Feature Enhancement with a Reservoir-based Denoising Auto Encoder

Azarakhsh Jalalvand, Kris Demuynck, Jean-Pierre Martens
Multimedia Lab, ELIS, Ghent University/iMinds, Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium
Email: Azarakhsh.Jalalvand@ugent.be

***Abstract*—Recently, automatic speech recognition has advanced significantly by the introduction of deep neural networks for acoustic modeling. However, there is no clear evidence yet that this does not come at the price of less generalization to conditions that were not present during training. On the other hand, acoustic modeling with Reservoir Computing (RC) did not offer improved clean speech recognition but it leads to good robustness against noise and channel distortions. In this paper, the aim is to establish whether adding feature denoising in the front-end can further improve the robustness of an RC-based recognizer, and if so, whether one can devise an RC-based Denoising Auto Encoder that outperforms a traditional denoiser like the ETSI Advanced Front-End. In order to answer these questions, experiments are conducted on the Aurora-2 benchmark.**

Keywords—recurrent neural networks, reservoir computing, denoising auto encoder, robust speech recognition

I. Introduction

If one wants to apply automatic speech recognition (ASR) on mobile devices, an ASR system is needed that is accurate and robust against the presence of noise and channel distortions. Despite many years of work, achieving robust recognition remains a big challenge.

Since an ASR is composed of a front-end (for the extracting acoustic feature vectors) and a back-end (for decoding these feature vectors) one can envision two types of techniques to tackle the problem. One is to enhance (denoise) the feature vectors in the front-end [1], [2], the other is to take account of the noise during feature decoding in the back-end [3], [4]. On top of that, one can develop acoustic model types that are intrinsicaally more resistant to noise [5]–[8].

Typical feature enhancement adopts signal processing techniques such as Wiener filtering (single-channel) and beamforming (multi-channel). They gave rise to an Advanced Front-end (AFE) [1] and the SPLICE [9] algorithms, respectively. Less typical is to adopt a neural network to convert the noisy feature vectors to clean vectors [10]. Such a network is called a Denoising Auto Encoder (DAE). Back-end techniques are usually restricted to GMM-based acoustic modeling (GMM = Gaussian Mixture Model) because for this type of model it is 'easy' to understand how to include the variations due to the noise into the decoder.

In previous work we investigated Reservoir Computing (RC) [11], [12] as a neural-based approach to acoustic modeling. The argument was that RC-networks are dynamical systems which are able to focus on meaningful speech dynamics which are bound to differ from the dynamics induced by the noise. We achieved quite competitive results for noise-robust continuous digit recognition (CDR) on Aurora-2 [7], [13]. A deep RC-HMM hybrid (HMM = Hidden Markov Model) can apparently compete with a traditional GMM-HMM system in clean conditions and clearly outperform it in noisy environments. More recently, we also discovered that supplying AFE features to the RC-HMM system can further increases its noise robustness [8].

In this work we further investigate the impact of the front-end in an RC-HMM system. In particular, we propose to apply an RC-network for feature denoising before recognition. We argue that such complex nonlinear dynamical system is bound to capture better the complex relationships between noisy and clean utterances than the traditional signal processing methods that have only a very short memory.

The rest of the paper is organized as follows: Section II provides a concise outline of the basic principles of RC, Section III describes ways of integrating reservoir networks in an RC-HMM hybrid speech recognizer, Section IV reviews the RC-based denoising of the acoustic features we propose and Sections V and VI summarize the experimental framework (Aurora-2) and the results obtained within this framework. The paper ends with conclusions and future work.

II. Reservoir Computing (RC)

The basic principle of RC is that one can retrieve information from sequential inputs by means of a two-layer RNN with the following characteristics (see Fig. 1). The inputs are

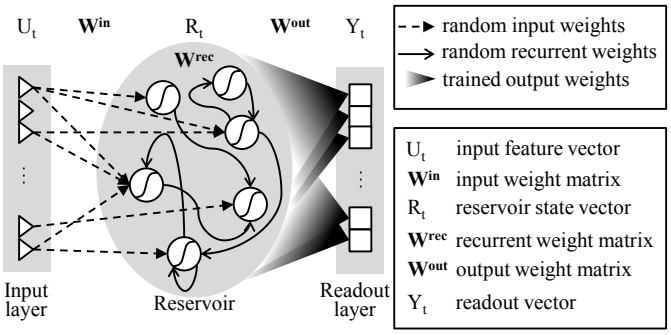


Fig. 1. A basic RC system consists of a reservoir and a readout layer. The reservoir is composed of interconnected **non-linear** neurons with randomly **fixed** weights. The readout layer consists of **linear** neurons with **trained** weights.

sparsely connected to a pool of recurrently interconnected non-linear neurons – forming so-called reservoir – and the outputs

are obtained as a linear combinations of the reservoir outputs. The reservoir neurons constitute a hidden layer that, at time t, is driven by inputs U_t and delayed hidden layer outputs R_{t-1}. Important is that (1) the weights of the hidden neurons are fixed, (2) the output neurons are linear, and consequently, (3) the output weights can be optimized by means of a least squared linear regression. The entire system is called a *reservoir network*. Its outputs Y_t are usually called *readouts* [11] so as to differentiate them unambiguously from the reservoir outputs R_t. In order to become more resistant to random inter-frame changes in the inputs (e.g. changes due to the spectral analysis or the ambient noise), one can create a reservoir of Leaky Integrator Neurons [14]). A network containing such a reservoir is governed by the following equations:

$$R_t = (1-\lambda)R_{t-1} + \lambda\, f_{res}(\mathbf{W}^{in}U_t + \mathbf{W}^{rec}R_{t-1}) \quad (1)$$

$$Y_t = \mathbf{W}^{out}R_t \quad (2)$$

with a leak rate λ between 0 and 1, with $\mathbf{W}^{in}$, $\mathbf{W}^{rec}$ containing the input and recurrent weights to the reservoir neurons, and with $\mathbf{W}^{out}$ containing the output weights.

The weights of the hidden neurons are fixed by means of a random process characterized by four control parameters (see [15] for more details): (1) α_U, the maximal absolute eigenvalue of the input weight matrix $\mathbf{W}^{in}$, (2) ρ, the maximal absolute eigenvalue of the recurrent weight matrix $\mathbf{W}^{rec}$, (3) K^{in}, the number of inputs driving each reservoir neuron and (4) K^{rec}, the number of delayed reservoir outputs driving each reservoir neuron. The first two parameters control the strengths of the input and the recurrent stimulations of a reservoir neuron, whereas the latter two control the sparsity of the input and recurrent weight matrices. Together with λ they constitute the reservoir control parameters which have to be set properly in order to assure the reservoir is well behaved. Any effective reservoir should at least have the so-called echo state property, stating that with time, the reservoir should forget about the initial state it was in. That is also why a reservoir network was originally called an Echo State Network [11]. It was shown in [11] that the echo state property holds if ρ – called the spectral radius – is smaller than 1.

The reservoir can be envisioned as a predefined but complex non-linear dynamical system that performs a temporal analysis of the input stream. Our claim is that such a system can extract features which are resistant to the presence of noise whose dynamics differ from the speech dynamics.

The output weights are determined so that they minimize the mean squared error between the readouts Y_t and the desired readouts D_t over the training examples [7]. As a consequence, they follow from a set of linear equations. If a reservoir network is trained for recognition, the desired output D_t is a unit vector with one non-zero entry encoding the desired HMM-state at time t. If it is trained for denoising, D_t is the desired clean speech feature vector at time t.

III. A HYBRID RC-HMM FOR SPEECH RECOGNITION

An RC-HMM hybrid works with an HMM that represents the task and a neural network that is supposed to convert the inputs U_t into HMM state likelihoods. The search for the best path through the HMM is found using a Viterbi search. In the case of an RC-HMM hybrid, the readouts $y_{t,i}$ (with i indexing the readouts) are assumed to resemble the posterior probabilities $P(q_t = i|U_1^t)$. This means that $z_{t,i} = y_{t,i}/P(q_t = i)$ is a scaled likelihood and consequently, that the best state sequence follows from

$$\hat{q} = \arg\max_q P(q, y) = \arg\max_q \prod_{t=1}^{T} z_{t,q_t}\, P(q_t|q_{t-1}),$$

Fig. 2 shows the architecture for the case of continuous digit recognition (CDR) and a multi-stage RC network in which each network output is supplied to the next stage [8]. The transition probability P_0 which is added to the digit loop controls the balance between deletions and insertions.

Since $y_{t,i}$ is not confined to [0,1] it is first mapped to that interval before computing $z_{t,i}$. The mapping is achieved by a simple clip-and-scale approach, as described in [8]. The different stages of the RC-network are trained independently, one after the other.

As in [8], [16] we use a bi-directional RC-network with one reservoir processing the frames from left-to-right and another one processing them from right-to-left. The readouts at time t are then computed as a linear function of the two reservoir outputs at time t.

IV. AN RC-BASED DENOISING AUTOENCODER

A system that aims at reconstructing clean feature vectors from noisy feature vectors is called a denoising autoencoder (DAE) [17]. A traditional example of a DAE is spectral subtraction: it is able to remove part of the noise by working frame-by-frame (= memory-less). One can argue that a complex dynamical system with memory should be able to do a better job. Therefore, we propose to create a DEA by means of an RC-network incorporating one unidirectional reservoir because such a system has memory and it permits to perform a non-linear transformation of the noisy features to 'clean' features. The back-side of this approach is of course that the DAE has to be trained and that during this training one needs the clean version of the noisy speech feature vectors.

Since we use MFCCs (Mels-scale Frequency Cepstral Coefiificients) as the feature vectors, and as the DAE has to denoise these MFCCs, one can adopt different strategies for achieving this. In fact, the MFCC-computation includes a chain of processing stages (See Fig. 3), and the denoising RC-network could be inserted at different positions in this chain. Traditional speech enhancement for instance often works in the Discrete-time Fourier Transform (DFT) domain [18], [19] (at position P1), whereas most work on robust ASR applies denoising in the log mel-frequency domain [20] (at position P2) and in the MFCC-domain [21], [22] (at position P3). Note that although the high dimensionality of the input normally leads to a higher computational cost [23], this may not necessarily be the case here due to the sparse interconnection between inputs and reservoir neurons, and also due to employing linear nodes as the readouts. Therefore, we consider a RC-network at all positions.

Furthermore, each RC-network is supposed to have access to static as well as the dynamic features, but the network is only expected to produce denoised static features. The dynamic parameters of the input to the ASR are then derived from these denoised static features.

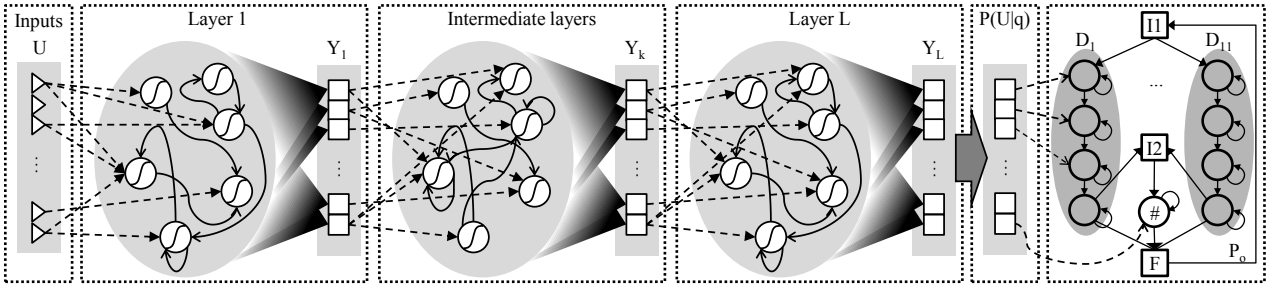


Fig. 2. Architecture of an RC-HMM hybrid comprising a multi-layer reservoir network for CDR. The HMM has two initial states (I1 and I2), one final state (F) and it comprises 11 multi-state digit models (D1 ... D11) and a single state silence model (#)

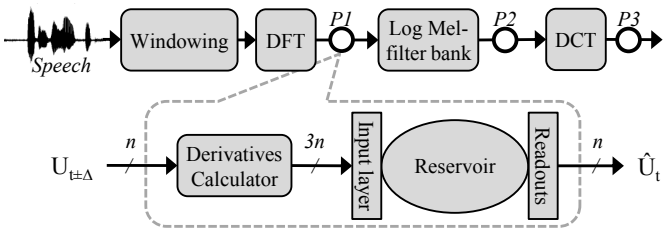


Fig. 3. Possible points to apply the denoising system (top) and their structure in more details (down)

V. Experimental setup

In this section we present the experimental framework that was adopted to investigate the potential of the different system configurations presented in the previous sections.

A. Speech corpus: Aurora-2

The Aurora-2 corpus consists of clean and noise corrupted digit sequences counting 1 to 7 digits per utterance. Each utterance is passed through a G712 or a MIRS filter [24], and then sampled at 8 kHz. Since there are two variants of '0' in American English, namely *zero* and *oh*, the vocabulary is composed of 11 digits.

The data is divided into a training part and an evaluation part. The framework supports two types of experiments: clean training experiments in which systems are developed on 8440 clean training utterances from 110 adults and multi-style training experiments in which systems are developed on 8440 noise corrupted versions of the same utterances. The corruption is randomly chosen out of four noise types and five SNRs (∞ (clean), 20, 15, 10 and 5 dB). The evaluation utterances come from speakers that are not present in the training data. They are divided into three tests. Tests A and B each contain 28,028 utterances covering 4004 different digit sequences, 4 noise types and 7 SNRs (∞ (clean), 20, 15, 10, 5, 0, and -5 dB). The noise types occurring in Test B do not occur in the multi-style training data, while those of Test A do. Test C contains 14,014 utterances covering 2002 different digit sequences, 2 noise types (one matched and one mismatched) and 7 SNRs. Unlike all other utterances they passed through a MIRS instead of a G712 filter (see Table I).

TABLE I. Noise types and filters used in Aurora-2 dataset

	Train & Test A	Test B	Test C
Noise types	N1: subway N2: babble N3: car noise N4: exhibition hall	N1: restaurant N2: street N3: airport N4: train station	N1: subway N2: street
Filter	G712	G712	MIRS

B. Evaluation results

We report average Word Error Rates (WERs) on tests A-C for all SNRs, and we consider both clean speech training and multi-style training. In multi-style training, clean utterances and utterances with SNRs between 20 and 5 dB are used for training.

In the final evaluation phase both the acoustic models and the DAE are trained on the complete training set, but using the control parameters that were found optimal in a development phase during which two thirds of the training set are used for training and the remaining third for control parameter optimization (e.g. the Viterbi decoder penalty, P_0 of the recognizer). In this paper, we only report the results of the final evaluation phase for each experiment.

C. Reference systems

In order to set a reference, we first report some state-of-the-art system performances (see Table II). In particular, we consider the ML-based GMM systems using AFE-features proposed in [25], the ML-based and MCE-based GMM systems proposed in [2] and [26], two GMM systems embedding more sophisticated back-ends based on joint uncertainty decoding (JUD) and Vector Tylor Series (VTS) respectively [4] and the tandem system embedding deep belief networks and GMMs (T-DBN-GMM), reported in [5]. The figures show that advanced back-end techniques (JUD and VTS) lead to a larger gain in noise robustness than advanced front-end techniques, but it is not clear from the papers how much degradation they induce for clean speech recognition.

D. Front-end setups

We will investigate three different acoustic feature sets: MFCCs (log energy and $c_1 \dots c_{12}$), Mel filterbank features (MelFB) (24 channel log energies), and the AFE features (denoised $c_0 \dots c_{12}$ without dropping non-speech frames) [1]. In all cases, the analysis is performed on 30 ms Hamming-windowed frames and the hop size between frames is $\tau_{fr} =$

TABLE II. COMPARING AVERAGE WERS (IN %) PER CONDITION FOR TEST SETS A-C OF AURORA-2 USING A 3-LAYER HYBRID RC-HMM FOR BOTH CLEAN AND MULTI-STYLE TRAINING.

System	Clean	Clean 0-20	Clean -5dB	Multi Clean	Multi 0-20	Multi -5dB
GMM (AFE) [25]	0.77	13.2	69.9	0.83	8.4	59.2
GMM (MFCC) [2]	0.84	19.7	82.2	1.77	8.5	59.1
GMM (SPLICE) [27]	0.55	17.6	83.7	-	12.7	-
GMM (MFCC-MCE) [26]	0.41	15.7	77.2	0.92	6.4	55.3
GMM (VTS) [4]	-	9.4	-	-	-	-
GMM (JUD) [4]	-	10.3	-	-	-	-
T-DBN-GMM [5]	1.26	21.0	74.6	-	-	-
RC (MelFB)	0.74	11.0	63.8	1.06	**5.4**	45.0
RC (MFCC)	0.93	10.7	59.7	1.46	6.2	46.5
RC (AFE)	0.84	**8.9**	**54.4**	1.40	5.7	**43.2**

10 ms. Each feature set is supplemented with Δ and $\Delta\Delta$ features.

Before supplying the feature vectors to the ASR, an utterance-wise normalization that creates zero-mean and unit-variance inputs per feature is performed.

E. RC-HMM hybrid setup

Following our previous work, the reservoir control parameters of each reservoir are determined in the same way. Defining $\tau_\lambda \doteq -\tau_{fr}/\ln(1-\lambda)$ as the leaky integration time constant and T as the expected state duration, we select $(\rho, \tau_\lambda, K^{in}, K^{rec}) = (0.8, T, 10, 10)$ and we chose α_U so that the average variance of the reserevoir outputs reaches a certain level [15]. In this work we utilize a 3-layer RC-HMM system and since layers 2 and 3 see basically the same inputs, α_U is taken the same for both layers.

The size of the reservoir – defined as the number of neurons it contains (N^{res}) – is considered to be an independent variable. Note that since K^{in} and K^{rec} are kept fixed to 10, the CPU-time needed for calculating the readouts scales linearly with the size of the reservoir.

The reservoir networks are trained by means of a Tikhonov regression [28] and each digit is modeled by a 7-state left-to-right HMM whilst the silence is modeled by a single state. The target outputs encode the visited HMM state at time t.

F. RC-based DAE setup

In line with [15], we contemplate that ρ and λ are the only task-dependent control parameters of a reservoir. Consequently, even though denoising is a completely different task than digit state recognition, we keep $Kin = K^{rec} = 10$ and we maintain the same method as before for computing α_U. Furthermore, since leaky integration is a form of smoothing corresponding to a some low-pass filter, we argue that no leaky integration should be applied here as the spectrum of the denoised inputs is not expected to be narrower than that of the noisy inputs. Consequently, there is only one parameter left to optimiza, namely ρ.

In all experiments, irrespective of the chosen feature set, we have used a 2-layer RC-network with one unidirectional reservoir of 1K neurons per layer.

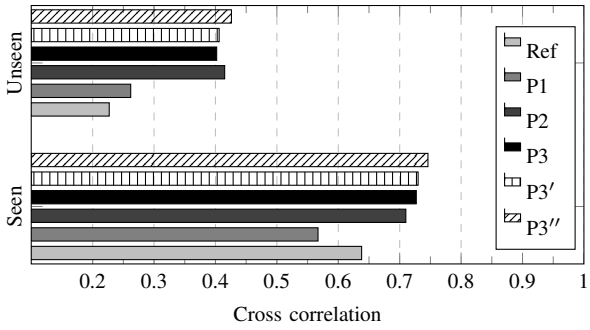


Fig. 4. The correlation between the output of the DAE and the clean normalized features, by denoising the data stream in different points.

VI. EXPERIMENTAL RESULTS

In this section we review the experiments we conducted to assess the capacity of an RC-network to denoise the features and the capacity it has to further raise the robustness of the CDR system.

A. Denoising capacity of an RC-network

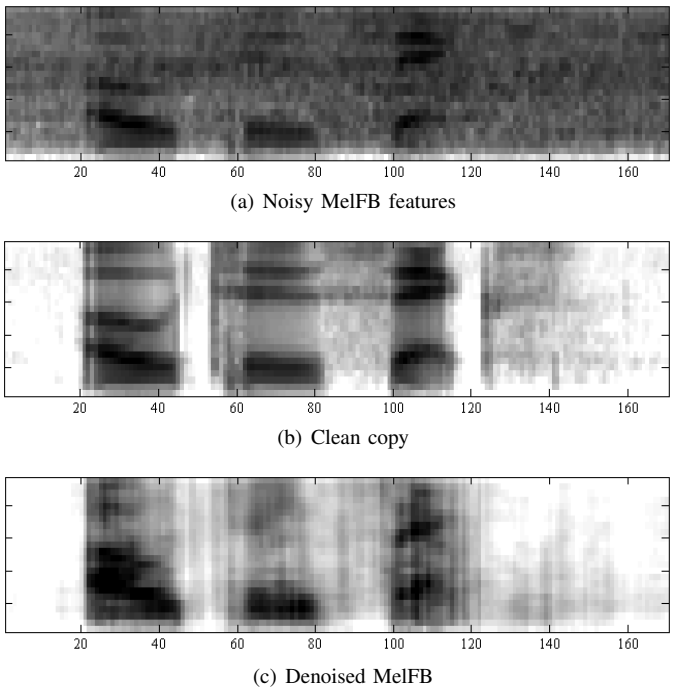
During assessment of the denoising capacity of the RC-networks we adopt the average correlations between the 39 denoised and the clean MFCCs features as the quality criterion. However, since the ASR will always work with utterance based mean and variance normalized features, the correlations are computed after this normalization step.

We conducted two experiments: one in which the test samples come from Test A and represent conditions that are present during training (= "Seen") and one in which the test samples come from Test C and represent another channel and unseen SNRs (= Unseen"). In each experiment, the DAE is imputed at all three positions P1, P2 and P3 (see Section IV). The results are depicted in Fig. 4. "Ref" denotes to the correlation existing between the raw (noisy) and the clean features. The data show that denoising in the DFT-domain is not working well but denoising in the two other domains does. In fact, irrespective of the experiment, denoising the MelFB and the MFCC features seem to be equally effective. We will therefore evaluate them both in combination with CDR.

In an additional experiment we constructed RC-networks that were trained to denoise the dynamic as well as the static MFCCs instead of just the static features. This approach lead to the average correlation marked by P3'. There seems to be no benefit in working this way.

Finally, we wanted to assess the limits of the RC-based DAE by raising the reservoir size from 1K to 4K neurons. The results obtained with the larger reservoirs (marked as P3") are only slightly better than the ones obtained with the smaller reservoirs.

In order to illustrate the effect of the DAE, we have depicted on Fig. 5 the MelFB spectrograms for a noisy speech sample (SNR = 5dB) before and after denoising together with the clean speech spectrogram. It is especially noteworthy that the DAE does an excellent job in the silence parts. This is partly due to the large number of non-speech (silent) frames in the training data.



(a) Noisy MelFB features

(b) Clean copy

(c) Denoised MelFB

Fig. 5. Denoising MelFB features of a sample with street noise of SNR 5 dB from Test C.

B. Need for a denoising front-end in an RC-based CDR

In a first recognition experiment we test the three acoustic feature sets described in the previous section in combination with a three-layer bi-directional RC-HMM hybrid with 8K neurons per layer. The results of this experiment are listed in the bottom section of Table II.

Apparently, in the clean speech training experiment there is only a little difference between the feature sets in the matched condition (clean). In the mismatched conditions however, the AFE does lead to more robustness, be it that the gain is very much less impressive than it is for GMM-based systems. This finding seems to confirm the hypothesis that a reservoir can filter out a large part of the noise without having been confronted with noise during training.

In the multi-style training experiment, where the mismatch between the test and the training remains much smaller, the positive effect of the AFE is also much smaller, but nevertheless remaining to some extent. Consequently, if an RC-based DAE could outperform the AFE it could lead to increased the robustness.

C. Can RC-based DAE outperform the AFE?

In order to answer the question, we conducted experiments with a front-end incorporating an RC-based DAE at position P3. We considered three configurations: (1) the baseline features and acoustic models are used (= Baseline), (2) the denoised features are used but the acoustic models are left unchanged (= Test on Baseline) and (3) the denoised features are used in combination with acoustic models that are retrained on these features (= Retrain & Test). In the case of retraining, we maintained the distinction between clean speech training and multi-style training, but of course, the clean speech training results have to be interpreted with care because after all, the DAE has already seen the noisy training examples that belong to the multi-style training corpus.

TABLE III. COMPARING AVERAGE WERS (IN %) ON TEST SETS A-C PER CONFIGURATION: (1) THE BASELINE RC-HMM RECOGNIZER TRAINED ON MFCC AND AFE, (2) THE BASELINE RECOGNIZER TRAINED ON MFCC AND AFE BUT TESTED ON DENOISED FEATURES, AND (3) THE RETRAINED RECOGNIZER TESTED ON DENOISED FEATURES.

		Clean	Clean 0-20	Clean -5dB	Multi	Multi 0-20	Multi -5dB
Baseline	MFCC	0.93	10.7	59.8	1.46	6.2	46.5
	AFE	0.84	8.9	54.4	1.40	5.7	43.3
Test on Baseline	MFCC	2.08	8.9	53.4	3.35	8.0	50.0
	AFE	2.56	9.4	49.7	3.57	8.1	45.9
Rerain & Test	MFCC	1.36	8.6	54.5	1.77	6.6	48.4
	AFE	1.38	7.7	50.0	1.85	6.2	43.6

The results listed in Table III show that the RC-based denoiser does not outperform the AFE as the performance of the baseline system with AFE features is slightly better than that of a system working with plain MFCCs and an RC-based DAE on all conditions.

In spite of this, comparing the AFE rows of the baseline and the new system, demonstrates that the RC-based DAE does lead to a small extra gain in strongly mismatched conditions (clean speech training and noisy test samples) but that this comes at the expense of a significant loss of the performance for matched conditions (clean test samples). In the multi-style training experiment where the mismatch remains moderate, the RC-based denoiser even has a (small) detrimental effect in all conditions. This can only mean that the DAE removes information from the feature stream that is otherwise exploitable by the RC back-end.

As could be expected, the Test on Baseline configuration is characterized by a strong detrimental effect in matched conditions because the features used during training and test have become different even in these conditions.

In a control experiment, we also trained and tested the RC-based denoising of MFCCs by inserting a DAE at position P2. In line with the correlations measured before, the emerging systems achieved very similar result showing no preference for positions P2 and P3.

VII. CONCLUSION AND FUTURE WORK

Recently, we were able to show that reservoir computing (RC) is a viable acoustic modeling technique that can lead to more robust automatic speech recognition (ASR). Experiments on the Aurora-2 benchmark demonstrated that our RC-based systems already outperform the most complex GMM-HMM based systems on the task of continuous digit recognition.

The main objectives of the present paper were: (1) to establish whether reservoir networks can be trained to denoise the feature vectors, (2) to investigate how much benefit can be attained by applying denoising in the front-end of the RC-recognizer, (3) to establish whether RC-based denoising can outperform traditional signal processing based techniques (e.g. like in the ETSI advanced front-end (AFE)), and (4)

to find out whether the two denoising techniques are maybe complementary.

First of all, we could show that an RC-based denoising auto encoder (DAE) imputed at a suitable position in the MFCC front-end can lead to increasing mean correlations between the features of the noisy and the clean speech utterances.

Next, we could demonstrate that adopting feature denoising in the front-end of an RC-based ASR is beneficial, but not as much as it is in the case of a traditional GMM-based ASR. A somewhat disappointing result is that an RC-based DAE fails to induce as much improvement as the AFE. However, by combining the two tecniques, a small gain is possible to achieve in severely mismatched conditions, be it at the expense of a small degradation in the matched condition.

In the near future we will compare the RC-based DAE with the AFE in situations where the noise is more non-stationary. Furthermore, we hope to apply some of the ideas which lead to uncertainty decoding in an RC-backend. Finally, we intend to extend our research to noise robust large vocabulary speech recognition (e.g. Aurora-4).

ACKNOWLEDGEMENT

The research leading to the results presented here has received funding from Flemish Science Foundation (FWO) under grant agreement G.0088.09N (RECAP).

REFERENCES

[1] ETSI, "Speech processing, transmission and quality aspects STQ; distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," ES 202 050, Tech. Rep., 2002.

[2] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 257–270, jan. 2007.

[3] M. Van Segbroeck and H. Van Hamme, "Advances in missing feature techniques for robust large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 123–137, 2011.

[4] H. Xu, M. Gales, and K. Chin, "Joint uncertainty decoding with predictive methods for noise robust speech recognition," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 6, pp. 1665–1676, 2011.

[5] O. Vinyals and S. Ravuri, "Comparing multilayer perceptron to deep belief network tandem features for robust ASR," in *Proc. ICASSP*, 2011, pp. 4596–4599.

[6] S.-X. Zhang and M. Gales, "Structured support vector machines for noise robust continuous speech recognition," in *Proc. Interspeech*, 2011, pp. 989–992.

[7] A. Jalalvand, F. Triefenbach, and J.-P. Martens, "Continuous digit recognition in noise: Reservoirs can do an excellent job!" in *Proc. Interspeech*, 2012, p. ID:644.

[8] A. Jalalvand, K. Demuynck, and J.-P. Martens, "Noise robust continuous digit recognition with reservoir-based acoustic models," in *Proc. ISPACS*, 2013, p. ID:99.

[9] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. ICASSP*, vol. 1, 2001, pp. 301–304.

[10] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *Proc. ICASSP*, apr 1988, pp. 553–556.

[11] H. Jaeger, "The ''echo state'' approach to analysing and training recurrent neural networks - with an erratum note," GMD Report 148, German National Research Center for Information Technology, Tech. Rep., 2001.

[12] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *IEEE Trans. Neural Networks*, vol. 20, pp. 391–403, 2007.

[13] A. Jalalvand, F. Triefenbach, D. Verstraeten, and J.-P. Martens, "Connected digit recognition by means of reservoir computing," in *Proc. Interspeech*, 2011, pp. 1725–1728.

[14] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, Apr 2007.

[15] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 659–686.

[16] F. Triefenbach, A. Jalalvand, K. Demuynck, and J.-P. Martens, "Acoustic modeling with hierarchical reservoirs," *IEEE Trans. Audio, Speech and Language Processing*, vol. PP, no. 99, 2013.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008, pp. 1096–1103.

[18] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoustics,Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[19] P. J. Wolfe and S. J. Godsill, "Efficient alternatives to the ephraim and malah suppression rule for audio signal enhancement," in *special issue) EURASIP JASP on Digital Audio for Multimedia Communications*, 2003, pp. 1043–1051.

[20] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "Robust speech recognition using a cepstral minimum-mean-square-error-motivated noise suppressor," *IEEE Trans. Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 1061–1070, 2008.

[21] K. Indrebo, R. Povinelli, and M. Johnson, "Minimum mean-squared error estimation of mel-frequency cepstral coefficients using a novel distortion model," *IEEE Trans. Audio, Speech and Language Processing*, vol. 16, no. 8, pp. 1654–1661, 2008.

[22] L. Deng, J. Droppo, and A. Acero, "Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features." *IEEE Trans. Speech Audio Processing*, vol. 12, no. 3, pp. 218–233, 2004.

[23] R. Rotili, E. Principi, S. Cifani, F. Piazza, and S. Squartini, "Multi-channel feature enhancement for robust speech recognition," *Speech technologies. InTech, ISBN*, pp. 978–953, 2011.

[24] H.-G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noise conditions," in *Automatic Speech Recognition: Challenges for the Next Millennium*. ISCA ITRW, 2000, pp. 181–188.

[25] H. G. Hirsch and D. Pearce, "Applying the advanced ETSI frontend to the Aurora-2 task," version 1.1, Tech. Rep., 2006.

[26] X. Xiao, J. Li, E.-S. Chng, H. Li, and C.-H. Lee, "A study on the generalization capability of acoustic models for robust speech recognition," *IEEE Trans. Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1158–1169, 2010.

[27] T. Kai, M. Suzuki, and K. Chijiiwa, "Combination of SPLICE and feature normalization for noise robust speech recognition," *Intl. Workshop on Nonlinear Circuits, Communications and Signal Processing*, vol. 16, no. 4, pp. 323–326, 2012.

[28] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Computation*, vol. 7, pp. 108–116, 1994.