

POSTER PRESENTATION

Open Access

Flexible and efficient recording of neuronal properties in large network simulations: The NEST Multimeter

Hans E Plesser^{1*}, Jochen M Eppler²

From Twentieth Annual Computational Neuroscience Meeting: CNS*2011
Stockholm, Sweden. 23-28 July 2011

In computer simulations, all quantities are known and thus in principle available for recording and analysis. Collecting data from large simulations in an efficient and flexible manner is still challenging. Dumping all available data to file generates huge amounts of data, requiring significant post-processing to extract relevant information. Instrumenting all parts of a complex simulator for recording is also not trivial, especially with respect to file management and data collection in parallel simulations. We have developed the NEST Multimeter to facilitate flexible and efficient data recording from large-scale parallel neuronal network simulations with the NEST simulator [1]. Design goals were to: (i) allow users to selectively record neuron properties, e.g., membrane potential; (ii) provide a unified user interface for all types of recordings; (iii) support recording in parallel simulations; (iv) minimize the computation time overhead; and (v) provide an API that allows model developers to make any scalar quantity in a model neuron available for recording.

From NEST version 2.0.0-RC1 onward (available from <http://www.nest-initiative.org>), all NEST neuron models support multimeter recording and the user can easily enquire the recordable quantities:

```
In [4]: nest.GetDefaults('aeif_cond_alpha')['recordables']  
Out[4]: ['V_m', 'g_ex', 'g_in', 'w']
```

reveals that the conductance-based adaptive exponential integrate-and-fire neuron model [2] supports recording of the membrane potential (V_m), excitatory and

inhibitory synaptic conductances (g_{ex} , g_{in}) and the adaptation current (w). We record membrane potential and adaptation current by connecting a properly configured multimeter:

```
In [5]: n = nest.Create('aeif_cond_alpha', params={'I_e': 100.})  
In [6]: m = nest.Create('multimeter', params={'record_from': ['V_m', 'w']})  
In [7]: nest.Connect(m, n)
```

After simulating for 5ms, we obtain the recorded values from the multimeter:

```
In [9]: nest.GetStatus(m)[0]['events']  
Out [9]: {'V_m' : array([-70.2624629, -69.9591348, -69.6865797, -69.4417065]),  
'w' : array([ 0.00476033, 0.01834889, 0.03980607, 0.06826917]),  
'times' : array([ 1. , 2. , 3. , 4. ])}
```

Data can also be recorded to files, with one file per virtual process, ensuring conflict-free recording in parallel simulations. When data is retrieved as illustrated above, it is combined across threads, but not MPI processes. A multimeter can record from an arbitrary number of neurons.

Model developers instrument a model for recording by adding a small amount of boilerplate code to models: a UniversalDataLogger (UDL) member collecting recorded values temporarily, and a RecordablesMap (RM):

```
template < void RecordablesMap<aeif_cond_alpha>::create() {  
    insert_(names::V_m , &aeif_cond_alpha::get_y_elem_<aeif_cond_alpha::State_::V_M>);  
    insert_(names::g_ex , &aeif_cond_alpha::get_y_elem_<aeif_cond_alpha::State_::G_EXC>);  
    insert_(names::g_in , &aeif_cond_alpha::get_y_elem_<aeif_cond_alpha::State_::G_INH>);  
    insert_(names::w , &aeif_cond_alpha::get_y_elem_<aeif_cond_alpha::State_::W>); }
```

When the multimeter is connected to a neuron, it informs the UniversalDataLogger about which entries of the RM shall be recorded. Once per time step, the model neuron calls `UDL::record_data()`, which collects the required data using the access functions specified in the RM. Once per minimal delay interval, the

* Correspondence: hans.ekkehard.plesser@umb.no

¹Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences, 1432 Aas, Norway

Full list of author information is available at the end of the article

multimeter collects the data from the UDLs of all connected neurons and stores it in memory or writes it to file.

Acknowledgements

This work is part of the long-term collaboration within the NEST Initiative, especially with M. Diesmann, H. Enger, M.-O. Gewaltig and A. Morrison. Supported by the Research Council of Norway under grant 178892/V30 *eNeuro*.

Author details

¹Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences, 1432 Aas, Norway. ²Institute of Neuroscience and Medicine, Computational and Systems Neuroscience (INM-6), Research Center Juelich, 52428 Juelich, Germany.

Published: 18 July 2011

References

1. Gewaltig MO, Diesmann M: NEST (NEural Simulation Tool). *Scholarpedia* 2007, **2**(4):1430.
2. Brette R, Gerstner W: Adaptive exponential integrate-and-fire model as an effective description of Neuronal activity. *J Neurophysiol* 2005, **94**:3637-3642.

doi:10.1186/1471-2202-12-S1-P92

Cite this article as: Plesser and Eppler: Flexible and efficient recording of neuronal properties in large network simulations: The NEST Multimeter. *BMC Neuroscience* 2011 **12**(Suppl 1):P92.

**Submit your next manuscript to BioMed Central
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

