System Modeling for Active Noise Control with Reservoir Computing

Jens Nyman*†, Ken Caluwaerts†, Tim Waegeman†, Benjamin Schrauwen†
*email: nymanjens.nj@gmail.com
†Reservoir Lab, Faculty of Engineering and Architecture
Ghent University
Ghent, Belgium

***Abstract*—This paper investigates the use of reservoir computing for active noise control (ANC). It is shown that the ANC problem can be solved by a concatenation of physically present subsystems. These subsystems can be modelled by reservoirs that are trained, using one shot learning. This approach is compared to genetic algorithms tuning a Volterra filter. Experimental results show that our approach works well as system model, meaning that a reservoir trained on white noise performs good on other input signals as well. This is a major advantage over genetic algorithms that generalize rather badly. Furthermore, our approach needs less data and this data can be gathered in one experiment only.**

***Index Terms*—Active Noise Control (ANC), System Modeling, Reservoir Computing**

I. Introduction

Unwanted noise has since long annoyed human beings and a lot has been done to reduce these disturbing signals. The classical approach to do this, is by passive noise control such as ear plugs and anti-noise screens along highways. The main drawback of these techniques is that they have problems with low-frequency noise, since the attenuation through blocking material is inversely proportional to the wavelength.

Active noise cancellation (ANC) produces an anti-noise signal with opposite phase as the incoming noise. Superposition of these waves leads to an attenuated leftover noise [1], [2]. As the output signal should destructively interfere in a larger space than only at the speaker, this method works best when the dimensions of the noise source and speakers are small with respect to the wavelength. Furthermore, if the signal has to be processed digitally, the Shannon theorem dictates a lower sampling frequency for lower frequencies, while also a possible prediction task is easier and processing time constraints are more relaxed. It is thus clear that ANC is easiest with low frequencies, which makes it an excellent addition to passive noise cancellation.

As early as 1936, Paul Lueg published a patent about ANC with a microphone and a loudspeaker. The technology has since then evolved with many systems and applications as a result [3]. The filtered-x LMS (FXLMS) algorithm became a popular choice for filtering recorded noise into a compensating signal and many implemented or improved it [4], [5]. One of these improvements was the Volterra FXLMS, which adds non-linear terms to the Finite Input Response (FIR) filter [6], [7]. This extension was created because some systems appeared to show non-linear behaviour and the linear FXLMS could not cope with this. These non-linearities can have many causes: non-linear properties of air at high pressure, AD/DA conversion, loudspeakers and amplifiers. Main disadvantages of these methods are that they may converge to local minima and that they need the identification of the secondary path, being the path from the compensating speaker to the error microphone.

An interesting alternative is using genetic algorithms. Russo and Sicuranza [6], [8] and Chang and Chen [9] state that these do not suffer the previously mentioned disadvantages, while still providing good performance. They use a standard genetic algorithm, where every entity in the population represents a set of FIR or Volterra filter coefficients. A new generation is then created by combining sets of two entities into children. The parents are selected based on their fitness, which in our case is based on the error signal (e.g. the mean square error).

Instead of performance-based learning, this paper focusses on supervised learning via *reservoir computing*. This technique starts with a recurrent neural network with randomly initialized weights. This network is called a *reservoir*. Instead of training the network itself, which is the usual approach for neural networks, only the readout is trained [10]–[12]. For single-input-single-output systems, this means that one inserts one sample every time step, so all N nodes in the reservoir have a value that is a (complicated) function of all past inputs. This N-dimensional sequence of node values are then matched to the one-dimensional output sequence via ridge regression, which is linear regression with regularization.

It has been shown that reservoir computing works well on digital signal processing tasks like speech recognition [10], [13] and chaotic time series prediction [14]. This last result is especially important for ANC because several noise mechanisms have shown to exhibit chaotic behaviour [15], [16].

The rest of this paper will be organised as follows. Section II will discuss the considered setup. Section III briefly explains the genetic algorithm implementation that is used as reference. Section IV discusses the core of our approach. The experimental setup is described in section V and results are shown in section VI. A conclusion is drawn in section VII.

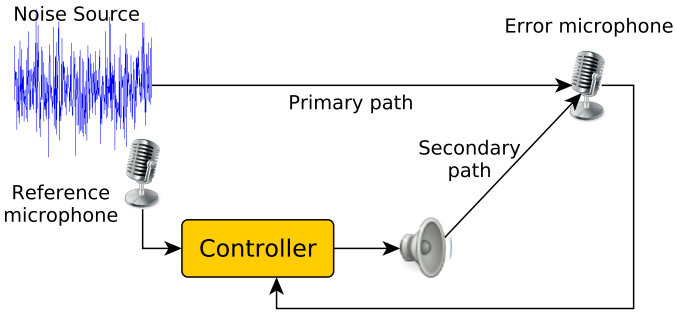

Fig. 1. ANC setup with two microphones. The controller is generally filtering the input from the reference microphone and is adapted by the error signal.

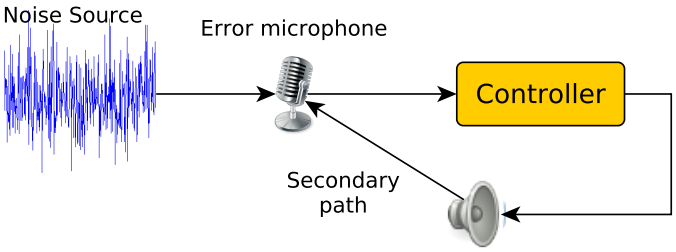

Fig. 2. ANC setup with one microphone. The controller generates a compensating signal from the error signal. The term "secondary path" is used for consistency with the first setup.

II. ANC SETUP

In this paper, we consider two ANC setups: one using two microphones and another using one microphone. The first setup is the same as in [6], [8], [9] and is depicted in Figure 1. Two paths are important here: the primary from the reference point to the error microphone and the secondary from the compensating speaker to the error microphone. The usual approach is to use the reference signal as input for generating a compensating signal, and to use the error signal to adapt the filter. This is also the approach we will follow.

The second setup is a simplification of the first in terms of hardware and acoustical separation. The first setup assumes perfect isolation between the speaker and the reference microphone. This is either not true or puts strict requirements on the mechanical layout. This simplification comes at the cost of less input signals.

III. ADAPTIVE GENETIC ALGORITHM

As reference, we implemented the adaptive genetic algorithm of [8] and [9] by applying the same system (Figure 3) with the same Volterra filter as controller block. We also used the same population (600) and number of generations (500) as in their work. As the reward function, the *Mean Square Error* (MSE) was taken from [8] and 200 train samples were considered. This means that the fitness function for all entities was evaluated using the same 200 training samples. After some optimization of the other parameters, the coefficients evolved to an acceptable solution. As reference, we also tried out *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) [17], [18] on the same error function. CMA-ES is another method to optimize coefficients based on a generic error function.

Genetic algorithms perform well for this task, but the adaptive nature of them can be questioned. The main problem is that the number of experiments for an acceptable solution is very large. Take 500 generations with a population of 600, this requires $600 \cdot 500 = 300.000$ experiments. Even more problematic is an expected performance drop when the applied noise characteristics change during training. While comparing genes of the same population, it will indeed lead to false comparisons if the hardness of the task differs.

IV. SYSTEM MODELLING WITH RESERVOIR COMPUTING

We propose solving above issues by system modelling with reservoir computing. To train the output layer of a reservoir, it only takes one experiment of a couple of seconds. As soon as the right models are made, the ideal compensating signal can be calculated. This will further be explained in more detail.

For modelling a system, we used a leaky reservoir with $\tanh$ as non-linear function followed by ridge regression. We optimized the ridge parameter, leak rate, input scaling, spectral radius and bias on a validation set. All reservoir models are compared to regular ridge regression, using a window of 80 samples including quadratic terms:

$$y(n) = \sum_{i=0}^{79} h_i^{(1)} x(n-i) + \sum_{i=0}^{79} h_i^{(2)} x^2(n-i) \qquad (1)$$

where x and y are input and output signals respectively of an individual system and $h_i^{(j)}$ are the weights that need to be trained. We optimized the ridge parameter on a validation set.

A. Two-microphone system

Working on the block diagram of Figure 3, it can easily be seen that the ideal controller is:

$$\begin{aligned} \text{Controller} &= S^{-1} \circ P && (2)\\ &= S^{-1} \circ M^{-1} \circ M \circ P && (3)\\ &= (M \circ S)^{-1} \circ (M \circ P) && (4) \end{aligned}$$

where the $\circ$-operator means 'after'. Since the systems could be non-linear, commutativity does not hold.

The problem is thus reduced to finding a model for $M \circ P$ and $(M \circ S)^{-1}$. Therefore, we need to capture meaningful signals at the input and output of the systems. For $M \circ P$, one sets $c(n)$ to zero and records the noise signal $x(n)$ and the error signal $e(n)$. Assuming an ideal reference microphone $(M'(x(n)) = x(n))$, this effectively captures $M \circ P$. For $(M \circ S)^{-1}$, the noise source $x(n)$ is silenced and a signal $c(n)$ is applied to the speaker. Since the nature of the compensating signal is not known beforehand, we chose white noise as $c(n)$. By recording the error signal, $M \circ S$ is captured and by swapping input and output, $(M \circ S)^{-1}$ is also captured. As soon as enough input-output pairs are found, a reservoir output layer can be trained to mimic the behaviour of the $(M \circ S)^{-1}$.

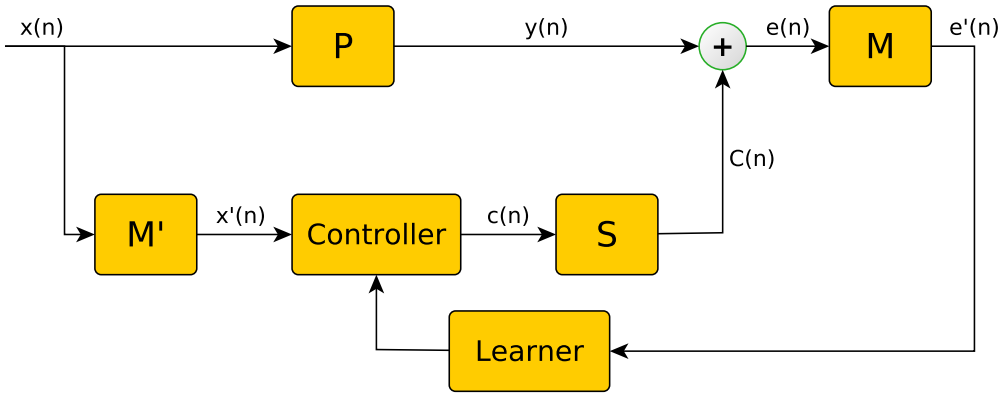

Fig. 3. Block diagram of the two-microphone system studied in this paper. P, S and $M^{(')}$ symbolize the systems governing the primary path, secondary path and microphone systems respectively. In general, these are non-linear systems.

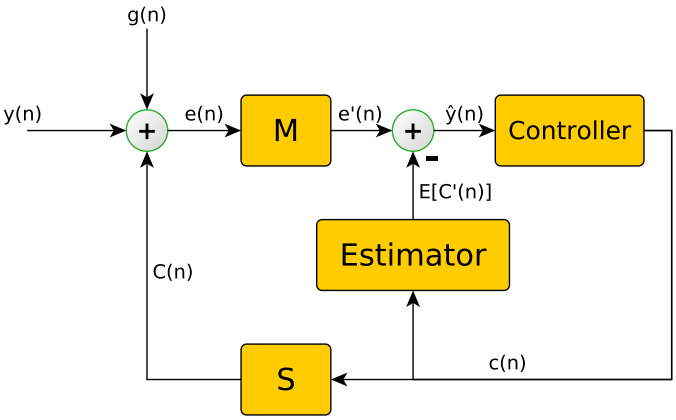

Fig. 4. Block diagram of the one-microphone system studied in this paper. S and M symbolize the secondary path and microphone system respectively. $g(n)$ is sensor noise and equal to zero if unspecified.

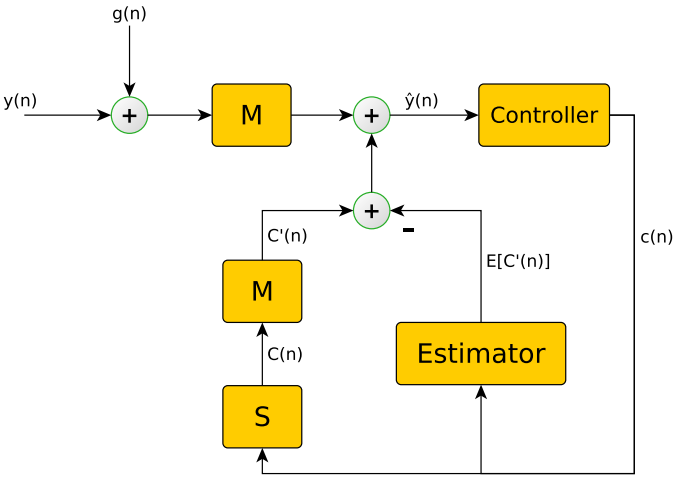

Fig. 5. Reorganized block diagram of Figure 4 with the assumption that M is linear.

If S introduces a larger delay than P, modelling $S^{-1} \circ P$ equals predicting the future. This is impossible for some signals and feasible for others. We do not consider this problem as it is the subject of numerous other works [19]–[22].

B. One-microphone system

Figure 4 shows the block diagram of the one-microphone system. We introduced an additional *Estimator* block because the error microphone receives $y(n)$, $g(n)$ and $C(n)$, but *Controller* should compensate $y(n)$ only. Consequently, we try to estimate $y(n)$ from $e(n)$ and $c(n)$.

If we assume that M is a linear system, we can reorganize this diagram to get Figure 5. It is clear that, in order to get $e'(n) \equiv 0$, *Estimator* should model $M \circ S$ and that *Controller* should model $(M \circ S)^{-1}$. The acquisition of train data for these is even easier because it requires only one experiment. Note that the controller will always need a prediction module.

V. Experimental setup

All simulations were performed with a frequency of $8\,\mathrm{kHz}$.

A. Simulation

We simulated our approach with the same (non-linear) path definitions as in [8] and [9]. These definitions are

$$P : x \to y \tag{5}$$

with

$$\begin{cases} s(n) = x(n-5) - 0.3x(n-6) + 0.2x(n-7) \\ y(n) = s(n-2) + 0.08s^2(n-2) - 0.04s^3(n-1) \end{cases} \tag{6}$$

for the primary path, and

$$S : c \to C \tag{7}$$

with

$$\begin{cases} r(n) = 0.66\tanh(1.5c(n)) \\ C(n) = r(n-2) + 1.5r(n-3) - r(n-4) \end{cases} \tag{8}$$

for the secondary path. As test signal, we use the logistic noise of [8] generated by the equation $x(n+1) = \lambda x(n)[1-x(n)]$ with $\lambda = 4$ and $x(0) = 0.9$. Additionally, we also used a white noise signal and a $200\,\mathrm{Hz}$ sine wave with some added white

Fig. 6. Real speaker and microphone setup.

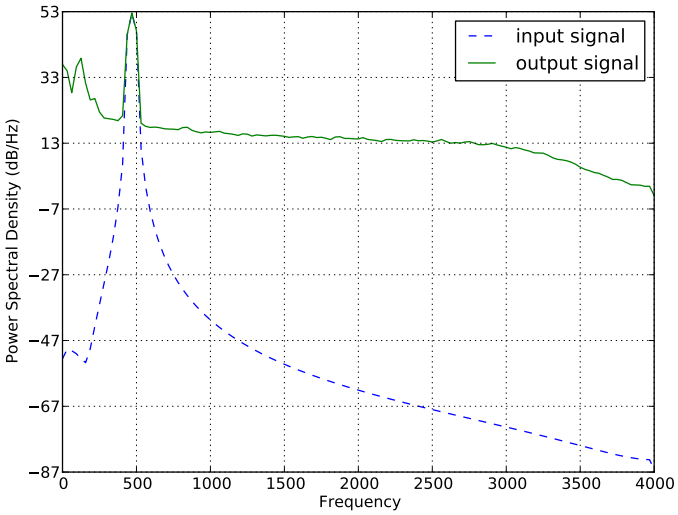

Fig. 7. Power spectral density of the speaker-to-microphone system with a 470 Hz sine wave as input. The output signal was normalized so both signals have equal energy.

noise. The added white noise had a standard deviation of 0.1 times the sine amplitude

B. Real speaker and microphone

We tested our approach in a practical setup consisting of a Shure SM58 microphone and a Harman/Kardon HK206 speaker as can be seen in Figure 6. The applied signal was a white noise signal. While a complete system has not yet been implemented, the necessary experiments were done to evaluate the system without predictor.

VI. Experimental results

All results are summarized in Table I and the optimal parameter values and intermediate results are stated in Table II. If we look at the genetic algorithms to which we compare, we see poor performance when applied to white noise. This suggests that the technique relies on the signal to be easily predictable meaning that the output signal is memorized rather than the system.

Our system suffers much less from this problem. A reservoir trained on white noise is able to to generalize quite well. For example, the $(M \circ S)^{-1}$ system, trained on white noise gave an MSE/E_t of $-10.39\,\mathrm{dB}$ for white noise and $-12.65\,\mathrm{dB}$ for logistic noise. This means that the reservoir actually performs better on an unknown (easier) signal than on the training signal. To show that this is not trivial, the ridge regression performs worse on the logistic data, as can be seen in Table II.

In simulation, the reservoir + ridge regression approach always outperformed ridge regression, what means that the non-linear nature of the filters plays a significant role and that a system with linear and second order terms cannot model this. This ability to model non-linearities is the strength of the reservoir approach.

The one-microphone setup in simulation has a similar performance as the two-microphone setup. This is partly because noise prediction is ignored (noise prediction on white noise is of course impossible). On the other hand, good performance is obtained because the $M \circ S$-model performs a lot better than the $(M \circ S)^{-1}$-model (Table II). Just like in all other simulations setups, the performance thus mainly depends on the $(M \circ S)^{-1}$-model.

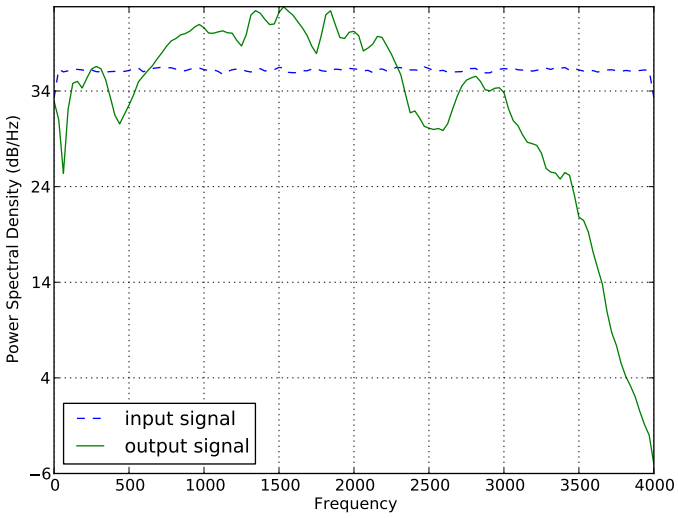

Fig. 8. Power spectral density of the speaker-to-microphone system with white noise as input. The output signal was normalized so both signals have equal energy.

For the practical system, results are quite different. Here, ridge regression performs better than the reservoir. Furthermore, omitting the second order terms of the ridge regression does not significantly influence performance. The model for this real-world system is thus a linear one. This linear hypothesis is supported by Figure 7 which suggests that the system is a linear filter plus noise. Assuming a linear filter and white added noise, the filter would look like the output signal of Figure 8.

		Reduction (dB)			
	Signal	Sine	Logistic	White noise	
	#Microphones	2	2	2	1
Simulation	Genetic	17.14	17.06	6.70	
	CMA-ES	17.67	17.20	1.80	
	Ridge regression	10.21	8.38	6.31	5.44
	Reservoir	17.42	16.17	13.64	12.62
Practical	Ridge regression				11.73
	Reservoir				10.60

TABLE I
EXPERIMENTAL RESULTS.

			$\mathbf{MSE/E_t}$ (dB)		Optimized reservoir parameters				
	System	**Signal**	**Reservoir**	**Ridge regr.**	**# Nodes**	**Leak rate**	**Input scaling**	**Bias**	**Spectral radius**
Simulation	$M \circ P$	Sine	-66.39	-40.23	100	1	0.1	0.4	0.5
	$M \circ P$	Logistic	-75.69	-36.99	100	0.8	0.1	0.4	0.5
	$M \circ P$	White noise	-41.79	-19.21	100	0.8	0.1	0.4	0.5
	$M \circ S$	White noise	-17.87	-9.04	100	1	0.5	0.4	0.5
	$(M \circ S)^{-1}$	White noise	-10.39	-8.77	100	0.8	0.1	0	0.5
	$(M \circ S)^{-1}$	Logistic*	-12.65	-7.53					
Practical	$M \circ S$	White noise	-13.41	-16.11	300	1	0.05	0	0.95
	$(M \circ S)^{-1}$	White noise	-8.35	-8.73	300	1	0.05	0	0.95

TABLE II
OPTIMIZED PARAMETER VALUES AND INTERMEDIATE EXPERIMENTAL RESULTS. E_t IS THE ENERGY OF THE TARGET SIGNAL. * THIS SYSTEM WAS TRAINED ON WHITE NOISE AND TESTED ON LOGISTIC CHAOTIC NOISE.

VII. CONCLUSION

In comparison with the genetic algorithms, our approach has some major improvements. Firstly, it takes just one experiment of a couple of seconds to train the whole system whereas genetic algorithm experiments take as long as the entire evolution which can easily be 15 minutes or more. Secondly, our approach is better in modelling the system for white noise as input signal. This model also turns out to generalize quite well for other signal types. This generalisation is an important property for practical systems, because it is unlikely that a noise source stays the same forever. Even for easy noise sources like fans, changes in the environment or ageing influences their nature. A final conclusion is that simplified setups in home environments turn out to be linear.

There are still a lot of improvements possible for enhancing our approach. Firstly, it would be interesting to see if performance could improve by feature engineering. Now, the reservoir has only one input (the most recent sample), but this could be replaced for example by a filter bank. Secondly, since the reservoir readout is standard ridge regression, techniques could be investigated to iteratively update the readout weights in order to adapt to new situations. This is necessary when the subsystems (P, S and M) are influenced by changing conditions. Finally, a predictor should be added to the system. It is possible that the reservoir itself could act as predictor [20], [21] and at the same time model the system. This combining would result in a compact model that requires a limited amount of resources.

REFERENCES

[1] S. Kuo and D. Morgan, *Active noise control systems: algorithms and DSP implementations*. John Wiley & Sons, Inc., 1995.

[2] S. Elliott and P. Nelson, "Active noise control," *Signal Processing Magazine, IEEE*, vol. 10, no. 4, pp. 12–35, 1993.

[3] S. Kuo and D. Morgan, "Active noise control: a tutorial review," *Proceedings of the IEEE*, vol. 87, no. 6, pp. 943–973, 1999.

[4] E. Bjarnason, "Analysis of the filtered-x lms algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 6, pp. 504–514, 1995.

[5] M. Rupp and A. Sayed, "Robust fxlms algorithms with improved convergence performance," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 78–85, 1998.

[6] F. Russo and G. Sicuranza, "Genetic optimization in nonlinear systems for active noise control: Accuracy and performance evaluation," in *Instrumentation and Measurement Technology Conference, 2006*. IEEE, 2006, pp. 1512–1517.

[7] D. Zhou and V. DeBrunner, "Efficient adaptive nonlinear filters for nonlinear active noise control," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 3, pp. 669–681, 2007.

[8] F. Russo and G. Sicuranza, "Accuracy and performance evaluation in the genetic optimization of nonlinear systems for active noise control," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 4, pp. 1443–1450, 2007.

[9] C. Chang and D. Chen, "Active noise cancellation without secondary path identification by using an adaptive genetic algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 9, pp. 2315–2327, 2010.

[10] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Networks*, vol. 20, no. 3, pp. 391–403, 2007.

[11] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[12] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*, 2007, p. 471.

[13] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J. Martens, "Phoneme recognition with large hierarchical reservoirs," *Advances in neural information processing systems*, vol. 23, pp. 2307–2315, 2010.

[14] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, p. 78, 2004.

[15] T. Matsuura, T. Hiei, H. Itoh, and K. Torikoshi, "Active noise control by using prediction of time series data with a neural network," in *IEEE International Conference on Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century.*, vol. 3. IEEE, 1995, pp. 2070–2075.

[16] P. Strauch and B. Mulgrew, "Active control of nonlinear noise processes in a linear duct," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2404–2412, 1998.

[17] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[18] N. Hansen, "The cma evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.

[19] E. Kayacan, B. Ulutas, and O. Kaynak, "Grey system theory-based models in time series prediction," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1784–1789, 2010.

[20] F. Wyffels and B. Schrauwen, "A comparative study of reservoir computing strategies for monthly time series prediction," *Neurocomputing*, vol. 73, no. 10, pp. 1958–1964, 2010.

[21] Q. Song and Z. Feng, "Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series," *Neurocomputing*, vol. 73, no. 10-12, pp. 2177–2185, 2010.

[22] M. Salmasi, H. Mahdavi-Nasab, and H. Pourghassem, "Evaluating the performance of mlp neural network and grnn in active cancellation of sound noise," *Canadian Journal on Artificial Intelligence*, vol. 2, no. 2, pp. 28–33, 2011.