Using TimeDelNetSim

# Files

The File Names in TimeDelNetSimMEX\_Lib are:

## Headers (Headers\) –

1. FiltRandomTBB.hpp
2. MexMem.hpp
3. Network.hpp
4. NeuronSim.hpp

## Source Files (Source\) –

1. MexFunctionInterface.cpp
2. ParallelStructuresII\_mex.cpp

## Matlab Codes For Testing (MatlabSource\) –

1. CompleteRandomNet.m
2. NeuronSim.m
3. NeuronSimMEX.m
4. RecurrentNetwork.m
5. RecurrentTesting.m

The File Names in TimeDelNetSimMEX\_Exe are irrelevant for now as that portion of the code has not been updated with the rest. When that happens, we will see what to do about it.

# Variable Categories

## Input Variables –

These variables are variables that are only relevant at the time of input in that their value does not get altered during the course of the simulation. These however, need to be given at the beginning as they are relevant parameters to the simulation.

## State Variables –

These variables are variables that change during the simulation and whose value is essential in the computation of the next simulation step. Of course in order to calculate the next simulation step, it is required to have the correct values of all State Variables and Input Variables.

State variables occur as a part of the input, processing, and output of the simulator. As part of the input they correspond to an initial condition in case one wishes to continue a simulation from a particular state. They are obviously a part of the processing. They may also be a part of the output if one requires to store / view the state of the simulation or any of the state variables of the simulation.

## Output Variables –

These are variables that are a function of State Variables and Input Variables. These variables cannot be used to simulate the next step but may correspond to quantities of interest that we may wish to observe.

# Matlab Interfacing

The Visual Studio project is configured with Release\_Lib and Debug\_Lib modes to build a mex file for the same. The input and output argument list is defined below.

## Input –

The input is a MATLAB Structure with fields corresponding to the Input and State Variables. The field names for the Input and state variables are given below.

At a minimum, the network must be completely specified as in the neuron parameters, and Sorted Edge List (Sorted lexicographically as (NStart, NEnd)).

The Initial Conditions (This corresponds to initial values to the State Variables) may be partially specified. Each State Value has a default initial value in case it is not specified.

If one wishes to not specify a field in the input structure, one may leave it blank (zero sized vector) or simply not define it.

The Input Struct has the following Fields (note that the names of these fields are case sensitive and if you make a mistake here you will unleash demons)

## Output –

There are two Output Structures that are returned –

### StateVarStruct:

This is a structure containing State Variables that are requested to be returned. Each Field of this will be a vector / matrix that contains the value of The State variable across all time instants for which it has been requested.

### OutputVarStruct:

Structure whose members store the value of Output Variables over the time instants for which it is requested.

# C++ Interfacing

## Data Structures Used in Simulation.

### Neuron

This is the struct that encapsulates the parameters of the Neuron Model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Field Type | Field Description |
| 1. | a | float | Izhikevich Neuron Parameter a |
| 2. | b | float | Izhikevich Neuron Parameter b |
| 3. | c | float | Izhikevich Neuron Parameter c |
| 4. | d | float | Izhikevich Neuron Parameter d |

### Synapse

This is the struct that encapsulates relevant details of the Synapses.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Field Type | Field Description |
| 1. | NStart | int | Presynaptic neuron |
| 2. | NEnd | int | PostSynaptic Neuron |
| 3. | Weight | float | Weight of Synapse |
| 4. | Delay | int | Delay of Synapse in time steps |

### XorShiftPlus::StateStruct

This is the Struct that encapsulates the state of the random number generator.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Field Type | Field Description |
| 1. | w | uint32\_t | RNo generator state variable w |
| 2. | x | uint32\_t | RNo generator state variable x |
| 3. | y | uint32\_t | RNo generator state variable y |
| 4. | z | uint32\_t | RNo generator state variable z |

### BandLimGaussVect::StateStruct

This is the struct that encapsulates the state of the bandlimited random Gaussian vector generator.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Field Type | Field Description |
| 1. | Generator1 | XorShiftPlus | RNo generator 1 |
| 2. | Generator2 | XorShiftPlus | RNo generator 2 |
| 3. | Values | MexVector<resTyp> | RNo vector values at current time |
| 4. | alpha | float | Value of alpha used in filtering |

In this, resTyp is the data Type of the random number generated. Either float or double.

## I/O Data Structures

Corresponding to the above MATLAB Structs, we have the following correspondingly defined C++ Structs. Note that heading represents the actual Type-name of the respective structure. These structures are the medium by which we give and retrieve information from the simulation function. The Explicit relation between the fields of the structs in C++ to those of the structs in MATLAB are discussed below.

### InputArgs

This is the structure that stores all the information that is required as an input to the simulation function.

### StateVarsOut

This is the structure in which the outputs corresponding to the State Variables are returned.

### OutputVars

This is the structure in which the outputs corresponding to the Output Variables are returned.

# Variable Information

The Following Correspond to the Input, State and Output Variables. This is basically the description of the quantities that correspond to these variables. The variable name is also exactly (case sensitive) the field name of MATLAB InputStruct field corresponding to the variable (naming convention ahead). DON’T MAKE A MISTAKE HERE.

NOTE: There is no detail about the types of these variables or the details regarding the vectors or scalars corresponding to these variables. Just Description and Name. These details are furnished in the section regarding the MATLAB and C++ Interfacing.

## Input Variables –

|  |  |  |
| --- | --- | --- |
|  | Variable Name | Description |
| 1. | onemsbyTstep | Number of simulation time steps in one millisecond. |
| 2. | NoOfms | Number of milliseconds for which the simulation has to run. |
| 3. | DelayRange | The maximum synaptic delay in milliseconds. |
| 4. | a | (vector of) Izhikevich Neuron Parameter **a** |
| 5. | b | (vector of) Izhikevich Neuron Parameter **b** |
| 6. | c | (vector of) Izhikevich Neuron Parameter **c** |
| 7. | d | (vector of) Izhikevich Neuron Parameter **d** |
| 8. | NStart | (vector of) Pre-synaptic Neuron Indices (beginning at 1) |
| 9. | NEnd | (vector of) Post-synaptic Neuron Indices (beginning at 1) |
| 10. | Weight \* | (vector of) Synaptic Delays |
| 11. | Delay | (vector of) Synaptic Delays |
| 12. | InterestingSyns | (vector of) Synapses whose weight we wish to output. |
| 13. | OutputControl | Variable Selection parameter of Output Control |
| 14. | StorageStepSize | Storage Step Size parameter of Output Control |
| 15. | StatusDisplayInterval | Number of simulation steps between status displays. |

**\*** Weight is technically a State Variable (in light of STDP) however, it is a necessary input as a part of network specifications and thus is included here just so. Will also be re-mentioned in the State variable List.

## State Variables

|  |  |  |
| --- | --- | --- |
|  | Variable Name | Description |
| 1. | Weight | (vector of) Synaptic Weights |
| 2. | V | (vector of) Neuron Voltages |
| 3. | U | (vector of) Neuron U’s |
| 4. | Iin | (vector of) Neuron Internal (From Spike Kernel) Input Currents |
| 5. | Time | Time instant of simulation (in Simulation Steps) |
| 6. | SpikeQueue | Spike Queue of simulation |
| 7. | CurrentQIndex | Index in SpikeQueue of spikes that are to arrive |
| 8. | LSTNeuron | (vector of) Last Spiked Timings of The Neurons |
| 9. | LSTSyn | (vector of) Last Spiked Timings of The Synapses |

## Output Variables

|  |  |  |
| --- | --- | --- |
|  | Variable Name | Description |
| 1. | Weight | (vector of) Synaptic Weights of **Interesting Synapses** |

# Adding a new State Variable

The Following Editing steps must be completed if one introduces a new state variable. Note that the following steps occur while editing an existing state variable too.

## Input / Output data structure related

1. Add a corresponding variable in InputArgs (WITH CONSTRUCTOR)
2. Add a corresponding variable in InternalVars (WITH CONSTRUCTOR)
3. Add a corresponding Matrix in State Struct (WITH CONSTRUCTOR)
4. Add a corresponding Matrix in Single Struct (WITH CONSTRUCTOR)
5. Add corresponding field names to MATLAB output structs (Remember to change the Number of Fields variable):
   1. Input Struct
   2. State Struct
   3. Single Struct

## Input / Output routines related

1. Add a corresponding Output Control Option in Enum.
2. Add a corresponding Output Control Option.
3. Add corresponding code to the 3 initialization routines in
   1. InternalVars
   2. StateVarsOutStruct
   3. FinalStateStruct
   4. InitialVarsStruct
4. Add corresponding code to the output routines in
   1. InternalVars::DoSingleStateOutput
   2. InternalVars::DoOutput
   3. InternalVars::DoFullOutput
   4. InternalVars::DoSparseOutput
5. Add corresponding code to the mex output routines
   1. takeInputFromMatlabStruct
   2. putStateToMatlabStruct
   3. putSingleStatetoMatlabStruct

## Computation Related

1. Nothing Specific. Just that do your shit carefully here
2. Create Alias in ParallelSimulate.
3. In the event of editing a variables name, ensure some level of   
   consistency.
4. Since this is state variable storage. No Computation except   
   type conversion may be done in the output routines

# Adding An Output Variable

## Full Procedure

1. Add the given variable to OutputVarsStruct.
2. Add OutputControl Option and Enum option (if necessary).
3. Add Initialization Routine to OutputVarsStruct.
4. Add computation code of This variable (this maybe done next step   
   too)
5. Add code to DoSparseOutput.
6. Add code to DoFullOutput.
7. Add Field Name to output Matlab struct (CHANGE STRUCT SIZE).
8. Add code to putOutputToMatlabStruct.

# Yo