International Neuroinformatics Coordinating Facility

Neuro 2009 informatics

2nd INCF Congress of Neuroinformatics

Pilsen, Czech Republic - September 6 - 8, 2009

Workshop topics:

- **Advances in the automatic analysis of multi-dimensional data**
  Chairs: Jaap Van Pelt, Ulla Ruotsalainen
  Bart ter Haar Romeny, Uri Eden, Klaus Linkenkaer-Hansen
- **Ontologies for neuroscience: applications and advances**
  Chair: Maryann Martone
  Tim Clark, Alan Ruttenberg, Jeffrey Grethe
- **How should a neuron be modeled: biophysical detail vs. abstraction**
  Chairs: Gaute Einevoll, Andreas Herz
  Arnd Roth, Wulfram Gerstner, Peter Hunter
- **High performance computing and grid infrastructure for neuroinformatics applications**
  Chairs: Luciano Milanesi, Shiro Usui
  Markus Diesmann, Andrey Semin, Pietro Liò
- **The neuroinformatics of neural connectivity**
  Chairs: David Willshaw, Kevan Martin
  Albert Cardona, Giorgio Ascoli, Rolf Kötter

Keynote speakers:

Kenji Doya
Alon Halevy
Astrid Prinz
Andrew Schwartz
Shankar Subramaniam
Arthur Toga

ted by the EU Special Support Action INCF, the INCF Central Fund and the Swedish Foundation for Strategic Research

www.neuroinformatics2009.org

incf

Poster Presentation

Simulating macroscale brain circuits with microscale resolution

Susanne Kunkel[1*], Tobias C Potjans[2,3], Abigail Morrison[4] and Markus Diesmann[1,4,3]

1 Bernstein Center for Computational Neuroscience, Freiburg, Germany

2 Research Center Jülich, Juelich, Germany

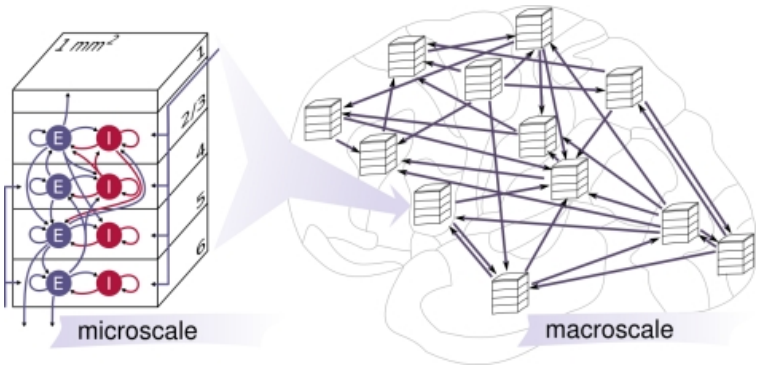
3 RIKEN Computational Science Research Program, Wako-shi, Japan

4 RIKEN Brain Science Institute, Wako-shi, Japan

Thanks to distributed computing techniques, today it is possible to routinely simulate local cortical networks of around 10^5 neurons with up to 10^9 synapses on clusters and multiple processor shared memory machines. Simulations of this type carried out with NEST [1] scale well up to at least 1000 processors. However, simulations of microscale networks corresponding to approximately $1mm^3$ of the cortex are limited in their explanatory power. To understand the functions of the brain, we need to simulate macroscale circuits involving multiple interacting areas [2]. One approach is to develop brain-scale networks (see Figure) in which the individual nodes are realized by microcircuits at the resolution of point neurons and synapses such as the layered local cortical network model we recently developed [3]. These networks will be one or two orders of magnitude larger than the previously studied models not only in terms of numbers of neurons and synapse but also in terms of computational load.

This presents a number of challenges to current simulation technology. Firstly, as the number of processors increases, the memory overhead due to serial data structures eventually dominates the total memory usage and so limits the parallelization. An example of this is the usage of proxies representing remote neurons on each machine and providing an interface to the local neurons. Although such representations of remote neurons require much less memory than local neurons, their proportion of the total neuronal memory usage approaches one as the number of machines increases. Secondly, the size of the maximal synaptic delay determines the size of the spike buffers each neuron uses to queue incoming spikes [4]. For local networks these buffers are small, as the synaptic delays are in the order of milliseconds. If interacting brain areas are to be considered, the synaptic delays increase by an order of magnitude, which entails a corresponding increase of memory usage for the spike buffers. It is also important that a distinction can be made between the axonal and the dendritic components of synaptic delays, as this affects the behavior of plasticity models dependent on relative spike timing [5,6]. We quantify the effects of these memory limitations in our application up to the order of 10k processors and present strategies for addressing these problems.

Next-Generation Supercomputer Project of MEXT, EU Grant 15879 (FACETS), BMBF Grant 01GQ0420, Helmholtz Alliance on Systems Biology



1. Gewaltig M-O & Diesmann M (2007) Scholarpedia, 2(4):1430
2. Sporns O et al. (2005) PLoS Comput Biol, 1(4):e42
3. Potjans TC & Diesmann M (2008) 38th SfN Meeting, 16.1
4. Morrison et al. (2005) Neural Comput, 17(8):1776-801
5. Morrison et al. (2008) Biol Cybern, 98(6):459-78
6. Lubenov EV & Siapas AG (2008) Neuron, 58:118-131

Poster Presentation

Identifying specific structural criteria for quasi-stable network dynamics

Sarah Jarvis[1*], Stefan Rotter[2] and Ulrich Egert[3]

1 Faculty of Applied Sciences , University of Freiburg, Freiburg, Germany

2 Faculty of Biology, BCCN, University of Freiburg, Freiburg, Germany

3 Biomicrotechnology, IMTEK, , University of Freiburg, Freiburg, Germany

Disassociated cortical cultures grown on micro-electrode arrays (MEA) have been established as a useful biological model in the analysis of network dynamics. Present in their dynamics are periods with high firing rates of network wide activity, termed 'bursting', whose purpose is not understood but that may saturate network dynamics. It has been demonstrated that bursts contain repeated spatio-temporal pattern motifs that could reflect the networks connectivity and structure. In such biological networks anatomical clustering of neurons has been demonstrated. While this does not prove that the networks are functionally inhomogeneous, clustering could be a structural basis for the bursting dynamics observed.

We implement a recurrent network model in order to understand what structural constraints are necessary to facilitate bursting. Our model extends an Echo State Network (ESN) architecture by introducing clustering within the reservoir, based on the modifications proposed for Scale-Free Highly Clustered ESN [1]. We have previously investigated [2] the effect of introducing hierarchical clustered reservoir architectures as well as the impact of modifying non-structural parameters, such as time constants of reservoir units. We reported that structural changes had the biggest impact on network dynamics and outlined some basic criteria required for bursting to be initiated, related to the ratio of number of clusters to the size of clusters. Furthermore we demonstrated that as the ratio is increased, bursting appears at the transition from stable to unstable dynamics. However, we were unable to reliably reproduce bursting behaviour within our networks, suggesting that additional criteria exist.

Here, we refine our findings and attempt to determine more specific criteria for reservoir architecture. Specifically, we examine the in and out degrees of cluster connectivity, eigenvalue spectra of connection weights and unequal weight distributions and report on how different distributions affect bursting reliability.

Lastly, we relate the presence of clusters to changes in network properties, and demonstrate that while simulated networks that display bursting have a reduced memory capacity, they are better able to preserve network activity. Thus, while bursting is only quasi-stable behaviour, it is likely symptomatic of network connectivity optimized for maintaining spontaneous activity.

Acknowledgements:
Supported by the German BMBF (FKZ 01GQ0420) and by the EC (NEURO, No. 12788)

1. Deng and Zhang (2007) ‘Collective behavior of a small-world recurrent neural system with scale-free distribution’, IEEE Trans Neural Netw 18, 1364–1375.
2. Jarvis, Rotter, Egert “Aiming for stable instability: introducing hierarchy to reservoir networks” submitted to Frontiers in Neuroinformatics

Poster Presentation

A testsuite for a neural simulation engine

Jochen M Eppler[1*], Rudiger Kupper[1], Hans E Plesser[2] and Markus Diesmann[3]

1 Honda Research Institute Europe GmbH, Offenbach, Germany

2 Norwegian University of Life Sciences, Oslo, Norway

3 RIKEN Brain Science Institute, Saitama, Japan

Testing is a standard activity of the software development process [1,2]. However, little research has been carried out on the specific problems of testing neuronal simulation engines. We found that with the growing complexity of the software and the growing number of developers, formalized and systematic testing becomes critical. This is because the changes made to the code for a new feature are more likely to break an existing function and no researcher has a full overview of the code. The rapid growth of neuroscientific knowledge and the changing research directions require an incremental/iterative development style [3] which extends over the full life time of the product. Thus, there is no single testing phase but the same tests need to be carried out repetitively over a time span of many years. Furthermore, there is usually no rigorous way to determine whether the result of a simulation is correct. The confidence of the scientist relies on comparisons of the results of different implementations and the correctness of critical components of the system.

Here we present the architecture of the testing framework we have developed for NEST [4]. The fully automatized testsuite consists of a collection of test scripts which are sequentially executed by a concise shell script. While high expressiveness of the test code is a goal, sometimes more primitive code is used in order not to condition a test on the correctness of a further component. NEST's built-in simulation language (SLI) is the lowest level at which tests are formulated. The C++ classes of the simulation kernel have no code for testing but are equipped with checks of invariants (assertions).

In designing the tests we found the following principles effective: (1) test the ability to report errors (self-test), (2) hierarchically organize tests from simple to complex, (3) test that objects have the expected default values and accept parameter changes, (4) compare simulation results with analytical results for simple scenarios, (5) check correctness of results with simpler algorithms, (6) test the convergence of results with decreasing simulation time step, (7) check for expected accuracy, (8) test the invariance of results with increasing numbers of processors used, (9) create regression tests for fixed problems.

It is essential to combine these principles. For example, a simulation may converge with decreasing time step, but to the wrong result. Convergence to the correct result still does not guarantee correctness of the implementation: if a signal delay is offset by one time step, the simulation may converge, but with unexpectedly large errors at a given resolution.

Acknowledgements: Next-Generation Supercomputer Project of MEXT, EU Grant 15879 (FACETS), BMBF Grant 01GQ0420, Helmholtz Alliance on Systems Biology

1. Sommerville I (2007) Software Engineering (8th edn), Harlow, Addison-Wesley
2. Beck K & Andres C (2004) Extreme Programming Explained (2nd edn), Boston, Addison-Wesley
3. Diesmann M and Gewaltig M-O (2002) GWDG Bericht, 58:43-70
4. Gewaltig M-O & Diesmann M (2007) Scholarpedia, 2(4):1430

Poster Presentation

Implementing neuromodulated plasticity in distributed simulations

Wiebke Potjans[1*], Abigail Morrison[2] and Markus Diesmann[2]

1 Institute of Neurosciences and Medicine, Research Center Jülich, Juelich, Germany

2 RIKEN Brain Science Institute, Saitama, Japan

A major challenge in computational neuroscience is to explain animal learning on the cellular level. Learning is thought to be implemented by changes in synaptic efficacy. Theoretical models of synaptic plasticity have been mainly influenced by Hebb's postulate that the sequential activation of two neurons strengthens the synapse connecting them. However, there is clear experimental evidence that synaptic plasticity can depend not only on the activity of the pre- and post-synaptic neurons but also on the presence of neuromodulators which act as a "third factor" [1]. The availability of a biological substrate for the third factor solves the puzzle of how dependence on a global reward signal in theories of system-level learning could be realized in a neuronal network with local synaptic plasticity. Such rules have recently been investigated in a number of theoretical studies (e.g. [2,3]).

Implementing neuromodulated plasticity in large-scale network simulations is challenging because the dynamics of these networks is commonly defined on the connectivity graph without explicit reference to the embedding of the nodes in physical space (e.g. NEST [4]). Simulations of models with closed functional circuits require the simultaneous representation of synapses with neuromodulated plasticity and neurons releasing a neuromodulator in particular target regions. Furthermore, the simulation of networks with biologically realistic connectivity entails the use of distributed computing techniques [5]. A modulated synapse must therefore be informed in an efficient way of neuronal activity which is typically generated on a different machine than either the pre- or the post-synaptic neuron.

We present a general framework to address this problem without reference to a particular implementation language, neuromodulator, or neuromodulated plasticity mechanism. To illustrate the technique, we implement a recurrent network with dopamine-modulated spike timing dependent plasticity [2] in the simulator NEST [4].

Next-Generation Supercomputer Project of MEXT, EU Grant 15879 (FACETS), BMBF Grant 01GQ0420, Helmholtz Alliance on Systems Biology

1. Reynolds JNJ et al. (2001) Nature, 413:67-70
2. Izhikevich EM (2007) Cereb Cortex, 17(10):2443-2452
3. Legenstein R et al. (2008) PLoS Comput Biol, 4(10):e1000180
4. Gewaltig M-O & Diesmann M (2007) Scholarpedia, 2(4):1430
5. Morrison A et al. (2005) Neural Comput, 17(8):1776-1801