



Grant Agreement:	604102	Project Title:	Human Brain Project
Document Title:	Algorithms, Cognitive Models and Computing Principles for Implementation in HBP, Neuromorphic and Neurobotic systems: Package Two		
Document Filename:	SP4_D4.6.4_FINAL.docx		
Deliverable Number:	D4.6.4		
Deliverable Type:	Other		
Work Package(s):	WPs 4.1, 4.2, 4.3, 4.4 & 4.6		
Dissemination Level:	PU		
Planned Delivery Date:	M30/31 Mar 2016		
Actual Delivery Date:	M30/31 Mar 2016		
Authors:	Alain Destexhe, CNRS (P7), SP4 leader		
Compiling Editors:	Katherine Fregnac, CNRS (P7), SP4 manager		
Contributors:	<p>Idan Segev, HUJI, (P21)</p> <p>Tomasz Gorski, Bartosz Telenzcuk, Zahara Girones, CNRS (P7), T411, T412, T433</p> <p>Wulfram Gerstner, EPFL (P1), T411, T422, T433</p> <p>Gaute Einevoll, UMB (P67), T412</p> <p>Marc de Kamps, Yi Ming Lai, ULEEDS (P110), T413</p> <p>Walter Senn, UBERN (P62), T421</p> <p>Andre Gruning, SURREY (P111), T423</p> <p>Gustavo Deco and Matthieu Gilson, UPF (P65), T431, T433</p> <p>Misha Tsodyks, WIS (P78), T432</p> <p>Abigail Morrison, Andrei Maksimov, Sacha van Albada, JUELICH (P17), T433</p> <p>Marja-Leena Line, TUT (P99), T434</p> <p>Wolfgang Maass, TUGRAZ (P54), T441</p> <p>Joni Dambre, UGent (P66), T442</p> <p>Olivier Marre, UPMC (P107), T443</p> <p>Olivier Faugeras, Inria (P26), T431</p>		
Coordinator Review:	<p>EPFL (P1): Jeff MULLER, Martin TELEFONT</p> <p>UHEI (P45): Sabine SCHNEIDER, Martina SCHMALHOLZ</p>		
Editorial Review:	EPFL (P1): Guy WILLIS, Lauren ORWIN		
Abstract:			
Keywords:			



Table of Contents

1. Introduction.....	10
2. WP4.1 Bridging Scales.....	13
2.1 Task 4.1.1 Derive simplified neuron and neural circuit models from biophysically morphologically detailed models.....	13
2.1.1 2.1.1 I. Segev (HUJI) _ Reducing complexity of single neuron models	13
2.1.1.1 Introduction	13
2.1.1.2 Model Description	13
2.1.1.3 Model Data	14
2.1.1.4 Model Results	15
2.1.1.5 Provenance.....	16
2.1.2 A. Destexhe (CNRS) _ Model of dendritic integration with excitable dendrites.....	16
2.1.2.1 Introduction	16
2.1.2.2 Model Description	17
2.1.2.3 Model Data	18
2.1.2.4 Model Results	19
2.1.2.5 Provenance.....	21
2.1.3 W. Gerstner (EPFL) _ Derive simplified neuron and neural circuit models from biophysically morphologically detailed models	21
2.1.3.1 Introduction	21
2.1.3.2 Model Description	23
2.1.3.3 Model Data	24
2.1.3.4 Model Results	27
2.1.3.5 Provenance.....	29
2.2 Task 4.1.2 Modelling brain signals at different scales, from intracellular, local field potentials, and VSD up to EEG.....	29
2.2.1 A. Destexhe (CNRS) _ simplified models of local field potential	29
2.2.1.1 Introduction	29
2.2.1.2 Model Description	29
2.2.1.3 Model Data	30
2.2.1.4 Model Results	32
2.2.1.5 Provenance.....	34
2.2.2 G. Einevoll (UMB) _ Modelling brain signals at different scales.....	34
2.2.2.1 Introduction	34
2.2.2.2 Model Description for LFP generation from active-dendrite neurons	34
2.2.2.3 Model Data	37
2.2.2.4 Provenance.....	37
2.2.2.5 Model Results	37



2.3 Task 4.1.3 Mechanistic Models of Cognition Linked to the Neural Substrate by Population Density Methods	39
2.3.1 M. de Kamps (ULEEDS)	39
2.3.1.1 Introduction	39
2.3.1.2 Model Description for Miind	40
2.3.1.2.1 Introduction.....	40
2.3.1.2.2 Demonstration.....	41
Starting with MIIND	41
Validating your Installation.....	41
2.3.1.2.3 Running Example Simulations	42
2.3.1.2.4 Large-scale Networks.....	42
2.3.1.3 Model Data	44
2.3.1.3.1 Single Populations and Simple Circuits	44
2.3.1.3.2 A Universal 2D Population Density Solver	44
2.3.1.3.3 Population Density Techniques for non-Markov Interspike Intervals...	48
2.3.1.3.4 A Model of Sentence Ambiguity	48
2.3.1.3.5 Outlook	49
2.3.1.4 Results	50
2.3.1.4.1 Conductance-based results	50
2.3.1.4.2 Results for non-Poisson input spike trains.....	50
2.3.1.4.3 Results on scaling	51
2.3.1.5 Publications	52
2.3.1.6 Provenance.....	52
2.3.2 Task 4.2.1 Derive learning rules from biophysical synapse models	52
2.3.2.1 Introduction	52
2.3.3 W. Gerstner (EPFL)	52
2.3.3.1 Introduction - A).....	52
2.3.3.2 Model Description A) ALGO STDPorchestrated:.....	53
2.3.3.3 Model Data -A) ALGO STDPorchestrated:	53
2.3.3.4 Model Results A) ALGO STDPorchstrated:	55
2.3.3.5 Provenance.....	57
2.3.4 W. Senn (UBERN)	57
2.3.4.1 Introduction B) ALGO STDPpredictive:.....	57
2.3.4.2 Model Data B) ALGO STDPpredictive:	58
2.3.4.3 Model Results	59
2.3.4.4 Provenance.....	60
2.3.4.5 Introduction C)	60
2.3.4.6 Model description C).....	60



2.3.4.7	Model data C) ALGO STDPbackprop:	61
2.3.4.8	Model results C) ALGO STDPbackprop:	62
2.3.4.9	Provenance	62
2.3.5	M. Tsodyks (WIS)	62
2.3.5.1	Introduction D) ALGO LT-memory	62
2.3.5.2	Model description	63
2.3.5.3	Model data D) ALGO LT-memory	64
2.3.5.4	Model results D) ALGO LT-memory	65
2.3.5.5	Provenance	66
2.4	Task 4.2.2 Unsupervised learning rules and emergent connectivity	66
2.4.1	W. Gerstner (EPFL)	66
2.4.1.1	Introduction	66
2.4.1.2	Model Description for Algo STDP-structural	67
2.4.1.3	Model Data for Algo STDP-structural	69
2.4.1.4	Model Results	70
2.4.1.5	Provenance	72
2.5	Task 4.2.3 Structures of Spiking Learning Algorithms	72
2.5.1	A. Gruning (SURREY)	72
2.5.1.1	Introduction	72
2.5.1.2	Model Description MultilayerSpiker	73
2.5.1.3	Model Data	74
2.5.1.4	Model Results	75
2.5.1.5	Provenance	77
2.6	T4.3.1 Models of perception-action	77
2.6.1	Cognitive model of whole cortex _G. Deco (UPF)	77
2.6.1.1	Introduction	77
2.6.1.2	Model Description	78
2.6.1.3	Model Data	79
2.6.1.4	Model Results	81
2.6.1.5	Provenance	83
2.6.2	O. Faugeras (INRIA)	83
2.6.2.1	Introduction	83
2.6.2.2	Model Description	84
2.6.2.3	Model Data: Pinwheel V1 model	84
2.6.2.4	Model Results	84
2.6.2.5	Provenance	85
2.6.3	F. Chersi & N. Burgess UCL (WP3.4 and WP4.3)	85
2.6.3.1	Introduction	85



2.6.3.2	Model details.....	86
2.6.3.2.1	The visual system	86
2.6.3.2.2	Neural architecture.....	86
2.6.3.2.3	The striatal circuit.....	88
2.6.3.3	Results	89
2.6.3.4	Provenance of the data:	93
2.6.3.5	Collaborations and interactions with other partners:	93
2.7	Task 4.3.2 Models of working memory and the effects of attention	93
2.7.1	M. Tsodyks (WIS)	93
2.7.1.1	Introduction.....	93
2.7.1.2	Model Description	94
2.7.1.3	Model Data	95
2.7.1.4	Model Results	96
2.7.1.5	Provenance.....	97
2.8	T4.3.3 Models of biologically realistic network states; wakefulness & sleep	97
2.8.1	G. Deco (UPF) _ Cortical model of wake and sleep.....	97
2.8.1.1	Introduction.....	97
2.8.1.2	Model Description	97
2.8.1.3	Model Data	98
2.8.1.4	Model Results	98
2.8.1.5	Provenance.....	101
2.8.2	A. Destexhe (CNRS) _ Models of biologically realistic network states	101
2.8.2.1	Introduction.....	101
2.8.2.2	Model Description	102
2.8.2.3	Model Data	104
2.8.2.4	Model Results	105
2.8.2.5	Provenance.....	106
2.8.3	M. Diesmann/A. Morrison (JUELICH) _ Models of biologically realistic network states; wakefulness & sleep.....	107
2.8.3.1	Introduction.....	107
2.8.3.2	Model Description	107
2.8.3.3	Model Data	109
2.8.3.4	Model Results	110
2.8.3.5	Provenance.....	112
2.9	Task 4.3.4 Computational model of astrocyte-neuron interaction for future large-scale simulations.....	112
2.9.1	ML. Linne (TUT)	112
2.9.1.1	Introduction.....	112
2.9.1.2	Model Description: Astrocyte-neuron interaction model (ANI) model	113



2.9.1.3	Model Data	115
2.9.1.4	Model Results	117
2.9.1.5	Provenance	119
2.10	Task 4.4.1 Principles of computation in single neurons and cortical microcircuits.....	119
2.10.1	W. Maass (TUGRAZ)_ Brain-Inspired Computing Principles for Implementation in Neuromorphic Computing Systems.....	120
2.10.1.1	Introduction	120
2.10.1.2	Model Description.....	120
2.10.1.3	Spike-based Winner-Take-All network model	121
2.10.1.4	Synaptic sampling in spike-based Winner-Take-All networks as stochastic STDP	122
2.10.1.5	Simulation and data analysis.....	122
2.10.1.6	Model Data	123
2.10.1.7	Model Results	124
2.10.1.8	Provenance	126
2.10.2	A. Destexhe (CNRS) _ Intergrative properties at the single neuron level.....	127
2.11	Task 4.4.2 Novel computing systems inspired by biology	127
2.11.1	W. Maass (TUGRAZ).....	127
2.11.1.1	Introduction	127
2.11.1.2	Model Description.....	127
2.11.1.3	Model Data	128
2.11.1.4	Model Results	129
2.11.1.5	Provenance	130
2.11.2	J. Dambre (UGent)	131
2.11.2.1	Introduction	131
2.11.2.2	Model Description : Quadruped Embodied gait generation	132
2.11.2.3	Model Data	134
2.11.2.4	Model Results.....	135
2.11.2.5	Provenance	137
2.12	Task 4.4.3 Closed-loop analysis of population coding	137
2.12.1	O. Marre (UPMC)	137
2.12.1.1	Introduction	137
2.12.1.2	Model Description.....	137
2.12.1.3	Model Data	137
2.12.1.4	Model Results	138
2.12.1.5	Provenance	139
3.	Data Delivery	140
4.	Annex A: References	141
	Annex B : Publications.....	148
	Annex C: Installation Procedure	150



4.1.1.1.1	Third Party Dependencies	150
4.1.1.2	Procedure	150
4.1.1.2.1	A Clean Ubuntu Install.....	151

List of Figures and Tables

Figure 1:	Sixty-one 3D reconstructed L2/3 pyramidal cells from human temporal cortex. .	14
Figure 2:	Reduction scheme for complex neuron models (“Eq_Cyl”).	15
Figure 3:	Steps in reducing 3D reconstructed and physiologically characterized nonlinear dendritic tree into a simplified model (“Neuron_reduce”).	16
Figure 4:	Velocity of spike increases with the diameter of dendrite.	20
Figure 5:	Left: Collision of dendritic spikes. Right: propagation of dendritic spikes.....	20
Figure 6:	Number of spikes which reached soma as a function of correlation coefficient ..	21
Figure 7:	Electrode properties for data preprocessing.....	23
Figure 8:	GIF Model	28
Figure 9:	GIF Model (2).....	29
Figure 10:	Simulation of single neuron spikes	33
Figure 11:	Latency of st-LFP peak.....	34
Figure 12:	Active conductances can shape the extracellular signature of synaptic inputs..	38
Figure 13:	Resonance is retained for asymmetric input from distributed synapses.	39
Figure 14:	File structure of the MIIND installation.....	41
Figure 15:	Monitoring running simulations	42
Figure 16:	Network of two-population circuits.....	43
Figure 17:	the response of a population of leaky-integrate-and-fire neurons to an input that has been switched on at $t = 0$	44
Figure 18:	AdExp model.....	45
Figure 19:	Mesh for a conductance-based model	46
Figure 20:	The same neurons (the same mass) moves through the grid from cell to cell. ..	46
Figure 21:	A fraction of neurons in bin (5,5) will be translated vertically due to synaptic input, as input spikes cause a jump in conductance.	47
Figure 22:	the neural blackboard architecture models linguistic trees in terms of populations.	49
Figure 23:	the density at three different moment in time ($t = 0, 8, 28$ ms).....	50
Figure 24:	Population firing rate as calculated by the 2D method (red line) and inferred from Monte Carlo events (blue line) for a total input frequency of 900 spikes/s.	50
Figure 25:	Non-Poisson input spike trains	51
Figure 26:	Simulation time as function of the number of rings in the network.	51
Figure 27:	<i>Neural network activity.</i>	66
Figure 28:	<i>Temporal properties of recall.</i>	66
Figure 29:	Model overview.....	71



Figure 30: Dynamics of synaptic contacts in the steady state.	72
Figure 31: Schematic diagram of the data flow in our model-based approach to interpret cortical activity.	78
Figure 32:	79
Figure 33: Tuned network model to reproduce empirical FC obtained from fMRI data. See text for details.	81
Figure 34: Interpretation of results from Fig. 33. See text for details.	82
Figure 35: EC estimated for rest (left) and movie (middle). Significant changes in inputs (right) between movie and rest conditions.	83
Figure 36:	85
Figure 37: Schematic representation of the neural representation of the rat's view field.	86
Figure 38: "Value function" that encodes the position of a specific object through the superposition of multiple receptive fields.	87
Figure 39: Striatal circuit employed in this architecture that assigns (a value of) an action to each sensory input.	88
Figure 40: Left panel: Locations of the hidden platform (white circle) and the corresponding landmark (black circles). Right panel: Corresponding distribution of place cells across the water maze.	89
Figure 41: Decision scheme for the navigation task.	90
Figure 42: Different types of behaviors and trajectories.	91
Figure 43: Left panel: average performance of the rats expressed as the time required to reach the escape platform. Right panel: performance of real rats as recorded by Pearce et al. (1998)	92
Figure 44: Behaviour of rats lesioned in caudate nucleus or hippocampus	92
Figure 45: Short term plasticity based working memory network model.	96
Figure 46: Empirical FC for the three states of wakefulness.	99
Figure 47: Effective connectivity differences between anesthesia and awake states.	99
Figure 48. Data analysis.	100
Figure 49. Parameter exploration of whole brain model	101
Figure 50: topological balanced network with conductance-based adaptive exponential integrate and fire neurons and a probability of connection which decays with distance, according to a Gaussian profile.	103
Figure 51: Raster plot and average firing rate obtained from the network simulation for $\sigma_{exc} = \sigma_{inh} = 0.15$	105
Figure 52: Parameter exploration of the network	105
Figure 53: Correlation coefficient versus distance of E-E pairs (blue) and I-I pairs (red) for $\sigma_{exc} = \sigma_{inh} = 0.15$	106
Figure 54: Correlation coefficient as a function of shared inputs for different firing rates.	106
Figure 55: Distributions of excitatory synaptic weights in the model with (blue) and without (red) ACh.	108



Figure 56: Raster plot showing spike times of, from top to bottom, populations 2/3E, 2/3I, 4E, 4I, 5E, 5I, 6E, and 6I. 110

Figure 57: Effect of ACh on firing rate distributions of excitatory neurons. 111

Figure 58: ACh enhances thalamocortical responses in the model, quantified as the difference between average driven PSTH and spontaneous rates. 112

Figure 59: Simulation of the full ANI model by varying the frequency of the stimuli. 118

Figure 60: Simulation of the full ANI model by varying the frequency of the stimuli and the volume of the postsynaptic terminal (equivalent to spine). 119

Figure 61 Inherent compensation for network perturbations. 126

Figure 62: Emergence of working memory encoded in neural assemblies through approximate HMM learning in a WTA circuit through STDP. 130

Figure 63: Schematic of the control system. 133

Figure 64 Robot trajectories 136

Figure 65: Schematic of model used & example of response prediction 138

Figure 66: sensitivity (defined from the Fisher information matrix) as a function of the cell position relative to the stimulus. 139

List of Tables

Table 1: Parameters of the cortical microcircuit model..... 107



Algorithms, Cognitive Models and Computing Principles for the HBP Human Brain Atlas

1. Introduction

This Deliverable describes in details the work done between M18 and M30, and 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 made available as supplementary information to this Deliverable (to be posted to the Human Brain Atlas). Meanwhile, DIC have been filled in on the HBP collaborative portal:

Task	Partner	Data set	DIC name
4.1.1	HUJI	Reducing complexity of single neuron models	Simplified neuron model
4.1.1	CNRS	Model of dendritic integration with excitable dendrites	Model of dendritic integration with excitable dendrites



4.1.1	EPFL	Derive simplified neuron and neural circuit models from biophysically morphologically detailed models	Fitting Generalized Integrate-and-Fire models
4.1.2	CNRS	Modelling brain signals at different scales, from single neuron activity to local field potential	simplified models of local field potential
4.1.2	UMB	Modelling of the effect of subthreshold active dendritic conductances in shaping the cortical LFP	Links given
4.1.3	ULEEDS	Simulator MINDS: Mechanistic Models of Cognition Linked to the Neural Substrate by Population Density Methods	MIIND code repository
4.2.1	UBERN	ALGO STDPpredictive, ALGO STDPbackprop	ALGO STDPpredictive, ALGO STDPbackprop
4.2.1	EPFL-LCN	ALGO STDPorchestrated	ALGO STDPorchestrated
4.2.1	WIS	ALGO LT-memory: Neural network modeling of memory	Link to paper
4.2.2	EPFL-LCN	Unsupervised learning rules and emergent connectivity: ALGO STDP-structural	Not published yet
4.2.3	SURREY	MultilayerSpiker: new supervised learning rule for multilayer spiking neural networks	Link given, just published
4.3.1	UPF	Whole-cortex dynamical model	
4.3.1	UCL	Models of navigation and spatial decision-making: Navigation: roles of the hippocampus and basal ganglia in navigation	Rat navigation simulation
4.3.1	INRIA	Model to investigate sensitivity of V1 to visual orientation	Pinwheel V1 model



4.3.2	WIS	Short term plasticity model: neural network model for memory	
4.3.3	CNRS	Models of biologically realistic network states	Models of biologically realistic network states
4.3.3	JUELICH	Spiking model of a cortical microcircuit to investigate how acetylcholine (ACh) influences cortical network dynamics	Not published yet
4.3.3	UPF	Cortical model of wake and sleep	
4.3.4	TUT	Astrocyte-neuron interaction (ANI) model	Astrocyte-neuron interaction model
4.4.1	TUGRAZ	Cortical microcircuits through STDP and rewiring (spine dynamics) principles	Matlab code for synaptic sampling
4.4.2	UGENT	Embodied computation (or morphological computation)	Reward-modulated Hebbian Learning embodiment Embodied gait generation quadruped Mixture of ESN experts
4.4.2	TUGRAZ	Tool for autonomous learning of working memory, input prediction, and sequence learning through STDP in a generic cortical microcircuit motif	Matlab code for autonomous learning of working memory
4.4.3	UPMC	Closed-loop analysis of population coding: model can be used to provide realistic spiking input from the retina to models of the thalamo-cortical network	



2. WP4.1 Bridging Scales

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

2.1.1 2.1.1 I. Segev (HUJI) _ *Reducing complexity of single neuron models*

2.1.1.1 Introduction

The aim of this task is to develop systematic methods for reducing the complexity of single neuron models (often consisting of hundreds/thousands of compartments per neuron) into simplified models (consisting of several tens of compartments), while preserving the essential I/O properties of the detailed model. This is part of “Bridging scales” effort within SP4; the reduced models are then accessible for large-scale network modelling (WP6.1 and WP6.2) and for building neuromorphic models of neurons - consisting of dendrites and synapses (WP9.1). Below is a description of two newly developed reduction algorithms and a summary of their related results.

2.1.1.2 Model Description

(i) An “Equivalent cylinder”, preserving transfer resistance from any original synapse to the soma.

The first novel reduction model approach proposed hereby is based on the following concept: Given a detailed (3D reconstructed and physiologically characterized) neuron (e.g., our newly rare set of data of L2/3 pyramidal neurons from human temporal cortex, Fig. 1), one starts with the detailed passive cable model of this cell (Eyal et al., submitted). The performance of this model (its response to a variety of synaptic inputs and somatic current injections) serves as a benchmark for the quality of the reduced model. Next, the input resistance at the soma (R_{soma}) and the transfer resistance from all dendritic terminals, i , to soma ($R_{i,\text{soma}} = V_{\text{soma}}/I_i$) is computed analytically; seeking for the minimal $R_{i,\text{soma}}$ (the branch with maximal voltage attenuation to the soma). Then an un-branched uniform cable is constructed that preserves two constraints; 1. The input resistance at one end of this cylindrical model is R_{soma} , whereas the transfer resistance from the other cylindrical end to the soma is the minimal $R_{i,\text{soma}}$, as computed in the detailed modelled neuron (Fig. 2 below). This provides a unique cylindrical model (entitled “Eq_Cyl”) for the detailed dendritic model. Note that for each input frequency (ω) there is unique cylinder that preserves both $Z_{\text{soma}}(\omega)$ and $Z_{i,\text{soma}}(\omega)$, the input impedance at the soma and the minimal transfer impedance to the soma, respectively. Finally, the location of synapse i of the detailed model is mapped into the reduced cylinder such that $R_{i,\text{soma}}$ is preserved in the two models (the detailed and the cylindrical). The idea is that by preserving the somatic input resistance and the transfer resistance from synaptic location to the soma in both models, the reduced model will closely approximate the synaptic I/O properties of the detailed model. Initial success of this approach is depicted in Figure 2. This work is in preparation for publication.

(i) Simplified branched tree that preserves the number of independent nonlinear dendritic subunits (“Neuron_Reduce”).

We have also started to develop an additional approach to tackle the case of reducing model complexity of neurons consisting of highly non-linear dendritic trees (e.g., generating local NMDA spikes at multiple dendritic branches). In this case, the idea for reduction is to first characterize how many independent nonlinear dendritic subunits exist in the full (detailed) tree (say 10 such independent subunits, see Fig. 3); then reduce the model to a branched tree containing these same number of nonlinear dendritic branches and general morphology of these nonlinear subunits (see Fig. 3). This reduction scheme is presently under its initial steps of exploration.

2.1.1.3 Model Data

The model reduction ideas presented above were conceived by Idan Segev and implemented computationally by three students involved under T4.1.1 from Segev's team.

The experimental data we have used for our modelling task arrives both from the large set of reconstructed and physiologically characterized neocortical neurons from P14 rats, available in the Blue Brain Project portal (<https://bbp.epfl.ch/nmc-portal/welcome>) and from recently established rare data set on L2/3 pyramidal neurons from human temporal cortex (see Figure 1); the whole human data is available at (<https://collab.humanbrainproject.eu/-/collab/528/nav/4671>).

Algorithms and simulations are implemented under Neuron/Python framework.

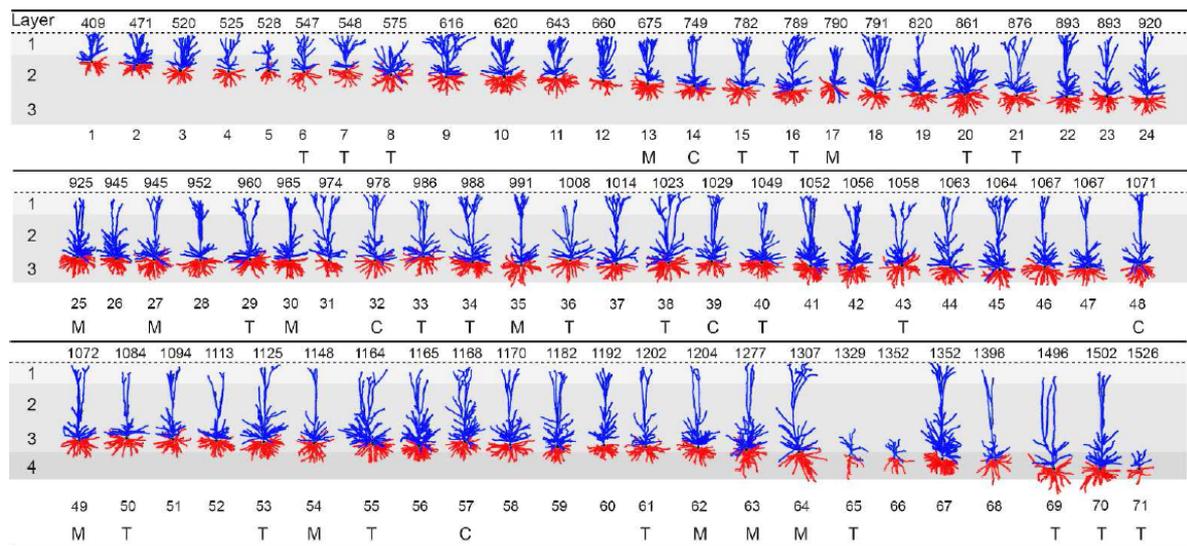


Figure 1: Sixty-one 3D reconstructed L2/3 pyramidal cells from human temporal cortex.

These cells serve as the 3D database; about 15 of them also include somatic recordings. These latter cells are used for both building detailed compartmental models (T6.1.3) and for using reduction schemes (T41.1) as explained in the text. Adapted from Mohan et al., (2015).

- Task 4.1.1 is handled by both Idan Segev and Alain Destexhe; the reduction algorithms developed hereby and the reduced models are aimed to serve to build large networks of neurons. They are communicated to SP6, and Destexhe's lab examines the behavior of neurons under in vivo conditions (“high conductance state”), with an aim to obtain efficient units that include dendritic nonlinearities. Destexhe is also responsible to conceive models in a format compatible with neuromorphic hardware.
- Data is based on BBP portal for rat cells (P14) and on recently established database on adult L2/3 Human pyramidal cells from the temporal cortex (following brain operation). This database, including the biophysical properties of the cells from in vitro experiments, stems from the collaboration with Huib Mansvelder and Christian deKock (Amsterdam, T2.1.4 VU VUMC, NL). Detailed 3D data on dendritic spines dimensions and densities from L2/3 human cells comes from post-mortem material, from Javier DeFelipe's team (Madrid, T2.1.3 UPM, ES). Details could be found in Mohan et al., Cereb Cortex. 2015 Dec; 25(12):4839-53 (2015).
- Neuron and Python are the essential simulation tools; they are used for reconciling anatomical and experimental data with detailed neuron models (the benchmark); then the same tools are used for the reduction schemes (and for checking the quality of the reduction) - as explained in the text attached hereby.

- As can be seen in Figs. 2&3 we have made significant progress in the last 6 months in examining the quality of our first reduction algorithm (Fig. 2) and also in developing a method for assessing the number of nonlinear dendritic subunits (e.g., in L2/3 human pyramidal cells, Fig. 3).

- References

Mohan H, Verhoog MB, Doreswamy KK, Eyal G, Aardse R, Lodder BN, Goriounova NA, Asamoah B, Brakspear AB, Groot C, van der Sluis S, Testa-Silva G, Obermayer J, Boudewijns ZS, Narayanan RT, Baayen JC, Segev I, Mansvelder HD, de Kock CP (2015) Dendritic and Axonal Architecture of Individual Pyramidal Neurons across Layers of Adult Human Neocortex. *Cereb Cortex*, Dec;25(12):4839-53. doi: 10.1093/cercor/bhv188. Epub 2015 Aug 28

Guy E, Matthijs B., Testa-Silva G, Deitcher Y, Lodder C., Benavides-Piccione R, Morales J, DeFelipe J, de Kock (Submitted). Unique Membrane Properties and Enhanced Computational Capabilities of Human Neurons.

2.1.1.4 Model Results

Results for the quality of reduction using the “equivalent transfer resistance” approach (the “Eq_Cyl” scheme) is shown in Figure 2. At left top (A) the classical reduction scheme proposed by W. Rall, whereby the original tree (red) is reduced to an “equivalent cable” (with variable diameter). In this reduction scheme the total membrane area of the reduced cable is identical to that of the full tree. In B, our new “equivalent cylinder” scheme is depicted. In this scheme the transfer resistance (or impedance, for a given frequency ω) from any synapse (i) to the soma is preserved. At right, the quality of these reduction schemes could be assessed when compared to the performance of the full tree (top). In this case the original detailed tree received 4,000 excitatory and 500 inhibitory synapses that were randomly activated and uniformly distributed over the dendritic tree (resulting in the firing of the black spikes at top). The newly reduced scheme (red traces) replicated closely the original firing pattern (and number) of spikes in the detailed model whereas the classical Rall model failed to do so.

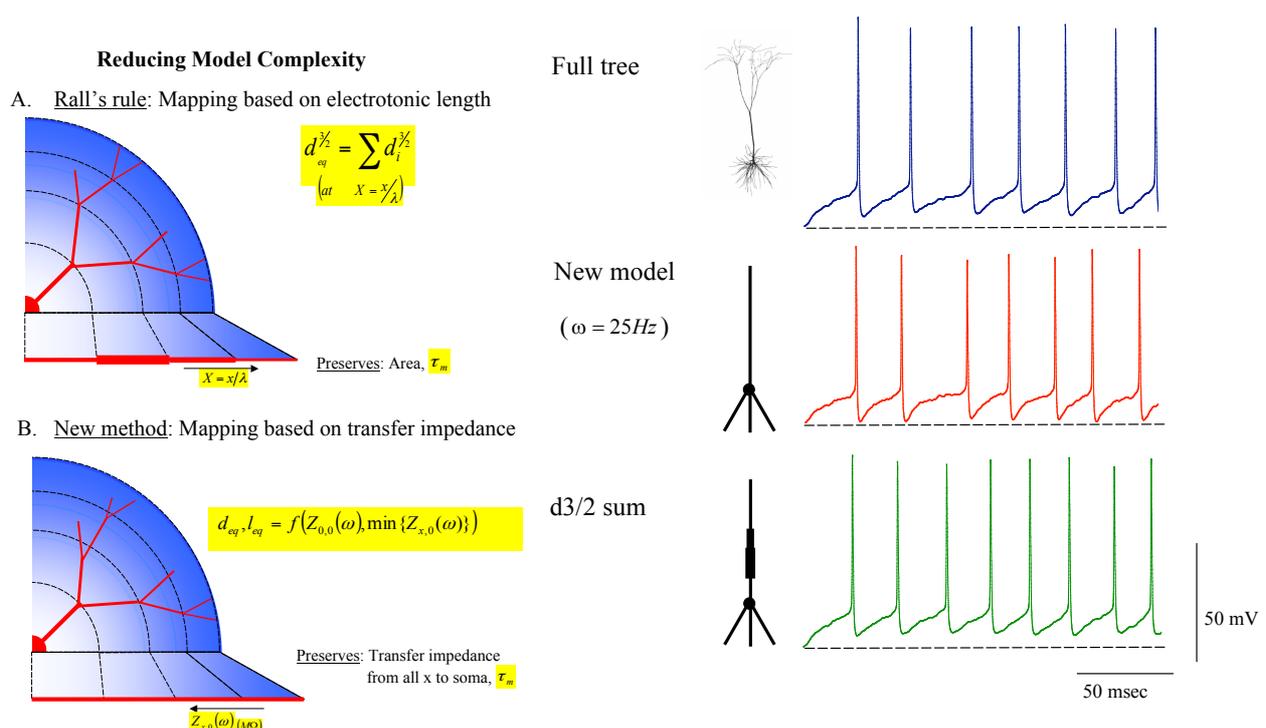


Figure 2: Reduction scheme for complex neuron models (“Eq_Cyl”).

A. The classical “Rall’s equivalent cable” (not a cylinder) approach that preserves the total dendritic area of the full tree. B. The new scheme proposed here (the Eq_Cyl”), preserving in a single equivalent cylindrical cable, the transfer resistance from all original synapses to the soma. This “equivalent cylinder” could be constructed for different frequencies (ω) whose transfer impedance (rather than transfer resistance) is preserved between the complete tree and the reduced cylinder. Right. The success of our reduced model (red traces) in preserving the firing pattern of the full model (blue traces) is shown; it can be appreciated that this reduction scheme (namely the “Eq_Cyl”) approach performs much better than Rall’s “equivalent cable” approach (green traces).

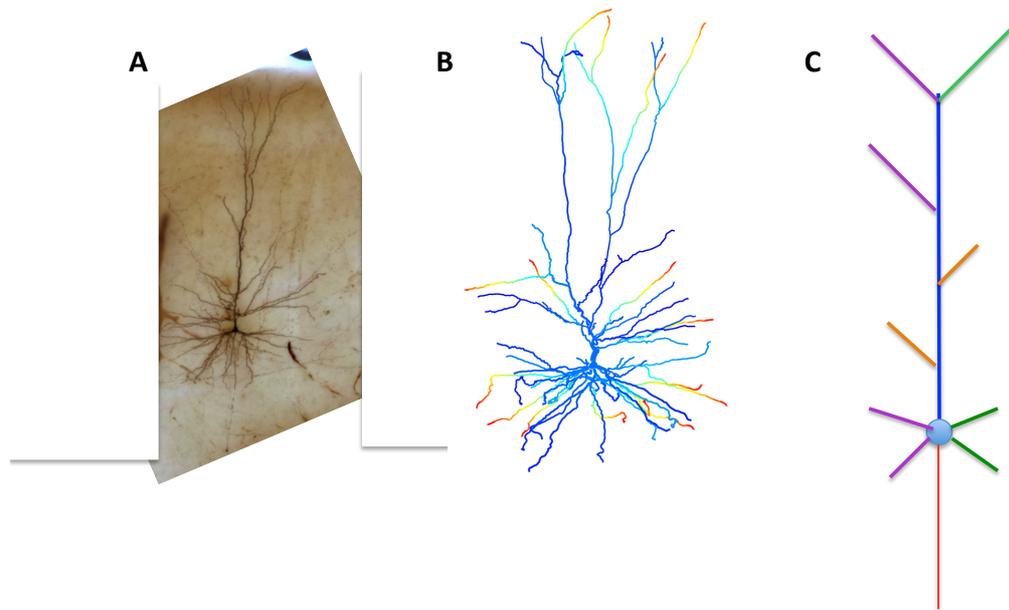


Figure 3: Steps in reducing 3D reconstructed and physiologically characterized nonlinear dendritic tree into a simplified model (“Neuron_reduce”).

A. A stained L2/3 pyramidal cells from a slice of human temporal cortex following brain operation. B. Same neuron is modelled in details, and the putative number of independent NMDA spikes (based on EPSP’s obtained from pair recordings and NMDA properties obtained from intracellular patch recordings from these cells (not shown). Red terminals denote BMDA spikes that could be generated simultaneously and are independent of each other. C. A reduced model consisting of the overall morphology/distribution of the NMDA generating dendritic subunits, neglecting the many hundreds of compartments used for reconstructing the full tree shown in A&B. The quality of this reduction compared to the performance of the full model is under examination.

2.1.1.5 Provenance

NA

2.1.2 A. Destexhe (CNRS) _ Model of dendritic integration with excitable dendrites

2.1.2.1 Introduction

In order to investigate the role of signal integration in dendritic trees for large networks of neurons it is crucial to construct simplified models of dendrites.

Detailed multi-compartmental Hodgkin-Huxley type models of neuron are computationally challenging and therefore inconvenient for constructing larger networks. On the other hand leaky integrate-and-fire model are faster, but they cannot reflect the diversity of neuronal spiking patterns.



In order overcome this problem we use the Adaptive Exponential (AdEx) model. The major advantage of AdEx model is its compatibility with the existing neuromorphic hardware, which is being developed in the SP9 platform.

Constructing a simplified model of excitable dendrites using the AdEx model is part of the 4.1.1 task.

2.1.2.2 Model Description

The AdEx model has arisen as a synthesis of two improvements:

- 1) the replacement of a fixed firing threshold of leaky integrate-and-fire by spike-triggering depolarizing exponential current
- 2) introducing neuronal adaptation by additional current governed by its own differential equation

The voltage in a single compartment changes according to:

$$c_m \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + g_e(t)(V - E_e) + g_i(t)(V - E_i)$$

where c_m is the specific membrane capacitance, g_L - the leak conductance, V_T - the spike threshold, Δ_T - a slope factor. w is an adaptation current governed by the equation

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w$$

When the potential is near threshold V_T depolarizing exponential current is able to surpass all other currents and the potential quickly tends to infinity. Whenever potential crosses detection value V_{det} we reset potential and adaptation current

$$V \rightarrow V_r \quad w \rightarrow w + b$$

Parameter a represents the sub threshold adaptation and parameter b - spike-triggered adaptation.

For our multi-compartmental model, the voltage changes according to

$$c_m \frac{\partial V}{\partial t} = \frac{d}{4r_i} \frac{\partial^2 V}{\partial t^2} - g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + g_e(t)(V - E_e) + g_i(t)(V - E_i)$$

where d is the diameter of the dendritic segment and r_i is the specific intracellular resistance.

For our simulations we have constructed a ball and stick neuron with n_c dendritic compartments. Each compartment is of AdEx type.

The synapses were located on the dendritic compartments and were of two types: excitatory AMPA synapses ($E_e = 0\text{mV}$) and inhibitory GABA A synapses ($E_i = -75\text{mV}$). Synaptic inputs were given by time-dependent conductance $g_e(t)$ and $g_i(t)$. On each compartment we have placed $n_{s,exc}$ excitatory synapses and $n_{s,inh}$ inhibitory synapses.

To simulate synaptic noise, for each synapse we have generated Ornstein-Uhlenbeck processes:

$$\tau d\nu = (r - \nu) + \sigma dW$$



where τ and σ are positive and dW is a Wiener process. These processes were then used as a fluctuating firing rates for inhomogeneous Poisson processes. Then each spike generated in Poisson process was translated into change of conductance. The spike trains could be then correlated according to the level of Ornstein-Uhlenbeck processes

$$\nu_c = C\nu$$

where C is a correlating matrix of dimension $n_c \times n_s$

$$C = \begin{bmatrix} D & S & \dots \\ S & D & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where

$$D = \begin{bmatrix} d & w_l & \dots \\ w_l & d & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad S = \begin{bmatrix} w_g & w_g & \dots \\ w_g & w_g & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

are square matrices of dimension $n_s \times n_s$ and

$$w_l = \frac{\sqrt{c_l}}{n_s n_c} \quad w_g = \frac{\sqrt{c_g}}{n_s n_s} \quad d = \frac{1}{n_s n_c} (1 - (n_s - 1)\sqrt{c_l} - n_s(n_c - 1)\sqrt{c_g})$$

where $c_l, c_g \in [0,1]$.

2.1.2.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.

T4.1.1 UNIC/EITN, Alain Dextexhe

- Data, algorithms, tools and methodologies storage location(s) (and links?)

The model is available at ...

- Description of algorithms/models/principles:

- Format, language if applicable.

Python

- Name of DICs/software catalogue/or HBP github project entries.

...

- Description of data:

- Species, sex, age, number of specimen/subjects.

N/A.

- Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

N/A.

- Completeness of data/algorithms/models:

The model is progressing as planned and its development is still on going.

- Data Quality and Value:



- Verification of data quality.

We have tasted the code comparing output data with analytical estimations (e.g. velocity of the dendritic spike)

- Your subjective analysis of the value of the data/algorithms for the users.

The parameters used in the program are well described in the comments. The output data is stored in easy to access npy. files and is visualized in a clear way by a number of plots. The program can be used by a user with a basic knowledge of Python language.

- Data/algorithm/model usage to date:

- Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

This code is still in development and it is not yet made available for the HBP community. We plan to release the code to the public at the end of ramp-up phase.

- Are the data/algorithms/models considered final?

No. The code is still in development. We expect much development in the functionality of the actual algorithms and at the level of user interface.

- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

The results haven't been published yet.

2.1.2.4 Model Results

Basic properties of dendritic spikes in the model

To measure the speed of the dendritic spike, we have generated a single spike at the end of the dendrite and measured the time of its propagation toward soma.

The measured speed can be explained by following approximation. We want to know how quick the potential changes in the compartment in front of the spike. We take into account only exponential and leak currents and the current which flows from the compartment with a spike:

$$\frac{dV}{dt} = \frac{g_L \Delta_T}{c_m} \exp\left(\frac{V - V_T}{\Delta_T}\right) + \frac{d}{4r_i l^2 c_m} (V_p - V) - \frac{g_L}{c_m} (V - V_R)$$

where V_p is the spike detection value. From this equation we can calculate the time needed for the potential to rise from the resting potential to the detection value. The velocity of the spike is the length of compartment divided by this time.

We checked how spike velocity depends on diameter and length of compartments and on specific axial resistance r_i , spike detection potential V_{det} and slope factor Δ_T . We compared the measured velocities with our approximation.

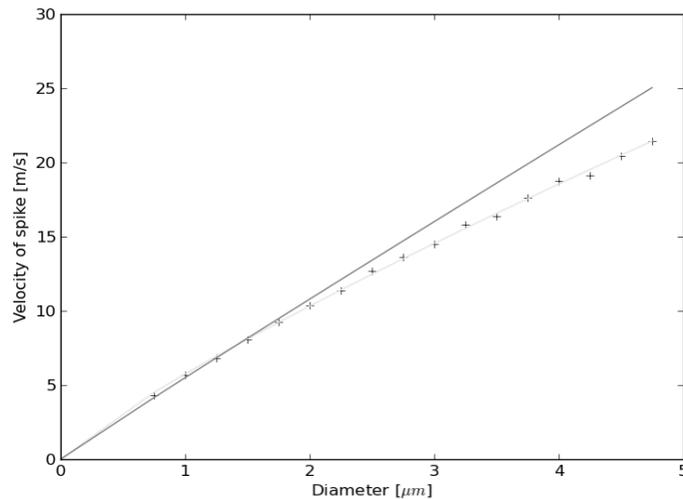


Figure 4: Velocity of spike increases with the diameter of dendrite.

Velocity measured in simulations (crosses) and fit function (grey). Approximated speed (black line).

When dendritic action potentials collide, they annihilate. To show this behavior with the AdEx model, we generated two spikes: one near soma and one at the end of the dendrite. Upon collision, the annihilation of spikes was complete.

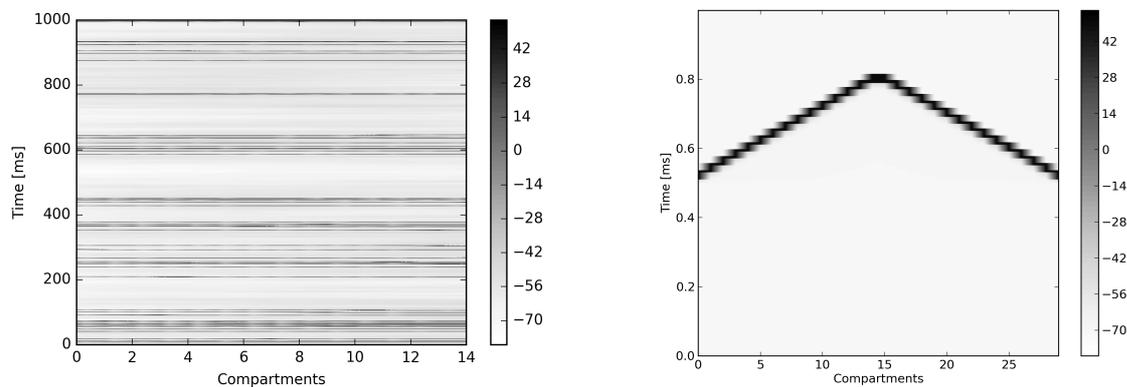


Figure 5: Left: Collision of dendritic spikes. Right: propagation of dendritic spikes

Influence of correlation of synaptic bombardment

Next, we checked how the number of spikes which reach soma depends on the global correlation coefficient c_g for different parameters of the Ornstein-Uhlenbeck noise.

It occurred that this relation depends mainly on time constant of the Ornstein-Uhlenbeck noise, see Figure 6.

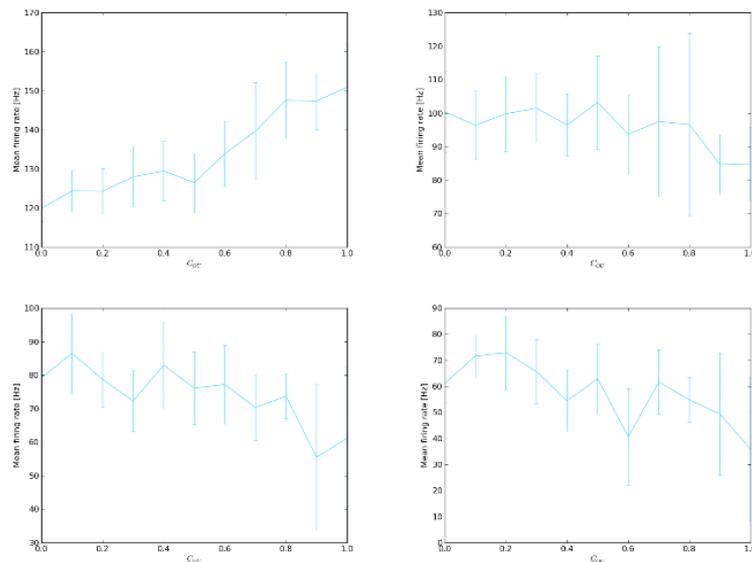


Figure 6: Number of spikes which reached soma as a function of correlation coefficient c_g for different τ , starting upper left plot 0.01, 0.1, 1, 10. $\sigma = 0.1$, and $r = 50\text{Hz}$

Other parameters such as velocity of spike also affects the relationship between a number of spikes in soma and a global correlation of synaptic bombardment. This is currently under investigation, as well as the role of synaptic weights.

2.1.2.5 Provenance

Not applicable.

2.1.3 W. Gerstner (EPFL) _ Derive simplified neuron and neural circuit models from biophysically morphologically detailed models

2.1.3.1 Introduction

Work done at EPFL-LCN: Neuron models from data: algorithmic pipeline to extract parameters from measurements. The methodology has been verified, written up in a paper in PLOS Computational Biology, and will be used in the future routinely in the Allan Institute for large-scale data collection.

At EPFL-LCN we have also same methodology to reduce complex neuron models from the simulation platform to simplified integrate-and-fire type models.

In the following quotation marks refer to quotes from the paper Pozzorini et al. in PLOS Comput. Biology. 2015

In task 4.1.1, we developed a pipeline for automated, high-throughput characterization of single neurons. The proposed approach consists of fitting and validating simplified neuron models (i.e., Generalized Integrate-and-Fire models) to electrophysiological recordings.

The pipeline can be used to: i) derive simplified neuron models from in vitro patch-clamp recordings and ii) derive simplified neuron models from biophysically morphologically detailed models. The methodology has been verified and written up in a paper published in PLOS Computational Biology as:

Pozzorini C, Mensi S, Hagens O, Naud R, Koch C, Gerstner W (2015) Automated High-Throughput Characterization of Single Neurons by Means of Simplified Spiking Models. PLoS Comput Biol 11(6): e1004275. doi:10.1371/journal.pcbi.1004275

The computational tools described in the manuscript are currently being used at the Allan Institute for Brain Science for the analysis of the Allen Cell Types Database



(<http://celltypes.brain-map.org/>). Within HBP, the pipeline is currently being used to simplify the detailed model of the cortical column recently published by Markram et al. Cell 2015. In the following, quotation marks refer to quotes from the paper Pozzorini et al. 2015.

“In vitro patch-clamping is the gold standard used to investigate the intrinsic electrophysiological properties of neurons, but remains labor intensive and requires a trained experimentalist with high technical skills. In the last years, several platforms have been developed that automatize electrophysiological recordings for ion-channel screening and drug discovery. In the future, this technology is likely to be transferred to more complex setups, such as in vitro brain slices. To make sense of the large amount of data that automated patch-clamp can produce, adequate computational tools and experimental protocols have to be developed.”

“Traditional protocols for single-neuron characterization rely on somatic current-clamp injections of stimuli (e.g., square current pulses, ramps of current) that are specifically designed to extract a small number of features (e.g. membrane time constant, firing threshold). We developed an alternative approach in which the electrophysiological properties of neurons are characterized by fitting point-neuron models (i.e., generalized integrate-and-fire models) to the response evoked by somatic current-clamp injection of in vivo-like fluctuating currents.”

“Nowadays, integrate-and-fire models are mainly used in large-scale simulations to study the emergent properties of neural circuits. In Pozzorini et al. 2015, we demonstrate that the same models (and their fitting procedures) can serve an equally important purpose, namely to characterize the electrical properties of single neurons. In this view, integrate-and-fire models (and their fitting procedures) are seen as computational tools to automatically compress the information contained in intracellular recordings into a set of unique and meaningful parameters, i.e., the model parameters. Summarizing the information of complex recordings can in turn enable systematic comparisons, clustering and identification of cell types.”

The proposed model is a Generalized Integrate-and-Fire model obtained by extending the standard leaky integrate-and-fire with: i) a spike-triggered current, whose functional shape is not assumed a priori, but is extracted from data; ii) a spike-triggered movement of the firing threshold, whose functional shape is not assumed a priori, but is extracted from data; iii) the exponential escape-rate mechanism for stochastic spike generation. More details about the model are provided in the next section. Model parameters can be efficiently extracted from a limited amount of data using the three-step procedure described in the next section.

“On the experimental side, the proposed approach relies on in vitro somatic injections of rapidly fluctuating currents that mimic natural inputs received in vivo at the soma of neurons.” The experimental protocol used to fit and validate at GIF model to data is shown and described in the following figure.

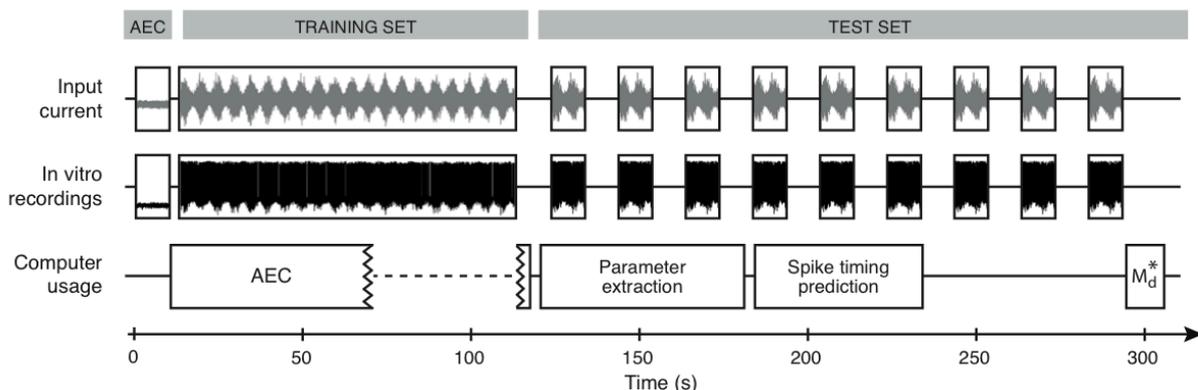




Figure 7: Electrode properties for data preprocessing

“To characterize the properties of the electrode required for data preprocessing by Active Electrode Compensation (AEC), the experimental protocol starts with the injection of a short subthreshold current. While the filtering properties of the electrode are estimated (AEC box, left part), the training dataset is collected that will be used to estimate the GIF model parameters. After training set collection, the raw data are preprocessed with AEC to compensate for the bias known to occur when the same electrode is used to both record and stimulate a neuron (AEC box, right part). Then, in parallel with GIF model parameter extraction and successive spike timing prediction, the test dataset is collected by injecting 9 repetitions of the same time-dependent current. Finally, after complete acquisition of the test set, the model performance Md^* is computed. Md^* quantify the similarity between the observed and the predicted spike trains (i.e., the fraction of spikes that the GIF model can correctly predict with a temporal precision of 4ms). Overall, GIF model parameter extraction and validation requires around 5 minutes of recording and CPU time. This method is thus suitable for high-throughput single-neuron characterization.”

Importantly, this experimental protocol, together with the GIF model and its fitting and validation procedure can also be used to derive simplified neuron models from more complex, biophysical multi-compartmental models (i.e., the main goal of Task 4.1.1.). In this case, the experimental data are not acquired from biological neurons, but are obtained *in silico* by simulating the response of a biophysical model of interest. Since *in silico* recordings are not biased, Active Electrode Compensation is not required and the experimental protocol directly starts with the training set acquisition.

As shown in the results section, the GIF model achieves good performance in describing the electrophysiological activity of cortical pyramidal neurons and complex biophysical models. The methods and results reported in Pozzorini et al. 2015 are currently being used in two different cortexes:

- 1) Automated modeling and analysis of the electrophysiological recordings performed at the Allen Institute for Brain Science (see Allen Cell Types Database: <http://celltypes.brain-map.org/>)
- 2) Systematic reduction/simplification of the biophysical models used for the detailed model of a cortical column developed within HBP and recently published by Markram et al. Cell 2015.

2.1.3.2 Model Description

“The GIF model discussed in Pozzorini et al. 2015 is a leaky integrate-and-fire model augmented with a spike-triggered current $\eta(t)$, a moving threshold $\gamma(t)$ and the escape rate mechanism for stochastic spike emission. In the model, the subthreshold membrane potential $V(t)$ evolves according to the following differential equation:

$$C \frac{dV(t)}{dt} = -g_L(V(t) - E_L) + I(t) - \sum_{t \in \{t\}} \eta(t - \hat{t})$$

where the parameters C , g_L and E_L define the passive properties of the neuron, $I(t)$ is the input current and $\{t\}$ are the spike times. Each time an action potential is fired, an intrinsic current with stereotypical shape $\eta(t)$ is triggered. Currents triggered by different spikes accumulate and produce spike-frequency adaptation, if $\eta(t) > 0$ (or facilitation, if $\eta(t) < 0$). The functional shape of $\eta(t)$ varies among neuron types as previously shown in Mensi et al. 2012. Consequently the time course of $\eta(t)$ is not assumed a priori but is extracted from intracellular recordings. Each time a spike is emitted, the numerical integration is stopped during a short absolute refractory period and the membrane potential is reset. Spikes are produced stochastically according to a point process with conditional firing intensity $\lambda(t|V, VT)$, which exponentially depends on the momentary difference between the membrane potential $V(t)$ and the firing threshold $VT(t)$:



$$\lambda(t|V, V_T) = \lambda_0 \exp\left(\frac{V - V_T}{\Delta V}\right)$$

where λ_0 has units of s⁻¹, so that $\lambda(t)$ is in Hz and ΔV defines the level of stochasticity. In the limit $\Delta V \rightarrow 0$, the model becomes deterministic and action potentials are emitted at the precise moment when the membrane potential crosses the firing threshold. Importantly, the value of ΔV is extracted from experimental data. The dynamics of the firing threshold $V_T(t)$ is given by:

$$V_T(t) = V_T^* + \sum_{t \in \{\hat{t}\}} \gamma(t - \hat{t})$$

where V_T^* is a constant and $\gamma(t)$ describes the stereotypical time course of the firing threshold after the emission of an action potential. Since the contribution of different spikes accumulates, the moving threshold constitutes an additional source of adaptation (or facilitation). Similar to $\eta(t)$, the functional shape of $\gamma(t)$ is not assumed a priori but is extracted from intracellular recordings.”

Fitting procedure

Given the intracellular voltage response $V_{data}(t)$ evoked in vitro by a fluctuating input current $I(t)$, all of the GIF model parameters can be extracted from experimental data using a three-step procedure.

Step 1: the parameters related to absolute refractoriness and voltage reset are extracted. First, the experimental spike train is defined as the collection of instants at which $V_{data}(t)$ crossed a certain threshold from below. The average spike shape $VSTA(t)$ is then obtained by computing the spike-triggered average (STA) of $V_{data}(t)$. The absolute refractory period T_{ref} is fixed to twice the spike width at half maximum and the reset potential is computed as $V_{reset} = VSTA(T_{ref})$.

Step 2: the parameters related to the subthreshold voltage dynamics are extracted. The first-order temporal derivative of the experimental voltage is estimated from the data and the parameters determining the membrane potential dynamics are extracted by fitting the model voltage derivative on data. This is done by exploiting the knowledge of the experimental voltage $V_{data}(t)$ and the external input $I(t)$. To avoid a priori assumptions on the functional shape of the spike-triggered current, $\eta(t)$ is expanded in a linear combination of rectangular basis functions. Consequently, optimal subthreshold parameters $\theta_{sub} = \{C, g_L, E_L, \eta(t)\}$ minimizing the sum of squared errors on the voltage derivative can be efficiently obtained by solving a multilinear regression problem. Since simplified threshold models do not describe the membrane potential dynamics during action potentials, all the data close to spikes are discarded.

Step 3: the parameters related to the firing threshold dynamics are extracted. The parameters estimated so far are first used to compute the subthreshold membrane potential of the model. The parameters $\theta_{th} = \{V_T^*, \Delta V, \gamma(t)\}$ defining the firing threshold dynamics are then extracted by maximizing the log-likelihood of the experimental spike train being produced by the GIF model. Similar to $\eta(t)$, the spike-triggered threshold movement is extracted nonparametrically by expanding $\gamma(t)$ in a linear combination of rectangular basis functions. In the GIF model, the nonlinear function of the escape-rate mechanism is convex and log-concave. Moreover, its argument is linear in the model parameters θ_{th} . Given these properties, the log-likelihood to maximize is guaranteed to be a concave function of the model parameters and the optimization problem can be solved using standard gradient ascent techniques.

2.1.3.3 Model Data

- **Task(s)/group(s) responsible for generating algorithms/models/principles.**



Simulation and model development by EPFL-LCN, Group of Wulfram Gerstner, responsible scientist has been Christian Pozzorini.

- **Data, algorithms, tools and methodologies storage location(s) (and links?)**

Computational tools for fitting GIF models to data, publicly available on:

<https://github.com/pozzorin/GIFFittingToolbox>

Published paper publicly available on:

<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004275>

- **Description of algorithms/models/principles:**

- **Format, language if applicable.**

Code written in Python. Tutorial publicly available on:

<https://github.com/pozzorin/GIFFittingToolbox/wiki/Automated-high-throughput-single-neuron-characterization>

- **Name of DICs/software catalogue/or HBP github project entries.**

DIC entry generated under task 4.1.1.

- **Description of data:**

- **Species, sex, age, number of specimen/subjects.**

In vitro patch clamp recordings used to validate the methods were performed in mice (C57Bl/6J), male, P13-15, 10 cells from different animals.

In silico patch clamp recordings used to validate the methods were performed using the multicompartmental model of a L5 pyramidal neuron introduced in Hay et al. 2011, PLOS Comp. Biol.:

<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002107>

- **Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.**

Neurons, physiology. The model is flexible and can be applied to data from different brain regions. The results in Pozzorini et al. 2015 were from layer 5 somatosensory cortex and provide a characterization of the electrical activity of single neurons.

- **Completeness of data/algorithms/models:**

- **Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?**

Model and algorithms meet the anticipated target. In particular, the GIF model accurately predicts the activity of the Hay model responding to somatic current-clamp injections of *in vivo* like fluctuating currents.

- **Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community**

- **The pipeline for single neuron characterization and modeling has been tested on a restricted number of neurons and biophysical models.**

Within HBP, the pipeline is currently being used to map 31000 biophysical, multi-compartmental neuron models used in Markram et al. 2015 into point-neuron models. The



pipeline is currently playing an important role in the effort of providing a simplified version of the cortical column model.

- Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.

Fitting neuron models to in vitro recordings has a long history. Several models and fitting procedures have been introduced both by previous scientists at EPFL-LCN as well as by other research groups. Most of the existing methods are based on deterministic integrate-and-fire models. While these models have been shown to achieve good performance in predicting spike times, their fitting procedures are not efficient and often require high computing power. Existing methods were thus not well suited for high-throughput modeling and characterization of single neurons.

The major contribution of Pozzorini et al. 2015 was to develop a stochastic version of the integrate-and-fire model, which allows for efficient and reliable parameter extraction. In contrast to previous stochastic models (such as the Generalized Linear Model), the GIF model has a clear biophysical interpretation. With the proposed method, a neuron model can be derived reliably and efficiently either from patch-clamp recordings or more complex biophysical models. In contrast to existing methods, our fitting procedure is convex. The set of model parameters found by the algorithm is thus guaranteed to be optimal.

Overall, less than 5 minutes of data acquisition and computing are sufficient to generate and validate a single neuron model from limited amounts of data. In contrast to previous methods, our method is suitable for high-throughput analysis and modeling.

- Outline state of validation work.

The validation was done by fitting the GIF model both to in vitro patch-clamp recordings from layer 5 pyramidal neurons as well as by fitting the GIF model to a state-of-the-art multi-compartmental model (Hay et al., PLOS Comp. Biol. 2011).

- Data Quality and Value:
 - Verification of data quality.

The main check for verification has been that the GIF model accurately predicts the electrical activity of biological neurons and complex biophysical models. In both cases, the GIF model was able to predict around 80% of the spikes with a temporal precision of 4ms. To avoid problems related to over fitting, model validation was performed on a new dataset (i.e., cross-validation) that was not used for parameter extraction.

- Your subjective analysis of the value of the data/algorithms for the users.

The algorithm for parameter extraction is robust and efficient. The resulting GIF models are of good quality and can mimic the electrical behavior of cortical neurons as well as of more complex neuron models.

- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

The methods are currently being used at Allen Institute for Brain Science to characterize the electrical activity of a large number of cells (see Allen Cell Types Database: <http://celltypes.brain-map.org/>).



Inside HBP, the model is currently being used to generate a simplified version of the detailed model of a neocortical column recently published by Markram et al. as:

Markram, Henry, et al. "Reconstruction and simulation of neocortical microcircuitry." *Cell* 163.2 (2015): 456-492.

- Are the data/algorithms/models considered final?

Yes.

- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

Computational methods, code and data are published in Pozzorini et al. 2015:

Pozzorini C, Mensi S, Hagens O, Naud R, Koch C, Gerstner W (2015) Automated High-Throughput Characterization of Single Neurons by Means of Simplified Spiking Models. *PLoS Comput Biol* 11(6): e1004275. doi:10.1371/journal.pcbi.1004275

2.1.3.4 Model Results

To verify its validity, the pipeline for single neuron characterization was tested on: i) in vitro recordings from layer 5 pyramidal neurons (obtained using standard patch clamping) and ii) in silico recordings obtained by simulating the activity of a multi-compartmental conductance-based model by Hay et al. *PLOS Comp. Biol.* 2011.

Results from in vitro recordings (single neuron characterization)

Despite its relative simplicity, the GIF model is able to predict both the subthreshold fluctuations of the membrane potential and the spikes evoked by in vivo-like fluctuating currents that have not been used for parameter extraction. The GIF model performance was quantified by the percentage of variance explained for the subthreshold membrane potential fluctuations, and by the percentage of spikes correctly predicted with a temporal precision of 4 ms, for the spiking activity. For layer 5 pyramidal neurons of the mouse somatosensory cortex, the percentage of variance explained was 80% and the percentage of spikes predicted 79%, on average. Fitting and validating a single neuron model to data required less than 5 minutes in total. The following figure illustrates the performance achieved by the model in capturing the activity of a layer 5 pyramidal neuron of the mouse somatosensory cortex.

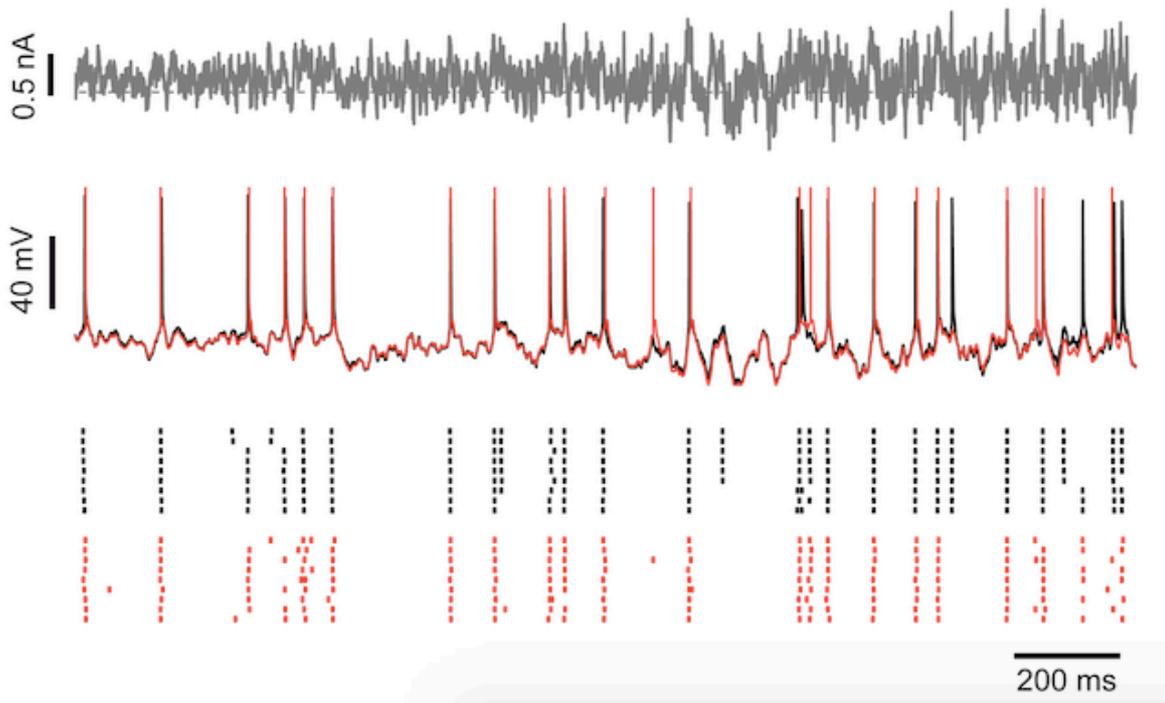


Figure 8: GIF Model

“The GIF model accurately predicts the occurrence of individual spikes with millisecond precision. To evaluate the predictive power of the GIF model, the response of a L5 pyramidal neuron to a fluctuating input current (top) has been recorded intracellularly (middle, black). The same protocol was repeated nine times to assess the reliability of the neural response (bottom, black raster). The GIF model (with parameters extracted using a different dataset) was able to accurately predict both the subthreshold (middle, red) and the spiking response (bottom, red raster) of the cell. The horizontal dashed line represents 0 pA.”

Results from in silico recordings (model simplification)

We also used our pipeline for automated high-throughput single neuron characterization to simplify a biophysical, multicompartmental model of a layer 5 pyramidal neuron developed by Hay et al. and published in PLOS Computational Biology in 2011. In this case, the input current was calibrated to obtain an average firing rate of 10 Hz. To model stochastic spike emission, a source of noise was introduced by corrupting the input current with additive white-noise. All in silico experiments were performed by delivering the current at the somatic compartment. Similar to the results obtained from layer 5 pyramidal neurons, the GIF model was able to predict 80% of the spikes with a temporal precision of 4ms and explained 74% of the variance of the subthreshold potential fluctuations. The following figure illustrates the performance achieved by the GIF model in capturing the activity of the Hay model (Hay et al. PLOS Comp. Biol. 2011).

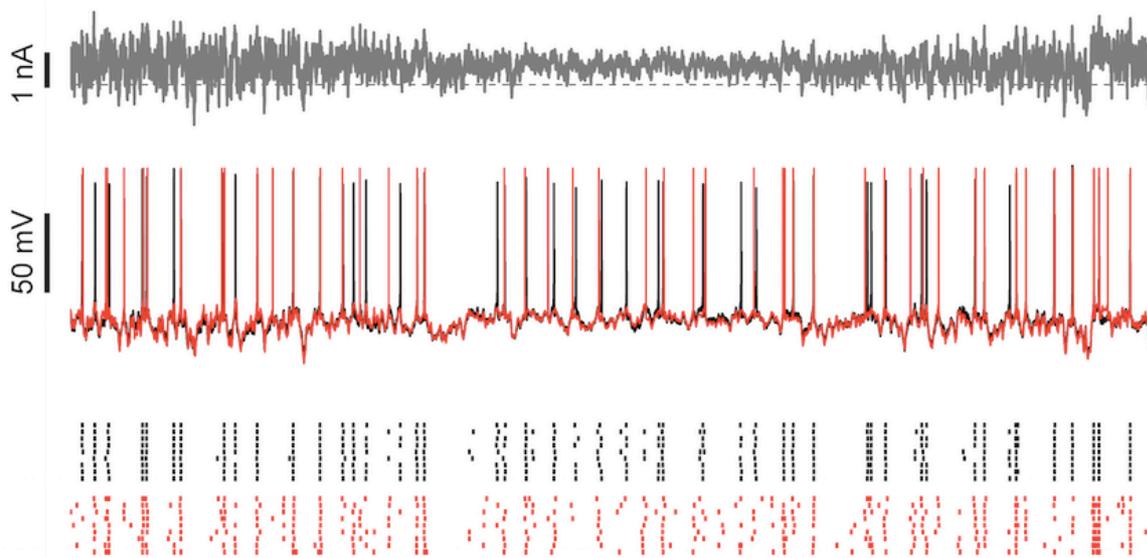


Figure 9: GIF Model (2)

“The GIF model (red) accurately predicts the subthreshold and spiking activity of the multi-compartmental biophysical model by Hay et al. PLOS Comp. Biol. 2011 (black) responding to a somatic current injection (gray). Top: input current. Middle: the membrane potential dynamics predicted by the GIF model (red) closely matches the one generated by with the biophysical model (black). Bottom: raster plot of the spiking activity generated in response to nine injections of the current shown in gray by the GIF model (red) and the multi-compartmental biophysical model (black).

These results indicate that the GIF model, together with its fitting and validation procedure, is a valid candidate to simplify the complex biophysical models developed in the context of HBP and recently used to simulate the activity of a cortical column (Markram et al. Cell 2015).

2.1.3.5 Provenance

The pipeline for single neuron modeling is currently been applied to data from Markram et al. Cell 2015 (i.e., simulation of cortical column based on multicompartmental models).

2.2 Task 4.1.2 Modelling brain signals at different scales, from intracellular, local field potentials, and VSD up to EEG

2.2.1 A. Destexhe (CNRS) _ simplified models of local field potential

2.2.1.1 Introduction

The brain signals can be represented in form of hierarchy going from microscopic (intracellular) to macroscopic (EEG/MEG) scales. Moving from one level of the hierarchy to another usually requires some kind of averaging of the signals from the level below (for example, to estimate the LFP one averages all synaptic currents generated within the population). Since the signals at the lower level are usually correlated the average measures are affected by such correlations. We developed a phenomenological model of correlated neuronal activity and studied its impact on the population (LFP) signal. Such simplified models will help us to interpret the collective activity recorded from animal models and humans (Tasks TODO) and it will be calibrated according to the measurement at single neuron level (Tasks TODO). The prediction of the model will be validated by detailed models including the cellular morphology and detailed ion channels (Tasks TODO).

2.2.1.2 Model Description

We considered the LFP signal generated by a population of neurons firing correlated spikes from homogeneous Poisson distribution. The LFP signal was calculated as a linear sum of unitary contributions from each spike, which was modelled as a LFP kernel. For the



purpose of this study, the same LFP kernel was used for all neurons. The choice of the kernel was such, that each neuron only contributed to the LFP in the nearest electrode. We chose a causal kernel consisting of a step at the time of the spike and an exponential tail:

$$k(t) = H(t)Aexp(-t/\tau)$$

where $H(t)$ is the Heaviside function, $A = 1$ (normalised units) and $\tau = 5$ ms. Then the LFP at a single electrode is:

$$LFP_j(t) = \sum_{i=1}^{K_j} k(t - t_i)$$

For each electrode j the spike times $\{t_i\}_{i=1}^{K_j}$ were concatenated over a population of N simulated Poisson neurons. Spikes of distinct neurons were correlated with the correlation coefficient r and exponentially distributed jitter with mean σ (for sample cross-correlograms, see Figure 2B). The correlated spike trains were generated using mother spike train technique as described by Brette (2009).

We implemented three methods for generating correlated spike trains:

- doubly stochastic process – correlated spike trains were generated from a inhomogeneous Poisson process with the same rate described by Ornstein-Uhlenbeck process. This method allows having an arbitrary correlation structure in space, but is limited in the magnitude of the correlations.
- mother spike train method – all spikes were generated from the same “mother template” by drawing spikes randomly and adding jitter. This method can generate spikes of arbitrary correlation coefficients, but with limited correlation structure in space.
- mixture model – this extends over the mother spike train method by taking N “mother templates” and drawing from it with different probabilities. This is most general methods allowing to generate spike trains with flexible correlation structure and high correlation coefficients.

2.2.1.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.

T4.1.2 UNIC/EITN, Alain Dextexhe

- Data, algorithms, tools and methodologies storage location(s) (and links?)

The model will be deposited in an internal repository and released to the public when the accompanying article is accepted for publication.

- Description of algorithms/models/principles:
 - Format, language if applicable.

Python.

- Name of DICs/software catalogue/or HBP github project entries.

Link to the repository will be provided

- Description of data:
 - Species, sex, age, number of specimen/subjects.

N/A.

- Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of



entities, e.g. morphological characterisation of basket cells of the hippocampus.

N/A.

- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?
 - Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
 - Outline state of validation work.

In the modelling part we implement well-accepted and validated algorithms for generation of the correlated spike trains (Brette 2009). The LFP simulation code will be validated against detailed simulations using compartmental models (implemented by the group of Einevoll) in the next phase of the project. The obtained simulation results were shown to agree qualitatively with the LFP recorded in human and monkey brain (see results below).

- Data Quality and Value:
 - Verification of data quality.

Unit tests are partially used to assure the code quality.

- Your subjective analysis of the value of the data/algorithms for the users.

The library may prove useful to the computational community at large. The ease of use and fast performance of the algorithms may encourage many groups from HBP and outside to adopt it. Our goal is to make it a community project, where other groups may contribute corrections, enhancements of new algorithm. For this to work, we will need to invest more time into documenting and testing the code and community building.

- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.
 - This code is still in development and it is not yet made available for the HBP community. We plan to release the code internally at the end of ramp-up phase. The code used for calculation the correlogram function was integrated with the [Elephant library](#) developed by Sonja Grün and Andrew Davison in SP5.
- Are the data/algorithms/models considered final?

No. The code is still in development. We expect much development in the functionality of the actual algorithms and at the level of user interface.

- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)



2.2.1.4 Model Results

To validate the model and compare it with the data (Telenczuk et al. 2016, in prep) we estimated the single spike contribution by means of so-called spike-triggered LFP average (st-LFP), that is the average of short LFP segments centered around each spike. This procedure can be applied to LFP signals from all electrodes, but keeping always firing of the same cell as the reference. This way we obtain a spatio-temporal map of the LFP components coincident with a spike of a given cell. The st-LFP is often used as the measure of the direct contribution of a neuron to LFP ("LFP kernel").

We estimated the st-LFP of the model and compared it with LFP kernel. The spatio-temporal st-LFP obtained in the model extends over the entire array despite the fact that the individual LFP kernels are spatially limited to a single electrode (Figure 1B). In addition, the st-LFPs start before the onset of the spike as observed in the experimental data, although the contribution of each spike starts only at the spike onset. We conclude that the non-locality and non-causality of the experimental st-LFP can be a direct consequence of the neuronal correlations.

To estimate the relative st-LFP contribution of the correlated neurons and compare it with the direct contribution of the trigger neuron, we calculated the st-LFP in the model with correlated and uncorrelated population (Figure 10). In case of uncorrelated population, the st-LFP (red) can perfectly capture the direct contribution of the trigger neuron (LFP kernel, black). However, even in a small population ($N=200$) of weakly correlated neurons (correlation coefficient, $r=0.1$; spike jitter $\sigma=5$ ms) the st-LFP (blue) was dominated by the correlated neurons diverging from the modelled kernel. Both theoretical analysis and simulations show that the magnitude of this "correlation st-LFP" increases linearly with the size of the population and correlation coefficient, whereas it decreases exponentially with spike jitter (Figure 10D).

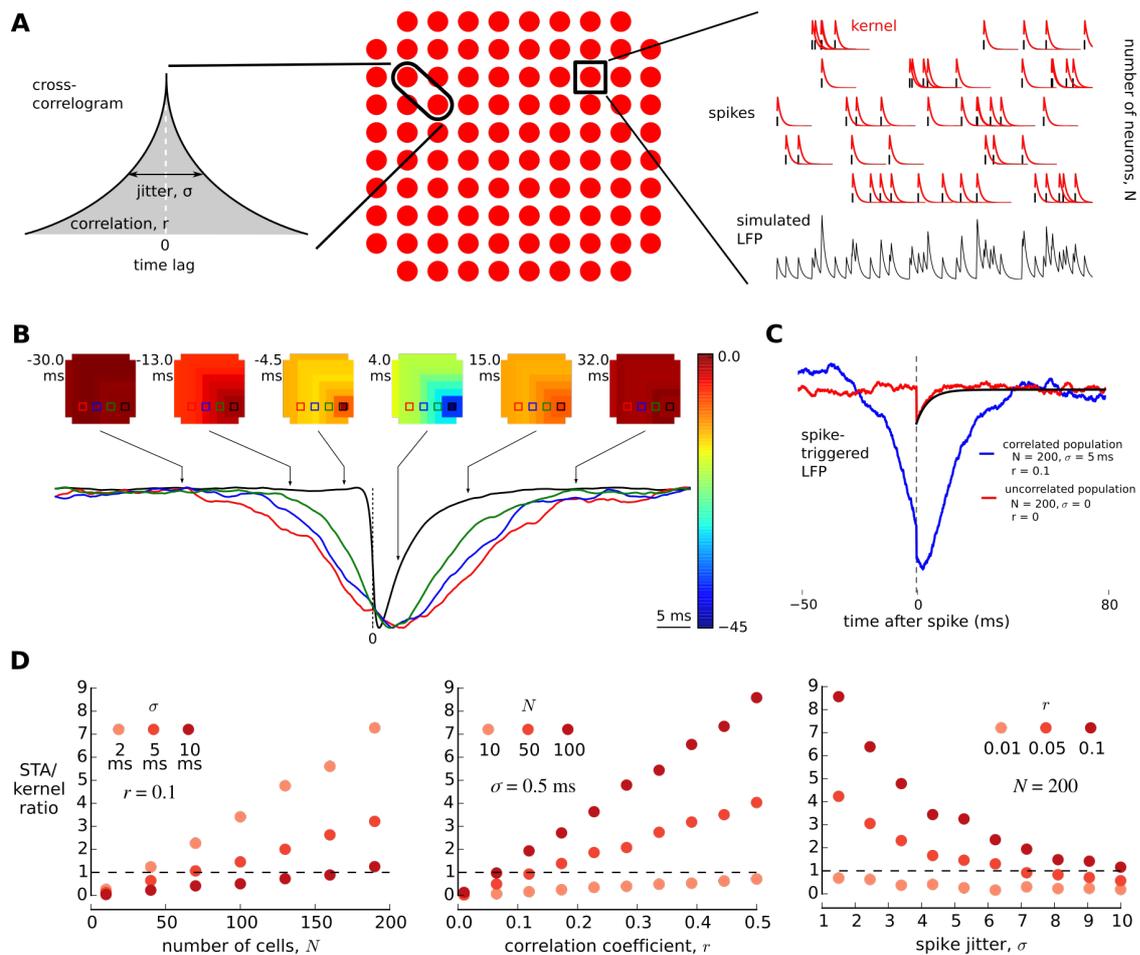


Figure 10: Simulation of single neuron spikes

(A) Single-neuron spikes are simulated using correlated Poisson processes with defined correlograms (left). Electrodes are arranged on a grid modelled after Utah array (middle). Contribution of a single spike to LFP is modelled by a LFP kernel (right). Sum over contributions from all neurons is the simulated LFP (bottom right). (B) st-LFP calculated from simulated LFP signals shows non-local and non-causal components. Traces of different colors in the lower panel show st-LFP four different locations in the array (marked with color squares in the heatmaps). (C) In non-correlated neuronal population st-LFP (red) recovers the direct contribution of a single spike to LFP (LFP kernel, black). When neurons are correlated the st-LFP is broader and substantially higher in amplitude (blue). (D) The amplitude ratio of direct contribution to LFP (LFP kernel) and st-LFP in correlated population depends linearly on the population size (N) and correlation coefficient (r) and non-linearly on the correlation widths (σ). For realistic population size ($N=1000$), correlation coefficients ($r=0.05$) and correlation widths ($\sigma=10$ ms) the contribution of the population is approximately 5 times higher than direct contribution.

Experiments show that the latency st-LFP increases with distance in a form of apparent propagation (Telenczuk et al. 2016, in prep, Figure 11A). Such st-LFP propagation can be reproduced in the model without axonal or synaptic delays. The latencies of the simulated st-LFP are gradually shifted with increasing distance from the trigger neuron (Figure 11B) due to asymmetry of the LFP kernel and increasing widths of cross-correlograms. Since we assumed that the correlation width depends only on distance and not direction, the propagation proceeds uniformly in all directions with equal speeds. In contrast, st-LFP derived from experimental data propagates preferentially in a specific direction.

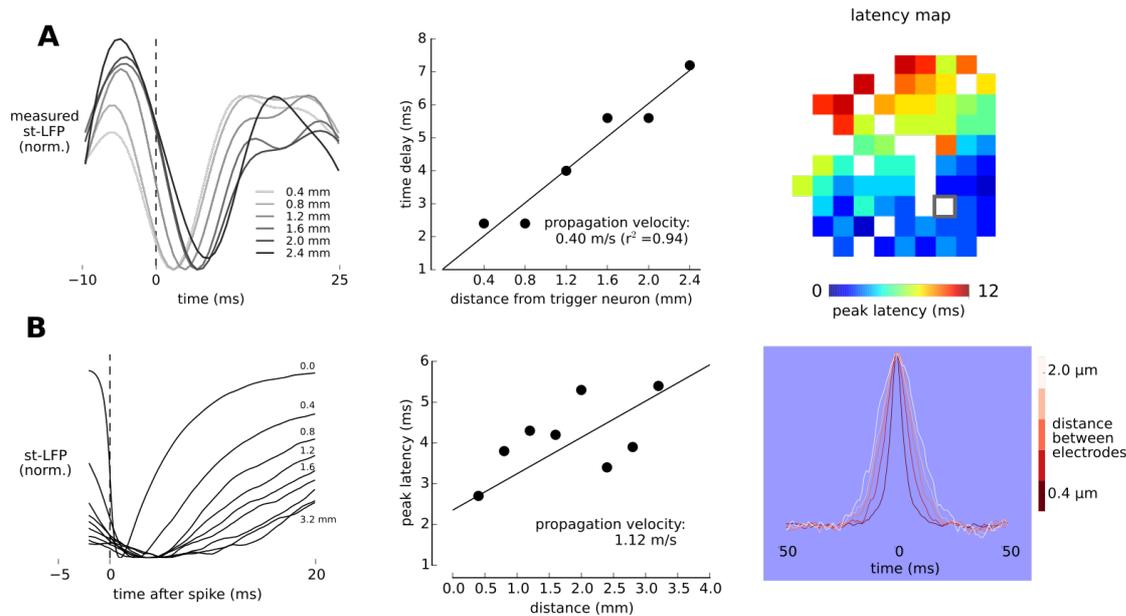


Figure 11: Latency of st-LFP peak

(A) The latency of st-LFP peak (left) increases with distance as a hallmark of st-LFP propagation (middle). The spike-evoked LFP propagates in a preferred direction (right). (B) Even without synaptic delays (cross-correlograms centered around zero, right) the simulated st-LFPs peak at higher latencies for longer distances (left) giving rise to apparent propagation (middle). Similarly to the experimental data, modelled cross-correlograms increase linearly in width with distance (right).

2.2.1.5 Provenance

Not applicable.

2.2.2 G. Einevoll (UMB) _ Modelling brain signals at different scales

2.2.2.1 Introduction

The local field potential (LFP), the low-frequency part of electrical potentials recorded in the brain, offers a unique window into the actions of cortical circuits comprising thousands of neurons. However, due to its numerous neural sources, the LFP is more complicated to interpret than spikes, and careful mathematical modeling is needed to properly analyze the signal. While the link between activity in neurons with passive dendrites and the generated LFP is fairly well understood, the role of active dendritic conductances remains elusive. We have performed a detailed modelling study of the effect of subthreshold active dendritic conductances in shaping the cortical LFP, and a putative key role of the I_h conductance in strongly modifying the signal, is identified.

This modeling project is important to the overall HBP project as the resulting formulas gives the recipe for implementing the computations of LFPs in brain simulation, for example, in CDP1 (contact: Marc-Oliver Gewaltig, EPFL).

2.2.2.2 Model Description for LFP generation from active-dendrite neurons

This model description is based on the description available in the arXiv paper “Active subthreshold dendritic conductances shape the local field potential”, that can be accessed from <http://arxiv.org/pdf/1512.04293v1.pdf>

Numerical simulations were carried out using a model of a cortical layer 5 pyramidal cell that was published by Hay et al. (2011). This model has a detailed morphology and includes ten active ionic conductances fitted to experimental data by multi-objective optimization with an evolutionary algorithm. For simulations of a passive model, the active conductances were removed from the model. Simulations with a so-called “frozen” h-type



conductance were performed by keeping the gating variable of the h-current constant, yielding an additional passive conductance.

The original model showed a voltage gradient from soma to distal apical dendrites. In order to simplify the interpretation of the simulation results, we adjusted the leak reversal potential of each compartment such that we could set the resting potential uniformly to a specified chosen potential (Carnevale and Hines, 2006).

The neuron model received either excitatory synaptic input or white-noise current input. Synaptic inputs were modeled as steps in the synaptic conductance followed by an exponential decay with a time constant of 2 ms and used a reversal potential of 0 mV. The white-noise current input consisted of a sum of sinusoids with identical amplitudes but random phases for each integer frequency from 1 Hz to 500 Hz (see Linden et al. (2010)). The resulting white-noise signal was scaled to obtain current fluctuations with a standard deviation of 8 pA. Injection of this input into the distal apical dendrite at a distance of 1094 μm away from the soma of the active cell held at -60 mV yielded local membrane potential fluctuations with a standard deviation of 1.5 mV. The exact same white-noise input was used in all simulations (i.e., so-called "frozen" noise).

Quasi-active approximation of voltage-dependent ion currents

Voltage-dependent membrane currents often behave in a near-linear fashion for small perturbations around a holding potential. This can be exploited by making linear approximations, so-called "quasi-active" models, of the nonlinear ionic currents (Mauro et al., 1970; Koch, 1984; Hutcheon and Yarom, 2000; Remme, 2014). In this way one can reduce the parameter space while retaining key dynamical features of the system. Results in Figure 2 used quasi-active currents to simplify the original, nonlinear cortical pyramidal cell model (see above) and to allow for a systematic study of the effects of active conductance on the LFP.

We here briefly describe the derivation of a quasi-active description of a single cellular compartment. The compartment includes one active current I_w that depends on the membrane potential $V(t)$ and is described by $I_w(t) = \bar{g}_w w(t)(V(t) - E_w)$, with peak conductance \bar{g}_w , reversal potential E_w , and gating variable $w(t)$. The passive leak current is given by $I_L(t) = g_L(V(t) - E_L)$, with conductance g_L and reversal potential E_L . Finally, the axial current, i.e., the net current entering or leaving to neighbouring cellular compartments, is denoted by $I_{axial}(t)$. The voltage of the compartment then evolves according to

$$c_m \frac{dV(t)}{dt} = -\bar{g}_w w(t)(V(t) - E_w) - g_L(V(t) - E_L) + I_{axial}(t),$$

where the term $c_m \frac{dV(t)}{dt}$ is the capacitive current with membrane capacitance c_m . The dynamics of the gating variable $w(t)$ is given by

$$\tau_w(V) \frac{dw(t)}{dt} = w_\infty(V) - w(t)$$

with voltage-dependent activation time constant $\tau_w(V)$ and activation function $w_\infty(V)$. The quasi-active description is obtained by linearizing V and w around resting-state values V_R and $w_\infty(V_R)$, respectively, by means of Taylor expansions. Defining the variable

$m(t) \equiv (w(t) - w_\infty(V_R)) / \frac{\partial}{\partial V} w_\infty(V_R)$, we can write the linearized equation describing the voltage dynamics of a single compartment:



$$c_m \frac{dV(t)}{dt} = -g_L \left(\gamma_R (V(t) - V_R) + \mu m(t) \right) + I_{axial}(t)$$

where $\gamma_R \equiv 1 + \bar{g}_w w_\infty(V_R)/g_L$, i.e., the ratio between the total membrane conductance (at V_R) and the leak conductance. The parameter $\mu \equiv (\bar{g}_w/g_L)(V_R - E_w) \frac{\partial}{\partial V} w_\infty(V_R)$ determines whether the quasi-active current functions as a positive feedback (when $\mu < 0$; i.e., a regenerative current) amplifying voltage deviations from the holding potential V_R , or as a negative feedback (when $\mu > 0$; i.e., a restorative current) counteracting changes in the voltage. When $\mu = 0$, the quasi-active current is frozen and functions as a static passive current (throughout the text we will refer to this as the "passive-frozen" case). The dynamics of the linear gating variable m is described by,

$$\tau_w(V_R) \frac{dm(t)}{dt} = V(t) - V_R - m(t).$$

The description of the ionic currents in a single compartment is easily extended to a multi-compartmental model where each compartment can have its own set of parameters to describe the passive and quasi-active currents.

For the simulations with a single linearized h-type current, we kept the passive parameters, as well as the peak conductance and activation time constant (at the specified holding potential) of the relevant active current, the same as in the original detailed model.

To systematically study the effect of the cellular distribution of a quasi-active current on the LFP, we used three different channel density distributions: (1) linearly increasing with distance from the soma, (2) linearly decreasing with distance from the soma, and (3) a uniform distribution. The slopes of the increasing (decreasing) distributions were set such that the most distal tip of the apical dendrite had a sixty-fold larger (smaller) density compared to that of the soma (in line with experimental estimates for I_h distributions: Mishra and Narayanan 2015; Lörincz et al. 2002; Nusser 2009; Kole et al. 2006), and the total membrane conductance of the quasi-active current (i.e., summed over all compartments) was the same as the total passive leak conductance g_L . The passive leak conductance was set uniformly to 50 $\mu\text{S}/\text{cm}^2$ for all cases. For $w_\infty(V_R) = 0.5$ the distance-dependent quasi-active peak conductance was $\bar{g}_w(x) = 5.29 + 0.242x$ $\mu\text{S}/\text{cm}^2$ for the linear increase, and $\bar{g}_w(x) = 143 - 0.109x$ $\mu\text{S}/\text{cm}^2$ for the linear decrease, where the distance x was measured in μm and had a maximum of 1291 μm . Note that the three distributions had the same total quasi-active membrane conductance summed over the neuronal membrane. The parameters μ and γ_R vary along the cell for the non-uniform channel distributions. We introduced $\mu^* \equiv \mu(x)g_L/\bar{g}_w(x)$, such that $\mu^* = (V_R - E_w) \frac{\partial}{\partial V} w_\infty(V_R)$ is independent of the distribution of the quasi-active conductance and can be specified as a single constant. We used $\mu^* = -0.5$ for the regenerative conductance, $\mu^* = 0$ for the passive-frozen conductance, and $\mu^* = 2$ for the restorative conductance. For the uniform distribution this gives the same values as those used in Remme and Rinzel (2011), namely $\mu = -1, 0, 4$ for the regenerative, passive-frozen and restorative cases, respectively. The activation time constant $\tau_w(V_R)$ of the quasi-active conductance was set to 50 ms in order to have dynamics similar to the h-type conductance. The intracellular resistivity was $R_a = 100 \Omega\text{cm}$, and the specific membrane capacitance $c_m = 1 \mu\text{F}/\text{cm}^2$.



Calculation of extracellular potentials

Extracellular potentials recorded inside the brain are generated by transmembrane currents from cells in the vicinity of the electrode contact (Nunez and Srinivasan, 2006). The biophysical origin of the recorded signals is well understood in the context of volume conduction theory. Extracellular potentials originating from a simulated multi-compartmental neuron model can be computed by first obtaining the transmembrane currents $I_n(t)$ from each compartment n at position \vec{r}_n . Next, the extracellular potential $\phi(\vec{r}, t)$ at position \vec{r} resulting from these transmembrane currents can be calculated (Holt and Koch, 1999; Lindén et al., 2014):

$$\phi(\vec{r}, t) = \frac{1}{4\pi\sigma} \sum_{n=1}^N I_n(t) \int \frac{d\vec{r}_n}{|\vec{r} - \vec{r}_n|},$$

where σ is the conductivity of the extracellular medium. This corresponds to the so-called line-source formula assuming the transmembrane currents to be evenly distributed along the axes of cylindrical neural compartments, see Lindén et al. (2014) for a detailed description.

All simulations and computations of the extracellular potentials were carried out using LFPy (Lindén et al., 2014), an open-source Python package that provides an interface to NEURON (Carnevale and Hines, 2006). The time step of the neural simulation was 0.0625 ms. For all simulations the first 1000 ms was discarded to avoid initialization effects. All simulation code used to produce the figures in the study is available.

2.2.2.3 Model Data

This work is the results of a collaboration between Torbjørn V Ness and Gaute T Einevoll at The Norwegian University of Life Sciences and Michiel W H Remme at The Institute for Theoretical Biology, Humboldt University Berlin. A repository containing all simulation code will be available at the HBP database, as well as at BitBucket (<https://bitbucket.org/torbness/alfp.git>). All simulation code is written in Python. Neural simulations were conducted using LFPy (Lindén et al., 2014), an open-source Python package that provides an interface to NEURON (Carnevale and Hines, 2006).

The main focus of this project was subthreshold cortical pyramidal cells, but using the quasi-active formulation, we were able to study generic features of the effects of subthreshold active conductance to the LFP, and we expect our findings to generalize to all brain regions and neural cells types that contributes to the LFP.

We have completed a thorough study of the effect of active conductances on the single cell contribution to the LFP (Ness et al. (2015), <http://arxiv.org/pdf/1512.04293v1.pdf>), and there is on-going work to extend these results to populations of cells. The first step will be to reproduce many of the results in Łęski et al. (2013).

2.2.2.4 Provenance

Not applicable

2.2.2.5 Model Results

With a morphologically detailed and experimentally constrained cortical pyramidal cell model, we demonstrated that active conductance could strongly shape the LFP stemming from subthreshold input. The precise effects depended on the position of the input, the position of the extracellular electrode, and the membrane potential of the cell (Figure 12). The subthreshold active conductance had distinct effects on the power spectral density (PSD) of the LFP, providing either amplification or attenuation of the low frequencies, in

the latter case leading to a resonance in the LFP-PSD. The effects on the LFP were observed for various spatial distributions of the active conductance across the cell, but generally required an asymmetric distribution of synaptic input onto the cell (Figure 13). We found that the effect of a subthreshold active conductance on the LFP was maximized for (i) an asymmetric distribution of the active conductance, (ii) when the input was targeted to regions where the active conductance was most strongly expressed, and (iii) the LFP was recorded on the opposite side of the cell with respect to the input.

Peaks in the LFP-PSD are commonly observed experimentally (see, e.g., Roberts et al., 2013; Hadjipapas et al., 2015), and such peaks are usually interpreted as the result of network oscillations, i.e., oscillatory firing activity driving the LFP-generating neurons (Buzsáki and Draguhn, 2004). Importantly, our work shows that such peaks may also be due to subthreshold restorative conductance molding the transmembrane return currents. In particular, in our simulations with the pyramidal cell model by Hay et al. (2011) we found that the h-type current had a prominent role in shaping the LFP and could cause a strong resonance in the LFP-PSD (Figure 1C). The h-type current is strongly expressed in cortical and hippocampal pyramidal neurons and has a particularly asymmetric distribution across the cell, increasing in density along the apical dendrites and peaking in the distal apical tuft dendrites (Magee, 1998; Williams and Stuart, 2000; Harnett et al., 2015). The h-type current can be expected to impact the LFP resulting from synaptic input that is predominantly targeting the apical dendritic tuft of populations of cortical or hippocampal pyramidal neurons. Indeed, various input pathways to pyramidal neurons target specific domains of the cell (see, e.g., Petreanu et al., 2009). An intriguing consequence of our work is that the LFP can contain information on the spatial distribution of subthreshold active channels, as well as on the location of synaptic input to the LFP generating cells. For example, given a known apical concentration of h-type conductance, a resonance in the LFP due to I_h would suggest that the cells receive asymmetric input. Vice versa, if asymmetrical input is provided, an absence of a resonance in the LFP (measured at multiple locations along the cell axis) suggests that the cell does not strongly express restorative currents.

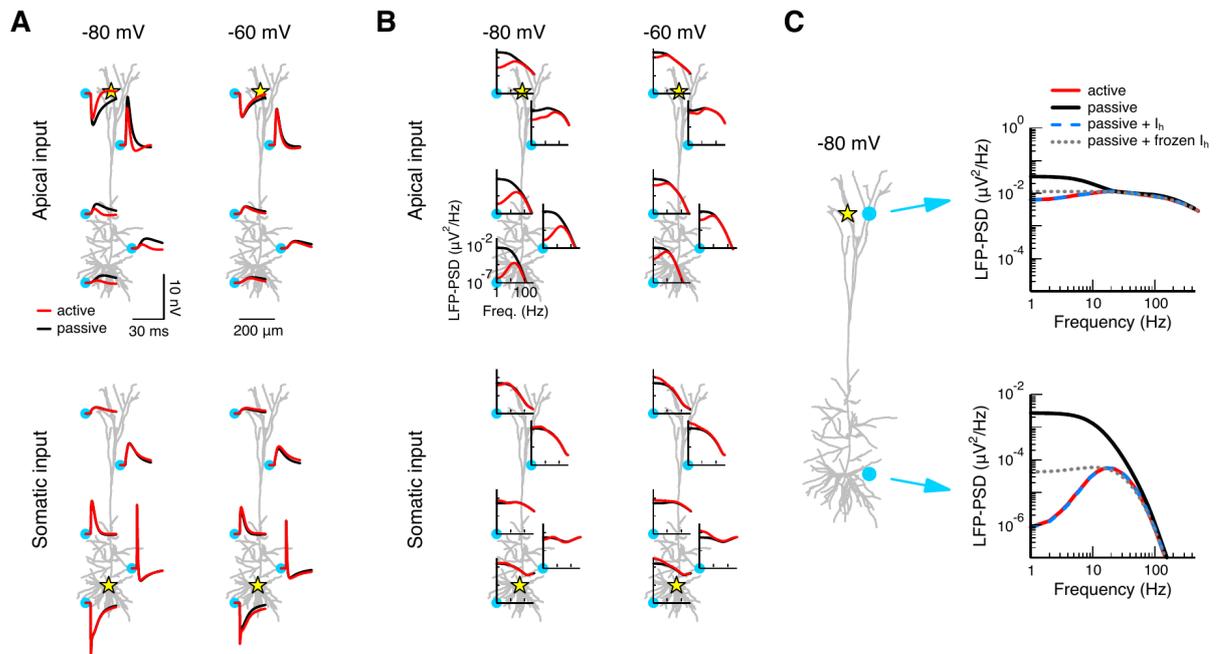


Figure 12: Active conductances can shape the extracellular signature of synaptic inputs.

A: A single synaptic input is provided to a cortical layer 5 pyramidal cell model. The extracellular response is shown at five positions (cyan dots) for two cases: the active model that includes various voltage-dependent conductances (red), or a passive model from which the active conductances have been

removed (black). The position of the input is marked by the yellow star: at the distal apical dendrite (top panels) or at the soma (bottom panels). The cell's resting potential was held uniformly at a hyperpolarized potential of -80 mV (left panels) or at a depolarized potential of -60 mV (right panels). The synaptic peak conductance was $0.001 \mu\text{S}$. The plots show the x,z-plane of the cell; the soma and the electrodes are positioned at $y=0$. B: As in panel A, but using white-noise current input instead of synaptic input, and displaying the response as the power spectral density (PSD). The PSD is calculated from 1000 ms long simulations. C: Apical input (marked by star) to the cell model held at a hyperpolarized potential (-80 mV). The extracellular potential is shown at two positions (cyan dots) for the active (red) and passive (black) model and for two additional versions of the model: the passive model supplemented by I_h (dashed blue), and the passive model supplemented by frozen I_h (dotted gray).

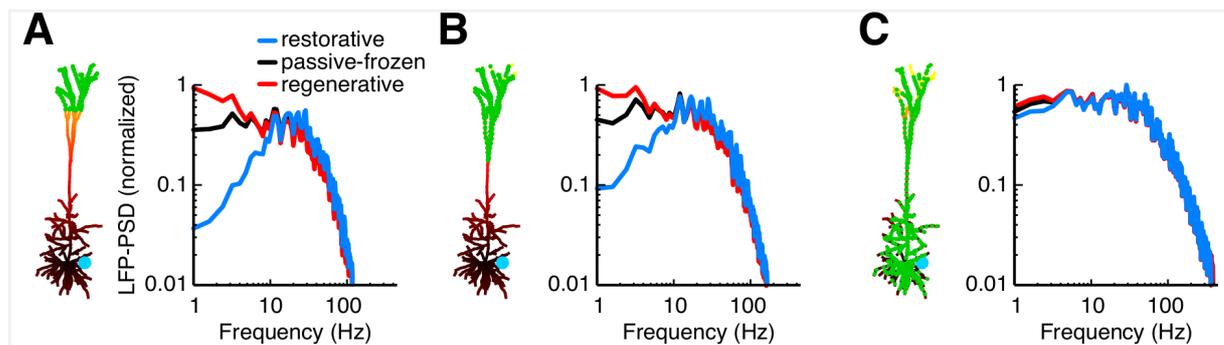


Figure 13: Resonance is retained for asymmetric input from distributed synapses.

A: The pyramidal cell model expressed a single linearly increasing quasi-active conductance that was either restorative (blue), passive-frozen (black), or regenerative (red). 1000 excitatory conductance-based synapses (green dots) with a peak conductance of $0.0001 \mu\text{S}$ were distributed across the distal apical tuft more than $900 \mu\text{m}$ away from soma. The synapses were activated by independent Poisson processes with a mean rate of 5 spikes per second. Simulations were run for 20 seconds and the LFP-PSD was calculated using Welch's method. B: As panel A, but with synapses distributed above the main bifurcation, $600 \mu\text{m}$ away from the soma. C: Synapses were distributed uniformly across the entire cell.

2.3 Task 4.1.3 Mechanistic Models of Cognition Linked to the Neural Substrate by Population Density Methods

2.3.1 M. de Kamps (ULEEDS)

2.3.1.1 Introduction

Task 4.1.3 has as objective the development of population level simulation methods, the production of a simulator available to the HBP - and any other interested parties - that allows more efficient simulation of large-scale neuronal networks, and can be applied in models of cognition. The point of population density techniques is that large populations of neurons can be simulated by modelling a single density function, and that population level quantities, such as firing rate, can be derived directly from the density function. We provide a simulation technique that is equivalent to spiking neuron simulations, at least when individual neuron identity does not matter, but require less run time and produce less data. Much effort has been invested into the development of efficient algorithms for population density techniques using 1-dimensional neural models, such as leaky-integrate-and-fire and quadratic-integrate-and-fire and exponential-integrate-and-fire neuron models.

Additionally, we have extended the methods theoretically and provided demonstrators to validate them: we have developed an algorithm that is applicable to any 2-dimensional neuron model. This means that we can model populations of model neurons as diverse as: adaptive-exponential-integrate-and-fire; Izhikevich; Fitzhugh-Nagumo; as well as conductance-based models (for a single conductance variable). We also have made progress towards spike trains with non-Markov distributed interspike intervals: we are able



to model populations subject to spike trains with gamma distributed spike intervals, for example. We found that in some cases superpositions of these processes can be analyzed, but that we cannot at present handle the general case of a superposition of such processes. We are currently in the process of integrating these novel algorithms into the main code base.

We apply our own work to models of cognition, in particular the problem of compositional representations in language. We are working with SP3 Pallier lab (NEUROSPIN, T3.6.2) and Twente University (non-HBP) to construct models of language representation, and below we will present a population level model on how humans handle some forms of ambiguity better than others. This model - the Neural Blackboard Architecture is used by Pallier lab in the analysis of MEG/fMRI data.

We are working with the Juelich (SP6 T6.2.3, now also SP4) group to apply population density methods in a reduced version of the Potjans-Diesmann (2014) model of a cortical column to allow for faster parameter sweeps and are discussing with UMB (Einevoll, T4.1.2) whether population density methods can be used in a more parsimonious model of Local Field Potentials (LFPs). In general, we believe our simulator is in a state where it can be used to great effect in reduced models of spiking neuron populations. We have improved the documentation and created a tutorial. In this document, we will present the simulator, explain how it can be installed and set up and provide many of the validation experiments that we have performed to ensure that it produces correct results. We would like this simulator to become a fixture in the simulator platform, and will strongly push its adaptation by other groups in the HBP. We are working with the Juelich group on a reimplementation of the Potjans-Diesmann model (T6.2.3) and hope to be able to contribute to T4.1.2. We will discuss the following aspects:

Installation and set up of the simulator

We have produced many examples, ranging from single populations to simple circuits of populations of leaky-integrate-and-fire and quadratic-integrate-and-fire neurons. These examples are described in <http://miind.sf.net/tutorial.pdf>. The examples are models. They are not novel, or original. Their purpose is to serve as validation and benchmarks.

A large-scale simulation, demonstration the MPI backend, and proof of scalability of the simulator. Large-scale network simulations will be based on this methodology. These examples are taken from <http://mind.sf.net/tutorial.pdf>

Possibilities and current limitations for the inclusion of spike train intervals that are drawn from a non-Markov processes, e.g. gamma distributions. Our results are summarized from <http://arxiv.org/abs/1601.07126>, which is currently under review in Physics Review E. The method is novel, and the examples can be considered to be models, but again, their purposes is validation.

Possibilities and current limitations for modelling 2-dimensional population densities. Earlier methods were restricted to one specific model. The method we have developed will handle any 2D model. We will present results on the quantitative validation of the method.

The models thus, serve an unusual purpose. They were necessary in the development of the simulation techniques and the simulator code, which is the bulk of the work for the ramp up phase.

2.3.1.2 Model Description for Miind

2.3.1.2.1 Introduction

The 1-dimensional population density methods, currently leaky- and-quadratic-integrate-and-fire neurons, are available at <http://miind.sf.net>, which serves as documentation. The code is available as a git repository there: 'git clone git://git.code.sf.net/p/miind/git miind-git'. A mirror repository is kept on github, in an anticipated move to that facility.



The git address mirrors the development version. Stable versions can be downloaded from: [‘https://sourceforge.net/projects/miind/’](https://sourceforge.net/projects/miind/). We will describe the installation process in Annex C. It details the required third party-packages and their installation, and then the installation of MIIND itself. We will now describe how to run the examples. These examples also cover comparison experiments between rabte-based models and population density techniques (PDT-RBMS;M283)

2.3.1.2.2 Demonstration

Starting with MIIND

We assume that you managed to install MIIND by the procedure explained in annex C. If not, please ask for assistance at M.deKamps@leeds.ac.uk.

In general there is no need to write C++ code, and you don’t need a background in C++ to work with MIIND. Simulations of a limited number of populations can be configured in an XML file, which are easier to create and store than the corresponding C++ code, and are more expressive. Once you have created an XML file that you want to run, you will invoke a Python script that translates the XML into C++ code and adds the resulting C++ code into the compilation tree. From that moment onward a C++ program corresponding to the XML file is part of your local MIIND installation until you decide to remove it. Invoking another Python script triggers compilation, will run the program and compile simulation results in a specific directory, which makes it easy to analyze them.

This set up is intended as an intermediate step towards a solution where XML files are submitted to an HPC facility and simulation results can be obtained from a server, possibly after a first remote analysis step. Even in such a set up a local MIIND installation will be useful.

Validating your Installation

Your top directory is called `miind-git`, we will refer to this directory as the **MIIND_ROOT**. In the **MIIND_ROOT**, you will find the directories `apps`, `libs`, `python` and `example`. Moreover, during installation you will have created a `build` directory, if it wasn’t there already. Within the `build` directory you will find the libraries and applications that were created during installation, also called `build` and `apps`. In the directory **MIIND_ROOT/build/apps/PerformanceGeom**, you will find the executables `LifOneCanvas`, `LifOne`, `LifTwoCanvas` and `LifTwo`. Run `LifOneCanvas`, by typing `./LifOneCanvas`, or by specifying the full path name. The program should run and a screen should pop-up that provides a running demonstration of the density display. You will see the full simulation of the single population of LIF neurons that was described in the last chapter.

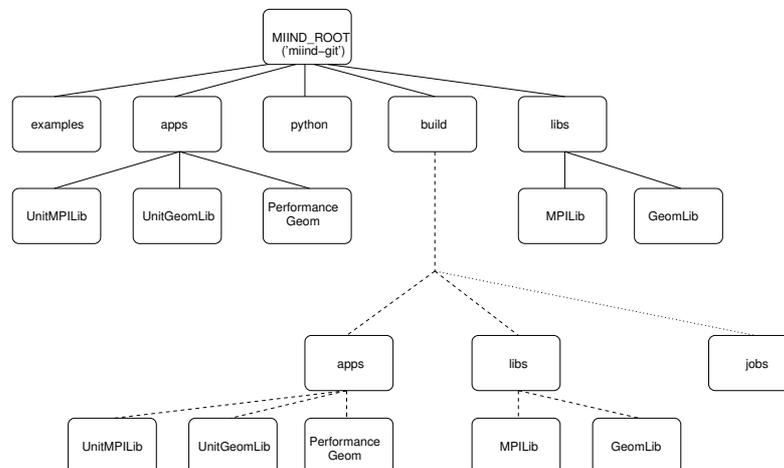


Figure 14: File structure of the MIIND installation.

Typically you will make a local installation, in a working directory of your choice, possibly on staging disks in an HPC environment. In your working directory you will find a directory called 'miind.git'. This is the MIIND_ROOT. The file structure is explained relative to this directory. A full explanation can be found in Section B, but you will want to add the python directory directly under the MIIND_ROOT to the PATH and PYTHONPATH environment variables. The window that pops up shows both the population density, the distribution of the population over the membrane potential, and the output firing rate: the fraction of neurons per unit time that are pushed across threshold and thereby emit a spike themselves. The pop up display is discussed in the figure below. You can also run `LifTwoCanvas` which describes a small circuit of two fully connected LIF populations, one excitatory and one inhibitory. You can run the executables: `LifOne` and `LifTwo` which perform the same simulation, but without online visualisation. As you can see these programs run much faster, and each produce a ROOT file that contains the simulation results.

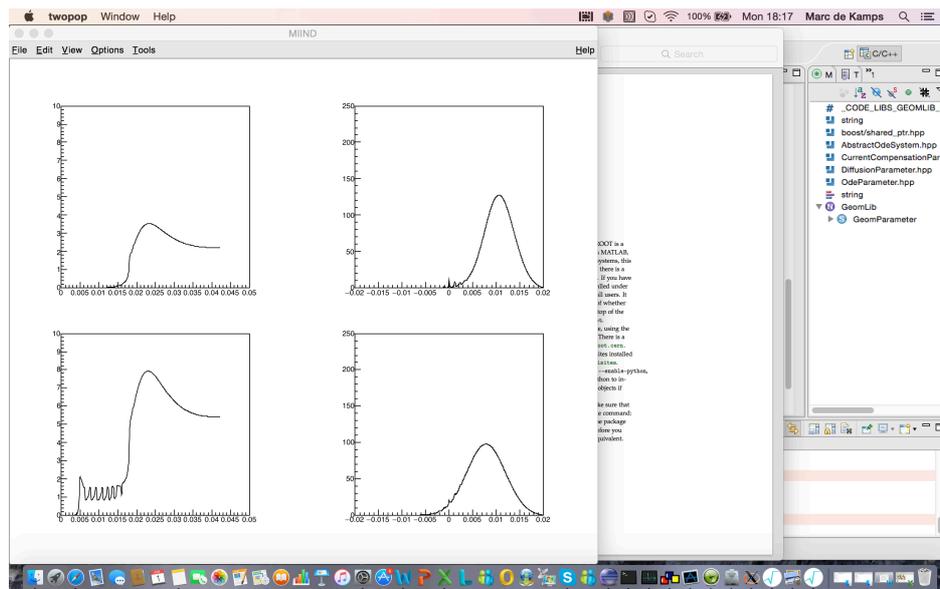


Figure 15: Monitoring running simulations

If the pop up screen is used, running simulations can be monitored. This is useful in setting up circuits and debugging. The use of a pop up screen slows down simulations considerably. On the right the population density function itself is shown as a function of membrane potential. On the left the population's output firing rate is shown as a function of time.

2.3.1.2.3 Running Example Simulations

In <http://miind.sf.net/tutorial.pdf> you will find a number of examples that provide a good introduction to population density techniques and running MIIND.

2.3.1.2.4 Large-scale Networks

We will use the example of the balanced excitation-inhibition in a two-population circuit to demonstrate performance of the simulator on a large-scale network. The two - population circuits are copied and these copies are placed in hexagonal rings around the original circuit.

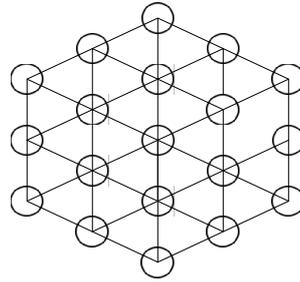


Figure 16: Network of two-population circuits

Each node represents a two-population circuit consisting of one excitatory and one inhibitory population, connected as described in the previous section. By increasing the number of rings, the number of populations in the network can be increased quadratically.

The inhibitory populations remain connected to the excitatory populations, but the excitatory populations acquire lateral connectivity. If it is assumed that the excitatory populations all fire at the same firing rate as the cortical background input, it is possible to work out how strong the lateral connections should be so that each excitatory receives on average as much input as in the two population circuit. Moreover, it is not difficult to do this as a function of the number of rings. By choosing an increasing number of rings, one can grow the network size quadratically in terms of number of populations. This makes the network suitable for scaling investigations.

The programme that resides in the [MIIND_ROOT/apps/largeNetwork](#) directory implements this network. This programme is built by default as part of MIIND's performance monitoring. At the moment this model is only used for investigating the scaling of parallelization, but we intend to use as an investigation into lateral wave phenomena in cortex. To investigate scaling, MPI must be used. This can be done as follows: restart the building process of MIIND with step 3: 'ccmake ..'. A toggle is present that allows the enabling of MPI. This requires the presence of BOOST.MPI in the boost libraries that the MIIND executable links to, and of course it assumes that MPI is installed on your host system.. If this is the case, you will be able to successfully compile MIIND. You must now use your host version of 'mpirun'. On the Leeds HPC cluster, we submit this as a batch job:

```
#$ -cwd -V
#$ -l h_rt=1:00:00
#$ -l np=64
mpirun ./largeNetwork
```

using 64 cores. It is now possible to investigate the scaling behavior of MIIND as follows: in `MIIND_ROOT/apps/largeNetwork/largeNetwork.hpp` is a variable:

```
const int NR_RINGS = 4
```

By changing the number of rings, the number of populations can be changed: the number of populations increases quadratically with the number of rings. For 25 rings, there are already 15000 populations in the network. In the model results section we will demonstrate that parallelization is nearly ideal: the total CPU time is almost inversely proportional to the number of cores used.



2.3.1.3 Model Data

2.3.1.3.1 Single Populations and Simple Circuits

A DIC has been created for the MIIND repository: '[MIIND code repository](#)'

Benchmark models are: a single population of leaky-integrate-and-fire neurons; a single population of quadratic-integrate-and-fire neurons; a circuit of two fully connected populations: one excitatory, one inhibitory, in a state of balanced excitation-inhibition; transfer curves for leaky-integrate-and-fire and quadratic-integrate-and-fire neurons. The core simulator is written in C++, the data format is that of the ROOT data analysis framework (<http://root.cern.ch>). Within ROOT a conversion is possible to NUMPY arrays.

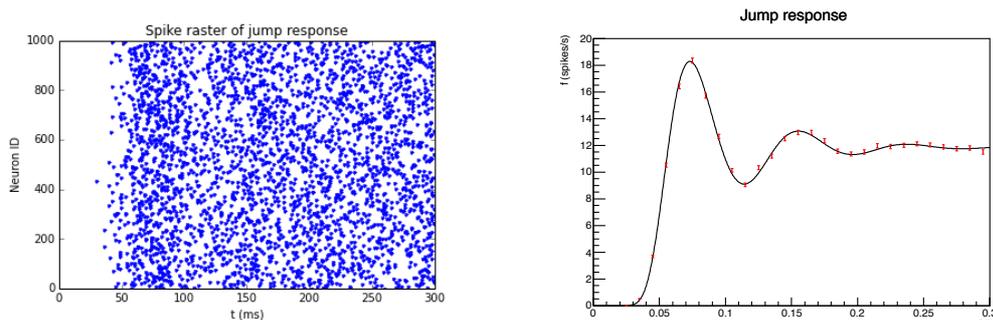


Figure 17: the response of a population of leaky-integrate-and-fire neurons to an input that has been switched on at $t = 0$.

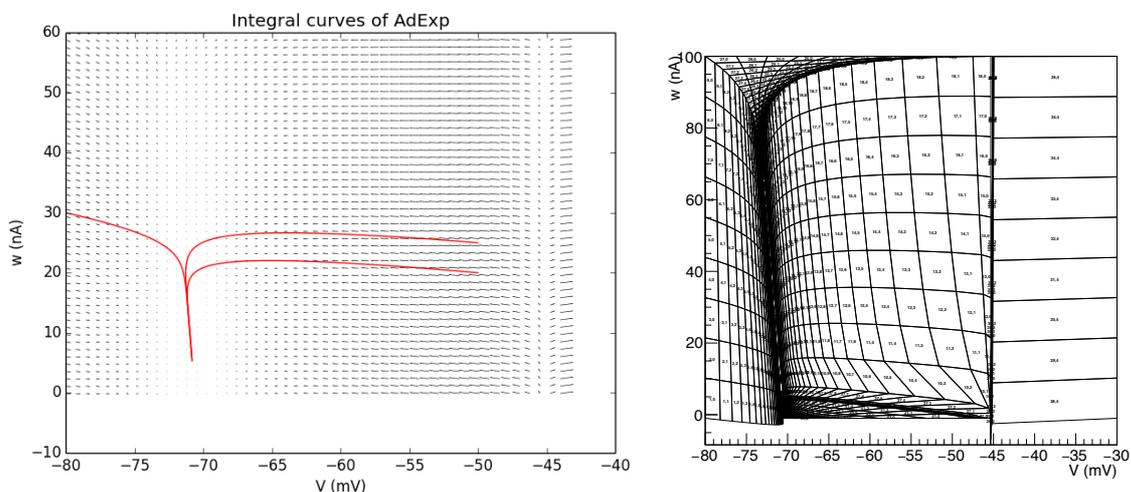
Left: a spiking neuron simulation, right: a calculation based on population density techniques (solid curve), the red markers are the population firing rate, calculated from the spikes of the spiking neuron simulation. Within statistics the agreement is perfect.

The response of a single population is shown in Fig.17. For reasons of space, we refer to the tutorial for the other examples. The tutorial has available on <http://miind.sf.net/tutorial.pdf>, and also has been submitted with this report.

2.3.1.3.2 A Universal 2D Population Density Solver

As explained above, we use population density methods to model populations. We will briefly outline the methodology for 2D neuronal models (1D neuronal models can be considered a special case and details can be found in <http://arxiv.org/abs/1309.1654>).

Consider an arbitrary 2D neuronal model, for example adaptive-exponential-integrate-and-fire. The 2D model is a two-dimensional vector field, and it is straightforward to compute its flow.



**Figure 18: AdExp model**

Left: vector field generated by AdExp model. Right: a geometric grid built from integral curves.

We use these integral curves to construct a mesh, built out of parallel strips, as shown on the right of Fig. 18. It is important to realise that neurons move through the grid in ascending order: the numbering of the grid is chosen in such a way that a group of neurons that is present in a grid cell is present in the neighbouring cell in the next time step. This is a so-called geometric grid. Importantly, such a grid can be constructed for any neuronal model. Fig. 19 shows a similarly constructed grid for a very different model: a conductance-based model, first considered in this form by Apfalter et al. (2006).

The horizontal axis describes the membrane potential, the vertical axis the conductance. As can be seen from the strip numbering, neurons at a high conductance move towards a higher membrane potential, and whilst they do this, the ion channel associated with the conductance closes: the conductance variable decreases. Below a certain conductance value, the neuron hyperpolarizes. Again, neurons move from cell to cell in each time step. It is now possible to define a density over the state space: consider the distribution of a group of neurons over the state space at $t = 0$. The density is then defined by the fraction of neurons, divided by bin area. If no synaptic input is present, the movement from cell to cell captures the evolution of the density. If a threshold is present (not shown in Fig.19), those neurons that move beyond, need to be reintroduced at a reset potential, but this is a trivial calculation. The firing rate can be computed as the fraction of neurons that cross the threshold per unit time.

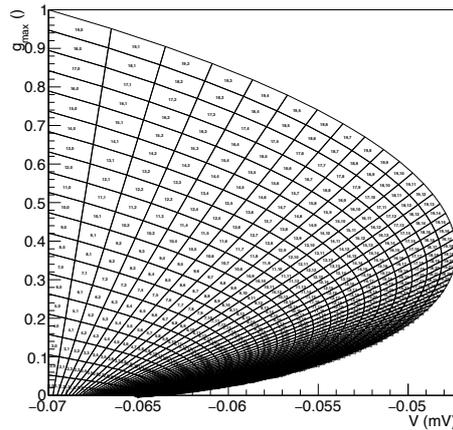


Figure 19: Mesh for a conductance-based model

The process is illustrated in Fig. 19. We store probability mass in an array that has as many elements as there are cells in the grid. The mass profile does not change in general, only the position of the mass in the grid is updated.

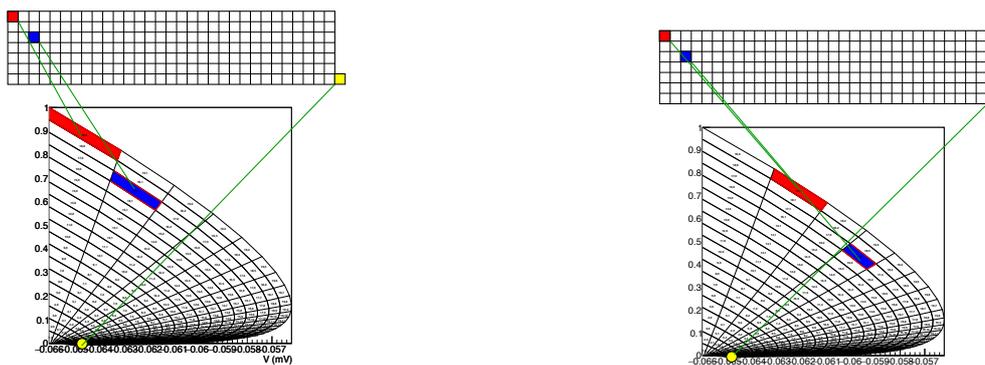


Figure 20: The same neurons (the same mass) moves through the grid from cell to cell.

It is sufficient to update the relationship between the mass array and the mass.

So the population density for any model neuron is trivial to model. This captures the so-called advective part of density evolution. With synaptic input things become more complex. We assume Poisson distributed input spikes. Each neuron in the population will see its individual realization of a Poisson process. In any neuronal model, an input spike will lead to an instantaneous change in the state. In LIF and QIF neurons with delta synapses, an input spike causes a jump in the membrane potential. In conductance-based models there will be a jump in the conductance. To model the evolution of the density, we will have to consider the Poisson Master equation as it manifests itself on the mesh. The problem is illustrated in Fig.21. It shows how neurons that are present in a given bin are distributed over other bins if these neurons receive a jump in conductance due to synaptic input.

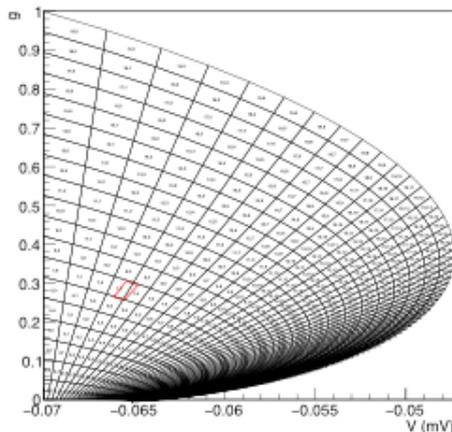


Figure 21: A fraction of neurons in bin (5,5) will be translated vertically due to synaptic input, as input spikes cause a jump in conductance.

To set up a Master equation, one needs to know what bins of the mesh the translated neurons cover. The problem can be stated as one of computational geometry: what proportion of which bins are covered by the translated bin (5,5)?

This problem is easily stated, but hard to compute efficiently. We do this offline, i.e. before simulation, using Monte Carlo techniques. When the translation coefficients have been calculated, the problem is that of solving an ordinary Poisson Master equation, which is a simple set of ordinary differential equation. We have developed the technique on AdExp neurons and qualitatively validated it on conductance-based models. The results will be demonstrated below. We consider this implementation successful. We hoped that we would obtain qualitative correct results for approximately 100000 bins, and this turns out to be the case. Our demonstrator in Python is too slow. It needs to be programmed more intelligently, and as a computational geometry application it is especially suitable for parallelization. We aim for a factor 40 speed up; we expect a factor 4 gain from moving to C, and from algorithmic improvements. We expect a more than factor 10 speedup by parallelization, which is currently being implemented. As a first step, we will produce an OpenMP version of the algorithm, and we hope to start soon on a Thrust-based implementation on a Tesla K80 GPU. We have underestimated the problem of computing the geometry of the Master equation, and found this more time consuming than expected. We now accept that this will probably have to be done offline, which is not in itself a fundamental problem. A potential solution to this problem may be the SpiNNaker architecture. Although not directly designed as such, it can function as a massively parallel database structure. It can in principle be used to find which grid cell a translated point resides in. We will explore this option with SP6. The current implementation is not yet part of the MIIND codebase, we will integrate the code after parallelization.

We consider the creation of a universally applicable 2D population density method a significant step. A few algorithms have been developed for individual models, e.g. by Casti et al. for the integrate-fire-and-burst model, and by Apfalter et al (2006) for the conductance based model, but the implementations were very specific to the model. Here, we have shown that any 2D model can be implemented, by solving the Master equation of the noise process on a geometric mesh. This is an inherently efficient method: it is easier to solve systems of ODEs than partial integro-differential equations. We consider the possibility to model 2D neurons a sea change. Just as 2D models of individual neurons are infinitely richer than 1D models such as leaky-integrate-and-fire models, but not as complex as e.g. a Hodgkin-Huxley model, and therefore more amenable to analysis, population versions of 2D models will offer a much richer toolset for modelling population dynamics.



2.3.1.3.3 Population Density Techniques for non-Markov Interspike Intervals

In population level modelling, the assumption that interspike intervals are drawn from a Poisson or white noise process is so commonly made that it is often not even stated explicitly. The assumption is implicitly present in diffusion-based approaches, such as Fokker-Planck equations. The potential impact non-Markov interspike intervals has been pointed out by Cateau and Reynes (2004). It is clearly an interesting question to see if population density techniques can be extended to include non Markov processes such as gamma distributed interspike intervals. There are some results: notably Ly and Tranchina (2009). For them, the starting point is usually a renewal process in two dimensions: the membrane potential and the time since last spike. The general framework that we presented for population density techniques above decouples deterministic motion from the stochastic process. The particular way in which we set up our solution means that we can model the stochastic process in its own terms: instead of solving the full partial integro-differential equation that describes the evolution of the density, we have to solve the Poisson Master equation on a geometric grid. This begs the question of whether generalizations of Master equations can be incorporated in our formalism. Such generalizations exist, for example in the form of the generalized Montroll-Weiss equation (Hoffmann et al, 2012), where the stochastic process is formulated in terms of waiting time distributions (WTDs), rather than transition probabilities. For a gamma distribution, this formalism leads to a Master-like equation, with a convolution of a finite kernel with the recent population density history. This approach is very attractive: first, there is no need to introduce an extra dimension. One-dimensional neuronal models require a one-dimensional density and as long as the kernel under consideration has a clear limited width, there is no need to consider the entire history. Second, generalizations to transient results are manifest; earlier results had only been derived for the steady state. We found the implementation straightforward: technical details are relegated to the write up: <http://arxiv.org/abs/1601.07126>.

In the next Section, we present our main results (M283; PDTS-RBMS). Here, we present a discussion of the limitations of the technique. We found that the technique worked well for a single channel. Both transient and steady state results were compared to Monte Carlo results and were satisfactory. The superposition of two renewal processes is in general not a renewal process, and we identified where in the generalized Montroll-Weiss equation the argument breaks down. It turns out that there the assumption is made that on an event the clocks for all WTDs reset, and that is in general not the case for computational neuroscience. We argued that for moderately high firing rates, this assumption is not dramatically violated, and proceeded to calculate the interaction of an excitatory and an inhibitory input process. We predicted that a 20% suppression compared to a linear addition of these processes should be expected and Monte Carlo validation bore this out, provided the processes were comparable in terms of firing rate, and synaptic efficacy. On the one hand, this is a nice result. On the other hand, it demonstrates that a study of balanced excitation and inhibition in full generality is difficult for non-Poisson processes.

Nonetheless, the method improves the state-of-the-art: it produces valid results for transients, and works similarly for excitatory and inhibitory processes. It inherently one dimensional, and shows transparently that the non-Markov character of spike intervals results in a convolution of a kernel with the recent density history. In earlier work this was shown to be the case for the steady state. We have shown this is generally true.

2.3.1.3.4 A Model of Sentence Ambiguity

Modelling language was the original driver for the development of MIIND. We have created a theory for the neural representation of linguistic trees, which was published as a target paper in *Brain and Behaviour Science* (van der Velde & de Kamps, 2006). We have further developed this architecture to account for so-called non-problematic ambiguities. Some sentences, garden-path sentences are clearly ambiguous and pose processing problems.

Other sentences are ambiguous, but can be easily processed, e.g. John knows Bill vs. John know Bill loves fish.

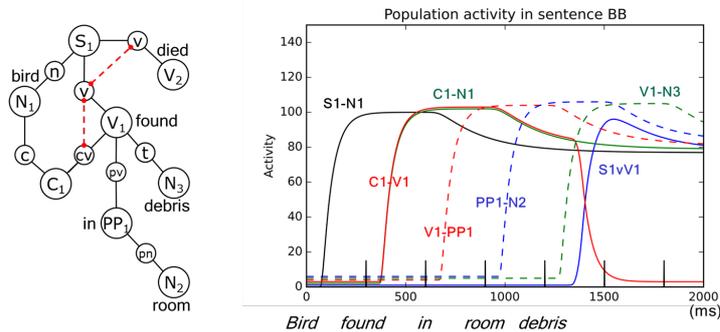


Figure 22: the neural blackboard architecture models linguistic trees in terms of populations.

Population dynamics plays an important role in ambiguity resolution by means of winner-take-all networks.

The purpose of the neural blackboard architecture is two-fold: first neural dynamics is instrumental in describing the computational processes that are required for language processing, but also should lead to a prediction of neural dynamics during sentence processing. With this document we submit our contribution to the NIPS workshop on Cognitive Computation.

2.3.1.3.5 Outlook

During the ramp up phase, our emphasis has been strongly on theory development and algorithm implementation. This is what we signed up for. Now that we have a functioning simulator, with most aspects necessary for modelling work implemented, we are keen to move away from this, and towards applications. We will continue the language modelling with the Pallier group (SP3), even if they will not be part of the HBP beyond the ramp up phase. Population level modelling is foreseen in co-development project 1 (CPD1), and the Juelich group has expressed an interest to adopt this simulation technique.

It is interesting to note that the Allen Brain Institute is developing a similar approach (Iyer et al, 2013): <http://alleninstitute.github.io/dipde/> DIPDE is similar for leaky-integrate-and-fire neurons, but does not at present have support for 2D neural models or non-Poisson statistics. We have recently invited Nicholas Cain to a workshop in the EITN to discuss commonalities and differences, and believe there is no fundamental problem with developing two population density simulators. First, as mentioned above, they have a different emphasis, second it allows the cross validation of results. Importantly, also for the Allen Brain Institute the motivation for the development is that spiking neuron simulations become to unwieldy, and need complementary lightweight techniques.

2.3.1.4 Results

2.3.1.4.1 Conductance-based results

Here, we show the evolution of density of a population of conductance-based neurons that

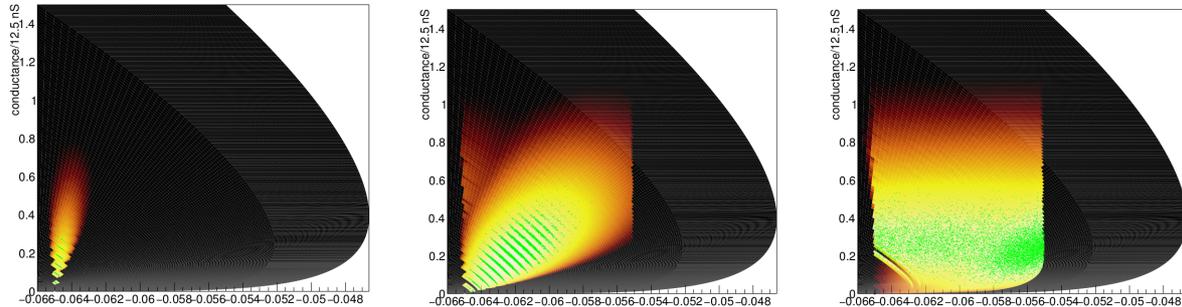


Figure 23 the density at three different moment in time ($t = 0, 8, 28$ ms).

Green dots are Monte Carlo events. The Monte Carlo events stick tightly to the white area. The heat plot is logarithmic so yellow and red areas will unlikely to have any events in them.

start at equilibrium. We have implemented a threshold at $V = -55$ mV in line with the original model by Apfaltrer et al (2006). The population firing rate can be calculated from the fraction of density that is being push over threshold per unit time. The population firing rate predictions are also accurate (Fig.24).

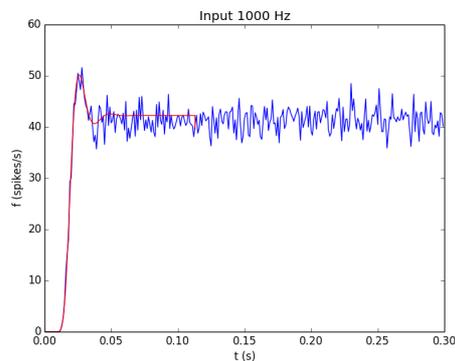


Figure 24: Population firing rate as calculated by the 2D method (red line) and inferred from Monte Carlo events (blue line) for a total input frequency of 900 spikes/s.

We conclude that the method is accurate.

2.3.1.4.2 Results for non-Poisson input spike trains

We start with a benchmark test that we have used before (originally from Omurtag et al (2000)). The benchmark is a population receiving a spike train of 800 Hz, with an efficacy h 3% of threshold value. For different shape factors, the efficacy is chosen such that the expectation value of the input is the same as for the Poisson case.

The results show that the population density predictions and Monte Carlo values correspond closely. The transients are clearly larger for larger shape factors. For this superthreshold stimulus the steady state firing rate is not dramatically affected.

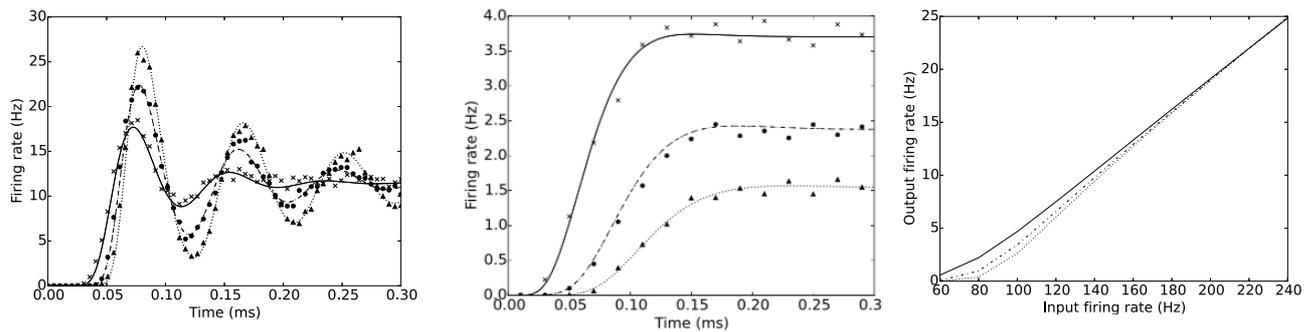


Figure 25: Non-Poisson input spike trains

Left: transient firing rate as a result of input being switched on at $t=0$. The results are dependent on shape factor, with stronger transients for higher shape values (circles:2, triangles 3). For superthreshold inputs the steady state is not affected. Middle: for subthreshold inputs there is a clear dependency of steady state results on the shape factor. Right: the effect is pronounced enough to affect the shape of the population's transfer function at low firing rates.

For subthreshold input, this is different, the steady state values are dependent on the shape factor. This effect is pronounced enough to affect the population's transfer function at low firing rates.

2.3.1.4.3 Results on scaling

We demonstrate the scalability of large-scale simulations, due to the MPI backend. MIIND is trivially parallel and an MPI implementation is relatively straightforward. Above, we have described how we can create a network of populations group in hexagonal rings. The number of population scales quadratically with the number of rings. The simulation time scales inversely with the number of cores used in simulation. It is therefore possible to determine the scaling theoretically, as well as to simply measure it. The results are shown in Fig. 26, and indicate almost perfect inverse linear scaling with number of cores used, i.e. a successful parallelization. The largest network has more than 15000 populations. Since each population represents hundreds or even thousands of neurons, truly large networks can now be simulated.

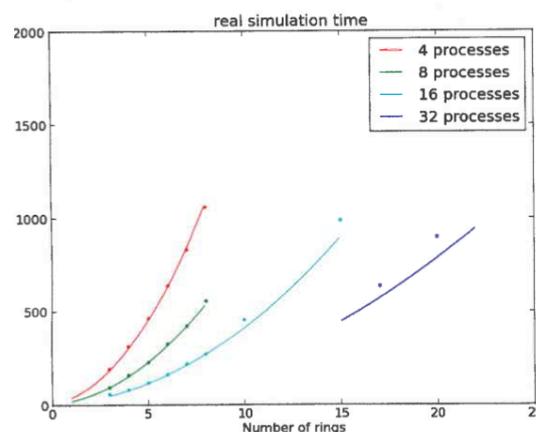


Figure 26: Simulation time as function of the number of rings in the network.

The measured simulation times are dot and lie mostly on the theoretic predictions (solid lines). This indicates nearly perfect scaling with the number of processors involved.



2.3.1.5 Publications

- Yi Ming Lai and Marc de Kamps (2016), <http://arxiv.org/abs/1601.07126>, submitted to Phys. Rev. E
- Frank van der Velde and Marc de Kamps (2015), Neural Information Processing Systems (NIPS) workshop on Cognitive Computation: Integrating neural and symbolic approaches, Combinatorial structures and processing in Neural Blackboard Architectures, Accepted oral presentation.
- M. de Kamps (2015), Using your MIIND (<http://miind.sf.net/tutorial.pdf>), In Preparation for a Neuroinformatics journal.
- Yi Ming Lai, Elaine Duffin and Marc de Kamps, Population Density Techniques for Modeling Neural Populations, ICMNS (2015), Accepted poster.

2.3.1.6 Provenance

All data is MIIND generated.

2.3.2 Task 4.2.1 Derive learning rules from biophysical synapse models

2.3.2.1 Introduction

In task 4.2.1, we have worked on four different models corresponding to four different algorithms, presented below. The models/algorithms were developed by the labs of Wulfram Gerstner (listed under A), Walter Senn (listed under B and C) and Misha Tsodyks (listed under D)

2.3.3 W. Gerstner (EPFL)

2.3.3.1 Introduction - A)

As part of task 4.2.1, EPFL-LCN (the Gerstner lab) developed the **ALGO STDPorchestrated** which combines Hebbian plasticity with heterosynaptic plasticity and transmitted induced plasticity. The main results have been published by Zenke et al. in ‘Nature Communications 2015’. Text in quotation marks in this report refers to quotations from that paper.

“The concepts of cell assembly and Hebbian learning have inspired generations of experimental and theoretical work. A cell assembly, loosely formulated as a group of neurons with strong connections among each other, can be interpreted as a functional circuit of brain activity. Cell assemblies may be activated during memory recall, as evidenced by delay activity of neurons during working memory tasks or during recognition of abstract items. While models of cell assemblies for fixed, preset connectivity can be readily constructed, the question of whether Hebbian learning rules can be used to *form* and *recall* such assemblies in a robust, stable manner is not well understood.

The reason why models fail to form functional memory assemblies in plastic networks of spiking neurons could be linked to either one specific or a combination of several features of biological networks, which were not addressed in these models. First, there are many different types of neurons in the brain, and experimental forms of plasticity depend on both the type of neuron and its connections. Second, plasticity manifests in multiple concurrently active forms. This includes, but is not limited to rate-dependent, voltage-dependent and spike-timing-dependent homosynaptic as well as heterosynaptic plasticity. Third, induction of synaptic plasticity needs to be distinguished from processes of synaptic consolidation and maintenance. Finally, additional nonstandard forms of plasticity such as structural plasticity, short-term plasticity (STP) or homeostatic synaptic changes complicate the picture.



Here we show that a well-orchestrated combination of a plausible Hebbian plasticity model³⁰ together with non-Hebbian forms of plasticity and globally modulated inhibitory plasticity leads to the formation of cell assemblies. Importantly, the emergent assemblies are stable and do not degrade or inflate during on-going activity and memory recall.”

2.3.3.2 Model Description A) ALGO STDPorchestrated:

C++ code based on Auryn simulation library; documentation on

https://www.fzenke.net/auryn/doku.php?id=examples:orchestrated_plasticity

“In order to distinguish different forms of plasticity in our model, we use the following terms and criteria. First, we call contributions to synaptic plasticity that depend only on the state of the postsynaptic neuron, but not on those of the presynaptic neurons, ‘heterosynaptic’. Manifestations of synaptic plasticity that depend jointly on pre- and post-synaptic activity are called ‘homosynaptic’. Similarly, changes of the synapse that depend only on the transmitter release, but not on the state of the post-synaptic neuron, are called ‘transmitter-induced’. By definition, heterosynaptic and transmitter-induced plasticity are non-Hebbian, while homosynaptic plasticity can either be Hebbian or anti-Hebbian. Second, in our terminology we also consider the timescale on which synaptic changes manifest themselves.”

“The stability in our model is a direct consequence of the orchestrated interplay of multiple plasticity mechanisms on different timescales. First, on the timescale of several hundred milliseconds the nonlinearity of STP creates the possibility for firing rate bistability in cell assemblies at intermediate levels of neuronal activity. Second, on the timescale of seconds, induction of plasticity is achieved by a combination of triplet STDP with heterosynaptic and transmitter-induced plasticity (see Methods). Transmitter-induced plasticity of strength δ , in our model, is proportional to the presynaptic activity $(pre)_j$ and ensures low neuronal baseline firing rates. Similar to earlier models, heterosynaptic plasticity of strength β changes all synapses on neuron i whenever the postsynaptic activity $(post)_i$ reaches a high value. The direction of change depends on the present value w_{ij} of the synaptic weight in relation to a reference weight \bar{w}_{ij} consistent with experiments of tetanic burst induction. The combination of transmitter-induced, heterosynaptic and Hebbian plasticity at the excitatory synapse between neuron j and the postsynaptic neuron i induces weight change schematically described by “the triplet STDP model where “LTP is quadratic in the postsynaptic variable and of strength A , whereas LTD is linear in the postsynaptic variable and of strength B_i . The fourth power in the description of heterosynaptic plasticity implements a threshold on the postsynaptic activity and acts as a burst detector.”

2.3.3.3 Model Data -A) ALGO STDPorchestrated:

Task(s)/group(s) responsible for generating algorithms/models/principles.

EPFL-LCN, Group of Wulfram GERSTNER, responsible Scientist has been Friedemann Zenke, now at Stanford University.

Simulation and model development by EPFL-LCN, Group of Wulfram GERSTNER, responsible Scientist has been Friedemann Zenke, now Stanford University.

Data, algorithms, tools and methodologies storage location(s) (and links?)

Simulation program, publicly available on

<https://github.com/fzenke/pub2015orchestrated>

Published paper publicly available on

<http://www.nature.com/ncomms/2015/150421/ncomms7922/full/ncomms7922.html>

Description of algorithms/models/principles:



Format, language if applicable.

C++ code based on Aurn simulation library; documentation l on https://www.fzenke.net/aurn/doku.php?id=examples:orchestrated_plasticity

Name of DICs/software catalogue/or HBP github project entries.

DIC entry generated under task 4.2.1

Description of data:

Species, sex, age, number of specimen/subjects.

Does not apply

Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

Brain region with memory function, network of excitatory and inhibitory neurons

Completeness of data/algorithms/models:

Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The model meets the anticipated target.

Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

Other aspects of plasticity, other neuron types, more brain areas involved in memory and learning should be added in the future

Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.

Online learning and maintenance of memory in the same network model has been always been considered as a challenging task. The fact that the model resulting from the work reported here came out at the level of Nature Communications shows the importance of the results. The scientific community remains interested in the problem as signified by a slightly earlier publication by the group of Brent Doiron that also appeared in Nature Communications. In contrast to their work, the synaptic memories are maintained in the model with orchestrated plasticity even in the absence of ongoing activity or prolonged and repeated recall of the same memory.

Outline state of validation work.

The validation was performed by checking the possibility of stable memory recall in a recurrent network with ongoing activity and ongoing synaptic plasticity.

Data Quality and Value:

Verification of data quality.

The main check for verification has been that model input on the level of individual synapses leads to collective network states that enable memory formation and memory recall via attractor dynamics.

Your subjective analysis of the value of the data/algorithms for the users.

The data resulting from simulations of the algorithm is of good quality.

Data/algorithm/model usage to date:



Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

The paper has been cited 4 times according to Google scholar from people outside HBP. Inside HBP, the aim is to transfer this algorithm (currently a stand-alone simulator) into the simulation platform and integrate it into larger software packages such as NEST.

Are the data/algorithms/models considered final?

The data resulting from simulations of the algorithm is of good quality

Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

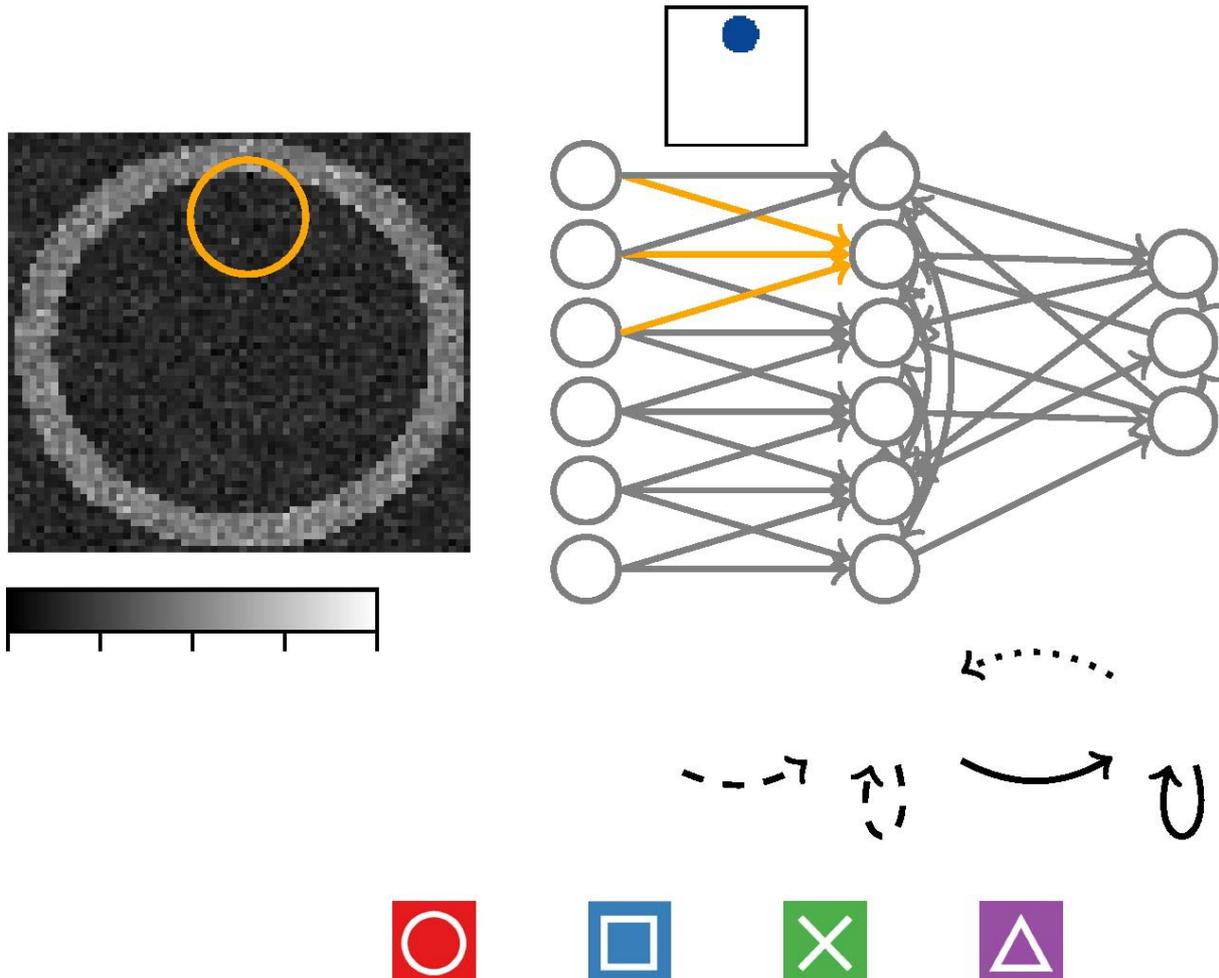
The paper has appeared as: F. Zenke, J.A. Agnes and W. Gerstner, Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks.

Nature Communication 6: 6922 doi:10.1038/ncomms7922

<http://www.nature.com/ncomms/2015/150421/ncomms7922/full/ncomms7922.html>

2.3.3.4 Model Results A) ALGO STDPorchstrated:

All results are reported here are quotations and images from the paper Zenke et al, cited above. This figure shows the layout of the network with an input layer (left) where four different pixel images are presented, connected to a network of excitatory and inhibitory neurons. Connections between excitatory neurons are plastic with the rule described above.



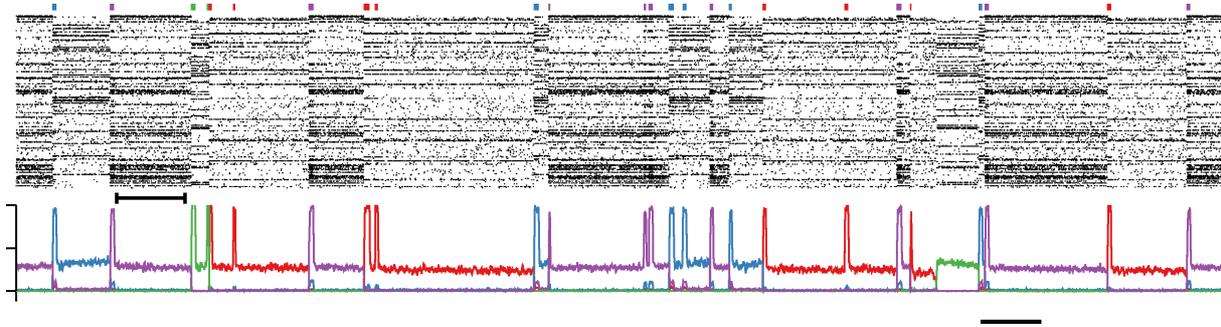
“We studied whether the combination of inhibitory plasticity with excitatory homosynaptic, heterosynaptic and transmitter-induced plasticity could work in symphony to enable stable assembly formation and recall in a spiking recurrent network model. To that end we implemented all forms of plasticity described above in the random network of excitatory and inhibitory neurons (Fig. above). Each excitatory neuron received recurrent input from the network, but also from a small patch of sensory neurons that defines the spatial location of its receptive field. All excitatory synapses were initialized with a common value such that the recurrent network exhibited asynchronous irregular firing. Synapses evolved freely according to the orchestrated plasticity rules described above. The network was then stimulated by applying repeatedly and stochastically one of four possible full-field input patterns (Fig. above). Stimulus identity, stimulus duration and interstimulus interval were randomized, while the stimulus intensity was kept fixed. Plasticity of feed-forward synapses induced the development of spatially structured feature detectors within the receptive fields that caused neurons to respond to specific input patterns.

Plasticity of recurrent excitatory connections led to the development of strongly connected assemblies (Fig. below), reminiscent of recent experimental findings in the sensory cortex. In our model, however, recurrent connections grew strong enough that assemblies could sustain selective delay activity following a brief stimulation of one of the patterns (Fig. below) consistent with signatures of attractor dynamics in experiments. Neurons that participated in an assembly exhibited a broad range of firing rates during delay activity. Background neurons had a firing rate of around 1 Hz or less and showed a large trial-to-trial variability. Some background neurons exhibited weakly inhibited or elevated responses to a specific assembly, whereas others did not. A large fraction of neurons did not belong to any of the assemblies (that is, never fired at high rate), which



suggests that there is a ‘reserve pool’ that could become sensitive to novel patterns not included in the stimulation paradigm.”

This figure below shows the spiking activity (black dots) of neurons in the network (vertical axis: neuron number) as a function of time (horizontal axis). Each of the four stimuli (color coded) is applied only for a very short time, but delay activity persists. The histograms in the lower part of the figure indicate the population activity in each of the four assemblies created through orchestrated plasticity (color code).



“To check whether recall is associative, we stimulated the plastic network with partial input by occluding up to three quarters of the input field. In most cases, we found activation of the appropriate assembly corresponding to the partial information, indicating memory recall from partial cues. Despite ongoing plasticity the learned assemblies were stable and did not degrade during days of ongoing network activity. Completely novel stimuli, unrelated to those previously encountered, or an ambiguous combination of known patterns could initiate memory recall of a single memory with overlap with the stimulated pattern”.

2.3.3.5 Provenance

Not applicable. Data is from other, previously published articles, prior to our outside HBP.

2.3.4 W. Senn (UBERN)

2.3.4.1 Introduction B) ALGO STDPpredictive:

The model documentation and all associated codes are available in pure Python at the GitHub repository https://github.com/dspicher/py_stdp. From this documentation:

“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, whereas 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 subserve 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 neocortex.

In a two-compartment formalism with somatic voltage U and dendritic voltage V , synaptic weights are updated according to

$$\Delta w(t) \sim (S(t) - \varphi(V^*))PSP^*(t)$$

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 as 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.”

2.3.4.2 Model Data B) ALGO STDPpredictive:

The code is available in pure Python from the GitHub repository https://github.com/dspicher/py_stdp. Note though that due to conceptual reasons, no NEST implementation currently exists (NEST so far does not allow to differentiate between somatic and dendritic potentials - this will only be addressed in SGA1, in particular in CDP5).

Task(s)/group(s) responsible for generating algorithms/models/principles.

UNIBE, Group of Walter SENN, responsible Scientist is Dominik Spicher (2nd year PhD student).

Data, algorithms, tools and methodologies storage location(s) (and links?)

Simulation program, publicly available on the repository

https://github.com/dspicher/py_stdp.

Description of algorithms/models/principles:

Format, language if applicable.

Currently in Python, but it is currently rewritten in the meta-language Brian to be made accessible to other scientists.

Name of DICs/software catalogue/or HBP github project entries.

DIC entry generated under task 4.2.1

Description of data:

Species, sex, age, number of specimen/subjects.

Does not apply

Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

Single cell with simplest dendritic morphology (two-compartment)

Completeness of data/algorithms/models:

Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The model is under development and evolves with the PhD project. On good track.

Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

The model should be extended to a more complex morphology (will be addressed in CDP5)



Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.

So far, optimal plasticity algorithms were tailored either to the supervised and reinforcement or to the unsupervised learning scenario. ALGO STDPpredictive, instead, represents a learning rule that can be used in all three learning scenarios. It is also a first algorithm that makes essential use of the intrinsic structure of a neuron.

Outline state of validation work.

The validation will be performed in the next period by applying the rule to specific learning tasks.

Data Quality and Value:

Verification of data quality.

As an intrinsic feature of an error-correcting rule, noisy perturbations are corrected out, making the algorithm very noise robust.

Your subjective analysis of the value of the data/algorithms for the users.

The algorithm is currently being implemented in VLSI.

Data/algorithm/model usage to date:

Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

This algorithm is one of the reasons to extend NEST during SGA1 such that NEST can also deal with dendritic structures.

Are the data/algorithms/models considered final?

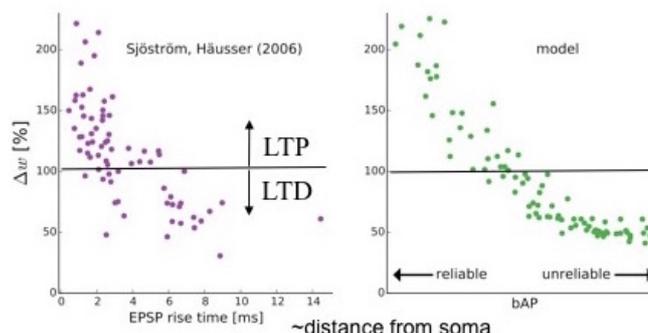
Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

The publication is still ahead (planned for the 3rd PhD year).

2.3.4.3 Model Results

Model Results - B) ALGO STDPpredictive: Only preliminary results are available that reproduce important experimental features of STDP such that the “switch” of LTP to LTD when the synaptic location moves distally (see Figure, data from Sjöström&Häusser, 2006).

In the model, the change from LTP to LTD with increasing distance of synaptic location from the soma is caused by failures in the AP backpropagation.





2.3.4.4 Provenance

Not applicable. Data is from other, previously published articles, prior to our outside HBP.

2.3.4.5 Introduction C)

In a third part the Senn lab developed the **ALGO STDPbackprop** which generalizes the optimal spike-dependent plasticity rule introduced by Pfister et al. 2006 to include dendritic NMDA-spikes. As other optimal rules it is derived as a gradient of a loss function. These can serve as a functional standard for a neuromorphic hardware implementation, and they may also guide the search for new biological forms of synaptic plasticity. This work was partly supported by the HBP, and the main research is Dr. Mathieu Schiess, former member of the Senn lab. The paper was published in PLoS Comp Biol (2016). The text in quotation marks refers to this paper.

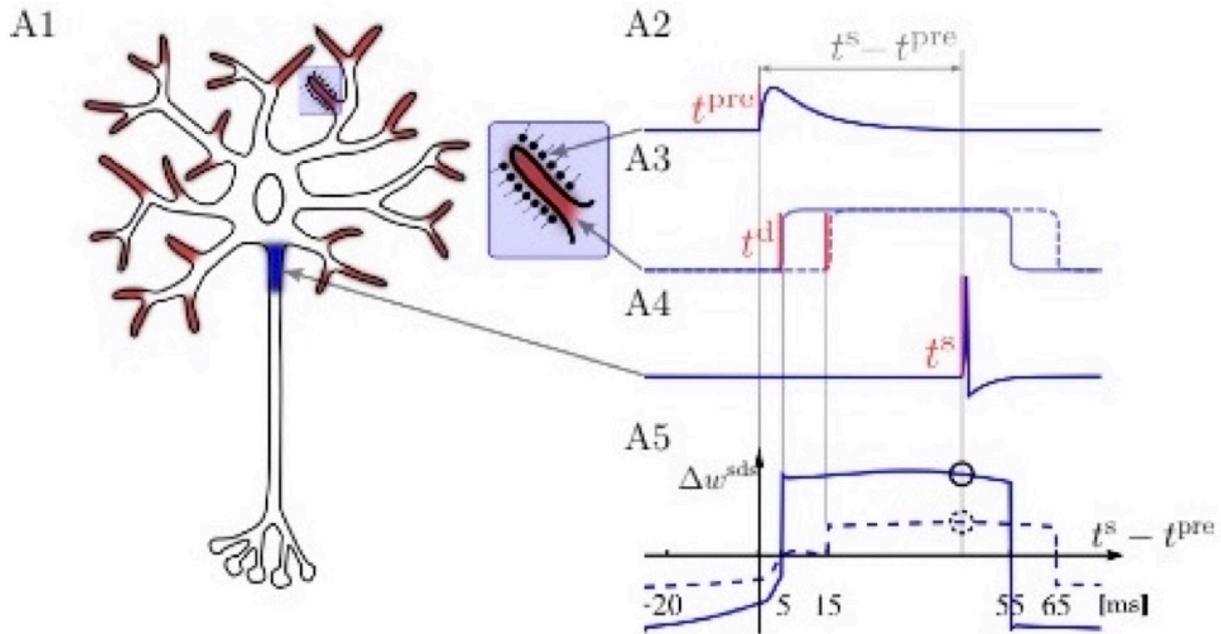
“Error-backpropagation is a successful algorithm for supervised learning in neural networks. Whether and how this technical algorithm is implemented in cortical structures, however, remains elusive. Here we show that this algorithm may be implemented within a single neuron equipped with nonlinear dendritic processing. An error expressed as mismatch between somatic firing and membrane potential may be backpropagated to the active dendritic branches where it modulates synaptic plasticity. This changes the classical view that learning in the brain is realized by rewiring simple processing units as formalized by the neural network theory. Instead, these processing units can themselves learn to implement much more complex input-output functions as previously thought. While the original algorithm only considered firing rates, the biological implementation enables learning for both a firing rate and a spike-timing code. Moreover, when modulated by a reward signal, the synaptic plasticity rule maximizes the expected reward in a reinforcement learning framework.”

2.3.4.6 Model description C)

ALGO STDPbackprop: The model generalizes the ALGO STDPpredictive to a neuron that is composed of a somatic compartment and many dendritic compartments that in parallel feed into to soma. As before a dendritic branch with local voltage V_w predicts the somatic firing rate as $\phi(\alpha V_w)$, where ϕ is the instantaneous rate function and α the dendritic attenuation factor. The dendritic weights (w_i) are adapted such that $\phi(\alpha V_w)$ is equal to the averaged somatic spike train $S(t)$ at each point in time. But different from ALGO STDPpredict, the learning rule is now modulated by a putative NMDA-spike in that branch and by the instantaneous rate of NMDA-spike triggering, $\phi(V_w)$. Overall, the rule becomes

$$\Delta w_i = \eta \left(S_i - \phi(\alpha V_w) \right) \left(\text{NMDA} + \phi(V_w) \right) \text{PSP}_i$$

where PSP_i is the postsynaptic potential of synapse i on the specific branch. Basically, the rule stretches the window of long-term potentiation if between the occurrence of a presynaptic and a subsequent postsynaptic spike (arriving at t^{pre} and t^{post} , respectively) when in between a NMDA-spike is triggered (at t^{d} , see Figure).



2.3.4.7 Model data C) ALGO STDPbackprop:

The code needs yet to be made available at GitHub. So far only Mathematica code is available. Because the algorithm at that stage represents a hypothetical rule for which no appropriate data is available, we do not (yet) plan a transcription into another language.

Task(s)/group(s) responsible for generating algorithms/models/principles.

UNIBE, Group of Walter SENN, responsible scientist is Mathieu Schiess (former PhD student, now private industry).

Data, algorithms, tools and methodologies storage location(s) (and links?)

- <https://github.com/mschiess/PLOS2016>

Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

Single cell with many parallel dendritic branches. Dendritic spikes are modeled as 50ms long depolarizations that are triggered by a voltage-dependent escape rate. It is shown that this model is realistic in the case of balanced excitatory and inhibitory input.

Completeness of data/algorithms/models:

Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The plasticity model has been shown to succeed in difficult learning tasks with delayed reward and linearly non-separable patterns.

Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

The algorithm yet waits on experimental validation or rejection (not planned within HBP).

Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.



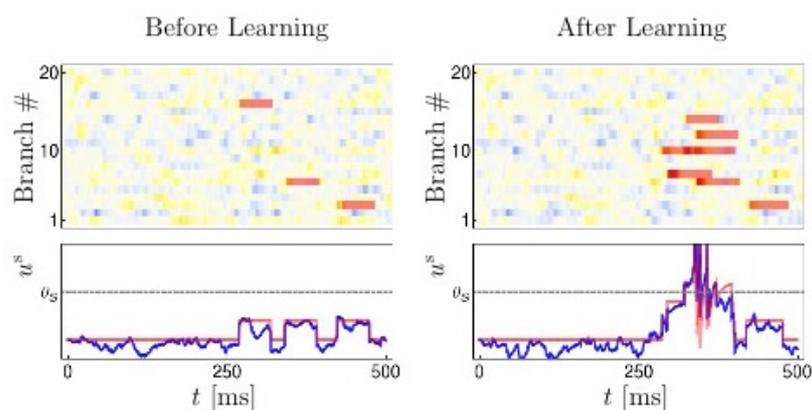
Various attempts exist to implement error-backpropagation in single neurons exist, but the current algorithm represents a mathematically rigorous way to formulate error-backpropagation in terms of a spike code.

Outline state of validation work.

The many simulations performed in the paper (Schiess et al., PLoS Comp. Biol. 2016) can be seen as a validation of theoretically deduced algorithm.

2.3.4.8 Model results C) ALGO STDPbackprop:

As an example of how the error-backprop algorithm for spikes works in dendritic structure we show the result of a simple learning task. A spatio-temporal presynaptic spike pattern that mimics the movement of a stimulus across the receptive field is repeatedly presented to the neuron. Before learning the stimulus does not evoke somatic spikes. But if during learning such spikes are imposed, the neuron learns to generate NMDA spikes (red) in a few dendritic branches that together depolarize the soma (with somatic voltage u^s) and generate a burst of action potentials. The neuron is shown to learn linearly non-separable tasks like the XOR-problem (Schiess et al. 2016).



2.3.4.9 Provenance

Not applicable. Data is from other, previously published articles, prior to our outside HBP.

2.3.5 M. Tsodyks (WIS)

2.3.5.1 Introduction D) ALGO LT-memory

The human long-term memory capacity for names, facts, episodes and other aspects of our lives is practically unlimited. Yet recalling this information is often challenging, especially when no precise cues are available. A striking example of this deficiency is provided by classical studies of free recall paradigm, where participants are asked to remember lists of unrelated words after a quick exposure. Even for short lists of 5-10 words most of participants are unable to reproduce them without omissions. These surprising results are still not accounted for in the framework of neural network modeling of memory. Many influential models of recall were developed in cognitive psychology. According to the 'search of associative memory' (SAM) model, items presented for recall acquire a set of mutual associations when stored temporarily in working memory buffers. These acquired associations are then used to retrieve words from memory. SAM can be fit to reproduce recall data with great precision, but since it has many parameters it cannot provide the first-principle explanation for very limited recall capacity observed in experiments. A recent model of (Romani et al 2013, Katkov et al 2015) introduced the notion that long-term associations between items determined by overlaps between their neuronal representations in memory networks, rather than short-term associations acquired during the experiment, are primarily responsible for recall process. With a simple phenomenological implementation of recall, this assumption results in a generic limit for



the recall capacity compatible with the data (Romani et al, 2013). Moreover, the neuronal representations determine the recall probability of different items ('easy' vs 'difficult' words) and the order of their recall. In the current contribution, we develop a more realistic neural network model where recall is mediated by the sequential reactivation of neuronal ensembles encoding different items in memory. As suggested in (Romani et al, 2013) the transitions between the memories is driven by periodic modulation of the feedback inhibition that pushes the network to oscillate between the attractor memory states and intersection between these states. We identify different phases of the model with mean-field analysis of the network dynamics and perform extensive numerical simulations to characterise the recall behavior of the model.

2.3.5.2 Model description

We consider a Hopfield neural network of N rate-neurons. The dynamics of the network is defined by the equation for the synaptic currents:

$$\tau \dot{c}_i(t) = -c_i(t) + \sum_{j=1}^N J_{ij} r_j(t) + \xi_i(t)$$

$$r_i = g(c_i)$$

where $c_i(t)$ is the synaptic current of neuron i at time t , r_i the firing rate, ξ_i is a white Gaussian noise. The gain function is:

$$g = x^\gamma \quad x > 0$$

$$g = 0 \quad x < 0$$

where $\gamma < 1$ resulting in the sublinear growth of gain function with increasing inputs.

The matrix of recurrent connections J_{ij} is determined by the set of P random memory patterns. Each memory is a binary vectors of N bits:

$$\eta^\mu = \{100011101001..1001\}$$

where each neuron belongs to a given memory with probability f , representing the sparseness of the memory representation.

These memories are stored in the network as attractors through the Hebbian learning rule for recurrent connections (Tsodyks & Feigelman, 1988)

$$J_{ij} = \frac{1}{N} \left(\sum_{\mu=1}^P J_1 (\eta_i^\mu - f)(\eta_j^\mu - f) - J_0 \right)$$

where J_1 and J_0 are two parameters that respectively define the strength of excitation and inhibition in the network. Their ratio is defined as $\phi = J_0 / J_1$. When simulating the network, all parameters are held constant except for the relative strength of inhibition, ϕ (see below).

We say that a particular memory item is 'recalled' when the corresponding memory pattern is active. To quantify the degree of memory activations we introduce the 'overlaps' defined as



$$m^\mu(t) = \frac{1}{N} \sum_{i=1}^N (\eta_i^\mu - f) r_i$$

$$m^0(t) = \frac{1}{N} \sum_{i=1}^N r_i(t)$$

While $m^0(t)$ measures the average firing rate in the network at time t , each m^μ measures the difference between the average firing rate of neurons encoding memory μ and all other neurons.

Simulation technique.

To study the influence of finite size effects and noise on the dynamics of the network we simulate the dynamic of a network of $N = 10^5$ neurons. To achieve this goal we simplify the dynamic equations above. This is dimensionality reduction of the network that reduces the number of simulated units. All the neurons that are in the same population \hat{v} such that $\hat{\eta}_i = \hat{v}$ can be described by a single unit. For these neurons the afferent connections given by the matrix J are identical. Each neuron receives the same input and projects equally on other neurons. It is not possible to differentiate their activity except for the effect of the noise term. But in the above dynamical equations we can average terms which share the same connections averaging also their noise. For a given realization of the network we can write the fraction of neurons in a given population \hat{v} as $S_{\hat{v}}$. Defining $c_{\hat{v}}(t)$, the averaging synaptic current for a neuron in population \hat{v} at time t , it is then possible to write an equation for the dynamics of $c_{\hat{v}}(t)$:

$$\dot{c}_{\hat{v}}(t) = -c_{\hat{v}}(t) + \sum_w J_{\hat{v}w} S_w g(c_w(t)) + \xi_{\hat{v}}(t)$$

These equations is a reduction of the original dynamical system, it has 2^P equations instead of the N . In this reduction the only piece of information which is not accessible is the precise value of the firing rate of each single neuron. Only the average firing rate of the population it belongs to is now accessible.

Although in principle the system has 2^P equations, in practice, due to the finite size of the network and its sparse connectivity, there are much less populations since $S_{\hat{v}} = 0$ for most \hat{v} . In this framework, for $P = 16$ we are able to simulate easily a large network of $N = 10^5$ neurons.

2.3.5.3 Model data D) ALGO LT-memory

- Task(s)/group(s) responsible for generating algorithms/models/principles.
- Data, algorithms, tools and methodologies storage location(s) (and links?)
- Description of algorithms/models/principles:
 - Format, language if applicable.
 - Name of DICs/software catalogue/or HBP github project entries.
- Description of data:
 - Species, sex, age, number of specimen/subjects.
 - Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.



- **Completeness of data/algorithms/models:**
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The model broadly fits the expectations.

- Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- **Outline state of validation work.**
 - **Data Quality and Value:**
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.
 - **Data/algorithm/model usage to date:**
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.
 - **Are the data/algorithms/models considered final?**

No. We are currently working on extending the model to account for recent observations on free recall data, in particular the spontaneous emergence of chunking.

- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

Recanatesi S, Romani S, Katkov M and Tsodyks M. ‘Neural network model of memory retrieval’. *Front. Comput. Neurosci.*, 17 December 2015

<http://dx.doi.org/10.3389/fncom.2015.00149>

2.3.5.4 Model results D) ALGO LT-memory

Time course of retrieval.

The example network activity generated by the model is presented in Fig. 27, showing the characteristic transitions between different memory attractors caused by oscillating inhibition. Since the recall of subsequent memories is a stochastic process triggered by noise in the input, we perform multiple simulations to characterize the average accumulation of recalled memories with time (see figure 28 below). We observe that after a quick initial accumulation of retrieved memories, the retrieval process slows down sharply, however the number of memories recalled continues to increase. This behavior is compatible with experimental observations and with results obtained by stochastic implementation of the free recall model presented in (Katkov et al, 2014). The time between the recall of subsequent items (inter-retrieval time, IRT) is highly variable as shown in the figure. Even after very long time-intervals it is possible to retrieve new items, in line with the experimental findings.

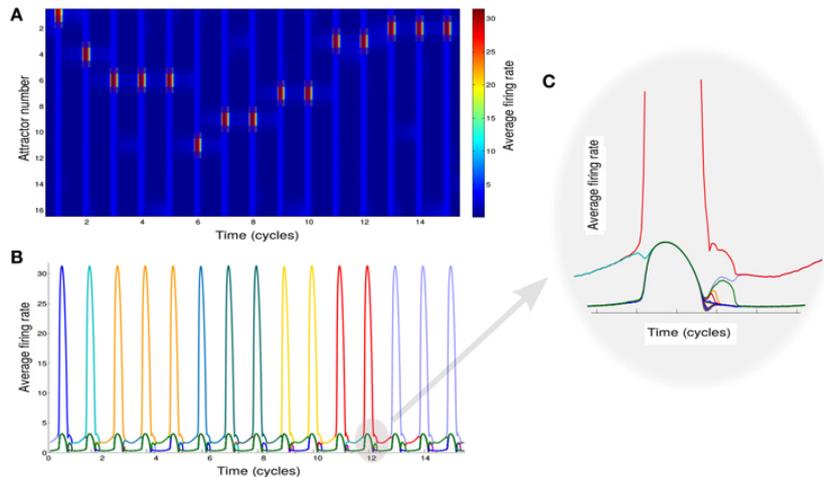


Figure 27. Neural network activity.

(A) Activity of the attractors in the network. Different rows correspond to the average firing rate of different memories for 15 cycles of oscillation of φ . (B) Activity of the attractors in the network. Each colored line corresponds to the average firing rate of a different memory. (C) Details of the neuronal dynamics.

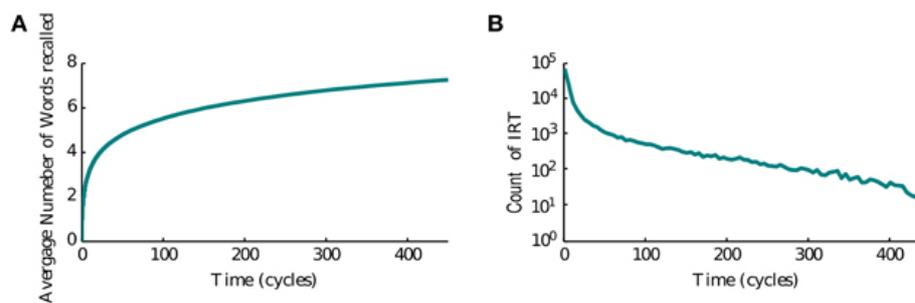


Figure 28. Temporal properties of recall.

(A) Average number of words recalled vs. time. (B) Distribution of the IRTs.

2.3.5.5 Provenance

Not applicable.

2.4 Task 4.2.2 Unsupervised learning rules and emergent connectivity

2.4.1 W. Gerstner (EPFL)

2.4.1.1 Introduction

All quoted text and the figures reported here are excerpts from a draft article by Moritz Deger & Wulfram Gerstner, which will be submitted for publication soon.

“Excitatory synaptic inputs on pyramidal cells of the mammalian neocortex are almost exclusively made on postsynaptic dendritic spines. Dendritic spines are small protrusions of the dendrite that vary in size and shape, and are subject to ongoing plasticity in the developing as well as in the adult brain [Yasumatsu2008, Holtmaat2009]. “

“Previous combined electrophysiological and anatomical studies in the somatosensory barrel cortex of the mouse have established that neocortical synapses between pyramidal neurons are sparse, but in case a synaptic connection exists between two neurons, it is typically made of several synaptic contacts [Feldmeyer1999, Fares2009] -- which means



that postsynaptically the connection involves several dendritic spines. The number of potential synaptic contacts between neuron pairs, estimated from reconstructed neuronal morphologies, is typically greater than the number of actual contacts (see Fig. 1D) [Fares2009]. Most importantly, the distribution of actual contact numbers cannot arise from independent synapse formation [Fares2009] but instead requires a mechanism of cooperation between synaptic contacts onto the same neuron. “

“We hypothesize that cooperative synaptic contact formation can be explained by spike-timing dependent plasticity models. Synaptic contacts connecting a neuron A to another neuron B share the same pre- and postsynaptic neuronal activity: presynaptic spikes cause Glutamate release and eventually leave a biochemical trace at each postsynaptic density, while information about postsynaptic spikes reaches the dendritic spines by means of backpropagating action potentials. Now if the plasticity rule that governs synaptic contact dynamics is sensitive to spike-timing correlations, the contacts indirectly influence each other because changes in the postsynaptic firing influence all synaptic contacts onto that cell. Previous modeling studies [see Deger2012 and references therein], however, have considered structural plasticity as being distinct from synaptic plasticity, by virtue of an assumed separation of time scales.”

“Thus STDP-based models of synaptic plasticity have not yet been linked quantitatively to data of dendritic spine plasticity and turnover [Yasumatsu2008, Holtmaat2009], and the statistics of synaptic contact numbers [Feldmeyer1999, Fares2009]. Here we present a quantitative model that links continuous synaptic plasticity of dendritic spine volume to discrete structural plasticity of synaptic contact formation and removal.”

see references

2.4.1.2 Model Description for Algo STDP-structural

All quoted text and the figures reported here are excerpts from a draft article by Moritz Deger & Wulfram Gerstner, which will be submitted for publication soon.

“Synaptic contacts in our model follow a variant of spike-timing dependent plasticity (STDP), see also Fig. 29. Each contact is described by its efficacy (weight) $w_{jk}(t)$, which is the amplitude of the excitatory postsynaptic effect that the contact k evokes upon transmission of an action potential of the presynaptic neuron j . Each presynaptic neuron j may be connected to the postsynaptic neuron by several synaptic contacts, up to a maximum of n_j , $1 \leq k \leq n_j$, cf. Fig. 1A. The number n_j of potential contacts of a connection is random, with a probability distribution (Fig. 29D, blue line) estimated by [Fares2009] for synapses connecting Layer 4 pyramidal neurons within a maximum distance of $50 \mu\text{m}$ in the barrel cortex of the rat. For computational reasons we limited n to a maximum of 10 and renormalized the distribution $P(n)$. Accordingly each possible value of n should appear (in expectation) $P(n) \cdot N$ times, where N is the number of presynaptic neurons. Therefore we randomly assigned groups of $P(n) \cdot N$ presynaptic neurons to each potential contact number n in order to exactly reproduce $P(n)$, the reported distribution of contact numbers per connection.”

“Synaptic contacts with a (unit-less) weight $w_{jk}(t) > 0$ (active contacts) evolve according to a local STDP rule

$$\frac{d}{dt} w_{jk}(t) = a_{2,corr} C_{jk,post}(t) - a_{4,corr} C_{jk,post}^2(t) - a_{4,post} R_{post}^4(t) - \alpha w_{jk}(t), \quad (1)$$

with the parameters $a_{2,corr} = 10^{-6} \text{s}$, $a_{4,corr} = 0.02453 \cdot 10^{-6} \text{s}^3$, $a_{4,post} = 0.01630 \cdot 10^{-6} \text{s}^3$ and $\alpha = 1.27142 \cdot 10^{-6} \text{s}^{-1}$. The terms in (1) are illustrated in Fig. 29C. As soon as a contact has weight $w_{jk}(t) \leq 0$, however, its weight is set to 0 and the dynamics cease (potential



contacts, see also Fig. 29B). However, with a rate of $\lambda_c=2.8\%/day$, each potential contact ($w_{jk}=0$) may be randomly transformed into an active contact again. In such an event, called creation, its weight w_{jk} is set to a low, non-zero value w_c , and its dynamics again follow (1) (see Fig. 29B).”

“The term $C_{jk,post}$ is a slow filtering trace of the correlations of pre- and postsynaptic spiking activity, and R_{post} of the postsynaptic activity. These traces are defined by the differential equations

$$\tau_{slow} \frac{d}{dt} C_{jk,post}(t) = -C_{jk,post}(t) + r_{jk}(t) \cdot r_{post}(t) \quad (2)$$

$$\tau_{slow} \frac{d}{dt} R_{post}(t) = -R_{post}(t) + S_{post}(t) \quad (3)$$

with a time-constant of $\tau_{slow}=2min$. The terms $r_{jk}(t)$ and $r_{post}(t)$ are filtered traces of the pre- and postsynaptic activities $S_{jk}(t)$ and $S_{post}(t)$, defined by the differential equations

$$\tau \frac{d}{dt} r_{jk}(t) = -r_{jk}(t) + S_{jk}(t), \quad (4)$$

$$\tau \frac{d}{dt} r_{post}(t) = -r_{post}(t) + S_{post}(t), \quad (5)$$

with a time constant $\tau=20ms$, as is typical for an excitatory postsynaptic potential (EPSP) or an STDP window function. This leads to a symmetric STDP window, but the formalism can be extended to a more standard STDP with long-term potentiation and depression by using more traces with different time constants.”

“Finally, the term

$$S_{jk}(t) = \sum_{t_{pre}} \delta(t - t_{pre}) z_{jk}(t_{pre}) \quad (6)$$

is the spike train of the presynaptic neuron j (with spike times t_{pre}) as it is received at the postsynaptic density of the contact k . Some spikes are not transmitted at each contact due to transmission failures. Specifically, the spike train $S_{jk}=S_j \cdot z_{jk}$ differs from the presynaptic spike train $S_j(t) = \sum \delta(t - t_{pre})$ through the (Bernoulli) random variables $z_{jk}(t_{pre}) \in \{0,1\}$ that describe stochastic failures of synaptic transmission. Such synaptic failures occur randomly and independently with a probability of $p_f=20\%$, so $z_{jk}(t_{pre})$ is 1 (successful transmission) with probability $1 - p_f$. “

“To enable us to perform mathematical analysis of the model’s dynamics, we assume a minimal model of the postsynaptic neuron. In the absence of synaptic input from the presynaptic neurons, the postsynaptic neuron fires with a baseline firing rate $\lambda_0=1/s$ (as a Poisson process). We further assume that synaptic inputs cause transient increases of the firing rate on the typical time-scale τ of an EPSP. We thus model the dynamics of the postsynaptic neuron’s firing rate $\lambda(t)$ as

$$\tau \frac{d}{dt} \lambda(t) = -(\lambda(t) - \lambda_0) + \sum_{j=1}^N \sum_{k=1}^{n_j} w_{jk}(t) S_{jk}(t), \quad (7)$$

here the second term sums all inputs across all n_j synaptic contacts over all N presynaptic neurons. We further assume that all presynaptic spike trains are independent Poisson processes with a constant firing rate of $\nu_{pre}=5/s$. “

“We performed direct simulations of the system by analytical integration of (1) between any successive pre- and postsynaptic spikes. The synapse model was implemented in NEST,



and this modified version of NEST was used to perform the simulations. The NEST neuron model that we used in this simulation is called “pp_psc_delta”, which implements (7). The complete system state (w_{jk} , r_{jk} , r_{post} , $C_{jk,\text{post}}$, R_{post} , λ) was recorded in intervals of 5min.”

2.4.1.3 Model Data for Algo STDP-structural

T4.2.2 “Unsupervised learning rules and emergent connectivity”

Task(s)/group(s) responsible for generating algorithms/models/principles.

4.2.2 / EPFL-LCN / Moritz Deger & Wulfram Gerstner

Data, algorithms, tools and methodologies storage location(s) (and links?)

Model will be published as part of NEST, <https://github.com/nest/nest-simulator>.

Description of algorithms/models/principles:

Format, language if applicable.

C++ code based on NEST simulation library

Name of DICs/software catalogue/or HBP github project entries.

DIC entry ‘ALGO STDP-structural’ generated under task 4.2.2

Description of data:

Species, sex, age, number of specimen/subjects.

Mostly more or less adult (around P16) rats. For details please refer to the cited publications from which the data has been obtained.

Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

Light (confocal and two-photon) and electron microscopy of dendritic spines in somatosensory cortex of the rat.

Completeness of data/algorithms/models:

Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The final publication of the model will include more analysis and discussion of plasticity dynamics than presented here, both for feed-forward and recurrent networks.

Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

Our plasticity model is well suited to be integrated in all kinds of network simulations of cortical circuits. For instance it will be interesting to study what it does in the BBP circuit simulations.

Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.

The model by Fauth, Wörgötter & Tetzlaff (PloS Comp Biol, 2015a,b) concerns an isolated synaptic connection between a single pair of neurons only. In networks, as we study here, there is cooperation and competition of synaptic contacts, which may lead to the formation and maintenance of structure in the network. Zheng, Dimitrakakis & Triesch (PloS Comp Biol, 2013) also model structural plasticity, but in an abstract network model of binary neurons with biologically implausible time scales.

**Outline state of validation work.**

The statistics and dynamics of the model system are consistent with experimental observations of dendritic spine turnover and synaptic contact numbers.

Data Quality and Value:**Verification of data quality.****Your subjective analysis of the value of the data/algorithms for the users.**

At this stage, the model represents a hypothesis how synaptic contact plasticity could take place, but other explanations are possible as well. Targeted experiments could help to increase confidence in this model, but these are out of scope of the present project.

Data/algorithm/model usage to date:

Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

Just EPFL-LCN has access to the model at the moment. Through integration in NEST, which is planned at the time of acceptance of the paper, the model will eventually become available to all users of the simulation platform of the HBP.

Are the data/algorithms/models considered final?

The model is final as far as the authors are concerned. The review process for publication may bring up additional concerns of the referees, but none are expected at this stage.

Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

Publication in preparation (Network simulations with structural plasticity are performed using NEST. The state of the plastic synapses is saved every 5 minutes. Analysis of the synapse states is performed on this dataset. The results are compared to experimental data on ongoing plasticity in adult rats. Simulations of the plasticity model with some manipulations of the parameters are also performed, to highlight the importance of certain components of the model).

2.4.1.4 Model Results

All quoted text and the figures reported here are excerpts from a draft article by Moritz Deger & Wulfram Gerstner, which will be submitted for publication soon.

“We have built a model of the collective plasticity of the synapses from presynaptic neurons onto a single postsynaptic neuron (Fig. 29A). The model (1) is a local, spike-timing dependent plasticity rule that we imagine to be realized by the biophysics of dendritic spines. The plasticity rule Eq. (1) maintains a certain level of spike timing correlations, as well as of postsynaptic firing rate.”

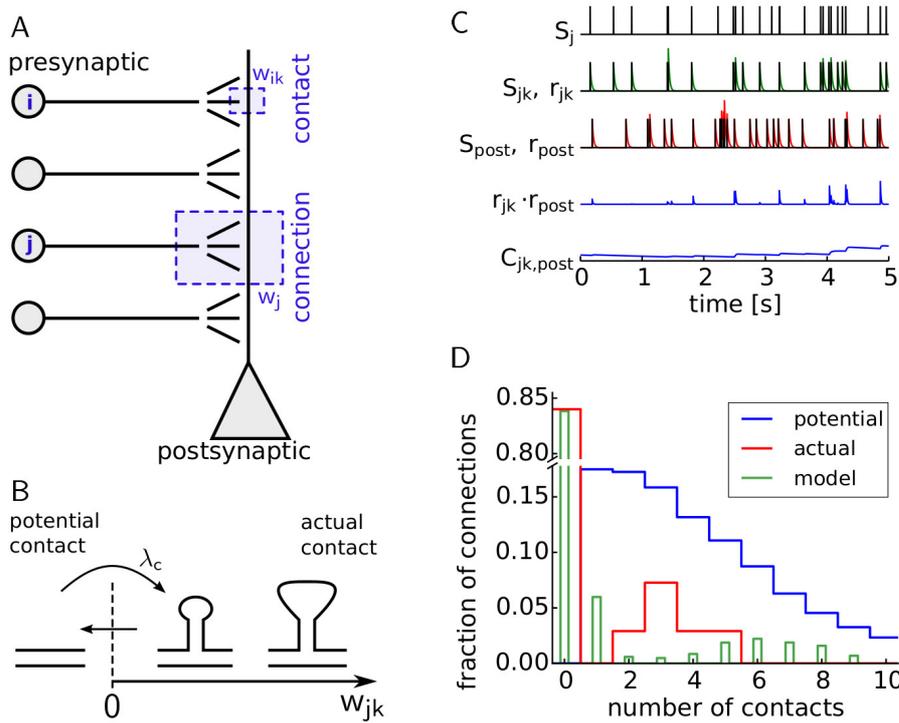


Figure 29: Model overview

A: Postsynaptic neuron (post) receiving multiple synaptic connections from several excitatory neurons (pre). Each connection consists of multiple synaptic contacts. The synaptic efficacy (weight) of contact k in connection j is denoted as w_{jk} . The total weight of this synapse is w_j , which is the sum of all its contact weights w_{jk} . B: Individual contact weights w_{jk} take continuous, positive values, and change in time according to spike-timing dependent plasticity. A small weight w_{jk} corresponds to a dendritic spine with a small volume, or a thin spine, and a large weight w_{jk} corresponds to a large, or mushroom-shaped dendritic spine. If w_{jk} is greater than 0 we call the contact an actual contact. In contrast, if at any time w_{jk} hits 0, the contact is pruned, its weight is kept fixed at $w_{jk}=0$, and its dynamics cease. Contacts with $w_{jk}=0$ are called potential (but inactive) contacts. They may occasionally form an actual contact (by being set to a positive w_{jk}) at random times, with a rate λ_c . C: Components of the plasticity model (top to bottom): presynaptic spike train S_j ; transmitted spike train S_{jk} at the contact (random synaptic failures occur), and its filtered trace r_{jk} ; postsynaptic spike train S_{post} and its filtered trace r_{post} ; product term $r_{jk} \cdot r_{post}$ composed of pre- and postsynaptic traces; slow low-pass filtered trace $C_{jk,post}$ of the pre- post product term $r_{jk} \cdot r_{post}$. D: Reference distribution of the number of actual (red) [Feldmeyer1999] and potential (blue) synaptic contacts of pairs of neurons in the adult somatosensory cortex [Fares2009] (recurrent synapses of layer 4 pyramidal neurons), truncated to $n \leq 10$ and renormalized. The steady state distribution generated by the model is shown in green (data pooled over 100 days of simulation).

“As a result of on-going STDP, synaptic weights move up and down. As soon as a contact weight $w_{jk}(t)$ hits zero, we fix it at zero and its dynamics cease. This contact is now called a potential, but inactive contact. However, with a rate of $\lambda_c=0.028/\text{day}$, each potential contact ($w_{jk}=0$) may be randomly transformed into an active contact again. In such an event, called creation, its weight w_{jk} is set to a low, non-zero value w_c , and its dynamics again follow (1) (see Fig. 29B).”

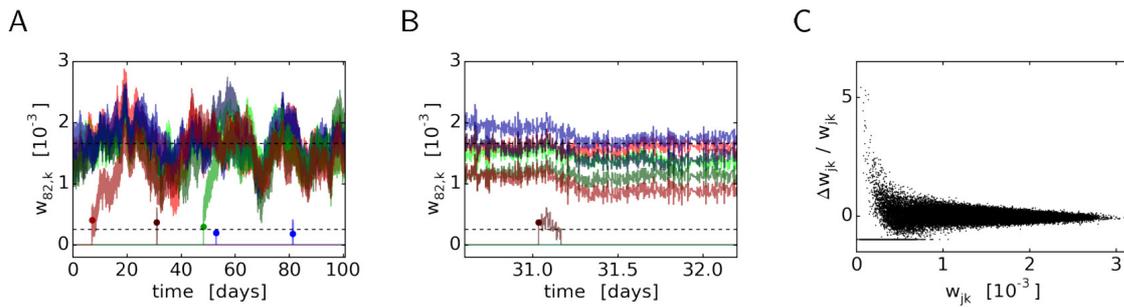


Figure 30: Dynamics of synaptic contacts in the steady state.

A: Example synaptic connection number $j=82$, different colors correspond to contact weights w_{jk} over time. New contacts (filled circles) are created with a weight given by the lower dashed line, but may evolve until the next recorded time step when they are first observed. Long-term stable contacts fluctuate close to the upper dashed line, which is the fixed point of w predicted by the theory (see Online Methods). **B:** Zoom into A at the time of creation and pruning of a persistent and of a transient synaptic contact. **C:** Relative changes of synaptic contact efficacy Δw_{jk} in one day, each dot corresponds to the change of one contact in one day, data pooled over all contacts and 100 days of simulation.

“Following an initial transient at the beginning of the simulation, the synaptic contacts fluctuate around a steady state. Sample trajectories of all contacts from a single presynaptic neuron to the postsynaptic cell are shown in Fig. 30A-B. Several contacts remain over the course of 100 days of simulation, occasionally new contacts are formed, which mature or are quickly removed (see Fig. 30B). Within one day, the relative change of the contact weight ranges up to 500% (see Fig. 2C) but fluctuations decrease with increasing contact weight. This is consistent with long-term time-lapse imaging data of spine volume in vitro [Yasumatsu2008].”

“The firing rate of the postsynaptic neuron, as well as the total synaptic weight and the average number of contacts are tightly controlled by the plasticity rule (1) (data not shown). The resulting distribution of synaptic contact numbers per connection (Fig. 29D) is bimodal, as found experimentally [Feldmeyer1999]. The turnover ratio of synaptic contacts in the model is $0.15 \pm 0.02/\text{day}$ (mean \pm SEM), which is consistent with the values found experimentally in somatosensory cortex [Holtmaat2009a]. “

2.4.1.5 Provenance

Not applicable. Data is from other, previously published articles, prior to our outside HBP.

2.5 Task 4.2.3 Structures of Spiking Learning Algorithms

2.5.1 A. Gruning (SURREY)

2.5.1.1 Introduction

Note All quoted text and the figures reported here are excerpts from [@gardner:neco:14] (as [1]) and from submitted print [@gardner:arxiv:16] (as [2]).

This task concentrated on the lack of efficient supervised learning algorithms for spiking neural network balancing biologically plausible plasticity mechanism as research in WP4.1 and WP4.2 and technical applicability as required for the neuromorphic and neuronformatic platforms SP6 and SP9. In particular this task has addressed the questions of general-purpose learning algorithms in [1], and that of the relation between different similar learning algorithms and their theoretical embedding in [2].

In particular, the research conducted on novel learning algorithms includes the following:



"Information encoding in the nervous system is supported through the precise spike timings of neurons; however, an understanding of the underlying processes by which such representations are formed in the first place remains an open question. Here we examine how multilayered networks of spiking neurons can learn to encode for input patterns using a fully temporal coding scheme. To this end, we introduce a new supervised learning rule, MultilayerSpiker, that can train spiking networks containing hidden layer neurons to perform transformations between spatiotemporal input and output spike patterns. The performance of the proposed learning rule is demonstrated in terms of the number of pattern mappings it can learn, the complexity of network structures it can be used on, and its classification accuracy when using multispikes-based encodings. In particular, the learning rule displays robustness against input noise and can generalize well on an example data set. Our approach contributes to both a systematic understanding of how computations might take place in the nervous system and a learning rule that displays strong technical capability." [from 1]

"In summary, we find the learning rule can encode for a large number of input spike patterns, being a substantial improvement over existing learning rules in this respect, and it provides increased accuracy when classifying inputs by the timings of multiple rather than single output spikes. Furthermore, we also explore the performance of the learning rule as applied to networks containing large numbers of output layer neurons, representing a unique contribution in the area of spike-based learning rules for multilayer networks." [from 1]

2.5.1.2 Model Description MultilayerSpiker

"[W]e propose a new supervised learning rule for multilayer spiking neural networks, termed MultilayerSpiker. Our rule extends the single-layer learning rule of Pfister, Toyozumi, Barber, and Gerstner (2006) to multiple layers by combining a maximum likelihood approach with error backpropagation. We demonstrate the efficacy of the proposed learning rule on several spike pattern transformation tasks: in terms of the accuracy of input pattern classifications based on multispikes codes and the time taken to converge in learning." [from 1]

"[We] propose a supervised learning rule, termed MultilayerSpiker, for training multilayer spiking neural networks to perform transformations between spatiotemporal input-output spike patterns. In the rule's formulation, we first consider a suitable likelihood function for generating desired output spike patterns, on which stochastic gradient ascent can be taken. The technique of backpropagation, as is traditionally used for rate-coded networks, is subsequently applied in finding hidden layer weight updates. In this way, our technique can be viewed as a generalization of the single-layer learning rule by Pfister et al. (2006) to multiple layers. Our multilayer learning rule differs from those proposed by Brea et al. (2013) and Rezende and Gerstner (2014), which have instead taken gradient descent on the KL divergence in a supervised and reinforcement setting, respectively. The novelty of our letter comes from the application of backpropagation, and its indicated high performance when encoding for a large number of input spike patterns as multiple and precisely timed output spikes." [from 1]

"The performance of the MultilayerSpiker learning rule was tested through simulations of multilayer networks trained to perform temporally precise spike pattern transformations. In our analysis, we considered networks containing either single- or multiple-output neurons. Example tasks for single-output networks include measuring the resilience of the network to input noise during learning, the solution of the XOR computation, a comparison between specific network structures, the memory capacity of the network, and its ability to generalize to new patterns on a synthetic data set. For multiple-output networks, the performance of the learning rule was tested for networks tasked with performing fully spatiotemporal spike pattern transformations." [from 1]



2.5.1.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles: ~ Task 423 "Structures of Spiking Learning Algorithms". PI: Andre Gruening. Executing Scientists: Brian Gardner (PhD student, postdoc), Ioana Sporea (postdoc, left), Eric Nichols (acceding postdoc).

- Data, algorithms, tools and methodologies storage location(s):

Algorithms and simulations are in GIT repository <https://feps-web.eps.surrey.ac.uk/gitlab/HBP>

- Description of algorithms/models/principles:

The algorithms are written in the Matlab/Octave language or C/C++.

- Name of DICs/software catalogue/or HBP github project entries.

see above

- Description of data:

n/a

Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities: ~ simulation of small and medium-sized neuronal networks.

- Completeness of data/algorithms/models:

codes are as released, and constantly being worked upon for the next stage of the HBP, especially with respect to implementation in the neuromorphic platforms. Contacts to the experts for the Spinnaker platform have been made and the modalities of model transfer are discussed in CDP5.

Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

~ The source codes provided are stand-alone in-principle implementations. Due to delay with the platforms and the evolving nature of plasticity implementation in them, codes are anticipated to run on eg Spinnaker during work on CDP5.

Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

~ Implementation on the neuromorphic platforms is in progress.

Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it:

~ There are various groups in this area designing learning algorithms for spiking neural networks with recent publications. The learning algorithms as developed by Westkott, Albers and Pawelzik <http://arxiv.org/abs/1407.6525> in particular follows a similar type of approach as ours. Other publications concentrate mainly on learning at single synapses / neurons and give greater emphasis to the biological plausibility, however at the cost of a loss in technical performance.

Outline state of validation work.

n/a

- Data Quality and Value:

At the current stage, the model represents a hypothesis on how



goal-oriented learning could be implemented on top of neural hardware. Its value consist in the demonstration that backpropagation can be implemented utilising spiking codes and neurons at a *technically* acceptable level (order of magnitude better performance than preexisting approaches) which opens up the road to technical use in the neuromorphic platforms.

- Data/algorithm/model usage to date:

The model/paper have just been published, hence too early to tell. At presentation to other research groups during the M18-M30 period, the suggested models have been well received.

- Who has used the data/algorithms/models, for what?

currently SP4 internal use.

It will be use for the "Learning Workbench" in Co-Design-Project 5 together with user in SP6/SP9.

- Are the data/algorithms/models considered final?

models are under constant improved to improve performance and biological plausibility.

2.5.1.4 Model Results

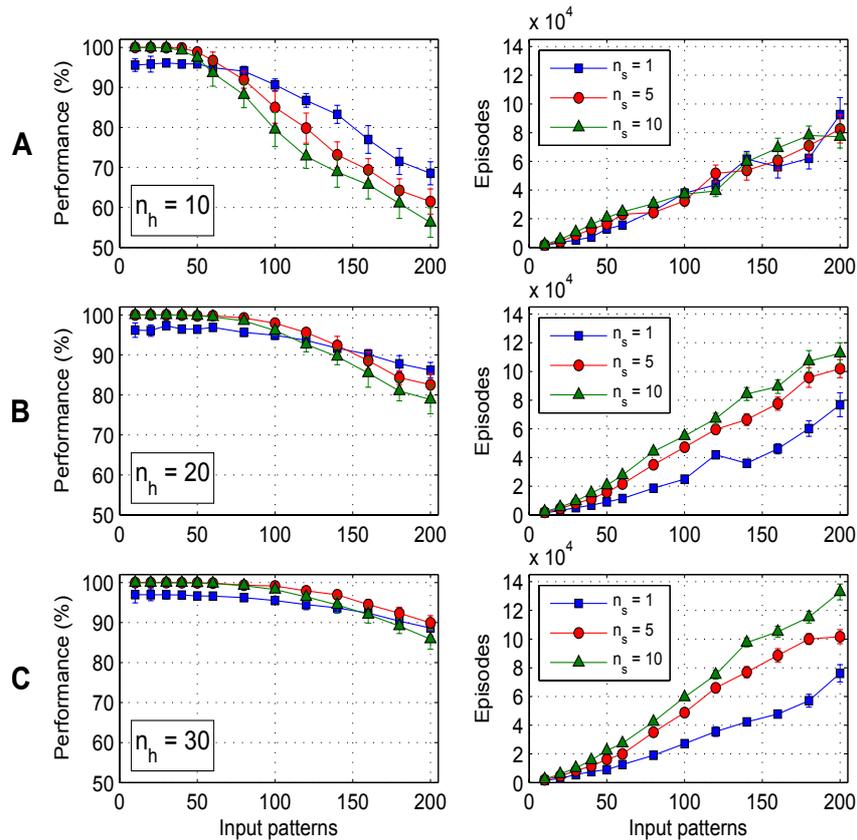
"[...], we have presented a new and technically efficient learning rule for training multilayer spiking neural networks, which has demonstrated a high performance level on several benchmark tests. The learning rule is capable of learning fully spatiotemporal input and output spike pattern transformations and can perform input classifications to a high level of accuracy using multiple output spikes. In our analysis we used the escape noise neuron model defined in Gerstner and Kistler (2002), which has been shown to closely approximate the variable firing activity of neurons in vivo (Jolivet, Rauch, Luscher, & Gerstner, 2006). Our choice of neuron model was primarily motivated by its general applicability to a wide range of learning paradigms, including supervised (Pfister et al., 2006; Brea et al., 2013) and reinforcement (Florian, 2007; Fremaux, Sprekeler, & Gerstner, 2010) learning. A key advantage of implementing escape noise neurons comes from being able to determine the likelihood of generating a specific output spike pattern (Pfister et al., 2006), which can then form the basis of a suitable objective function. Here we took the approach of maximizing the log likelihood of generating a desired output spike pattern in a multilayer network through a combination of gradient ascent and backpropagation, that is, an extension of the single-layer learning rule proposed by Pfister et al. (2006) to multilayer networks. In terms of the output layer, weight updates result from a product of locally available pre- and postsynaptic activity terms that bears a resemblance to Hebbian-like learning: the presynaptic term originates from filtered hidden neuron spike trains in the form of PSPs, and the postsynaptic term an output error signal that controls the direction and magnitude of weight changes [...]. Hidden layer weight updates, however, appear as a three-factor rule: PSPs due to input spikes are combined with hidden spike trains, to then be modulated by backpropagated error signals to allow hidden weight changes [...]. In training multilayer networks to map between spike patterns, it proved necessary to represent input patterns with sufficiently rich spiking activity at each input neuron; sparse representations otherwise led to decreased performance. This requirement is apparent from an examination of the hidden layer weight update rule, which has an explicit dependence on hidden neuron spike trains: a lack of input-driven hidden layer activity prevented weight updates from taking place, thereby resulting in diminished learning.

Previous multilayer learning rules (Bohte et al., 2002; Sporea & Gruning, 2013) have faced a similar challenge in effectively presenting input patterns to the network, but instead took the approach of introducing multiple synaptic connections with varying conduction delays between neurons of neighboring layers. We were motivated to introduce synaptic



scaling to the network to maintain an optimal range of hidden firing rates (van Rossum et al., 2000), a process that has been observed in biological networks (Turrigiano, Leslie, Desai, Rutherford, & Nelson, 1998). Aside from stabilizing the firing rate, the introduction of synaptic scaling also has side benefits, such as decreasing the network sensitivity to initial synaptic weight values (Sporea & Gruning, 2013), a critical issue that was identified in Bohte et al. (2002). An important contribution of our letter is the large number of accurate pattern encodings that can be performed by MultilayerSpiker. In comparison with multilayer ReSuMe (Sporea & Gruning, 2013) for a network containing 100 input neurons, a variable number of hidden neurons and a single output neuron tasked with mapping between arbitrary input-output pattern pairs, MultilayerSpiker was capable of at least 10x as many pattern classifications at a 90% performance level but requiring less than 1/10th the number of hidden neurons (see table 1 in appendix C, and table 7 in Sporea & Gruning, 2013). In addition to this, MultilayerSpiker scaled well with both the input and hidden layer sizes and performed classifications with higher accuracy when using multispikes based encodings. It is worth noting that most standard ANNs—for example, those containing perceptron units—are fundamentally incapable of performing such temporally based pattern mappings and fall behind spiking networks in terms of their computational power (Maass, 1997).

We believe our encoding method better takes advantage of spike-timing than many alternative methods (Gütig & Sompolinsky, 2006; Florian, 2012; Mohemmed et al., 2012). For example, the Tempotron (Gütig & Sompolinsky, 2006) is limited to binary classifications using a spike/no-spike coding scheme, and the experiments run for the Chronotron (Florian, 2012) and SPAN (Mohemmed et al., 2012) required precisely matched actual and target output spikes, which would invariably be detrimental to the network performance on generalization tasks given that actual output spikes would fluctuate about their respective target timings in response to input noise. As in most existing learning rules for spiking networks, we have assumed the presence of a supervisory signal, which allows a continuous comparison between actual and target output spike patterns during learning. Biologically, however, the source of such a signal remains unclear. Knudsen (1994) has posited that a separate, external network might exist that is capable of providing continuous feedback during learning, hence acting as a form of activity template that is to be mimicked elsewhere in the brain. Such a mechanism has been offered as an explanation for functional plasticity changes in neurons encoding for auditory stimuli in the barn owl (Knudsen, 2002). Finally, reinforcement learning might offer a plausible alternative, especially in light of the promising evidence that the firing activity of dopaminergic neurons can encode a form of reward-prediction error signal (Schultz, Dayan, & Montague, 1997; Schultz, 2000). In our previous work (Gardner & Gruning, 2013), we have shown how reinforcement learning can be applied in training single-layer networks to perform temporally precise input-output spike pattern transformations." [1]



The dependence of the network performance on the number of input patterns, the number of hidden neurons n_h , and the number of target output spikes n_s . In all cases, the network contained one output neuron. In this experiment, input patterns p were equally assigned between $c = 10$ classes. (Left) The performance as a function of the number of input patterns, for $n_h = 10$ (A), $n_h = 20$ (B), and $n_h = 30$ (C) hidden neurons. In each panel, different curves correspond to the number of target output spikes identifying each class of input. (Right) The number of episodes to convergence in learning. Results were averaged over 20 independent runs. at an early stage before being. Fig.8 from 1

2.5.1.5 Provenance

n/a

2.6 T4.3.1 Models of perception-action

2.6.1 Cognitive model of whole cortex _G. Deco (UPF)

2.6.1.1 Introduction

Our purpose is to develop a dynamical model of the whole cortex activity to reproduce activity patterns, or functional connectivity (FC), 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 differs from the structural connectivity (SC), namely the density of white-matter connections between distant cortical areas, which can be estimated by diffusion tensor imaging (DTI). Rather, EC measures the corresponding strengths that depend in the biology on neurotransmitter types, receptor concentration, local excitability, etc. Therefore, the aim of our project is to infer the macroscopic EC for the whole cortex from observed activity using fMRI, while incorporating DTI information for the connectivity skeleton. In practice, this means about 2000-3000

connections for 70-100 cortical regions and the model is tuned to reproduce empirical FC defined as the covariances of BOLD time series. The results will be used to constrain more detailed models (e.g., populations of spiking neurons) that aim to exhibit given correlated spatio-temporal patterns of activity in a specific neural system or for the whole cortex (SP6). The estimated EC values are biomarkers to be incorporated in the atlas (SP5), as they describe the dynamic properties of the corresponding neural populations. We also plan to characterize a “normal” profile for healthy subjects in order to examine abnormalities for individuals, as a tool for diagnostics (SP12).

2.6.1.2 Model Description

Figure 31 illustrates the global aspects of the whole-cortex dynamical model. DTI and fMRI information is combined to estimate the strengths of intracortical connections. The key here is to use the temporal information in FC to estimate the direction of connectivity at the scale of the whole cortex.

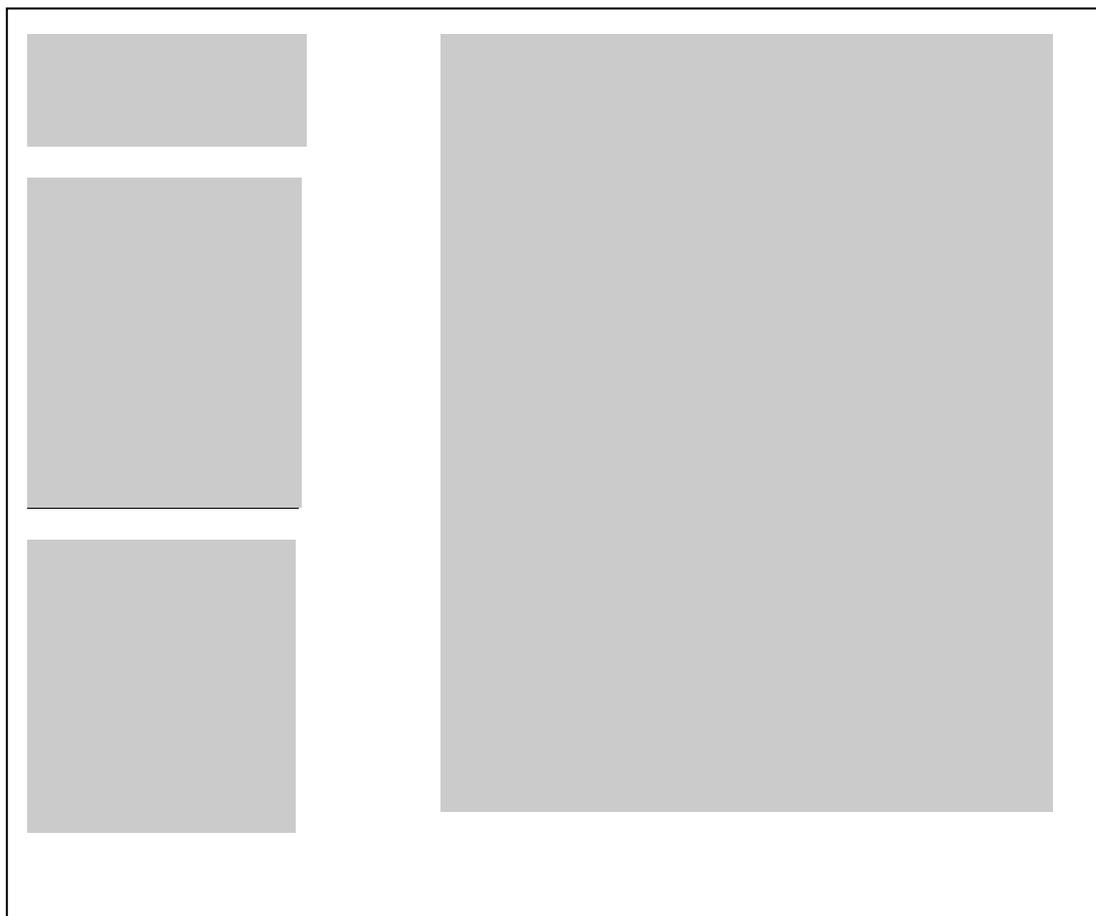


Figure 31: Schematic diagram of the data flow in our model-based approach to interpret cortical activity.

DTI information (bottom left) and a parcellation in 68 cortical areas (middle left) is used to build the skeleton of the network model. The strengths of the network are then tuned such that the model (right) fits the empirical FC (top left).

The network model in Fig.32 (left) is a nonlinear version of a multivariate Ornstein-Uhlenbeck process. As an illustration, the minimum model describes each population with a synaptic variable x_i , which exponentially decays with time constant τ_x and receives a current U_i that lumps excitation/inhibition from itself and other populations:

$\frac{dx_i}{dt} = \frac{-x_i}{\tau_x} + \Phi(u_i) + \sigma_i dB_i$ where Φ is the activation function that applies to the current
 $u_i = \sum_{j \neq i} C_{ij} x_j + e$, dB_i is unitary white noise and e the homogeneous external input. The
 intracortical connectivity (EC) by the matrix C . FC is the second-order moments of the
 variables x_i , namely $Q_{ij}^\tau = \langle (x_i^t - \bar{x}_i)(x_j^{t+\tau} - \bar{x}_j) \rangle$, where τ is the time shift and 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 C and Q^0 are
 related via a Lyapunov equation: $JQ^0 + Q^0 J^T + \Sigma = 0$, where the Jacobian of the
 dynamical system is $J_{ij} = \frac{-1}{\tau_x} + C_{ij} \Phi'(U_i)$, and Σ is the diagonal noise matrix such that
 $\Sigma_{ii} = \sigma_i^2$. Moreover, the time shifted covariance is given by $Q^\tau = Q^0 \expm(J^T \tau)$. The code
 is a Lyapunov optimization algorithm (Fig. 32 right) to infer C and Σ from $Q^{0/\tau}$ based on
 the previous consistency equations: the Jacobian update is
 $\Delta J = (Q^0)^{-1} [\Delta Q^0 + \Delta Q^\tau \expm(-J^\tau)]$ for the model FC error corresponding to the difference
 between the empirical and model FC matrices $\Delta Q^0 = \hat{Q}^0 - Q^0$ and $\Delta Q^\tau = \hat{Q}^\tau - Q^\tau$. Finally,
 the update of EC is $\Delta C_{ij} \propto \Delta J_{ij} / \Phi'(U_i)$ for existing connections (cf DTI). This also allows
 for the use of constraints on the connectivity during the optimization, namely non-
 negativity. The input variances are tuned according to the heuristic optimization
 $\Delta \Sigma_{ii} \propto \Delta Q_{ii}^0$.



Figure 32:

Left: Example of network of interconnected populations (purple circles). EC corresponds to the recurrent
 connectivity in red, which is directed. Each node in the network receives individual noise (Σ) that is
 shaped by EC to generate FC (activity covariances between all pairs of node). Right: Algorithm that tunes
 the network parameters EC and Σ for the model to reproduce empirical FC.

Details for the linear model (Φ is the identity) can be found in a recently accepted
 publication (uploaded preprint PLoS_CB_dircon_R3.pdf). The algorithm has been checked
 on surrogate networks by generating activity from known underlying network parameters,
 then measuring the performance to retrieve that original connectivity. The focus is on
 directed connectivity, which requires the use of time-shifted covariances in addition to
 zero-shift covariances.

2.6.1.3 Model Data

T4.3.1 is responsible for generating algorithms/models/principles.

- Code stored on EITN owncloud (www.unic.cnrs-gif.fr/owncloud)



- Description of algorithms/models/principles:
 - 2 equivalent versions, in Matlab and Python (in ‘HBP_code_WP43’ folder).
 - Matlab version: the script ‘extract_data.m’ computes from BOLD time series the average empirical FC matrices, as well as SC from DTI; the script ‘optimization.m’ estimates the “best” network parameters EC and Σ that give the best fit for the empirical FC used as objectives in the optimization.
 - Python version: see ‘how to use’ document.
- Description of experimental data:
 - fMRI/BOLD time series recorded from 25 subjects aged from 19 to 40 years during rest (eyes closed); DTI data; all available online, see Schirner M, Rothmeier S, Jirsa V, McIntosh A, Ritter P. An automated pipeline for constructing personalised virtual brains from multimodal neuroimaging data. *Neuroimage* (2015) 117: 343-357; Parcellation of cortex in 68 areas.
 - Twenty-four right-handed young, healthy volunteers (15 females, 20-31 years old) participated in the fMRI recording for 2 conditions: a resting state and a natural viewing condition. In the resting state, participants fixated a red target with a diameter of 0.3 visual degrees on a black screen. In the natural viewing condition, subjects watched (and listened) to 30 minutes of the movie “The Good, the Bad and the Ugly”. Data published previously in: Hlinka J, Palus M, Vejmelka M, Mantini D, Corbetta M (2011) Functional connectivity in resting-state fMRI: is linear correlation sufficient? *Neuroimage* 54:2218-25
- Completeness of algorithms/models:
 - The method has been verified using fMRI and DTI data from three distinct laboratories (Prof P Ritter from la Charite in Berlin, Prof M Kringelbach in Oxford, Prof Kourtzi from Cambridge), and a currently developed extension focuses on interpreting task evoked activity / other condition in addition to resting state. In particular, we collaborate with Prof S Dehaene in SP3 for anaesthesia, cf T4.3.3.
 - The algorithms and models are final in the sense they can be used in the current version, but will be extended in the future.
- Model Quality and Value:
 - A limitation lies in the temporal resolution (~2s) and duration of fMRI time series during resting state, which raises the issue of stationarity of the network parameters to estimate.
 - So far, we have focused on 100 cortical areas (e.g., AAL90). Detailed parcellations with more than 100 areas imply a larger number of connectivity parameters and require further investigation for the optimization robustness. In particular, the amount of empirical BOLD data (recordings duration) for individual subjects is a crucial issue.
 - The nature of the nonlinearity for the local dynamics (activation function of each cortical area) requires further examination.
- Data/algorithm/model usage to date:
 - The algorithm will be passed and adapted to SP6/SP12 for applications in large-scale simulations of cortical network with realistic macroscopic connectivity and applications to individual profiling of subjects/patients.
- Publications with details of model and algorithm:

Gilson, Moreno-Bote, Ponce-Alvarez and Deco, Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome, PLoS Comput Biol (in press)

Deco, Ponce-Alvarez, Hagmann, Romani, Mantini and Corbetta. How Local Excitation-Inhibition Ratio Impacts the Whole Brain Dynamics. J Neurosci (2014) 34: 7886-7898

Deco and Kringelbach. Great Expectations: Using Whole-Brain Computational Connectomics for Understanding Neuropsychiatric Disorders. Neuron (2014) 84: 892-905

- References

Deco, Tononi, Boly and Kringelbach. Rethinking segregation and integration: contributions of whole-brain modelling. Nat Rev Neurosci (2015) 16: 430-439

Hlinka, Palus, Vejmelka, Mantini and Corbetta. Functional connectivity in resting-state fMRI: is linear correlation sufficient? Neuroimage (2011) 54:2218-25

Schirner, Rothmeier, Jirsa, McIntosh and Ritter. An automated pipeline for constructing personalised virtual brains from multimodal neuroimaging data. Neuroimage (2015) 117: 343-357

2.6.1.4 Model Results

From BOLD time series, we calculate two empirical FC matrices, one with spot covariances and one with a time shift of 4s; here the average over 25 subjects in Fig. 33A. The autocovariances are close to an exponential decay, with a time constant of 5.3s evaluated in a log plot in Fig. 3B. The distribution of time constants for all areas is narrow, as shown in Fig. 3C, which justifies the use of a single time constant here. This calibrates the model before the optimization. SC values obtained from DTI are thresholded to determine existing connections (in black in Fig. 33D), constraining the model to be tuned. The optimization gives the “best” EC in Fig. 33F, which corresponds to the minimum FC error (blue curve in Fig. 33E) between the model and empirical matrices. Fig. 33G illustrates the fit for all matrix elements (diagonal in cyan, existing connections in blue, absent connections in black) for the two FC matrices.

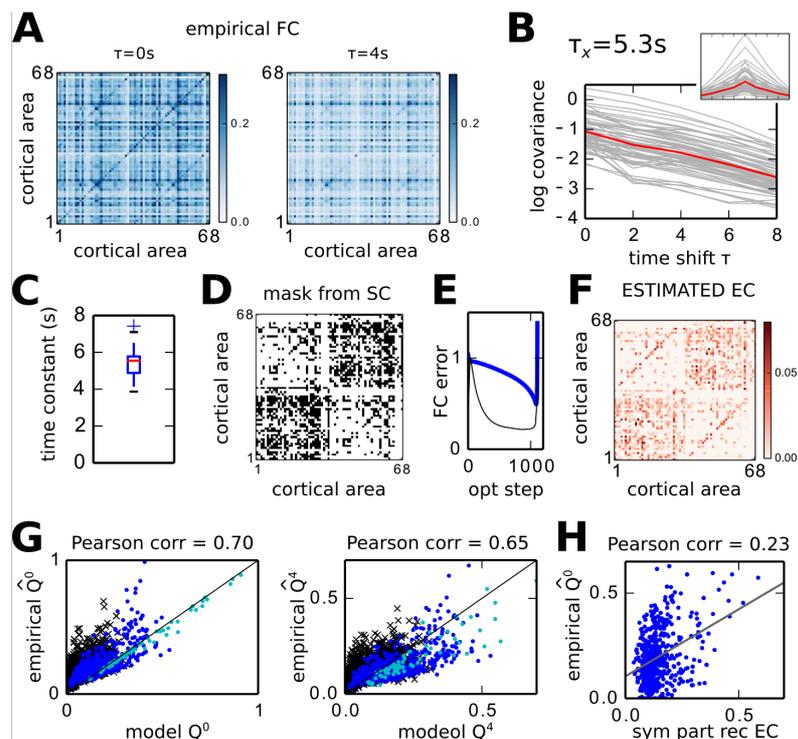


Figure 33: Tuned network model to reproduce empirical FC obtained from fMRI data. See text for details.

We present briefly the results in the last figure of our publication, in order to interpret EC. As shown in Fig. 34A, EC values differ from SC values, suggesting that the strengths of dynamic interactions are not only predicted by the density of neural fibres estimated by DTI. EC has information about individual connections, in particular representing directed connectivity, in contrast to SC. This corresponds to the input-output asymmetry of EC weights for each cortical area (red crosses in Fig. 34B). Interestingly, hubs are either receivers with strong inputs or feeders with strong outputs, but not both. Further interpretation of the results are ongoing, to understand the flow of information across the whole cortex, which is measured via the proxy of EC and Σ in Fig. 34C. These biomarkers can then serve to constrain the connectivity between cortical regions in more detailed models, and should be integrated in the human atlas.

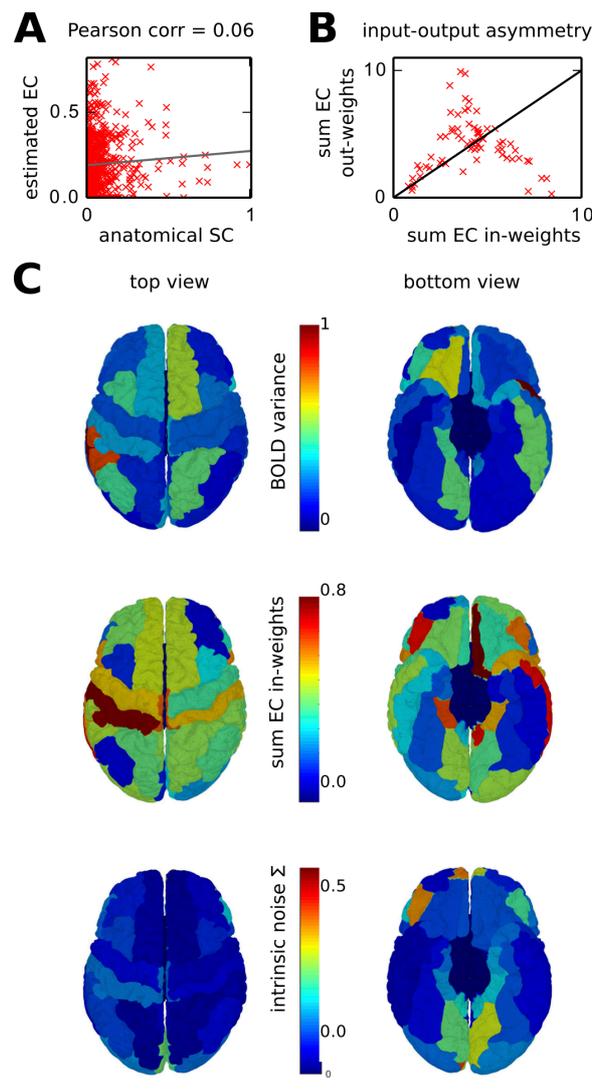


Figure 34: Interpretation of results from Fig. 33. See text for details.

Finally, we give unpublished results related to task condition, passive viewing and listening versus resting state. Fig. 35 shows that the average estimated EC for the 23 subjects is not dramatically affected in the visual and auditory systems. Rather, the change of inputs is more pronounced (right).

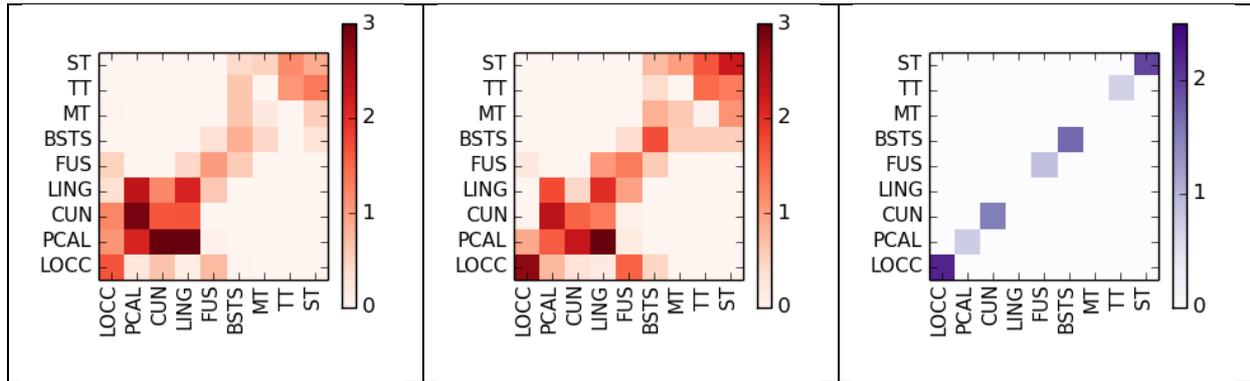


Figure 35: EC estimated for rest (left) and movie (middle). Significant changes in inputs (right) between movie and rest conditions.

Disentangling the contributions in changing FC from inputs and connectivity is at the core of the concept of effective connectivity. It gives quantitative estimates of how the activity/information flows in the network, uncovering the dynamic emergence of functional communities. Results in Fig. 35 based on data from Hlinka et al Neuroimage (2011) are a first step in that direction toward a more systematic approach.

What we aim to obtain then is a description of the average human profile of dynamical biomarkers (excitability, interaction strengths) in various conditions, such as rest, specific tasks involving sensory perception or task performance. We aim to uncover principles for the regulations the dynamical interactions at the cortex level that can be used to constrain large-scale models, for example, relating estimated excitability with neuromodulators in more detailed models (see Deco et al Neuron 2014, Nat Rev Neurosci 2015). We will also define mesoscopic measures for the goodness of fit for models that aim to reproduce cortical regions.

2.6.1.5 Provenance

T4.3.1

2.6.2 O. Faugeras (INRIA)

2.6.2.1 Introduction

Our work challenges and extends earlier seminal work. We consider the problem of describing mathematically the spontaneous activity of V1 by combining several important experimental observations including (1) the organization of the visual cortex into a spatially periodic network of hyper columns structured around pinwheels, (2) the difference between short-range and long-range intracortical connections, the first ones being rather isotropic and producing naturally doubly periodic patterns by Turing mechanisms, the second one being patchy, and (3) the fact that the Turing patterns spontaneously produced by the short-range connections and the network of pinwheels have similar periods. By analysing the PO maps, we are able to classify all possible singular points (the pinwheels) as having symmetries described by a small subset of the wallpaper groups. We then propose a description of the spontaneous activity of V1 using a classical voltage-based neural field model that features isotropic short-range connectivity modulated by non-isotropic long-range connectivity. A key observation is that, with only short-range connections and because the problem has full translational invariance in this case, a spontaneous doubly periodic pattern generates a 2-torus in a suitable functional space which persists as a flow-invariant manifold under small perturbations, for example when turning on the long-range connections. Through a complete analysis of the symmetries of the resulting neural field equation and motivated by a numerical investigation of the bi-furcation of their solutions, we conclude that the branches of solutions which are stable over an extended range of parameters are those that correspond



to patterns with an hexagonal (or nearly hexagonal) symmetry. The question of which patterns persist when turning on the long-range connections is answered by (1) analysing the remaining symmetries on the perturbed torus and (2) combining this information with the Poincaré-Hopf theorem. We have developed a numerical implementation of the theory that has allowed us to produce the predicted patterns of activities, the planforms. In particular we generalize the contoured and non-contoured planforms predicted by previous authors.

2.6.2.2 Model Description

NOT APPLICABLE

2.6.2.3 Model Data: Pinwheel V1 model

- Task(s)/group(s) responsible for generating algorithms/models/principles.
- P26, Olivier Faugeras
- Data, algorithms, tools and methodologies storage location(s) (and links?)
- <http://mathematical-neuroscience.springeropen.com/articles/10.1186/s13408-015-0023-8>
- Description of algorithms/models/principles:
 - Code is written in Python
 - Name of DICs/software catalogue/or HBP github project entries.

<https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=5f9067feb3f144ba9b34df7e24eb538d>

- Description of data:

Not applicable

- Completeness of data/algorithms/models:

Not applicable

- Outline state of validation work.

We have verified that the model can account for many known visual hallucinations

- Data Quality and Value:

The model we have developed can be used in conjunction with experimentalists in order to verify the validity of some of the predicted visual hallucinations corresponding to spontaneous V1 activity.

- Data/algorithm/model usage to date

So far the model has only been used internally by the Inria group

- Are the data/algorithms/models considered final?

The model is not final: we plan to refine it to account for more biological data.

2.6.2.4 Model Results

The figure below shows the following: In the left column, two examples of cortical activations, as predicted by our model, overlaid on the Preferred Orientation map. The pinwheels are shown in white. In the right column: interpretation of these activations in cortical coordinates, obtained by pooling the different activated orientations around each pinwheel. See the reference below for more information.

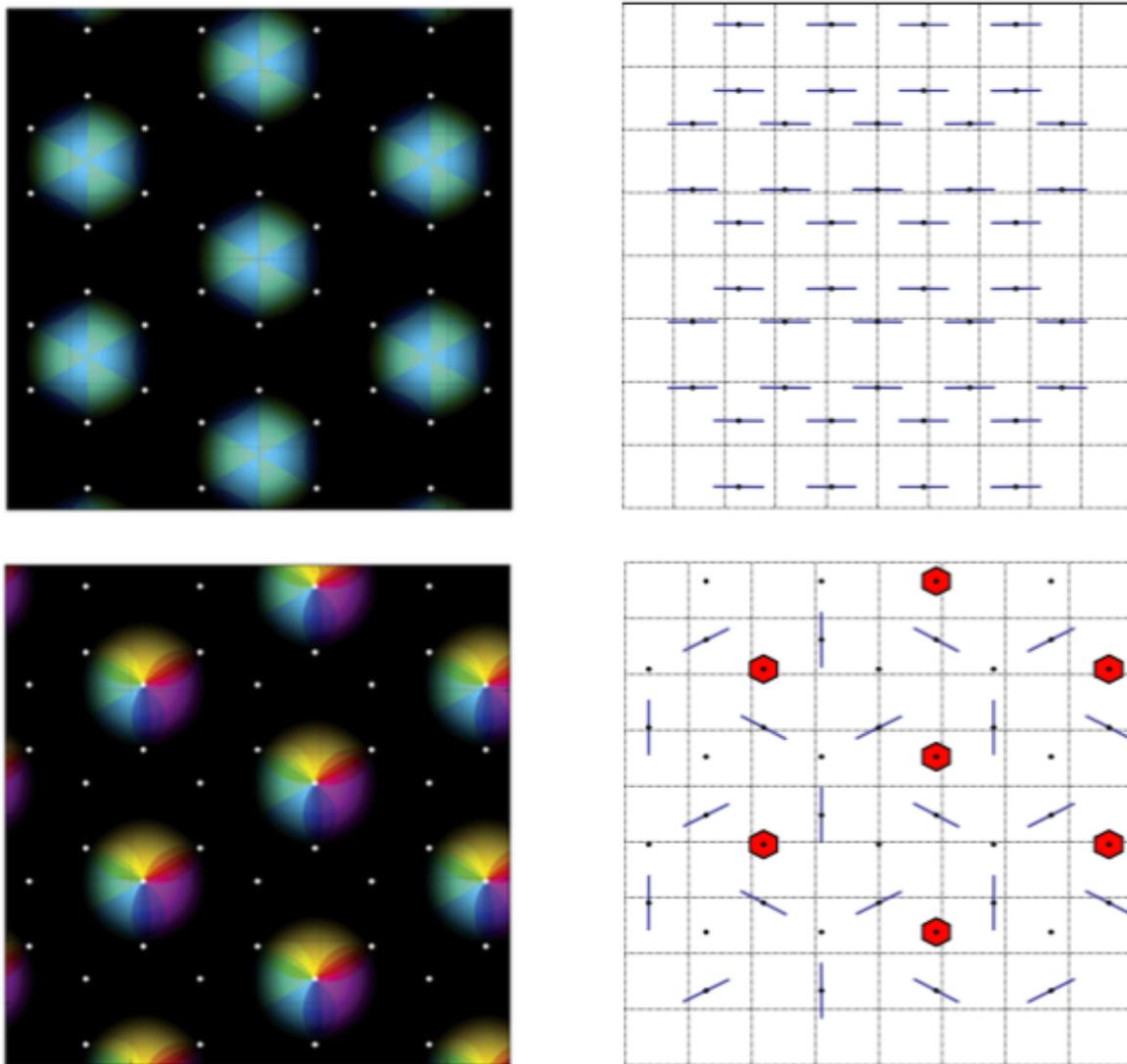


Figure 36:

Romain Veltz, Pascal Chossat, Olivier Faugeras. On the Effects on Cortical Spontaneous Activity of the Symmetries of the Network of Pinwheels in Visual Area V1. *The Journal of Mathematical Neuroscience (JMN)* 2015 5:11. Link: <http://mathematical-neuroscience.springeropen.com/articles/10.1186/s13408-015-0023-8>

2.6.2.5 Provenance

Not applicable

2.6.3 *F. Chersi & N. Burgess UCL (WP3.4 and WP4.3)*

2.6.3.1 Introduction

Spatial navigation, although one of the most common actions for humans and animals, is a complex task that involves the processing of a variety of sensory and proprioceptive stimuli (e.g. visual, vestibular and motor information), the storage and recall of memories about location and events, and the elaboration of plans.

There is now an unparalleled literature concerning the neural representations involved in spatial cognition and a vast array of experimental data relating behavior in spatial navigation tasks to manipulations of the environment or learning procedure and to the operation or impairment of different brain systems.

There are two main mechanisms utilized in spatial navigation. The first one relies on following a well-learned sequence of actions, each depending on the previous action or a sensory cue (“response strategy”), the second one by following a flexible internal representation of spatial layout (“place strategy”).

During the first phase of the HBP we have implemented a biologically inspired model of the two main circuits involved in spatial navigation, more precisely the hippocampus and the striatum.

We have also developed a simple simulator of the environment and the rat that can be connected to a control architecture that guides its behavior.

2.6.3.2 Model details

2.6.3.2.1 The visual system

The simulated rat has been endowed with a simple visual system that allows it to acquire two types of information about the environment: the color of the observed objects and their distance (from the observer). In the current implementation the visual field extends from -160 to +160 degrees and is subdivided into small regions (see Figure 37), each assigned to one neuron.

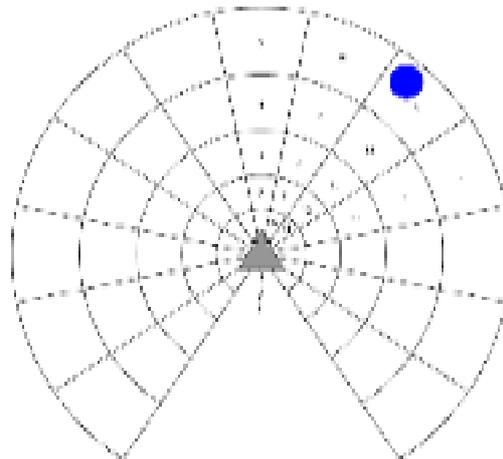


Figure 37: Schematic representation of the neural representation of the rat's view field.

Each region is associated to a specific neuron (numbered). When an object, e.g. the landmark, is spotted, neurons corresponding to the interested regions become active

The transformed sensory representation will be encoded as a single layer of neurons that are mostly silent, with few neurons encoding the presence of an object in the view field, firing at a high rate.

2.6.3.2.2 Neural architecture

The neural circuits implemented in this model utilize firing rate-based neurons with synaptic currents. The behavior of each neuron is described by the following set of equations:

$$\begin{cases} \tau_v \frac{dv}{dt} = -v + F(I_{syn}) \\ F(I_{syn}) = v_{max} \cdot \tanh^+[\gamma(I_{syn} - I_0)] \end{cases} \quad (1)$$

where I_{syn} is the total synaptic current, v is the firing rate, $\tau_v = 25$ ms the firing rate time constant, $v_{max} = 100$ spk/s is the maximum firing rate, $F()$ is the current-to-firing rate

transfer function, \tanh^+ is the positive part of the hyperbolic tangent, $\gamma = 1.56 \cdot 10^9$ is the slope of the (I-f) response function, $I_0 = 0.4$ nA is the threshold for the input current below which the neuron does not fire.

The hippocampal circuit

The hippocampus has here been implemented as single layer of firing rate-based neurons with no lateral connections. For sake of simplicity we assume that its neurons encode the position of the agent in an absolute reference frame. The typical response function of place cells is a Gaussian-like activity profile centered on the receptive field.

In the current model, we have implemented a mechanism that generates a new place cell whenever the distance to the closest field center is higher than a specific value.

When the rat reaches the goal location (i.e. the hidden platform) the connections between the place cells and the goal cell (located in the striatum/nucleus accumbens layer) are strengthened proportionally to their firing rate according to the following Hebbian-like equation;

$$\Delta w_{ij} = \eta \cdot v_i^{(HPC)} \cdot v_j^{(G)}$$

where w_{ij} is the connection weight between neuron i and neuron j , η is the learning rate, $v_i^{(HPC)}$ and $v_j^{(G)}$ are the firing rates of the i -th neuron in the hippocampus and of the j -th goal neuron, respectively.

The end result is the formation of a “goal surface” that has it highest value centered on the location in the environment where the object (i.e. the platform) is. This surface can be used to guide behavior. In particular, given the goal surface and the current coordinates of the rat, one can test adjacent positions in order to determine the gradient of the surface. If the surface is convex this method allows to iteratively find the path that leads to the maximum.

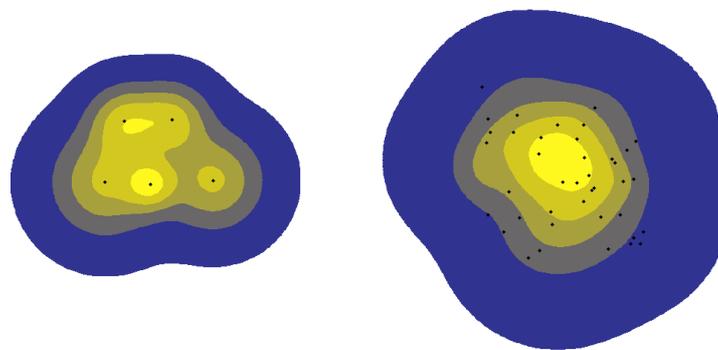


Figure 38: "Value function" that encodes the position of a specific object through the superposition of multiple receptive fields.

2.6.3.2.3 The striatal circuit

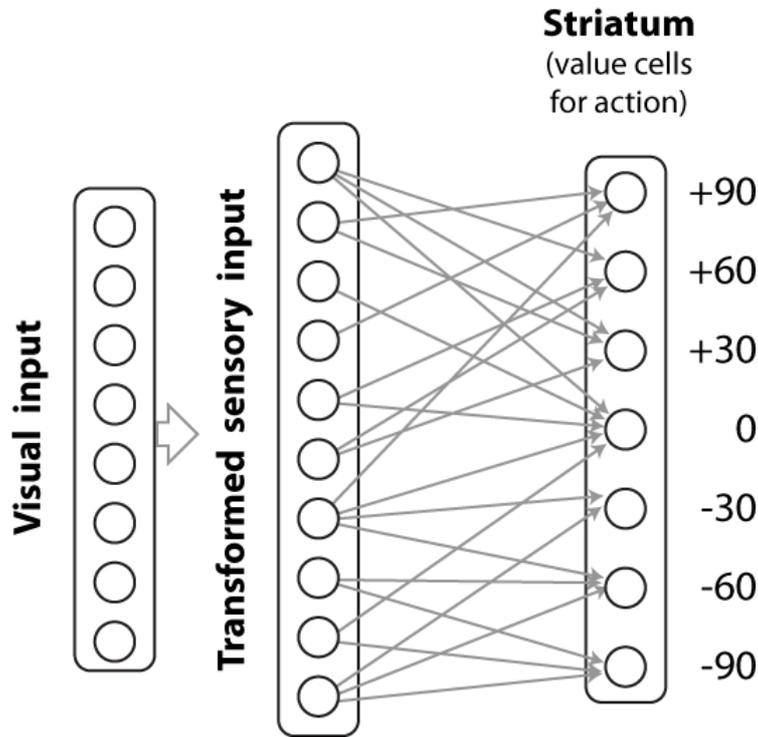


Figure 39: Striatal circuit employed in this architecture that assigns (a value of) an action to each sensory input.

The sensory input for the striatum is obtained by transforming the visual signals in the way described above. Note that the striatum does not receive information about the head direction, so the sensory vector is aligned to the heading direction and to a global reference point.

Neurons in the transformed sensory layer project to neurons in the dorsal striatum in an all-to-all manner. The latter neurons represent the values of the possible actions associated with any given sensory state playing here the role of “critics” for the state-action associations.

The sensory and striatal neurons can be thought of as representing state-action combinations for reinforcement learning.

In the current implementation the weights between the sensory and the striatum layer are updated by means of the Q-learning rule, as follows:

$$Q_a = v_a^{(STR)} = F \left[\sum_{i=1}^N v_i^{(Sens)} \cdot w_{i,a} \right]$$

where Q_a is the expected discounted return obtained by performing action a (in our case one of the 12 angles of rotation) in the current state, $v_a^{(STR)}$ is the firing rate of the striatum neuron corresponding to action a , F is the response function of the neurons, N is the total number of sensory neurons, $v_i^{(Sens)}$ is the firing rate of the active sensory neuron, and $w_{i,a}$ is the weight between the sensory neuron and the striatum neuron.

At every time step the rat can use the available information about the current and the past Q values and a possible reward to update its internal model by means of the following equations:

$$\Delta Q_{s-1,a-1} = \eta \cdot [R + \gamma \cdot \max_{a'} (Q_{s,a'}) - Q_{s-1,a}]$$



$$\Delta w_{i,a-l} = \Delta Q_{s-l,a-l} \cdot v_i^{(Sens)} \cdot \left(\sum_{j=1}^N v_j^{(Sens)} \right)^{-1}$$

where $Q_{s-l,a-l}$ is the Q value of action $a-l$ in state $s-l$, η is the learning rate, R is the reward, γ is the discount factor, and $\max_{a'} (Q_{s,a'})$ is the maximum Q value that can be reached from the current state s computed on all possible actions a' .

Note that the computation can only be done backwards in time, hence $Q_{s-l,a-l}$, because the agent does not know the transition function between states and has to remember the last state and action.

Additionally, Note that the striatum does not receive information about the head direction, so the sensory vector is aligned to the heading direction and not to a global reference direction.

2.6.3.3 Results

The above described architecture has been used to control a simulated rat in a variant of the Morris Water Maze described by Pearce et al. (1998). In this set up, in contrast to the original experiment, a landmark indicates the close-by location of a submerged platform.

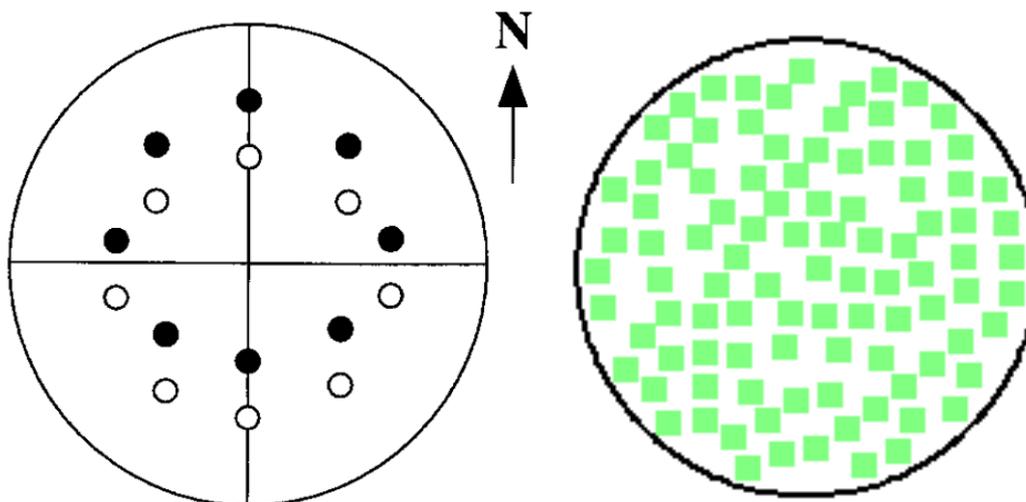


Figure 40: Left panel: Locations of the hidden platform (white circle) and the corresponding landmark (black circles). Right panel: Corresponding distribution of place cells across the water maze.

At the start of each trial, the rat is placed in a location chosen randomly among the potential locations of the platform, obviously excluding the location where the platform actually is.

Every 4th trial the platform and the landmark are randomly and conjunctively moved to one of the 8 locations shown in the left panel of Oure 39.

As the rat explores the maze, place cells specialize to cover the whole environment as shown in the right panel of Figure 39.

When the rat is placed for the first time into the maze it has no knowledge about the location of the submerged platform and about the relation between the (visible) landmark and the platform, thus it initially randomly explores the maze. As it learns more about how to find the location of the platform, its behavior shifts from random to goal directed, utilizing the hippocampus and the striatum to make reasoned decisions.

The movement of the rat is determined by the following set of equations:



$$\begin{cases} \frac{ds}{dt} = -s + v \\ \frac{d\theta}{dt} = -\theta + \omega \end{cases}$$

where s is the linear movement, v is the linear speed and is determined by the controller, θ is the rotation angle (relative to the direction of movement), ω is the angular speed and is determined by the controller.

The complete navigation controller is composed of several components (see Figure 40). At the lowest level there is the obstacle avoidance mechanism that ensures that the rat does not run into walls and obstacles. It has the highest priority and is active at every time step. It utilizes the distance information provided by the sensory areas to produce a motor reaction that distantiates the rat from the obstacle.

Besides this we have implemented a mechanism that every 250 ms triggers the decision procedure which selects a new movement direction. In particular, 5% of the times the decision is completely random. This ensures that the rat does not get trapped in cyclical movements. In the remaining 95% of the times, utilizing the mechanisms described above the hippocampus and the striatum each produce an output vector that contains the “values” (encoded as neuronal firing rates) of the 12 directions in which the rat can move.

The final decision is taken by the prefrontal cortex that determines which direction most likely leads most rapidly to the goal location. In case none of the values is above a minimum threshold, a new direction is chosen randomly among the available ones.

In order to render the simulations biologically more realistic we introduced a “neural computation time”, i.e. the time that a biological neural network would approximately take to execute the computations in the hippocampus, which in our case has been simplified and implemented algorithmically for sake of simplicity.

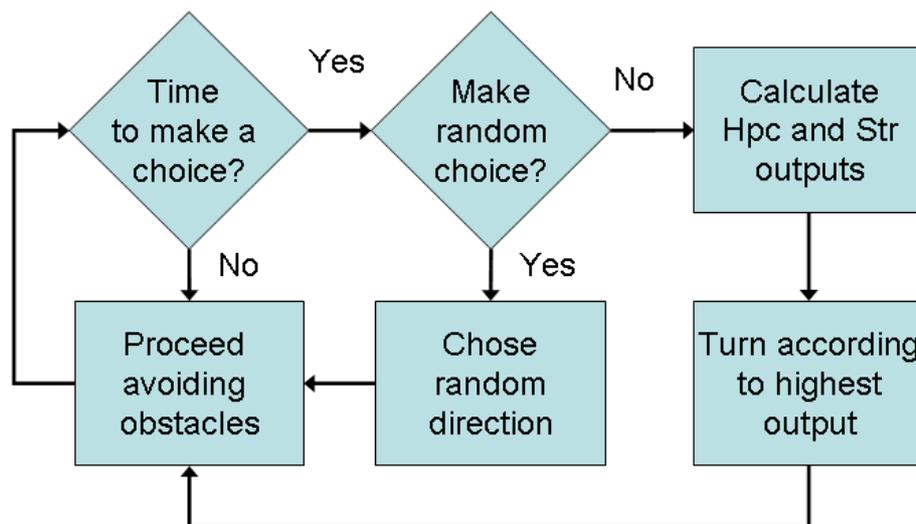


Figure 41: Decision scheme for the navigation task.

A low level controller has been implemented which assures that the rat avoids obstacles and walls. Every 250 ms the controller engages the direction decision mechanism. In 5% of the cases a random direction is chosen. In the other cases the responses of the hippocampal and the striatal circuit are compared and the response with the highest value is chosen.

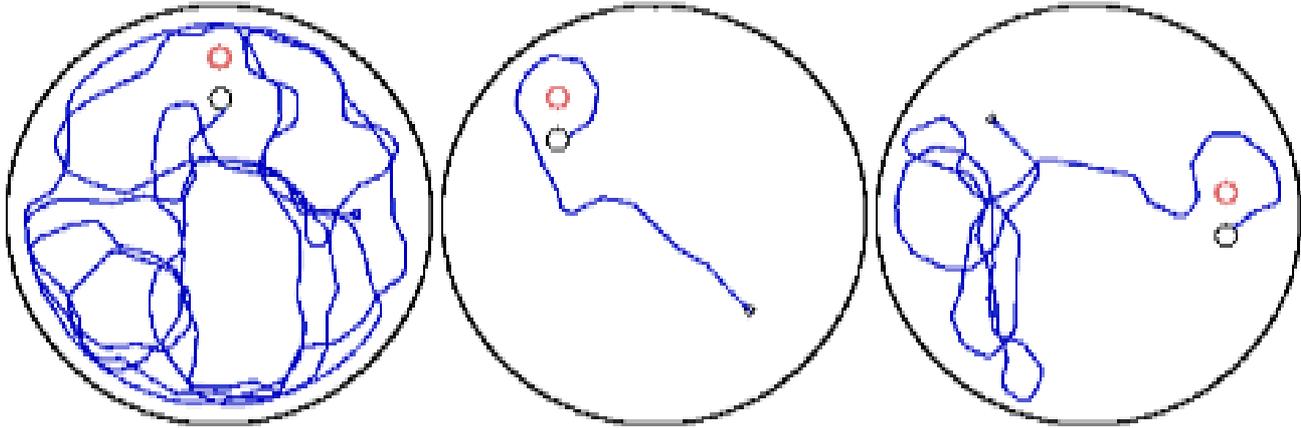


Figure 42: Different types of behaviors and trajectories.

Left panel: random exploration. Central panel: after several learning trials, when action is controlled predominantly by the striatum, the movement becomes straight in the direction of the hidden platform (the rat knows where to go). Even if the rat initially slightly misses the location it turns around and rapidly finds it. Right panel: at the beginning of the experiment the rat's behavior is strongly driven by the hippocampus. As a result the rat initially tends to return to the previous location of the platform even if the landmark and platform are moved. After repeated failures the rat engages in exploratory behavior and eventually reaches the platform.

The left panel in Figure 43 shows the average performance of 4 groups of 30 simulated rats over the course of 11 sessions, each composed of a set of 4 similar trials. Following the original experiment we have deactivated the hippocampus of 2 of the 4 four groups of animals. The corresponding results are indicated by the curves with the full dots. The curves with the full dots represent the performance of animals with intact hippocampus and striatum. The upper curve represents the performance on the first trial of each set, while the lower curve represents the performance on the fourth trial. As can be seen the performance on the first trial is much worse. This is because the hippocampus drives the choice mechanism to reach the location where the platform was previously situated (as opposed to the location where the landmark is). When the rat reaches that location and realizes that the platform is not present, it starts to unlearn the hippocampal goal coding in favor of a random search strategy. In a few trials the rat is able to learn the new platform location and the new goal representation. Note that the place cells do not remap, i.e. their place field remains stable, instead only their connections to the goal cells are modified.

Rats with no working hippocampus are able to solve the task and reach the hidden platform in a time that is shorter the control rats on their first trial because in those specific cases the hippocampus misleads the rat to return to the presumed location of the platform which in fact has been moved elsewhere (see trajectories in Figure 42).

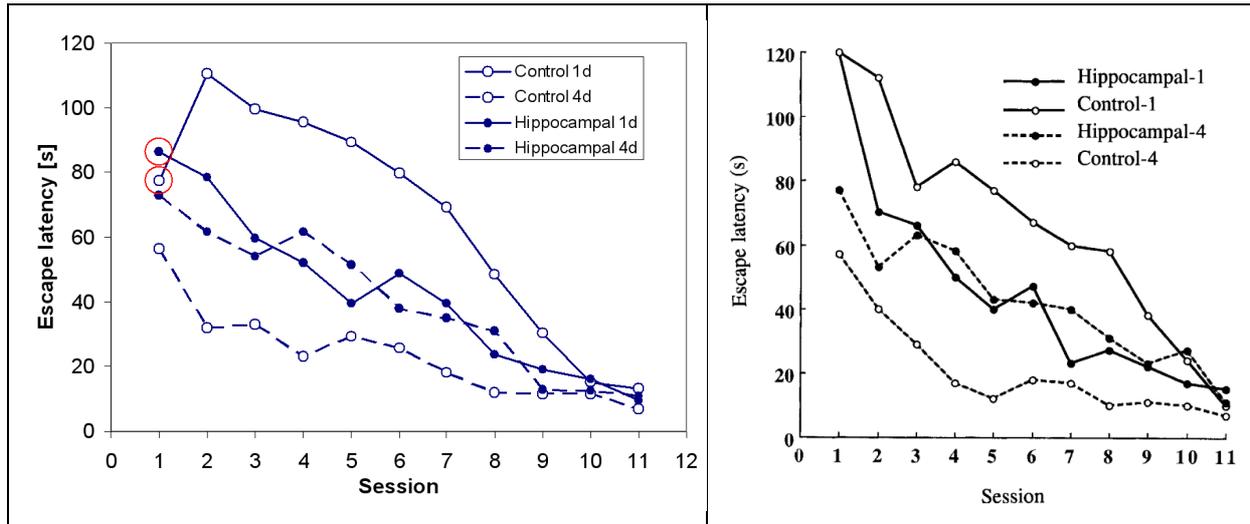


Figure 43: Left panel: average performance of the rats expressed as the time required to reach the escape platform. Right panel: performance of real rats as recorded by Pearce et al. (1998)

Figure 44 represents the percentage of choice type, whether based on a place strategy or on a response strategy, of all control rats in session 1, in session 6 (middle of the experiment) and in session 11 (end of experiment). As can be seen, as the rats spend more time in the maze solving the tasks they switch from a goal directed strategy to more stimulus-response one, indicating the formation of habits.

As a comparison, in the right panel of Figure 44 we report statistics on the behavior of rats lesioned either in the caudate nucleus or in the hippocampus as found in the paper mentioned above.

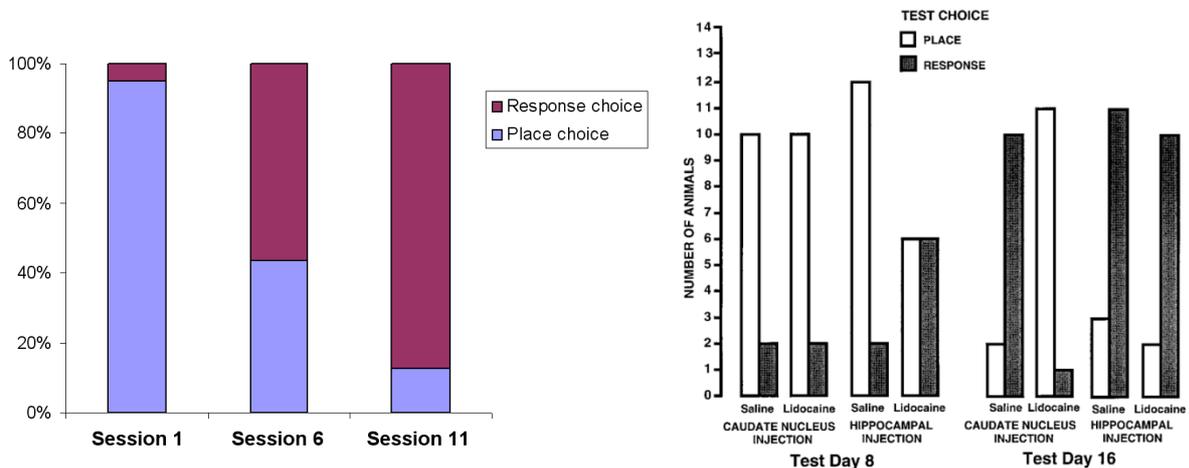


Figure 44: Behaviour of rats lesioned in caudate nucleus or hippocampus

Left panel: distribution of strategy choices (response or place strategy) during session 1, session 6 and session 11. The last two approximately correspond to Test Day 8 and Test Day 16 in the graph in the right panel reporting the results of a similar experiment conducted by Packard and McGough (1996).

Conclusion

By examining the existing literature, we identified a simplified cognitive architecture of spatial navigation (Chersi and Burgess, 2014). We have implemented this architecture at the level of firing rate neurons and Hebbian and Reinforcement learning rules for synaptic plasticity and instantiated it in a simulated agent. Here we have validated the model by demonstrating that it is capable of reproducing some of the classic experimental tests of



spatial navigation. Our next steps will be to see how this same cognitive architecture generalizes to classic non-spatial tests of learning and planning (e.g. Daw et al., 2008).

2.6.3.4 Provenance of the data:

The data we utilized in our work was taken from 2 studies published by Packard and McGaugh (1996) and Pearce et al. (1998).

The data we produced is located at <http://se/data/kit/edu/SP3/3.4.1/Simulation Results>

Self-analysis of the value and completeness of data (vs DoW):

The utilized data is of great value and complete. The data produced by us is perfectly in line with the existing data.

Indication of who has used this data so far and for what

As far as we know, nobody has used our data yet. The TUM partner is interested in using our model (which for us is a different thing than data), but has still not practically started.

We will collaborate with Manchester to get help on the implementation of our model on the Spinnaker board.

2.6.3.5 Collaborations and interactions with other partners:

University of Leeds (Marc de Kamps)

University of Manchester

University of Munich

EITN Paris

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

2.7.1 M. Tsodyks (WIS)

2.7.1.1 Introduction

There is hardly a cognitive task that does not involve working memory (WM), including visual processing, speech comprehension and episodic memory. In view of its fundamental importance in cognition, it is puzzling that WM capacity is extremely limited, ranging between 3 and 6 items for most of healthy human participants. In psychological literature, it is often postulated that the brain possesses a specialized buffer, or 'focus of attention', where memory items can be temporarily placed for short periods of time and removed when needed, and hence working memory capacity corresponds to the size of this buffer. The neuronal implementation of the focus of attention and its size, as well as the way memory items can be placed and removed from it, are not yet understood. The most popular hypothesis in theoretical modeling is that working memory is mediated by persistent activity of neurons encoding the corresponding items in long-term memory. The maximal number of items simultaneously active depends on the characteristics of the network in a complex way, but there does not seem to be a fundamental upper limit on WM capacity in this model.

Recently, we proposed a synapse-based theory for short-term information storage in neural circuits. In this model, memory is retained by item-specific pattern of synaptic facilitation in recurrent connections between the corresponding neurons. Compared to the conventional model, this mechanism does not require neurons to fire with elevated rate for the whole duration of the memory task, resulting in a robust and metabolically more efficient scheme. Moreover, several items can be maintained in the WM via consecutive brief reactivations of the corresponding neuronal groups, such that at each time only one



item can be active, making the task of interpreting the activity by readout circuits more straightforward. Here we analyze this model with the goal of analytically estimating the maximal number of items that can be maintained in WM. The advantage of analytical expressions is that it allows one to make predictions about how WM capacity depends on the various synaptic, neuronal and circuit parameters that can potentially be tested by genetic manipulations.

2.7.1.2 Model Description

We consider a neural network model with long-term representations of memory items as interconnected neuronal ensembles with short-term plasticity (STP) of recurrent connections. To achieve an analytical hold on capacity estimation, we drastically reduce the complexity of the network to only leave the most essential features that allow it to function as working memory. We neglect the overlaps between different representation so each item is represented by a single excitatory unit (cluster) with self-excitation reflecting the strengthened connections between neurons encoding a given item in long-term memory. Different clusters are only interacting via reciprocal connections to a common inhibitory pool. The role of the inhibitory pool is to orchestrate the activities of excitatory clusters, such that only one cluster (i.e., only one memory item) can be active at any single moment.

Following the STP model developed in our previous work, recurrent connections are characterized by fixed 'absolute synaptic efficacy' and two dynamic variables; u , which stands for release probability, and x , the fraction of available neurotransmitter. If J_{ij} denotes the absolute synaptic efficacy between two excitatory neurons, then the instantaneous synaptic efficacy subject to STP is given by $J_{ij} u x$. Upon arrival of a spike, the release probability u temporarily increases due to the influx of calcium into the axon terminal of the presynaptic neuron, resulting in short-term facilitation. Meanwhile, the fraction of available neurotransmitters x decreases due to neural signalling consuming resources, resulting in short-term depression. After neuronal spiking, u returns to its baseline value u_0 with a time constant τ_u , and x recovers to its maximum value $x = 1$ with a time constant τ_x .

The resulting network model has 3 differential equations for each of P excitatory clusters (synaptic current h_μ and two STP variables u_μ and x_μ for each cluster and one other equation for the inhibitory pool current h_i):

$$\tau \frac{dh_\mu}{dt} = -h_\mu + J_{\mu\mu} u_\mu x_\mu R_\mu - J_{\mu i} R_i + I_\mu$$

$$\frac{du_\mu}{dt} = \frac{U - u_\mu}{\tau_u} + U(1 - u_\mu) R_\mu$$

$$\frac{dx_\mu}{dt} = \frac{1 - x_\mu}{\tau_x} - u_\mu x_\mu R_\mu$$

$$\tau \frac{dh_i}{dt} = -h_i + J_{i\mu} \sum_\mu R_\mu$$

where τ is the neuronal time constant, for simplicity taken to be the same for excitation and inhibition, and $R(h) = \epsilon \log(1 + e^{h/\epsilon})$ is neuronal gain chosen in the form of a smoothed



threshold-linear function, also same for excitatory and inhibitory neurons. I_h is the constant background excitation that we assume to reflect the attentional state of the network (see below).

2.7.1.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.
- Data, algorithms, tools and methodologies storage location(s) (and links?)
- Description of algorithms/models/principles:
 - Format, language if applicable.

Matlab

- Name of DICs/software catalogue/or HBP github project entries.
- Description of data:
 - Species, sex, age, number of specimen/subjects.
 - Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.
- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

The model itself fits what was expected, the analytical results exceed the expectations.

- Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
- Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- Outline state of validation work.
- Data Quality and Value:
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.

Working memory is a fundamental component of many cognitive processes, so I hope the model can be used in many different contexts when studying cognition.

- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.
- Are the data/algorithms/models considered final?

For the time being yes, but the model can be adapted to different contexts as mentioned above.

- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

The paper by Yuanyuan Mi and Misha Tsodyks ‘Synaptic theory of working memory capacity’ is about to be submitted.

2.7.1.4 Model Results

With proper parameters that are compatible with experimental measurements of inter-pyramidal connections in the prefrontal cortex, our model can extract 4 items from long-term memory by activating the corresponding clusters with transient external inputs, and retain them in WM in the form of brief reactivations which we call population spikes (PSs; see figure). Each time a certain cluster emits a PS, it triggers a modification in the instantaneous synaptic efficacy of recurrent connections within this cluster, characterized by transient depression and subsequent facilitation that maintains the corresponding item in WM until the next PS of the cluster (Fig. D). When 5 items were loaded, only 4 of them could be maintained successfully while the fifth one fades away (see Fig. E, ~F), indicating that for this set of parameters, the capacity of WM is 4.

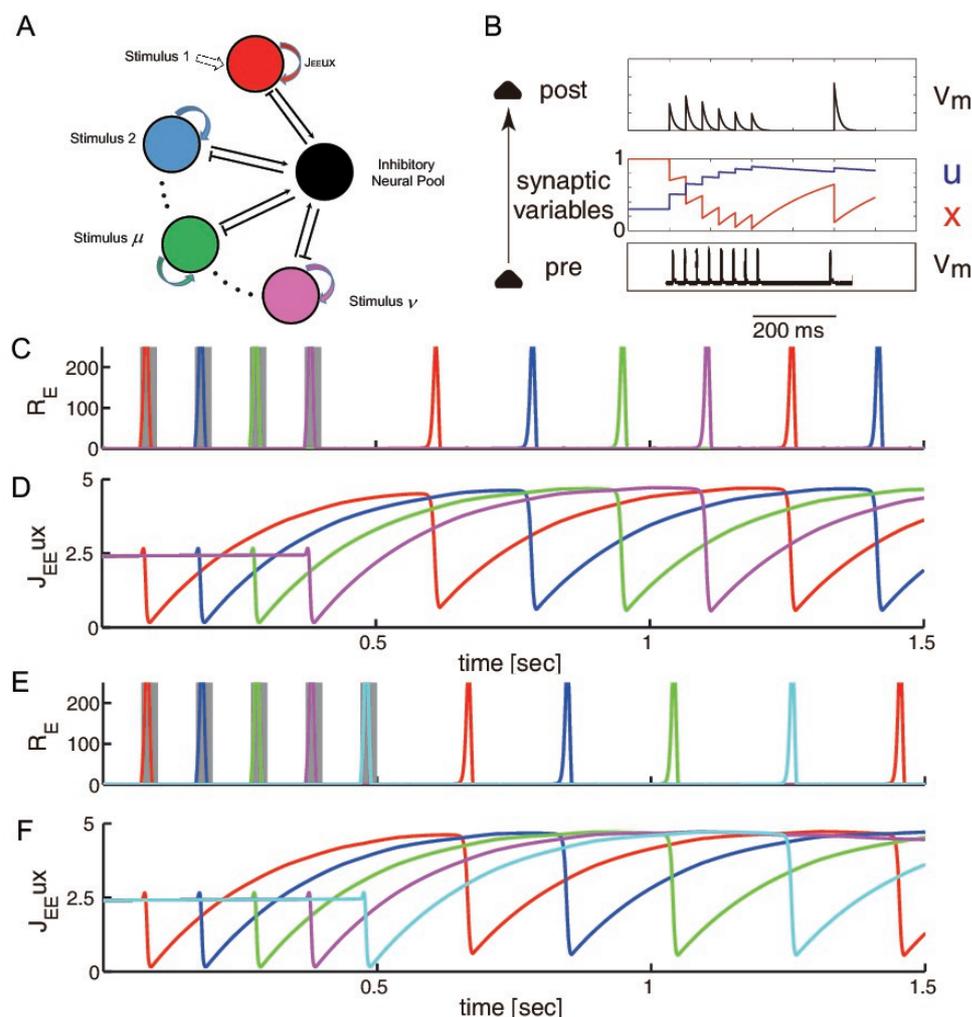


Figure 45: Short term plasticity based working memory network model.

A. Network architecture: a number of recurrent excitatory neural clusters, shown in different colors, reciprocally connected to an inhibitory neuron pool, shown in black. Each excitatory cluster represents one memory item. Within each excitatory cluster, recurrent excitatory connections exhibit short-term synaptic plasticity. B. Model of a synaptic connection with STP. In response to a presynaptic spike train (lower panel), the neurotransmitter release probability u increases and the fraction of available



neurotransmitter x decreases (middle panel), representing, correspondingly, synaptic facilitation and depression. V_m is the membrane potential (upper panel).

C. Network simulation with four loaded memory items. Upper panel: Firing rates of different clusters after transient simulation. Four clusters are sequentially stimulated by brief external excitation (shaded colored rectangles). Different colors correspond to different clusters as in A. Following the stimulation, all four clusters continue sequential activation in the form of PSs. Lower panel: The instantaneous synaptic efficacy J_{ij} for stimulated neural clusters during loading and subsequent reactivations. D. Same as C with five loaded items. The parameters are:

$$J_{ii} = 8, \tau_f = 1.5s, \tau_d = 0.3s, U = 0.3, \tau = 8ms, J_d = 3, J_{ij} = 1.75, J_{ij} = 1.1, \alpha = 1.5, P = 16$$

We analytically estimated the maximum number of items that can be maintained in WM by the ratio of two temporal scales: (i) the maximal period T_{max} of the limit cycle of the network, i.e. the maximal time between subsequent reactivation of each cluster; and (ii) the temporal separation between two consecutive PSs, referred to as t_s , the capacity of WM is given by the maximum number of PSs that can be accommodated in a single period of the limit-circle, i.e. by $C \approx T_{max} / t_s$. The resulting estimated expression for capacity is

$$C : \frac{\tau_d}{\tau} G(J, \tau, \tau_d, \tau_f; I_{back} - I_{crit})$$

where G is a weak function of parameters. This analytical result was confirmed with numerical simulations of the model. The above expression shows that WM capacity is increasing with the time constant of synaptic depression in recurrent connections and decreasing with the neuronal time constants in the network. These analytical predictions could be amenable for experimental verifications by genetic manipulations of these parameters. The model also predicts that synaptic facilitation time constants determines the minimal duration of pauses that allows the stream of inputs to be segmented into different chunks. These effects could be relevant for many cognitive processes where chunking was shown to play a crucial role, such as e.g. speech processing.

2.7.1.5 Provenance

Not applicable.

2.8 T4.3.3 Models of biologically realistic network states; wakefulness & sleep

2.8.1 G. Deco (UPF) _ Cortical model of wake and sleep

2.8.1.1 Introduction

The group under Prof. Deco has started a collaboration with Prof. Dehaene (WP3.1) to examine dynamical markers of wakefulness in fMRI data.

2.8.1.2 Model Description

The cortex model is parcellated into 432 cortical and subcortical areas. The aim is to interpret fMRI data acquired from four anesthetized (propofol; deep: 31 trials, light: 25 trials) and awake monkeys during quiet wakefulness (21 trials). To do so, we estimate directed effective connectivity from the data with the help of a recently developed whole brain model (cf. T4.3.1). It is based on a noise diffusion process and optimizes the correspondence between the functional connectivity (covariances of the fMRI time series data) and the intracortical interaction strengths. Details are provided in T4.3.1 and a recent publication: Gilson, Moreno-Bote, Ponce-Alvarez and Deco, Estimation of Directed



Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome, PLoS Comput Biol (in press).

2.8.1.3 Model Data

- T4.3.3 for data analysis, T4.3.1 for model and code development, WP3.1 for experimental data.
- Description of algorithms/models/principles:
 - Model and code described in T4.3.1; cf. www.unic.cnrs-gif.fr/owncloud.
- Description of experimental data:
 - Secondary use of data published: P Barttfeld, L Uhrig, JD Sitt, M Sigman, B Jarraya, S Dehaene. Signature of consciousness in the dynamics of resting-state brain activity. PNAS (2015)
 - Parcellation of 432 cortical and subcortical areas. fMRI data acquired from four anesthetized (propofol; deep: 31 trials, light: 25 trials) and awake monkeys during quiet wakefulness (21 trials).
- Completeness of data/algorithms/models:
 - The method and model have been verified on many datasets.
 - The data analysis focuses on information about the directed cortico-cortical connectivity to examine the information flow in the three states of wakefulness. This is supported by recent results about the spatio-temporal structure of fMRI time series (Mitra et al, eLife 2015). We aim to provide a mechanistic explanation to this phenomenological analysis.
- The analysis is ongoing to uncover principles of cortical branching patterns that underlie the changes observed in fMRI.
- Data/algorithm/model usage to date:
 - Algorithm developed and used by T4.3.1.
- References:

Barttfeld, Uhrig, Sitt, Sigman, Jarraya and Dehaene. Signature of consciousness in the dynamics of resting-state brain activity. PNAS (2015)

Gilson, Moreno-Bote, Ponce-Alvarez and Deco. Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome, PLoS Comput Biol (in press).

Mitra, Snyder, Tagliazucchi, Laufs and Raichle. Propagated infra-slow intrinsic brain activity reorganizes across wake and slow wave sleep, eLife (2015) 4:e10781

Tagliazucchi, von Wegner, Morzelewski, Borisov, Jahnke and Laufs. Automatic sleep staging using fMRI functional connectivity data. Neuroimage (2012) 63: 63-72

Boly, Perlberg, Marrelec, Schabus, Laureys, Doyon, Péligrini-Issac, Maqueta and Benali. Hierarchical clustering of brain activity during human nonrapid eye movement sleep. PNAS (2012) 109: 5856-5861

2.8.1.4 Model Results

Fig. 46 shows the empirical FC for the 3 states. FC spatio-temporal information (covariances with various time lags of correlation, not just “spot” covariances) is used to extract information about the directionality of intracortical connections. This analysis sits in the context of ‘feedforward’ and ‘feedback’ pathways in the brain, namely from sensory to high-level areas.

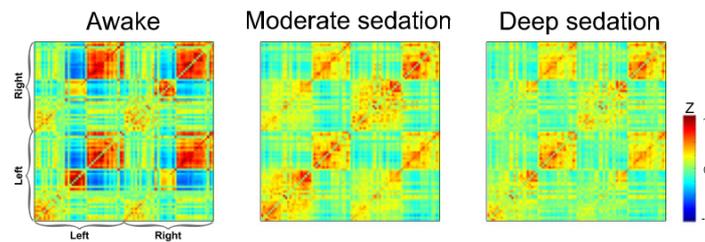


Figure 46: Empirical FC for the three states of wakefulness.

Here the spot covariances between the BOKD time series of all regions are represented. Data from Barttfeld et al, PNAS (2015).

The model is tuned to individual sessions and statistical tests are used to assess the significant differences between the estimated connectivities in the various states. We focus on differences in the relative strength of feedforward and feedback connections that could serve as a marker for conscious and unconscious vigilance states. Initial results indeed indicate that asymmetry is generally larger during anesthesia and more pronounced in frontotemporal areas and midline structures (Fig. 47). This is in line with recent results (Mitra et al, eLife 2015); we aim to go beyond a phenomenological description and bring a mechanistic explanation for the observed changes in FC. Further analysis will examine whether different functional connectivity states as found by the group of Prof. Dehaene during both anesthesia and wakefulness are also accompanied by different effective connectivity profiles.

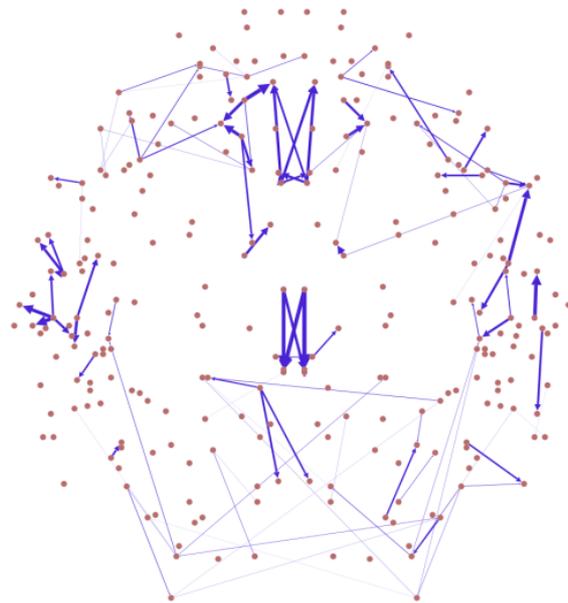


Figure 47: Effective connectivity differences between anesthesia and awake states.

Cortical areas are depicted as red circles and the blue lines reflect significantly stronger effective connectivity during anesthesia as compared to wakefulness with the preferred activity flow pointing into the direction of the arrow.

With another set of human fMRI data during wake and sleep (Tagliazucchi et al., Neuroimage 2012; Boly et al., PNAS 2012), we examine the temporal and spatial dynamics. The focus is on distinguishing these two states with data analysis and model-based methods. The data analyses in Fig. 48, based on two different data sets, show significant differences in various measures between the two states:

- The mean functional connectivity (FC), the mean over nodes of the Pearson correlation between the BOLD time series between each node, is significantly lower in slow-wave sleep than in awake.

- The global phase synchrony, the mean of the phase synchronisation over time, is significantly lower in sleep than in awake.
- The integration, the length of the largest component in the binarized, thresholded FC matrices, is significantly lower in slow-wave sleep than in awake.

The differences have been tested using permutation testing (p-value: 0.0099).

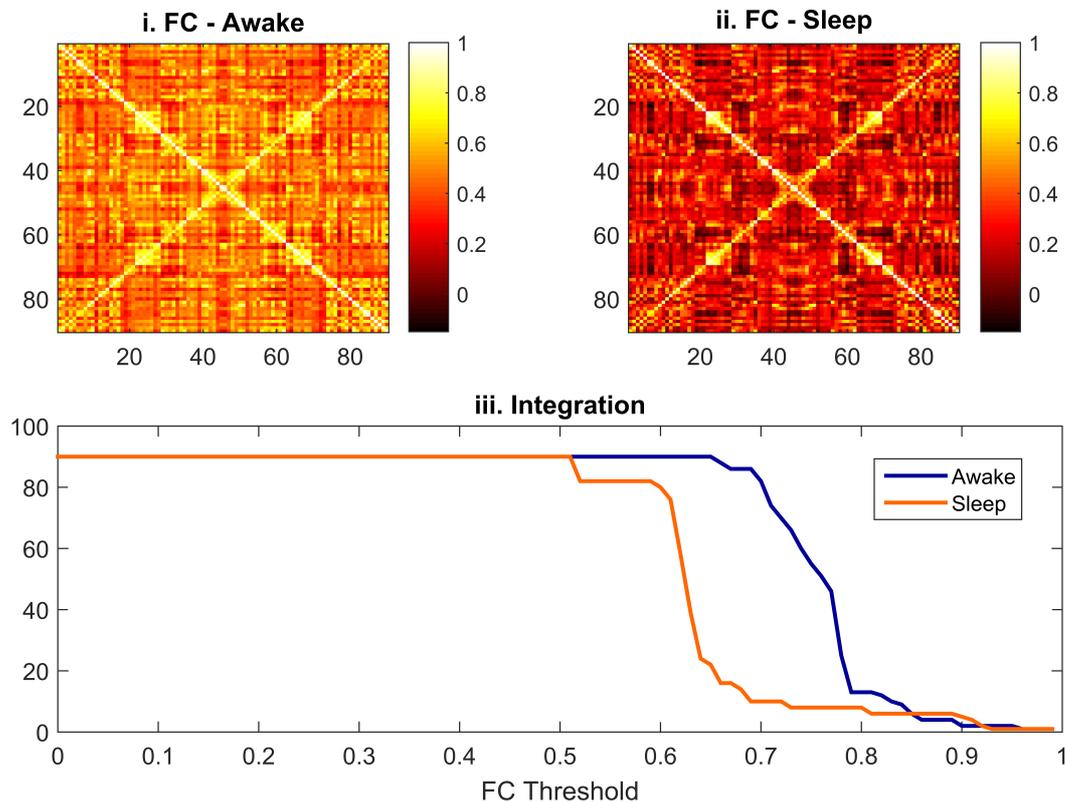


Figure 48. Data analysis

(i) Functional connectivity matrix for the awake state. (ii) Functional connectivity matrix for the sleep state. (iii) Integration measure as a function of the FC Threshold.

The model-based analyses suggest that these results are due to a global shift of the dynamical system to the asynchronous regime during sleep (see Fig.49). The optimal fit between the empirical data and the model moves with increasing global coupling during sleep to more negative bifurcation parameter (p-value: 0.0099, permutation test), meaning asynchronous dynamics, whereas in the awake state the system stays closer to the edge of the bifurcation, the brink between asynchronous and oscillatory dynamics.

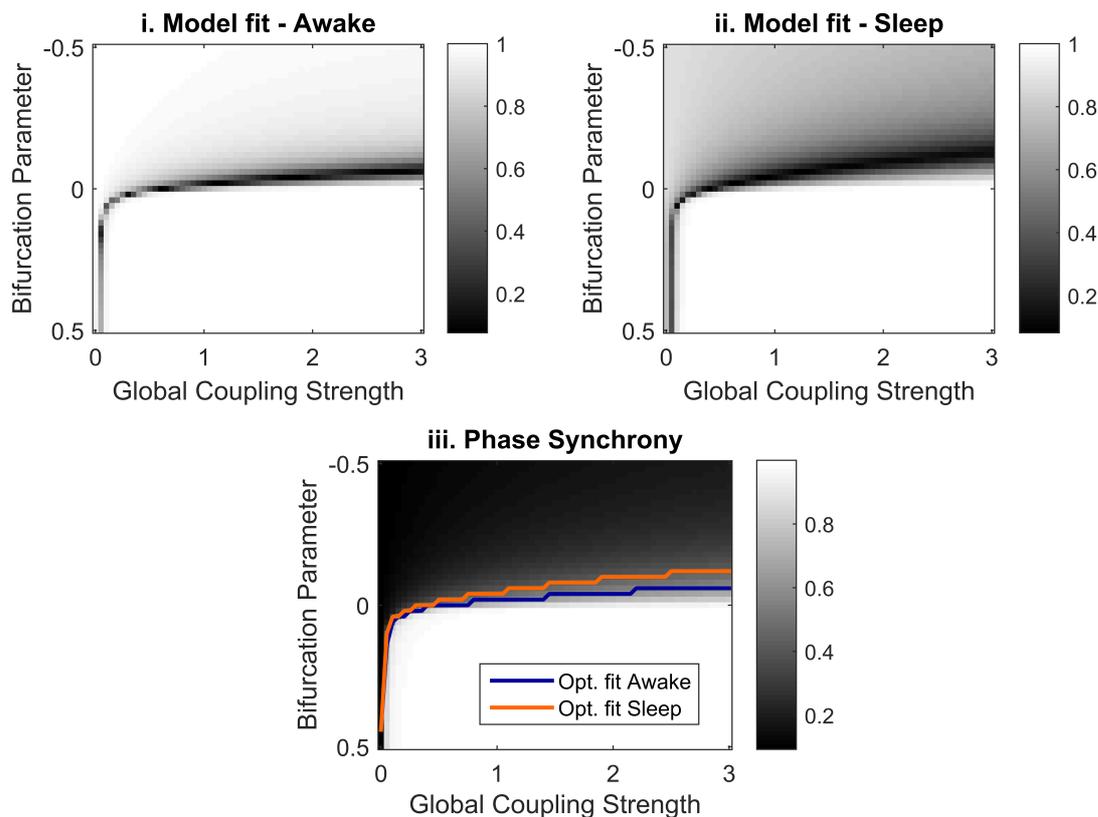


Figure 49. Parameter exploration of whole brain model

(i) Model fit (Kolmogorov-Smirnov-Distance between the distribution of empirical FC values and simulated FC values) for the awake state: the optimal fit is the minimum KS-Distance. (ii) Model fit (Kolmogorov-Smirnov-Distance between the distribution of empirical FC values and simulated FC values) for the sleep state: the optimal fit is the minimum KS-Distance. (iii) Simulated global phase synchrony and the optimal fit for both awake and sleep

The profile of dynamical interactions will be used to constrain large-scale networks of neural-mass models or spiking neurons (SP6) to investigate transition between the activity patterns in wake and sleep, thanks to our estimation of biomarkers (connection strengths and excitability).

2.8.1.5 Provenance

Data from WP 3.1 (Prof S Dehaene) and model from T4.3.1 (Prof G Deco, related team).

2.8.2 A. Destexhe (CNRS) _ Models of biologically realistic network states

2.8.2.1 Introduction

Comparing waking and sleep states is the main source of knowledge for understanding how spontaneous activity in the brain affects neuronal responses and information coding. Understanding the role of spontaneous activity, and its interaction with sensory inputs, is crucial for understanding how information is represented and processed in the brain. In fact, spontaneous activity is a major aspect in which the brain and engineered information processing systems differ. EEG recordings from cortex and thalamus show very different patterns of spontaneous activity in different brain states. During slow wave sleep and anesthesia, the activity across large populations of neurons is dominated by a synchronous regular pattern, showing slow-frequency oscillations. In contrast, during waking and REM sleep, so called activated states, the population activity exhibits an asynchronous irregular network state with a highly complex dynamics. Intracellular recordings in vivo show that, in such complex dynamics, cortical and thalamic neurons are in a stable depolarized state



close to the firing threshold, embedded in a “noisy” irregular pattern of background synaptic activity which, occasionally, lead to neuronal spiking. It is during the apparently noisy dynamical regime, which characterizes activated states, that the main computational tasks are preformed. Ongoing spontaneous activity in the network is generated by the combination of intrinsic electrophysiological properties of single neurons and their synaptic interactions; it is globally affected by external neuromodulatory inputs and correlated with the functional state of the brain. Within this context, we are specifically working on understanding the role of correlations between pairs of neurons in information processing during different brain states. Pair-wise response correlations can arise when neurons received shared sensory inputs, but also when they are subject to correlated modulatory influences that are not sensory in origin.

From published 2D multielectrode array recordings in human neocortex from epileptic patients during sleep, Peyrache et. al. (PNAS 2012) reported a detailed functional characterization of putative excitatory (E) and inhibitory (I) cells. The absolute value of the spatial correlation coefficient (CC) between homogeneous pairs, E-E and I-I, was reported. E-E correlations showed a marked exponential decay with distance while I-I correlations did not. However, synaptic connections between I neurons are believed to be more local than the E ones. This result opens a new intriguing question, which we aim to address in our model. First, we have implemented the data analysis presented in Peyrache et. al. by studying the impact of silent periods, or down states, in the network dynamics. Second, we are developing mechanistic spiking models of different brain states. We have designed topological balanced networks characterized by adaptive exponential integrate and fire neurons with conductance based synapses and a probability of connection which decays with distance, according to a Gaussian profile. We are still exploring the dynamics of the network in different parameter regimes and considering different implementations. To shed some light in the implementations we need to make, we are also developing a phenomenological model, which recast the qualitative features of the correlation-distance dependence observed in the data.

2.8.2.2 Model Description

In order to understand relevant aspects of the correlation-distance dependence of neuronal pairs during different brain states, we have first refined and implemented the data analysis presented in Peyrache et. al. We have refined the analysis by including the sign of the correlation coefficients to explore the impact of negative correlations between neuronal pairs. We have also included correlations between heterogeneous pairs, E-I, in our analysis. As an implementation, we have characterized up and down states from the multiunit activity; as expected, down states were more abundant during slow wave sleep than during awake or REM. With this characterization we could isolate the time intervals of activation, or up states, and compute pair-wise correlations only considering those. For this step, we followed Renart et. al. (Science 2010): every ISI of the MUA longer than 50 ms was associated with a down state, to create the margins of each down state we smoothed the MUA with a Gaussian density of width 10 ms and we set a threshold at 20% of its maximum of the whole recording session, the down state began at the first point in time within the associated ISI at which the smoothed MUA reaches a value under threshold, and ended at the first point in time within the ISI at which it reaches a value over threshold. The time intervals characterized as down states were removed from all spike trains and the up-state time intervals were merged together (unless they were shorter than 50 ms). From the resulting spike trains, we computed again the pair-wise correlations. We find that, considering positive and negative correlation coefficients, fluctuations between up and down states have a relevant impact on the magnitude of the correlation coefficient of E-E pairs, as we show below. Surprisingly, the correlation coefficients between I-I pairs, do not show to be affected by activated and silent fluctuations in the state of the network. We hope to be able to reproduce this behavior in a mechanistic model.

From the modeling perspective, we are, in parallel, developing a simulation and a more phenomenological model which conceptually recast the features to be implemented in the simulation.

We use PyNN and the NEST simulator to design 2D topological balanced networks. The simulation is composed of $N = 10000$ neurons (80% excitatory and 20% inhibitory). The neurons are model as conductance-based adaptive exponential integrate-and-fire (AdEx IF) with a membrane time constant $\tau_m = 20$ ms. The membrane potential spiking threshold is set to $V_t = -50$ mV, the reset potential to $V_r = -60$ mV and the refractory period to $\tau_{ref} = 5$ ms. These parameters were chosen as biologically realistic according to previous studies. Through adaptation regular spiking (excitatory) neurons and fast spiking (inhibitory) neurons are generated. The synaptic connections between neurons are modeled as transient conductance changes in which the synaptic time course is an instantaneous rise followed by an exponential decay. The synaptic time constants are $\tau_{exc} = 5$ ms for excitation and $\tau_{inh} = 10$ ms for inhibition and the reversal potentials are $E_{exc} = 0$ mV and $E_{inh} = -80$ mV. Every neuron is connected with the rest of the network with a connection probability that depends on the distance between two neurons in the network, according to a Gaussian profile with σ_{exc} and σ_{inh} the variances of the connectivity profile, i.e., the spatial spread of the Gaussian profile. For each neuron, the total number of synapses is fixed and the incoming connections are random. In Fig. 50 we show a diagram of the network.

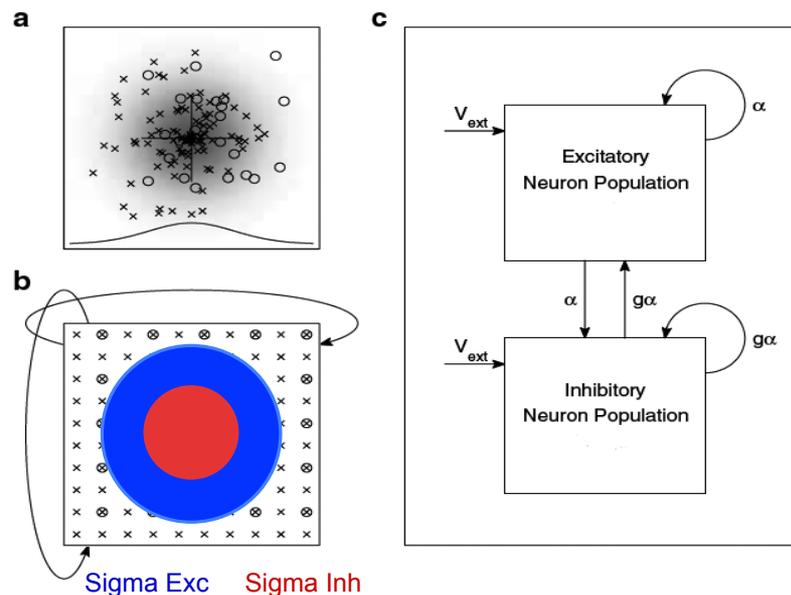


Figure 50: topological balanced network with conductance-based adaptive exponential integrate and fire neurons and a probability of connection which decays with distance, according to a Gaussian profile.

On the other hand, we built a simple phenomenological model of pair-wised correlated neurons which receive a fraction of shared inputs from a pool of presynaptic neurons. The firing of a single neuron in the model is represented by a double-stochastic process. Each neuron have two types of variability: one is related to the Poisson-like variability inherent to spike generation and the other to random fluctuations in the (time-dependent) firing rate. The fluctuations could represent gain modulations in sensory systems or modulations of excitability (Goris, Movshon, and Simoncelli, Nature Neuroscience 2014). We assume that the time-course of gain fluctuations is the same for the whole population, but their amplitude varies from neuron to neuron. In other words, we can write the firing rate of the



presynaptic pool as the product of a (Gamma-distributed) random process (with norm 1,), and the population-firing rate. The realization of the double-stochastic process is the numbers of spikes per bin given by a non-homogeneous Poisson distribution. Each neuron within a pair gets inputs from such gain-modulated population. Some of the inputs are common for both neurons, but others are not. We assume that with increasing distance neurons share a lower fraction of common inputs.

2.8.2.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.

T4.3.3 UNIC/EITN, Alain Destexhe

- Data, algorithms, tools and methodologies storage location(s) (and links?)
- Description of algorithms/models/principles:
 - Format, language if applicable.

PyNN and NEST simulator for the network simulation

Python for the phenomenological model

- Name of DICs/software catalogue/or HBP github project entries.
- Description of data:
 - Species, sex, age, number of specimen/subjects.
 - Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.
- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?
 - Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- Outline state of validation work.
- Data Quality and Value:
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.
- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.

- Are the data/algorithms/models considered final?
- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

2.8.2.4 Model Results

In the simulation the parameters are tuned within the regime in which the network shows self-sustained asynchronous irregular activity (see Fig. 51). We have performed an exploration of the network dynamics mainly by varying the spatial spread of the Gaussian profile σ_{exc} and σ_{inh} (see Fig. 52).

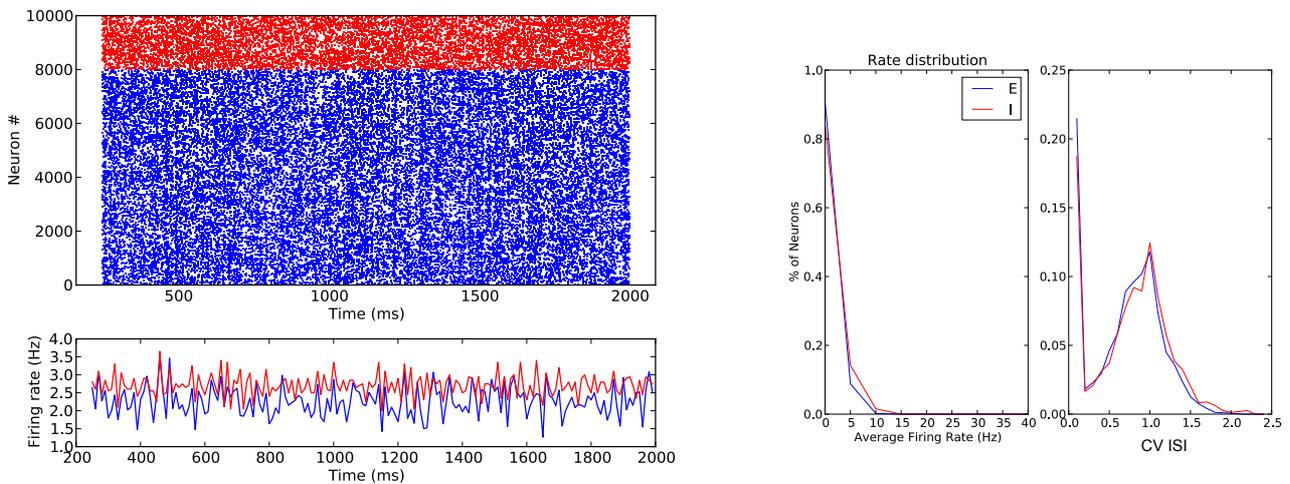


Figure 51: Raster plot and average firing rate obtained from the network simulation for $\sigma_{exc} = \sigma_{inh} = 0.15$.

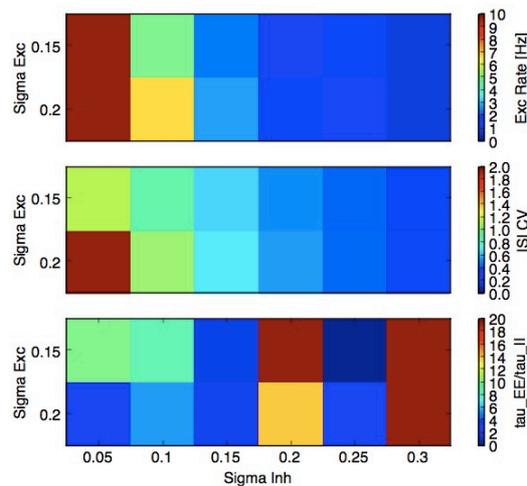


Figure 52: Parameter exploration of the network

By varying σ_{exc} and σ_{inh} we compute the excitatory firing rate, the coefficient of variation of the interspike interval (ISI CV) and the ration between the spatial decay of E-E pairs (τ_{EE}) and the spatial decay of I-I pairs (τ_{II}).

The model is still in progress and we need to implement the network simulation in order to reproduce the pair-wise correlations-distance dependence obtained from the analysis of the data. As an example, we show in Fig. 53 the correlation coefficient versus distance for $\sigma_{exc} = \sigma_{inh} = 0.15$.

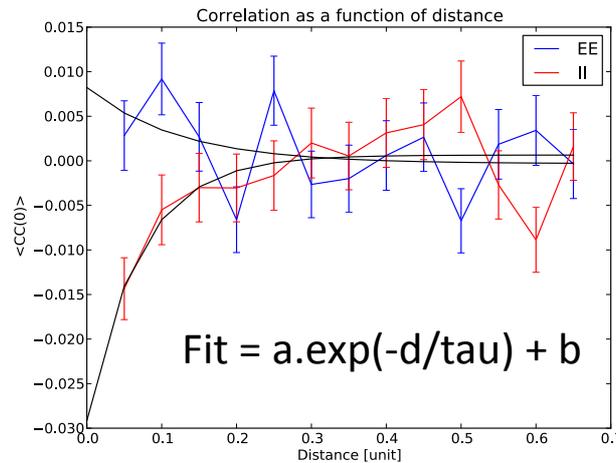


Figure 53: Correlation coefficient versus distance of E-E pairs (blue) and I-I pairs (red) for $\sigma_{exc} = \sigma_{inh} = 0.15$.

From the more phenomenological model, as expected (see Fig. 54), the correlation coefficient decreases when neurons have less synaptic inputs in common. But, the sensitivity of the correlation coefficient to this fraction depends on the firing rate. For fast firing neurons, the function decays much slower with distance, as we obtained from the data analysis. In the mechanistic implementation, a pair of neurons is receiving conductance-based inputs from a shared pool of pre-synaptic inhibitory and excitatory neurons. The modulation of the gain will be implemented as the fluctuations of the inhibition and excitation, keeping the balance. The differences in the firing rate between the two post-synaptic neurons will be implemented by means of neuronal adaptation.

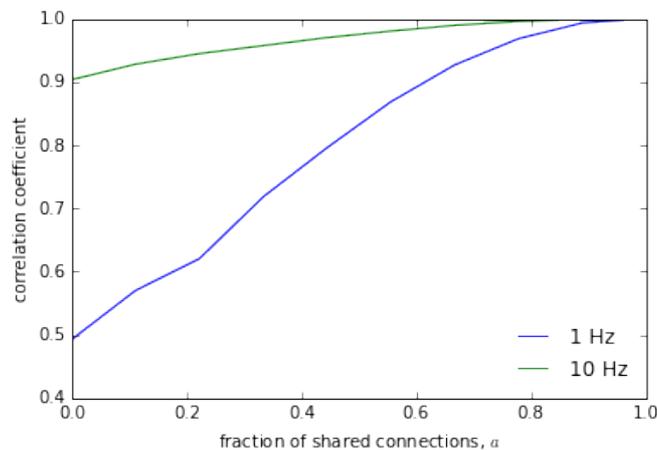


Figure 54: Correlation coefficient as a function of shared inputs for different firing rates.

2.8.2.5 Provenance

NA



2.8.3 M. Diesmann/A. Morrison (JUELICH) – Models of biologically realistic network states; wakefulness & sleep

2.8.3.1 Introduction

Acetylcholine (ACh) is an important neuromodulator in the regulation of brain states, including the sleep-wake cycle^{1,2}. Since ACh is related to attention and arousal, and can modify the signal-to-noise ratio of sensory responses, understanding its actions can contribute to models of cognition. While both single-neuron and network effects of ACh on cortex have been measured experimentally, a model systematically linking these levels taking into account layer and cell-type specificity is missing to date. In this task, Jülich aimed to establish such a model.

2.8.3.2 Model Description

Jülich created a full-density spiking model of a cortical microcircuit to investigate how acetylcholine (ACh) influences cortical network dynamics. Layers 2/3, 4, 5, and 6 are modeled with an excitatory (E) and an inhibitory (I) population of leaky integrate-and-fire neurons each. The model is based on the cortical microcircuit model of Potjans and Diesmann³ but the numbers of neurons are adjusted to more closely reflect mouse barrel cortex. The numbers of neurons are [2/3E: 1700, 2/3I: 170, 4E: 1700, 4I: 170, 5E: 1100, 5I: 110, 6E: 1300, 6I: 130]. Furthermore, the relative inhibitory synaptic weight is adjusted, as are the delays and synaptic weight distributions. The synaptic weight distributions are modeled as lognormal rather than the original Gaussian distributions. External Poisson input was applied to all neurons (independent across neurons) at rates found by an automatic procedure to yield realistic population-specific firing rates. The Poisson rates are [2/3E: 1129, 2/3I: 975, 4E: 1345, 4I: 1198, 5E: 2892, 5I: 2875, 6E: 1570, 6I: 1498] spikes/s. Table XXX lists the remaining parameters that differ with respect to the model of Potjans and Diesmann.

Table 1: Parameters of the cortical microcircuit model

Parameter	Symbol	Value
Transmission delay from E neurons	d_E	1.2 ms
Transmission delay from I neurons	d_I	0.7 ms
Median excitatory synaptic strength	μ_E	2.7 mV
Mean excitatory synaptic strength	w_E	3 mV
Median inhibitory synaptic strength	μ_I	27 mV
Mean inhibitory synaptic strength	w_I	30 mV
Membrane time constant	τ_m	20 ms
Membrane capacitance	C_m	200 pF
Leak potential	E_L	-67 mV
Synaptic time constant	τ_s	5 ms

Cell-type-specific depolarizing and hyperpolarizing effects of ACh^{4,5,6} were taken into account by changing the resting membrane potentials by the following amounts: [2/3E: +1 mV, 2/3I: +1 mV, 4E: -1 mV, 4I: 0 mV, 5E: +1 mV, 5I: +1 mV, 6E: +1 mV, 6I: +1 mV]. The

effects on layer 6 neurons are based on unpublished observations of Günter and Feldmeyer (Jülich). ACh can have opposite effects even on different inhibitory neuron types in a single layer, but we tried to take into account the main effects.

ACh sparsifies cortical activity, with the most active neurons remaining active while the least active neurons become even less active^{7,8}. Furthermore, ACh has been shown to weaken the synapses of several connection types, including excitatory connections onto excitatory neurons in layers 3-5^{5,9,10}. We hypothesized that the differential effects of ACh on the activity of low-rate and high-rate neurons are due to selective weakening of already weak synapses. Figure 55 shows the synaptic weight distributions in the model in the presence and absence of ACh.

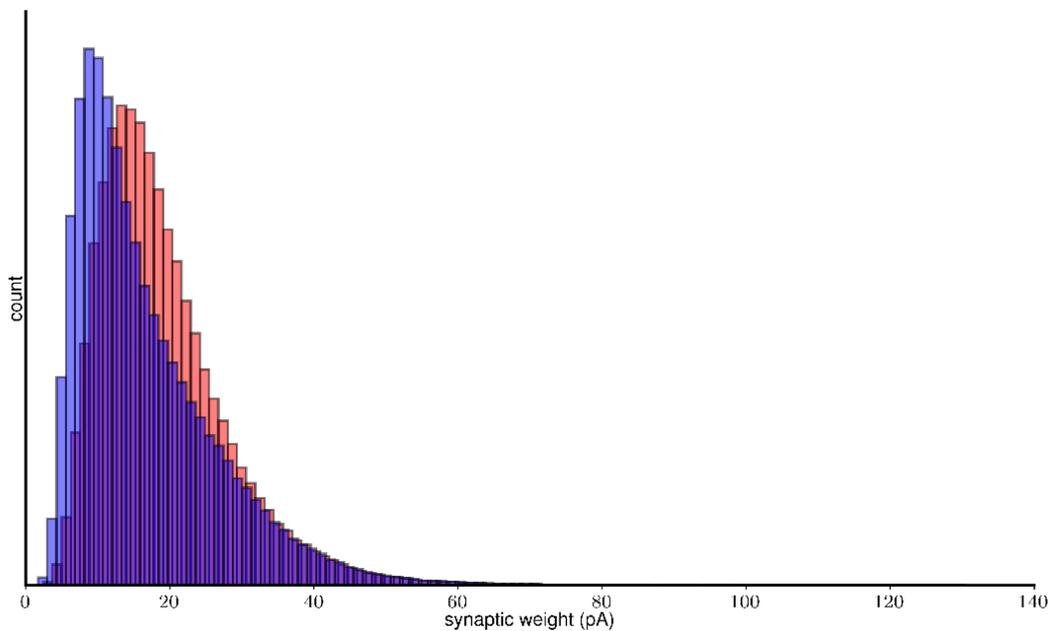


Figure 55: Distributions of excitatory synaptic weights in the model with (blue) and without (red) ACh.

We quantified the neuromodulator (NM) effect as the mean firing rate of each neuron in the presence of ACh divided by its mean firing rate in the absence of ACh. In experiments where the cholinomimetic drug Carbachol was applied to mouse primary visual cortex, the distributions of effect sizes across neurons at several Carbachol concentrations were found to be negatively skewed⁸. We tested whether the same holds in our model.

Another notable effect of ACh for which we tested is the decorrelation of spontaneous activity⁷.

In terms of evoked activity, ACh is thought to increase the signal-to-noise ratio of responses to sensory stimuli¹¹. Correspondingly, ACh enhances the strength of thalamocortical synapses via nicotinic receptors⁸. We simulated transient thalamocortical inputs by increasing the Poisson rates impinging on layer 4 neurons by 500 spikes/s for 100 ms. In the control condition, thalamocortical synaptic strengths equaled the mean excitatory synaptic strength in the circuit. To model the condition with ACh, they were doubled. We quantified responses to thalamic stimuli as the difference in the peristimulus time histogram (PSTH) between spontaneous and driven conditions, and tested whether ACh enhanced these responses.

The network was simulated with NEST 2.8 for 10 seconds of biological time and the first 100 ms were discarded for the calculation of dynamical measures.



2.8.3.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.

Jülich model: This work was done by Andrei Maximov and Sacha van Albada. The responsible person is Abigail Morrison.

- Data, algorithms, tools and methodologies storage location(s) (and links?)

Python scripts are in a git repository administrated by Jülich

- Description of algorithms/models/principles:
 - Format, language if applicable.

PyNEST

- Name of DICs/software catalogue/or HBP github project entries.

not applicable

- Description of data:
 - Species, sex, age, number of specimen/subjects.

Not applicable

- Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.

Simulations of a cortical microcircuit at cellular and synaptic resolution

- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?

This model was intended as a first, exploratory version and that is what was achieved.

- Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community

The project is being continued in Jülich outside the HBP to ensure preservation and extension of the knowledge gained. Once a more mature version has been developed, the code will be made available in a public repository, for instance github or Open Source Brain.

- Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.

The model mostly uses data from the literature, but in addition some new, unpublished data from the group of Dirk Feldmeyer in Jülich. The latter determined cell-type-specific effects of acetylcholine on the resting membrane potential of neurons in cortical layer 6. These data help determine the parameter changes applied in the model.

- Outline state of validation work.

Based on parameter changes reflecting experimental results at the single-neuron level, several network effects were qualitatively reproduced. The work builds on an existing model of a cortical microcircuit that has been more extensively validated and used in several follow-up projects such as a multi-area model of macaque visual cortex that is also part of SP4.



- Data Quality and Value:
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.
- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.
- Are the data/algorithms/models considered final?

No

- No publications yet

2.8.3.4 Model Results

Simulations of the control condition without ACh-induced changes yield irregular activity with reasonable firing rates and a low level of synchrony (Fig. 56).

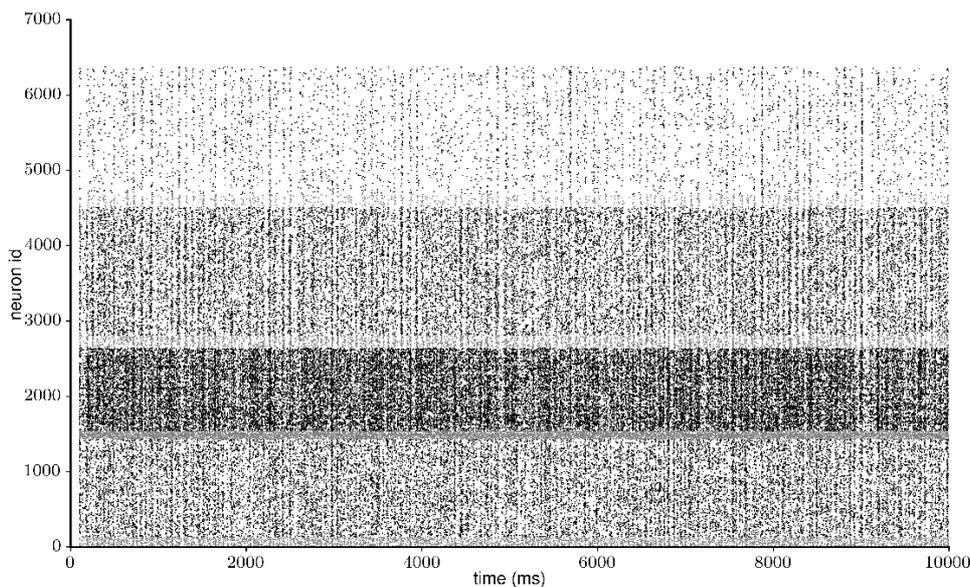


Figure 56: Raster plot showing spike times of, from top to bottom, populations 2/3E, 2/3I, 4E, 4I, 5E, 5I, 6E, and 6I.

Simulations with parameters mimicking ACh effects give altered firing rates distributions compared to the control condition, with negatively skewed effect size distributions for layers 2-5 (Fig. 57). Furthermore, ACh decorrelates the spiking activity in these layers. Both effects are in line with experimental results^{7,8}.

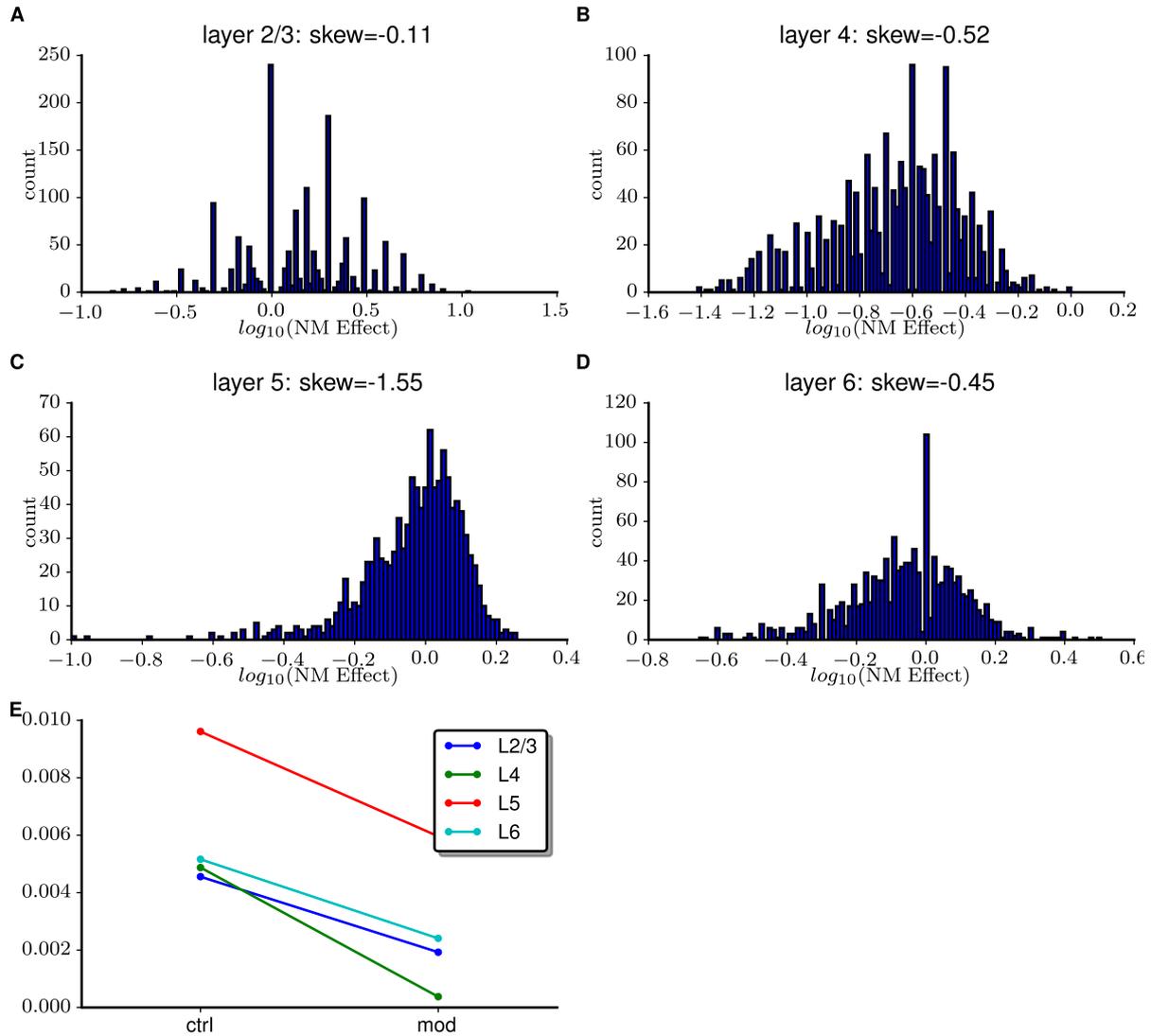


Figure 57: Effect of ACh on firing rate distributions of excitatory neurons.

A-D: Neuromodulator (NM) effect quantified as the ratio of firing rates with and without ACh. Distributions across neurons for layer 2-6 are shown. E: ACh reduced mean pairwise cross-correlation coefficients between excitatory neurons in each of layers 2/3, 4, and 5.

Also in line with experimental findings, ACh enhances responses to modeled transient thalamocortical inputs (Fig. 58). This enhancement is most prominent in layer 4.

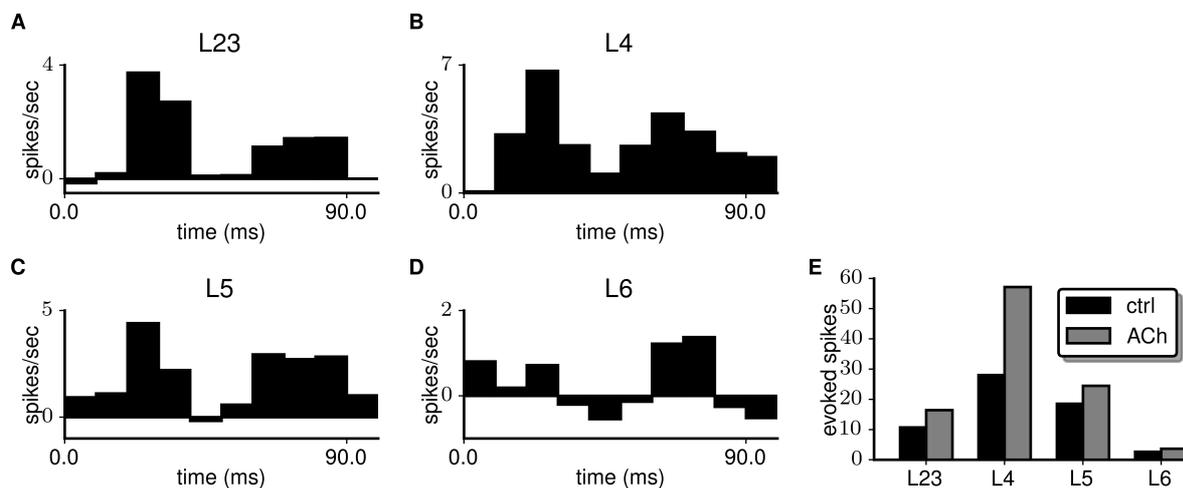




Figure 58: ACh enhances thalamocortical responses in the model, quantified as the difference between average driven PSTH and spontaneous rates.

A-D: PSTH of thalamically evoked activity in layers 2-6 in control condition with no ACh-derived modifications. E: Total number of evoked spikes (calculated as an integral of PSTH) for layers 2-6 in control (black) and modified by ACh (grey) conditions.

In conclusion, the Jülich model reproduces cholinergic effects on several aspects of spontaneous and evoked cortical activity. Future work will consider more systematic parameter scans, a more precise matching of parameter changes in the model to experimental results, and potentially a further distinction between subpopulations that undergo differential effects in response to acetylcholine.

2.8.3.5 Provenance

NA

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

2.9.1 ML. Linne (TUT)

2.9.1.1 Introduction

Recent evidence indicates that astrocytes actively participate in the functions of cortical synapses *in vivo*, and may therefore be crucial for understanding information processing, plasticity and learning in the cerebral cortex. Our goal in T434 was to develop a biophysically and biochemically detailed model of astrocyte-neuron interactions (ANI) to test the accumulating hypotheses about the involvement of astrocyte molecular signalling mechanisms in synaptic functions. We tailored the model for future large-scale simulations of mouse somatosensory cortex functions *in vivo*. The work utilizes our previous work in computational modelling of astrocytic and neuronal mechanisms in excitability, plasticity and long-term potentiation/depression (Manninen et al. 2010, Toivari et al. 2011, Hituri and Linne 2013, Linne and Jalonen 2014, Saudargiene and Graham, 2015). The ANI model incorporates key biochemical and biophysical mechanisms for the tripartite synapse, in other words for the pre- and postsynaptic elements of the synapse and the astrocyte. The model is utilized to simulate the time courses of various events, in response to various stimulation paradigms including plasticity paradigms, in postsynaptic terminal, and time courses of subsequent induction of increased calcium oscillations in astrocytes. In addition, the ANI model is planned to be used for the following purposes within the HBP:

- 1) develop systematic algorithms in order to reduce the complexity of models, in collaboration with SP4 tasks and external collaborators,
- 2) develop computing principles of astrocytes, in collaboration with SP4 tasks,
- 3) develop neuromorphic chips, inspired by the ANI model, in collaboration with SP9 (the first T434-SP9 workshop was organized in Nov 2015),
- 4) guide production of additional validation data on calcium transients in postsynaptic terminal and astrocyte in specific learning paradigms in SP1 and SP2. Several researchers are contacted through contacts organized by Prof. Katrin Amunds (SP2),
- 5) integrate the developed ANI model as part of the molecular models of neuro-vascular-glia coupling, developed by Dan Keller (T6.4.3.). First meetings and workshops were held in 2015: Geneva May 2015 and EITN Paris Dec 2015.
- 6) promote advanced use of The Simulation Platform, in collaboration with SP5 and SP6, to validate ANI model using additional validation data (obtained in item 4). First meetings and workshops were held in 2015: Geneva May 2015 and EITN Paris Dec 2015,



7) integrate the ANI model as part of the somatosensory cortex simulation, in collaboration with SP6.

We received funding in the competitive call in 2013. T434 started in M7 with institutional funding from TUT, and continued in M14 with HBP funding.

2.9.1.2 Model Description: Astrocyte-neuron interaction model (ANI) model

The astrocyte-neuron interaction (ANI) model is a detailed biophysical excitability model that involves the astrocyte, presynaptic and the postsynaptic terminals of the so-called tripartite synapse. The model describes key cell membrane, as well as 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 use descriptions of voltage-gated conductance in order to better assess the time scales of events associated with cell membrane level events and events that occur inside the terminals. Moreover, the model includes key interactions between postsynaptic terminal and the astrocyte. Studies related to the selection of model components and reproducibility issues of T434 have already been accepted for publication in *Manninen et al. 2016*. Altogether 61 astrocyte models have been published through the year 2014 and our recent work (*Manninen et al. 2016*), with detailed analysis of model characteristics, provided us important insights into the development of the ANI model.

Our hypothesis for the present ANI model is that a retrograde signalling mechanism, such as the one provided by endocannabinoids, is required in addition to neuro- and gliotransmission to fully understand synaptic information transfer in a cortical synapse under study. One of the few models that describe both pre- and postsynaptic signalling, including the release of endocannabinoids, is the study by Zachariou et al. (2013). Moreover, the only existing astrocyte model that can take endocannabinoid as input is by Wade et al. (2012). These two modelling projects were used as the basis of the new ANI model which is extended beyond to incorporate detailed cell membrane level mechanisms. For details of model selection see *Manninen et al. 2016*).

In short, we implemented the following mechanisms of the tripartite synapse:

Presynaptic terminal: **i)** conventional cell membrane ionic currents (transient sodium current, delayed rectifier potassium current), **ii)** fraction of willing calcium channels, **iii)** fraction of bound G proteins (cannabinoid receptors).

Postsynaptic terminal: **i)** conventional cell membrane ionic currents (sodium current, delayed rectifier type potassium current), **ii)** L-type calcium current, **iii)** AMPA receptor current, **iv)** PMCA pump (plasma membrane Ca²⁺-ATPase), **v)** intracellular calcium buffer, and **vi)** SERCA pump (sarco/endoplasmic reticulum Ca²⁺-ATPase).

Astrocyte terminal: **i)** IP₃ receptor channel, **ii)** SERCA pump (sarco/endoplasmic reticulum Ca²⁺-ATPase), and **iii)** leak current into the cytoplasm from the endoplasmic reticulum (ER).

An example of the equations used for the astrocyte terminal is as follows:

Cytosolic IP₃ concentration in the astrocyte (Wade et al. 2011):

$$\frac{d[IP_3]_{astro}}{dt} = \frac{[IP_3^*]_{astro} - [IP_3]_{astro}}{\tau_{IP_3 astro}}$$

Cytosolic calcium concentration in the astrocyte (Li and Rinzel 1994, Wade et al. 2011):

$$\frac{d[Ca^{2+}]_{astro}}{dt} = J_{IP_3}^{astro} - J_{SERCA}^{astro} + J_{leakER}^{astro}$$

Calcium flux through the IP₃R (Li and Rinzel 1994, Volman et al. 2007, Wade et al. 2011):

$$J_{IP3}^{astro} = \bar{p}_{IP3_astro} m_{\infty IP3_astro}^3 n_{\infty IP3_astro}^3 h_{\infty IP3_astro}^3 \left([Ca^{2+}]_{total_free_astro} - (1 + c_1)[Ca^{2+}]_{astro} \right).$$

IP₃R gating variables (Li and Rinzel 1994, Wade et al. 2011):

$$m_{\infty IP3_astro} = \frac{IP3_astro}{IP3_astro + k_{mIP3_astro}},$$

$$n_{\infty IP3_astro} = \frac{[Ca^{2+}]_{astro}}{[Ca^{2+}]_{astro} + k_{nIP3_astro}},$$

$$h_{\infty IP3_astro} = \frac{Q2_astro}{Q2_astro + [Ca^{2+}]_{astro}},$$

$$Q2_astro = d_{2_astro} \frac{IP3_astro + d_{1_astro}}{IP3_astro + d_{3_astro}}$$

$$\tau_{h_astro} = \frac{1}{a_{2_astro} (Q2_astro + [Ca^{2+}]_{astro})},$$

$$\frac{dh_{IP3_astro}}{dt} = \frac{h_{\infty IP3_astro} - h_{IP3_astro}}{\tau_{h_astro}}.$$

The SERCA pump (Li and Rinzel, 1994, Wade et al. 2011):

$$J_{SERCA}^{astro} = V_{\max_SERCA_astro} \frac{[Ca^{2+}]_{astro}^2}{[Ca^{2+}]_{astro}^2 + k_{SERCA_astro}^2}.$$

Leak from ER (Li and Rinzel 1994, Wade et al. 2011):

$$J_{leakER}^{astro} = \bar{p}_{leakER_astro} \left([Ca^{2+}]_{total_free_astro} - (1 + c_1)[Ca^{2+}]_{astro} \right).$$

\bar{p}_{leakER_astro} adjusted such that net calcium flux from the ER is zero at the resting calcium concentration (Blackwell 2012).

Each terminal and their submodules were first validated separately. The astrocyte-neuron equations provided by Wade et al. (2012) did not match the simulation results provided in their original publication. However, since we were only interested in the astrocyte terminal of Wade et al. (2012) model, a part that is commonly known as Li-Rinzel model (Li and Rinzel, 1994), we can strongly assume that the equations are exceptionally well validated. Additional validation of astrocyte responses were done based on results of Shigetomi et al. (2010). Equations taken from Zachariou et al. (2013) 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. To validate the postsynaptic calcium transients we used data from Larkum et al. 2003 and Kaiser et al. 2004. Additional validation data was produced by T434 for the presynaptic vesicle release (**Teppola et al. 2015**). There exist very few datasets to validate the model equations for intracellular signalling, but we partly used our previous modelling work to validate the models (Manninen et al. 2010, Toivari et al. 2011, Hituri and Linne, 2013).

For details of the model, see the code available in EITN repositories (file name: T434_D464_ANImodel.m, which can be run in Matlab version R2010b to obtain the simulation results presented in Results section of this document).



2.9.1.3 Model Data

- **The group developing the model and group's expertise:** The task has been carried out by two computational modellers, Tiina Manninen (TUT, Finland; postdoctoral researcher) and Ausra Saudargiene (Vytautas Magnus University, Lithuania, collaborator of T434), Marja-Leena Linne (TUT, Finland) is responsible, and Riikka Havela (TUT; PhD student) provided expertise on biological data and computational models. Heidi Teppola (TUT; PhD student) provided additional biological expertise and Mikko Lehtimäki (TUT; MSc student of Prof. Linne) in model simplification/reduction. Rolandas Stonkus (Vilnius University, Lithuania; MSc student of Prof. Saudargiene) has additionally started work towards linking models developed in T434 to neuromorphic engineering in SP9. The group has the following expertise/ background education: Manninen (theoretical and computational modelling in neuroscience/ mathematics), Saudargiene (theoretical and computational modelling in neuroscience/ informatics), Linne (theoretical and computational modelling in neuroscience/ electrical engineering), Havela (computational modelling in neuroscience/ biochemistry and medicine), Teppola (electrophysiology/ biochemistry), and Lehtimäki (signal processing/computational biology).
- **Data, algorithms, tools and methodologies storage location(s):** The Matlab code of the ANI model, as well as the publications produced by T434, are stored in the UNIC owncloud database.
- **Description of algorithms/models/principles:**
 - Model format: Matlab (use Matlab version R2010b to obtain the simulation results presented in Results section of this document).
 - Name of DICs: 'Astrocyte neuron interaction model'.
- **Description of data:**
 - **Species, sex, age, number of specimen/subjects:** juvenile (appr. P16-21) rat and mouse data is used to validate the model.
 - **Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus:** Scale: cellular and molecular level computational models, as well as molecular reaction data and physiological data, is used to build and validate the model.
- **Completeness of data/algorithms/models:**
 - 1) The models delivered at M30 are as anticipated. The here produced ANI model is a completely new approach and goes beyond the existing approaches. Since we also started simplification/reduction of models we exceed the anticipated results.
 - 2) The models produced by the whole scientific community during M1-30 are presented and analysed in detail by T434 and results are published in *Manninen et al. 2016* (in press). In short, the research on astrocytes and on astrocyte-neuron signaling has resulted in several attempts to computationally model astrocytes' roles in synaptic and neuronal network dynamics. We evaluated existing computational models developed for astrocyte functions and for astrocyte-neuron interactions. To the best of our knowledge, this study is the first to report detailed evaluation of astrocyte-neuron models. Models were characterized in detail to help their utilization in the future work. Based on our evaluation, we realized that the same component models were often used repeatedly in successive publications, with no spelled-out, explicit justification on how suitable the specific component model was for the study in question. Thus, most models describing the subcellular events of astrocytes had



limitations and shortcomings. A model typically describes a limited set of molecular mechanisms and provides explanations to a relatively narrow set of astroglial dynamics.

We evaluated some of the models also by implementing the equations according to what was presented in the original publication, with an aim to reproduce the simulation results of the original publication. We discovered that many publications lack crucial details in how the models were presented, preventing the implementation of the models without first contacting the authors. Moreover, in some cases it was impossible, after several trials, to reproduce the simulated results of the original publication. This meant often that the equations and simulation results of the original publication did not match.

Please see for more details the accepted manuscript (*Manninen et al. 2016*, in press).

As a conclusion, the here produced ANI model is a completely new approach and goes beyond the existing approaches.

- **Outline state of validation work:**

The three terminals of the ANI model are validated with several sets of data obtained from the literature. Validation data is described in Model Description. We are in the process of establishing better data than what we have now on calcium dynamics in astrocyte processes, but this data does not exist in large quantities/is difficult to obtain. We are in the process of establishing collaborations with SP1 and SP2 and several external collaborators.

- **Data Quality and Value:**

- Verification of data quality: We have used several different sources (different research groups) of validation data.
- Subjective analysis of the value of the data/algorithms for the users: We expect that the value of the ANI model will be large for the following reasons: the model is one of the first attempts to model the biophysical details of the synapse including the astrocyte component, the model is tuned to a specific brain region, and the model is described to the best of our capability (we define set of criteria how to define astrocyte-neuron models in our accepted T434 publication, *Manninen et al. 2016*).

- **Data/algorithm/model usage to date:**

a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group): DIC 'Astrocyte-neuron interaction model', there is a plan that SP6 (Daniel Keller et al) and T423 (Gruning) will use the model.

b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models: T422 (Linne) of SGA1 will develop network models involving influences from astrocytes and the model produced by T434 of Ramp-Up will be used as a core model. In SGA1, also SP6 (Daniel Keller et al) and T433 (Andre Gruning) will utilize the model.

- **Are the data/algorithms/models considered final:**

Models are final, but we may still replace some submodels of the postsynaptic terminal with more state-of-the-art submodels. These submodels are currently under validation in the group. Submodels describing the astrocyte terminal are final.

- **Publications connected to the gathered data:**

Model construction:



Kim et al., 2013: For the postsynaptic part, we took the detailed biochemical reactions for activation of metabotropic glutamate receptors (mGluRs) and endocannabinoid production (2-arachidonyl glycerol (2AG)) from the model by Kim et al., 2013. In more detail, the reactions included glutamate binding to mGluRs, which then activates G proteins. Calcium (Ca), with or without the activated G protein, then binds and activates phospholipase C (PLC), which produces diacylglycerol (DAG) and inositol trisphosphate (IP3). DAG lipase then converts DAG into 2AG. Furthermore, DAG and Ca activate protein kinase C.

Zachariou et al., 2013: To make the postsynaptic model more detailed, we decided to add the equations for postsynaptic plasma membrane Ca-ATPase (PMCA) pumps and diacylglycerol (DAG) degradation from the model by Zachariou et al., 2013. In addition, we modified the L-type Ca channel model by Zachariou et al. (2013) to fit better our purposes.

Wade et al., 2012 and Li and Rinzel, 1994: We took the equations for astrocytic IP3, Ca, and fraction of activated IP3 receptors (IP3Rs) from the model by Wade et al., 2012. Basically the equations for astrocytic Ca and fraction of activated IP3Rs originated from the work of Li and Rinzel, 1994. Astrocytic Ca concentration depends on IP3 and Ca -dependent Ca release, sarco/endoplasmic reticulum Ca-ATPase (SERCA pump), and leak current into the cytoplasm from the endoplasmic reticulum. The fraction of activated IP3Rs depend on Ca and IP3. IP3 concentration depends on postsynaptic 2AG concentration from the model by Kim et al., 2013.

T. Manninen, R. Havela, and M.-L. Linne (2016). Computational models of astrocytes and astrocyte-neuron interactions: Categorization, analysis, and future perspectives. Accepted to Computational Glioscience: One of the goals of T434 was to evaluate existing models. This publication presents the characteristics of all astrocyte and astrocyte-neuron models and shows the reproducibility of the original simulation results for five models. The publication serves as an important background material to produce the ANI model in T434.

Model validation:

Kaiser et al. 2003: Data used to validate the calcium transients in the postsynaptic terminal.

Larkum et al. 2004: Data used to validate the calcium transients in the postsynaptic terminal.

H. Teppola, R. Sarkanen, T.O. Jalonen, and M.-L. Linne (2015). Morphological Differentiation Towards Neuronal Phenotype of SH-SY5Y Neuroblastoma Cells by Estradiol, Retinoic Acid and Cholesterol. *Neurochem Res.* 2015 Oct 30: The publication quantifies vesicle release in neuroblastoma cells and makes comparisons to astrocytic and other neuronal cells. The paper is used as background material in T434.

2.9.1.4 Model Results

The ANI model was simulated to assess a variety of calcium-related signalling mechanisms in a tripartite synapse. In the present work, we use postsynaptic stimulation with varying frequencies to activate the model and to illustrate its behaviour. We conclude that synaptic stimulation induces endocannabinoid production and delayed astrocytic calcium responses (Figure 59). We also conclude that the volume of the postsynaptic terminal (equivalent to spine) significantly affects the astrocytic calcium responses (Figure 60).



Postsynaptic calcium and 2-AG and astrocytic calcium

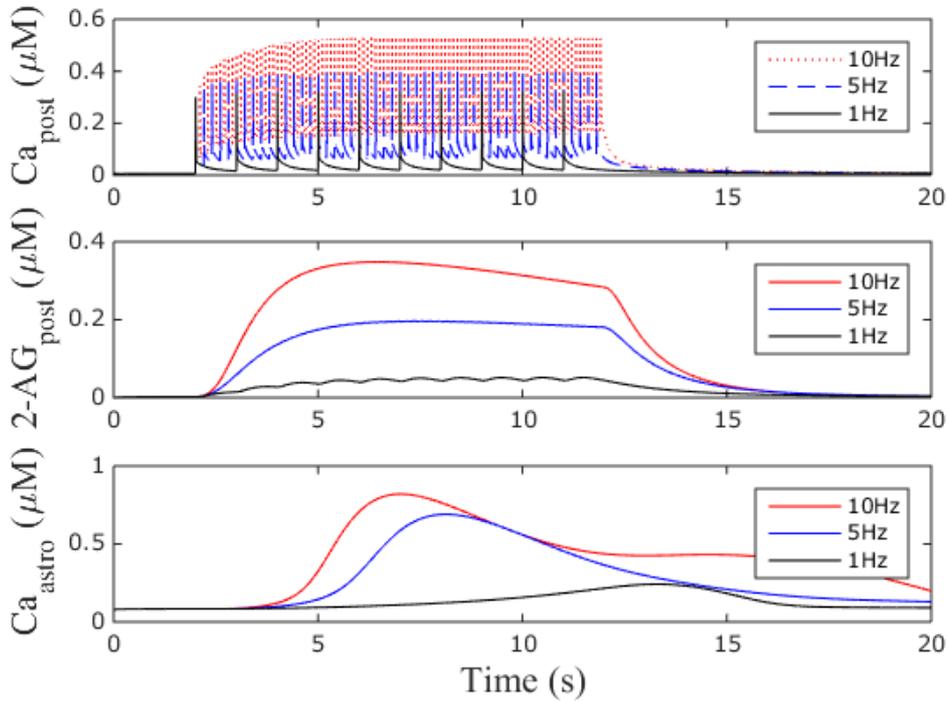


Figure 59: Simulation of the full ANI model by varying the frequency of the stimuli.

Top) Postsynaptic calcium levels with 1, 5, and 10 Hz stimulation. Middle) Endocannabinoid (2-AG) production with 1, 5, and 10 Hz stimulation. Bottom) Astrocytic calcium responses with 1, 5 and 10 Hz stimulation. The results show that 5 Hz stimulation is required to induce marked changes in the intracellular calcium concentration of the model astrocyte. Similar changes in astrocyte calcium concentration have been observed in vivo in response to stimulation of astrocytes.

Postsynaptic calcium and 2-AG and astrocytic calcium

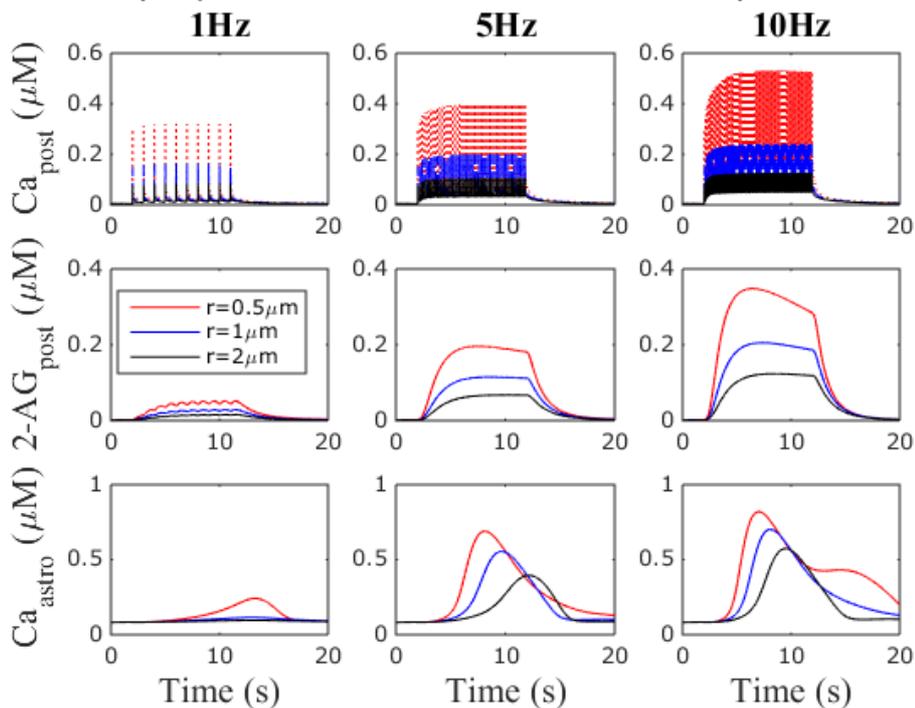




Figure 60: Simulation of the full ANI model by varying the frequency of the stimuli and the volume of the postsynaptic terminal (equivalent to spine).

Increase in spine volume (increase in the radius, defining the volume of the postsynaptic terminal, from 0.5 μm to 2 μm) causes significant changes in the astrocytic calcium responses. Substantial decrease in the amplitude of the calcium response is detected, together with a marked delay in the activation of the astrocytic response, in comparison to original 0.5 μm radius.

The model serves as a tool to test several other hypotheses stemming from the experimental community, including how transmission of information between astrocytes and neurons at synapses affects the neural network function in the cerebral cortex. Moreover, it helps in clarifying 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., Agulhon et al. 2010, Navarrete et al. 2012). The time scales of the molecular signalling mechanism inherently present in the model provide a step towards bridging scales in the future neural network models that incorporate both the neuronal and astrocytic influences.

We are also developing systematic methods to reduce the complexity of models. Model order reduction of molecular signalling mechanisms advances development of neuromorphic chips for astrocytes.

In summary, towards the end of the Ramp-Up Phase the TUT Partner has developed the following models, methods and tools:

- 1) We have provided guidelines how the astrocyte models should be described and presented. One publication is accepted.
- 2) We have developed a detailed biophysical model of the astrocyte-neuron interactions. Manuscript in preparation.
- 3) We are at the midst of applying new model order reduction methods that preserve, very closely, the calcium dynamics observed in the full (non-reduced) ANI model. Manuscript in preparation.

In addition, T434 organized The Neuron-Glia Workshop at the European Institute of Theoretical Neuroscience (EITN) in 2015, in order to interact with the scientific community outside HBP. The workshop greatly helped to sort out relevant validation data for the benefit of T434 goals.

2.9.1.5 Provenance

The ANI model is yet a theoretical model which will be used to test various predictions in the field of glioscience. We have mainly used data from literature which we report carefully.

We are currently negotiating with SP1 and SP2, initiated by Prof. Katrin Amunds (SP2), to initiate collaboration within the HBP. External collaborators are obtained through workshops organized at EITN and they include: Prof. Joachim Luebke (Julich; additional validation data on calcium transients in the postsynaptic terminal), Prof. Vladimir Parpura (University of Alabama at Birmingham, USA; calcium concentrations and calcium transients in astrocytes in vivo), and Prof. Cendra Agulhon (Paris Descartes University, France; calcium transients in astrocytes in vitro). Moreover, we are negotiating with Prof. Pierre Magistretti and Dr. Corrado Cali (T6.4.3.) on reconstructions of astrocyte morphologies. We expect to make all this data that is used to validate the model available. This would allow to track back the origin of data used for model construction and validation.

2.10 Task 4.4.1 Principles of computation in single neurons and cortical microcircuits



2.10.1 *W. Maass (TUGRAZ)_ Brain-Inspired Computing Principles for Implementation in Neuromorphic Computing Systems*

2.10.1.1 Introduction

This model addresses computational properties that emerge in cortical microcircuits through STDP and rewiring (spine dynamics). It also addresses the robustness of these emergent computational properties with regard to network lesions. Details of the model can be found in two publications:

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Synaptic sampling: A Bayesian approach to neural network plasticity and rewiring. Proc. of NIPS 2015: Advances in Neural Information Processing Systems, 2015.

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Network plasticity as Bayesian inference. PLOS Computational Biology, 11(11):e1004485, 2015.

This model can readily be transferred to neuromorphic hardware developed in SP9, since it employs neuron models and microcircuit motifs that have already been implemented in neuromorphic hardware.

2.10.1.2 Model Description

This model was motivated by preceding work in neuroscience, where it was shown that networks of neurons do not typically assume a fixed setting of parameters and connections, but remain to some degree plastic even in the adult brain [Grashow, 2010, Marder, 2011, Marder and Taylor, 2011]. It provides a demonstration and mathematical insight into a resulting new perspective of network learning, where one assumes that the high-dimensional parameter vector θ of a biological neural network tends to move around within some low-dimensional manifold of the parameter space where all parameter vectors provide stable network function (degeneracy).

The model proposes that this low-dimensional manifold is a manifestation of Bayesian learning: A biological neural network \mathcal{N} strives to find parameters θ that

- (1) satisfy a set of constraints (such as genetically encoded connectivity rules) mathematically modelled by a prior distribution $p_s(\theta)$, and
- (2), simultaneously maximize the fit between sensory experiences \mathbf{x} from the outside world and the network's internal representation, quantified by the likelihood function $p_{\mathcal{N}}(\mathbf{x}|\theta)$.

Bayes rule allows us to make this learning goal precise:

Learn the posterior distribution $p^*(\theta|\mathbf{x})$ defined up to normalization by (1)

$$p_s(\theta) p_{\mathcal{N}}(\mathbf{x}|\theta) .$$

The posterior distribution Eq. (1) is not explicitly represented by any neural code in this model but implicitly in the stationary distribution of the permanently changing network parameters θ . The model demonstrates that stochastic local plasticity rules for the parameters θ_i enable a network to achieve the learning goal (1): The distribution of network parameters θ will converge after a while to the posterior distribution $p^*(\theta|\mathbf{x})$ - and produce samples from it - if each network parameter θ_i obeys the dynamics

$$d\theta_i = b \left(\frac{\partial}{\partial \theta_i} p_s(\theta) + \frac{\partial}{\partial \theta_i} \log p_{\mathcal{N}}(\mathbf{x}|\theta) \right) dt + \sqrt{2b} dW_i , \quad (2)$$

where the learning rate $b > 0$ controls the speed of the parameter dynamics and dW_i is a stochastic term that describes infinitesimal stochastic increments and decrements of a Wiener process W_i (see Kappel et al. 2015 for details). We refer to this model for the temporal evolution of network parameters according to Eq. (2) as *synaptic sampling*.



The underlying continuous stochastic fluctuation dW_i provides in this model a driving force that automatically moves network parameters (with high probability) to a functionally more attractive regime when the current solution performs worse because of perturbations, such as lesions of neurons or network connections. We demonstrate this inherent compensation capability in Fig. 1 for a generative spiking neural network with synaptic parameters θ that regulate simultaneously structural plasticity and synaptic plasticity (dynamics of weights).

2.10.1.3 Spike-based Winner-Take-All network model

Network neurons of the model network shown in Fig. 1A were modelled as stochastic spike response neurons with a firing rate that depends exponentially on the membrane voltage [Jolivet et al. 2006, Mensi et al. 2011].

The membrane potential $u_k(t)$ of neuron k is given by

$$u_k(t) = \sum_i w_{ki} x_i(t) + \beta_k(t), \quad (3)$$

where $x_i(t)$ denotes the (unweighted) input from input neuron i , w_{ki} denotes the efficacy of the synapse from input neuron i , and $\beta_k(t)$ denotes a homeostatic adaptation current (see below). The input $x_i(t)$ models the influence of additive excitatory postsynaptic potentials (EPSPs) on the membrane potential of the neuron. Let $t_i^{(1)}, t_i^{(2)}, \dots$ denote the spike times of input neuron i . Then, $x_i(t)$ is given by

$$x_i(t) = \sum_f \epsilon(t - t_i^{(f)}), \quad (4)$$

where ϵ is the response kernel for synaptic input, i.e., the shape of the EPSP, that had a double-exponential form in our simulations:

$$\epsilon(s) = \theta(s) \left(e^{\frac{s}{\tau_r}} - e^{\frac{s}{\tau_f}} \right), \quad (5)$$

with the rise-time constant $\tau_r = 2ms$, and the time constant of the falling edge $\tau_f = 20ms$. $\theta(s)$ denotes the Heaviside step function.

The network neurons were subject to a soft winner-take-all (WTA) mechanism that normalizes the network activity (see below). The instantaneous firing rate $\rho_k(t)$ of network neuron k depends exponentially on the membrane potential and is subject to divisive lateral inhibition $I_{lat}(t)$:

$$\rho_k(t) = \frac{\rho_{net}}{I_{lat}(t)} \exp(u_k(t)), \quad (6)$$

where $\rho_{net} = 100Hz$ scales the firing rate of all neurons. Such exponential relationship between the membrane potential and the firing rate has been proposed as a good approximation to the firing properties of cortical pyramidal neurons [Jolivet et al. 2006]. Spike trains were then drawn from independent Poisson processes with instantaneous rate $\rho_k(t)$ for each neuron. We denote the resulting spike train of the k th neuron at time t by $S_k(t)$.

Lateral inhibition. Divisive inhibition between the neurons in the WTA network was implemented in an idealized form [Nessler et al. 2013]

$$I_{lat}(t) = \sum_{l=1}^K \exp(u_l(t)). \quad (8)$$

This form of lateral inhibition, that assumes an idealized access to neuronal membrane potentials, has been shown to implement a well-defined generative network model [Nessler et al. 2013].

Homeostatic adaptation current. Each output spike caused a slow depressing current, giving rise to the adaptation current $\beta_k(t)$. This implements a slow homeostatic mechanism



that regulates the output rate of individual neurons (see Habenschuss et al. 2012 for details). It was implemented as

$$\beta_k(t) = \gamma \sum_f K(t - t_k^{(f)}), \quad (7)$$

where $t_k^{(f)}$ denotes the f th spike of neuron k and K is an adaptation kernel that was modelled as a double-exponential (Eq. 5) with time constants $\tau_r = 12s$ and $\tau_f = 30s$. The scaling parameter γ was set to $\gamma = -8$.

2.10.1.4 Synaptic sampling in spike-based Winner-Take-All networks as stochastic STDP

The synaptic sampling framework outlined in Sec. **Erreur ! Source du renvoi introuvable.** s used here to model automatic adaptation and rewiring in recurrent networks. We used a single synaptic parameter θ_{ki} per synapse. The synaptic efficacy w_{ki} of a synapse was given by

$$w_{ki} = \exp(\theta_{ki} - \theta_0), \quad (9)$$

where θ_0 is a parameter that scales the synaptic weights. To model network rewiring we interpret synapses with negative synaptic parameters θ_{ki} as disconnected (see Kappel et al. 2015, for details).

To arrive at a concrete learning scheme in terms of synaptic sampling (Eq. 2) we define the probabilistic model $p_{\mathcal{N}}(\mathbf{x}|\boldsymbol{\theta})$ that underlies the WTA network. It has been shown in [Habenschuss et al. 2013] that a WTA-network of the form given above implicitly defines a generative mode. In this generative model, inputs $x_i(t)$ are assumed to be generated depending on the value of a hidden multinomial random variable encoded by the activity of the WTA circuit.

The network implements inference in this generative model, i.e., for a given input $x_i(t)$, the firing rate $\rho_k(t)$ of network neuron k is proportional to the posterior probability of the corresponding hidden cause (see Kappel et al. 2015 for details). An online maximum likelihood learning rule for this generative model was derived in [Habenschuss et al. 2013]. It changes synaptic weights according to

$$\Delta w_{ki}(t) = S_k(t)(x_i(t) - \alpha e^{w_{ki}}), \quad (10)$$

where $S_k(t)$ denotes the spike train of the postsynaptic neuron and $x_i(t)$ denotes the weight-normalized value of the sum of EPSPs from presynaptic neuron i at time t .

To define the synaptic sampling learning rule Eq. (2), we also need to define the parameter prior. In our experiments, we used a simple Gaussian prior on each parameter with mean $\mu = 0.5$ and $\sigma = 1$. The derivative of the log-prior is given by

$$\frac{\partial}{\partial \theta_{ki}} \log p_S(\boldsymbol{\theta}) = \frac{1}{\sigma^2} (\mu - \theta_{ki}). \quad (11)$$

Inserting Eq. (10) and Eq. (11) into the general form (2), we find that the synaptic sampling rule is given by

$$d\theta_{ki} = b \left(\frac{1}{\sigma^2} (\mu - \theta_{ki}) + N w_{ki} S_k(t) (x_i(t) - \alpha e^{w_{ki}}) \right) dt + \sqrt{2b} dW_{ki}, \quad (12)$$

see [Kappel et al. 2015] for a detailed derivation.

2.10.1.5 Simulation and data analysis

Computer simulations of spiking neural networks were done in Matlab 2011b (Mathworks). In all spiking neural network simulations, synaptic weights were updated according to the rule Eq. (12) with parameters $N = 100$, $\alpha = e^{-2}$, and $b = 10^{-4}$. In the simulations, we directly implemented the time-continuous evolution of the network parameters in an



event-based update scheme. Before learning, initial synaptic parameters were independently drawn from the prior distribution $p_S(\theta)$.

For the mapping Eq. (9) from synaptic parameters θ_{ki} to synaptic efficacies w_{ki} , we used as offset $\theta_0 = 3$. This results in synaptic weights that shrink to small values (< 0.05) when synaptic parameters are below zero (disconnected synapses). In the simulation, we clipped the synaptic weights to zero for negative synaptic parameters θ_{ki} to account for retracted synapses (see Kappel et al. 2015 for details).

Network model. In the network model shown in Fig. 61A we used two recurrently connected ensembles, each consisting of four WTA circuits, with arbitrary excitatory synaptic connections between neurons within the same or different ones of these WTA circuits. The parameters of neuron and synapse dynamics were as described in Sec. 4.1.2.1. All synapses, lateral and feedforward, were subject to the same learning rule Eq. (12). Lateral connections within and between the WTA Circuit neurons were unconstrained (allowing potentially all-to-all connectivity). Connections from input neurons were constraint as shown in Fig. 1. The lateral synapses were treated in the same way as synapses from input neurons but had a synaptic delay of 5 ms.

Network Inputs. The spoken digit presentations in Fig. 61 were given by reconstructed cochleagrams of speech samples of isolated spoken digits from the TI 46 dataset (also used in Klampfl and Maass, 2013, and Hopfield and Brody, 2001). Each of the 77 channels of the cochleagrams was represented by 10 afferent neurons, giving a total of 770. Cochleagrams were normalized between 0 Hz and 80 Hz and used to draw individual Poisson spike trains for each afferent neuron. In addition 1 Hz Poisson noise was added on top. We used 10 different utterances of digits 1 and 2 of a single speaker. We selected 7 utterances as training set and 3 for testing. One randomly selected utterance from the training set was presented together with a randomly chosen instance of the corresponding handwritten digit. Handwritten digit images were taken from the MNIST dataset [LeCun et al. 1998]. Each pixel was represented by a single afferent neuron. Gray scale values were scaled to 0 - 50 Hz Poisson input rate and 1 Hz input noise was added on top. These Poisson rates were kept fixed throughout the duration of the corresponding spoken digits. After each digit presentation, a 50 ms window of 1 Hz Poisson noise on all input channels was presented before the next digit was shown.

For test trials in which only the auditory stimulus was presented, the activity of the visual input neurons was set to 1 Hz throughout the whole pattern presentation. The learning rate b was set to zero during these trials. The PETH plots in Fig. 61B were computed over 100 trial responses of the network to the same stimulus class (e.g. presentation of digit 1). Spike trains were filtered with a Gaussian filter with $\sigma = 50$ ms and summed in a time-discrete matrix with 10 ms bin length. Maximum firing times were assigned to the time bin with the highest PETH amplitude for each neuron.

2.10.1.6 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.
- Data, algorithms, tools and methodologies storage location(s) (and links?)
- Description of algorithms/models/principles:
 - Format, language if applicable. Matlab Code (2011b)
 - Name of DICs/software catalogue/or HBP github project entries.
- Description of data:
 - Species, sex, age, number of specimen/subjects.



- Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.
- **Completeness of data/algorithms/models:**
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?
 - Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- **Outline state of validation work.**
- **Data Quality and Value:**
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.
- **Data/algorithm/model usage to date:**
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.
- **Are the data/algorithms/models considered final?**
- **Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)**

2.10.1.7 Model Results

This model supports the study of the self-organization of connections and synaptic weights between hidden neurons, and resulting emergent computational properties. It addresses multi-modal sensory integration and association in a simplified manner (see Fig. 61A). Two populations of “auditory” and “visual” input neurons x_A and x_V project onto corresponding populations z_A and z_V of hidden neurons. Each of the two populations z_A and z_V consist of four WTA circuits (as described above) and receives exclusively auditory or visual inputs. In addition, arbitrary lateral excitatory connections between these “hidden” neurons are allowed.

The synaptic parameters of this network where subject to learning via the parameter dynamics Eq. (12). One can test emergent associations between the two populations z_A and z_V of hidden neurons by presenting auditory input only and observing the activity of the “visual” hidden neurons. Fig. 61B shows the emergent activity of the neurons z_V when only the auditory stimulus was presented (visual input neurons x_V remained silent). Assemblies of hidden neurons emerge that encode the presented digit (1 or 2). Top panel shows the average activity of all neurons from z_A for stimulus 1 (left) and 2 (right) after learning, when only an auditory stimulus is presented.

One can further demonstrate the generative aspect of the network by reconstructing the visual stimulus from the activity of the “visual” hidden neurons z_V . Fig. 61B shows



reconstructed visual stimuli from a single run where only the auditory stimuli x_A for digits 1 (left) and 2 (right) were presented to the network. Digit images were reconstructed by multiplying the synaptic efficacies of synapses from neurons in x_V to neurons in z_V (which did not receive any input from x_V in this experiment) with the instantaneous firing rates of the corresponding z_V -neurons.

To study the inherent compensation capability of synaptic sampling one can apply lesions to the network and study the compensation for these perturbations. Fig. 61C shows the first three PCA components of a subset of the network parameters for a trial in which two major lesions were applied to the network within a learning session of more than 7 hours. In the first lesion all neurons (16 out of 40) that became tuned for digit 2 in the preceding learning (see Fig. 61B) were removed. The lesion significantly impaired the performance of the network in stimulus reconstruction, but it was able to recover from the lesion after about one hour of continuing network plasticity according to Eq. (12) (see Fig. 61D). The reconstruction performance of the network was measured here continuously through the capability of a linear readout neuron from the visual ensemble to classify the current auditory stimulus (1 or 2).

Insets at the top of Fig. 61D show the synaptic weights of neurons in z_V at 4 time points t_1, \dots, t_4 , projected back to the input. Network diagrams in the middle show ongoing network rewiring for synaptic connections between the hidden neurons z_A and z_V . Each arrow indicates a functional connection between two neurons. To keep the figure uncluttered only subsets of synapses are shown (1% randomly drawn from the total set of possible lateral connections). Connections at time t_2 that were already functional at time t_1 are plotted in gray. The neuron whose parameter vector is tracked in (C) is highlighted in red. The text under the network diagrams shows the total number of functional connections between hidden neurons at the time point.

In the second lesion (transition to green in Fig. 61C,D) all currently existing synaptic connections between neuron in z_A and z_V were removed, and not allowed to regrow. After about two hours of continuing learning 294 new synaptic connections between hidden neurons emerged. These new connections made it again possible to infer the auditory stimulus from the activity of the remaining 24 hidden neurons in the population z_V . The PCA analysis in Fig. 61C shows that after each lesion the network parameters migrate to a new manifold.

Altogether this experiment showed that continuously ongoing synaptic sampling maintains stable network function and automatic self-repair in a recurrent network architecture.

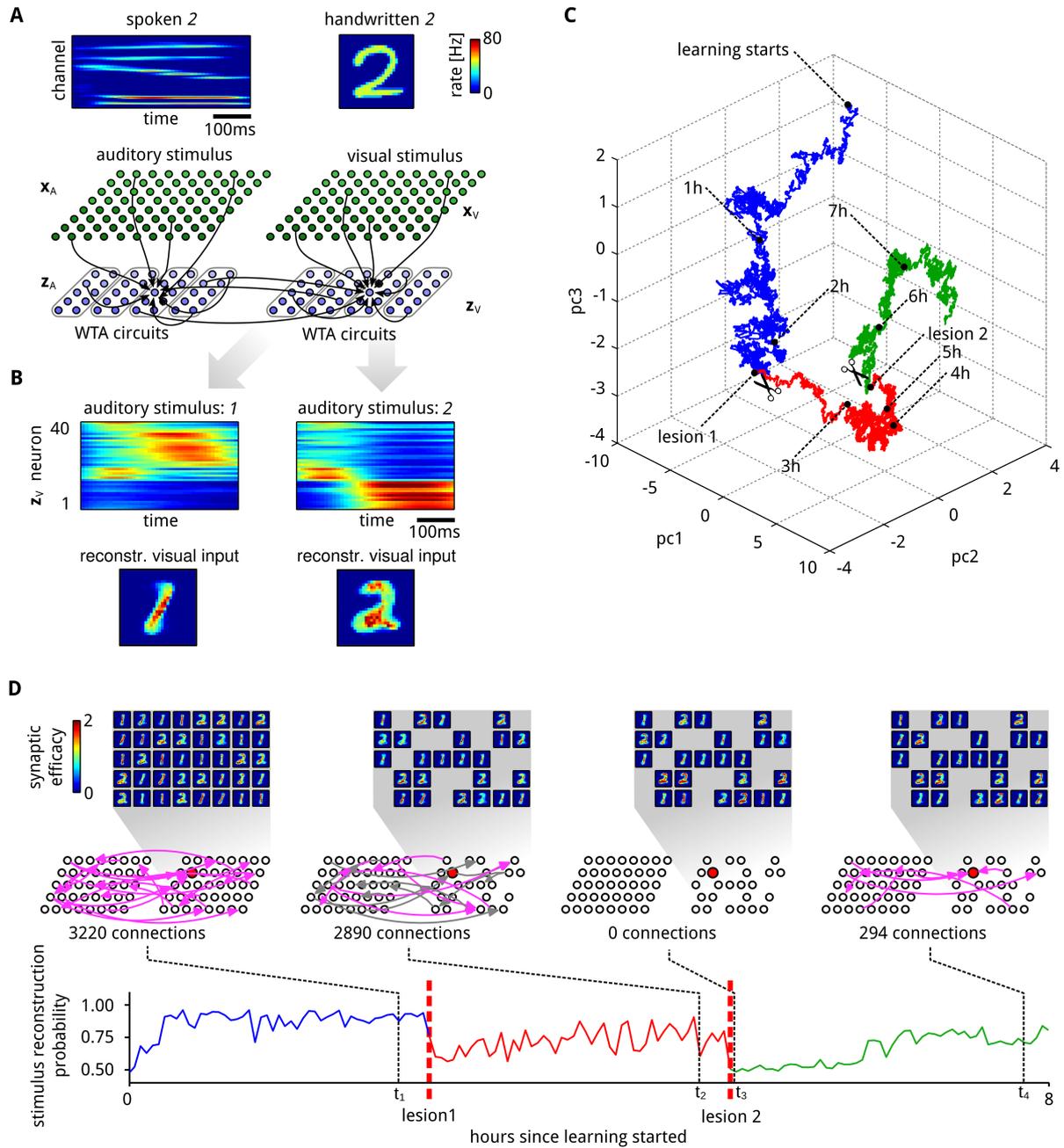


Figure 61 Inherent compensation for network perturbations.

A: The network received simultaneously spoken and handwritten representations of the same digit (1 or 2). Stimulus examples for spoken and written digit 2 are shown at the top. These inputs are presented to the recurrent network through corresponding firing rates of “auditory” (x_A) and “visual” (x_V) input neurons. B: Assemblies of hidden neurons emerge that encode the presented digit (1 or 2). Neurons are sorted by the time of their highest average firing. Although only auditory stimuli are presented, it is possible to reconstruct an internally generated “guessed” visual stimulus that represents the same digit (bottom). C: First three PCA components of the temporal evolution of a subset of the network parameters θ , while two lesions were applied to the network. After each lesion the network parameters θ migrate to a new manifold. D: The generative reconstruction performance of the “visual” neurons z_V for the test case when only an auditory stimulus is presented was tracked throughout the whole learning session, including lesions 1 and 2 (bottom panel). After each lesion the performance strongly degrades, but reliably recovers.

2.10.1.8 Provenance

NA



2.10.2 A. Destexhe (CNRS) – Integrative properties at the single neuron level

The work in this task from CNRS partner is reported in Task 4.1.1 - it consisted in investigating the integrative properties of excitable dendrites and the effect of multiple interacting dendritic spikes.

2.11 Task 4.4.2 Novel computing systems inspired by biology

2.11.1 W. Maass (TUGRAZ)

2.11.1.1 Introduction

This model provides a tool for autonomous learning of working memory, input prediction, and sequence learning through STDP in a generic cortical microcircuit motif: recurrently connected populations of excitatory and inhibitory neurons. This microcircuit motif has already been implemented in neuromorphic hardware in SP9. Hence it provides additional functionality to such neuromorphic hardware. The model shows how networks of spiking neurons in neuromorphic hardware can acquire through STDP important capabilities of Hidden Markov Models (HMMs). HMMs are standard tools in numerous engineering applications, from speech understanding to robot navigation.

The basic principles of this model have been published in

D. Kappel, B. Nessler, and W. Maass. STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning. *PLOS Computational Biology*, 10(3):e1003511, 2014.

Variations and further application paradigms for this model are currently developed in Graz.

2.11.1.2 Model Description

We first define the spiking neural network model for the winner-take-all (WTA) circuit. The network consists of stochastic spiking neurons, which receive excitatory input from an afferent population **and** from lateral excitatory connections from competing neurons in the WTA circuit. To clarify the distinction between these connections, we denote the synaptic efficacies of feedforward and lateral synapses by different weight matrices W and V , respectively. Thus, the membrane potential of neuron k at time t is given by

$$u_k(t) = \sum_i w_{ki} x_i(t) + \sum_j v_{kj} y_j(t) + b_k, \quad (1)$$

where $w_{ki} x_i(t)$ and $v_{kj} y_j(t)$ denote the time courses of the excitatory postsynaptic potentials (EPSP) under the feedforward and lateral synapses respectively, and b_k is a parameter that controls the excitability of the neuron.

Divisive inhibition between the neurons in the WTA network was implemented in an idealized form [Nessler et al. 2013]

$$I_{lat}(t) = \sum_{l=1}^K \exp(u_l(t)). \quad (2)$$

The instantaneous firing rate of neuron k at time t is then given by

$$\rho_k(t) = \frac{\rho_{net}}{I_{lat}(t)} \exp(u_k(t)). \quad (3)$$

This probabilistic formulation allows us to derive parameter estimation algorithms and to compare them with biological mechanisms for synaptic plasticity. In [Kappel et al. 2014] we derive this algorithm for the WTA circuit and show that it evaluates to weight updates that need to be applied whenever neuron k emits a spike at time \hat{t} , according to

$$\Delta w_{ki}(\hat{t}) = \eta (e^{-w_{ki} x_i(\hat{t})} - 1) \quad \text{and} \quad \Delta v_{kj}(\hat{t}) = \eta (e^{-v_{kj} y_j(\hat{t})} - 1), \quad (4)$$



where η is a positive constant that controls the learning rate. Note that the update rules for the feedforward and the recurrent connections are identical, and thus all excitatory synapses in the network are handled uniformly. These plasticity rules are equivalent to the updates that previously emerged as theoretically optimal synaptic weight changes, for learning to recognize repeating high-dimensional patterns in spike trains from afferent neurons, in related studies [Habenschuss et al. 2013, Nessler et al. 2013].

The update rules consist of two parts: A Hebbian long-term potentiating (LTP) part that depends on presynaptic activity and a constant depression term. The dependence on the EPSP time courses makes the first part implicitly dependent on the history of presynaptic spikes. Potentiation is triggered when the postsynaptic neuron fires after the presynaptic neuron. This term is commonly found in synaptic plasticity measured in biological neurons, and for common EPSP windows it closely resembles the shape of the pre-before-post part of standard forms of STDP [Caporale and Dan 2008, Markram et al. 2011].

2.11.1.3 Model Data

- Task(s)/group(s) responsible for generating algorithms/models/principles.
- Data, algorithms, tools and methodologies storage location(s) (and links?)
- Description of algorithms/models/principles:
 - Format, language if applicable: Matlab Code (2011b)
 - Name of DICs/software catalogue/or HBP github project entries.
- Description of data:
 - Species, sex, age, number of specimen/subjects.
 - Scale (brain, brain region, cells, molecules), features (morphology/physiology/expression, etc.), locations, and description of entities, e.g. morphological characterisation of basket cells of the hippocampus.
- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models?
 - Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- Outline state of validation work.
- Data Quality and Value:
 - Verification of data quality.
 - Your subjective analysis of the value of the data/algorithms for the users.
- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2.



- Are the data/algorithms/models considered final?
- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.)

2.11.1.4 Model Results

Computer simulations demonstrate that the WTA circuit described above is able to learn to encode the hidden state that underlies the statistics of input spike trains presented to the circuit via the learning rule Eq. (4). We demonstrate this for a simple working memory task and analyse how the hidden state underlying this task is represented in the network.

The task consisted of three phases: An initial cue phase, a delay phase and a recall phase. Each phase is characterized by a different input sequence, where the cue sequence defines the identity of the recall sequence. We used four cue/recall pairs in this experiment. The structure of this task is illustrated in Fig. 62A. The graph represents a finite state grammar that can be used to generate symbol sequences by following a path from *Start* to *Exit*. On each arc that is passed, the symbol next to the arc is generated, e.g. *AB-delay-ab* is one possible symbolic sequence. The symbolic sequences are presented to the WTA circuit encoded by afferent spike trains. Every symbol *A, B, C, D, a, b, c, d, delay*, is represented by a rate pattern with fixed length of 50ms. The *delay* symbol is unspecific to the sequence and it is thus necessary to memorize the cue phase throughout the delay phase in order to correctly predict the recall phase. We show that this working memory emerges automatically through unsupervised learning in our WTA circuit model, and that it is represented in the form of specialized neural assemblies that encode the input patterns and their temporal context.

To show this, we trained a WTA circuit with 200 afferent cells and 100 circuit neurons by randomly presenting input spike sequences until convergence. In this experiment, the patterns were presented as a continuous stream of input spikes, without intermediate pauses or resetting the network activity at the beginning of the sequences. Training started from random initial weights, and therefore the observation and prediction model had to be learned from the presented spike sequences. Prior to learning the neural activity was unspecific to the patterns and their temporal context (see Fig. 62B). Fig. 62C shows the evoked activities for all four sequences after training. The output of the network is represented by the averaged network activity over 100 trial runs. The neurons are sorted according to the time of highest firing rates. Each sequence is encoded by a different assembly of neurons. This reflects the structure of the working memory that underlies the task. Since the input is presented as continuous spike train, the network has also learned intermediate states that represent a gradual blending between patterns. About 25 neurons were used to encode the information required to represent the hidden state of each sequence.

This coding scheme installs different representations of the patterns depending on the temporal context they appeared in, e.g. the pattern *delay* within the sequence *AB-delay-ab* was represented by another assembly of neurons than the one in the sequence *BA-delay-ba*. Small assemblies of about five neurons became tuned for each pattern and temporal context. This sparse representation emerged through learning and is not merely a consequence of the inherent sparseness of the WTA dynamics. Prior to learning all WTA neurons are broadly tuned and show firing patterns that are unordered and nonspecific (see Fig. 62B). After learning their afferent synapses are tuned for specific input patterns, whereas the temporal contexts in which they appear are encoded in the excitatory lateral synapses (see Fig. 62D). These weights reflect the sparse code and also the sequential order in which the neurons are activated.

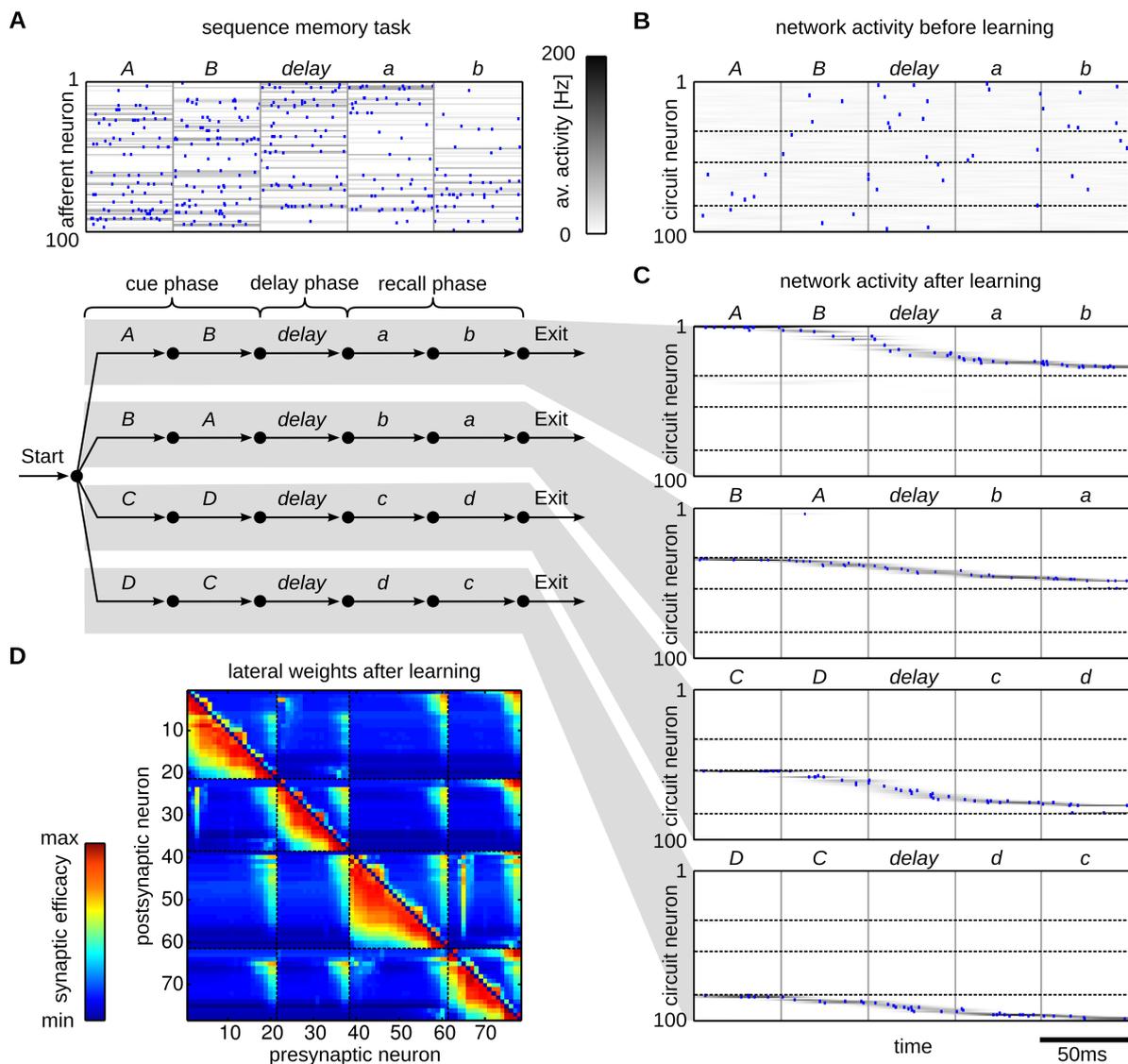


Figure 62: Emergence of working memory encoded in neural assemblies through approximate HMM learning in a WTA circuit through STDP.

A: Illustration of the input encoding for sequence *AB-delay-ab*. The upper plot shows one example input spike train (blue dots) plotted on top of the mean firing rate averaged over 100 trials. The lower panel shows the finite state grammar graph that represents the simple working memory task. The graph can be used to generate symbol sequences by following any path from *Start* to *Exit*. In the first state (*Start*) a random decision is made, which of the four paths to take. On each arc that is passed the symbol next to the arc is emitted (and provided as input to the WTA circuit in the form of a rate pattern). **B,C:** Evoked activity of the WTA circuit for one example input sequence before learning (**B**) and for each of the four sequences after learning (**C**). The network activity is averaged and smoothed over 100 trial runs (grey traces) the blue dots show the spiking activity for one trial run. The neurons are sorted by the time of their highest average activity over all four sequences, after learning. For each sequence a different assembly of neurons becomes active in the WTA circuit. Dotted black lines indicate the boundaries between assemblies. **D:** The lateral weights that emerged through STDP. The neurons are sorted as in (**B,C**). Each neuron has learned to fire after a distinct set of predecessors, which reflects the sequential order of assembly firing. The stochastic switches between sequences are represented by enhanced weights between neurons active at the sequence onsets.

2.11.1.5 Provenance

NA



2.11.2 *J. Dambre (UGent)*

2.11.2.1 Introduction

Throughout the ramp-up phase, UGent has investigated how brain-inspired computing principles can help to generate building blocks of robotic motor control and gait generation. This work is aimed in particular at highly compliant robots, for which detailed analytical dynamical and kinematic models are not easily obtained. In those cases, the controller should autonomously learn how to interact with and control the robot dynamics. Since this learning has to be done with the actual physical robot in the loop, this is only feasible if it requires as little adaptation of the controller as possible, and is therefore likely to converge more rapidly. This is the main reason why UGent has focused on reservoir computing, a brain-inspired computation paradigm in which the desired computation is obtained by optimally exploiting the dynamics that are available in a dynamical system (a reservoir, consisting of neural network and/or physical system). This is achieved by observing a large set of internal dynamical responses of the system and optimally combining them to the desired output signals. The term embodied reservoir computing refers to those cases in which the physical robot's dynamical response is explicitly used as part of the 'reservoir'.

A first contribution of UGent that was reported previously in HBP consists of a reservoir computing mixture-of-experts model that can predict robotic 'proprioceptive' sensor outputs and that was learnt and experimentally validated on a highly compliant tensegrity robot [UGent1]. This model also contains latent variables that can be used to classify, e.g., properties of the terrain the robot was moving on. A second, previously reported contribution of UGent focuses on how embodied motor control and gait generation can be learned in a biologically plausible way, i.e., when the 'fitness' of the controller is evaluated based on external observations of the body's (loco)motion and the individual desired actuator signals for motors or muscles are not known. In this setting, UGent has evaluated the use of noise-based Reward-modulated Hebbian learning to optimise the output signals of reservoir computing in general, as well as for certain embodied motor control tasks involving a highly compliant tensegrity robot [UGent2]. This work has shown that our approaches that are based on reservoir computing using traditional machine learning techniques and supervised learning are transferable to biologically plausible neural systems.

The final contribution of UGent in this context, and the one reported in what follows, involves the demonstration of partially embodied tunable gait generation on the Oncilla compliant quadruped robot. For highly compliant robots, it was previously shown that a reservoir computing approach using only the body as a reservoir is adequate for obtaining embodied gait generation. The morphology of quadrupedal robots has previously been used for sensing and for control of a compliant spine, but never for gait generation. However, our work shows that the less compliant nature of the Oncilla robot gives rise to the need for a partially embodied approach, in which the body needs to be augmented with additional dynamic computation to obtain a suitable reservoir. Part of the aim of this work was to study in which ways the body needs to be augmented, and in particular whether it is lacking in memory or in computational (nonlinear) richness. For this reason, in the present work, the augmentation consists of the combination of explicit memory with an untrained memoryless neural network (extreme learning machine), but similar results could equally be achieved by an untrained recurrent neural network. This work has led to the demonstration of the Oncilla robot walking in a stable way, including tenability of the gait frequency and the walking direction (steering). Although in this case known desired actuator patterns were used, no simulation was involved and the gait generation controllers were learnt directly on the physical robot. This work was published in [UGent3].



The next step in this line of research (SGA1 - T10.4.5) will be the extension to more complex (higher DOF) robots, the conversion to more biologically plausible neural networks and the re-introduction of reward-based biologically plausible learning in the context of the neurorobotics subproject.

2.11.2.2 Model Description : Quadruped Embodied gait generation

In earlier work it was shown that for a tensegrity robot, a linear transformation from the sensor signals to the motor signals was enough to generate stable locomotion. The idea behind this is that the body of the robot itself has computing power, and that this power is being harvested by using it as a reservoir. This embodiment of computation allows the robot to generate stable locomotion without the requirement to explicitly use the state of its compliant elements in a digital control algorithm. On our robot however, we found that we could indeed generate a gait this way, but that it was not stable and did not always return to its limit cycle. In other words, the robot's internal dynamics in response to the environment do not exactly match the required dynamics for stable gait generation. However, the fact that they suffice to generate a close but unstable approximation indicates that the mismatch is not very large. Therefore, we propose to digitally add additional transformations to the sensor signals. In this way, the computations are still partially embodied in the morphology of the robot. In this setting, we want to quantify the minimal complexity of these transformations as expressed by their memory and nonlinear complexity. The next paragraphs describe its parts in more detail.

Linear Transformation

The aim of the linear transformation is to find the $M \times N$ transformation matrix W that optimally maps the $N \times 1$ vector of the N normalized input signals x to the $M \times 1$ vector of the M output signal \hat{y} :

$$\hat{y} = W \cdot x$$

Optimality is defined as the minimisation of the mean squared error (MSE) between the output signals \hat{y} and the target output signals y . This can be achieved by using linear regression. To achieve the right bias, we add a constant signal to the inputs x . This approach is limited to one-shot learning. In order to continue optimizing this relation while running, we will use the recursive least squares (RLS) algorithm, an online method for linear regression.

Adding Non-Linear Dynamics

Between the robot body and the linear transformation, we now add an additional transformation layer in order to increase the richness of signals received by the linear transformation. In order to be able to separately explore the need for memory and nonlinearity, we introduce two separate modules: a nonlinear layer and a memory buffer.

The nonlinear transformations are generated by introducing a hidden layer of H nonlinear neurons, each of which receives a random mixture of the sensor signals (again augmented with a constant bias signal). This technique is known as Extreme Learning Machines (ELM). We initialize all elements in the weight matrices of the ELM by sampling them from the standard normal distribution. These elements are not optimized.

The memory buffer of length B is added to each nonlinearly transformed signal, such that the RLS algorithm obtains direct access to the signals from previous time steps in $x_{rls}(t)$. x_{rls} thus contains all signal values from a small time window in the past. This allows us to explore a further richness of the dynamics which were added by the morphology of the robot. The resulting detailed controller architecture is schematically represented in Figure 63.

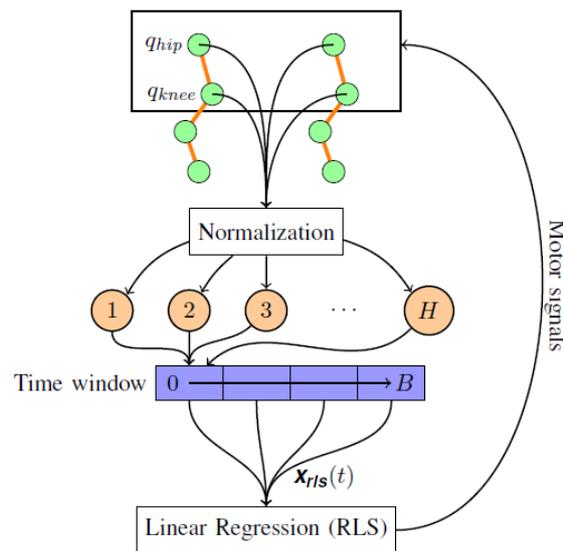


Figure 63: Schematic of the control system.

The signals measured in the motor encoders are first normalized, are then sent through a layer of hidden, untrained neurons, the outputs of these neurons are buffered and a linear transformation is performed on these buffered signals to generate the motor signals. It is only this linear transformation which is learned through a linear regression method (RLS).

Experimental setup

We want our robot to realise a stable gait based on the feedback from the 8 rotary encoders. Using these sensors, the proposed controller must be trained to derive a motor command which is sent to the end-effectors. In sequential operation, this should result in the robot moving with a stable gait. The training procedure outlined in this section enables the controller to discover the relation between the received sensor signals and the output it needs to generate at that moment. As target signals, we use motor signals for stable gaits resulting from previous work. We have the robot trot at a frequency of 1.7 Hz, corresponding to a speed of about 0.76m/s. Before each experiment, the robot runs for 5 seconds with this gait, using the desired signal as input, in order to reach steady state. During these 5 seconds, we measure the mean and the standard deviation of the sensor signals. These are used to normalize the input signals such that they have a mean of 0 and a standard deviation of 1. After normalization, we add Gaussian noise with an amplitude of 0.01 to each input signal for regularization during training.

In the first training phase, the linear combination is optimized using RLS. As a result of this training, the control system finds a relation between the input and the output signals, but it fails to find a stable attractor. Every time the robot has a small error in the output signal, this error is reflected in the input signals of the next time step. Since the controller has never learned to handle those errors, they accumulate and destabilize the attractor. For the controller to learn to deal with its own errors, we add a second training phase, with RLS still active. In this stage, the output signals sent to the motors are mixtures of the target signals and the signals generated by the linear transformation. The fraction of the target signals is reduced over time until it becomes zero. After this phase, the RLS learning is switched off and the resulting gait is evaluated.

We thus split up the learning process into multiple phases:

- 1) The normalization phase: We wait for transient effects caused by starting from standstill to fade out, and when we record the average and variance of each sensor to normalize them. This stage takes 5 s.



2) The first training phase: We send the teacher signal to the motors, and use RLS to learn the relation between these outputs and the normalized inputs from the sensors. This stage takes 10 s, unless noted differently.

3) The second training phase: The motor signals are mixed between the teacher signal and the signals generated by the linear transformation. The RLS-algorithm still updates Wrls. This stage takes 10 s..

4) The running phase: The robot stops optimizing the linear transformation, but continues to run and where we test the stability of the attractor.

2.11.2.3 Model Data

- Responsible T 4.4.2, UGent
- Data, algorithms, tools and methodologies storage location(s) (and links?)

Source code:

<https://www.unic.cnrs-gif.fr/owncloud/index.php/s/204531670df05930ffbec2504376325c/download?path=%2F&files=Oncilla-workstation.tar.gz>

Paper:

https://www.unic.cnrs-gif.fr/owncloud/index.php/s/204531670df05930ffbec2504376325c/download?path=%2F&files=IROS_2015_Jonas.pdf

Video of walking robot:

<https://www.unic.cnrs-gif.fr/owncloud/index.php/s/204531670df05930ffbec2504376325c/download?path=%2F&files=oncilla.wmv>

- Description of algorithms/models/principles:
 - Source code for the experiments: Python (note: this code can only be used in combination with the Oncilla robot)
 - DICC card: “Embodied gait generation quadruped”
- Description of data:
 - Source code for the experiments on embodied gait generation for quadruped Oncilla robot
- Completeness of data/algorithms/models:
 - Completed and published
 - Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community
 - Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it.
- This work was experimentally validated on the physical robot
- Data Quality and Value:
 - Verification of data quality.
 - We expect that the approach followed in this work is portable to other 4- or 6-legged robots. The combination of source code and published paper should suffice to achieve this.



- Data/algorithm/model usage to date:
 - a) Ramp-Up data: DICCC card: “Embodied gait generation quadruped”, only used by UGent, the approach has since been applied to a second and even more compliant quadruped robot.
 - b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models: SP10 - task 10.4.5
- Are the data/algorithms/models considered final? Within their own right: yes, but extensions and improvements will be made in SGA1
- Publication:

“Developing an embodied gait on a compliant quadrupedal robot”,

Jonas Degraeve, Ken Caluwaerts, Joni Dambre and Francis Wyffels, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 4486-4491

2.11.2.4 Model Results

To evaluate our approach, we optimized a controller with $H = 50$ hidden nodes and a time window B of 5 time steps (44 ms or 7.5% of the gait period). For these settings, an attractor was found that generated a stable gait and was able to return to its limit cycle after stopping. The resulting attractor is depicted in Figure 64.

In order to prove that this result is reproducible on different setups, we trained the Oncilla robot to perform a walking gait. For this situation, we again used $H = 50$ hidden nodes, but we had to increase the time window B to 12 time steps (103 ms or 17.5% of the gait period). The resulting attractor is shown in Figure 64. We also needed to increase the length of the first and second training stage to 30 s. The fact that we needed to increase both B and the training time is explained by the increased complexity of a walking gait. In this gait, each leg has a different phase which means that there is less dependence between the motor signals that need to be generated. Therefore, we had to increase the number of inputs to the RLS layer to the maximum we could compute in real time. Additionally, we needed to increase the training time to find the proper relation between the inputs and the outputs.

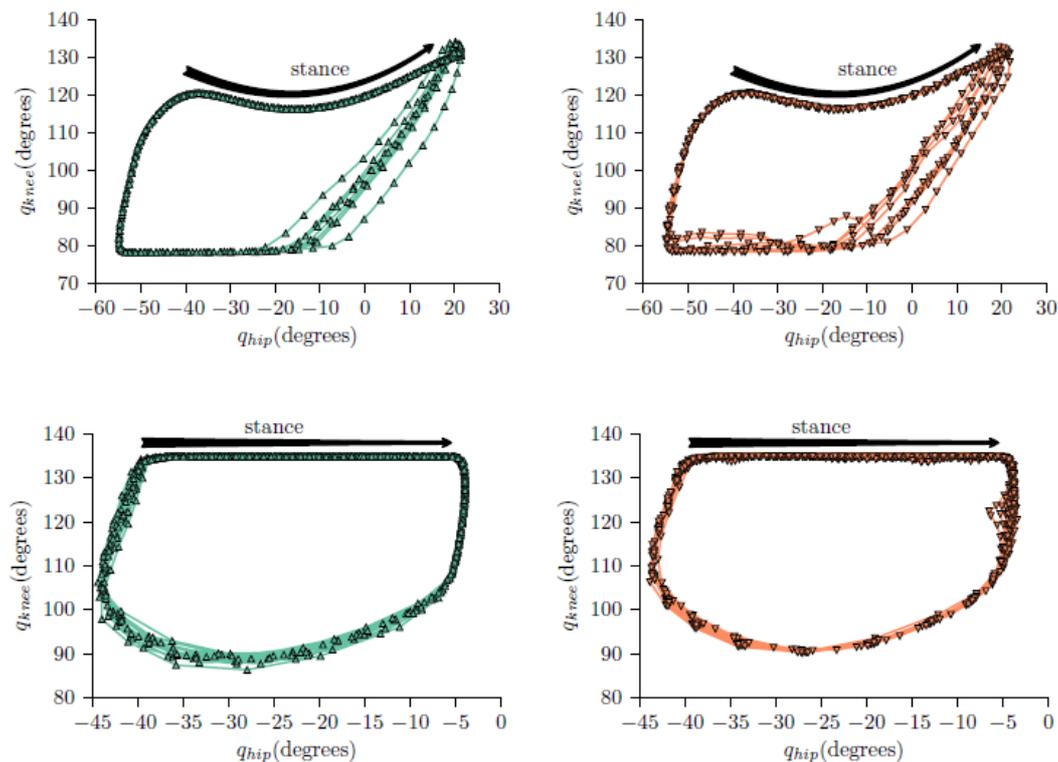


Figure 64 Robot trajectories

On the left, the trajectory of the robot's right hind leg in the joint space is shown while the robot is performing a trotting gait (top) or walking gait (bottom), with q_{hip} and q_{knee} being the angle measured over time by the hip and knee encoder respectively. These were recorded during the first 5 seconds of the first training stage. A higher angle means the leg is moved to the front or the knee is extended.

On the right, the same trajectory is shown, but with the learned controller, with $B = 5$ and $H = 50$ on top and $B = 12$ and $H = 50$ on the bottom. These were recorded between 25 and 30 seconds into the running stage.

Since our approach was reproducible, we consequently tried to reduce the parameters for the easier trot gait to identify the point at which the controller fails to find a stable attractor. We first reduced the number of hidden neurons H . It makes little sense to have $H < 12$, since we have 12 independent outputs to generate. We found that with 12 hidden neurons, we needed B to be at least 16 time steps (115 ms or 20% of the gait period) for a stable gait. Secondly, we removed the buffer ($B = 0$), and searched for the minimal number of hidden neurons required for finding a stable attractor. We found that without a time window, we need at least $H = 128$ 32 hidden neurons. Thirdly, we tried to reduce the training time. For this, we used a controller without a buffer and with $H = 250$ hidden neurons. We found that 1.18 s, or two gait periods are enough for both the first and second training stage, or 2.36 s in total. In this paper, we demonstrated how an embodied control

Conclusions

We have demonstrated how an embodied control system with memory-less nonlinear feedback can generate a stable trot on a compliant quadrupedal robot. When adding memory in the feedback loop, the complexity of the nonlinear feedback can be reduced. Our feedback controller, based on extreme learning machines, learns the desired relation between the input and the output signals in the time span of only a couple of strides.

We have shown that this method can be extended to other gaits, such as a walk, when increasing the training time and the model complexity. The incorporation of either additional memory or additional non-linearities contribute approximately equally to the controller performance. The parameter that mainly determines the performance is the number of signals that is fed into the the linear transformation.



As our controller was trained directly on the actual robot, we did not have to rely on a simulation model, which is often unreliable on a compliant robot. In addition, the controller optimisation was fast, happened entirely online and automatically. We believe that the proposed method can provide a useful tool for transferring knowledge from open loop to closed loop control.

2.11.2.5 Provenance

T4.4.2

2.12 Task 4.4.3 Closed-loop analysis of population coding

2.12.1 O. Marre (UPMC)

2.12.1.1 Introduction

The purpose of this task was to better characterize the sensitivity of the retina to perturbations of the stimulus, in order to find models that give a good account of the sensitivity of the retinal network.

In our approach we have found that several subtypes of retinal ganglion cells could be modelled with a cascade model of two layers. This model allows predicting not only the responses of ganglion cells to stimuli inside their receptive field, but also outside, for surround stimulation. The model is described below.

We found that ganglion cells extract two different features from the stimulus depending on its position compared to their receptive field. For Stimuli inside their receptive field, they were highly sensitive to the absolute position of the object, while outside of their receptive field, they were mostly sensitive to the velocity of the object.

2.12.1.2 Model Description

For the firing rate $r_k(t)$ of the k -th neuron, the equation of this model is:

$$r_k(t) = g\left(\sum_{j, \tau_2} W_{j,k, \tau_2} f\left(\sum_{i, \tau_1} V_{i,j, \tau_1} S(i, t - \tau_1 - \tau_2)\right)\right) + b_k$$

where S represents the visual stimulus, V the connectivity from the photoreceptors to the bipolar cells, f and g are non-linear functions (of the form $f(x) = \log(1 + \exp(x))$), W is the connectivity matrix from subunits to ganglion cells. Parameter values were fitted to the data using a gradient descent similar to (McFarland and Butts, 2013).

2.12.1.3 Model Data

Task(s)/group(s) responsible for generating algorithms/models/principles: WP 4.4.3

- Data, algorithms, tools and methodologies storage location(s) (and links?): model in the SP4 repository
- Description of algorithms/models/principles:
 - Format, language if applicable: matlab program
 - Name of DICs/software catalogue/or HBP github project entries: NA
- Description of data: NA
- Completeness of data/algorithms/models:
 - Comparison of data set/algorithms/models anticipated versus those actually delivered in M30—to what extent does it fall short or exceed the anticipated data set/algorithms/models? It corresponds to what was anticipated.

- Current data set/algorithms/models versus a projected full data set/algorithms/models to be generated by the research community: NA
- Give a short review (1-2 paragraphs) of data/algorithms/models generated by the community over the past 30 months, and how these validate the data/algorithms/models gathered by the HBP Task, and/or complement it: NA
- Outline state of validation work. The model has been validated on rodent retina recordings
- Data Quality and Value:
 - Verification of data quality: fit to some real retina data.
 - Your subjective analysis of the value of the data/algorithms for the users. This model can be used to provide realistic spiking input from the retina to models of the thalamo-cortical network.
- Data/algorithm/model usage to date:
 - Who has used the data/algorithms/models, for what? Please list a) Ramp-Up data (please use DIC name) used for validation or input, and the number and name of the corresponding Ramp-Up Phase Task (and subsidiary group), b) Task number and name of the SGA1 Task that will use the developed models/approach to generate models, or c) Tasks that will build modelling tools that allow usage of the model/approach in SGA2: T 4.4.2 in phase 2.
- Are the data/algorithms/models considered final? Yes, as model for single retinal cells responding to objects moving with complex trajectories.
- Publications connected to the gathered data (please put in parenthesis a short description how they are connected, e.g. description of method used generate data, analysis results, models built using the data, etc.) For the full model, the manuscript is in preparation. Some preliminary experimental data can be found in Marre et al (2015).

2.12.1.4 Model Results

We have shown that predicting the responses of ganglion cells of several subtypes is possible with a two-layer model, while a linear model did not work in that case (Figure 65). If the stimulus is a randomly moving bar, we have been able to fit a network model composed of two layers to the neural responses. Each layer combined a linear filtering and a non-linear function. This model can predict the responses of individual ganglion cells to new sequences of randomly moving bar.

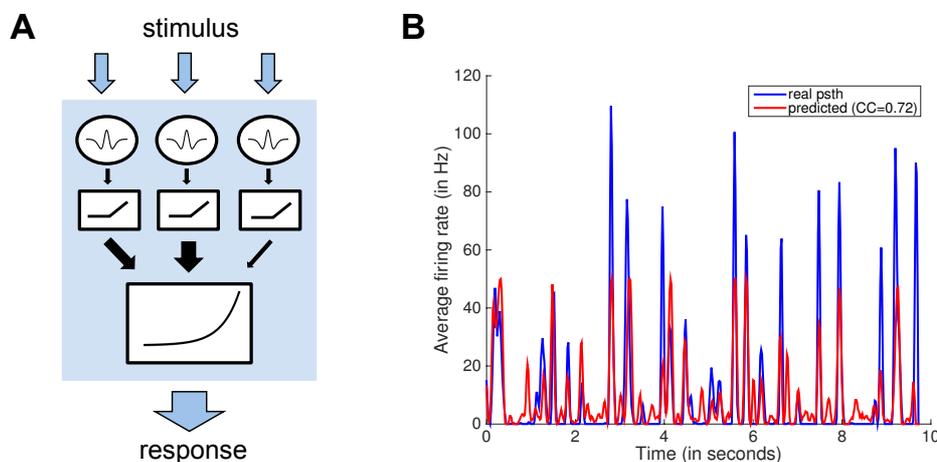


Figure 65: Schematic of model used & example of response prediction

A: schematic of the model we used. The stimulus (here, a randomly moving bar) is filtered by different subunits. For each of these units the result of the filtering goes through a non-linear function. These outputs are pooled linearly, and this weight sum goes through another non-linear function. The final output predicts the responses of ganglion cells to this dynamical stimulus. B: an example of prediction of the response of a subtype of ganglion cell (blue) and the prediction using a cascade model (red). The prediction performance is high (pearson coeff 0.72).

We developed a method to characterize the computations performed by a network is to determine what changes in the input can be reliably distinguished based on the network outputs, and what changes cannot be distinguished. This defines the discrimination capacity of the network, i.e. its sensitivity to some changes, and its invariance to others. In some preliminary analysis we have developed a method to analyze a multi-layered model by local linear approximation. Starting from an initial stimulus, we linearized the input-output function in a local neighborhood around this point of the stimulus space. We then used the Fisher Information Matrix to estimate how sensitive is the model to change in all directions of the stimulus space:

$$I_{i,j}(S) = \left\langle \left(\frac{\partial}{\partial S_i} \log P(R; S) \right) \left(\frac{\partial}{\partial S_j} \log P(R; S) \right) \middle| S \right\rangle$$

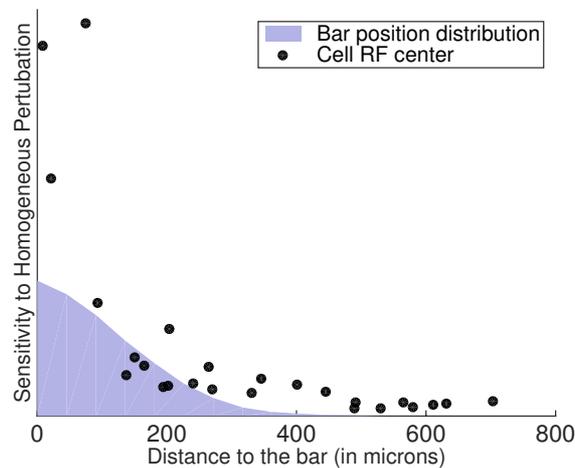


Figure 66: sensitivity (defined from the Fisher information matrix) as a function of the cell position relative to the stimulus.

This matrix quantifies how much change there will be in the model response R , when the stimulus S is changed along a given stimulus direction D . We can thus estimate if the model will be particularly sensitive to changes in a chosen direction D of the stimulus space by computing ${}^T D I(S) D$, which we define as the *sensitivity* of the model. We used this tool to understand the function of the cascade model described before, where the retina responded to a randomly moving bar. With this analysis, we showed that cells activated by distant stimulus are not very sensitive to change in the stimulus mean position, but carry information about the velocity of the bar. On the contrary, cells close to the stimulus are very sensitive to the mean position. Therefore, cells close to the stimulus signal the exact position of the stimulus, while distant cells use a code that is more “position-invariant”. This gives a characterization of the separation and invariance property of the retinal network.

2.12.1.5 Provenance

NA



3. Data Delivery

All the models described in this deliverable correspond to the work done in the last 12 months of the ramp-up phase. All models are provided in sufficient detail to reproduce them, and the main results are described in a 2-3 pages summary with figures.

The program codes of these models were documented and were assembled into a single repository accessible on demand:

Partner	Link
CNRS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b4409fa4a4258edb75d7f0422a26143e
EPFL	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=cf2d9cef0bfa2c8da96278d6d5afad202
HUJI	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=76187736a66948015e3e64c1b2c4d257
INRIA	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=5f9067feb3f144ba9b34df7e24eb538d
JUELICH	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=111776feb0861c45105087ae2f43e4d7
SURREY	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=df23313e33dcee2ccd4ee0038a6e237d
TUGRAZ	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b9b29a3b1f655b72bc66876364aab5d4
TUT	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=aba85ff52a3d3c0b822377fe28361661
UBERN	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=360c876f32b61ec9d4001acf87718cab
UCL	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=e538cb9acee782923933cdc078bcece1e5
UGENT	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=204531670df05930ffbec2504376325c
ULEEDS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=db94bbf7272cf6ca2ea2669e654c2e3a
UMB	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=7752dbcbf7e0f13ac04f210ea03d2bd1
UPF	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b7e08bda0c02f256a92411133d7ec62d
UPMC	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=428018f766d307e14264b0f3a230210c
WIS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=2e2598664a34bc427352c4a8a1acb129



4. Annex A: References

FROM CNRS

Brette, Romain. 2009. "Generation of Correlated Spike Trains." *Neural Comput* 21 (1): 188-215. doi:[10.1162/neco.2008.12-07-657](https://doi.org/10.1162/neco.2008.12-07-657).

FROM HUJI

Mohan H, Verhoog MB, Doreswamy KK, Eyal G, Aardse R, Lodder BN, Goriounova NA, Asamoah B, Brakspear AB, Groot C, van der Sluis S, Testa-Silva G, Obermayer J, Boudewijns ZS, Narayanan RT, Baayen JC, Segev I, Mansvelder HD, de Kock CP (2015) Dendritic and Axonal Architecture of Individual Pyramidal Neurons across Layers of Adult Human Neocortex. *Cereb Cortex*, Dec;25(12):4839-53. doi: 10.1093/cercor/bhv188. Epub 2015 Aug 28

Guy E, Matthijs B., Testa-Silva G, Deitcher Y, Lodder C., Benavides-Piccione R, Morales J, DeFelipe J, de Kock (Submitted). Unique Membrane Properties and Enhanced Computational Capabilities of Human Neurons.

FROM UMB

Buzsáki G & Draguhn A (2004). *Neuronal Oscillations in Cortical Networks*. Science 304, 1926-1930.

Carnevale NT & Hines ML (2006). *The NEURON Book*. Cambridge University Press, Cambridge, UK.

Hadjipapas A, Lowet E, Roberts M, Peter A & De Weerd P (2015). *Parametric variation of gamma frequency and power with luminance contrast: A comparative study of human MEG and monkey LFP and spike responses*. *NeuroImage* 112, 327-340.

Harnett MT, Magee JC & Williams SR (2015). *Distribution and Function of HCN Channels in the Apical Dendritic Tuft of Neocortical Pyramidal Neurons*. *J Neurosci* 35, 1024-1037.

Hay E, Hill S, Schürmann F, Markram H & Segev I (2011). *Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties*. *PLoS Comp Biol* 7, 1-18.

Holt GR & Koch C (1999). *Electrical interactions via the extracellular potential near cell bodies*. *J Comput Neurosci* 6, 169-184.

Hutcheon B & Yarom Y (2000). *Resonance, oscillation and the intrinsic frequency preferences of neurons*. *Trends Neurosci* 23, 216-22.

Koch C (1984). *Cable Theory in Neurons with Active, Linearized Membranes*. *Biol Cybern* 33, 15-33.

Kole MHP, Hallermann S & Stuart GJ (2006). *Single Ih channels in pyramidal neuron dendrites: properties, distribution, and impact on action potential output*. *J Neurosci* 26, 1677-87.

Łęski S, Lindén H, Tetzlaff T, Pettersen KH & Einevoll GT (2013). *Frequency dependence of signal power and spatial reach of the local field potential*. *PLoS Comp Biol* 9, 1-23.



Lindén H, Hagen E, Leski S, Norheim ES, Pettersen KH & Einevoll GT (2014). *LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons*. Front Neuroinform 7, 1-15.

Lindén H, Pettersen KH & Einevoll GT (2010). *Intrinsic dendritic filtering gives low-pass power spectra of local field potentials*. J Comput Neurosci 29, 423-44.

Magee JC (1998). *Dendritic hyperpolarization-activated currents modify the integrative properties of hippocampal CA1 pyramidal neurons*. J Neurosci 18, 7613-24.

Mauro A, Conti F, Dodge F & Schor R (1970). *Subthreshold behavior and phenomenological impedance of the squid giant axon*. J Gen Physiol 55, 497-523.

Mishra P & Narayanan R (2015). *High-conductance states and A-type K⁺ channels are potential regulators of the conductance-current balance triggered by HCN channels*. J Neurophysiol 113, 23-43.

Nunez PL & Srinivasan R (2006). *Electric Fields of the Brain*, Oxford University Press, New York.

Nusser Z (2009). *Variability in the subcellular distribution of ion channels increases neuronal diversity*. Trends Neurosci 32, 267-274.

Petreaunu L, Mao T, Sternson SM & Svoboda K (2009). *The subcellular organization of neocortical excitatory connections*. Nature 457, 1142-1145.

Remme MWH (2014). *Quasi-active approximation of nonlinear dendritic cables* In Jaeger D & Jung R, editors, Encyclopedia of Computational Neuroscience, pp. 1-5. Springer New York.

Remme MWH & Rinzel J (2011). *Role of active dendritic conductances in subthreshold input integration*. J Comput Neurosci 31, 13-30.

Roberts M, Lowet E, Brunet N, TerWal M, Tiesinga P, Fries P & DeWeerd P (2013). *Robust Gamma Coherence between Macaque V1 and V2 by Dynamic Frequency Matching*. Neuron 78, 523-536.

Williams SR & Stuart GJ (2000). *Site independence of EPSP time course is mediated by dendritic I(h) in neocortical pyramidal neurons*. J Neurophysiol 83, 3177-3182.

FROM UPF

Deco, Ponce-Alvarez, Hagmann, Romani, Mantini and Corbetta. *How Local Excitation-Inhibition Ratio Impacts the Whole Brain Dynamics*. J Neurosci (2014) 34: 7886-7898

Deco and Kringelbach. *Great Expectations: Using Whole-Brain Computational Connectomics for Understanding Neuropsychiatric Disorders*. Neuron (2014) 84: 892-905

Deco, Tononi, Boly and Kringelbach. *Rethinking segregation and integration: contributions of whole-brain modelling*. Nat Rev Neurosci (2015) 16: 430-439

Gilson, Moreno-Bote, Ponce-Alvarez and Deco. *Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome*, PLoS Comput Biol (in press).

Hlinka, Palus, Vejmelka, Mantini and Corbetta. *Functional connectivity in resting-state fMRI: is linear correlation sufficient?* Neuroimage (2011) 54:2218-25

Schirner, Rothmeier, Jirsa, McIntosh and Ritter. *An automated pipeline for constructing personalised virtual brains from multimodal neuroimaging data*. Neuroimage (2015) 117: 343-357



Barttfeld, Uhrig, Sitt, Sigman, Jarraya and Dehaene. Signature of consciousness in the dynamics of resting-state brain activity. *PNAS* (2015)

Gilson, Moreno-Bote, Ponce-Alvarez and Deco. Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome, *PLoS Comput Biol* (in press).

Mitra, Snyder, Tagliazucchi, Laufs and Raichle. Propagated infra-slow intrinsic brain activity reorganizes across wake and slow wave sleep, *eLife* (2015) 4:e10781

Tagliazucchi, von Wegner, Morzelewski, Borisov, Jahnke and Laufs. Automatic sleep staging using fMRI functional connectivity data. *Neuroimage* (2012) 63: 63-72

Boly, Perlberg, Marrelec, Schabus, Laureys, Doyon, Péligrini-Issac, Maqueta and Benali. Hierarchical clustering of brain activity during human nonrapid eye movement sleep. *PNAS* (2012) 109: 5856-5861

FROM JÜLICH

1. Jones BE. The organization of central cholinergic systems and their functional importance in sleep-waking states. *Progress in Brain Research* 98 (1993): 61.
2. Jacobs BL, Jones BE. The role of central monoamine and acetylcholine systems in sleep-wakefulness states: mediation or modulation? In *Cholinergic-Monoaminergic interactions in the brain* (pp. 271-290). Academic Press New York (1978).
3. Potjans TC, Diesmann M. The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex* 24.3 (2014): 785-806.
4. Xiang, Z., Huguenard, J. R., & Prince, D. A. (1998). Cholinergic switching within neocortical inhibitory networks. *Science*, 281(5379), 985-988.
5. Eggermann E, Feldmeyer D. Cholinergic filtering in the recurrent excitatory microcircuit of cortical layer 4. *PNAS* 106.28 (2009): 11753-11758.
6. Kawaguchi, Y. (1997). Selective cholinergic modulation of cortical GABAergic cell subtypes. *Journal of Neurophysiology*, 78(3), 1743-1747.
7. Runfeldt MJ, Sadovalsky AJ, MacLean JN. Acetylcholine functionally reorganizes neocortical microcircuits. *J Neurophysiol* 112 (2014): 1205-1216.
8. Schmidt, S. L., Chew, E. Y., Bennett, D. V., Hammad, M. A., & Fröhlich, F. (2013). Differential effects of cholinergic and noradrenergic neuromodulation on spontaneous cortical network dynamics. *Neuropharmacology*, 72, 259-273.
9. Levy, R. B., Reyes, A. D., & Aoki, C. (2006). Nicotinic and muscarinic reduction of unitary excitatory postsynaptic potentials in sensory cortex; dual intracellular recording in vitro. *Journal of neurophysiology*, 95(4), 2155-2166.
10. Gil, Z., Connors, B. W., & Amitai, Y. (1997). Differential regulation of neocortical synapses by neuromodulators and activity. *Neuron*, 19(3), 679-686.
11. Wester, J. C., & Contreras, D. (2013). Differential modulation of spontaneous and evoked thalamocortical network activity by acetylcholine level in vitro. *The Journal of Neuroscience*, 33(45), 17951-17966.

**FROM TUT**

- [1] Agulhon C., Fiacco T.A., McCarthy K.D. Hippocampal short- and long-term plasticity are not modulated by astrocyte Ca²⁺ signaling. *Science* 327(5970):1250-4, 2010.
- [2] Blackwell, Calcium waves and closure of potassium channels in response to GABA stimulation in *Hermissenda* type B photoreceptors, *J. Neurophysiol.* 87(2):776-792, 2002.
- [3] Hituri K., Linne M.-L. Comparison of models for IP₃ receptor kinetics using stochastic simulations. *PLoS ONE*, 8(4): e59618, 2013.
- [4] Kaiser K.M.M., Luebke J., Zilberter Y., Sakmann B. Postsynaptic Calcium Influx at Single Synaptic Contacts between Pyramidal Neurons and Bitufted Interneurons in Layer 2/3 of Rat Neocortex Is Enhanced by Backpropagating Action Potentials. *The Journal of Neuroscience* 24(6): 1319-1329, 2004.
- [5] Kim B., Hawes S.L., Gillani F., Wallace L.J., Blackwell K.T. Signaling pathways involved in striatal synaptic plasticity are sensitive to temporal pattern and exhibit spatial specificity. *PLoS Comput. Biol.* 9(3), e1002953, 2013.
- [6] Larkum M.E., Watanabe S., Nakamura T., Lasser-Ross N., Ross W.N. Synaptically activated Ca²⁺ waves in layer 2/3 and layer 5 rat neocortical pyramidal neurons. *J. Physiol.* 549(Pt 2): 471-488, 2003.
- [7] Li Y., 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.
- [8] Linne M.-L., 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.
- [9] Manninen T., Hituri K., Hellgren-Kotaleski J., Blackwell K. T., Linne M.-L. Postsynaptic signal transduction models for long-term potentiation and depression. *Frontiers in Computational Neuroscience* 4:152, 2010.
- [10] Navarrete M., Perea G., Fernandez de Sevilla D., Gomez-Gonzalo M., Nunez A., Martin E.D., Araque A. Astrocytes Mediate In Vivo Cholinergic-Induced Synaptic Plasticity. *PLoS Biology* 10(2): e1001259, 2012.
- [11] Saudargiene A, Graham BP. Inhibitory control of site-specific synaptic plasticity in a model CA1 pyramidal neuron. *Biosystems* 130:37-50, 2015.
- [12] Shigetomi E., Kracun S., Sofroniew M.V., Khakh B.S. A genetically targeted optical sensor to monitor calcium signals in astrocyte processes. *Nature Neuroscience* 13, 759-766, 2010.
- [13] Toivari E, Manninen T., Nahata AK., Jalonen TO., 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.
- [14] Wade J., McDaid L., Harkin J., Crunelli V., Kelso S. Self-repair in a bidirectionally coupled Astrocyte-Neuron (AN) system based on retrograde signaling. *Frontiers in Computational Neuroscience* 6:76, 2012.
- [15] Zachariou et al., A biophysical model of endocannabinoid-mediated short term depression in hippocampal inhibition, *PLoS ONE*, volume 8, issue 3, e58926, 2013.
- [16] Volman V., Ben-Jacob E., Levine H. The astrocyte as a gatekeeper of synaptic information transfer. *Neural Comput.* 19, 303-326, 2007, 2007.

From ULEEDS



de Kamps, Marc. "A simple and stable numerical solution for the population density equation." *Neural computation* 15.9 (2003): 2129-2146.

Omurtag, A., Bruce W. Knight, and L. Sirovich. "On the simulation of large populations of neurons." *Journal of computational neuroscience* 8.1 (2000): 51-63.

Apfaltrer, Felix, Cheng Ly, and Daniel Tranchina. "Population density methods for stochastic neurons with realistic synaptic kinetics: Firing rate dynamics and fast computational methods." *Network: Computation in Neural Systems* 17.4 (2006): 373-418.

Van Der Velde, Frank, and Marc De Kamps. "Neural blackboard architectures of combinatorial structures in cognition." *Behavioral and Brain Sciences* 29.01 (2006): 37-70.

Hoffmann, Till, Mason A. Porter, and Renaud Lambiotte. "Generalized master equations for non-Poisson dynamics on networks." *Physical Review E* 86.4 (2012): 046102.

Ly, Cheng, and Daniel Tranchina. "Spike train statistics and dynamics with synaptic input from any renewal process: a population density approach." *Neural Computation* 21.2 (2009): 360-396.

Iyer, Ramakrishnan, et al. "The influence of synaptic weight distribution on neuronal population dynamics." *PLoS Comput Biol* 9.10 (2013): e1003248.

Potjans, Tobias C., and Markus Diesmann. "The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model." *Cerebral Cortex* 24.3 (2014): 785-806.

Casti, Alexander Robert Richard, et al. "A population study of integrate-and-fire-or-burst neurons." *Neural Computation* 14.5 (2002): 957-986.

FROM EPFL

[Yasumatsu2008] Yasumatsu N, Matsuzaki M, Miyazaki T, Noguchi J, Kasai H, "Principles of long-term dynamics of dendritic spines.", *J Neurosci* 28, 50 (2008), pp. 13592--13608

[Holtmaat2009]

Holtmaat A, Svoboda K, "Experience-dependent structural synaptic plasticity in the mammalian brain.", *Nat Rev Neurosci* 10, 9 (2009), pp. 647--658

[Feldmeyer1999] Feldmeyer D, Egger V, Lübke J, Sakmann B, "Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single 'barrel' of developing rat somatosensory cortex", *J Physiol* 521 Pt 1 (1999), pp. 169--190

[Fares2009] Fares T, Stepanyants A, "Cooperative synapse formation in the neocortex.", *Proc Natl Acad Sci USA* 106, 38 (2009), pp. 16463--16468.

[Deger2012] Deger M, Helias M, Rotter S, Diesmann M, "Spike-timing dependence of structural plasticity explains cooperative synapse formation in the neocortex.", *PLoS Comput Biol* 8, 9 (2012), pp. e1002689.

[Pfister2006] Pfister JP, Toyoizumi T, Barber D, Gerstner W., "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning." *Neural Computation*, 18(6), pp. 1318-1348

[Schiess2016] Schiess M, Urbanczik R, Senn W, "Somato-dendritic Synaptic Plasticity and Error-backpropagation in Active Dendrites.", *PLoS Comput. Biol* 12(2), pp. 1-18

[Sjöström2006] Sjöström PJ, Häusser M, "A Cooperative Switch Determines the Sign of Synaptic Plasticity in Distal Dendrites of Neocortical Pyramidal Neurons", *Neuron* 51 (2006), pp. 227--238.



[Urbanczik2014] Urbanczik R, Senn, W, "Learning by the dendritic prediction of somatic spiking." *Neuron* 81.3 (2014), pp. 521--528.

FROM SURREY

Publications connected to the gathered data:

[1] . (Derivation, Description and Test of novel spiking learning algorithms).

[2] . (Comparison and joint derivation of a number of spiking learning algorithms).

FROM UCL

Burgess N., O'Keefe J., 1996. Neuronal computations underlying the firing of place cells and their role in navigation, *Hippocampus*, 6 749-762.

Chersi F., Burgess N., 2015. The cognitive architecture of spatial navigation: Hippocampal and Striatal contributions. *Neuron* 88: 64-77.

Daw N. D., Gershman S. J., Seymour B., Dayan P., Dolan R. J., 2008. Model-Based Influences on Humans' Choices and Striatal Prediction Errors. *Neuron* 69, 1204-1215

Packard M.G., McGaugh, J.L., 1996. Inactivation of Hippocampus or Caudate Nucleus with Lidocaine Differentially Affects Expression of Place and Response Learning. *Neurobiol. Learn. Mem.* 65, 65-72.

Pearce J.M., Roberts A.D., Good M., 1998. Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. *Nature* 396, 75-77.

Watkins C.J.C.H., 1989. Learning from Delayed Rewards. PhD thesis, Cambridge University, Cambridge, England.

FROM TUGRAZ

Caporale N, Dan Y (2008) Spike timing-dependent plasticity: a Hebbian learning rule. *Annu Rev Neuroscience* 31: 25-46.

Grashow R, Brookings T, Marder E. Compensation for variable intrinsic neuronal excitability by circuit-synaptic interactions. *The Journal of Neuroscience*. 2010;20(27):9145-9156.

Habenschuss S, Bill J, Nessler B. Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints. In: *Advances in Neural Information Processing Systems*. vol. 25; 2012. p. 782-790.

Habenschuss S, Pühr H, Maass W. Emergence of optimal decoding of population codes through STDP. *Neural Computation*. 2013;25:1-37.

Hopfield JJ, Brody CD. What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences*. 2001;98(3):1282-1287.

Jolivet R, Rauch A, Lüscher HR, Gerstner W. Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience*. 2006;21(1):35-49.

Klampfl S, Maass W. Emergence of dynamic memory traces in cortical microcircuit models through STDP. *The Journal of Neuroscience*. 2013;33(28):11515-11529.

LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998;86(11):2278-2324.



Marder E. Variability, compensation and modulation in neurons and circuits. *Proceedings of the National Academy of Sciences*. 2011;108(3):15542-15548.

Marder E, Taylor AL. Multiple models to capture the variability in biological neurons and networks. *Nature Neuroscience*. 2011;14(2):133-138.

Markram H, Gerstner W, Sjöström PJ (2011) A history of spike-timing-dependent plasticity. *Frontiers in Synaptic Neuroscience* 3: 1-4.

Mensi S, Naud R, Gerstner W. From stochastic nonlinear integrate-and-fire to generalized linear models. In: *Advances in Neural Information Processing Systems*. vol. 24; 2011. p. 1377-1385.

Nessler B, Pfeiffer M, Buesing L, Maass W. Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity. *PLoS Computational Biology*. 2013;9(4):e1003037.

FROM UPMC

Marre, O., Tkacik, G., Simmons, K., Mora, T., Berry, M.J. (2015) Precise decoding of dynamical motion from a large retinal population. *PloS Computational Biology*, Jul 1;11(7):e1004304.

McFarland JM, Cui Y, Butts DA (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Computational Biology* 9: e1003143.



Annex B : Publications

FROM UMB

Ness, Remme, and Einevoll (2015). Active subthreshold dendritic conductances shape the local field potential: <http://arxiv.org/pdf/1512.04293v1.pdf>

FROM ULEEDS

Frank van der Velde and Marc de Kamps, Combinatorial Structures and Processing in Neural Blackboard Architectures, NIPS 2015 workshop: Cognitive Computation: Integrating neural and Symbolic Approaches

Yi Ming Lai and Marc de Kamps, Population Density Equations for Stochastic Processes with Memory Kernels, under review in Phys. Rev. E

Marc de Kamps, Using your MIIND, <http://miind.sf.net/tutorial.pdf>

Marc de Kamps and Yi Ming Lai, A General Solution Method for Two-Dimensional Population Density Equations, In Preparation

FROM UCL

Chersi F., Burgess N., 2015. The cognitive architecture of spatial navigation: Hippocampal and Striatal contributions. *Neuron* 88: 64-77.

FROM TUT

Publications related to HBP (HBP acknowledged). Publications are indicated with bold-cursive in the text.

[1] **T. Manninen, R. Havela, and M.-L. Linne. Computational models of astrocytes and astrocyte-neuron interactions: Categorization, analysis, and future perspectives. Accepted to Computational Glioscience.**

[One of the goals of T434 was to evaluate existing models. This publication presents the characteristics of all astrocyte and astrocyte-neuron models and shows the reproducibility of the original simulation results for five models.].

[2] **H. Teppola, R. Sarkanen, T.O. Jalonen, and M.-L. Linne. Morphological Differentiation Towards Neuronal Phenotype of SH-SY5Y Neuroblastoma Cells by Estradiol, Retinoic Acid and Cholesterol. Neurochem Res. 2015 Oct 30.**

[The paper quantifies vesicle release in neuroblastoma cells and makes comparisons to astrocytic and other neuronal literature. The paper is used as background material in T434].

[3] **Tiina Manninen*, Ausra Saudargiene*, Riikka Havela, Marja-Leena Linne. Computational model of astrocyte-neuron interactions in a cortical synapse. Manuscript in preparation. * Equal contribution.**

[4] **Mikko Lehtimäki, Tiina Manninen, Riikka Havela, Marja-Leena Linne. Method for reduction of dimensionality in neural calcium dynamics models. Manuscript in preparation.**

FROM TUGRAZ



D. Kappel, B. Nessler, and W. Maass. STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning. *PLOS Computational Biology*, 10(3):e1003511, 2014.

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Synaptic sampling: A Bayesian approach to neural network plasticity and rewiring. *Proc. of NIPS 2015: Advances in Neural Information Processing Systems*, 2015.

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Network plasticity as Bayesian inference. *PLOS Computational Biology*, 11(11):e1004485, 2015.

FROM UGENT

Publications connected to the gathered data:

"Developing an Embodied Gait on a Compliant Quadruped", Jonas Degraeve, Ken Caluwaerts, Joni Dambre, Francis wyffels, proceedings of IROS, Sept. 28 - Oct. 2, 2015

(publication of the approach and the experimental results)



Annex C: Installation Procedure

4.1.1.1.1 Third Party Dependencies

It is necessary to install ROOT (<http://cern.root.ch>). ROOT is a powerful analysis platform that has similar capabilities as MATLAB, but is optimized for high end performance. Under Unix systems, this is straightforward. For almost all UNIX-like distributions there is a binary. This can be installed in a directory of your choice. If you have *sudo* or *root* permission, you can opt to have ROOT installed under */usr/local*, thereby making the framework available to all users. It is also possible to install the package locally. Regardless of whether you install the package locally or under */usr/local*, the top of the ROOT directory structure is always a directory called *root*.

Alternatively, you can compile the package from source, using the configure script in the top directory of the download. There is a comprehensive description on how to do this: <https://root.cern.ch/building-root>. Make sure you have all the prerequisites installed that are listed at <https://root.cern.ch/build-prerequisites>.

Make sure that the version you use is configured with `--enable-python`, `--enable-table` and `--enable-mathmore`. You can use Python to inspect the simulation results, and convert them to numpy objects if you feel the need.

Whether you install ROOT locally or system-wide, make sure that the script `root/bin/thisroot.sh` is sourced, e.g. issue the command: `source ~/root/bin/thisroot.sh` if you have installed the package in your home directory. You have to do this every time before you use ROOT, so it is worth to include in a `.bashrc` file or equivalent.

You will also need:

- The GNU Scientific Library, GSL for short
- A recent (> 1.48) version of BOOST.

4.1.1.2 Procedure

Whether you download the tar file or checkout the code from the repository, you will have a top directory called 'miind-git'. This is the **MIIND_ROOT**. Where you place this is immaterial. Perform the following steps:

Directly below 'code', create a directory called 'build'. This 'build' directory will be at the same level as 'apps' and 'libs'.

```
'cd build'
```

```
'ccmake ..'
```

Indicate whether you want a Release or a Debug version, by setting the **CMAKE_BUILD_TYPE** field.

You may have to indicate where ROOT is. CMake is intelligent enough to work out where, once you have provided the location of the root executable.

```
Configure ('c')
```

```
Generate the Make file ('g')
```

```
Quit ('q')
```

```
Type 'make'
```



The libraries will be built in `build/libs`, the executables in `build/apps`. In general, you want to set your `PATH` and `PYTHONPATH` such that they include the path to `MIIND_ROOT/python`. CMake can be used directly from the command line, e.g.:

```
cmake      path_to_miind_src      -DENABLE_MPI=TRUE      -  
DCMAKE_BUILD_TYPE=Debug -DMPIEXEC=/opt/local/bin/openmpiexec
```

To compile the project type: `'make'`. To build the documentation type: `'make doc'`

4.1.1.2.1 A Clean Ubuntu Install

We start with a clean Ubuntu 14-04 machine. Install the following packages with `sudo apt-get install`:

- `g++` (this should be at least `g++ 4.8`, which you should get by default.)
- `python-scipy` • `cmake-curses-gui`
- `libboost-all-dev` • `libgsl0-dev` • `git`

Do not use a Ubuntu package for `ROOT`! It does exist, but misses a few libraries that `MIIND` depends on. Go to the `ROOT` web site: <http://root.cern.ch>, go to Download and click on the most recent version. Download the Ubuntu binary and unpack it in a directory of your choice. Issue the command `source cwd/root/bin/thisroot.sh`, where `'cwd'` should be replaced by the name of the directory where you unpacked. If you now type `root` anywhere in a shell, `ROOT`'s CINT interpreter should start and a splash screen should appear. Exit CINT by typing `.q`. Start a Python shell, and type `import ROOT`. This module should now load without any comment. You may want to incorporate the source command described above in a `.bashrc` file or equivalent.



3. Data Delivery

All the models described in this deliverable correspond to the work done in the last 12 months of the ramp-up phase. All models are provided in sufficient detail to reproduce them, and the main results are described in a 2-3 pages summary with figures.

The program codes of these models were documented and were assembled into a single repository accessible on demand:

Partner	Link
CNRS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b4409fa4a4258edb75d7f0422a26143e
EPFL	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=cf2d9cef0bfa2c8da96278d6d5af4202
HUJI	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=76187736a66948015e3e64c1b2c4d257
INRIA	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=5f9067feb3f144ba9b34df7e24eb538d
JUELICH	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=111776feb0861c45105087ae2f43e4d7
SURREY	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=df23313e33dcee2ccd4ee0038a6e237d
TUGRAZ	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b9b29a3b1f655b72bc66876364aab5d4
TUT	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=aba85ff52a3d3c0b822377fe28361661
UBERN	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=360c876f32b61ec9d4001acf87718cab
UCL	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=e538cb9acee78292393cddc078bcece1e5
UGENT	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=204531670df05930ffbec2504376325c
ULEEDS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=db94bbf7272cf6ca2ea2669e654c2e3a
UMB	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=7752dbcbf7e0f13ac04f210ea03d2bd1
UPF	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=b7e08bda0c02f256a92411133d7ec62d
UPMC	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=428018f766d307e14264b0f3a230210c
WIS	https://www.unic.cnrs-gif.fr/owncloud/public.php?service=files&t=2e2598664a34bc427352c4a8a1acb129



4. Annex A: References

FROM CNRS

Brette, Romain. 2009. "Generation of Correlated Spike Trains." *Neural Comput* 21 (1): 188-215. doi:[10.1162/neco.2008.12-07-657](https://doi.org/10.1162/neco.2008.12-07-657).

FROM HUJI

Mohan H, Verhoog MB, Doreswamy KK, Eyal G, Aardse R, Lodder BN, Goriounova NA, Asamoah B, Brakspear AB, Groot C, van der Sluis S, Testa-Silva G, Obermayer J, Boudewijns ZS, Narayanan RT, Baayen JC, Segev I, Mansvelder HD, de Kock CP (2015) Dendritic and Axonal Architecture of Individual Pyramidal Neurons across Layers of Adult Human Neocortex. *Cereb Cortex*, Dec;25(12):4839-53. doi: 10.1093/cercor/bhv188. Epub 2015 Aug 28

Guy E, Matthijs B., Testa-Silva G, Deitcher Y, Lodder C., Benavides-Piccione R, Morales J, DeFelipe J, de Kock (Submitted). Unique Membrane Properties and Enhanced Computational Capabilities of Human Neurons.

FROM UMB

Buzsáki G & Draguhn A (2004). *Neuronal Oscillations in Cortical Networks*. Science 304, 1926-1930.

Carnevale NT & Hines ML (2006). *The NEURON Book*. Cambridge University Press, Cambridge, UK.

Hadjipapas A, Lowet E, Roberts M, Peter A & De Weerd P (2015). *Parametric variation of gamma frequency and power with luminance contrast: A comparative study of human MEG and monkey LFP and spike responses*. *NeuroImage* 112, 327-340.

Harnett MT, Magee JC & Williams SR (2015). *Distribution and Function of HCN Channels in the Apical Dendritic Tuft of Neocortical Pyramidal Neurons*. *J Neurosci* 35, 1024-1037.

Hay E, Hill S, Schürmann F, Markram H & Segev I (2011). *Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties*. *PLoS Comp Biol* 7, 1-18.

Holt GR & Koch C (1999). *Electrical interactions via the extracellular potential near cell bodies*. *J Comput Neurosci* 6, 169-184.

Hutcheon B & Yarom Y (2000). *Resonance, oscillation and the intrinsic frequency preferences of neurons*. *Trends Neurosci* 23, 216-22.

Koch C (1984). *Cable Theory in Neurons with Active, Linearized Membranes*. *Biol Cybern* 33, 15-33.

Kole MHP, Hallermann S & Stuart GJ (2006). *Single Ih channels in pyramidal neuron dendrites: properties, distribution, and impact on action potential output*. *J Neurosci* 26, 1677-87.

Łęski S, Lindén H, Tetzlaff T, Pettersen KH & Einevoll GT (2013). *Frequency dependence of signal power and spatial reach of the local field potential*. *PLoS Comp Biol* 9, 1-23.



Lindén H, Hagen E, Leski S, Norheim ES, Pettersen KH & Einevoll GT (2014). *LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons*. Front Neuroinform 7, 1-15.

Lindén H, Pettersen KH & Einevoll GT (2010). *Intrinsic dendritic filtering gives low-pass power spectra of local field potentials*. J Comput Neurosci 29, 423-44.

Magee JC (1998). *Dendritic hyperpolarization-activated currents modify the integrative properties of hippocampal CA1 pyramidal neurons*. J Neurosci 18, 7613-24.

Mauro A, Conti F, Dodge F & Schor R (1970). *Subthreshold behavior and phenomenological impedance of the squid giant axon*. J Gen Physiol 55, 497-523.

Mishra P & Narayanan R (2015). *High-conductance states and A-type K⁺ channels are potential regulators of the conductance-current balance triggered by HCN channels*. J Neurophysiol 113, 23-43.

Nunez PL & Srinivasan R (2006). *Electric Fields of the Brain*, Oxford University Press, New York.

Nusser Z (2009). *Variability in the subcellular distribution of ion channels increases neuronal diversity*. Trends Neurosci 32, 267-274.

Petreaanu L, Mao T, Sternson SM & Svoboda K (2009). *The subcellular organization of neocortical excitatory connections*. Nature 457, 1142-1145.

Remme MWH (2014). *Quasi-active approximation of nonlinear dendritic cables* In Jaeger D & Jung R, editors, Encyclopedia of Computational Neuroscience, pp. 1-5. Springer New York.

Remme MWH & Rinzel J (2011). *Role of active dendritic conductances in subthreshold input integration*. J Comput Neurosci 31, 13-30.

Roberts M, Lowet E, Brunet N, TerWal M, Tiesinga P, Fries P & DeWeerd P (2013). *Robust Gamma Coherence between Macaque V1 and V2 by Dynamic Frequency Matching*. Neuron 78, 523-536.

Williams SR & Stuart GJ (2000). *Site independence of EPSP time course is mediated by dendritic I(h) in neocortical pyramidal neurons*. J Neurophysiol 83, 3177-3182.

FROM UPF

Deco, Ponce-Alvarez, Hagmann, Romani, Mantini and Corbetta. *How Local Excitation-Inhibition Ratio Impacts the Whole Brain Dynamics*. J Neurosci (2014) 34: 7886-7898

Deco and Kringelbach. *Great Expectations: Using Whole-Brain Computational Connectomics for Understanding Neuropsychiatric Disorders*. Neuron (2014) 84: 892-905

Deco, Tononi, Boly and Kringelbach. *Rethinking segregation and integration: contributions of whole-brain modelling*. Nat Rev Neurosci (2015) 16: 430-439

Gilson, Moreno-Bote, Ponce-Alvarez and Deco. *Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome*, PLoS Comput Biol (in press).

Hlinka, Palus, Vejmelka, Mantini and Corbetta. *Functional connectivity in resting-state fMRI: is linear correlation sufficient?* Neuroimage (2011) 54:2218-25

Schirner, Rothmeier, Jirsa, McIntosh and Ritter. *An automated pipeline for constructing personalised virtual brains from multimodal neuroimaging data*. Neuroimage (2015) 117: 343-357



Barttfeld, Uhrig, Sitt, Sigman, Jarraya and Dehaene. Signature of consciousness in the dynamics of resting-state brain activity. *PNAS* (2015)

Gilson, Moreno-Bote, Ponce-Alvarez and Deco. Estimation of Directed Effective Connectivity from fMRI Functional Connectivity Hints at Asymmetries in Cortical Connectome, *PLoS Comput Biol* (in press).

Mitra, Snyder, Tagliazucchi, Laufs and Raichle. Propagated infra-slow intrinsic brain activity reorganizes across wake and slow wave sleep, *eLife* (2015) 4:e10781

Tagliazucchi, von Wegner, Morzelewski, Borisov, Jahnke and Laufs. Automatic sleep staging using fMRI functional connectivity data. *Neuroimage* (2012) 63: 63-72

Boly, Perlberg, Marrelec, Schabus, Laureys, Doyon, Péligrini-Issac, Maqueta and Benali. Hierarchical clustering of brain activity during human nonrapid eye movement sleep. *PNAS* (2012) 109: 5856-5861

FROM JÜLICH

1. Jones BE. The organization of central cholinergic systems and their functional importance in sleep-waking states. *Progress in Brain Research* 98 (1993): 61.
2. Jacobs BL, Jones BE. The role of central monoamine and acetylcholine systems in sleep-wakefulness states: mediation or modulation? In *Cholinergic-Monoaminergic interactions in the brain* (pp. 271-290). Academic Press New York (1978).
3. Potjans TC, Diesmann M. The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cerebral Cortex* 24.3 (2014): 785-806.
4. Xiang, Z., Huguenard, J. R., & Prince, D. A. (1998). Cholinergic switching within neocortical inhibitory networks. *Science*, 281(5379), 985-988.
5. Eggermann E, Feldmeyer D. Cholinergic filtering in the recurrent excitatory microcircuit of cortical layer 4. *PNAS* 106.28 (2009): 11753-11758.
6. Kawaguchi, Y. (1997). Selective cholinergic modulation of cortical GABAergic cell subtypes. *Journal of Neurophysiology*, 78(3), 1743-1747.
7. Runfeldt MJ, Sadvosky AJ, MacLean JN. Acetylcholine functionally reorganizes neocortical microcircuits. *J Neurophysiol* 112 (2014): 1205-1216.
8. Schmidt, S. L., Chew, E. Y., Bennett, D. V., Hammad, M. A., & Fröhlich, F. (2013). Differential effects of cholinergic and noradrenergic neuromodulation on spontaneous cortical network dynamics. *Neuropharmacology*, 72, 259-273.
9. Levy, R. B., Reyes, A. D., & Aoki, C. (2006). Nicotinic and muscarinic reduction of unitary excitatory postsynaptic potentials in sensory cortex; dual intracellular recording in vitro. *Journal of neurophysiology*, 95(4), 2155-2166.
10. Gil, Z., Connors, B. W., & Amitai, Y. (1997). Differential regulation of neocortical synapses by neuromodulators and activity. *Neuron*, 19(3), 679-686.
11. Wester, J. C., & Contreras, D. (2013). Differential modulation of spontaneous and evoked thalamocortical network activity by acetylcholine level in vitro. *The Journal of Neuroscience*, 33(45), 17951-17966.

**FROM TUT**

- [1] Agulhon C., Fiacco T.A., McCarthy K.D. Hippocampal short- and long-term plasticity are not modulated by astrocyte Ca²⁺ signaling. *Science* 327(5970):1250-4, 2010.
- [2] Blackwell, Calcium waves and closure of potassium channels in response to GABA stimulation in *Hermissenda* type B photoreceptors, *J. Neurophysiol.* 87(2):776-792, 2002.
- [3] Hituri K., Linne M.-L. Comparison of models for IP₃ receptor kinetics using stochastic simulations. *PLoS ONE*, 8(4): e59618, 2013.
- [4] Kaiser K.M.M., Luebke J., Zilberter Y., Sakmann B. Postsynaptic Calcium Influx at Single Synaptic Contacts between Pyramidal Neurons and Bitufted Interneurons in Layer 2/3 of Rat Neocortex Is Enhanced by Backpropagating Action Potentials. *The Journal of Neuroscience* 24(6): 1319-1329, 2004.
- [5] Kim B., Hawes S.L., Gillani F., Wallace L.J., Blackwell K.T. Signaling pathways involved in striatal synaptic plasticity are sensitive to temporal pattern and exhibit spatial specificity. *PLoS Comput. Biol.* 9(3), e1002953, 2013.
- [6] Larkum M.E., Watanabe S., Nakamura T., Lasser-Ross N., Ross W.N. Synaptically activated Ca²⁺ waves in layer 2/3 and layer 5 rat neocortical pyramidal neurons. *J. Physiol.* 549(Pt 2): 471-488, 2003.
- [7] Li Y., 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.
- [8] Linne M.-L., 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.
- [9] Manninen T., Hituri K., Hellgren-Kotaleski J., Blackwell K. T., Linne M.-L. Postsynaptic signal transduction models for long-term potentiation and depression. *Frontiers in Computational Neuroscience* 4:152, 2010.
- [10] Navarrete M., Perea G., Fernandez de Sevilla D., Gomez-Gonzalo M., Nunez A., Martin E.D., Araque A. Astrocytes Mediate In Vivo Cholinergic-Induced Synaptic Plasticity. *PLoS Biology* 10(2): e1001259, 2012.
- [11] Saudargiene A, Graham BP. Inhibitory control of site-specific synaptic plasticity in a model CA1 pyramidal neuron. *Biosystems* 130:37-50, 2015.
- [12] Shigetomi E., Kracun S., Sofroniew M.V., Khakh B.S. A genetically targeted optical sensor to monitor calcium signals in astrocyte processes. *Nature Neuroscience* 13, 759-766, 2010.
- [13] Toivari E, Manninen T., Nahata AK., Jalonen TO., 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.
- [14] Wade J., McDaid L., Harkin J., Crunelli V., Kelso S. Self-repair in a bidirectionally coupled Astrocyte-Neuron (AN) system based on retrograde signaling. *Frontiers in Computational Neuroscience* 6:76, 2012.
- [15] Zachariou et al., A biophysical model of endocannabinoid-mediated short term depression in hippocampal inhibition, *PLoS ONE*, volume 8, issue 3, e58926, 2013.
- [16] Volman V., Ben-Jacob E., Levine H. The astrocyte as a gatekeeper of synaptic information transfer. *Neural Comput.* 19, 303-326, 2007, 2007.

From ULEEDS



de Kamps, Marc. "A simple and stable numerical solution for the population density equation." *Neural computation* 15.9 (2003): 2129-2146.

Omurtag, A., Bruce W. Knight, and L. Sirovich. "On the simulation of large populations of neurons." *Journal of computational neuroscience* 8.1 (2000): 51-63.

Apfaltrer, Felix, Cheng Ly, and Daniel Tranchina. "Population density methods for stochastic neurons with realistic synaptic kinetics: Firing rate dynamics and fast computational methods." *Network: Computation in Neural Systems* 17.4 (2006): 373-418.

Van Der Velde, Frank, and Marc De Kamps. "Neural blackboard architectures of combinatorial structures in cognition." *Behavioral and Brain Sciences* 29.01 (2006): 37-70.

Hoffmann, Till, Mason A. Porter, and Renaud Lambiotte. "Generalized master equations for non-Poisson dynamics on networks." *Physical Review E* 86.4 (2012): 046102.

Ly, Cheng, and Daniel Tranchina. "Spike train statistics and dynamics with synaptic input from any renewal process: a population density approach." *Neural Computation* 21.2 (2009): 360-396.

Iyer, Ramakrishnan, et al. "The influence of synaptic weight distribution on neuronal population dynamics." *PLoS Comput Biol* 9.10 (2013): e1003248.

Potjans, Tobias C., and Markus Diesmann. "The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model." *Cerebral Cortex* 24.3 (2014): 785-806.

Casti, Alexander Robert Richard, et al. "A population study of integrate-and-fire-or-burst neurons." *Neural Computation* 14.5 (2002): 957-986.

FROM EPFL

[Yasumatsu2008] Yasumatsu N, Matsuzaki M, Miyazaki T, Noguchi J, Kasai H, "Principles of long-term dynamics of dendritic spines.", *J Neurosci* 28, 50 (2008), pp. 13592--13608

[Holtmaat2009]

Holtmaat A, Svoboda K, "Experience-dependent structural synaptic plasticity in the mammalian brain.", *Nat Rev Neurosci* 10, 9 (2009), pp. 647--658

[Feldmeyer1999] Feldmeyer D, Egger V, Lübke J, Sakmann B, "Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single 'barrel' of developing rat somatosensory cortex", *J Physiol* 521 Pt 1 (1999), pp. 169--190

[Fares2009] Fares T, Stepanyants A, "Cooperative synapse formation in the neocortex.", *Proc Natl Acad Sci USA* 106, 38 (2009), pp. 16463--16468.

[Deger2012] Deger M, Helias M, Rotter S, Diesmann M, "Spike-timing dependence of structural plasticity explains cooperative synapse formation in the neocortex.", *PLoS Comput Biol* 8, 9 (2012), pp. e1002689.

[Pfister2006] Pfister JP, Toyoizumi T, Barber D, Gerstner W., "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning." *Neural Computation*, 18(6), pp. 1318-1348

[Schiess2016] Schiess M, Urbanczik R, Senn W, "Somato-dendritic Synaptic Plasticity and Error-backpropagation in Active Dendrites.", *PLoS Comput. Biol* 12(2), pp. 1-18

[Sjöström2006] Sjöström PJ, Häusser M, "A Cooperative Switch Determines the Sign of Synaptic Plasticity in Distal Dendrites of Neocortical Pyramidal Neurons", *Neuron* 51 (2006), pp. 227--238.



[Urbanczik2014] Urbanczik R, Senn, W, "Learning by the dendritic prediction of somatic spiking." *Neuron* 81.3 (2014), pp. 521--528.

FROM SURREY

Publications connected to the gathered data:

[1] . (Derivation, Description and Test of novel spiking learning algorithms).

[2] . (Comparison and joint derivation of a number of spiking learning algorithms).

FROM UCL

Burgess N., O'Keefe J., 1996. Neuronal computations underlying the firing of place cells and their role in navigation, *Hippocampus*, 6 749-762.

Chersi F., Burgess N., 2015. The cognitive architecture of spatial navigation: Hippocampal and Striatal contributions. *Neuron* 88: 64-77.

Daw N. D., Gershman S. J., Seymour B., Dayan P., Dolan R. J., 2008. Model-Based Influences on Humans' Choices and Striatal Prediction Errors. *Neuron* 69, 1204-1215

Packard M.G., McGaugh, J.L., 1996. Inactivation of Hippocampus or Caudate Nucleus with Lidocaine Differentially Affects Expression of Place and Response Learning. *Neurobiol. Learn. Mem.* 65, 65-72.

Pearce J.M., Roberts A.D., Good M., 1998. Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. *Nature* 396, 75-77.

Watkins C.J.C.H., 1989. Learning from Delayed Rewards. PhD thesis, Cambridge University, Cambridge, England.

FROM TUGRAZ

Caporale N, Dan Y (2008) Spike timing-dependent plasticity: a Hebbian learning rule. *Annu Rev Neuroscience* 31: 25-46.

Grashow R, Brookings T, Marder E. Compensation for variable intrinsic neuronal excitability by circuit-synaptic interactions. *The Journal of Neuroscience*. 2010;20(27):9145-9156.

Habenschuss S, Bill J, Nessler B. Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints. In: *Advances in Neural Information Processing Systems*. vol. 25; 2012. p. 782-790.

Habenschuss S, Puhf H, Maass W. Emergence of optimal decoding of population codes through STDP. *Neural Computation*. 2013;25:1-37.

Hopfield JJ, Brody CD. What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences*. 2001;98(3):1282-1287.

Jolivet R, Rauch A, Lüscher HR, Gerstner W. Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience*. 2006;21(1):35-49.

Klampfl S, Maass W. Emergence of dynamic memory traces in cortical microcircuit models through STDP. *The Journal of Neuroscience*. 2013;33(28):11515-11529.

LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998;86(11):2278-2324.



Marder E. Variability, compensation and modulation in neurons and circuits. *Proceedings of the National Academy of Sciences*. 2011;108(3):15542-15548.

Marder E, Taylor AL. Multiple models to capture the variability in biological neurons and networks. *Nature Neuroscience*. 2011;14(2):133-138.

Markram H, Gerstner W, Sjöström PJ (2011) A history of spike-timing-dependent plasticity. *Frontiers in Synaptic Neuroscience* 3: 1-4.

Mensi S, Naud R, Gerstner W. From stochastic nonlinear integrate-and-fire to generalized linear models. In: *Advances in Neural Information Processing Systems*. vol. 24; 2011. p. 1377-1385.

Nessler B, Pfeiffer M, Buesing L, Maass W. Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity. *PLoS Computational Biology*. 2013;9(4):e1003037.

FROM UPMC

Marre, O., Tkacik, G., Simmons, K., Mora, T., Berry, M.J. (2015) Precise decoding of dynamical motion from a large retinal population. *PloS Computational Biology*, Jul 1;11(7):e1004304.

McFarland JM, Cui Y, Butts DA (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Computational Biology* 9: e1003143.



Annex B : Publications

FROM UMB

Ness, Remme, and Einevoll (2015). Active subthreshold dendritic conductances shape the local field potential: <http://arxiv.org/pdf/1512.04293v1.pdf>

FROM ULEEDS

Frank van der Velde and Marc de Kamps, Combinatorial Structures and Processing in Neural Blackboard Architectures, NIPS 2015 workshop: Cognitive Computation: Integrating neural and Symbolic Approaches

Yi Ming Lai and Marc de Kamps, Population Density Equations for Stochastic Processes with Memory Kernels, under review in Phys. Rev. E

Marc de Kamps, Using your MIIND, <http://miind.sf.net/tutorial.pdf>

Marc de Kamps and Yi Ming Lai, A General Solution Method for Two-Dimensional Population Density Equations, In Preparation

FROM UCL

Chersi F., Burgess N., 2015. The cognitive architecture of spatial navigation: Hippocampal and Striatal contributions. *Neuron* 88: 64-77.

FROM TUT

Publications related to HBP (HBP acknowledged). Publications are indicated with bold-cursive in the text.

[1] **T. Manninen, R. Havela, and M.-L. Linne. Computational models of astrocytes and astrocyte-neuron interactions: Categorization, analysis, and future perspectives. Accepted to Computational Glioscience.**

[One of the goals of T434 was to evaluate existing models. This publication presents the characteristics of all astrocyte and astrocyte-neuron models and shows the reproducibility of the original simulation results for five models.]

[2] **H. Teppola, R. Sarkanen, T.O. Jalonen, and M.-L. Linne. Morphological Differentiation Towards Neuronal Phenotype of SH-SY5Y Neuroblastoma Cells by Estradiol, Retinoic Acid and Cholesterol. Neurochem Res. 2015 Oct 30.**

[The paper quantifies vesicle release in neuroblastoma cells and makes comparisons to astrocytic and other neuronal literature. The paper is used as background material in T434].

[3] **Tiina Manninen*, Ausra Saudargiene*, Riikka Havela, Marja-Leena Linne. Computational model of astrocyte-neuron interactions in a cortical synapse. Manuscript in preparation. * Equal contribution.**

[4] **Mikko Lehtimäki, Tiina Manninen, Riikka Havela, Marja-Leena Linne. Method for reduction of dimensionality in neural calcium dynamics models. Manuscript in preparation.**

FROM TUGRAZ



D. Kappel, B. Nessler, and W. Maass. STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning. *PLOS Computational Biology*, 10(3):e1003511, 2014.

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Synaptic sampling: A Bayesian approach to neural network plasticity and rewiring. *Proc. of NIPS 2015: Advances in Neural Information Processing Systems*, 2015.

D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass. Network plasticity as Bayesian inference. *PLOS Computational Biology*, 11(11):e1004485, 2015.

FROM UGENT

Publications connected to the gathered data:

"Developing an Embodied Gait on a Compliant Quadruped", Jonas Degraeve, Ken Caluwaerts, Joni Dambre, Francis wyffels, proceedings of IROS, Sept. 28 - Oct. 2, 2015

(publication of the approach and the experimental results)



Annex C: Installation Procedure

4.1.1.1.1 Third Party Dependencies

It is necessary to install ROOT (<http://cern.root.ch>). ROOT is a powerful analysis platform that has similar capabilities as MATLAB, but is optimized for high end performance. Under Unix systems, this is straightforward. For almost all UNIX-like distributions there is a binary. This can be installed in a directory of your choice. If you have *sudo* or *root* permission, you can opt to have ROOT installed under */usr/local*, thereby making the framework available to all users. It is also possible to install the package locally. Regardless of whether you install the package locally or under */usr/local*, the top of the ROOT directory structure is always a directory called *root*.

Alternatively, you can compile the package from source, using the configure script in the top directory of the download. There is a comprehensive description on how to do this: <https://root.cern.ch/building-root>. Make sure you have all the prerequisites installed that are listed at <https://root.cern.ch/build-prerequisites>.

Make sure that the version you use is configured with `--enable-python`, `--enable-table` and `--enable-mathmore`. You can use Python to inspect the simulation results, and convert them to numpy objects if you feel the need.

Whether you install ROOT locally or system-wide, make sure that the script `root/bin/thisroot.sh` is sourced, e.g. issue the command: `source ~/root/bin/thisroot.sh` if you have installed the package in your home directory. You have to do this every time before you use ROOT, so it is worth to include in a `.bashrc` file or equivalent.

You will also need:

- The GNU Scientific Library, GSL for short
- A recent (> 1.48) version of BOOST.

4.1.1.2 Procedure

Whether you download the tar file or checkout the code from the repository, you will have a top directory called 'miind-git'. This is the **MIIND_ROOT**. Where you place this is immaterial. Perform the following steps:

Directly below 'code', create a directory called 'build'. This 'build' directory will be at the same level as 'apps' and 'libs'.

```
'cd build'
```

```
'ccmake ..'
```

Indicate whether you want a Release or a Debug version, by setting the **CMAKE_BUILD_TYPE** field.

You may have to indicate where ROOT is. CMake is intelligent enough to work out where, once you have provided the location of the root executable.

```
Configure ('c')
```

```
Generate the Make file ('g')
```

```
Quit ('q')
```

```
Type 'make'
```



The libraries will be built in `build/libs`, the executables in `build/apps`. In general, you want to set your `PATH` and `PYTHONPATH` such that they include the path to `MIIND_ROOT/python`. CMake can be used directly from the command line, e.g.:

```
cmake path_to_miind_src -DENABLE_MPI=TRUE -  
DCMAKE_BUILD_TYPE=Debug -DMPIEXEC=/opt/local/bin/openmpiexec
```

To compile the project type: `'make'`. To build the documentation type: `'make doc'`

4.1.1.2.1 A Clean Ubuntu Install

We start with a clean Ubuntu 14-04 machine. Install the following packages with `sudo apt-get install`:

- `g++` (this should be at least `g++ 4.8`, which you should get by default.)
- `python-scipy` • `cmake-curses-gui`
- `libboost-all-dev` • `libgsl0-dev` • `git`

Do not use a Ubuntu package for `ROOT`! It does exist, but misses a few libraries that `MIIND` depends on. Go to the `ROOT` web site: <http://root.cern.ch>, go to Download and click on the most recent version. Download the Ubuntu binary and unpack it in a directory of your choice. Issue the command `source cwd/root/bin/thisroot.sh`, where `'cwd'` should be replaced by the name of the directory where you unpacked. If you now type `root` anywhere in a shell, `ROOT`'s CINT interpreter should start and a splash screen should appear. Exit CINT by typing `.q`. Start a Python shell, and type `import ROOT`. This module should now load without any comment. You may want to incorporate the source command described above in a `.bashrc` file or equivalent.