

---

# Litterature summaries

---

## Contents

<b>1</b>	<b>Spiking Neurons</b>	<b>1</b>
1.1	Simple model of spiking neurons [?]	1
<b>2</b>	<b>Spike-based Synapses</b>	<b>1</b>
2.1	Spike-timing-dependent plasticity (STDP) [?]	1
2.2	Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning [?]	2
2.3	ReSuMe - new supervised learning method for Spiking Neural Networks [?]	2
2.4	Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks [?]	2
2.5	Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting [?]	2
2.6	Apparatus and methods for gating analog and spiking signals in artificial neural networks [?]	3
2.7	Gradient Descent for Spiking Neural Network [?]	4
2.8	Comparison of supervised learning methods for spike time coding in spiking neural networks [?]	4
2.9	Temptron-like learning with ReSuMe [?]	4
2.10	Span: Spike pattern association neuron for learning spatio-temporal spike patterns [?]	4
2.11	Training spiking neural networks to associate spatio-temporal input-output spike patterns [?]	5
2.12	Learning precisely timed spikes [?]	5
2.13	Delay learning and polychronization for reservoir computing [?]	5
<b>3</b>	<b>Spiking microcircuits</b>	<b>6</b>
3.1	Real-time computing without stable states: A new framework for neural computation based on perturbations [?]	6
3.2	Polychronization: Computation with spikes [?]	7
3.3	Generating coherent patterns of activity from chaotic neural networks [?]	7
3.4	Supervised learning in spiking neural networks with FORCE training [?]	9
3.5	Reservoir computing trends [?]	10
3.6	Design of the spiking neuron having learning capabilities based on FPGA circuits [?]	10
3.7	Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication [?]	10
<b>4</b>	<b>Speech synthesis</b>	<b>11</b>

4.1	A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation [?]	11
4.2	Phonetic feature encoding in human superior temporal gyrus [?]	11
4.3	The control of vocal pitch in human laryngeal motor cortex [?]	11
4.4	Intonation Modelling for Speech Synthesis and Emphasis Preservation [?]	13
4.5	Research plan (P.N. Garner)	14
<b>5</b>	<b>Other</b>	<b>14</b>
5.1	Software for brain network simulations: a comparative study [?]	14
5.2	Rate coding versus temporal order coding: a theoretical approach [?]	14

# 1 Spiking Neurons

## 1.1 Simple model of spiking neurons [?]

**Abstract :** A model is presented that reproduces spiking and bursting behavior of known types of cortical neurons. The model combines the biological plausibility of Hodgkin-Huxley-type dynamics and the computational efficiency of integrate-and-fire neurons. Using this model, one can simulate tens of thousands of spiking cortical neurons in real time (1 ms resolution) using a desktop PC.

**Equation :**

$$\dot{v} = 0.04v^2 + 5v + 140 - u + I \quad (1)$$

$$\dot{u} = a(bv - u) \quad (2)$$

with the auxiliary after-spike resetting

$$\text{if } v \geq 30 \text{ mV, then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (3)$$

$v$  represents the membrane potential of the neuron and  $u$  represents a membrane recovery variable.  $a$  and  $b$  describe respectively the time scale and the sensitivity of the recovery variable  $u$ .  $c$  and  $d$  describes the after-spike reset value respectively of the membrane potential  $v$  and the recovery variable  $u$ .

Typical values are :  $a = 0.02$   $b = 0.2$ ,  $c = -65 \text{ mV}$ ,  $d = 2$

## 2 Spike-based Synapses

### 2.1 Spike-timing-dependent plasticity (STDP) [?]

The synapse is the biological element present in between two connected neurons called presynaptic and postsynaptic neurons. The synapse regulates the amount of current from the presynaptic neuron that is transmitted to the postsynaptic neuron. The presynaptic current can be both reduced or amplified by the synapse and then injected in the postsynaptic neuron. The synaptic plasticity includes the mechanisms that involve changes of the synaptic weight : the factor from presynaptic current to postsynaptic current. Hebbian learning is a general mathematical formulation of Hebb's postulate based on two particular aspects : the change of the synaptic efficacy depend only on (spatially) local variables and depend on the joint activity of pre- and postsynaptic neurons. Numerous learning rules have been derived from rate-based models of synaptic plasticity. Pair-based models of STDP consider the temporal difference between pre- and postsynaptic spikes for the change in weight of a synapse. If we introduce  $S_j = \sum_f \delta(t - t_j^f)$  and  $S_i = \sum_f \delta(t - t_i^f)$  for the spike trains of pre- and postsynaptic neurons respectively, the update rule is :

$$\frac{dw_{ij}}{dt} = -A_-(w_{ij})e^{-\frac{|\Delta t|}{\tau_-}} S_j + A_+(w_{ij})e^{-\frac{|\Delta t|}{\tau_+}} S_i \quad (4)$$

where  $A_{\pm}(w)$  describes the dependence of the update on the current weight of the synapse,  $\Delta t = |t_{post} - t_{pre}|$  the temporal difference and  $\tau_{\pm}$  the time constant of the exponential decay.

Figure 1: ReSuMe learning rule

## 2.2 Optimal spike-timing dependent plasticity for precise action potential firing in supervised learning [?]

Supervised learning paradigm to derive a synaptic update rule that optimizes by gradient ascent the likelihood of postsynaptic firing at one or several desired firing times. Define the probability density of an entire spike train. It allows to describe explicitly the likelihood of emitting a set of spikes. Since the likelihood is a smooth function (density dist.) it is differentiable with respect to any variable. Thus learning by gradient ascent.

Strong mathematical proof. Relation to STDP. Only single spike, or limited to few spikes.

## 2.3 ReSuMe - new supervised learning method for Spiking Neural Networks [?]

Original paper, presenting ReSuMe learning rule (first time) for movement control. Combined with a Liquid State Machine. See figure 1

## 2.4 Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks [?]

Experimental demonstration that ReSuMe has the following properties:

1. it can learn temporal sequences of spikes
2. it can learn model object's I/O properties
3. it is scalable
4. it is computationally simple
5. it is fast converging

See figure 1

## 2.5 Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting [?]

Robust theory describing ReSuMe (Remote Supervised Method) learning rule. **how do neurons learn to reproduce template signals encoded in precisely timed sequence of spikes?**

*Supervised learning is believed to be utilized by the neural motor centres to form the internal representation of the body and the environment (cerebellum and cerebellar cortex)*

Figure 2: ReSuMe learning rule

*It (ReSuMe) is based on the interaction between two spike-timing dependent plasticity (STDP) processes. It appears that ReSuMe represents a spiking analogy to the classical Widrow-Hoff algorithm proposed for the rate-based neuron model (Widrow and Hoff, 1960; Widrow, 1962)*

*It has been demonstrated that ReSuMe enables effective learning of complex temporal and spatio-temporal firing patterns with an arbitrary high accuracy*

*Due to the ability of the method to operate online and due to its fast convergence the method is suitable for real-life applications. [...] SNN trained with ReSuMe become efficient neurocontrollers for movement generation and control*

Widrow-Hoff learning rule (weight change  $\Delta w_{oi}$ ):

$$\Delta w_{oi} = \alpha x_i (y_d - y_o) \quad (5)$$

$\alpha$ : learning rate,  $x_i$ : input signal,  $y_o$ : output signal,  $y_d$ : teaching signal

ReSuMe learning rule (weight change  $\frac{d}{dt}w_{oi}(t)$ ):

$$\frac{d}{dt}w_{oi}(t) = [S_d(t) - S_o(t)] \left[ a_d + \int_0^\infty a_{di}(s) S_i(t-s) ds \right] \quad (6)$$

$S_i(t)$ ,  $S_o(t)$  and  $S_d(t)$ : presynaptic, postsynaptic and target spike trains. The role of the non-correlative factor (Non-hebbian term)  $a_d$  is to adjust the average strength of the synaptic inputs so to impose on a neuron a desired level of activity such that the actual mean firing rate of  $S_o(t)$  approaches the mean firing rate of signal  $S_d(t)$ . Thus the task of setting up the precise timing of spikes, is attributed mainly to the Hebbian term  $a_{di}(s)$ . The kernel  $a_{di}(s)$  define the shape of a learning window:

$$a_{di}(s) = A_{di} \exp\left(-\frac{s}{\tau_{di}}\right) \quad (7)$$

1. Learning: sequence of spikes / with various neuron models / in the presence of noise and uncertainty
2. Classification
3. Spike shifting (by setting different parameters)

(The main advantage of VLSI (Maass and Bishop, 1999) systems over processor-based designs lies in their truly parallel mode of operation which fully utilizes the essential property of parallelism found in spiking neural networks.) Advantages: simplicity, scalability, online training suitability, and fast convergence.

See figure 2

## 2.6 Apparatus and methods for gating analog and spiking signals in artificial neural networks [?]

Universal node (neuron) design implementing a universal learning rule in a mixed (analog and spiking) signal spiking network.

Figure 3: Comparison of supervised learning methods for SNN

## 2.7 Gradient Descent for Spiking Neural Network [?]

Introduce a differentiable synaptic model, which is based on the presynaptic membrane potential. The synaptic current is thus activated gradually throughout the active zone (margin below the threshold potential), this allows to compute a gradient. Uses a regularizer. Predictive coding task, auto-encoder task, delayed-memory XOR task.

## 2.8 Comparison of supervised learning methods for spike time coding in spiking neural networks [?]

See figure 3

## 2.9 Tempotron-like learning with ReSuMe [?]

description of the tempotron as a special case of ReSuMe learning rule. Classification task : categorize spike trains not only by firing or not, but also by firing given spike trains.

*Despite its limitations, the tempotron has been proved quite efficient for spoken digit recognition, outperforming with only 15 spiking neurons complex state-of-the-art HMM word recognition systems (Gutig and Sompolinsky, 2008). This shows that spiking neural networks are quite powerful, and using appropriate learning methods for training them might reveal even more of their potential*

## 2.10 Span: Spike pattern association neuron for learning spatio-temporal spike patterns [?]

Transform spike trains during the learning phase into analog signals (filtered spike trains), based on Widrow-Hoff rule. Compared with ReSuMe and Chronotron.

*Some pattern recognition tasks such as the recognition of colors, visual patterns, odors and sound qualities can not be easily solved by rate-based neural models. Temporal information encoding can also reduce the number of neurons that are necessary to perform a given task.*

*SNN has proved itself adequate for a number of computation and engineering problems. It is considered as a suitable tool to perform temporal pattern recognition and real-time computation*

Learning rule  $\Delta w_i$ :

1. with  $\alpha$ -kernel:  $\tilde{s}(t) = \sum_{t^f} e\tau^{-1}(t - t^f)e^{-\frac{(t-t^f)}{\tau}}H(t - t^f)$

$$\Delta w_i = \lambda\left(\frac{e}{2}\right)^2 \left[ \sum_g \sum_f (|t_i^f - t_d^g| + \tau)e^{-\frac{|t_i^f - t_d^g|}{\tau}} - \sum_h \sum_f (|t_i^f - t_a^h| + \tau)e^{-\frac{|t_i^f - t_a^h|}{\tau}} \right] \quad (8)$$

fig/SPAN1.png

Figure 4: SPAN learning rule

2. with exponential kernel:  $\tilde{s}(t - t^f) = H(t - t^f)e^{-\frac{t-t^f}{\tau}}$

$$\delta w_i = \frac{\tau\lambda}{2} \left[ \sum_g \sum_f e^{-\frac{|t_d^g - t_i^f|}{\tau}} - \sum_h \sum_f e^{-\frac{|t_a^h - t_i^f|}{\tau}} \right] \quad (9)$$

Illustration of the learning rule on figure 4

The paper explores: Output spike sequence learning, learning with noise, memory capacity (number of patterns memorized depending on the number of synapses), classification.

## 2.11 Training spiking neural networks to associate spatio-temporal input-output spike patterns [?]

Very similar to [?]. Application to various methods of classification.

## 2.12 Learning precisely timed spikes [?]

**Summary** Capacity of feedforward networks to generate desired spike sequences : we find the maximum number of desired output spikes a neuron can implement to 0.1-0.3 per synapses. We present a biologically plausible learning rule that allows feedforward and recurrent networks to learn multiple mappings between inputs and desired spike sequences.

**Introduction** Here we characterize the capacity of spiking neurons to implement desired transformations between input and output spike patterns. We evaluate the maximum number and length of mappings that can be implemented by a neuron and describe its dependence on neuronal time constants and input and output firing statistics.

Here we present a simple, efficient, and biologically plausible neuronal learning algorithm capable of training neurons to generate desired spike trains with a specified temporal tolerance, in response to their associated inputs. Three challenging problems:

1. a data analysis tool for reconstructing synaptic connections from observed spike patterns.
2. temporal information about an ongoing vocal behavior embedded in the spiking patterns of the songbird motor cortex and to model the decoding of this information by downstream neurons.
3. learning of multiple stable, precisely timed patterns of spikes in networks with recurrent topology.

## 2.13 Delay learning and polychronization for reservoir computing [?]

A network of spiking neurons, sparsely connected, without pre-imposed topology, and output neurons, with adaptable connections. Our learning rule for readout neurons is justified by the appealing notion

of polychronization. Talk about intrinsic plasticity (IP) and STDP. A current trend is to propose computational justifications for plasticity-based learning rules, in terms of entropy minimization as well as log-likelihood or mutual information maximization.

Our multi-timescale learning rule for RC comprises STDP, modifying the weights inside the reservoir, and a supervised adaptation of axonal transmission delays toward readout neurons coding, via their times of spike firing, for different classes. Similarity between RC and SVM.

Network similar to Izhikevich network, neuron model SRM (Gerstner)

Delay adaptation algorithm minimizes:

$$C = \sum_{p \in \text{class1}} |t_1(p) - t_2(p) + \mu|_+ + \sum_{p \in \text{class2}} |t_2(p) - t_1(p) + \mu|_+ \quad (10)$$

This learning rule seems classification oriented (with the margin  $\mu$ )

### 3 Spiking microcircuits

#### 3.1 Real-time computing without stable states: A new framework for neural computation based on perturbations [?]

- alternative to paradigms based on Turing machines or attractor neural networks
- high-dimensional dynamical system (task-independent)
- universal analog fading memory and universal computational power
- readouts neurons can learn to extract in real-time from the current neural circuit information about current and past inputs
- from observation of neocortex's circuits: extremely complex but surprisingly stereotopic microcircuits that can perform a wide spectrum of tasks
- multiple readouts can be trained to perform different tasks on the same state trajectory of a recurrent neural circuit  $\rightarrow$  parallel real-time computing.

The liquid has only one attractor state: the resting state. Perturbed state of the liquid represents present and past inputs  $\rightarrow$  potentially providing the information needed for an analysis of various dynamic aspects of the environment.

1. **Separation property (SP)**: amount of separation between the trajectories of internal states of the system that are caused by two different input streams.
2. **Approximation property (AP)**: resolution and recoding capabilities of the readout mechanisms (capability to distinguish and transform different internal states of the liquid into given target outputs)

The liquid state  $x^M(t)$  is simply the current output of some operator or filter  $L^M$  (called liquid filter or liquid circuit) that maps input function  $u(\cdot)$  onto functions  $x^M(t)$ :

$$x^M(t) = (L^M u)(t) \quad (11)$$



The second component of an LSM  $M$  is a memoryless readout map  $f^M$  that transforms, at every time  $t$ , the current liquid state  $x^M(t)$  into the output:

$$y(t) = f^M(x^M(t)) \quad (12)$$

The liquid state  $x^M(t)$  can be defined as the vector of output values at time  $t$  of linear filters with exponential filters with exponential decay (time constant equal to  $\tau_m$ ) applied to the spike trains emitted by the liquid neurons.

## 3.2 Polychronization: Computation with spikes [?]

Spiking neural network with STDP synaptic weights evolution (1000 Izhikevich neurons). It describes the different rhythms observed, the balance of excitation and inhibition, the rate to spike-timing conversion.

The **originality of the paper** is the consideration of a large range of synaptic delays, it enables the emergence of polychronous groups (PG), subsets of interconnected neurons activated by a precise spike timing sequence.

PGs appear in the the context of gamma frequencies. the number of PGs far exceeds the number of neurons in the network, resulting in a large memory capacity of the system.

The paper provides a way to detect PGs from an anatomical point of view.

Needs 24h for STDP synapses to converge.

*Twofolds changes of some parameters, such as maximal synaptic weight, the amount of depression in STDP, or the thalamic input, produce only transient changes in network dynamics. Neurons adjust their synaptic weights, balance excitation and inhibition, and return to the mean firing rate (btw 2 and 7 Hz)*

*Since STDP is always ON in the network, groups constantly appear and disappear*

*Not all groups corresponding to a pattern activate when the network is stimulated. Because the groups share some neurons and have exc. and inh. interconnections, they are in a constant state of competition and cooperation. As a result, each p[re]sentation of a stimulus activates only two to three groups (15 per cent) in a random manner.*

*Polychronous groups: strongly interconneconnected groups of neurons having matching conduction delays and capable of firing stereotypical time-locked spikes with millisecond precision.*

## 3.3 Generating coherent patterns of activity from chaotic neural networks [?]

**Abstract** *We develop a procedure called FORCE learning for modifying synaptic strengths either external to or within a model neural network to change chaotic spontaneous activity into a wide variety of desired activity patterns (in particular, input-output transformations that require memory). Also, Our results [...] suggest that synaptic plasticity may be a more rapid and powerful modulator of network activity than generally appreciated.*

**Introduction** They show how external feedback loops or internal synaptic modifications can be used to alter the chaotic activity of a recurrently connected neural network and generate complex but controlled outputs.

Three problems to solve :

1. Feeding erroneous output back into a network during training can cause its activity to deviate so far from what is needed that learning fails to converge. They show how the synaptic modification procedure itself can be used to control the feedback signal, without any mechanism being required, in a manner that allows fluctuations to be sampled and stabilized.
2. Credit assignement for output errors. Credit assignement amounts to figure out which neurons and synapses are most responsible for output errors and therefor most in need of modification. First solution: restricting modification solely to synapses directly driving network output. Second solution: FORCE learning allows to train network in which modifications are not restricted to network outputs.
3. Training in the face of chaotic spontaneous activity. The solution for learning in a recurrent network and for suppressing chaos turn out to be one and the same: synaptic modifications must be strong and rapid during the initial phases of training.

**Results** Learning cannot be accomplished in a network receiving no external input without including a feedback loop. The strength of the feedback synapses is of order 1 whereas that of synapses between neurons of the recurrent network is of order 1 over the square root of the number of recurrent synapses per neuron. The feedback synapses are made stronger so that the feedback pathway has an appreciable effect on the activity of the recurrent network.

As long as output errors are small enough, they can be fed back without disrupting learning, i.e., without introducing significant delayed, reverberating effects. Small differences between the actual and desired output of the network during training allow the learning procedur to sample instabilities in the recurrent network and stabilize them.

Network output ( $t - \Delta t$  means before weight update):  $\mathbf{w}^T(t - \Delta t)\mathbf{r}(t)$

Comparing output with the desired output  $f(t)$  before weight update:  $e_-(t) = \mathbf{w}^T(t - \Delta t)\mathbf{r}(t) - f(t)$

After the weight update:  $e_+(t) = \mathbf{w}^T(t)\mathbf{r}(t) - f(t)$

Recursive Least-Square (RLS) modification:

$$\mathbf{w}(t) = \mathbf{w}(t - \Delta t) - e_-(t)\mathbf{P}(t)\mathbf{r}(t) \quad (13)$$

$$\mathbf{P}(t) = \mathbf{P}(t - \Delta t) - \frac{\mathbf{P}(t - \Delta t)\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{P}(t - \Delta t)}{1 + \mathbf{r}^T(t)\mathbf{P}(t - \Delta t)\mathbf{r}(t)} \quad \mathbf{P}(0) = \frac{\mathbf{I}}{\alpha} \quad (14)$$

RLS rule satisfies the conditions for FORCE learning. The parameter  $\alpha$ , which acts as a learning rate, should be adjusted depending on the particular target function being learned. Small  $\alpha$  values result in fast learning but sometimes make weight changes so rapid that the algorithm becomes unstable. But if  $\alpha$  is too large, the force algorithm may not keep the output close to the target function for a long enough time, causing learning to fail. In practice, values from 1 to 100 are effective, depending on the task.

Provided that the feedback signal is of sufficient amplitude and frequency to induce a transition to a nonchaotic state, learning can take place in the absence of chaotic activity, even though the network is chaotic prior to learning and afterwards there may exist additional chaotic trajectories.

The progression of learning can be tracked by monitoring the size of the fluctuations in the readout weights, which diminish over time [...].

The activity of a network that has been modified by the FORCE procedure to produce a particular output can be analyzed by principal component analysis (PCA).

Advantages of chaotic spontaneous activity: faster training, lower error, lower magnitude of the weight vector.

Nonlinear distortions of the feedback signal can be introduced as long as they do not diminish the temporal fluctuations of the output to the point where chaos cannot be suppressed. Delays can be more problematic if they are too long. The feedback does not have to be identical for each neuron of the generator network.

To illustrate how FORCE learning works, we express the total current into each neuron of the generator network as the sum of two terms. One is the current produced by the original synaptic strength prior to learning,  $\sum_j \mathbf{J}_{ij}(0)\mathbf{r}_j(t)$  for neuron  $i$ . The other is the extra current generated by the learning-introduced changes in these synapses,  $\sum_j (\mathbf{J}_{ij}(t) - \mathbf{J}_{ij}(0))\mathbf{r}_j(t)$

### 3.4 Supervised learning in spiking neural networks with FORCE training [?]

To be read again...

Examples include the FORCE method, a novel technique that harnesses chaos to perform computations. We demonstrate the direct applicability of FORCE training to spiking neurons by training networks to mimic various dynamical systems in addition to reproduce more elaborate tasks such as input classification, storing sequences, reproducing the singing behavior of songbirds, and recalling a scene from a movie.

The synaptic weight matrix is the sum of a set of static weights and a set of dynamically varying weights. The time dependent weights are learned using a supervised learning method called Recursive Least Mean Squares (RLMS) so that the squared error between the target dynamics (i.e. the task,  $x(t)$ ) and the network dynamics ( $\hat{x}(t)$ ) is minimized. For the dynamics under consideration, the Izhikevich model had the greatest accuracy and fastest training times. This is partially due to the fact that the Izhikevich neuron has the spike frequency adaptation variable ( $u_i$ ) which operates on a longer time scale (i.e. 100 ms). We set the parameters such that the average firing rate was low ( $< 60$  Hz). To test whether a network of spiking neurons can function as a statistical classifier, we trained networks of Izhikevich neurons to classify inputs into two classes separated by either linear or nonlinear boundaries.

The weight matrix :

$$\omega_{ij} = G\omega_{ij}^0 + Q\eta_i\phi_j^T \quad (15)$$

$\omega_{ij}^0$  the static weight matrix that induces chaos,  $G$  chaos factor,  $Q$  factor increasing desired recurrent dynamics,  $\eta_i$  encoding variables,  $\phi_j$  decoding variables that are learned with recursive least mean squares:

$$\phi(t) = \phi(t - \Delta t) - e(t)P(t)r(t) \quad (16)$$

$e(t) = \hat{x}(t) - x(t)$  the error,  $P(t)$  estimate for the inverse of the correlation matrix (learning rule) and  $r(t)$  the firing rate.

### 3.5 Reservoir computing trends [?]

The reservoir can be seen as a non-linear high-dimensional expansion of the input signal. At the same time, the reservoir serves as a memory, providing the temporal context.

Even if the reservoir is kept fixed, for some tasks the trained readouts are fed back to the reservoir and thus the training process changes its dynamics. This extends the power of RC (RC no longer relies on fixed random input-driven dynamics, dynamics are adapted to the task), but this power has its price, because stability issues arise. Two strategies are used :

1. disengage the recurrent relationship between the reservoir and the readout using teacher forcing and treat output learning as a feedforward task. Feeding the desired output through the feedback connections instead of the real output while learning. After learning real outputs replace target outputs. The approach works well if the output can be learned precisely. Otherwise quick divergence. The problem can be alleviated by some kind of regularization of the weights or immunization of the state and/or feedbacks
2. FORCE learning uses a powerful 2nd-order online learning algorithm to vigorously adapt  $W^{out}$  in the presence of the real feedbacks. It appears that FORCE training is well suited to yield very stable and accurate neural pattern generators.

### 3.6 Design of the spiking neuron having learning capabilities based on FPGA circuits [?]

Hardware real-time implementations of Spiking Neuron Networks (SNN) with ReSuMe learning rule.

### 3.7 Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication [?]

Method for learning nonlinear systems, echo state networks. Computationally efficient, accuracy is improved by a factor of 2400.

Learning algorithms that incrementally adapt synaptic weights of an RNN have not been widely employed in technical applications because of slow convergence and suboptimal solutions. ESN approach differs ... only the synaptic connections from the RNN to the output readout are modified by learning.

The output neuron was equipped with random connections that project back into the reservoir.

ESNs have been developed from a mathematical and engineering perspective, but exhibit typical features of biological RNNs: a large number of neurons, recurrent pathways, sparse random connectivity, and local modification of synaptic weights.

## 4 Speech synthesis

### 4.1 A Neural Model to Predict Parameters for a Generalized Command Response Model of Intonation [?]

Bastian paper. Presents Generalised Command Response (GCR) model, which is extracted from an intonation contour by a matching pursuit algorithm. Constituted by atoms.

*GCR also enables analysis of the underlying physiological process (approximate muscle response).*

*Given that GCR approximate muscle responses to neural spikes, it would make sense to use a Spiking Neural Network to generate these atoms. The generated spikes would be filtered by muscle responses to generate the pitch contour.*

*We generalize the Spike Pattern Association Neuron (SPAN) algorithm from the litterature to construct a loss from GCR atoms, and show thatit can be backpropagated to allow the system to generate natural prosody.*

The proposed GCR model is a physiologically based intonation model which has the same representative power as the CR model of Fujisaki. It generate the LF0 contour by a superposition of impulse responses to critically damped second order systems modelling muscle responses.

The impulse response of a critically damped second order system is a gamma kernel :

$$G_{k,\theta}(t) = \frac{1}{\theta^k \Gamma(k)} t^{k-1} e^{-\frac{t}{\theta}} \quad (17)$$

a

$$\frac{d^2x}{dt^2} + \frac{2}{\theta} \frac{dx}{dt} + \frac{1}{\theta^2} x = 0 \quad \text{with} \quad x(0) = 0, \quad \frac{dx}{dt}(0) = \frac{1}{Z\theta^2} \quad (18)$$

a

$$\frac{dy}{dt} = -\frac{2}{\theta} y - \frac{1}{\theta^2} x, \quad \frac{dx}{dt} = y, \quad y += \frac{1}{Z\theta^2} \quad (19)$$

### 4.2 Phonetic feature encoding in human superior temporal gyrus [?]

The superior temporal gyrus (STG) participates in high-order auditory processing of speech, but how it encodes information is poorly understood. At single electrodes, we found response selectivity to distinct phonetic features. Phonetic features could be directly related to tuning for spectrotemporal acoustic cues, some of which were encoded in a nonlinear fashion or by integration of multiple cues. Acoustic-phonetic representation of speech in human STG.

We analysed high gamma (75-100 Hz) cortical field potential, which correlate with neuron spiking.

### 4.3 The control of vocal pitch in human laryngeal motor cortex [?]

Moulouation of vocal pitch creates intonation patterns that speakers use to convey linguistic meaning.

We found neural population in bilateral dorsal laryngeal motor cortex (dLMC) that selectively encoded produced pitch but not non-larygeal articulatory movements (control short speech accents). Other

larynx cortical representations controlling voicing and longer pitch phrase contours were found at separate sites. ... reveal the neural basis for the voluntary control of vocal pitch in human speech.

In speech, the two dominant functions of the larynx are to generate voicing and modulate speech. Voicing is created by bringing the vocal folds into close proximity, so they vibrate when air is passed through. In contrast, the pitch of the voice is modulated primarily by fine changes in the tension of the vocal folds. Greater tension in the vocal folds causes them to vibrate at higher frequency during voicing, producing a higher speech sound. (primarily mediated by flexing the cricothyroid muscle.

Two distinct regions in the human sensorimotor cortex that are correlated with laryngeal movements: the ventral laryngeal motor cortex (vLMC) and a completely separate dorsal premotor region (dLMC). Candidate representations include the control of larynx movements, their acoustic sensory goals, or even higher-order linguistic-level encoding of prosody.

We sought to determine:

1. whether there is separable encoding for functionally distinct pitch components
2. whether the same pitch control mechanisms are engaged during non-speech vocalization (singing melody)
3. whether dLMC activity reflects the causal, feedforward, and proportional control of laryngeal muscles.

**Pitch encoding in the dLMC during natural speech intonation:** High gamma range (70-150 Hz) which has been found to correlate with multi-unit neuronal firing rates and has been shown to reliably track neural activity associated with speech articulation and other movements. All pitch-encoding electrodes showed a positive monotonic relationship between neural activity and pitch.

**dLMC has both motor and auditory sensory representations:** we consider whether they represented sensory feedback versus motor feedforward commands. We found that dLMC electrodes that were correlated with pitch during speaking also had auditory responses that were correlated with pitch during listening, although with distinct response latencies. We found that most electrodes activated before acoustic onset when speaking, with a mean lead of 0.09 s, and that when listening, all electrodes responded after acoustic onset, with a mean lag of 0.39 s.

**dLMC, vLMC encoding of pitch components: accent, phrase and voicing.** We adapted a well-known mathematical formalization called the Fujisaki model. For each sentence, the components in the model include a fast 'accent' component (emphasized words or syllables), and a slow 'phrase' component (the declination in pitch over the course of phrase). We found a clear and striking dissociation between electrodes that encoded accent, phrase and voicing. A subset of dLMC electrodes were mostly strongly associated with pitch accent. In contrast, phrase encoding electrodes were found in the vLMC and dLMC. Finally, voicing was localized to a distinct subset of dLMC and vLMC electrodes. These results demonstrate a functional-anatomical distinction between the dorsal and ventral LMCs, as well as an evidence for independent and heterogeneous encoding for different pitch components within the dLMC.

dLMC activity reflects a task-independent representation of vocal pitch that is not specific to speech or singing, and may therefore reflect feed-forward control of specific laryngeal movements.

(+ direct stimulation) evidence that dLMC neural activity reflects feed-forward encoding of motor commands in the larynx, though they also suggest that the representation might be more complex than control of a single muscle or even the larynx alone.

Prosody is enabled by highly specialized sensorimotor neural control of laryngeal function in the human brain.

Dissociate accent, phrase and voicing at different discrete sites. demonstrate how multiple dimensions of vocal pitch can be independently controlled by the cortex.

## 4.4 Intonation Modelling for Speech Synthesis and Emphasis Preservation [?]

Statistical Parametric Speech Synthesis:

1. Source-Filter Model
2. HMM-based Speech Synthesis
3. Modelling Context
4. DNN-based speech synthesis

**Prosody:** (The components of speech that are not individual phonetic segments. Not what is said but rather how it is said). Speakers voice, emotional state, speaking style, socio-linguistic background, intentions.

In the speech signal, prosody is associated with 3 components:

1. intonation (speech melody) : fundamental frequency, F0
2. rhythm (speech rate and its variations) : speed and pauses
3. intensity : loudness, energy of the speech signal

Two main approach to model intonation:

1. directly modelling pitch **Momel algo**: sequence of specific F0 target points forms tonal segments. **Tilt model**: measure of shape and amplitude of an event in prosodic (intonation) contour. **Superposition of Functional Contours**: elementary contours extracted with NN
2. trying to simulate the pitch production process: **Command Response (CR) model (Fujisaki)**: decomposes intonation into additive physiologically meaningful components. Two types of components: phrase components (long term) assumed to model the effect of the translation of the thyroid cartilage on the vocal fold tension and accent components (short term) assumed to model the effect of the rotation of the thyroid cartilage on the vocal fold tension. relation between vocal fold tension and F0 is then straightforward. **Stem-ML**: maximisation of two functions, ease of production and speakers estimate of the efficiency of the specific prosody on the listener.

In DNN-based synthesis, F0 is modelled frame by frame, with some general smoothing generally done after generating F0. The linguistic context, input of the DNN, being at the frame level, contains information of higher level than phonemes and word, but also intra-phone information, with some features

as *position in the phoneme*, to model patterns. Continuous F0 improves the perceived naturalness of synthesis. Improved by hierarchical modelling using continuous wavelet decomposition to separate the different levels of variations in F0 into multiple continuous features, on different scales (Suni et al. 2013 and extended by Ribeiro et al. 2016)

**Emphasis:** specific segment of a sentence which is given particular importance (prominence, focus, saliency)

## 4.5 Research plan (P.N. Garner)

Recent research has tended to move towards the architectures of the neural solutions. In some cases, it is because we wish to introduce some sense of the underlying generative model.

It has been shown by Zen et al. (2013) that DNNs can produce more natural sounding synthetic speech. WaveNet (van den Oord et al., 2016) is the first of several DNNs for TTs that output audio waveforms directly. Juvela et al. (2018) show that a WaveNet can be used as the excitation in an otherwise conventional all-pole model of linear prediction. This approach is particularly interesting as it harks back to the source filter model.

Prosody refers to the rhythm and intonation patterns in speech; the aspects that are not typically conveyed lexically, but represent meaning and emotion. Concretely, it is the pitch, energy and duration of segments of speech. Fujisaki (1995) and Strick (1994) tried to explain the physiological aspects of intonation production and, in particular, to relate muscle activity to intonation. Fujisaki's CR model decomposes the pitch contour into a long term component (the phrase component) and short term variations (accent components). These components are modelled as responses of critically-damped second order linear systems to impulses and step functions respectively.

Attempts to implement DNNs in the spirit of physiological systems tend to focus on the outputs of neurons being real numbers that can be interpreted as spiking rate. Biologically plausible algorithms can be cast in a machine learning light.

There have been very few attempts to apply spiking networks to speech processing.

## 5 Other

### 5.1 Software for brain network simulations: a comparative study [?]

### 5.2 Rate coding versus temporal order coding: a theoretical approach [?]

The efficiency of a coding strategy based on rate coding is surprisingly poor. A simple mathematical analysis reveals that, due to the stochastic nature of spike generation, even transmitting the simplest signals reliably would require either: (1) excessively long observation periods incompatible with the speed of sensory processing or (2) excessively large numbers of redundant neurones, incompatible with the anatomical constraints imposed by sensory pathways.



Alternative temporal code: asynchrony of firing across a population of afferent neurons. Simple way of using asynchrony is to use the order in which the neurones spike as a code. In this case, the exact latency at which a neurone fires is not critical - only the rank order of each neurone is important. (more robust, the order is fully invariant wrt. changes, transmit a large amount of information very rapidly)