Human Brain Project Co-funded by the European Commission

DISCLAIMER: More detailed information on the results and/or performance obtained and their use is available in the Subproject's subsequent Deliverable (D4.6.4) and/or Periodic Report.

Project Number:	604102	Project Title:	Human Brain Project

Document Title:	Algorithms, Cognitive Models and Computing Principles for the HBP Human Brain Atlas: Package 1
Document Filename:	SP4 D4.6.3 FINAL.docx
Deliverable Number:	D4.6.3
Deliverable Type:	Other
Work Package(s):	WPs 4.1, 4.2, 4.3, 4.4, 4.6
Dissemination Level:	PU = Public
Planned Delivery Date:	Month 18 / 31 March 2015
Actual Delivery Date:	Month 20 / 15 May 2015

Authors:	Alain DESTEXHE, CNRS (P7)
Compiling Editors:	Katherine FREGNAC, CNRS (P7)
Contributors:	Joni DAMBRE, UGENT (P66) Wulfram GERSTNER, EPFL (P1) Neil BURGESS, UCL (P70) Andre GRUNING, Brian GARDNER, SURREY (P111) Wolfgang MAASS, TUGRAZ (P54) Marja-Leena LINE, TUT (P99) Walter SENN, UBERN (P62) Marc DE KAMPS, ULEEDS (P10) Gustavo DECO, Mathieu GILSON, Gerald HAHN, UPF (P65) Abigail MORISSON, JUELICH (P17) Olivier MARRE, UPMC (P107)
STO Review:	UHEI (P45): Björn KINDLER, Sabine SCHNEIDER, Martina SCHMALHOLZ
Editorial Review:	EPFL (P1): Guy WILLIS,

Abstract:	Deliverable D4.6.3 Algorithms, Cognitive Models and Computing Principles for the HBP Human Brain Atlas: Package 1. This Deliverable will provide an initial set of brain models, algorithms and computing principles. These will include models for perception-action, learning algorithms, algorithms for mapping models with morphologically detailed neurons to point neuron models, and brain-inspired principles of computing suitable for implementation in neuromorphic computing systems. A short report will describe each algorithm, model and principle, outline the state of validation work and provide details on how to access the model in the Human Brain Atlas.
Keywords:	Computational models, algorithms, simulation codes

Table of Contents

List of Figures and Tables

1. Executive Summary

This Deliverable provides an initial set of brain models, algorithms and computing principles. These include models for perception-action, learning algorithms, algorithms for mapping models with morphologically detailed neurons to point neuron models, and brain-inspired principles of computing suitable for implementation in neuromorphic computing systems.

The Deliverable follows from the first description of the different models given in D4.6.2. The different models are updated here, with a detailed description for each model, as well as programme codes for most of them. It is important to note that the programme codes given here are "first drafts" of codes and they represent the present status of the models. All of these models are still under study, and for some of them it is too premature to give codes at this stage (in particular for Tasks 4.1.2 and 4.3.3, which started only 6 months ago).

Our plan is that the posting of programme codes will be an on-going process, independent of any Deliverable. As soon as a new version of the code is obtained and is stable, it will be posted in the Human Brain Atlas. The last Deliverable D4.6.4, planned for Month 30 (March 2016), will contain a description of the models and the final version of the programme codes.

Another important point is that it is also our plan to publish all the models developed here and make the programme codes available publically in known model databases such as ModelDB. The latter requires that a model code must correspond to a publication where all details can be found. It is our plan to follow this strategy of open-access.

Lastly, it must be noted that the Human Brain Atlas is not yet ready for posting codes, so in the mean time we have set up our own repository for SP4, where codes are posted and continuously updated. They are available to the whole HBP, while waiting for the Neuroinformatics Platform to be functional.

2. Introduction

This Deliverable has two objectives. The first is to give an overview of the different models investigated in SP4 and how they relate to various other aspects of the HBP. The second is to provide the programme codes used to simulate most of these models.

The models investigated in SP4 concern many different aspects of brain function and different scales. We summarize below the different modelling approaches considered in this Deliverable.

A first class of models concern the investigation of brain cells, brain networks and brain signals at different scales. Here, models investigate the integrative properties of dendrites and the approaches followed are first to simplify the complex dendritic morphology of neurons into simplified models that capture the essence of dendritic integration and nonlinearities (such as dendritic spikes). Such models are conceived so that they are compatible with the hardware. Another important effort is to provide a set of models to simulate brain signals at different scales, first focusing on local signals such as the local field potential. The goal here is to provide such models for both detailed and simplified neuron representations. We also investigate population-level models, for which a simulator has been conceived and released for general use.

A second main effort concerns the plasticity algorithms and their consequences in learning and memory paradigms. Several algorithms are described, including plasticity algorithms that explicitly use dendrites. We hope that appropriate modification of simulation programs (NEST) will allow the implementation of these algorithms.

A third main field of modelling in SP4 focuses on large-scale models of cognitive functions. The different models investigated concern models of perception-action and spatial navigation, models of attentional processes, models of realistic network states and network models including glial cells. These models should be available in a format that should allow them to be included in the Brain Simulation and Neuromorphic Platforms.

Finally, SP4 also investigates general principles of brain computation. The work described here is about computations in dendrites, computations in circuits, and general principles of computation inspired from biology and with possible applications in robotics. Retinal computations (at the population level) are also investigated with a goal to provide a working model of the retinal “input” to thalamocortical models.

In all cases, the programme codes are either made available as a supplementary information to this Deliverable (posted to the Human Brain Atlas), or will be made available and posted later this year.

3. Work Package 4.1: Bridging Scales

The “Bridging Scales” WP comprises three Tasks. The first Task is about obtaining simplified models of dendrites. The second Task is about models to generate brain signals at different scales, and the third Task is about population models of brain activity.

3.1 Task 4.1.1: Derive simplified neuron and neural circuit models from biophysically morphologically detailed models

3.1.1 HUJI (Idan Segev, Task Leader)

The HUJI (Hebrew University of Jerusalem) team is modelling the L2/3 Human pyramidal cell and developing systematic methods for reducing the complexity of the model. To this end, HUJI has developed the following models and tools.

1) Realistic (morphologically-based) models of individual human spines (based on electron microscope and light microscope data).
2) A method for incorporating globally the large numbers of dendritic spines (up to 15,000 spines/cell) into the L2/3 cell model.
3) An automated method for matching recorded somatic EPSP from a pair of connected L2/3 cells to the corresponding EPSP model. This provides direct prediction on the dendritic *loci* of the synaptic connection between the two cells.
4) An automated method for studying conditions for the generation of local NMDA spikes in human neurons.
5) A new statistical method for “morphing” rat/mouse L2/3 pyramidal neurons into human L2/3 neurons.
6) HUJI is currently exploring a new reduction/simplification method that preserves, very closely, the I/O relationship of the full (non-reduced) L2/3 human neuron model in a simplified model of the same cell.

Models are all written either in HOC (Neuron) file or in Python and will be soon available publically.

3.1.2 CNRS (Alain Destexhe)

The UNIC partner developed several models of dendritic excitability under *in vivo* conditions (joint work with Mathieu Galtier, Romain Veltz and Alain Destexhe).

3.1.2.1 Models

Three models were investigated. They represent different levels of abstractions for excitable dendrites.

1) Biophysical model of excitable cable. The excitable cable was modelled by the Fitzhugh-Nagumo (FN) model (a reduction of the Hodgkin-Huxley model), which was placed in a chain of several dendritic compartments, each equipped with the FN model. This model generates spikes, and the spikes propagate across the dendrite structure.
2) Simplified binary model of dendritic cable. This model consists of a binary reduction of the more complex models. A spike is represented by "1", while the resting membrane potential is "0". A spike in one compartment evokes a spike in the neighbouring compartment, and one can simulate this by a cellular automaton, which will simulate the propagation of spikes in dendritic cables.
3) A dendritic cable model using the AdEx model, fully compatible with neuromorphic hardware. This model is under development at the moment and will be posted later.

3.1.2.2 Model structure

We consider a single ball and stick neuron, i.e. with only one branch. The single branch is separated in a compartment that displays an excitable dynamics: a small perturbation generates a small activation, which fades away; yet, a slightly larger perturbation may trigger a large activation that we call a dendritic spike. With more specification, this kind of dynamics gathers most of the behaviours due to voltage-dependent ion channels observed in dendrites (i.e. sodium, calcium or NDMA spikes). Because all the dendritic compartments are connected through a linear cable equation, dendritic spikes propagate in both sides. An important assumption that we made is that the dendritic compartment is homogeneous, i.e. there are all structurally identical. This guarantees that the propagation in one direction (e.g. to the soma) is identical to the propagation in the other direction (e.g. away from the soma).

The dendrite is assumed to be stimulated with randomly generated trains of spikes. This corresponds to an *in vivo* situation where synaptic bombardment occurs. Importantly, we consider that this synaptic bombardment may have a given correlation structure. For simplicity, we will assume that the cross covariance function between two different trains of spikes (stimulating two different compartments) is of the form

$$\tau \rightarrow 2l\, e{-}l/\tau\, /$$

where l controls the degree of correlations between the trains of spikes (larger l means larger correlations).

All models were programmed in Python format and are available on the UNIC-CNRS server (see implementation details in the README file). The behaviour and computation principles associated with these models are described under Task 4.4.1.

3.1.3 EPFL (Wulfram Gerstner)

The EPFL-LCN partner developed two algorithms so as to extract simplified neuron models from data or more detailed models.

1) Neuron models from data: In collaboration with Christof Koch from the Allen Institute, we developed an algorithmic pipeline that enables us to extract neuronal parameters from less than 60 seconds-worth of single-cell current injection into neurons. The methodology has been verified and written up in a paper submitted to PLOS Computational Biology. It will be used routinely by the Allan Institute for large-scale data collection. The resulting data will be public and made available to the HBP. The paper in PLOS Computational Biology (to appear) acknowledges HBP support.
2) Neuron models from more complex neuron models. We have designed a workflow and stimulation paradigm to extract simple generalized integrate-and-fire models from detailed HBP neuron models.

The basic idea is to stimulate the model at many synapses in parallel and extract linear response kernels on a per-synapse basis, using systematic optimization methods. Note that the dendrites of complex neuron models are non-linear, so that a combination of linear responses can only be an approximation. In future validation work, we will quantify the quality of this approximation using different stimulation paradigms.

Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

3.2 Task 4.1.2 Modelling brain signals at different scales

3.2.1 CNRS (Alain Destexhe)

Task 4.1.2 started 6 months ago. UNIC (the Edex-Educational Excellence Corporation of the University of Nicosia) has started to analyse the relationship between local field potential (LFP) and single units. This analysis is performed based on Utah-array recordings in human and monkey, during natural wake and sleep states. We calculate the spike-LFP relation, separately for regular-spiking (presumed excitatory) and fast-spiking (presumed inhibitory) cells (the excitatory or inhibitory nature of some of these cells was confirmed from functional interactions). The results from this analysis will give us the typical "kernel" of the relation between LFPs and spikes, and this kernel will be used to obtain a phenomenological model of LFP that can be applied to spiking neural networks.

This model will be developed this year, once the analysis is done, and the corresponding codes will be posted in Deliverable D4-6.4 in March 2016.

3.2.2 UMB (Gaute Einevoll):

UMB (Norges Miljø- og Biovitenskapelige Universitetet) has explored how active sub threshold dendritic conductance affects the LFP generated by synaptically activated pyramidal neurons. Using published biophysical model neurons, in particular a layer-5 pyramidal neuron developed by Hay *et al*, PLoS Comp Biol (2011), we have found that the so-called Ih channel can significantly affect the low-frequency part of the generated LFP. Moreover, we have found that the effects may be incorporated in multi-compartmental models by means of so called ‘quasi-active’ linearized conductance allowing for efficient numerical implementation in large-scale brain simulations. A publication is in preparation. Furthermore, a paper on the generation of benchmarking data for validation of spike-sorting algorithms (accompanied by the Python toolbox ViSAPy), as well as a paper on the modelling and analysis of electrical potentials in micro electrode arrays (accompanied by the Python toolbox ViMEAPy), have been published. Both VISAPy and ViMEAPy have been added to the HBP repository.

3.3 Task 4.1.3 Mechanistic models of cognition linked to the neural substrate by population density methods

3.3.1 ULEEDS (Marc de Kamps, Task Leader)

The ULEEDS (University of Leeds) group has developed the simulator MINDS. This simulator operates at the population level and implements population-density techniques with the aim of simulating large networks of populations, whilst retaining biological realism. The theory underpinning the simulator is described in http://arxiv.org/abs/1309.1654. (See also Apfaltrer et al.,2006; Omurtag et al. 2000; de Kamps, 2003) If the identity of individual neurons in a simulation is irrelevant then these population level simulations are equivalent to the simulation of large numbers of point model neurons (see Figure 1), unlike for example rate-based methods such as Wilson-Cowan dynamics (Wilson, H. R. and Cowan, J. D. (1972). The simulator is written in C++ and has an MPI engine. The simulator departs from the state-of-the-art in the following way: (1) we can handle arbitrary synaptic distributions, i.e. we are not restricted to the diffusion limit. Fokker-Planck equations are a special case; (2) we can handle any 1D point model. Figure 2 shows a population of quadratic-integrate-and-fire neurons that is partially in synchrony. Currently we are extending this method to higher dimensional neural models.

In Appendix B we to describe the installation process and explain a few simple programmes that serve as demonstrators. As a simulator, this is a data-generating tool. We believe that this tool is important for the HBP and the community at large, and strongly advocate its inclusion in the Brain Simulation Platform (SP6).

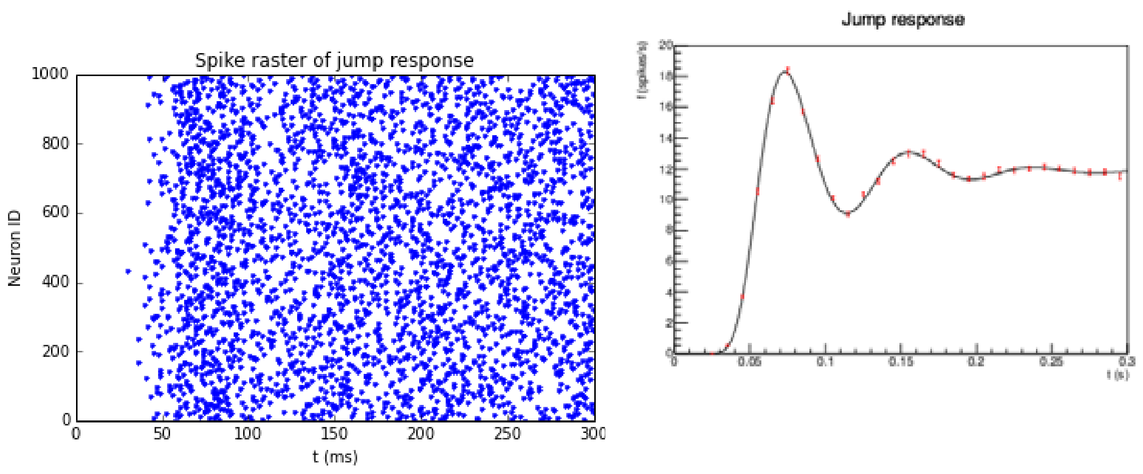

Figure 1: Simulation of a neuronal population-using NEST (left); same simulation using population density techniques using MIIND (right)

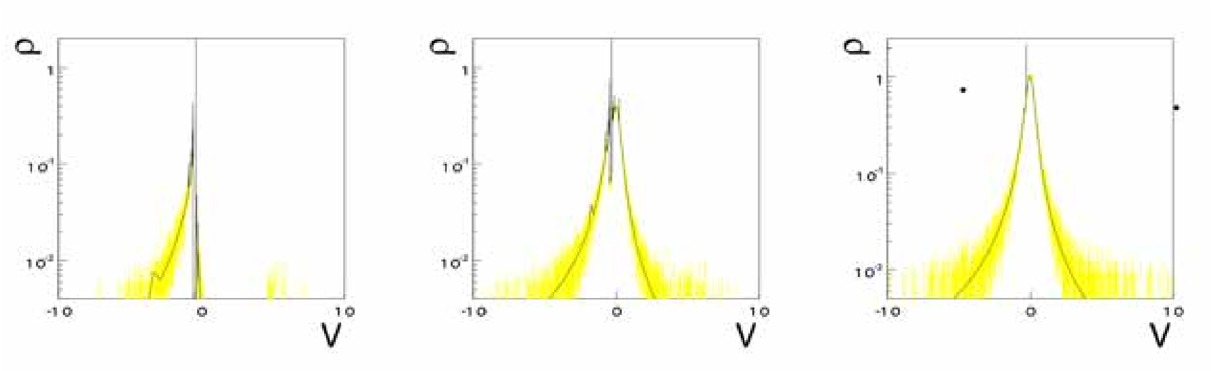

Figure 2: A population of Quadratic-Integrate-and-Fire neurons in partial synchrony

Line: MIIND calculation; yellow markers: Monte Carlo.The sharp peak is a travelling delta function. The population is slowly decorrelated by synaptic input.

Further Reading from Marc de Kamps

F. Apfaltrer, C. Ly, and D. Tranchina (2006). Population density methods for stochastic neurons with realistic synaptic kinetics: Firing rate dynamics and fast computational methods. Network,17(4):pp. 373–418.

De Kamps, M (2013) http://arxiv.org/abs/1309.1654).

M.-O. Gewaltig and M. Diesmann (2007). Nest (neural simulation tool). Scholarpedia, 2(4):p. 1430.

A. Omurtag, B. W. Knight, and L. Sirovich (2000). On the simulation of large populations of neurons. Journal of Computational Neuro-science, 8(1):pp. 51–63.

H. R. Wilson and J. D. Cowan (1972). Excitatory and inhibitory interactions in localized populations of model neurons. Biophysical journal, 12(1):pp. 1–24.

4. WP4.2 - Synaptic Plasticity, Learning and Memory

This Work Package contains three tasks, the first task is about the formulation of learning rules from biophysical models, the second task is about unsupervised learning rules, and the third task is about the structure of different learning algorithms. These three tasks are successively considered below.

4.1 Task 4.2.1 Derive learning rules from biophysical synapse models

4.1.1 UBERN (Walter Senn, Task Leader):

ALGO STDPpredictive. In theoretical abstractions of synaptic plasticity, it has been difficult to reconcile approaches that strive for biological plausibility with models mathematically developed within an error-minimization framework. Specifically for spike-timing-dependent plasticity, provably optimal models have mostly relied on the point neuron assumption, because it has been difficult to establish the computational relevance of detailed biophysical models. Based on a model published before the HBP funding period [1], we propose the ALGO STDPpredictive which sits at the intersection of these two approaches. The key idea is that synaptic plasticity is driven by a prediction error, where the dendritic voltage serves as a prediction of somatic spiking in a compartmental neuron model. Taking this perspective, a single learning rule derived from first principles has been shown to sub serve diverse learning paradigms, depending on the structure of synaptic input, thus providing an intriguing proposal of how computation might be implemented on the single-cell level in the neocortex. In a two-compartment formalism with somatic voltage U and dendritic voltage V, synaptic weights are updated according to

$$\Delta w(t) \sim \left(S(t) - \varphi(V\ \ )\right) PSP\ \ (t)$$

where $\varphi(V\ \ )$ is the dendritic prediction of somatic spiking, based on a low-pass filtered version V of the dendritic potential. Spikes at the soma follow an inhomogeneous Poisson process based on the firing rate $\varphi(U)$ with some voltage-to-rate transfer function φ) leading to a spike train $S(t)$. Finally, learning is modulated by presynaptic activity according to a filtered version of the EPSP $PSP\ \ (t)$.

This learning rule has a self-correcting component, in the sense that large weights will lead to large dendritic predictions resulting in depression on average, whereas small weights will be more likely to be potentiated. Thus, there is no need for special homeostatic terms on the short time scale to prevent unbounded weight growth, which is a common problem in standard Hebbian learning schemes, although a slow homeostatic process that moves the strength of the somatic synapses into a working regime is still required.

Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

Data provenance for Algo STDPpredictive:

As the proposed model is the result of an error minimization scheme that results in provably optimal plasticity dynamics, the reproduction of experimental data is not built into the approach by design. However, in recent yet unpublished work, biological plausibility for the learning rule was established by demonstrating that somato-dendritic prediction error learning exhibits a diverse set of experimentally observed characteristics of STDP, such as frequency dependence [2], postsynaptic depolarization dependence [3] or modulation by action potential backpropagation failure [4]. All these preliminary results are available for reproduction in the accompanying GitHub repository.

Validation Work for ALGO STDPpredictive

Extensive validation of functional relevance was provided in [1] where it was shown that the rule can implement supervised, unsupervised and reinforcement learning. As described above, the on-going effort in establishing biological plausibility has shown that the learning rule offers a new perspective on spike-timing dependent plasticity. Note also that, as postsynaptic depolarization has emerged as the crucial variable guiding plasticity, the compartmental model offers a much more natural setting to investigate biological phenomena than models that rely on the point neuron assumption.

Platform requirements for ALGO STDPpredictive

The algorithm crucially relies on a compartmental description, separating dendritic and somatic voltage into at least two compartments. In addition, filtered versions of the dendritic voltage and EPSPs need to be maintained. Thus, the algorithm is currently not implemented in a specific neural network simulator such as NEST but in pure Python (available from the GitHub repository).

Cross-SP Aspect for ALGO STDPpredictive

ALGO STDPpredictive may serve as a guide to implement more biological features in neuromorphic chips (SP9) that, at the same time, satisfy optimality constraints in terms of learning. Similarly, the algorithm is a candidate to be implemented in the HBP Brain Simulation Platform (SP6). As shown in [1], the algorithm allows for learning of spatio-temporal spike patterns in a recurrent network with hidden neurons. When implemented in a cortical column, it may underlie the formation of micro-columns, motor pattern, or spatio-temporal memory sequences.

Neuroinformatics Links for ALGO STDPpredictive

The model and all associated version history is available in pure Python from the GitHub repository https://github.com/dspicher/py_stdp . Note that, due to conceptual reasons (see above), no NEST implementation currently exists.

4.1.2 EPFL (Wulfram Gerstner)

ALGO STDPorchestrated. Synaptic plasticity is thought to be the basis of memory formation. Many models are influenced by ideas of Hebb, who formulated 'Hebbian learning', together with the concept of 'Hebbian assemblies' that should represent abstract concepts, corresponding to memory items. However, it has remained a big challenge for modellers to use Hebbian learning rules in a continuous online flow of activities while the neural network is stimulated by a continuous stream of external inputs. Most standard Hebbian learning rules will lead to unstable network activity. The purpose of the ALGO STDPorchestrated is to provide a stable learning rule for online formation and retrieval of memories.

The ALGO STDPorchestrated combines Hebbian plasticity (also called homosynaptic plasticity) with heterosynaptic plasticity and transmitted induced plasticity. Hebbian plasticity is formulated as the triplet STDP model [12]. In the triplet STDP model, potentiation needs the combined effect of three spikes: firstly, a presynaptic spike has left an exponentially decaying trace; secondly, a previous postsynaptic spike has left an exponentially decaying trace; and thirdly, a further postsynaptic spike is generated before the two traces have decayed back to zero. The weight change is then proportional to the product of the values of the presynaptic trace and the postsynaptic trace at the moment of the present postsynaptic spike [13]. Long-term depression needs a pair of spikes post-before-pre, implemented via an additional trace left by each postsynaptic spike. How traces are formulated mathematically is indicated below in the context of ALGO STDPStructural.

Heterosynaptic plasticity is implemented with similar traces. Each postsynaptic spike leaves an exponentially decaying trace with a slower time constant. If further postsynaptic spikes occur before the trace has decayed to zero, each novel postsynaptic spike adds to this trace. Heterosynaptic plasticity is proportional to the fourth power of this postsynaptic trace. Heterosynaptic plasticity is therefore sensitive to bursts of postsynaptic spikes. At the moment of each postsynaptic spike, all synapses onto the same neuron are affected (hence the terminology 'heterosynaptic'). The direction of weight change depends on the momentary value of the weight of the synapses compared to a reference weight that evolves on a slower time scale. The algorithm also includes a form of inhibitory plasticity modulated by a global activity-dependent factor. For a full description of the orchestrated plasticity model, see [15].

Model codes for two algorithms (*ALGO STDPorchestrated* and *ALGO STDPStructural*) are available on the UNIC-CNRS server – the other algorithms will be posted later this year and will be included in the final D4.6.4 Deliverable in March 2016.

Data provenance for ALGO STDPorchestrated

This algorithm works via a combination of different terms in an orchestrated plasticity rule. The Hebbian terms driving Long-Term Potentiation and Long-Term Depression are identical to Triplet STDP rule [12] and uses the data of several experimental groups cited therein. The heterosynaptic plasticity term is based on and compatible with the data of [13]. The transmitter-triggered term is based on indirect evidence and has not been reported separately in papers, but is akin to the drift observed in many experiments. Inhibitory plasticity is based on indirect evidence and observations reported in several papers. In the algorithm we use a hypothetical form of inhibitory plasticity modulated by a global factor [14].

Validation Work for ALGO STDPorchestrated

We have validated the algorithm after extensive simulations on a paradigm of online learning and memory formation. The validation work led to a sequence of impressive results that were published by Zenke et al. in the Nature Communications (2015) with acknowledgement of HBP support.

Platform Requirements for ALGO STDPorchestrated

The plasticity algorithm presented here relies on filtered traces of the pre-and postsynaptic activity with several time scales. A general framework for computing and accessing such traces at the synapse would be a very useful addition to simulation software such as NEST [7], because most spike-timing-dependent plasticity rules can be formulated as a functional of such traces. At present, however, such a generic infrastructure is not available in NEST. Therefore, our prototype implementation works on independent stand-alone software in the framework of AURYN [11]. However, to facilitate interfacing with the simulation Platform, we suggest that NEST developers include a framework for synaptic traces in the NEST software. This request was communicated in the cross-subproject workshop at the EITN in Paris on 2-4 March (Are we building the right thing: requirements from theory to simulation environments and neuromorphic computing) and was well received by the Simulation Platform developers.

Because the ALGO STDPorchestrated uses only pre- and postsynaptic traces, it should eventually be possible to implement it in the Neuromorphic Computing Platform. The model would also be suitable for implementation on the Neuromorphic Computing Platform, as soon as the next generation of neuromorphic chips with on-chip synaptic traces becomes available. We remark that several exponentially decaying traces with different time constants are necessary for each synapse. The neuromorphic chip offers the desirable speed-up compared to biological time to allow extensive simulation of the model.

Cross-SP Aspect for ALGO STDPorchestrated

This algorithm addresses, from a computational perspective, some important problems for cognitive neuroscience (SP3), namely memory storage, memory formation, and memory retrieval. The algorithm is formulated at a level where it could be directly implemented in the HBP Brain Simulation Platform (SP6). Since it is relatively efficient (it does not use complex molecular signals at synapses), it is well suited for large-scale simulation experiments (SP6). Moreover, given its formulation with formal spiking neurons of the integrate-and-fire type, it can also be used for large network emulation experiments using the Neuromorphic Computing Platform (SP9). In this sense, the ALGO STDPorchestrated is a true example of cross-cutting work.

Neuroinformatics Links for ALGO STDPorchestrated

This model will be implemented in NEST once the required infrastructure of pre- and postsynaptic activity traces becomes available. As NEST is part of the neuroinformatics Platform, the model will then be available to all users of the Platform.

4.1.3 WIS (Misha Tsodyks)

This Task started late because of the delay in hiring a postdoc. We are pursuing two approaches. In the first one, we consider a learning rule that is a combination of Hebbian plasticity and homeostatic scaling of synaptic strength:

$$\frac{dw_{i,j}^{+}}{dt} = \mu \left(\underbrace{F_i F_j}_{\substack{\text{Syn.}\\ \text{plasticity}}} + \kappa^{-1} \underbrace{\left(F^T - F_i\right)\left(w_{i,j}^{+}\right)^2}_{\text{Syn.Scaling}} \right)$$

We simulate the network receiving a stream of different inputs and see how neuronal ensembles representing different inputs are formed. The code for network simulations will be shared with the HBP Partners in Deliverable D4.6.4 in March 2016.

In the second approach, we generalize the paper by Guetig and Sompolinsky on tempotrons (Guetig and Sompolinsky, 2008) to the case of dynamic synapses. In particular, we derive a learning rule for modifications of all 3 parameters of the short-term plasticity model of (Tsodyks *et al*, 1998). This derivation results in the predictions that can potentially be tested with future STDP experiments.

4.2 Task 4.2.2 Unsupervised learning rules and emergent connectivity

4.2.1 *EPFL (Wulfram Gerstner)*

ALGO STDPstructural T4.2.2 Synaptic connectivity in cortical neuronal networks is subject to continuous remodelling. Dendritic spines, which form most synaptic connections, are formed, grow, shrink, and may even be removed again. On the time scale of minutes to hours, gradual changes in the strength of a synaptic connection can be observed, which are correlated with changes in dendritic spine volume [5]. On the time scale of days, new spines appear and old ones are removed [6, 2]. We propose a new model for the slow processes of formation, maturation, shrinkage and removal of excitatory synaptic contacts, based on spike timing. The model will soon be introduced and motivated in greater detail in a dedicated scientific article. In this report, we just present the algorithm specifications.

Spike-timing-dependent plasticity (STDP) rules can be expressed in terms of traces of the pre- and postsynaptic neuronal activity [4]. Here we use:

$$d/dt\, T_i^{\alpha}(t) \;=\; -1/\tau_{\alpha}\, T_i^{\alpha}(t) + S_i(t)$$

where T_i^{α} is the trace of the spike train $S_i(t)$ of neuron i with time constant τ_{α}. This trace $T_i^{\alpha}(t)$ is identical to the result of applying a low-pass filter (exponential kernel) to the spike train. From pre- and postsynaptic spike trains and their traces, our synapse model computes a correlation trace q_k at each synaptic contact k that connects the neurons "pre" and "post". The dynamics of q_k are defined as:

$$\begin{aligned} d/dt\, q_k(t) \;=\; & a_+\, S_{\text{post}}(t)\, T_{\text{pre}}{}^{\text{fast}}(t) + a_-\, S_{\text{pre}}(t)\, T_{\text{post}}{}^{\text{fast}}(t) \\ & - 1/\tau\, q_k(t) + \sigma \xi_k(t) - f_{\text{h}}(T_{\text{post}}{}^{\text{slow}}(t)) \end{aligned}$$

Here $\tau = 3620.0\ \text{s}$ is the integration time constant of the correlation trace q. The parameters $a_+ = -a_- = 31.54$ scale the Hebbian terms in response to coincident pre- and postsynaptic activity, on a time scale $\tau_{\text{fast}} = 50\ \text{ms}$ of the activity traces $T_i^{\text{fast}}(t)$. If seen as a STDP window function [4], a_+ is the scale of the weight change for positive time differences $t_{\text{post}} - t_{\text{pre}}$, and a_- the scale of the negative part. Furthermore, we add a Gaussian white noise term $\xi(t)$, scaled by $\sigma = 379.75 / \sqrt{(\ \text{s})}$, and a heterosynaptic term similar to [10]

$$f_{\text{h}}(T_{\text{post}}{}^{\text{slow}}(t)) \;=\; a_{\text{h}}\,([(1/\tau_{\text{slow}})\, T_{\text{post}}{}^{\text{slow}} - \nu_0]_+)^2$$

where $\nu_0 = 10 /\ \text{s}$ is the baseline firing rate and $a_{\text{h}} = 13.88\ \ \text{s}$ is a scaling factor, and $[.]_+$ denotes rectification. The term $T_{\text{post}}{}^{\text{slow}}(t) / \tau_{\text{slow}}$ estimates the firing rate of the postsynaptic neuron by low-pass filtering the spikes $S_{\text{post}}(t)$ with a time constant of $\tau_{\text{slow}} = 10\ \text{s}$.

Note that here $q_k(t)$ is not the synaptic weight, but merely an internal synaptic state variable with an arbitrary scale, as in a previous model [8]. Synaptic weights in this model derive from the state of the synaptic contacts as

$$w(t)\ \text{ prop to }\ x(t)$$

where $x(t)$ is the number of active contacts. Apart from being in the active state, contacts can also be in inactive or potential states. However, each contact can only be in one state at a time. The possible state transitions of each contact k are:

- Creation: potential contact becomes inactive contact. Occurs randomly, according to a Poisson process with rate $\lambda\text{c} = 9.6{=}10 - 7 /\ \text{s}$. Creation resets qk to 0.

- Maturation: inactive contact becomes active contact. Triggered if $\mathrm{qk} \geq \mathrm{Dm} = 22.30\tau\ /\ \mathrm{s}$. Maturation resets qk to Dm.
- Shrinkage: active contact becomes inactive contact. Triggered if $\mathrm{qk} \leq \mathrm{Ds} = -15.91\tau\ /\ \mathrm{s}$. Shrinkage resets qk to 0.
- Pruning: inactive contact becomes potential contact. Triggered if $\mathrm{qk} \leq \mathrm{Dp} = -10.68\tau\ /\ \mathrm{s}$. Pruning resets qk to 0.

These four state transitions of the model represent, respectively, the morphological changes of dendritic spine formation, maturation, shrinkage and removal. Potential contacts do not form the correlation trace $q_k(t)$, because they are assumed to lack the biomolecular apparatus to do so.

We simulate this plasticity model for pairs of neurons. The presynaptic spike train $S_{\mathrm{pre}}(t)$ is modelled as a stationary Poisson process with rate $\nu_0 = 10\ /\ \mathrm{s}$, the postsynaptic neuron as an inhomogeneous Poisson process with time-dependent rate

$$\nu_{\mathrm{post}}(t) = \nu_0 + m\, x(t)\, T_{\mathrm{pre}}{}^{\mathrm{m}}(t)\,,$$

so that it filters the input spikes using an exponential filter with time constant $\tau_{\mathrm{m}} = 50\ \mathrm{ms}$. In this model, each presynaptic spike transiently increases the postsynaptic rate by $m\ x(t)$, where $m = 0.5\ /\ \mathrm{s}$ is the effective synaptic effect on the postsynaptic rate per active contact and presynaptic spike.

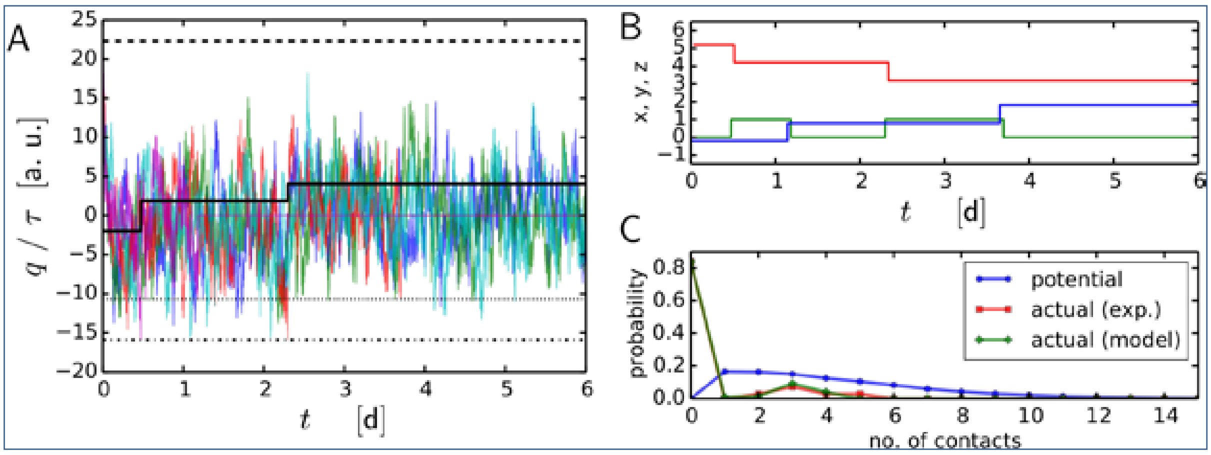

Figure 3: A-B: Simulation of the synaptic plasticity rule with parameters for cortical layer 4 connections

Initially, all contacts are in the active state, with $\mathrm{qk}(0) = \mathrm{Dm}$. A: correlation traces $\mathrm{qk}(\mathrm{t})$, $1 \leq \mathrm{k} \leq 5$, (colours) for a synapse consisting of 5 synaptic contacts. Thresholds are shown in grey: Dm (dashed), Ds (dash-dotted), Dp (dotted). The solid black line shows the theoretically expected mean level of qk, which depends on the number of active contacts $\mathrm{x(t)}$. B: Numbers of active (x, red), inactive (y, green) and potential (z, blue) synaptic contacts in the same simulation as in A. The lines were slightly shifted to improve visibility. C: Theoretically expected steady-state distribution of the number of contacts for this parameter set, compared to experimental data [3]. The parameters have been chosen to reproduce this data, as well as additional criteria concerning the temporal dynamics of spine formation and removal.

Model codes for two algorithms (ALGO STDPorchestrated and ALGO STDPStructural) are available on the UNIC-CNRS server – the other algorithms will be posted later this year and will be included in the final D4.6.4 Deliverable in March 2016.

Data provenance for ALGO *STDPstructural*

This algorithm concerns a new model of structural synaptic plasticity. The research was done at EPFL-LCN. A previous analysis [8] of the synaptic connection of neocortical neuron pairs, which typically consists of multiple synaptic contacts, established that the distributions of actual synapse numbers can be reproduced by structural plasticity based on spike timing. However, the study [8] did not propose a specific unsupervised learning rule to implement that mechanism. Here, guided by the results of [8], we developed a candidate STDP algorithm driving the morphological changes of individual excitatory synaptic contacts. The experimental data is the same as used in [8].

Valdiation work for ALGO STDPstructural

We have performed preliminary simulations of the algorithm for three different types of neocortical connections and varying numbers synaptic contacts, of which we show just one example in the Figure above. These simulations were run for 106 sec ≈ 12 days of biological time, simulated in time steps of 1 ms. Synaptic contacts were updated every 1 sec. Much longer simulations are required to confirm that the model reproduces the experimental distributions of the number of synaptic contacts in cortical microcircuits in the steady state (Figure C), as predicted by our theoretical analysis (not presented here). Furthermore, our analysis predicts that synaptic contact lifetimes and turnover ratios in this model are on the same order of magnitude as found experimentally [6, 2]. However, much longer simulations are necessary to confirm these theoretical results in direct simulations of the model for very long time (100–1,000 days).

Platform requirements for ALGO STDPstructural

The plasticity algorithm presented here relies on filtered traces of the pre-and postsynaptic activity with several time scales. A general framework for computing and accessing such traces at the synapse would be a very useful addition to simulation software such as NEST [7], because most spike-timing dependent plasticity rules can be formulated as a functional of such traces. At present, however, such a generic infrastructure is not available in NEST. Therefore, our prototype implementation computes the synaptic state and $\mathrm{qk}(t)$ updates outside of NEST, in the main loop of a python control script. However, to facilitate interfacing with the simulation Platform, we suggest that NEST developers include a framework for synaptic traces in the NEST software. This request was communicated in the cross-subproject workshop at the EITN in Paris on 2-4 March (Are we building the right thing: requirements from theory to simulation environments and neuromorphic computing) and was well received by the simulation Platform developers.

The model is a local synaptic plasticity rule and can be implemented in any spike-based neuronal network simulation. However, the dynamics of structural plasticity are slow, and so very long simulations are necessary to observe its effects. Hence very efficient simulations, and large computing resources, are required. The simulations presented in the Figure, of ≈ 12 days biological time, comprising 20 pairs of neurons, connected by 1, …, 20 synaptic contacts, respectively, took approximately one day to complete, using 8 CPUs in parallel.

The model would also be suitable for simulation on the Neuromorphic Computing Platform, as soon as the next generation of neuromorphic chips with on-chip synaptic traces becomes available. The neuromorphic chip offers the desirable speed-up compared to biological time to allow extensive simulation of the model.

Cross-SP aspects for ALGO STDPStructural

Synaptic spine turnover, also called structural synaptic plasticity, is an important phenomenon in the brain, for even adult networks continuously rewire themselves. For long-term simulation of cortical circuits, it is thus an important ingredient. The model presented here can explain how the distributions of synapse numbers in experiments can come about through continuously active synaptic plasticity. Such data on synaptic contact numbers are used in the HBP Brain Simulation Platform (SP6) to reconstitute the connectivity of real cortex in a detailed biophysical model. Our ALGO STDPStructual can serve as theoretical underpinning of the employed network structure by making connectivity a result of on-going plastic changes.

However, very long simulations are necessary, which poses a significant challenge to simulation technology. Accelerated neuromorphic hardware, on the other hand, can simulate for long durations much faster. The Neuromorphic Computing Platform (SP9) might thus be helpful to observe the long-term effects of structural plasticity models in networks in future research.

Structural plasticity may also be an important mechanism to increase memory capacity of cortical circuits [1]. In particular, in motor learning tasks, massive structural changes (spine formation) have been observed in animal experiments [9]. Thus also learning agents or robots controlled by neuronal networks might eventually benefit from realistic structural plasticity models (SP10).

Neuroinformatics links for ALGO STDPStructural

This model will be implemented in NEST once the required infrastructure of pre- and postsynaptic activity traces becomes available. As NEST is part of the neuroinformatics Platform, the model will then be available to all users of the Platform.

4.3 Task 4.2.3 Structures of spiking learning algorithms

4.3.1 SURREY (Andre Gruning, Task Leader)

ALGO MultilayerSpiker. The importance of precise spike timing in neural and cognitive information processing has been indicated in a variety of studies. For example, in the olfactory system, the precision of spike-timing has been associated with accurate odour-classifications, and populations of auditory neurons are known to signal input features by the relative timing of spikes. However, we lack learning algorithms for spiking networks that take full advantage of such a temporal coding scheme and yet retain the technical efficiency and versatility of more traditional, artificial neural network learning algorithms such as back propagation. To address this, we propose a learning algorithm for training spiking neural networks containing hidden layer neurons to encode for spatio-temporal spike patterns using a fully temporal coding scheme, such that spatio-temporal input patterns are identified by the timings of multiple timed spikes across one or several output neurons, rather than just single output spikes, as in many previous schemes.

ALGO MultilayerSpiker considers a feed forward multilayer spiking neural network containing a single hidden layer. Mechanistically, output weight updates result from a product of locally available pre- and postsynaptic activity terms that bears a close resemblance to Hebbian-like learning; the presynaptic term originates from filtered hidden neuron spike trains, and the postsynaptic term from an output error signal that guides the direction and magnitude of weight changes. Hidden weight updates, however, appear as a three-factor rule: filtered input spike trains are first combined with hidden spike trains, to then be modulated by a linear combination of back-propagated error signals to effect persistent weight changes. Such back-propagated error signals appear to mimic the functionality of a diffusive neuromodulator such as dopamine, lending biological support to our spiking analogue of back propagation. More biologically plausible implementations, in which the targeted back-propagated error signal is replaced with summary error information identical to all hidden neurons, also show a good technical performance.

Finally, the directly supervised nature of the scheme can be traded off, with performance against provision instead of a reinforcement signal only.

A more detailed, illustrative description of the multilayer learning rule can be found in [1]. Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

Data provenance for ALGO MultilayerSpiker

The ALGO MultilayerSpiker extends the single-layer learning rule from [2] to multiple layers by combining the methods of gradient ascent/descent and backpropagation. Specifically, a stochastic neuron model for distributing output spikes is considered, which has the advantage of allowing for the determination of the likelihood of generating a prescribed output spike pattern. From taking the log-likelihood, a suitable objective function which has a smooth dependence on network parameters can be formed, upon which gradient ascent can be taken with respect to output and hidden layer weights. Hence, in a manner analogous to backpropagation for rate-coded networks, hidden weight updates for spiking neurons can be rigorously derived. The mathematical formulation of MultilayerSpiker can be found in [1].

Validation Work for ALGO MultilayerSpiker

The algorithm has been tested thoroughly over a wide range of benchmark learning tasks, and has been found to successfully solve the linearly non-separable Exclusive-OR (XOR) computation, encode for a very large number of spatio-temporal spike patterns and generalise well on a synthetic dataset. We also demonstrate that the introduction of a hidden layer makes complicated tasks more easily trainable in spiking neural networks, compared to the classical two-layer architecture.

In our approach, we use a fully temporal coding scheme to perform input classifications, i.e. classifications are performed based on the timings of multiple rather than just single output spikes. Importantly, we find that multiple-spike-based codes counteract background noise, thereby resulting in increased classification performance. Our results compare favourably with previous studies, especially in terms of the network storage capacity, number of spikes in output trains, number of spike-train input-output mapping that can be learnt and encourage future investigation of multi-spike based encoding methods. Results are shown in [1] and acknowledge HBP support.

ALGO MultilayerSpiker offers strong promise as a general-purpose learning algorithm for spiking networks. It would be of great future interest to test this algorithm on datasets from other HBP Subprojects, e.g. from SP3 (Cognitive Architectures) and SP4 (Mathematical and Theoretical Foundations of Brain Research), apply it to problems in Neurorobotics (SP10 and SP11), and also implement it on the Neuromorphic Computing Platform (both Spinnaker and the Heidelberg neuromorphic systems) as a reference and to benchmark its performance against alternative spiking algorithms.

Platform requirements for ALGO MultilayerSpiker

These requirements pertain mainly to the Neuromorphic Computing Platform in SP9 (Physical model and Many Core model) as well as to the development of NEST, Neuron, STEPS (SP6):

1) Stochastic Neuron Models: MultilayerSpiker relies on the implementation of the stochastic ‘exponential escape rate’ neuron model, Chapter 5 in [3]. This model effectively simulates leaky integrate-and-fire neurons subject to background synaptic noise, such that output spikes are instead distributed according to an underlying, instantaneous firing rate at each neuron. Currently, this neuron model is not included as standard in the NEST environment: we would therefore welcome its addition by the

NEST developer team, and especially given its increasing popularity for formulating spiking learning algorithms.

2) Modulatory Signals: As a supervised or reinforcement scheme MultilayerSpiker also requires the delivery of modulatory signals to individual neurons that describe to witch degree accumulated eligibility traces from pre- and postsynaptic activity are to become actual long-term changes of the synaptic weights. Hence provision for the delivery of such “reward” signal, possibly temporally and spatially filtered to neurons or a group of thereof is desirable in the neuromorphic systems.
3) Pre-/postsynaptic Traces Finally, similar to other learning schemes proposed in this document, MultilayerSpiker also requires access to filtered traces of spike events at pre- and postsynaptic synapses in the neuromorphic systems. Ideally the filter function should not be the classical exponentially decaying STDP-window, but flexible and adjustable, at least to other observed forms of spike-timing dependent plasticity (such as Mexican hat forms, tripartite etc), but ideally freely adaptable.

Cross-SP aspects for ALGO MultilayerSpiker

ALGO MultilayerSpiker offers strong promise as a general-purpose learning algorithm for spiking networks. It would be of great future interest to test this algorithm on datasets from other HBP Subprojects, e.g. from SP3 (Cognitive Architectures) and SP4 (Mathematical and Theoretical Foundations of Brain Research). We would therefore welcome data sharing with dataset providers.

We want to see MultilayerSpiker applied in SP10 and SP11. Its technical performance outperforms previous learning algorithms for spiking neural networks with respect to the number of pattern mappings that can be learnt and the number of spikes these may contain. This makes it an ideal tool to form neural networks that are used to model cognitive tasks in SP3 and, more especially, to train neuro-inspired robots and agents in SP10 and applications in SP11.

We want to see this algorithm implemented on the Neuromorphic Computing Platform (both the Spinnaker and physical model neuromorphic systems within SP9) as a reference and benchmark its performance against alternative spiking algorithms.

Neuroinformatics Links for ALGO MultilayerSpiker

MultilayerSpiker will be implemented in NEST and the Neuromorphic systems once the required infrastructure of pre- and postsynaptic traces and modulatory signals becomes available. Our algorithms will then be made available through the inclusion into NEST within the Neuroinformatics Platforms (SP5). Preliminary MatLab or C/Python source codes will be made available trough an open source repository.

For the time being, [1] contains a detailed reproducible description of MultilayerSpiker.

5. WP4.3 - Large-scale and Cognitive Models

This WP is structured in four Tasks. The first Task is about models of perception-action, the second is about models of working memory and attention, the third is about generating models of different brain states, and the fourth is about network models that include glial cells. These Tasks are examined in turn below.

5.1 Task 4.3.1 Models for perception-action

5.1.1 UPF (Gustavo Deco, Task Leader)

Our purpose aim is to develop a dynamical model of activity of the whole cortex, activity to reproduce activity patterns observed during rest or when performing a task (e.g., using fMRI, EEG, MEG). The tuned model is then used as a fingerprint of the brain activity, to discriminate between brain states, categories of processing, etc.

What the model captures is effective connectivity (EC), namely the interaction strengths between neural populations in a network. In this sense, EC describes, for the a given levelconfiguration of structural connectivity (SC, physical connections), the corresponding strengths, which depend in the biology on neurotransmitter types, receptor concentration, etc. The aim of our project is to infer the macroscopic EC from observed activity. This will be used to constrain more detailed models that aim to reproduce a given correlated pattern of cortical activity in a given system. Our current focus is on estimating white-matter intracortical connections, to help model the whole cortex activity. In the near future, we expect to use human fMRI data from our collaboration with the group of Prof. Stanislas Dehaene (WP3.1) to study how effective connectivity is modified for various states of wakefulness. Later, we also plan to investigate the between-layer connectivity that applies to within a “cortical column”.

The method has been applied so far on data from the lab of Prof. Petra Ritter (BCCN Berlin) and Morten Kringelbach (Oxford). The data set associated with the present code are is a surrogate version of those. In parallel, a non-human data set (monkey) has been obtained from the group of Prof. Stan Dehaene (WP3.1) in the context of our collaboration and is currently being processed.

Meanwhile, the algorithm has been checked on surrogate networks by generating activity from known underlying network connectivity, and checking whether the algorithm can retrieve that original connectivity. The focus has been on directed connectivity, which requires the use of time-shifted covariance in addition to zero-shift covariance. Details can be found in a recently submitted publication (available in https://www.dropbox.com/sh/7tkm9g0afwrd5v5/AAAC2BmKW-OQXHVkzRqayCQ9a?dl=0xxx).

The model is a nonlinear version of Ornstein-Uhlenbeck processes that are recurrently coupled. As an illustration, we provide here a minimum model that describes the neural/MRI activity for each population with a synaptic variable x_i, which obeys the following equation: $\frac{dx_i}{dt} = \frac{x_i}{\tau} + \varphi\left(\sum_j C_{ij} x_j\right) + \vartheta_i$ where φ is a nonlinear activation function that applies to the contributions from other populations, and ϑ_i is white noise. The intracortical connectivity (EC) by the matrix C_{ij}. FC is the second-order moments of the variables x_i , namely $\langle (x_i - \bar{x}_i)(x_j - \bar{x}_j) \rangle$ where the bar indicates the mean over an observation period; the angular brackets denote the averaging over the noise in the network, related to the white noise. The matrices FC and EC are related via a Lyapunov equation, which allows for the derivation of an optimization method to adjust EC such that the model reproduces a desired FC. The code for this gradient-descent algorithm is included in the deliverable; the “how to use” file details the role of the different scripts.

5.1.2 UCL (Neil Burgess):

Space and Memory Lab (Neil Burgess), UCL (London, UK) is developing models of navigation and spatial decision-making. The goal is to identify, theoretically and computationally, the roles of the hippocampus and basal ganglia in navigation, by examining experimental results in rodents and humans that illuminate the distinct representations and learning rules present in each area. To achieve this, we are simulating the activity in the relevant areas during navigation of the simulated agent using neural networks of firing rate-based neurons. Each unit of the network represents a small subpopulation of neurons that encodes information specific to the area to which it belongs: locations in hippocampus, value in ventral striatum, and perceptions and movements in neocortex. A firing rate model with synaptic inputs describes the behaviour of each neuronal pool. The various areas are connected as shown in the image below.

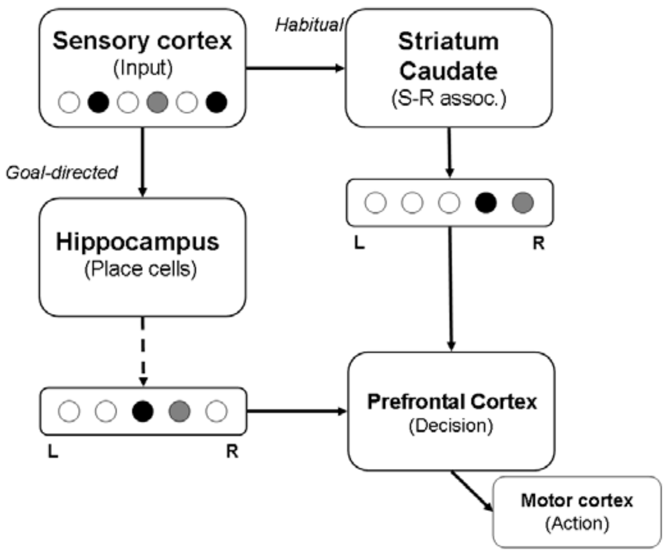

Figure 4: The various areas are connected as shown in a firing rate model with synaptic inputs

The whole code is written in (Visual) C++. This choice has been motivated by the need to obtain high performance on standard computers.

The data we have modelled derives from two publications: Pearce *et al.* (1998) Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. Nature 396, 75-77; Packard and McGaugh (1996) Inactivation of hippocampus or caudate nucleus with lidocaine differentially affects expression of place and response learning. Neurobiol Learn Mem. 65:65-72.

The validation of our result is still in progress.

In order to run our algorithms with more biological realism (spiking instead of firing rate-based neurons) and higher speed, we would need a high-performance computer on which it is possible to run generic code (not just spiking neuron networks).

At the beginning, we will utilize the neuronal population density simulator developed by Partner P110 (Marc De Kamps, ULEEDS) to run fast versions of our rodent navigation task. We will also collaborate with Partner P1 (Marc-Oliver Gewaltig, EPFL) to run our software exploiting high-performance computing architectures in closed-loop configuration. We will receive a Spinnaker Board from Partner P73 (David Lester, UMAN) to run accelerated spiking neuron versions of our Task. On this board, we will implement neuron models developed by Partner P7 (Alain Destexhe, CNRS). Once the Task is running correctly in our own simulator, we will collaborate with Partner P53 (TUM) to implement it in their newly developed physics engine and, later, on a real robot.

5.1.3 INRIA (Olivier Faugeras)

In the last period, Inria has continued investigating a model of the sensitivity of V1 to visual orientation. The model assumes that the pinwheels are arranged in a doubly periodic lattice. These pinwheels are the points where the orientation preference map is discontinuous.

This idealization naturally introduces symmetries in the problem, which make deeper analysis accessible. As long as the pinwheels are nearly arranged on a periodic lattice, we can expect that the main conclusions of our analysis will be still valid for the ”real” lattice. Combining the existence local and long-range connections between neurons in V1, using the fact the strength of long-range connections is significantly weaker than that of local connections, which allows for treating the long-range connections as a perturbation of the local ones, and exploiting the symmetries of the network of pinwheel, we have been able to investigate the bifurcation structure of this model and predict experimentally observable patterns. We have investigated in detail two hypotheses concerning the symmetries. One is the square symmetry, the other the hexagonal symmetry. In the first case, we have been able to provide a complete analysis of the possible bifurcations and to classify all spontaneous activity patterns. The second case is still under investigation but we have already obtained a number of significant results. A software package allowing simulation of the model has been developed and made available to the HBP community. A report has been uploaded to HAL: https://hal.inria.fr/hal-01079055v3.

The report has also been submitted for publication to the Journal of Mathematical Neuroscience. The first round of reviews has been favourable and we are in the process of revising the paper.

5.2 Task 4.3.2 Models of working memory and the effects of attention

5.2.1 WIS (Misha Tsodyks, Task Leader)

In this Task we are pursuing two goals. The first goal emerged from our earlier **synaptic theory of working memory** (Mongillo et al, 2008), according to which information is maintained in working memory by means of selective short-term facilitation of recurrent connections within a neuron population encoding a given item. A natural question that emerges in this picture is how many items can potentially be stored in a working memory. This ‘working memory capacity’ is well studied in classical psychological studies of working memory but its neuronal underpinnings are still unclear. We are currently looking ata network of interacting neuronal populations, each representing a different memory item. The activity of each population is described as in (Mongillo *et al.*, 2008):

$$\begin{aligned}
\tau\frac{dE}{dt} &= -E + g(JuxE + E_0) \\
\frac{du}{dt} &= \frac{U-u}{\tau_F} + U(1-u)E \\
\frac{dx}{dt} &= \frac{1-x}{\tau_D} - uxE
\end{aligned}$$

where E stands for the firing rate of a population and u and x are facilitation and depression variables, respectively. Different populations are connected by weak excitatory connections, but the activity of the whole network is controlled by global inhibition, so the effective interaction between the populations is inhibitory, which prevents them from being activated simultaneously. When an item is loaded into working memory by the activation of the corresponding population, the connections of this population become facilitation that may lead to subsequent spontaneous reactivation, or 'recall' from working memory. Now we consider the situation where several items are loaded consecutively. If the number of loaded items is not too big, they may then be reactivated spontaneously one by one, as shown in Figure 4 below, in the case of two items.

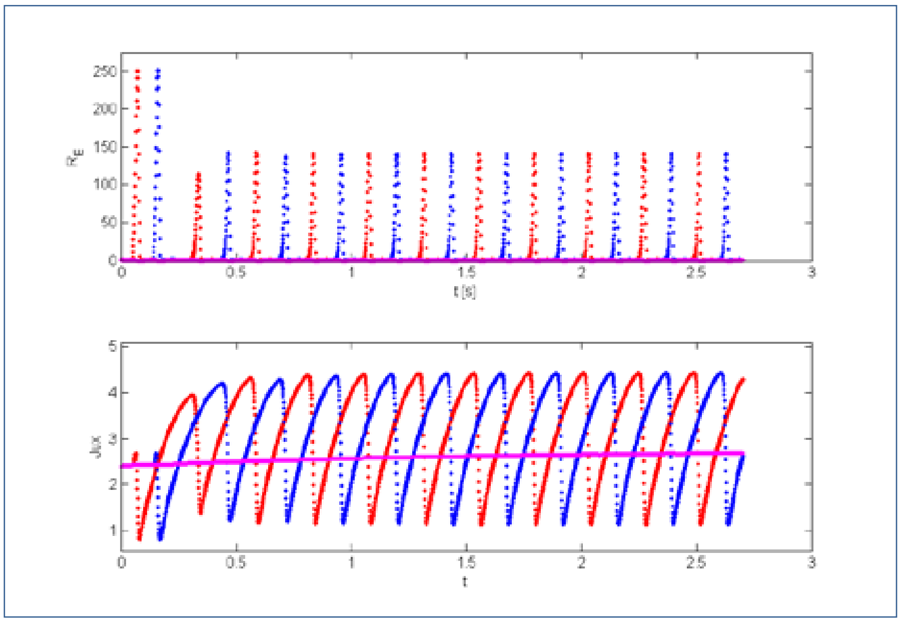

Figure 5: Loading and reactivation of two items into a working memory network with short-term synaptic facilitation. Upper panel shows the firing rates of two memory populations, the lower panel shows the corresponding effective synaptic strength for the same populations.

The activity of the corresponding populations is shown in red and blue colours (top panel): first two peaks show the loading and subsequent peaks represent the recall. The lower panel shows the trace of synaptic efficacy.

The main question that we address is how many items can be recalled in the network (capacity) and how the capacity depends on the parameters of the network. To study these issues, we simulate the network with different initial conditions and search for limit cycles corresponding to periodic reactivation of different items. We also try to estimate analytically the capacity by computing the largest possible time between two consecutive reactivations of a single item. Finally, we study the possible effects of attention by considering the effect of external excitatory input on the capacity of the working memory. We found that increasing the external excitation can increase the capacity to a certain degree, which could be the neuronal mechanism of the attentional effects on working memory. Our code for simulating the network is currently under development, but we expect to make a more robust version available soon to the HBP Consortium.

The second goal being investigate within the current Task relates to the neural network implementation of free recall. As in the previous model, we simulate a network of interacting populations representing different items in memory. However, we are interested in how a memory can be recalled from the long-term form directly into the active form, without working memory. To this end, we simulate the dynamics of the network, without synaptic facilitation and with periodically modulated global inhibition strength. The idea is that when the first item is recalled (reactivated), it may trigger the recall of the next one if the overlap between the corresponding neuronal representations is big enough. The question we address is again one of capacity, namely how many items can be recalled from memory depending on the total number of items to be recalled and parameters of the network. Some of the features of this model resemble classical observations on free recall in the psychological literature. An example of the network behaviour is shown in the Figure 5, which simulates the subject attempting to recall a list of 16 words.

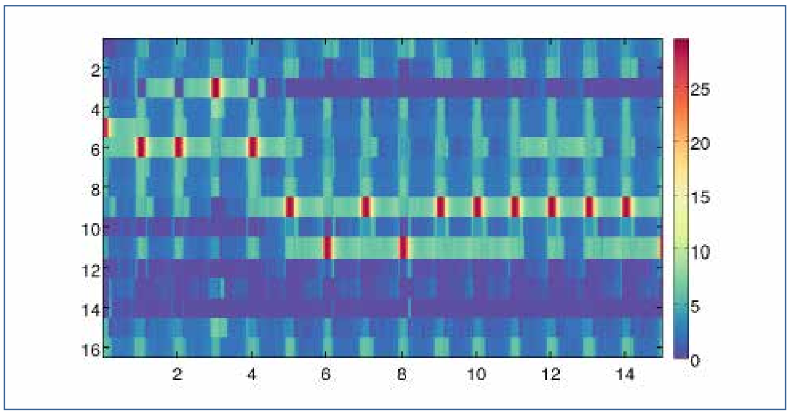

Figure 6: Free recall of a list of 16 words in attractor neural network with periodically modulated global inhibition

The activity of the corresponding encoding populations is shown in colour code. Out of 16 memorized words, only 4 were recalled in 14 cycles of oscillation.

Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

5.3 Task 4.3.3 Models of biologically realistic network states; wakefulness & sleep

5.3.1 CNRS (Alain Destexhe, Task Leader)

Task 4.3.3 started 6 months ago. The UNIC Partner started to develop spiking models of different brain states, starting directly from models that are fully compatible with the neuromorphic hardware. We are designing 2D topographic balanced networks using NEST and PyNN. These networks are characterized by adaptive exponential (AdEx) integrate-and-fire neurons and conductance-based synapses with a Gaussian profile for the probability of connections. The parameters are tuned to be within the regime in which the networks dynamics show self-sustained asynchronous-irregular (AI) activity states. An exploration of the dynamics of the network in regimes with different parameters and with different connectivity structures has been started, to reproduce the profile of pair-wise correlations observed experimentally.

As soon as these models are ready, the simulation codes will be posted in the Human Brain Atlas.

5.3.2 UPF (Gustavo Deco)

For the past 6 months, the UPF Partner (one post-doc) has started using a whole cortical dynamic model to study the influence of wakefulness/sleep on functional connectivity (FC), namely correlated activity observed in fMRI/BOLD signals. The goal of our model-based approach is to infer from experimental data what determines the changes of correlation patterns between all cortical areas. We focus on a network model where nodes are canonical Hopf bifurcations. In the model, the FC changes can arise from the modifications of the intracortical connectivity and/or the intrinsic variability of the cortical areas. The nodes can exhibit two types of intrinsic variability: the Hopf-bifurcation model can switch from noise diffusion (cf. Task 4.3.1) to oscillatory behaviour. We collaborate with the group of Prof. Stanislas Dehaene (WP3.1) to obtain experimental data to feed our models; we have started using a non-human (monkey) data set with different levels of sedation. Human data set with both sleep and awake states will be collected later. The simulation code will be made available via the HBP Platform in due course.

5.3.3 JULICH (Abigail Morrison):

A PhD student at JUELICH has worked on this Task as a parallel project since October 2014. The idea is to model the cell-type-specific effects of acetylcholine, a neuromodulator that influences cortical states, on the dynamics of a cortical microcircuit. Information on cell-type-specific effects of acetylcholine on the single-neuron level is taken from the literature and is complemented with findings from experimentalists at JUELICH. The model consists of leaky integrate-and-fire neurons divided into excitatory and inhibitory populations in the different cortical layers, forming the full-scale network under about 1 square mm of cortical surface. It is based on a model by Potjans and Diesmann (Cerebral Cortex, 2014) and is implemented in NEST. First simulations suggest that acetylcholine tends to desynchronize the network and to reduce its mean firing rate in the spontaneous condition mainly through the depolarization of inhibitory neurons. Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

5.4 Task 4.3.4 Computational model of astrocyte-neuron interaction for future large-scale simulations

5.4.1 TUT (Maja-Leena Linne, Task Leader)

MODEL DEVELOPERS: Marja-Leena Linne, Tiina Manninen, Ausra Saudargiene, Riikka Havela

BIOLOGICAL BACKGROUND: Astrocytes represent the main homeostatic element of the central nervous system. Moreover, they are capable of sensing and secreting neurotransmitters and neuromodulators, and are thus integral parts of the chemical and electrical transmission system operating in neural networks in the brain. Different cortical areas in mammals are shown to contain large amounts of astrocytes, many times equal to the number of neurons. Our aim is to study the effects of different biochemical and biophysical mechanisms for the integrated functioning of tripartite synapses that involve three elements: the presynaptic terminal, the postsynaptic terminal and the astrocyte. The ultimate goal of our work is to understand better the involvement of astrocytes in information transmission and in different forms of plasticity in the brain. Our modelling work will provide us with information about the time scales of astrocyte events and, consequently, on bridging scales from molecular level events up to cognitive level events. In order to clarify these issues, we have previously done two intensive reviews that have been used as the basis for model selections in the present work [4, 5].

THE MODEL: The model is a detailed biophysical excitability model that involves postsynaptic, presynaptic and astrocyte terminals of the so-called tripartite synapse. The model describes key cell membrane and intracellular calcium processes shown to take part in the information transfer in the synapse. We did not use a simple leaky-integrate-and-fire type of approach for spikes, but instead used descriptions of voltage-gated conductance to better assess the time scales of events associated with cell membrane level events and events that occur inside the three terminals. Moreover, the model includes key interactions between postsynaptic terminal and the astrocyte. Because of the complexity of the biophysical processes, our strategy was to build on top of existing modelling projects. Our hypothesis is that a retrograde signalling mechanism, such as the one provided by endocannabinoids, is required in addition to neuro- and glio-transmission to fully understand synaptic information transfer in a longer term. One of the few models that describe both pre- and postsynaptic signalling, including the release of endocannabinoids, is the study by Zachariou et al. [10] . Moreover, the only existing astrocyte model that can take endocannabinoid as input is by Wade et al. [9] . These two modelling projects were used as the basis of our new model. As a result of the model implementation, we are able to simulate a variety of calcium-related signalling mechanisms in a tripartite synapse and assess their role in information transfer under various stimulation protocols. This is expected to clarify certain controversies in the field, such as the involvement of astrocytes in the induction and/or maintenance of long-term potentiation and depression (see, e.g., [1, 6]). The model simulations will also shed light on the time scales at which certain events occur at the tripartite synapse, as well as help bridging scales in the future neural network models involving simplified versions of the present tripartite synapse model.

In short, we have selected to implement the following well-known mechanism of the three synaptic elements (for code see file in the EITN database: T434_D463_tripartitesynapse.m, which can be run in Matlab version R2010b):

Presynaptic terminal:

1) Conventional cell membrane ionic currents (transient sodium current, delayed rectifier potassium current),

2) Fraction of willing calcium channels,

3) Fraction of bound G proteins (cannabinoid receptors).

Postsynaptic terminal:

1) Conventional cell membrane ionic currents (sodium current, delayed rectifier type potassium current),

2) L-type calcium current,

3) AMPA receptor current,

4) GABA receptor current,

5) PMCA pump (plasma membrane Ca2+ ATPase),

6) Intracellular calcium buffer,

7) SERCA pump (sarco/endoplasmic reticulum Ca2+-ATPase).

Astrocyte terminal:

1) SERCA pump (sarco/endoplasmic reticulum Ca2+-ATPase),

2) Leak current into the cytoplasm from the endoplasmic reticulum,

3) IP3 receptor channel.

VALIDATION OF THE MODEL: The sub-modules for each terminal were first validated separately. The astrocyte-neuron equations provided by Wade et al. [9] did not match the simulation results provided in the original publication. However, since we were only interested in the astrocyte terminal of Wade et al. [9] model, a part that is commonly known as Li-Rinzel model [3] , we can strongly assume that the equations are exceptionally well validated. Additional validation of astrocyte responses were done based on results of

[7]. Equations taken from Zachariou et al. [10] for the pre- and postsynaptic terminal were partly modified, for the reason that not all equations and parameter values are provided in the original publication. There exist very few datasets for validating the model equations for intracellular signalling, but we partly used our previous modelling work to validate the models [2, 5, 8]. The excitability models for both pre- and postsynaptic terminals were considered as prototypical models that have been validated extensively in the past.

COLLABORATIONS WITHIN THE HBP: The model will be validated with HBP SP6, and collaborations with SP6 have been sought out. Possible simplification of the model can be done with other Tasks in SP4.

PUBLICATIONS: We received funding in the Competitive Call and started our work only 8 months ago. The presented work is thus work-in-progress and will be continued during HBP Months 19-30. There are several publications planned, but not yet published, but dissemination via conference posters and oral presentations has already started.

The codes for the present version of the model are available on the UNIC-CNRS server, and will be submitted to HBP Brain Atlas repository as soon as it is ready.

6. WP4.4 - Principles of Brain Computation

This WP is structured in three Tasks. The first Task investigates principles of computation in single neurons and microcircuits; the second Task is about novel computing systems inspired from biology, and the third Task is about closed loop analysis of population coding. These Tasks are described below.

6.1 Task 4.4.1 Principles of computation in single neurons and neural microcircuits

6.1.1 TUGRAZ (Wolfgang Maass, Task Leader):

TUGRAZ has identified a remarkable computational property of another common microcircuit motif: interconnected ensembles of pyramidal cells with lateral inhibition. It was shown through theoretical analysis and computer simulations that this microcircuit motif can emulate Bayesian filtering (or particle filtering), one of the most powerful tools known for integrating information over time in a close-to-optimal manner. Furthermore, a substantial number of experimental data that address integration of information in the cortex can be modelled on the basis of this computational paradigm. As a methodology, this new computational model provides an improvement of the well-known neural sampling paradigm, that is also implemented in neuromorphic hardware: one can now allow that random variables are represented by the firing activity of ensembles of pyramidal cells, rather than by single cells. This makes the resulting computational model more robust against noise and failures of units. The results of this research were published in R. Legenstein and W. Maass, Ensembles of spiking neurons with noise support optimal probabilistic inference in a dynamically changing environment. PLOS Computational Biology, 10(10):e1003859, 2014.

Model codes will be made available later this year and will be included in the final D4.6.4 Deliverable in March 2016.

6.1.2 CNRS (Alain Destexhe)

In Task 4.4.1, the UNIC Partner studied the computational properties of dendrites subject to noise (representing *in vivo* conditions), and in the presence of local excitability properties (dendritic spikes). The models used (Fizhugh-Nagumo and cellular automaton) were developed in Task 4.1.1 and the code is given there. This code can be used to simulate the main finding of this model: models with excitable dendrites are inversely sensitive to correlations, compared to point neuron models. This can be simulated by introducing correlations in the codes provided, and see how correlations affect the firing activity. We describe these results below in more detail (joint work with Mathieu Galtier, Romain Veltz and Alain Destexhe).

6.1.2.1 Cancellation of colliding dendritic spikes

Two dendritic spikes propagating in opposite directions will cancel out when they collide. Actually, this can be shown for any kind of non-linear dynamics governing the behaviour of the dendritic compartments. The proof can be made rigorous and goes as follow: consider the precise position where the two dendritic spikes collide. For simplicity, assume that is occurs at the separation between two compartments. The linear cable equation (responsible for propagation) says the first compartment is influencing the second by a factor of the difference between the two activities (and vice versa). However, because we are at the point where spikes collide, both compartments have the same activity and therefore do not communicate. The propagation is therefore stopped and the activity of each compartment will eventually fade away. Note that this relies highly on the homogeneity of the dendritic spikes: they have to be of the same shape.

The fact that the dendritic spikes display a refractory period is not necessary for the proof above. However, we believe it will make the phenomenon more robust to noise and also make the result valid in more heterogeneous cases.

Figure 6 illustrates how several dendritic spikes can interact destructively when they re-generated simultaneously at different positions in the dendrite. From this illustration, it is clear that the larger the correlation, the more numerous the collisions.

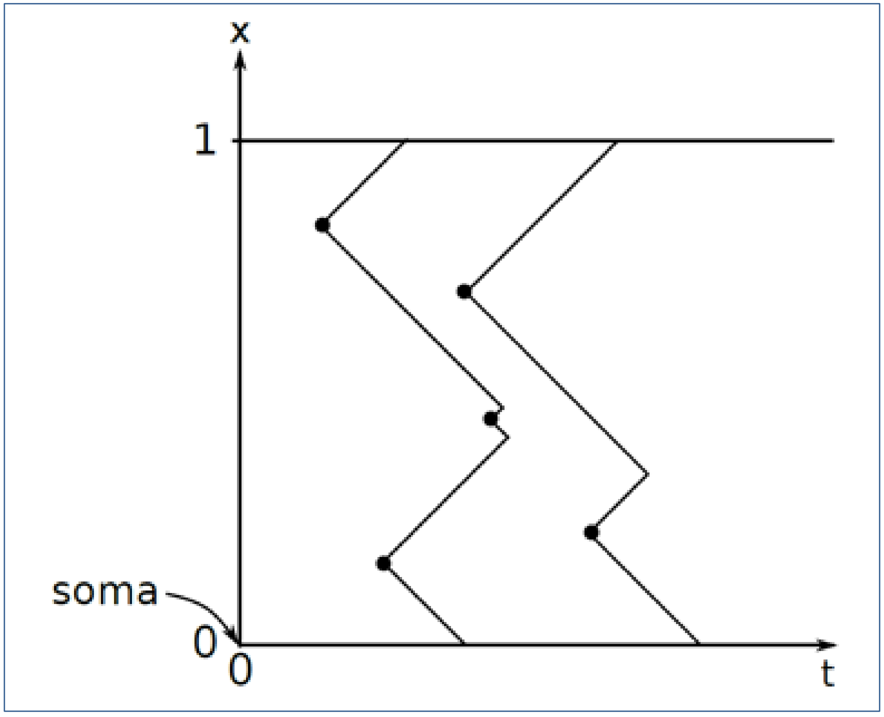

Figure 7: Example of destructive spikes interaction in cellular automaton model of excitable dendrites

The x-axis represents time. The y-axis represents the spatial extension of the ball and stick neuron considered. Black dots correspond to dendritic spike emission. Oblique lines correspond to the propagation of spikes along the dendrite.

6.1.2.2 Mathematical proof

We have designed a simple cellular automaton model that reproduces the phenomena explained above, and we have been able to compute explicitly the expectation of the number of dendritic spikes reaching the soma as a function of the correlation level between input spike trains (the parameter l).

The cellular automaton model consists in discretizing space and time for the ball and stick neuron. Each compartment can be in four states: rest, emission of dendritic spike, dendritic spike propagating to the left, and dendritic spike propagating to the right. The rule of the cellular automaton, corresponding to 64 cases, embodies the notions of propagations and cancellation of colliding dendritic spikes.

We have deployed sophisticated mathematical tools, such as the theory of partially ordered sets and advanced probabilistic methods, in order to compute the desired result. We were able to find an analytic, though complicated, expression for the expectation of the number of dendritic spikes reaching the soma as a function. From this expression, it is easy to show that the former decreases with the latter.

6.1.2.3 Numerical simulations

We have done numerical simulations, where each compartment was modelled as an excitable system. We first considered Fitzhugh-Nagumo models, the dynamics of which correspond well to the behaviour of dendritic spikes. These numerical simulations compare the behaviour of a point neuron with the behaviour of a ball and stick neurons (with active dendrites) when exposed to correlated synaptic bombardment (Fig. 7). As expected, we observe two different behaviours: while the activity of the point neuron increases with correlation, the activity of the ball and stick neuron decreases with correlation. Thus, we highlight a qualitative feature of active dendrite: they filter out correlations.

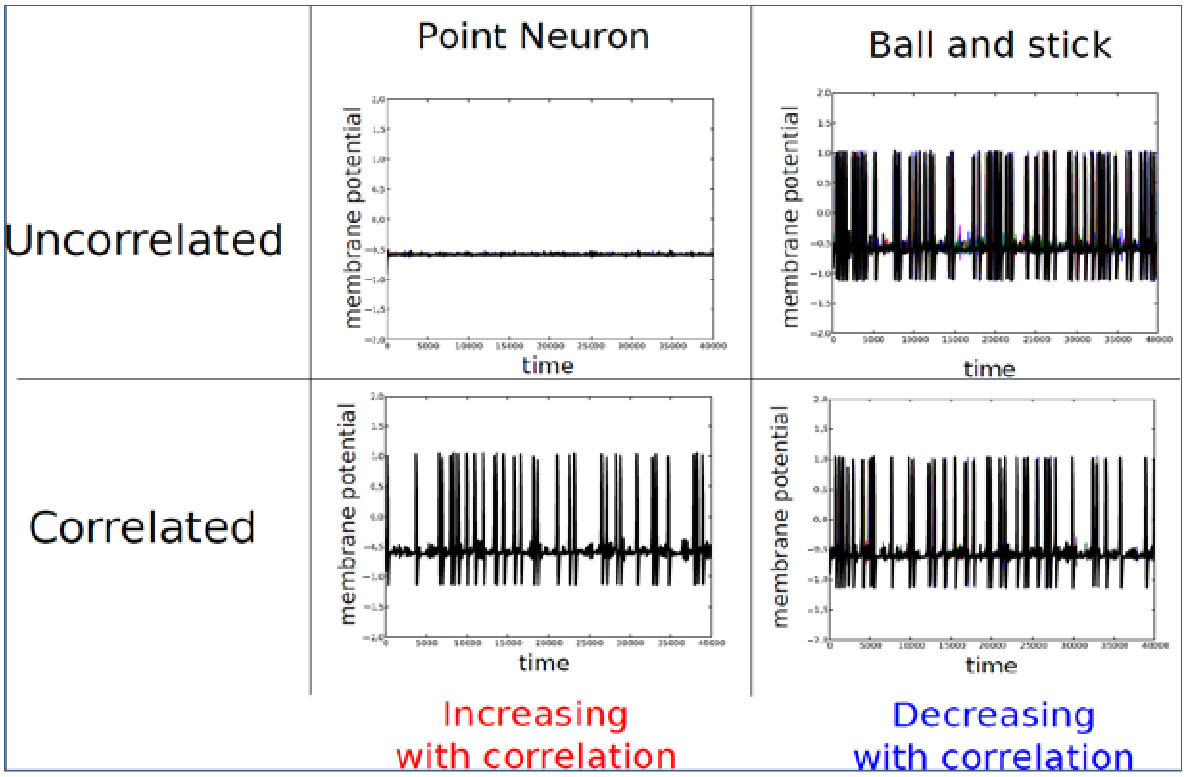

Figure 8: Qualitatively different correlation processing in neurons with and without dendrites

The Fizhugh-Nagumo model was used to simulate spikes, in response to a set of synapses releasing randomly, without (top) and with (bottom) correlations. In a point neuron (left), the firing increases with correlation, but in a model with dendrite (ball-and-stick, right), correlations decrease the firing of the neuron.

It is important to note that these simulations can be easily extended to Adaptive Exponential models, which are implemented in the neuromorphic hardware. In that sense, we believe that the phenomenon we put forward here will be of much help for designing biological neural networks, with dendrites, in neuromorphic computers.

An additional point revealed by these simulations is that the optimal environment for generating a large number of somatic spikes is to have locally correlated synapses (so as to generate a lot of dendritic spikes) and globally uncorrelated synapses (so as to avoid dendritic spike collisions). Such interactions between local and global correlations are presently under study.

6.1.2.4 Conclusion

In this theory-driven study, we argue that active dendrites *in vivo* filter out correlations. This puts in perspective the generally held idea that dendrites are correlation detectors. We do not call into question that precisely timed stimulations can reinforce each other. However, we argue that, in the situation of dense and random synaptic bombardment, such precise events rarely occur. Rather, we believe that correlations between spike trains induce a lot of correlations between dendritic spikes and thus a lot of cancellation.

Models codes are the same as for Task 4.1.1, and they are available on the UNIC-CNRS server.

6.2 Task 4.4.2 Novel computing systems inspired by biology

6.2.1 UGENT (Joni Dambre, Task Leader)

In embodied computation (or morphological computation), part of the complexity of motor control is offloaded to the body dynamics. UGent investigates how embodied computation in low-level motor control can be achieved using reward-modulated Hebbian learning. In our previous work, gait motor patterns for a compliant tensegrity robot were first learned using evolutionary optimisation (CMA-ES). Direct proprioceptive feedback was then used to mimic those desired, using noise-based supervised Hebbian learning. Here, the reward signal directly expressed the similarity to the desired actuator signals, which is not biologically plausible.

UGent has now used reward-modulated Hebbian learning in a more biologically plausible setting to achieve end-effector control for the tensegrity robot. For each targeted trajectory, the actuators are driven by a sequence of approximate actuator signals that could have been generated from a self-organised static predictive body model. As these control signals do not take the body dynamics into account, the resulting end-effector trajectories bore little resemblance to the desired ones. Again, using reward-modulated Hebbian learning, a linear feedback controller (using analogue neurons as an approximation for rate-codes) was trained to correct the actuator signals and achieve the right trajectory. Between the proprioceptive signals and the feedback neurons, a decorrelation layer was added. The reward, representing the average deviation from the desired trajectory, was presented after each run of a trajectory was completed. Our algorithm, which is analogous to existing work in the context of neural networks, adds exploration noise to the actuator signals. By correlating the effects of the noise and the resulting change in reward, the learning rule attempts to improve the average reward.

The results of this work were submitted for a special issue of Frontiers in Neurorobotics (draft version available with the source code). As our robotics experiments were performed using our in-house robotics simulator, the code submitted for D4.6.3 consists of an illustration of the learning rule in which a neural network is trained to drive an untrained neural network is used as dummy for a body. The script and library (Python code) can be used to represent one of the experiments described in the submitted paper.

At present, we are also evaluating our model for self-organised gait generation. The next stages will consist of bringing the different parts of the model (reward estimation, learning, decorrelation, etc.) closer to biological plausibility.

This work is related to SP10 (Neurorobotics) as it involves motor control for compliant bodies.

To access to these data within the Neuroinformatics Platform, the following link is available https://github.com/jdambre/HBP_D4.6.3.git

6.2.2 TUGRAZ (Wolfgang Maass)

For Task 4.4.2 (Novel computing systems inspired by biology), TU Graz has developed and validated new design methods that enable networks of spiking neurons to solve difficult constraint satisfaction problems. These methods were validated through extensive computer simulations for a number of spike transmission delays, and a threshold was determined for spike transmission delays below which the network can be expected to work close to the theoretical optimum. The University of Manchester has already implemented in Spinnaker a related simpler algorithm developed earlier at TU Graz for solving Sudoku puzzles in networks of spiking neurons. The new results from TU Graz will allow the University of Manchester and the University of Heidelberg also to develop now spike-based networks for approximately solving well-known (and practically relevant) NP hard-constraint satisfaction problems, such as SATISFIABILITY (3SAT), that are relevant for logical inference and verification.

The new design methods and algorithms, as well as validation results, are described in detail in the report Z. Jonke, S. Habenschuss, and W. Maass. A theoretical basis for efficient computations with noisy spiking neurons. *arXiv.org*, arXiv:1412.5862, 2014. A publication will be submitted shortly.

Software for implementing these algorithms are available on the UNIC-CNRS server.

6.3 Task 4.4.3 Closed loop analysis of population coding

6.3.1 UPMC (Olivier Marre, Task Leader)

6.3.1.1 Retinal model

Our purpose is to provide realistic models of the retina, whose output can be used as an input for thalamo-cortical models. Part of this work is also done in collaboration with the team of Ryad Benosman to put these models in a neuromorphic chip (see e.g. Akolkar *et al.*, 2015).

Here we provide a simple matlab code to simulate the response of a basic retina to complex stimuli. This model will be improved stepwise as we make further progress in this project.

In this matlab script, the user needs to specify the stimulus it wants to display line 4. This should a 3D array (X, Y and time). We assume that the display has a constant refresh rate.

The parameters of the retinal model are set between lines 10 and 20.

Once the script is run, it should give back two variables: lambda and Response. Lambda contains the firing rate over time for each cell, while Response contains the number of spikes at each time point.

7. How Can Platform Developers Provide You With Feedback on The Data?

The contact persons are primarily the Task leaders and HBP partners as enumerated above. We suggest using the collaboration portal to keep track of the exchanges between SP4 and the Platforms. The most useful feedback of course concerns how the Platform developers can integrate the models developed here, and what SP4 partners can do to facilitate such an integration.

An important issue is the format in which the models should be exchanged. At the moment, it seems that Python appears as the most-used format for model specification and simulation (through Python frontends of different simulators such as NEURON, NEST or BRIAN). In particular, we try to formulate our models using the PyNN description, because PyNN codes are compatible with many Platforms, such as NEURON, NEST, BRIAN and neuromorphic hardware.

Some more specific feedback is provided below.

7.1 ALGO STDPorchestrated T4.2.1

The contact person is Friedemann Zenke at EPFL-LCN. In order to facilitate interfaces, our main request to Simulation Platform developers is that a framework be developed for a class of STDP rules that is implemented as a combination of presynaptic traces and postsynaptic traces (see description of traces in Section 2). In particular, such a framework for pre-and postsynaptic traces should be provided in the NEST simulator.

7.2 ALGO STDPStructural T4.2.2

The contact person is Moritz Deger at EPFL-LCN. In order to facilitate interfaces, our main request to Simulation Platform developers is that a framework be developed for a class of STDP rules that is implemented as a combination of presynaptic traces and postsynaptic traces (see description of traces in Section 2). In particular, such a framework for pre-and postsynaptic traces should be provided in the NEST simulator.

7.3 ALGO STDPpredictive T4.2.1

The contact person is Dominik Spicher at UBERN

7.4 Population density models

Contact:

- M.deKamps@leeds.ac.uk
- Initially, we expect installation problems. We are prepared to help with them. We are also greatly interested in comments on the proposed workflow.

7.5 ALGO MultilayerSpiker

The contact person is André Grüning at SURREY, [a.gruning@surrey.ac.uk]. In addition to the points made for the other algorithms in this section, traces of reward and feedback signals should be made available for use in the Simulators and the Neuromorphic systems (see Section 2), as well as for more flexible neuron models.

Annex A: implementation details of synaptic algorithms

Learning algorithms summarize rules of synaptic plasticity in a concise format that should enable simulation of plasticity phenomena and relate the results to the function of 'learning'. Learning algorithms come in different forms, depending on the functional context.

ALGO STDPorchestrated T 4.2.1 - Friedemann Zenke (EPFL-LCN) has been carrying out this task, Wulfram Gerstner is responsible.

ALGO STDPStructural T 4.2.2 - (EPFL-LCN). Moritz Deger is carrying out this task, Wulfram Gerstner is responsible.

ALGO STDPpredictive T 4.2.1 - (UBERN). Dominik Spicher is carrying out this task, Walter Senn is responsible.

ALGO MultilayerSpiker: Brian Gardner and Ioana Sporea (SURREY) have been carrying out this task, André Grüning (SURREY) is responsible

Annex B: Implementation Details of MIIND

B1. Installing MIIND

Some third party software is required, or MIIND cannot be installed. In all but one case, the required software is available through standard package managers:

Prerequisites:

- Gnu Scientific Library (GSL). Any recent version will do.
- Boost C++ libraries, version 1.54 or later. When MPI is required, the boost MPI libraries must be available. This is not built by default.
- g++ 4.7 or later. There are problems with the Intel compiler.
- CMake 2.8 or later.
- Python 2.7. It is assumed that Matplotlib and SciPy are available for analysis of simulation results.
- Root (https://root.cern.ch/drupal/). For now, we recommend version 5.34

It is strongly recommended to build ROOT from source, rather than obtain it through a package manager, as some libraries are not built by default. Building ROOT by source has itself some perquisites that are listed here: https://root.cern.ch/drupal/content/build-prerequisites . On most platforms, package managers are able to deliver them quickly.

After download and extraction of the ROOT source tar ball, which can be placed anywhere, there will be a main directory called ‘root’. There is a configure script there, which should be run with the following enabling options: --enable-table, --enable-explicitlink, --enable-python, --enable-mathmore.

The configure script will generate a Makefile and typing ‘make’ will start the build process. The install then is idiosyncratic, but this is the recommended procedure by the ROOT developers: leave the code in place where it is compiled. In the shell (and later presumably in .bashrc, or equivalent) run:

```
‘source ~/root/bin/thisroot.sh’
```

In this example, it is assumed that the user has extracted the ROOT tar ball into his/her home directory. The command should of course be adapted to match the actual extraction directory.

After this, some environment variables have been defined that allows the MIIND install procedure to find ROOT, wherever it was placed. The success of the previous procedure can be tested simply by typing ‘root’ in the shell where the source command was run. A splash screen should pop up, and a ROOT shell should become available.

Assuming this has been successfully completed, MIIND can be installed. The current version of MIIND can be obtained with:

```
git clone git://git.code.sf.net/p/miind/git miind-git
```

After execution, there will be a directory called mind-git. One should ‘cd’ there, and create a directory ‘build’, then ‘cd build’. In that directory ‘ccmake..’ will bring a GUI to configure the simulation. In the first instance set CMAKE_BUILD_TYPE to ‘Release’, and switch the testing off. ENABLE_MPI can be left to ‘OFF’ at first, as the demonstrators are not very CPU intensive. The GUI has a configure option ‘C’, and upon successful configuration, there will be a ‘G’ option available. It may be necessary to hit ‘C’ twice, before this happens. After hitting the ‘G’ option, ccmake will exit and a Makefile will be available. Now type ‘make’. MIIND should be build. If the building process completes at 100% the MIIND libraries and demonstrators will have been built.

B2. Running the Demonstrators

Assuming the current working directory is the ‘build’ directory where the install was performed, there will now be new directories present. The MIIND libraries are present in the ‘libs’ directories, the demonstrators in ‘apps/PerformanceGeom’. It is necessary to create a ‘test’ directory before running the programs described below, the demonstrators will exit ungracefully when it is not present.

Present are:

- LifOneCanvas. A single population of leaky-integrate-and-fire (LIF) neurons at rest at the beginning of the simulation, then after $t=0$ each neuron in the population receives Poisson distributed spike trains with a rate of 800 Hz. The simulation will show a running canvas. The right hand panel describes the population density of the population. The left-hand panel the output firing rate of the population as a response to input.
- LifTwoCanvas. A fully connected circuit of two LIF populations: one excitatory, one inhibitory driven by an external stimulus, in a state of balanced excitation-inhibition
- QifOnePoisson. The quadratic-integrate-and-fire (QIF) equivalent of LifOneCanvas.
- QifCorrelated. A single population of QIF neurons spiking coherently, initially. Over time, they are decorrelated by a low frequency input with high efficacy. This is a prime of example of a simulation that could not be performed by rate-based models (see http://arxiv.org/abs/1309.1654) .

In apps/largeNetwork an executable is present that simulates local cortical circuits arranged in a hexagonal network. The number of rings in the network is configurable (in the source code; requires recompilation) and allows the generation of large networks, so as to test MPI performance.

All networks can be run using MPI. MIIND has to be compiled with MPI_ENABLED switched to ‘ON’ (this can be done in the CMake GUI). Then you need to run your executables with the local version of ‘mpirun’.

B3. Analysing Simulation Results

Running the miind executable will produce a ’.root’ file. You can analyse the results conveniently in Python, using either the ROOT objects directly, or converting them into numpy objects that can be analysed in numpy, scipy and visualized with Matplotlib.

Copy the root file to a directory of your choice. Make sure that the PYTHONPATH variable is set to pick up the ROOT module. Open a Python shell in your favouring Python environment and type

```
import ROOT
f = ROOT.TFile( ’yourfile.root’ )
```

You can inspect the file:

```
f.ls()
```

You will get a list of names of TGraph objects. To get them from the file, do, for example:

```
g = f.Get( 'rate_1' )
g.Draw( 'AP ' )
```

A canvas with the firing rate graph should now pop up.

ROOT is a very powerful analysis environment, with more extensive visualization capabilities than Matplotlib, but nothing prevents you from using SciPy for your analysis. Consider a ROOT.TGraph object with name ‘data’:

```
x_buff = data.GetX()
y_buff = data.GetY()
N = data.GetN()
x_buff.SetSize(N)
y_buff.SetSize(N)
# Create numpy arrays from buffers, copy to prevent data loss
x_arr = np.array( x_buff , copy=True )
y_arr = np.array( y_buff , copy=True )
```

B4. Workflow as Envisaged

Consider the ‘miind-git’ directory that was created upon cloning the git repository. There is a ‘python’ subdirectory. If you type ‘python mind.py xml/omurtag.py’ you will find that when you go to the ‘build’ directory and type ‘make’ again, there will be a new executable, called ‘omurtag’. The XML file has been converted into a C++ executable, which is now part of the build structure. You can do this with other XML files as well.

The XML files are human readable, almost self-documenting definitions of a simulation. In the future, we envisage researchers creating XML files like these, submitting their simulations to the Brain Simulation Platform through a web interface, and receiving a message telling them when and where they can download the simulation results in the form of ROOT files. This clearly requires extra work, in conjunction with the simulator Platform.

B5. Current and Future Plans

It is important that to get across that MIIND can simulate really large networks. Currently, the emphasis has been on validation the simulation techniques. Over the next weeks, we will collect a large number of showcases, demonstrating that MIIND can be used to reproduce published results easily. This compilation will form the basis of a tutorial that is currently being written.

Annex C: References

References from Marja-Leena Line

[1] Agulhon C., Fiacco T.A., and McCarthy K.D. Hippocampal short- and long-term plasticity are not modulated by astrocyte Ca2+ signaling. Science 327(5970):1250-4, 2010.

[2] Hituri K. and Linne M.-L. Comparison of models for IP3 receptor kinetics using stochastic simulations. PLoS ONE, 8(4): e59618, 2013.

[3] Li Y. and Rinzel J. Equations for InsP3 receptor-mediated calcium oscillations derived from a detailed kinetic model: a Hodgkin-Huxley like formalism. Journal of Theoretical Biology 166, 461–473, 1994.

[4] Linne M.-L. and Jalonen T.O. Astrocyte-neuron interactions: From experimental research based models to translational medicine. Progress in Molecular Biology and Translational Science 123:191-217, 2014.

[5] Manninen T., Hituri K., Hellgren-Kotaleski J., Blackwell K. T., and Linne M.-L. Postsynaptic signal transduction models for long-term potentiation and depression. Frontiers in Computational Neuroscience 4:152, 2010.

[6] Navarrete M., Perea G., Fernandez de Sevilla D., Gomez-Gonzalo M., Nunez A., Martin E.D., and Araque A. Astrocytes Mediate In Vivo Cholinergic-Induced Synaptic Plasticity. PLoS Biology 10(2): e1001259, 2012.[7] Shigetomi E., Kracun S., Sofroniew M.V. and Khakh B.S. A genetically targeted optical sensor to monitor calcium signals in astrocyte processes. Nature Neuroscience 13, 759–766, 2010.[8] Toivari E., Manninen T., Nahata A.K., Jalonen T.O., and Linne M.-L. Effects of transmitters and amyloid-beta peptide on calcium signals in rat cortical astrocytes: Fura-2AM measurements and stochastic model simulations. PLoS ONE 6(3): e17914, 2011.[9] Wade J.,McDaid L., Harkin J., Crunelli V. and Kelso S. Self-repair in a bidirectionally coupled Astrocyte-Neuron (AN) system based on retrograde signaling. Frontiers in Computational Neuroscience 6:76, 2012.[10] Zachariou M., Alexander S.P.H., Coombes S., and Christodoulou C. A Biophysical Model of Endocannabinoid-Mediated Short Term Depression in Hippocampal Inhibition. PLoS ONE 8(3): e58926, 2013.

References from Andre Gruning

[1] Gardner, Brian, Ioana Sporea, and André Grüning. “Encoding Spike Patterns in Multilayer Spiking Neural Networks.” Submitteed. Preprint arXiv:1503.09129 (2015). http://arxiv.org/abs/1503.09129

[2] Pfister, J.P., et al. “Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning.” Neural computation 18.6 (2006): 1318-1348.

[3] Wulfram Gerstner, Werner M. Kistler: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, 2002.

References from Wulfram Gerstner

[1] Andreas Knoblauch, Günther Palm, Friedrich T Sommer: “Memory capacities for synaptic and structural plasticity.”, *Neural Comput*, pp. 289—341, 2010.

[2] Anthony J G D Holtmaat, Joshua T Trachtenberg, Linda Wilbrecht, Gordon M Shepherd, Xiaoqun Zhang, Graham W Knott, Karel Svoboda: “Transient and persistent dendritic spines in the neocortex in vivo.”, *Neuron*, pp. 279—291, 2005.

[3] D. Feldmeyer, V. Egger, J. Lubke, B. Sakmann: “Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single 'barrel' of developing rat somatosensory cortex.”, *J Physiol*, pp. 169—190, 1999.

[4] Wulfram Gerstner, Werner M. Kistler: *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[5] Haruo Kasai, Masahiro Fukuda, Satoshi Watanabe, Akiko Hayashi-Takagi, Jun Noguchi: “Structural dynamics of dendritic spines in memory and cognition.”, *Trends Neurosci*, pp. 121—129, 2010.

[6] Joshua T Trachtenberg, Brian E Chen, Graham W Knott, Guoping Feng, Joshua R Sanes, Egbert Welker, Karel Svoboda: “Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex.”, *Nature*, pp. 788—794, 2002.

[7] Marc-Oliver Gewaltig, Markus Diesmann: “NEST (NEural Simulation Tool)”, *Scholarpedia*, pp. 1430, 2007.

[8] Moritz Deger, Moritz Helias, Stefan Rotter, Markus Diesmann: “Spike-timing dependence of structural plasticity explains cooperative synapse formation in the neocortex.”, *PLoS Comput Biol*, pp. e1002689, 2012.

[9] Tonghui Xu, Xinzhu Yu, Andrew J. Perlik, Willie F. Tobin, Jonathan A. Zweig, Kelly Tennant, Theresa Jones, Yi Zuo: “Rapid formation and selective stabilization of synapses for enduring motor memories.”, *Nature*, pp. 915—919, 2009.

[10] Friedemann Zenke, Guillaume Hennequin, Wulfram Gerstner: “Synaptic plasticity in neural networks needs homeostasis with a fast rate detector.”, *PLoS Comput Biol*, pp. e1003330, 2013.

[11] F. Zenke and W. Gerstner. Limits to high-speed simulations of spiking neural networks using general-purpose computers, in Frontiers in neuroinformatics, vol. 8, p. 76, 2014.

[12] J.P. Pfister and W. Gerstner, Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity, *J. Neuroscience*, Vol. 26, Nr. 38, pp. 9673-9682, 2006.

[13] J.-Y. Chen et al. Heterosynaptic plasticity prevents runaway synaptic dynamics. J. Neuroscience 33, 15915-15929, 2013.

[14] G.G. Turrigiano, Homeostatic synaptic plasticity: Local and global mechanisms for stabilizing neuronal function. Cold Spring Harb Perspect Biol 4, a005736, 2012.

[15] F. Zenke, E. Agnes, W. Gerstner, A diversity of synaptic plasticity mechanisms orchestrated to form and retrieve memories of spiking neural networks. Nature Comm. , 2015.

References from Walter Senn

[1] Urbanczik, Robert, and Walter Senn. "Learning by the dendritic prediction of somatic spiking." *Neuron* 81.3 (2014): 521-528.

[2] Sjöström, Per Jesper, Gina G. Turrigiano, and Sacha B. Nelson. "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity." *Neuron* 32.6 (2001): 1149-1164.

[3] Ngezahayo, Anaclet, Melitta Schachner, and Alain Artola. "Synaptic activity modulates the induction of bidirectional synaptic changes in adult mouse hippocampus." *The Journal of Neuroscience* 20.7 (2000): 2451-2458.

[4] Sjöström, Per Jesper, and Michael Häusser. "A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons." *Neuron* 51.2 (2006): 227-238.