*Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 2, 129–137*
DOI: 10.2478/v10006-008-0012-0

amcs

# TOWARDS SPIKE-BASED SPEECH PROCESSING: A BIOLOGICALLY PLAUSIBLE APPROACH TO SIMPLE ACOUSTIC CLASSIFICATION

ISMAIL UYSAL, HARSHA SATHYENDRA, JOHN G. HARRIS

Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL 32611, USA
e-mail: {ismail,sathyendra,harris}@cnel.ufl.edu

Shortcomings of automatic speech recognition (ASR) applications are becoming more evident as they are more widely used in real life. The inherent non-stationarity associated with the timing of speech signals as well as the dynamical changes in the environment make the ensuing analysis and recognition extremely difficult. Researchers often turn to biology seeking clues to make better engineered systems, and ASR is no exception with the usage of feature sets such as Mel frequency cepstral coefficients, which employ filter banks similar to cochlear filter banks in frequency distribution and bandwidth. In this paper, we delve deeper into the mechanics of the human auditory system to take this biological inspiration to the next level. The main goal of this research is to investigate the computation potential of spike trains produced at the early stages of the auditory system for a simple acoustic classification task. First, various spike coding schemes from temporal to rate coding are explored, together with various spike-based encoders with various simplicity levels such as rank order coding and liquid state machine. Based on these findings, a biologically plausible system architecture is proposed for the recognition of phonetically simple acoustic signals which makes exclusive use of spikes for computation. The performance tests show superior performance on a noisy vowel data set when compared with a conventional ASR system.

**Keywords:** Spike coding, synchrony coding, phase locking, speech perception, psychoacoustics, speech recognition.

## 1. Introduction

The research discussed in this paper has two mutually inclusive goals. While exploring the spike coding mechanisms employed by the auditory system at different sound pressure levels (SPLs) and signal-to-noise ratios (SNRs), we aim to introduce a novel way of information processing for speech signals trying to imitate the auditory neural computation. The biggest motivation behind this research is the fact that human engineered automatic speech recognition (ASR) systems perform poorly as the variability associated with the speech signal increases, especially in noisy environments. On the other hand, the human brain is capable of processing the ever-continuous stream of input with an unparalleled accuracy. Even though much is still unknown about how brain exactly works, it is well known that neurons in the brain use action potentials to communicate the timing information from the sensory to more complex levels of processing in the cortex. We strongly believe that computation with spike trains is not a mere artifact of biology, but instead it holds the key to the robustness and performance of the auditory system.

The idea of using bio-inspired techniques for machine recognition tasks is certainly not a new concept. In ASR, the most commonly used feature set, Mel frequency cepstral coefficients (MFCC), imitates the distribution of cochlear filter banks by employing logarithmically distributed filters along the frequency axis (Davis and Mermelstein, 1980). More elaborated approaches include human factor cepstral coefficients, which use known facts from human psychoacoustics such as the relationship between center frequency and critical bandwidth to decouple filter bandwidth from filter spacing (Skowronski and Harris, 2004). The performance and robustness of those techniques compared with previous commonly used feature extractors, such as linear predictive coding, clearly shows the advantage of using inspiration from biology and psychoacoustics in particular (Atal and Hanauer, 1971).

The objective of this research is to take this inspiration one step further to include the neural computation used in the human auditory system for both the feature extraction and recognition stages of a simple acoustic classification problem. In order to accomplish this goal, we have followed a systematic approach where dif-

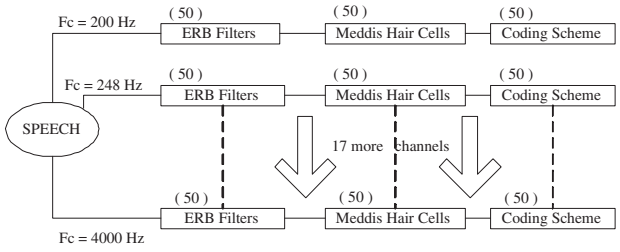


Fig. 1. Speech-to-spike conversion block which shows the transduction of sound waves into trains of action potentials at nerve fibers connected to inner hair cells.

ferent spike-based operators are evaluated along with different spike coding techniques for different types and intensities of noise and input acoustic stimuli (Uysal *et al.*, 2006; Uysal *et al.*, 2007a; Uysal *et al.*, 2007b).

Section 2 briefly discusses the mechanism which converts incoming acoustic signals to action potentials on nerve fibers. Various coding schemes applied on these spike trains are discussed in Section 3. Sections 4 and 5 deal with spike-based operators and how they are used in conjunction with spike coding schemes. The acoustic classification problem and test results are provided in Section 6.

## 2. Front end: Speech-to-spikes

In order to build a spike-based architecture for speech recognition, one should use a biologically realistic conversion from acoustic stimuli to action potentials. In the human ear, the cochlea is the region where this electromechanical conversion takes place. One of the most commonly used and realistic models of the cochlea is due to Meddis (1986). According to this model, sound pressure waves are converted into a mechanical motion at the basilar membrane (BM), which has a tonotopic distribution of inner hair cells. When the sound pressure wave vibrates the BM, the attached hair cells deflect, which changes the permeability of the cell membranes that make synaptic connections to nerve fibers. A change in permeability also changes the amount of neurotransmitters currently present in the synaptic cleft, which starts the mechanism for generating the action potentials at the nerve fibers. The model used in this paper takes into account many improvements made over this basic concept throughout the years such as the introduction of adaptation as seen in the human cochlea and the non-linear temporal properties associated with spike generation (Sumner and Lopez-Poveda, 2002; Sumner *et al.*, 2003)

Figure 1 shows the front end of the overall spike-based architecture. The speech is passed through a series of equivalent rectangular bandwidth (ERB) filters spanning the frequency range of 200Hz to 4kHz, which includes most of the frequency content present in a typical speech signal. The center frequencies of these filters are distributed logarithmically and follow the frequency resolution observed in real cochlear filters. Hence, the resolution is a decreasing function of frequency with more filters centered around lower frequencies. There are 20 frequency channels and 50 filters in each channel for more accurate application of some of the coding schemes such as synchrony and rate coding, while also having a reasonably low computational cost. The outputs of these filters are passed onto the Meddis hair cell model and are associated with probabilities of spike firing on respective neuron fibers. Finally, spike trains observed on the nerve fibers are analyzed and encoded using various spike coding schemes and spike-based operators for performance comparisons.

## 3. Spike coding techniques and their application to nerve fibers

This section explains the three different spike coding schemes used in this paper in greater detail. The three schemes are: rate coding, which encodes information in the frequency of spike occurrence, direct temporal coding, which uses the timing of each and every spike for computation, and synchrony coding, which groups neurons with similar firing times.

At this step, it is important to note that there have been previous attempts to use spike trains for speech classification (Hopfield and Brody, 2001; Verstraeten *et al.*, 2005). One of the major differences between the proposed architecture and these approaches is the enforcement of the spike coding schemes on top of the spike trains from the cochlea. Spike coding is not only used as a feature extractor but also to investigate how robustness might be encoded within these spike trains. Without any applied coding scheme, the spike-based operator is forced to differentiate between the higher order features contained within a spike train, thus decreasing the robustness of the overall system.

**3.1. Rate coding.** Rate coding, without a doubt, is still among the most common schemes applied by scientists to real world problems to discover possible connections between behavior and observed spike train data (Rieke *et al.*, 1999). Averaging over time simply assumes that the frequency of spike firing on a particular nerve fiber is the feature which carries the information onward. Other rate coding examples include averaging over a number of experimental trials to yield spike density, or averaging over populations of neurons as population activity.

On the other hand, rate coding has some important restrictions especially when it comes to nerve fibers right after the cochlear region. It has been shown that, during a regular conversation, the input SPLs fluctuate around 60dB and most nerve fibers are simply saturated, firing

as fast as they can (Sachs, 1984). This means, given two different inputs, that as long as their SPLs are around conversational levels, the rate coding at the end of nerve fibers will not be able to differentiate between the two different acoustic stimuli.

Hence, this paper investigates rate coding at two different SPLs, namely 60 dB and 10 dB, pointing towards a duplex theory of spike coding, which might depend on the input SPL and noise levels.

**3.2. Direct temporal coding.** In contrast to rate coding, temporal coding relies on the precise timing of action potentials. Phase coding, time-to-first spike coding are all examples of temporal coding commonly used by neuroscientists (Dayan and Abbott, 2001; VanRullen *et al.*, 2005; Terman and Wang, 1995). In particular, direct temporal coding uses the spike firing times as-is, and all the timing information is supplied to the spike-based operator. For the architecture in the system, this corresponds to 1000 individual spike trains being fed into the spike-based operator. Depending on the classifier architecture, the process is simplified by using fewer spike trains per frequency channel, which will be discussed in greater detail in Section 5.

**3.3. Synchrony coding.** Synchrony coding can be viewed as a special type of temporal coding which groups neurons with similar firing times. It is used to explain the group communications of neurons especially on the sensory level (Terman and Wang, 1995).

We believe that the redundancy caused by the number of nerve fibers packed densely along the basilar membrane gives rise to such synchrony, which is one of the factors explaining the robustness of the auditory system to noise.

In the literature one can find many different definitions for synchrony amongst spike trains (Moissl and Meyer-Base, 2000). The concept of synchrony coding as applied in this paper's architecture is best explained with a simple example. Consider the vowel /iy/ as in "beet". For simplicity, let us assume there are 20 nerve fibers placed at a particular frequency channel. When the acoustic signal, /iy/, is given as input to the frequency channel centered at 300Hz, the spike trains that are observed along these nerve fibers are shown in Fig. 2.

The input signal is corrupted with white noise at a very low level of the SNR around 5 dB and is barely audible at such a noise intensity. When the spike trains on the 10 nerve fibers are observed, it is difficult to see a trivial pattern regarding action potential timings.

The next step is to look at the inter-spike time interval distribution which is shown in Fig. 3. As is clearly observed from the figure, these nerve fibers are phase locked to integer multiples of $T = 3.26$ ms. Hence, the phase locked frequency is $F_{pl} \approx 305$ Hz. Figure 4 shows the

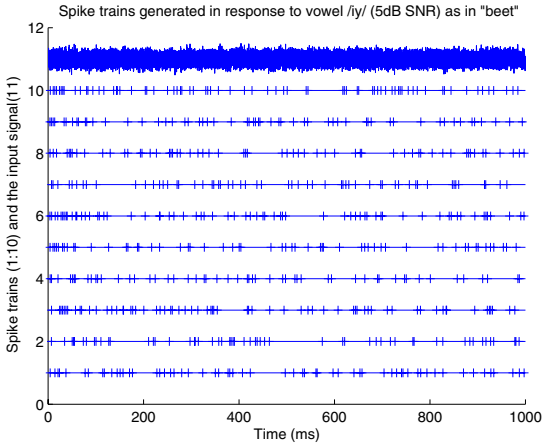


Fig. 2. Spike train outputs for a set of 10 hair cells all with a central frequency of 300Hz.

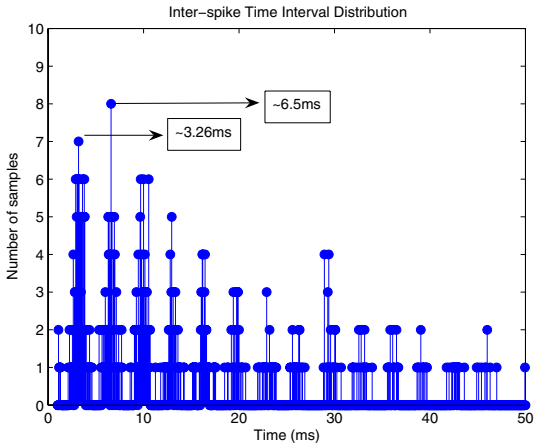


Fig. 3. Inter-spike time intervals for a set of hair cells centered at 300Hz.

spectral magnitude of the inter-spike time interval distribution compared to the log-magnitude spectral envelope of the input vowel. Vowels are predominantly defined by their first three formant frequencies, which are the peaks shown in the top plot. In particular, the vowel /iy/ has its first formant frequency at $F_1 = 305$ Hz.

Figure 4 shows that the spectral magnitude plot has a peak at $F_p = 305$ Hz, indicating that the nerve fibers centered at $F_c = 300$ Hz were able to phase lock to the first formant frequency even for such a noisy signal. However, the noise robust feature which will carry this information forward is not the frequency they are phase locked to, but the degree of phase locking, which is simply the magnitude of the peak at that particular frequency. The reasoning behind this will become more apparent in Fig. 5.

Figure 5 shows the spectral magnitude of the inter-spike interval distribution corresponding to two different sets of hair cells centered at $F_{c1} = 300$ Hz and $F_{c2} = 200$

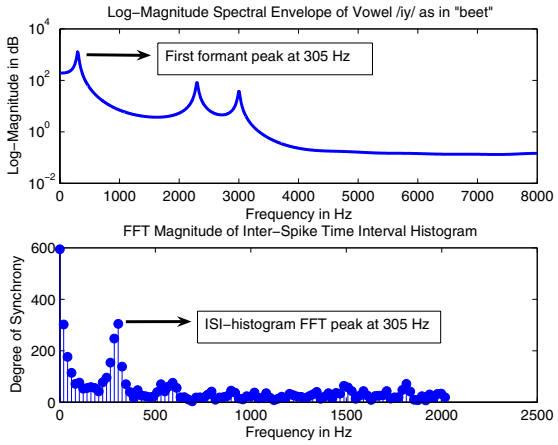


Fig. 4. Log-magnitude spectral envelope for /iy/ and the corresponding degree of synchrony for a set of hair cells centered at $F_c = 300$ Hz (computed for a noisy utterance with a 5 dB SNR).

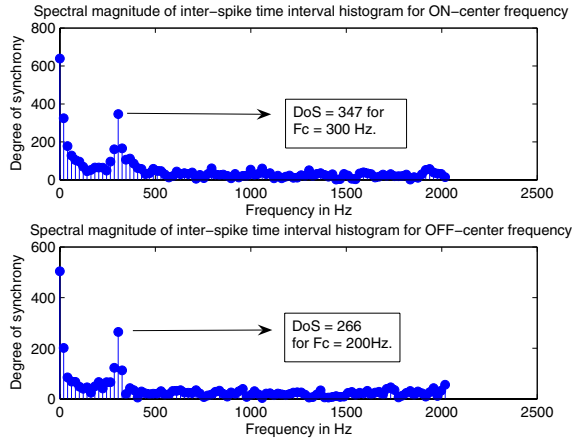


Fig. 5. Degree of synchrony within 2 sets of hair cells centered at $F_c = 300$ Hz and $F_c = 250$ Hz in response to a noisy vowel signal with $F_1 = 300$ Hz.

Hz. We have already explained why the first plot has a peak at 300 Hz. Interestingly, when we look at the second plot, even though the other set of nerve fibers are centered 100 Hz off the vowels first formant, they are still able to phase lock to that particular frequency. This is because of the fact that they have a finite bandwidth associated with their reception of acoustic stimuli.

However, the most crucial difference lies within the magnitudes of the peaks observed in the phase locked frequency for the two sets of nerve fibers. As the figure clearly shows, the peak magnitude for the second plot is less than the first plot indicating a weaker degree of synchrony or phase locking for nerve fibers centered further apart from the dominant frequency in the input stimuli. This result immediately leads to a new set of features defined as follows: "highest spectral magnitude peak value of the inter-spike time interval histogram at a non-zero frequency for each channel", which we will call the degrees of synchrony (DoS). Our experiments showed that, independent of the spike-based classifier operator, this novel feature set is extremely robust to noise, and for a particular vowel it remains unaffected by the change in noise type or intensity.

## 4. Spike-based operators

After the introduction of neural networks, the research on bio-inspired algorithms led to several spike-based operators that proved to be competitive in real world applications. Two most striking examples are rank order coding (ROC) and the liquid state machine (LSM) (Thorpe and Gautrais, 1998; Delorme and Thorpe, 2001; Maass *et al.*, 2002).

**4.1. Rank order coding.** ROC can be considered a special type of temporal coding where the information is carried in the order of spike arrival to the post-synaptic neuron. This is a great simplification which gets rid of all the precise timing information yet can still encode complicated information depending on the number of presynaptic neurons. Recent experimental studies on the auditory system of cats and somatosensory system of humans show that ROC might be responsible for coding sensory information with only one spike per neuron (Rullen *et al.*, 2005).

As is evident from the algorithm description, the response times are much shorter when compared with other alternatives such as rate coding and are not affected by the input intensity considering the presynaptic neurons will simply fire faster without changing the order of firing. These properties render the algorithm very useful for image processing applications (Rullen *et al.*, 1998; Delorme and Thorpe, 2001).

Training a system using this type of coding is extremely simple and fast, and slightly resembles that of a multi-layer perceptron. Figure 6 shows a simple block diagram for a classification example where there are $L$ number of classes and $L$ corresponding decoder neurons. The activation level of a decoding (post-synaptic) neuron $i$ at time $t$ is given as follows:

$$Activation(i,t) = \sum_{j=1}^{m} k^{order}(j) \times w_{j,i}.$$

$Order(j)$ is the firing order of the presynaptic neuron $j$ and $k$ is chosen to be any number between 0 and 1. $w_{j,i}$ is the synaptic weight between the decoding neuron $i$ and the presynaptic neuron $j$. The decoding neuron will simply fire when its activation energy reaches a certain threshold:

$$Activation(i,t) > Threshold(i).$$

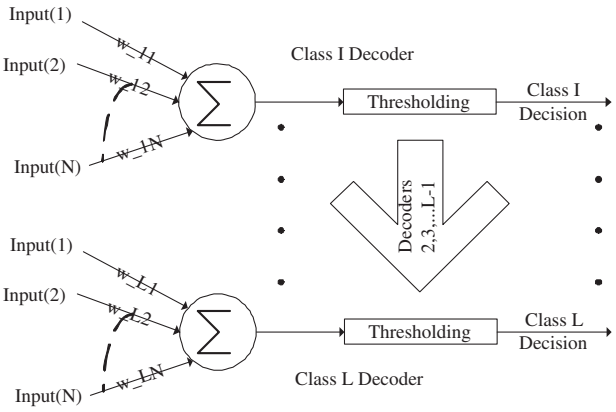


Fig. 6. Basic structure of a rank order decoder for $L$ different number of classes and $N$ different number of presynaptic neurons.

During training, the only updated variable is the synaptic weight between the pre- and postsynaptic neurons:

$$\Delta w_{j,i} = \frac{k^{order(j)}}{N},$$

where *N* is simply the number of training samples. Hence, at the end of training, for a particular decoder neuron, the neurons that fire faster will have stronger synaptic connections which will enable the activation energy to reach its threshold when the right test sample order arrives at the postsynaptic neuron. Thus, in a setup as in Fig. 6, when Class 1 is presented, since the order of firing will correspond best to the presynaptic weights for the first decoder neuron, it will reach its threshold and fire a spike signaling the detection of Class 1.

The synaptic weight value at the end of the training phase will be proportional to the mean modulation of the synapse. Hence, in case one knows the mean modulation for the training data samples belonging to a specific class, just one sample which is closer to the mean modulation can be used for training, which results in an exceptionally fast training phase, very much unlike in the case of the conventional machine learning engines.

For speech, without some type of pre-coding enforcement, it is not possible to classify signals with just the order of firing among afferent auditory neurons. The reason is that the neurons centered at higher frequencies will always fire later than those centered at lower frequencies, no matter what the input stimuli is.

**4.2. Liquid state machine.** Both Maass and Jaeger independently introduced the concept of using transients inherent in high dimensional dynamic structures to perform machine learning computations in the form of liquid state machines and echo state networks, respectively (Maass *et al.*, 2002; Jaeger, 2001). The overall structure for both

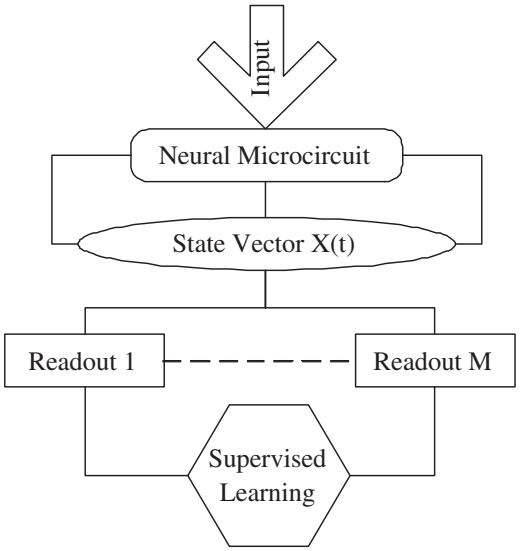


Fig. 7. Basic structure of a liquid state machine with $M$ readouts trained using supervised learning.

of these approaches is quite similar to a multi-layer perceptron except the fact that there are recurrent connections within the reservoir/neural microcircuit and the weights corresponding to these connections are not trained. Only the output weights that are used to extract readouts from the state of the reservoir/neural microcircuit are trained using supervised learning techniques. A major difference between the two architectures is the fact that the LSM uses spikes for computation and thus is more suitable for the algorithm discussed in the paper.

Figure 7 shows the basic structure of a typical LSM. The neural microcircuit is a network of spiking neurons randomly connected to each other with a desired ratio of inhibitory and excitatory synapses. The input is supplied to this network via analog or spiking synapses and the state of the neural circuit is recorded over time. One can define this state to be an internal variable and for a spike-based classification problem it can be chosen as the low-pass filtered spike trains generated by the individual elements of the neural microcircuit,

$$\vec{X}(t) = NMC(\vec{u}(t)).$$

Here the state vector is defined by the liquid filter NMC, operating on the input vector *u(t)*. For a particular classification task, the last step is to map the state vector to the desired output via memoryless readout functions:

$$\vec{y}_{desired}(t) = F_{1:L}(\vec{X}(t)),$$

assuming $M$ different classes in this example. We have chosen to use a feed-forward multi-layer neural network to train the readout functions. The next section will discuss the different system architectures combining the coding schemes and the spike-based operators in greater detail.

## 5. Hybrid system architectures and parameters

With the possibility of using two different spike-based operators and three spike coding schemes, there are six different possible architectural combinations. This section will discuss four of the more competitive combinations under different types and intensities of noise and different SPLs.

Both spike-based operators have different advantages. While ROC is a very simple and efficient technique which requires very few training data, an LSM with supervised learning is more complex and suitable for better generalization of the proposed algorithm to different classification tasks.

**5.1. ROC-Synchrony Coding Architecture (ROC–DoS).** This architecture uses ROC as its spike-based operator and encodes the spike information using synchrony coding. The degree of synchrony is detected for each of the 20 channels using all 50 nerve fibers per channel and a corresponding order of firing is assumed amongst the 20 presynaptic neurons (one for each channel) with channels having higher degrees of synchrony firing their first spike faster. Hence, the overall architecture can be thought of as 20 presynaptic connections per decoding neuron with the first spike firing times depending on the degree of synchrony at each channel.

**5.2. LSM-Rate, Direct Temporal and Synchrony Coding Architecture (LSM-RC, LSM-DoS, LSM-DTC).** These architectures use an LSM as the common spike-based operator and encode the spike information using all the three coding schemes possible: rate, synchrony and direct temporal coding. The way the LSM is applicable to all coding schemes makes it possible to see the degree of correlation between the performances of coding schemes and the type/intensity of noise and acoustic stimuli.

As an example, the architecture which uses an LSM and the degree of synchrony as its feature set is shown in Fig. 8.

Since the DoS feature set is an analog vector, it is supplied to the neural microcircuit using analog synapses modeling membrane potentials. The rate coding structure is similar in the sense that, instead of the degree of synchrony, the frequency of spike firing per channel is fed into the neural microcircuit with similar analog synapses. On the other hand, the direct temporal code uses spiking input synapses as it consists of spike trains at each channel, rather than feature vectors as in the case of synchrony and rate coding. Since there are a total of $50 \times 20 = 1000$ spike trains, only 10 nerve fibers per channel will be used for direct temporal coding to simplify computation.

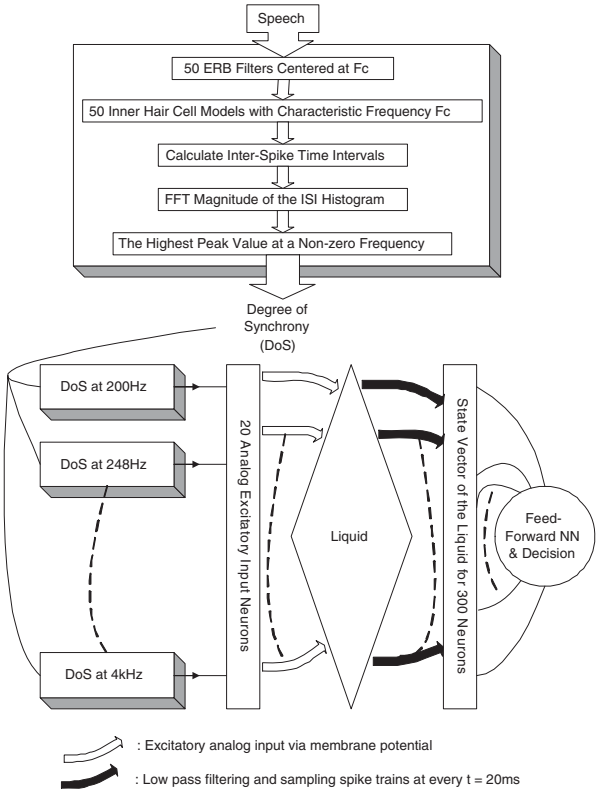


Fig. 8. Block diagram for an LSM-synchrony coding architecture.

The LSM structure and parameters used for all combinations are the same. The neural microcircuit is chosen to have 300 leaky integrate-and-fire neurons, 20% of which are chosen to be inhibitory. The local recurrent connections are modeled using a Gaussian distribution resulting in denser connections amongst neighboring neurons. All connections used in the microcircuit are dynamic spiking synapses with spike timing-dependent plasticity (Markram *et al.*, 1997).

During the training phase, the first state vector is obtained via low-pass filtering the spike outputs of the 300 neurons in the microcircuit with a 300 Hz cut-off frequency. The state vector is then sampled at every 20 ms and associated with a class label corresponding to the input signal. This generates input-desired output pairs to be used to train a single hidden layer feed-forward neural network with the well known back-propagation algorithm. The neural network uses a tangential sigmoid output which is quantized by the number of available input classes and is used to approximate the memoryless readout function mapping the state of the microcircuit to the desired class label.

## 6. Classification problem and test results

As an initial step to gauge the possibility of using spike-based architectures for speech recognition, we have cho-

sen a 5-class vowel classification problem as a performance measure. Table 1 shows five most commonly used vowels in the English language, which are the vowels that the systems will try to classify.

Table 1. Mean formant frequencies for common English vowels.

| *Vowel* | $F_1$ (Hz) | $F_2$ (Hz) | $F_3$ (Hz) |
|---|---|---|---|
| /iy/ "beet" | 270 | 2290 | 3010 |
| /ae/ "bat" | 660 | 1720 | 2410 |
| /aa/ "hot" | 730 | 1090 | 2240 |
| /ao/ "bought" | 570 | 840 | 2410 |
| /uh/ "foot" | 440 | 1020 | 2240 |

The tests will be done for all the four architectural combinations discussed in Section 5 using two different types of noise: white and pink; and three different SNR levels: 25 dB, 15 dB, 5 dB. Also, in order to see the viability of the three spike-coding schemes for different SPLs, LSM-rate, direct temporal and synchrony coding structures will also be tested at two levels of sound pressure as well: 60 dB and 10 dB.

**6.1. Test settings and results.** It is important to note that the proposed system architectures do not yet target the state-of-the-art ASR systems built with years of knowledge, operating on words/sentences with the help of elaborate language models. However, the final algorithm will still be compared to the conventional ASR duo of an MFCC-Hidden Markov Model (HMM).

200 training and 200 testing utterances are chosen for each of the vowels from the multi-speaker, multi-gender TIMIT database. White and pink noise signals are added with different SNR values and at different SPLs.

Table 2 shows the performance of the architecture using a degree of synchrony and rank order coding at a 60 dB SPL. As can be observed from the table, for both noise types, even at a 5 dB SNR, the performance is roughly the same as a 25 dB SNR, which shows the robustness of the degree of synchrony feature set to noise.

Rate coding being unreliable at high SPL levels, the same experiment is performed at a 10 dB SPL using the LSM as the spike-based operator. Table 3 shows the results for pink noise (white noise being very similar) for RC, DoS and DTC. At the lowest SNR value, RC manages to outperform the other coding schemes. This is mainly due to the fact that there are just not enough spike occurrences on nerve fibers to yield reliable timing information necessary for DoS and DTC. Thus, the results in Table 3 indicate that rate coding might be preferred over temporal coding at low SPLs.

The third test compares all algorithms for a 60dB SPL using the LSM as the spike-based operator again. Table 4 shows the results for pink noise, and DoS is superior especially at very low SNR levels due to its robustness to noise explained in the previous chapters.

Table 2. Percentage of vowels correctly classified for ROC-synchrony coding.

| Noise \ SNR [dB] | 25 | 10 | 5 |
|---|---|---|---|
| Pink Noise | 80.6% | 80.2% | 79.9% |
| White Noise | 79.8% | 78.9% | 77.5% |

Table 3. Percentage of vowels correctly classified at a 10 dB SPL for LSM-rate, direct temporal and synchrony coding.

| Code \ SNR [dB] | 25 | 10 | 5 |
|---|---|---|---|
| RC | 77.9% | 74.2% | 63.0% |
| DTC | 77.8% | 72.0% | 59.8% |
| DoS | 76.2% | 71.6% | 58.4% |

These tests support the theory of duplex spike coding in the auditory system depending on the intensity and noise level of the input acoustic stimuli. At low SPLs, rate coding is preferred due to the linear change in spike firing rates with the input intensity, and the number of spikes needed for reliable DoS and DTC coding is non-existent. However, at high SPLs, due to nerve fibers saturating, RC can no longer be used and DoS and DTC take over. However, since DoS is extracted from a population of neurons, it proves to be more noise robust and its performance unaffected by the significant drops in SNR values.

Taking into account all the three tests, the first thing to note is the superiority and robustness of DoS regardless of the spike-based operator. With DoS as the choice of spike coding, for typical conversational SPLs, the LSM outperforms ROC, which makes up for it having faster training and processing times. Nevertheless, due to increased classification performance and the ability to generalize to different classification tasks, this paper chooses the DoS-LSM as the proposed architecture.

The final step is to compare the proposed feature set along with its operator with a conventional ASR system from an engineering point of view. Table 5 shows the comparison of the LSM-synchrony coding architecture and the MFCC-HMM (64 Gaussians, 1 state) engine with 13 MFCC coefficients and their first and second derivatives as the feature set (a common practice in ASR).

At high SNR levels, both algorithms perform comparably well, with the MFCC-HMM having slightly better classification. However, as the SNR gets lower, LSM-DoS starts to outperform the MFCC-HMM with differences as

Table 4. Percentage of vowels correctly classified at a 60 dB SPL for LSM-rate, direct temporal and synchrony coding.

| Code \ SNR [dB] | 25 | 10 | 5 |
|---|---|---|---|
| RC | 36.2% | 35.4% | 35.2% |
| DTC | 80.2% | 72.5% | 64.9% |
| DoS | 93.0% | 92.5% | 91.0% |

Table 5. Percentage of vowels correctly classified for LSM-synchrony coding and the MFCC-HMM engine.

| Code \ SNR [dB] | 25 | 10 | 5 |
|---|---|---|---|
| MFCC - White N. | 92.5% | 84.0% | 76.0% |
| DoS - White N. | 91.0% | 90.0% | 87.5% |
| MFCC - Pink N. | 94.0% | 88.0% | 78.5% |
| DoS - Pink N. | 93.0% | 92.5% | 91.0% |

much as 12.5% and 11.5% for pink and white noise, respectively. These results not only support the claim of the degree of synchrony being a noise-robust feature, but also indicate the possibility of using a fully spike-based architecture for common speech recognition tasks in the future.

## 7. Conclusions and discussion

Two major goals of this research are to have a competitive and fully spike-based speech processing algorithm with heavy inspiration from psychoacoustics and neuroscience and to develop a possible process description to account for simple acoustic classification mechanisms in the auditory system. As a first step towards this goal, various spike coding schemes, including a novel degree of synchrony metric as a feature set, were evaluated with different spike-based operators.

The findings not only support the duplex theory of spike coding in the auditory system, but also suggest a possible reason for its unparalleled robustness. From an engineering point of view, the algorithm using the proposed feature set with a liquid state machine operator manages to outperform the typical MFCC-HMM ASR engine on a noisy vowel dataset proving the robustness and information potential of synchrony coding along with spike-based computation.

Future work includes the design of a spike-based extractor for measuring the degree of synchrony among nerve fibers. The next important step towards a fully featured ASR engine is the integration of a top-down hierarchical network to account for multi-syllable words. The authors believe that the results presented in this paper will pave the way to an alternative way of information processing for speech signals.

## References


Atal B. S. and Hanauer S. L. (1971). Speech analysis and synthesis by linear prediction, *Journal of the Acoustical Society of America* **50**(2B): 637–655.

Davis S. B. and Mermelstein P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Transactions on Acoustics, Speech, Signal Processing* **28**(4): 357–366.

Dayan P. and Abbott L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, Cambridge, MA.

Delorme A. and Thorpe S. J. (2001). Face identification using one spike per neuron: resistance to image degradations, *Neural Networks* **14**(7): 795–803.

Hopfield J. J. and Brody C. D. (2001). What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration, *Proceedings of the National Academy of Sciences USA* **98**(3): 1282–1287.

Jaeger H. (2001). The "echo state" approach to analysing and training recurrent neural networks, *Technical Report GMD Report 148*, German National Research Center for Information Technology.

Maass W. Natschlager T. and Markram H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Computation* **14**(11): 2531–2560.

Markram H., Lubke J., Frotscher M. and Sakmann B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs, *Science* **275**(5297): 213–215.

Meddis R. (1986). Simulation of mechanical to neural transduction in the auditory receptor, *Journal of the Acoustical Society of America* **79**(3): 702–711.

Moissl U. and Meyer-Base U. (2000). A comparison of different methods to assess phase-locking in auditory neurons, *International Conference of IEEE-EMBS on Information Technology Applications in Biomedicine*, Vol. 2, Arlington, USA, pp. 840–843.

Rieke F., Warland D., de Ruyter can Steveninck R. and Bialek W. (1999). *Spikes — Exploring the Neural Code*, MIT Press, Cambridge, MA.

Rullen R. V., Gautrais J., Delorme A. and Thorpe S. J. (1998). Face processing using one spike per neuron, *Biosystems* **48**(1–3): 229–239.

Rullen R. V., Guyonneau R. and Thorpe S. J. (2005). Spike times make sense, *Trends in Neurosciences* **28**(1): 1–4.

Sachs M. B. (1984). Neural coding of complex sounds: Speech, *Annual Review of Physiology* **46**: 261–273.

Skowronski M. D. and Harris J. G. (2004). Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition, *Journal of the Acoustical Society of America* **116**(3): 1774–1780.

Sumner C. J. and Lopez-Poveda E. A. (2002). A revised model of the inner-hair cell and auditory-nerve complex, *Journal of the Acoustical Society of America* **111**(5): 2178–2188.

Sumner C. J., Lopez-Poveda E. A., O'Mard L. P. and Meddis R. (2003). Adaptation in a revised inner-hair cell model, *Journal of the Acoustical Society of America* **113**(2): 893–901.

Terman D. and Wang D. (1995). Global competition and local cooperation in a network of neural oscillators, *Physica D.* **81**(1–2): 148–176.

Thorpe S. J. and Gautrais J. (1998). Rank order coding, *in* J. Bower (ed.), *Computational Neuroscience: Trends in Research*, New York: Plenum Press, pp. 113–119.

Uysal I., Sathyendra H. and Harris J. G. (2006). A biologically plausible system approach for noise robust vowel recognition, *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, Vol. 1, San Juan, Puerto Rico, pp. 245–249.

Uysal I., Sathyendra H. and Harris J. G. (2007a). A duplex theory of spike coding in the early stages of the auditory system, *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Vol. 4, Honolulu, USA, pp. 733-736.

Uysal I., Sathyendra H. and Harris J. G. (2007b). Spike-based feature extraction for noise robust speech recognition using phase synchrony coding, *Proceedings of the IEEE International Symposiom on Circuits and Systems*, New Orleans, USA, pp. 1529–1532.

VanRullen R., Guyonneau R. and Thorpe S. J. (2005). Spike times make sense, *Trends in Neurosciences* **28**(1): 1–4.

Verstraeten D., Schrauwen B., Stroobandt D. and Campenhout J. V. (2005). Isolated word recognition with the liquid state machine: A case study, *Information Processing Letters* **95**(6): 521–528.

Received: 17 April 2007
Revised: 6 September 2007