

Using TimeDelNetSim

1 FILES

The File Names in TimeDelNetSimMEX_Lib are:

1.1 HEADERS (HEADERS\)-

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

1.2 SOURCE FILES (SOURCE\)-

1. MexFunctionInterface.cpp
2. ParallelStructuresII_mex.cpp

1.3 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.

2 VARIABLE CATEGORIES

2.1 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.

2.2 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.

2.3 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.

3 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.

3.1 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. InterestingSyms	(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.

3.2 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.	lin	(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

3.3 OUTPUT VARIABLES

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

4 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.

4.1 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)

4.2 OUTPUT –

There are two Output Structures that are returned –

4.2.1 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.

4.2.2 OutputVarStruct:

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

5 C++ INTERFACING

5.1 DATA STRUCTURES USED IN SIMULATION.

5.1.1 Neuron

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

	<i>Field Name</i>	<i>Field Type</i>	<i>Field Description</i>
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

5.1.2 Synapse

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

	<i>Field Name</i>	<i>Field Type</i>	<i>Field Description</i>
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

5.1.3 XorShiftPlus::StateStruct

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

	<i>Field Name</i>	<i>Field Type</i>	<i>Field Description</i>
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

5.1.4 BandLimGaussVect::StateStruct

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

	<i>Field Name</i>	<i>Field Type</i>	<i>Field Description</i>
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.

5.2 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.

5.2.1 InputArgs

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

5.2.2 StateVarsOut

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

5.2.3 OutputVars

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