Gap Junctions Implementation for IO EGLIF

Niccolò Frassetto

June 2025

1 NESTML Documentation

Gap junctions (electrical synapses)

Each neuron model can be endowed with gap junctions. The model does not need to be (necessarily) modified itself, but additional flags are passed during code generation that identify which model variables correspond to the membrane potential and the gap junction current. For instance, the code generator options can look as follows:

```
"gap_junctions": {
    "enable": True,
    "membrane_potential_variable": "V_m",
    "gap_current_port": "I_gap"
}
```

Read more in the official documentation.

2 Repository and Code Availability

The source code for the olivary network reconstruction is available on GitHub at:

https://github.com/FrassettoN/io-gap-junctions-network

2.1 Repository Structure

The repository is organized as follows:

- `configurations/` – Configuration files for BSB and NEST simulations
- `nest_models/` – NESTML source code for EGLIF neuron model and compilation utilities
- `test.ipynb` – Jupyter Notebook for testing network compilation and running simulations

3 Updating `build_models.py`

To enable gap junctions connections, update the `_build_nest_models()` function in `nest_models/build_models.py` by passing the `codegen_opts` argument to `generate_target()` as specified below.

```
generate_target(
    ...
    codegen_opts={"gap_junctions": {
                    "enable": True,
                    "membrane_potential_variable": "V_m",
                    "gap_current_port": "I_gap"
                  }
    }
)
```

4 Modifying Configuration Files

4.1 BSB Connectivity Declaration

Gap junctions are defined using the `FixedIndegree` strategy in the BSB connectivity configuration. This can later be adapted based on physiological evidence.

```
connectivity:
  gap_io:
    strategy: bsb.connectivity.FixedIndegree
    indegree: 5
    presynaptic:
      cell_types:
        - io
    postsynaptic:
      cell_types:
        - io
```

4.2 NEST Connection Model for Simulation

In the simulation block, the `gap_junction` synapse model is explicitly defined. Although the delay is unsupported by NEST for this model, BSB requires it.

```
simulations:
  basal_activity:
    ...
    connection_models:
      gap_io:
        synapse:
          model: gap_junction
          weight: 0.5
          delay: 0.0
        rule: one_to_one
        make_symmetric: True
```

5 Gap Junctions Delay Errors

NEST does not support defining delays for gap junctions, while BSB requires it. Errors are raised depending on whether a delay is defined or omitted.

5.1 Delay Not Defined or Set to Null

```
RequirementError: Missing required attribute 'delay' in {standalone}.gap_io.synapse
```

This error originates from the class `NestSynapseSettings` class definition in `bsb_nest/connection.py`, where delay is configured as a required parameter:

```
@config.node
class NestSynapseSettings:
    model = config.attr(type=str, default="static_synapse")
    weight = config.attr(type=float, required=True)
    delay = config.attr(type=float, required=True)
    receptor_type = config.attr(type=int)
    constants = config.catch_all(type=nest_parameter())
```

5.2 Delay Defined

```
NESTErrors.BadProperty: BadProperty in SLI function Connect_g_g_D_D: gap_junction connection
    has no delay
```

This error originates from the implementation of gap junctions in `nest_simulator/models/gap_junction.h`, where any attempt to set a delay raises a `BadProperty` exception:

```
void
set_delay( double )
{
  throw BadProperty( "gap_junction connection has no delay" );
}
```

6 Testing BSB-Exclusive Connectivity

6.1 Configuration

Following the `gap_goc` example in `cerebellum/configurations/mouse/mouse_cerebellar_cortex.yaml`, gap junctions were implemented exclusively using the BSB connectivity system. To evaluate whether this implementation produced any effect, we examined how BSB-only connectivity influences inferior olive (IO) and global network dynamics during basal activity simulations. The configuration file `cerebellum/configurations/mouse/dcn_io/dcn_io_vitro_nest_gap.yaml` was updated to include the `gap_io` connectivity specification.

Table 1: Cell Type Summary

Cell Type	**Cell Count**	**Cell Density (cells/µm³)**
IO	12	2.00×10^{-6}

Table 2: IO to IO Connection Properties

Connection	**Nb. Synapses**	**Synapses per Pair**	**Convergence**	**Divergence**
IO ↔ IO	60	1.0 ± 0.0	5.0 ± 0.0	5.0 ± 1.6

6.2 Basal Activity

No significant differences were detected between gap and no-gap conditions during basal activity simulation across all measured parameters: mean firing rate, inter-spike intervals (ISIs), and frequency spectrum. This finding was consistent across all network cell types.

6.3 MF and CF Stimulus

During MF and CF stimulation, no significant differences were observed between gap and no-gap conditions for mean firing rate, inter-spike intervals (ISIs) and frequency spectrum across all network cells.

6.4 Modified Ie Parameter with MF/CF Stimulus

Modifying the Ie parameter from -18.101 to -9 during MF/CF stimulation did not reveal any significant differences between gap junctions and no-gap junctions conditions either.

7 Possible Next Steps

1. Update the local `bsb-nest` installation to permit synapse definitions without delay parameters.
2. Modify the local `gap_junction.h` file to enable delay parameter definition.
3. While awaiting the official `bsb-nest` update, conduct a comprehensive review of IO EGLIF parameters.