Advanced plasticity rules in NEST

Clopath and Urbanczik-Senn plasticity

Jonas Stapmanns[1,2], Jan Hahne[3], David Dahmen[1], Moritz Helias[1,2], Matthias Bolten[3], Markus Diesmann[1,4,5]

[1] Institute of Neuroscience and Medicine (INM-6) and Institute for Advanced Simulation (IAS-6) and JARA-Institut Brain Structure-Function Relationships (INM-10), Jülich Research Centre, Jülich, Germany

[2] Institute for Theoretical Solid State Physics, RWTH Aachen University, 52074 Aachen, Germany

[3] School of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal, Germany

[4] Department of Psychiatry, Psychotherapy and Psychosomatics, School of Medicine, RWTH Aachen University, Aachen, Germany

[5] Department of Physics, Faculty 1, RWTH Aachen University, Aachen, Germany

Contact: j.stapmanns@fz-juelich.de

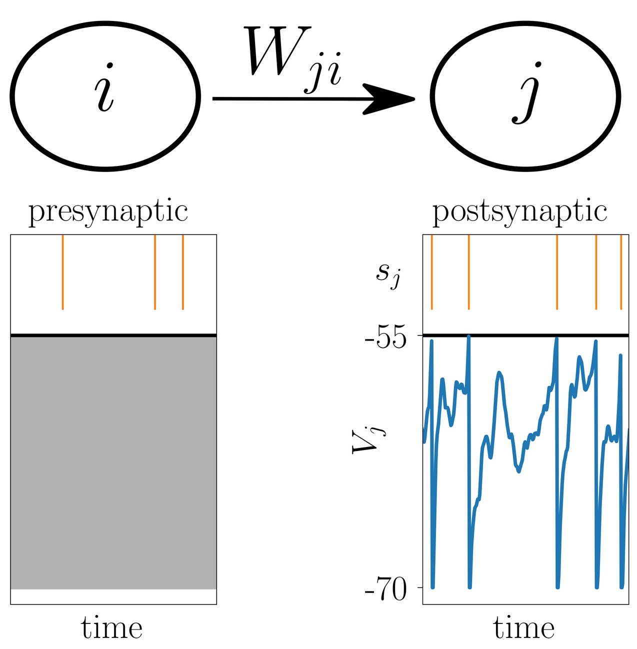
Learning rules

- Spike-timing dependent plasticity (STDP) is a form of Hebbian plasticity that relies on exact **spike times of pre- and postsynaptic neurons**
  - synapse requires history of spikes s (events)
  - suitable for event-based synapse updates
- Experimental evidence [1] and functional motivations [2] ask for plasticity features beyond STDP that rely on **postsynaptic membrane potential**
  - synapse requires history of spikes s (events) and postsynaptic membrane potential V_j (continuous signal)
  - a priori demands for time-driven synapse updates
- Here, show how to embed such third-factor plasticity rules in event-driven synapse update scheme in **NEST**

General update rule for synaptic weights:

$$\frac{dW_{ji}}{dt} = F[s_i, s_j, V_j]$$

Functional F depends on pre- (i) and postsynaptic (j) spikes and postsynaptic membrane potential.



Clopath rule:

$$F[s_i, s_j, V_j] = -A_- \, s_i \, (\kappa_- * V_j - \theta_-)_+ + A_+ \, K * s_i \, (\kappa_+ * V_j - \theta_-)_+ (V_j - \theta_+)_+$$

with exponential filter kernels κ_+, κ_- and K.

Urbanczik-Senn rule:

$$F[s_i, s_j, V_j] = \eta \, \kappa * \Big((s_j - \phi(V_j)) \, h(V_j) \, K * s_i \Big)$$

with exponential filter kernels κ and K and nonlinearities ϕ, h.

Implementation in NEST

- Postsynaptic neuron: storage buffer for time trace of membrane potential
- Synapse: access to membrane potential at time points of spike delivery
- Different summation schemes:
  1) separate summation for each synapse (Fig. 2A)
  2) summation carried out once and result used for all synapses ("backward summation") (Fig. 2B)
  - backward summation scheme potentially faster for expensive summations
  - backward summation requires strictly chronological processing of spikes → in NEST only possible if delay equal to resolution of simulation
- Current status:
  - Clopath rule available in NEST 2.18.0
  - Pull request with Urbanczik-Senn rule under review

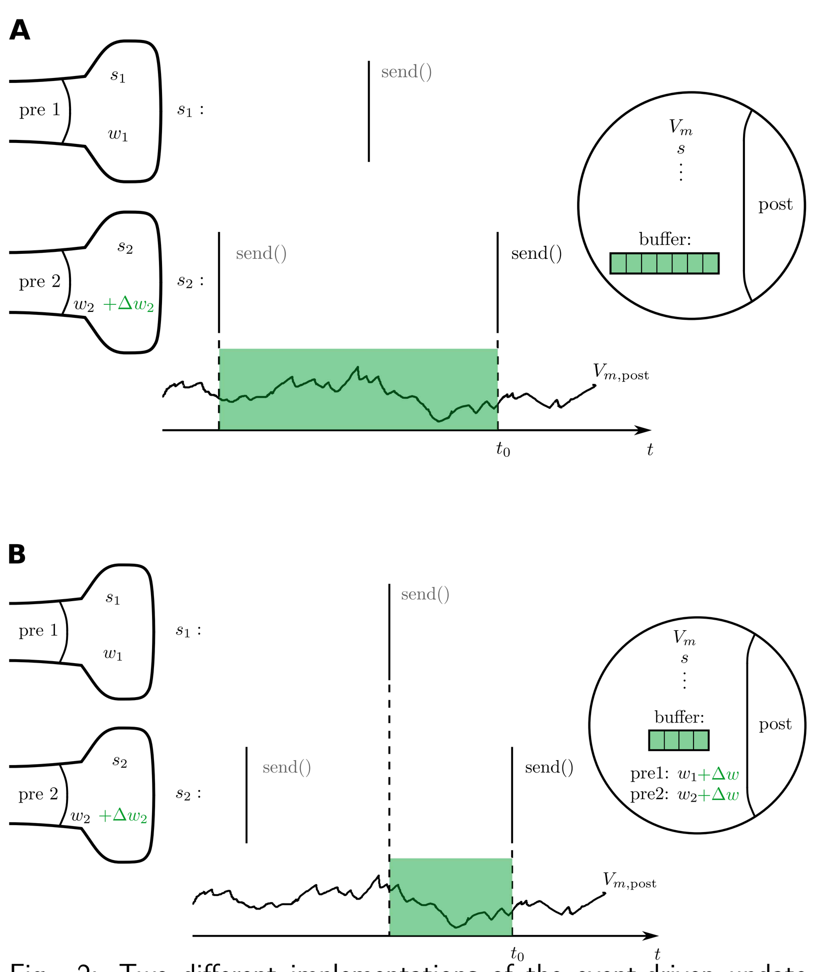


Fig. 2: Two different implementations of the event-driven update scheme in NEST. Two presynaptic neurons ($\mathrm{pre}_{1/2}$) send spike trains ($s_{1/2}$) to a postsynaptic neuron (post). The synaptic weights ($w_{1/2}$) depend on the spike times (vertical bars) and the postsynaptic membrane potential ($V_{m,\mathrm{post}}$). In **A** the weight change of a synapse is computed when this synapse sends a spike, whereas in **B** the weight change for all synapses is computed whenever there is an incoming spike at the postsynaptic neuron.

Outlook

- Discuss how the combinatorial explosion caused by the combinations of neuron and synapse models can be avoided using NESTML
- Use infrastructure to implement biologically plausible approximations to back-propagation through time
  - Network of few-compartment cells [4]
  - *E-prop* algorithm [5] for recurrent networks of spiking neurons

Reproduction of results

Clopath rule

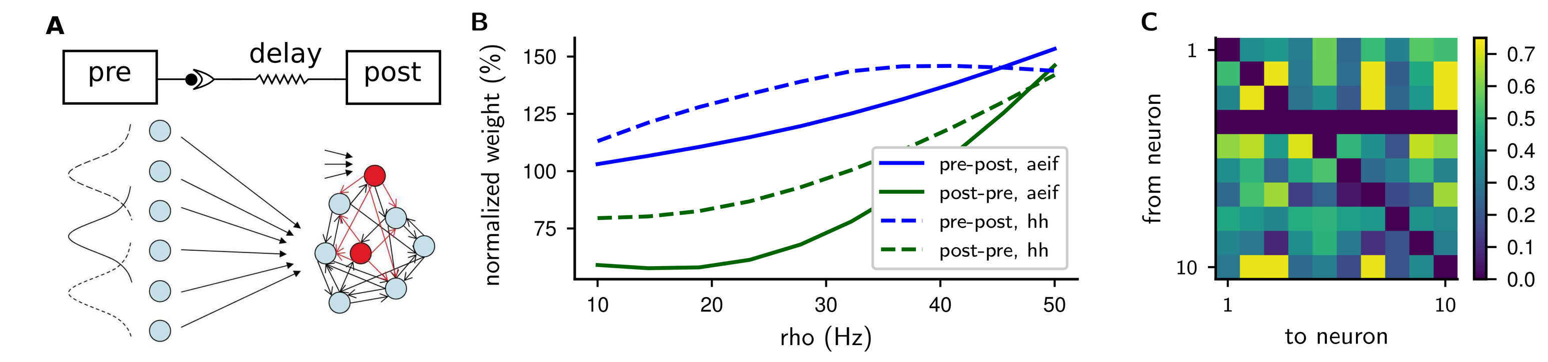


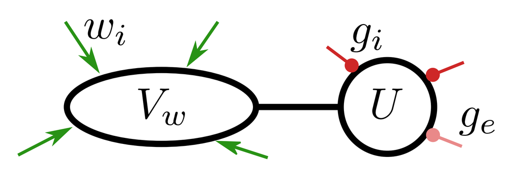
Fig. 3: **A** Setups of two experiments that use the Clopath synapse. Spike pairing experiment (top) and small network driven by external input (bottom, adapted from [1]). **B** Normalized weight in spike pairing experiments with AdEx neurons (solid lines) and Hodgkin-Huxkey neurons (dashed liness) for $t^k_{\mathrm{post}} - t^k_{\mathrm{pre}} = +10\,\mathrm{ms}$ (blue) and $t^k_{\mathrm{post}} - t^k_{\mathrm{pre}} = -10\,\mathrm{ms}$ (green). Corresponds to figure 2b in [1]. **C** Emergence of strong bidirectional couplings between neurons of the excitatory population. Corresponds to figure 5 in [1].

Urbanczik-Senn rule

Plasticity of dendritic synapses w_i:

- Aim: prediction of somatic firing ('evidence') from dendritic membrane potential ('expectation')
- No somatic input: trivial prediction
- Somatic input: firing deviates from dendritic prediction

→ Adjust dendritic weights to minimize error



Sketch adapted from [2].

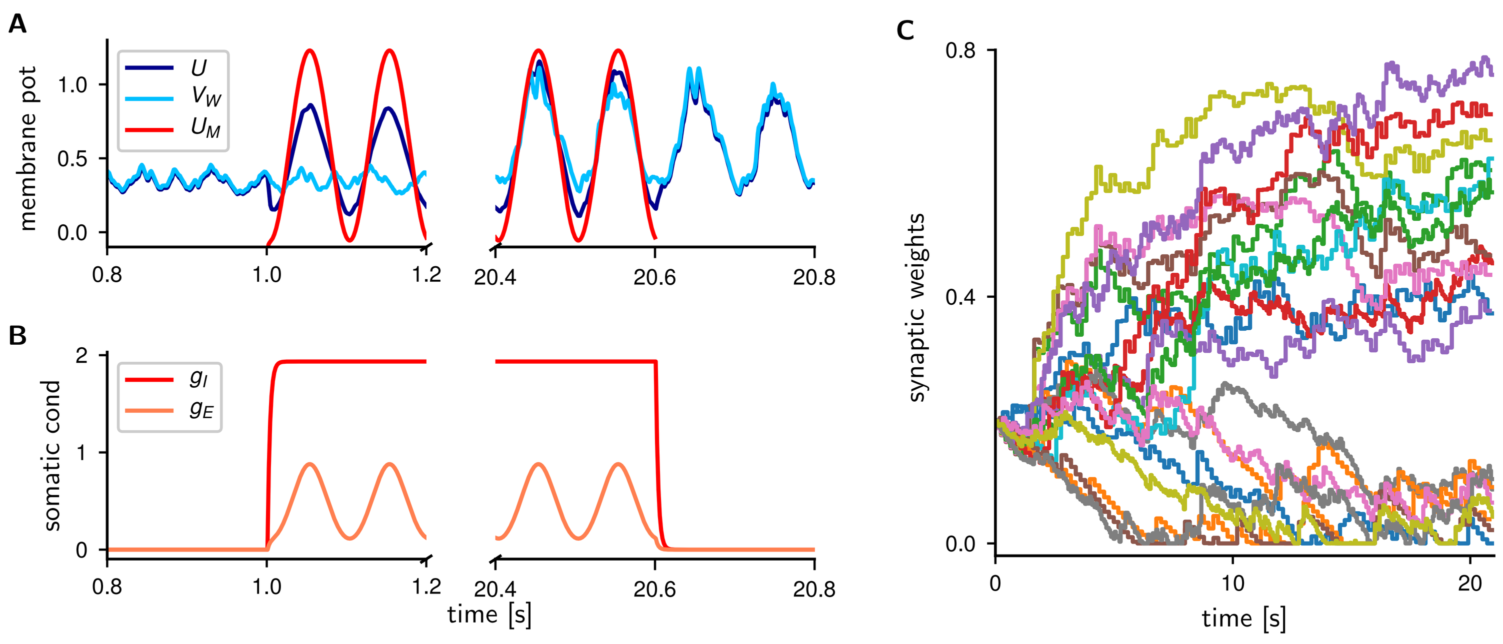


Fig. 4: **A** Simple learning task using the Urbanczik-Senn plasticity rule. Membrane potential of the soma U (dark blue) and the dendrite V_W (light blue). The red curve denotes the nudging potential U_M resulting from somatic input (panel B). **B** Excitatory (g_E) and inhibitory (g_I) somatic conductances that produce the teaching signal. Corresponds to figure 1b in [2]. **C** Temporal evolution of the synaptic weights during learning.

Scaling

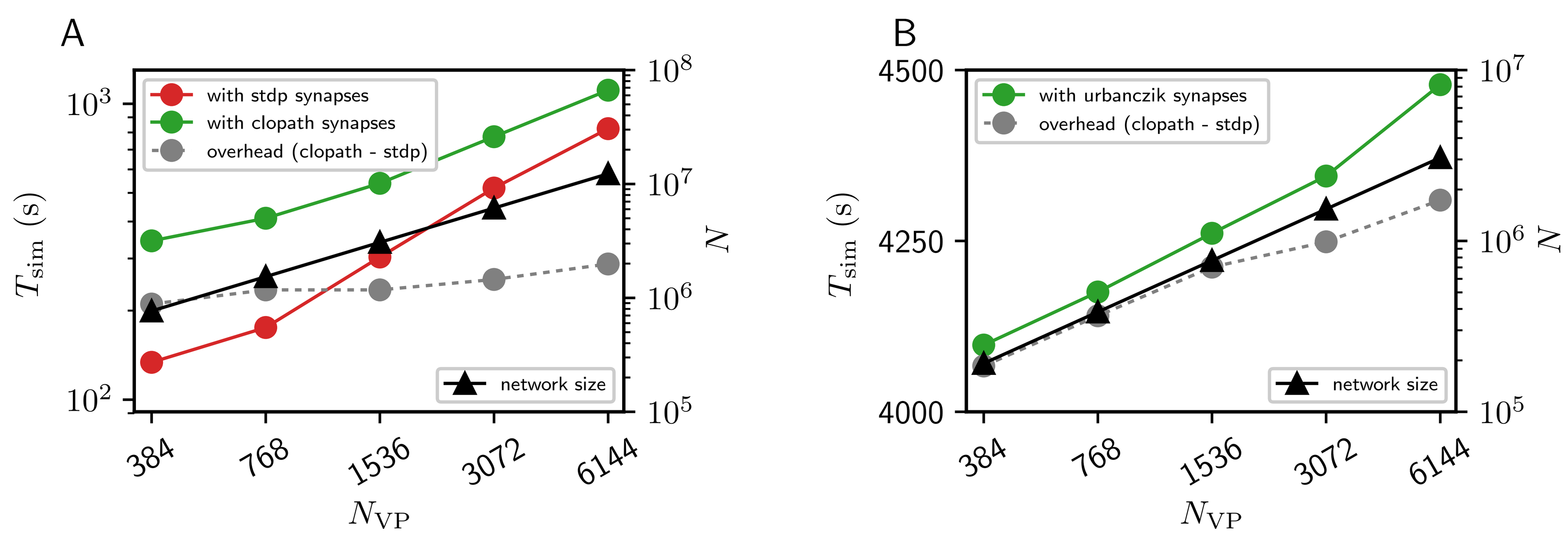


Fig. 5: Comparison of simulation times for the simulation of a Brunel network [3] with i) stdp synapses (red circles) and ii) Clopath synapses (panel **A**, green cricles) or Urbanczik-Senn synapses (panel **B**, green circles), respectively. Gray circles denote the difference in runtime between the two simulations. Since the simulation with the Urbanczik-Senn synapses takes much longer than that with the stdp synapse, the latter is not shown but only the difference. The figure shows results for a weak scaling on JURECA with fixed indegree $K = 5000$. The black triangles indicate the number of neurons N in the simulations.

- Simulations with the Clopath synapse show the same scaling behavior as simulations with stdp synapses
- The additional computations result in a constant overhead in a weak scaling scenario
- Build times for the network are identical compared to stdp (not shown)
- Scaling behavior of the Urbanczik synapse is similar (note the linear scale on the y-axis) but simulation time much longer due to large, consecutive buffers
- Backward summation (see Fig. 2B) is advantageous if spikes are exchanged in strict temporal order

References

[1] C. Clopath, L. Büsing, E. Vasilaki, W. Gersnter (2010): Learning by the Dendritic Prediction of Somatic Spiking. Neuron, 81, 521 - 528

[2] R. Urbanczik, W. Senn (2014): Learning by the Dendritic Prediction of Somatic Spiking. Neuron, 81, 521 - 528

[3] N. Brunel (2000): Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. J. Comput. Neurosci. 8, 183–208.

[4] J. Sacramento, R. P. Costa, Y. Bengio, W. Senn (2017): Dendritic Error Backpropagation in Deep Cortical Microcircuits. arxiv:1801.00062v1

[5] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, W. Maass (2019): Biologically Inspired Alternatives to Backpropagation Through Time for Learning in Recurrent Neural Nets. arxiv:1901.09049v2

Acknowledgments: This work was partly supported by the Exploratory Research Space seed funds MSCALE and G:(DE-82)ZUK2-SF-CLS002 (partly financed by Hans Herrmann Voss Stiftung) of the RWTH university; the Jülich-Aachen Research Alliance Center for Simulation and Data Science (JARA-CSD) School for Simulation and Data Science (SSD); the Helmholtz association: Young investigator's grant VH-NG-1028; European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2); Juelich Aachen Research Alliance (JARA).