than the VTS approach with respect to WER. The fast acceptance test reduces the particle decimation problem and performs greatly better in terms of MSER. Furthermore, it reduces the WER of the unadapted pass, which could be of interest especially for ASR systems without MLLR adaptation.

## 7. REFERENCES


[1] P.J. Moreno, B. Raj, and R.M. Stern, "A vector taylor series approach for environment-independent speech recognition," *Proc. of ICASSP*, 1996.

[2] N. S. Kim, "Nonstationary environment compensation based on sequential estimation," *IEEE Signal Processing Letters*, vol. Vol. 5, No. 3, pp. 57–59, Mar. 1998.

[3] N. S. Kim, "Imm-based estimation for slowly evolving environments," *IEEE Signal Processing Letters*, vol. Vol. 5, No. 6, pp. 146–149, Jun. 1998.

[4] B. Raj, R. Singh, and R. Stern, "On tracking noise with linear dynamical system models," *Proc. of ICASSP*, 2004.

[5] M. Fujimoto and S. Nakamura, "Particle filter based non-stationary noise tracking for robust speech feature enhancement," *Proc. of ICASSP*, 2005.

[6] R. Haeb-Umbach and J Schmalenstroeer, "A comparison of particle filtering variants for speech feature enhancement," *Proc. of Interspeech*, 2005.

[7] F. Faubel and M. Wölfel, "Coupling particle filters with automatic speech recognition for speech feature enhancement," *Proc. of Interspeech*, Sep. 2006.

[8] F. Faubel and M. Wölfel, *Speech Feature Enhancement for Speech Recognition by Sequential Monte Carlo Methods*, Diploma Thesis. Universität Karlsruhe (TH), Germany, Aug. 2006.

[9] S. Julier and Uhlmann J.K., "A general method for approximating nonlinear transformations of probability distributions," Nov. 1996.

[10] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer Texts in Statistics. Springer, second edition, 2004.

[11] M. Wölfel and J.W. McDonough, "Minimum variance distortionless response spectral estimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.

[12] The Freesound Project, "garbage.coll.serv.ds70p.mp3," *freesound.iua.upf.edu/samplesViewSingle.php?id=6986*.

[13] The Freesound Project, "[hydroelectric powerstation] centrale de st-marc - st-martin-terressus (france).mp3," *freesound.iua.upf.edu/samplesViewSingle.php?id=6986*.