1 SPIKE BASED NORMALIZING HEBBIAN LEARNING IN AN ANALOG VLSI ARTIFICIAL NEURON

Philipp Häfliger and Misha Mahowald

Institute of Neuroinformatics UNIZ/ETHZ, Gloriastrasse 32, CH-8006 Zürich
hafliger@ini.phys.ethz.ch

Recent neurophysiological results indicate that changes in synaptic efficacy are dependent on co-occurrence of a pre and a postsynaptic spike at the synapse [11, 7]. There are only a few models of parts of the nervous system that use temporal correlation of single spikes in learning [3]. In most artificial neural networks neurons communicate by analog signals representing frequencies, and their learning rules are also defined on these continuous signals. Timing of single spikes is not used, nor is it represented. This simplification has proven useful in many applications and it makes simulations in software simpler and faster. Spiking systems have been avoided because they are computationally more difficult. However, by implementing spiking and learning neurons in analog VLSI it is possible to examine the behaviour of these more detailed models in real time. This is why we and others [1] have started to use silicon models of spiking learning neurons. We have formulated one possible mechanism of weight normalization: a Hebbian learning rule that makes use of temporal correlations between single spikes. We have implemented it on a CMOS chip and demonstrate its normalizing behaviour.

1.1 INTRODUCTION

Recent *in vitro* experiments in young rat neocortex have shown that the relative timing of excitatory postsynaptic potential (EPSP) and action potential (AP) is a main factor in determining changes in synaptic efficacy [11, 7]. It seems that a causal relationship between a pre- and a postsynaptic spike strengthens the synapse whereas presynaptic spikes that follow a postsynaptic action potential weaken it. Following this rule, synapses that contributed directly to the generation of an AP get enhanced.

Such rules are like Hebbian learning rules that are sensitive to temporal correlations of single input and output spikes rather than simply activity correlations. Gerstner et al. in [3] used such a rule to tune artificial neurons, sensitive to azimuthal sound localization, in their simulation of the Barn Owl's auditory system. Schultz and Jabri in [9] proposed a circuit, that uses such temporal correlations, to achieve biologically realistic Short Term Potentiation. We too presented a spike based learning rule, the Modified Riccati Rule (MRR), and showed that it preferes temporally correlated input spike-trains over uncorrelated ones, even if there were no differnces in their average frequencies [4].

For these learning rules to function, the information as to when the cell has produced an action potential must be known at the synapse. In biology this information may be carried by the AP traveling back through the dendritic tree [10]. Furthermore the back-propagating postsynaptic AP is a shared signal among all synapses and could therefore lead to some coordinated behaviour among them. In the MRR [4] for instance, we exploited this property to achieve approximate weight vector normalization. Similarly Akers et al. [1] have implemented Oja's Rule[1] on a CMOS chip. Since this rule is more complicated than the Riccati Rule, it needs a more sophisticated implementation: their chip uses external phase signals to compute the average and square of the output frequency. Like the original Oja's Rule, it is sensitive to average frequencies of input and output and not to temporal correlations of single spikes.

At NIPS '96 [4] we introduced a preliminary chip that implemented a learning rule that is slightly different to the MRR and shows no normalizing behaviour. Now we have an analog VLSI (aVLSI) implementation of the MRR in hand that is sensitive to temporal correlations and performs approximate weight vector normalization. To achieve this normalizing property no phases need to be imposed on the chip nor is there any average explicitly computed.

1.2 A SPIKE BASED LEARNING RULE

The learning rule we briefly describe here has been proposed by us in [4]. It is inspired by biological findings and the Riccati Equation, a learning rule formulated for rate coded input and output [5]. We call it the Modified Riccati Rule (MRR). Although it is purely artificial, it has some interesting features in common with those observed in biological synapses:

1. The information that an AP has been generated is shared among the input synapses.

2. Temporal correlations between pre- and postsynaptic spikes determine the changes in synaptic efficacy.

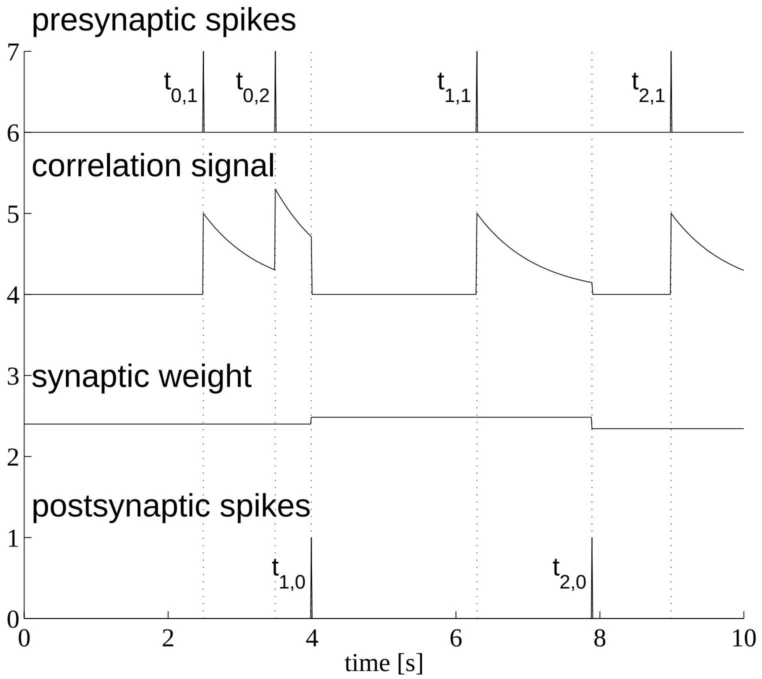

Figure 1.1 A snapshot of the simulation variables involved at one synapse. $\tau = 0.8306$sec.The correlation signal corresponds to c in (1.1), the synaptic weight to w. The pre- and postsynaptic spikes determine the events $t_{i,j}$, where i numbers the postsynaptic spikes and j counts occurrences of presynaptic spikes after the last postsynaptic spike. If the weight is increasing or decreasing with a postsynaptic spike depends on its present size and on the size of the correlation signal at that moment (see (1.1)).

3. Causal relationships lead to an increase in synaptic efficacy whereas ’anti-causal’ relationships weaken the connection.

In order to detect the temporal correlations between input and output spikes, the synapse must keep track of the recent events. In nature, calcium dynamics seem to be involved in this [12]. In our learning rule a simulation variable c, which we named correlation signal, plays this role. It stores information about the recent presynaptic activity: every presynaptic spike increments c by one and it decays in time with a time constant τ. In our rule, weight updates happen whenever an output spike is released and their signs and magnitudes are dependent on this correlation signal c.

1.2.1 Definition

The following rule defines the function of one synapse:

$$
\begin{aligned}
c(t_{m,0}) &= \begin{cases} e^{-\frac{t_{m,0}-t_{m-1,s}}{\tau}} c(t_{m-1,s}) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases} \\
c(t_{m,1}) &= 1 \\
c(t_{m,n}) &= e^{-\frac{t_{m,n}-t_{m,n-1}}{\tau}} c(t_{m,n-1}) + 1 \\
&\text{if} \\
n &> 1 \\
t_{m,n} &\leq t_{m+1,0} \\
s &= max\{v : t_{m-1,v} \leq t_{m,0}\}
\end{aligned}
\tag{1.1}
$$

$$
w(t_{m,0}) = w(t_{m-1,0}) + \alpha c(t_{m,0}) - \beta w(t_{m-1,0}) \tag{1.2}
$$

where w is the weight at this synapse, $t_{m,0}$ is the time of the m’th postsynaptic spike and $t_{m,n} \quad (n > 0)$ is the time of the n’th presynaptic spike after the m’th postsynaptic spike. s is the number of presynaptic spikes between the $(m-1)$’th and the m’th postsynaptic spike, so $t_{m-1,s}$ is the last event (presynaptic or postsynaptic spike) before the m’th postsynaptic spike; α and β are parameters influencing the learning speed and the weight vector normalization (see (1.5)) and τ is the time constant for the correlation signal’s decay.

If τ approximates infinity, so that the decay can be neglected, it can be shown that this rule becomes equivalent to the Riccati Equation as described in [5]:

$$
\frac{\delta}{\delta t}\vec{w} = \alpha \vec{I} - \beta \vec{w} O \tag{1.3}
$$

where O is the neuron's output frequency and $\vec{I}$ is the vector of input frequencies.

The equivalence of the MRR to the Riccati Rule becomes clear, if the correlation signal c is thought of as an input spike counter. With every output spike, w is incremented by αc. After one second the sum of all those increments is αI, with I being the average input frequency arriving at this synapse. Because w is decremented with every output spike by βw, the sum of those decrements during one second will be $\beta w O$.

The Riccati Equation is a non-Hebbian learning rule that normalizes the weight vector, and sets its direction to be the same as the input vector's direction.

With $\tau < \infty$ we gain the following features:

1. MRR is a Hebbian learning rule.

2. MRR considers temporal correlations between input and output spikes and favours synapses that received input shortly before an AP occurs.

If the inputs do not show temporal correlations, the MRR behaves as a normal, frequency based, weight normalizing, Hebbian learning rule. We used it in that manner in unpublished Matlab simulations to solve some small-scale problems, for instance learning direction sensitivity in an array of neurons that are connected excitatorily to their neighbours with temporal delays; and letting two inhibitorily interconnected neurons become pattern selective, by presenting them two alternating input patterns.

If however the inputs do show temporal correlations of single spikes, the MRR reacts differently: In [4] we showed (in simulation) its ability to choose two input spike-trains with 40% coincident spikes over two uncorrelated ones, even if all were firing with the same average frequency.

1.3 IMPLEMENTATION IN ANALOG VLSI

We have fabricated a chip, in 2μm CMOS technology, which simulates this learning rule in analog hardware. Figure 1.2 is a block diagramm of the whole neuron. Figure 1.3 shows a repeatedly used component in the circuits: The so called trigger circuit produces digital pulses of adjustable width given a rising edge as input. Figure 1.4 describes the soma, and figure 1.5 shows a learning synapse, of which our silicon neuron has three. Figure 1.6 is an example of data from the chip, produced by the trigger circuits to time different events at the synapses.

We tried to make this implementation as compact as possible and to match it closely to the theoretical learning rule. Since we had to find a compromise between these two criteria, the chip differs in some points from the theory. In particular, the influence of the correlation signal on the weight increases is exponential rather than linear. By adapting the range of the correlation signal with the parameters tau and $delta$ to a small 'piece' of this exponential one can make this influence as 'linear' as possible.

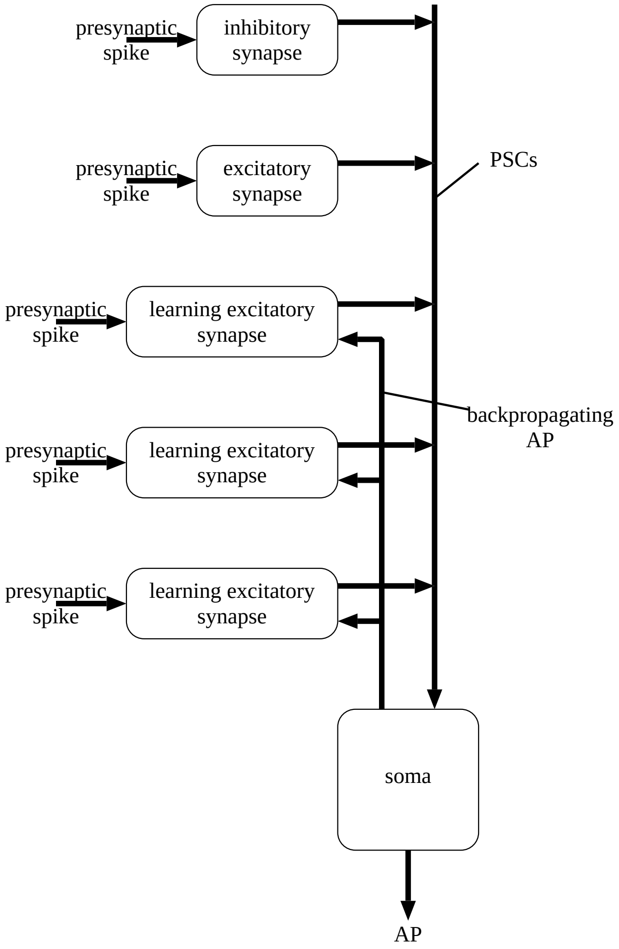

Figure 1.2 A block diagram of the whole neuron. We equipped it with five synapses, 3 of which are learning and two non learning. The synpases send a postsynaptic current (PSC) to the soma when stimulated and receive a pulse back whenever an action potential (AP) is produced. This figure demonstrates that the only information shared among the learning synapses is the moment of the occurance of an AP. That is enough for this local learning rule to coordinate the growth of the weights such that weight vector normalization is achieved.

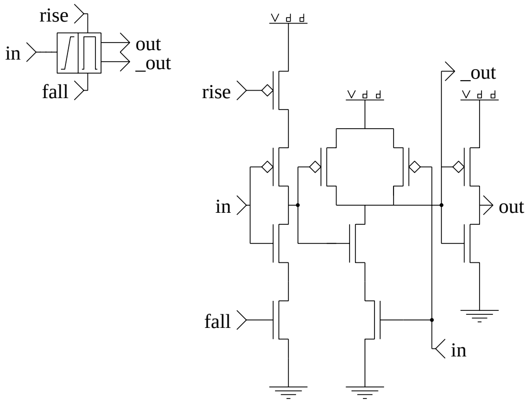

Figure 1.3 This 'trigger circuit' produces a digital pulse of adjustable width as response to a rising edge. Later in this paper we will use its icon (in the upper left). The circuit is made up of a NAND gate and two inverters. The NAND and one inverter form a AND gate. The input signal is given directly to one input of the AND and delayed and inverted to the other. $fall$ determines that delay and therefore the width of the trigger circuits output pulse. $rise$ being slightly smaller than Vdd ensures that the falling edge of the signal does not produce a glitch.

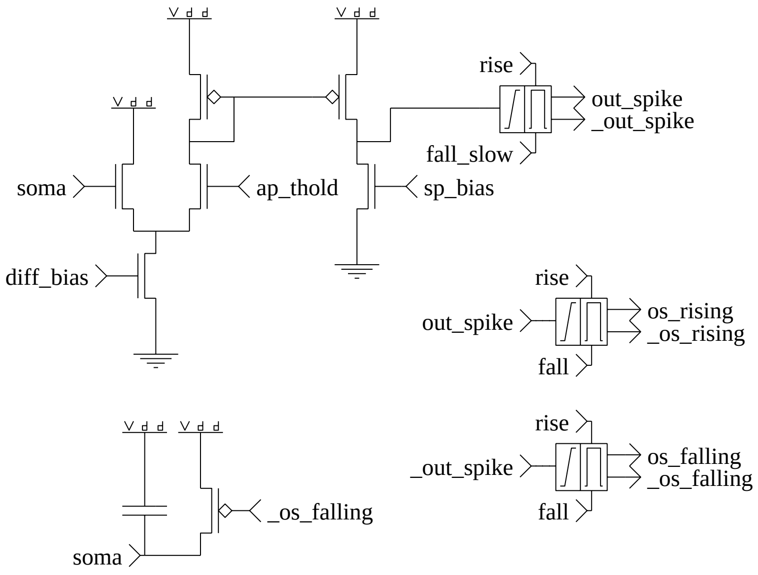

Figure 1.4 The circuit for the soma: It contains three trigger circuits. The two lower ones provide timing information to the synapses about the rising and falling of the action potential (os_rising, $os\_falling$). The third embedded in the top circuit, tuned by $fall\_slow$ to produce wider pulses, provides the action potential. It is activated as soon as the soma voltage (inverted sense compared to nature) falls below a threshold (ap_thold). A differential pair, a current mirror and a simple amplifier change that information into a rising voltage. The rising edge then triggers the trigger circuit. The soma is resetted to Vdd by $\_os\_falling$.

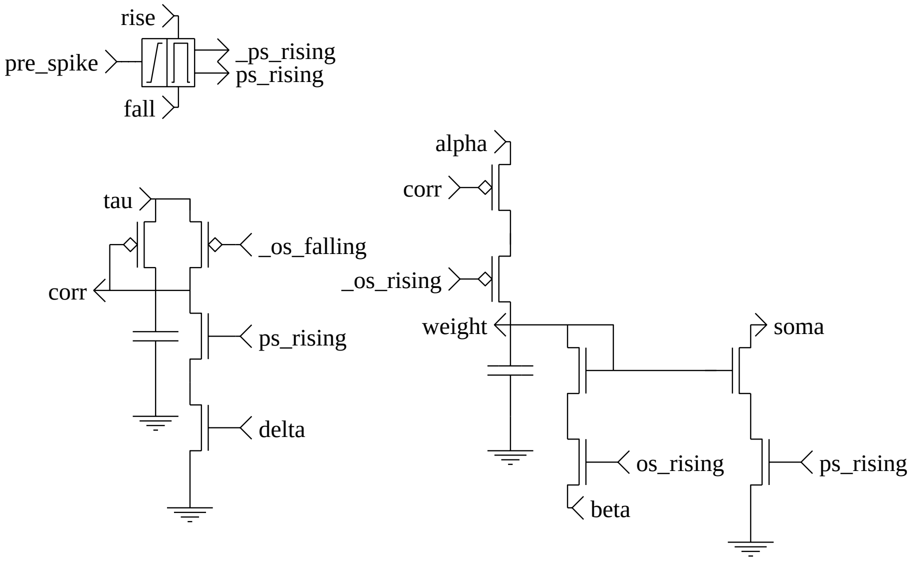

Figure 1.5 The circuit for one synapse: It first captures an incoming spike with a trigger circuit and translates it to a pulse of well defined width (ps_rising). This pulse opens a gate which allows a current (EPSC) to flow from the $soma$ to ground, by an amount given by the synapse's $weight$ (The polarity of the membrane voltage is opposite to that observed in nature). It also decreases (increase in theoretical formula) the correlation signal limited by the voltage $delta$. $alpha$ and $beta$ as in the formula set the amount of the weight change at a AP-event. Further control over the theoretical values α and β is provided by the length of the os_rising pulse. (see the circuit for the soma in figure 1.4). The correlation signal's decay and its lower limit is controlled by the parameter tau. os_rising, $\_os\_rising$ and $\_os\_falling$ are signals generated in the cells AP. and are summarized in figure 1.2 as 'backpropagating AP'. They could also have been produced locally at the synapses but since they are needed at every synapse it is more economical in chip area to produce them once at the soma and to distribute them. os_rising and $\_os\_rising$ cause a weight update and $\_os\_falling$ resets the correlation signal.

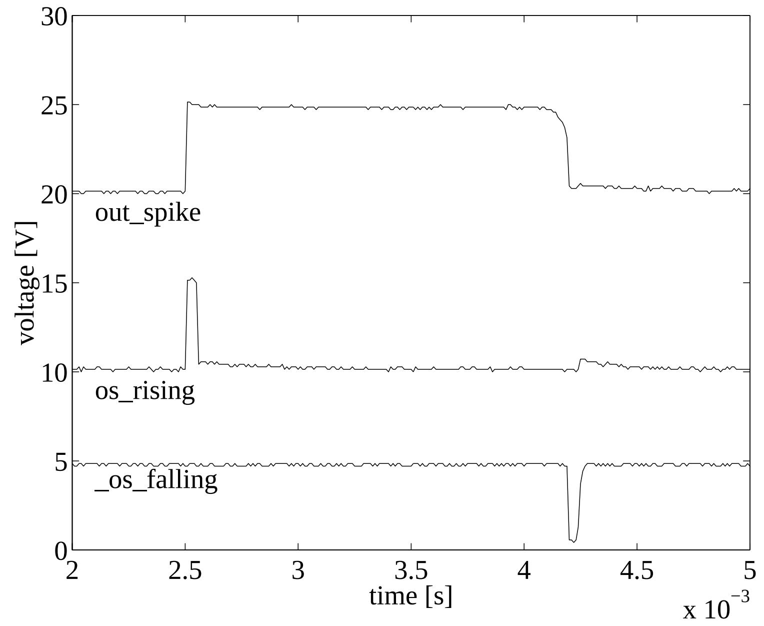

Figure 1.6 Scope traces of internal timing signals^3, taken from a longer recording sequence. When os_rising is high the weights get updated and then when $\_os\_falling$ is low, $soma$ and $corr$ are reset. A presynaptic spike also produces a similar timing signal (ps_rising, not shown here). ps_rising allows synaptic current to flow to the soma and increments the correlation signal $corr$.

1.4 NORMALIZATION

We define normalization as a mechanism to maintain the length of the weight vector constant. This prevents the synaptic weights from growing infinitely, which is a danger in more basic Hebbian learning rules, especially in recurrent networks. On the other hand it also prevents the case that all synapses of one neuron become ineffective. It keeps the gain of a neuron constant: $O = ||\vec{w}|| ||\vec{I}||$ if $\vec{I}$ is an optimal stimulus.

In many learning rules this normalization is performed as an extra step after the weights have been adjusted. In other cases, like the Riccati or the related Oja's rule, the normalization is included in the learning rule and the extra step is not required. The MRR is inspired by the Riccati rule and has inherited an approximate weight vector normalization from it.

To see how this is achieved one can view the output spikes as an adaptive clock. A bigger input load accelerates this clock and if there is less input to the cell the clock slows down. The clock rate normalizes the input load, which makes it a lot easier to formulate a learning mechanism that normalizes the weights too. The rule presented here is an example for this.

1.4.1 Normalization in the MRR

If we assume that the condition given by (1.4) for the output firing rate O is met (for example by using a non leaky integrate and fire neuron) and we neglect the decay in the MRR by letting τ approach infinity, then in the equilibrium state (assuming $\frac{\delta}{\delta t}\vec{w} = 0$), using (1.3) we can compute the length of the weight vector $||\vec{w}||$ as described in [5]:

$$O = \vec{w}^T \vec{I} \tag{1.4}$$

Multiplying (1.3) from the left with $\vec{w}^T$, then substituting $\vec{w}^T \vec{I}$ with O and finally solving for $\vec{w}^T \vec{w}$ leads finally to (1.5).

$$||\vec{w}|| = \sqrt{\frac{\alpha}{\beta}} \tag{1.5}$$

If $\tau < \infty$ this normalization gets corrupted. Since in every situation the incrementing term in (1.2) will be smaller with the correlation signal decaying, the weight vector norm will be smaller too. So one would need to replace the '$=$' in (1.5) by a '$\leq$'. This effect can be countered by using a leaky integrate and fire spike mechanism, using the same time-constant τ for the leak. This way the decrementing term in (1.2) would be smaller too. As the present aVLSI implementation uses a non leaky integrate and fire mechanism, the normalization is not exact, but at least the weight vector length is restricted. As simulations and experiments show (figures 1.7 and 1.8), it tends to be not far away from that upper limit (approximately the maximum value in the graphs). The main factor determining the difference of the weight vector length and its upper limit is the ratio of τ to the average inter-spike interval of the output spikes: the bigger the output spike intervals, the bigger the norms deviation from (1.5), because then the decay has more time to affect the correlation signals substantially during these intervals.

1.4.2 Normalization in our aVLSI implementation

Figures 1.7 and 1.8 illustrate test runs on the chip where two synapses received Poisson distributed spike inputs[4]. Two effects make the chip deviate from the optimal normalization: due to fabrication asymmetries, synapse 1 tends to be stronger than synapse 2. That is why the value at the right end of the graph is bigger than at the left end.

The second effect is the one expected by the theory: With the MRR, in contrast to the original Riccati rule, where the output frequency is smaller, the weight norm gets smaller too. The depression where the two average input frequencies are similar is caused by this effect. In the original Riccati rule the weight-vector follows the average input-vector

$$\frac{\vec{w}}{\|\vec{w}\|} = \frac{< \vec{I} >}{\| < \vec{I} > \|} \ . \tag{1.6}$$

Therefore also the output-frequency O remains constant for a constant length of the input vector ($O = \|\vec{I}\|\|\vec{w}\|$). With the MRR however the input- and the weight-vector will not align and they will deviate the most when the input-vectors values are all equal. The more they deviate the smaller the output-frequency O will become and the more severely will the weightvector length decrease.

1.5 CONCLUSION

We propose that a back-propagating action potential can be used to obtain coordinated changes in synaptic efficacy among all synapses of one neuron. For example weight vector normalization can be achieved as demonstrated here in a theoretical learning rule (MRR) and in an aVLSI artificial learning neuron.

Clearly the use of volatile weight storage in learning is not ideal. Work in progress is addressing the issue of non volatile weight storage in on chip learning, using floating gates [2], and therefore enabling our chips to preserve their learned state even when they are switched off or inactive, without having to store weights digitally.

We plan to extend our work on learning with single synapses to networks of neurons. Since that requires inter-chip communication, we intend to equip future chips with the ability to use the address event protocol [8, 6] for that purpose.

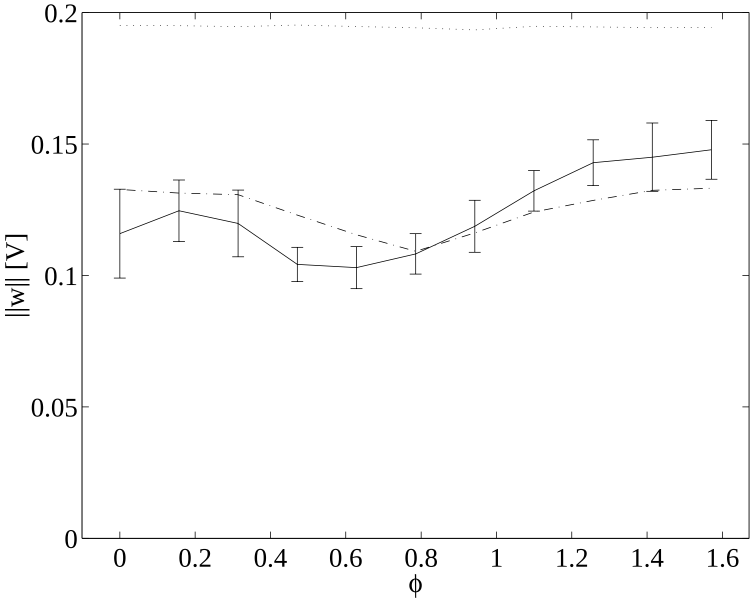

Figure 1.7 The graph shows the weight vector length on the y-axis for an experiment with our chip using two synapses, both of which get Poisson distributed spike-trains with constant average frequency as input. The squared sum of the average input frequencies was kept constant. The x-axis is an angle ϕ which is the angle of the polar coordinates of the input vector. In other 'words' the input frequencies ν_1 and ν_1 are defined as $\nu_1 = 100sin(\phi) \quad \nu_2 = 100cos(\phi)$. The weights were given time to settle and were then taken as the mean of 50 oscilloscope traces of 200ms each. The bars show the standard deviations in these sets of 50 samples. For comparison we also show two results of computer simulations of the MRR. The dash-dotted line is the result of a simulation with equivalent parameters to the chip. It is more nicely symmetric than the chip data, as is to be expected, since in the simulation there are no mismatches between the two synapses. On the chip the first synapse tends to be stronger. The dotted line results when the somas membrane voltage leaks out with the same timeconstant as the correlation signal. This could not be replicated on the chip, since we used a simple integrate-and-fire mechanism, without leakage through the membrane.

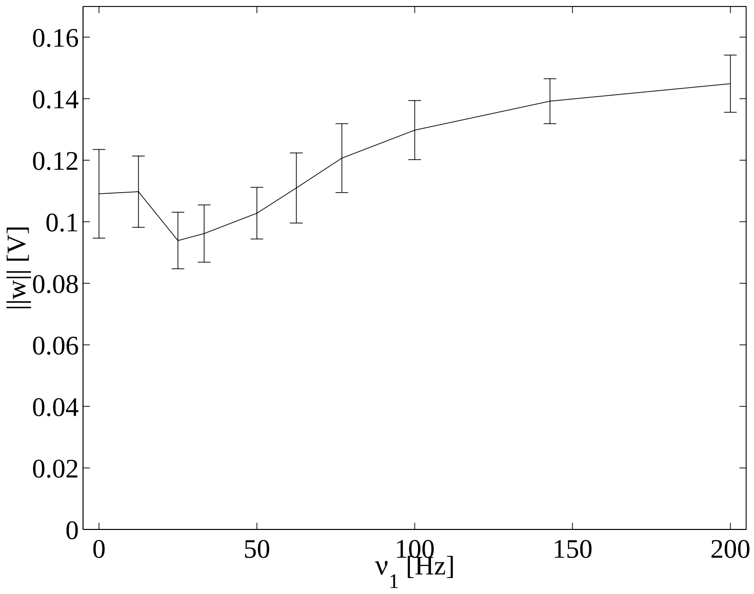

Figure 1.8 Again (see figure 1.7) the weight vector length for an experiment with two synapses stimulated with Poisson distributed spike signals is shown. This time the average frequency of synapse 2 is kept constant at 50 Hz. The x-axis shows the average frequency given to the other synapse (ν_1). The weights were taken as the mean of 200 oscilloscope traces of a duration of 200ms each. The bars show the standard deviation in these sets of 200 samples.

Acknowledgments

We thank the following organizations for their support: SPP Neuroinformatik des Schweizerischen Nationalfonds, Centre Swiss d'Electronique et de Microtechnique, U.S. Office of Naval Research and the Gatsby Charitable Foundation.

Notes

1. Oja's rule is closely related to the Riccati Rule, that also normalizes the weight vector
2. Compare the signal names in the circuits in figures 1.4 and 1.5.
3. Compare the signal names in the circuits in figures 1.4 and 1.5.
4. We used a National Instruments Lab-PC card and a PC running Linux to provide these spike trains. The program and the altered driver can be obtained from the corresponding author.

References

[1] L.A. Akers, D.K. Ferry, and R.O. Grondin. *An introduction to neural and electronic networks*, chapter Synthetic neural systems in the 1990s, pages 359–387. Academic Press, 1995.

[2] C. Diorio, S. Mahajan, P. Hasler, B. Minch, and C. Mead. A high-resolution non-volatile analog memory cell. *Proc. 1995 IEEE Intl. Symp. on Circuits and Systems*, 3:2233–2236, 1995.

[3] W. Gerstner, R. Kempter, J.L. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78, 1996.

[4] Ph. Häfliger, M. Mahowald, and L. Watts. A spike based learning neuron in analog VLSI. *Advances in neural information processing systems*, 9, 1996.

[5] Tuevo Kohonen. *Self-Organization and Associative Memory*. Springer, Berlin, 1984.

[6] Misha Mahowald. *An Analog VLSI System for Stereoscopic Vision*, chapter 3, pages 66–117. Kluwer Academic Publishers, 1994.

[7] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215, 1997.

[8] A. Mortara and E. A. Vittoz. A communication architecture tailored for analog VLSI artificial neural networks: intrinsic performance and limitations. *IEEE Transactions on Neural Networks*, 5:459–466, 1994.

[9] S. R. Schultz and M. A. Jabri. A silicon basis for synaptic plasticity. *Neural Processing Letters*, 2(6):17–22, 1995.

[10] G. J. Stuart and B. Sakmann. Active propagation of somatic actionpotentials into neocortical pyramidal cell dendrites. *Nature*, 367:69ff, 1994.

[11] M. V. Tsodyks and H. Markram. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382:807–810, August 1996.

[12] R. Yuste and W. Denk. Dendritic spines as basic functional units of neuronal integration. *Nature*, 375:682–684, 1995.