Release notes for simulator version 1.0.29
August 28, 2020

Bug Fixes

1. Fixed a problem that I have seen intermittently that was difficult to reproduce. Sometimes after editing a model which included deleting nodes and/or adding new nodes, after saving and reloading the model there would be connectors that did not connect to anything. It was also possible that a connector belonging to a node that had been deleted was connected to a new node, which was not the intention of the modeler. I finally discovered a way to reproduce this at will, determined the problem, and fixed it.

2. Fix error when upgrading newsned .snd files to the usfsim format where the length of the simulation was not correctly adjusted. The simulation length was displayed in the global parameters page as zero, but when running the simulation, the number of steps was treated as seconds, resulting in very long simulation run times, perhaps for hours.

3. Drag and drop in the BDT and Plot tables was broken. The dragged row was dropped, but it was a copy operation, not a move operation, so you wound up with duplicate rows.

4. In simviewer, the fiber number for fiber plot rows as always 1. The correct fiber information was plotted, but the text was not always correct.

5. If the plot table had a non-existent cell or fiber number in it, simrun would provide random values for magnitude and spike events. Now a flat line is always returned.

New Features

1. Plasticity in the form of Hebbian learning has been added to the simulator. The MacGregor program LRSYS30 is the basis for this. This is the SYSTM11 program that is the basis for the simulator with Hebbian learning added. The details of how to use this are in the User's Manual.

Briefly, you add a Learning synapse type to the synapse list. Depending on the parameter values, it can be a synapse that increases or decreases in strength when it learns. You pick this type as the connector between fiber and cell populations. Several parameters control the learning process. When a fiber or cell fires, it adds the event to a history list for the synapse, which includes an arrival time in ticks. The arrival times are decremented each tick. Entries are deleted from the list when the arrival time of an event has passed. When the target cell fires, it consults this list and if it finds an entry that is within the arrival time window, it rewards the synapse by modifying the strength by a small amount until it hits the limiting value.

There are a lot of variations on how to model "forgetting", or adjusting the strength towards the initial value. There are suggestions that even if the strength is adjusted as determined by some set of conditions, it perhaps should not adjusted completely to the initial value, but to an intermediate value. In a sense, the synapse is predisposed to learn more easily after an initial learn process.

I am certain there will be several iterations of the plasticity feature.

The new package includes learn_demo.snd and learn_demo.ols, which are located in /usr/local/share/usfsim on Linux systems. The Windows version includes learn_demoWIN.snd and learn_demoWIN.ols. These will be in whatever directory you unzip the simulator zip file.

Something I did not include in the manual because I expect it to change shortly is this: Hebbian learning has the problem that in can only adjust the synapse strength in one direction. Once the value hits the limit, it stays there forever.

There are a variety of methods to address this problem. Almost all of the articles/papers I read were too technical for me. They assumed physiological and mathematical knowledge that I do not possess. In addition, I could not find any formula in any of the articles I read that looks like the MacGregor formula.

One method that did make sense was that if a cell fires and there is no history, adjust the strength towards the initial value. The current version of the simulator does this. It sort of works, but not really. If the cell never fires, no adjustment is made so it remains stuck on a fixed value. This suggests that the adjustment should be a function of perhaps just purely time. Feedback from modelers who use this feature would be most instructive.

Release notes for simulator version 1.0.28
June 24, 2020

Bug Fixes

1. If you create a standard fiber population, send a member to the plot table in the Launcher window, and then change the population to an afferent fiber type and saved the model to a .snd file, simbuild would crash when loading the model file. This has been fixed.

A problem in newsned and simbuild is that changes to a population in the model designer does not update the bdt or plot tables in the launch window. This has been addressed somewhat, in that type changes are propagated to the plot view table when the Apply button is clicked. Some changes are not. If you have a population of 100 members and send one to the plot view table, by default a random member is selected, e.g., 83. If you change the population size to 50, the plot view table is not affected. The program will run correctly, but since there is no longer a member 83, there will never be any events plotted for it.

New Features

1. On laptops and smaller monitors, some portions of the simbuild window are not visible even if the window is maximized. In Linux, it can be resized and the top of the window can be moved above the top of the monitor. Windows will not let you do this. The non-menu, non-toolbar portion of the GUI is called the central widget. Simbuild now creates a control called a scroll view and the central widget is is added to it. When required, scroll bars are added to the GUI and the user can scroll hidden portions of the GUI into view. The default size of the central widget is 1900 by 940 because this shows all of the controls when the window is maximized on my monitor. Since the drawing area has adjacent scroll bars, I altered the appearance of the outer ones slightly to differentiate them.

If the mouse pointer is over the vertical or horizontal scroll bar, the mouse scroll wheel will scroll the central widget.

If portions of the central widget are still not visible with the default width and height, you can adjust these numbers with an optional geometry argument when starting the simbuild program from a command line window, like so:

```
simbuild -g 1900x1000
```

The windows shortcut is this:

```
"C:\path\to\your\simbuild_folder\simbuild.exe" -g 1900x1000
```

There is a space before the "-", after the "g", the "x" must be lower case, and no spaces in the numbers. The first number is the width and the second is the height and both must be present. You may have to experiment with the numbers. It is likely the width is okay and you may need to increase the height value.

Release notes for simulator version 1.0.27
June 17, 2020

Bug Fixes

1. Fix bug in Windows related to "File is in use" message. Windows treats deleting a non-existent file as an error, Linux does not. Simbuild previously treated that error as a "file in use" error. It now checks explicitly for a "file in use" condition and ignores the non-existent deletion error.

2. There were a couple of paths through the code setting up for a launch that could leave out a couple of lines of text from the sim file depending on the options selected. A lot of combinations: use wave files, don't use wave files, make a bdt file, don't make a bdt file, include analog info, or not, and others. etc. Some depend on others, it does not make sense to select analog and not create a bdt file. I think all combinations of the cases are now properly handled.

3. A model with just a fiber population in it crashed simrun in random ways. It assumes there is at least one cell and one connector in a model and did not check to see if this is true, which resulted in uninitialized pointer values. You can now have a non-crashing simulation with just one fiber population.

4. Simbuild and simviewer save some state information, such as window size and location, when these applications close. If these are open when the operating system is shut down, it was possible for incorrect information to be saved, such as a window size of 0 because the window had already been removed from the GUI before the settings were saved. Settings are no longer saved if the applications are running when shutting down. Any changes will not be saved in this case. It is better to close the applications, then shut down the system. I thought of adding an explicit "Save Settings" menu item, but that is just one more thing to have to remember to do.

New Features

1. I compressed some more of the controls in simbuild, particularly on the parameters pages. We are working at home on laptops and systems with smaller monitors. I tried to make everything visible at the cost of a rather cluttered display. This is hard to test because I do not have a test lab with many system configurations. It also varies depending on the font. Send me screen captures if controls or text fields are being clipped or overlapped. In at least one case, one control was not visible at all. The GUI is getting rather crowded as features are added. An option is to put a block of controls into a scrollable container. If the contents are not all visible, scrollbars are provided.

2. The big addition to this version is a prototype implementation of incorporating external afferent signals into the model. You create one of these as a fiber population. It is visually different because it is a square made of two lines. One of the fiber type options is Afferent. You get to pick a spike2 file that is the source of the signal. Simrun assumes the first ADC channel in the file is the one to use. It is best to use spike2 to export a single channel to an smr or smrx file and to time shift it if required so tick 0 is the first sample in the file.

The Big Idea is this:

Instead of a fixed firing probability, simrun varies the probability of firing as a function of the value of the signal at each simulation tick. You control this by setting up a range of values and associate a firing probability for each. Think of these as points in the plane. You may have something like (0,0.0), (40,0.01), (100,0.2), (150,1.0), (180,0.2), (200,0.0). This means e.g., that a signal value between 0 and 40 will have an interpolated probability between 0.0 and 0.01.

The table on the Afferent Fiber parameter page can have up to 32 rows. This is arbitrary, it can be more if you want them but it needs to be a fixed number that we choose as part of the prototyping cycle and stick with. The values in the table can be any range of floating point numbers and must be in monotonic ascending order. They do not have to be entered in order, simbuild will sort the table when you click on Apply. The probabilities must be between 0.0 and 1.0. The values do not have to be equally spaced. You can greatly vary the probabilities over a narrow range and also have larger ranges where the probability is the same.

When you send an afferent fiber population member to the plot table in the Launcher window, you get several choices on what should be plotted. The default is member events. You can also choose just the signal or events and the signal.

In my testing, I used a blood pressure file from one of our experiments. I arranged it so lower values had a lower probability and larger values had a larger one. You can also invert the probabilities so the opposite is true.

You can add as many afferent population as you want and use the same file and construct different probability profiles for each. You can use different files with other signal sources.

The file name convention is that if you pick a .smr/.smrx file in the same directory as the .snd file, the path is not saved and simrun assumes the file is in the current directory. This lets you share models with others independent of how your local file system is structured. It also allows you to duplicate a directory without having to make changes to the model. If you want to share the same .smr/.smrx file among multiple models, put in in a different directory. If the path to the file is not the same as the path to the .snd file, the entire path will be saved and will be used by simrun. This is not as portable as assuming all files are in the current directory, but it does avoid having many copies of what could be a large file. The file name control is editable, so if you really do want the full path to the current directory saved to the .snd file, you can type it in manually.

It was quite a bit of work to sync the external signal with the simulator. By default, the simulator runs at 2000 Hz. Our recordings are 25,000 Hz. That is 12.5 BP sample ticks per simulator tick. The opposite can also be the case. Simrun up or down samples the incoming signal. The goal of the code is that a sample taken at second 3 of the afferent signal should be the same as the simulator's second 3.

The spike2 scripting language can be used to generate may types of waveforms at any sampling rate and saved to a file that the simulator can use. Processed signals of existing recordings could be used.

The ultimate goal is to communicate with an external program that is simulating some physiological subsystem and which provides the information to the simulator in (more or less) real time. The details of this remain to be defined. You may pick a program, some networking information, a URL, it could be many things.

This is the first cut at a prototype and there are no doubt improvements to be made. I am particularly suspicious on how I implemented the up/down sampling. In the BP file I used, readings tended to change very slowly in just 12 or 13 simulator ticks, so as a first approximation, I think it is usable.

The next step after this would be to become a signal source to another simulation. I plan on making that a variation of the Cell setup and draw a circle with two lines. This will probably have it's own parameter page. It is conceptually different in that it exports information to an external sink, I don't think it does any calculations on its own, but it will have one or more connectors from the simulation, so, not really sure. All to be defined.

3. I needed a way to test various sampling rates, so I wrote a utility that generates a spike2 file that is a single channel with a sine wave in it. This is a command line program. By default the sample rate is 2000 Hz, the length is 60 seconds, and the frequency of the sine wave is 3 Hz. The name of the program is makesine. There is a Windows versions that is included in the Windows .zip file. This is the help:

```
Usage: makesine [-r sample rate in HZ]  [-t length of plot in seconds]  [-f frequency in times per second]
```

The curve is translated so all values are positive and range from 0 to 100. The simviewer program and its predecessor assumes that voltage values will be mostly positive. The X axis where Y = 0 is near the bottom of each simviewer row, not in the middle of the row. The spike2 file can be loaded into spike2 and adjusted as required.

Release notes for simulator version 1.0.26
May 6, 2020

Bug Fixes

1. In the launcher window, if you deselected saving a bdt file and an smr file but did select saving an smr wave file, the output file would have a random filename.

2. In lower-res monitors, some of the controls are not visible. The control box above the tables in the launch window has been resized to show the controls. The Undo and Apply buttons in simbuild on all of the parameter pages have been moved from the bottom to the top region. We have more horizontal resolution. It is generally frowned upon to move elements of the GUI around because it will take a bit of retraining to learn the new locations. The trade-off is now you can see the controls. This made the parameter panels a bit wider, but you can always use the splitter to show more of the drawing panel or resize the window so it is part of it is visible on a second monitor. There is some inconsistency in the placement. Depending upon the real estate available in each parameter panel, sometimes the buttons are one the same row and sometimes in the same column.

3. If you have sent items to the bdt or plot tables in the launcher window, and if you edit an existing model and change a normal fiber population to an electric stimulation population, the member number for the one row of the population number is set to 1. If there are multiple members, the rest are not changed. Fiber populations have only one member, and if, for example, the only item from the population was, say, 34, there would never be an event for that member because it does not exist. It is not important in the plot table, but entries in the bdt should be unique because some down-stream tools break if this is not the case.

There are lot of ways to produce unexpected results, such as creating some table entries for a fiber population with 500 members in it, adding some items to the launcher tables, then changing the number of members to 10. It is likely there are table entries with no-longer-valid members. The amount of software effort to check for errors like these often outweigh the benefits.

4. The Fit To Screen function in simviewer now does what it claims to do.

New Features

1. Simviewer now displays the comment text for fiber populations. Since cell and fiber populations are numbered independently, you can have cell population 1 and fiber population 1. A "C" or an "F" is prepended to the population number.

2. The Send To Launcher buttons on the Cell and Fiber parameter pages are managed differently. It there are unapplied changes, the buttons are disabled. This is because the changes you see in the panel

are not what would be sent if you were able to click on one of the Send buttons, which will lead to confusing and unexpected result. You need to Apply or Undo the changes before sending an item to the launch window tables.

Release notes for simulator version 1.0.25
April 27, 2020

Bug Fixes

1. In simviewer, when changing the time marker values, the marker movement was jumpy and obviously inaccurate. The cause of this is that the graphics package we are using uses a floating point coordinates and the coordinate calculations were being done as integers. Now they are done as floats, which results in smoother line movements and a more accurate display. There really are points (and pixels) between (4,1) and (4,2).

New Features

1. Fiber populations can be sent to the plot table and the results can be seen in simviewer. These are plotted as events. If creating a spike2 file, they are saved as event channels.

There were some changes to the some of the files such that older versions of the simulator package will not know what to do with some of the information and may show fiber populations as cells, or perhaps not at all.

Release notes for simulator version 1.0.24
March 27, 2020

Bug Fixes

1. In simviewer, when using wave files instead of a network connection, a typo in the line of code that checks to see if a new wave file has been written causes the check to always fail. Wave files would never be displayed by the simviewer. Fixed.

New Features

1. There is an option in the simbuild launcher window to save the waveform data that is plotted by the simviewer program to a spike2 file. Action potential plots use two spike2 rows because there is no obvious way to have waveform and event data in the same row. The first row shows action potentials and the second shows the waveform data. All other plot types use a single row and show the value at each tick. The interpretation of what the value represents varies with the plot type.

Spike2 limits the row labels to 9 characters, so longer label text from simbuild is truncated.

Release notes for simulator version 1.0.23
February 25, 2020

Bug Fixes

1. Fix a crash in simviewer when trying to zoom an empty plot.

New Features

1. The model can be saved to a PDF file. This was already part of the code, but for some reason there was no menu item to select. It is in the File menu. Like simviewer, it saves what you see in the model graphic panel. You can save part, or all of a model.

2. A feature was added to the Options Menu to switch the model to monochrome mode. It displays the background as white and all foreground objects as black. The intended use of this is to save a pdf that can be incorporated in a document or to print a model without using all the black ink in the toner cartridge.

If you want to print a complex model on multiple pages, load the pdf file into Adobe reader, select poster in the print setup, and set the scale to a larger value, such as 150%. It will print the model on multiple pages that you can cut and tape together.

3. Minor text change in a few menu items. For example, "Load .snd File" is now "Load .snd Model File." A couple of people have asked, "What is an snd file?"

4. The XY cursor position in simviewer is now always displayed. There is also some text that explains what the numbers at the top of the drawing area are, which are the left time, right time, and width of display in seconds. Changing the time slider will change this. You can also resize the window and see the width in seconds change.

Release notes for simulator version 1.0.22
February 11, 2020

Bug Fixes

1. Deleting rows in the bdt and plot tables in the Launcher Window could produce a corrupted pointer with undefined results. This would only happen when selecting File-New and if there was more than one row in either table.

New Features

1. Add optional different paths to input and output files to simrun. These are only available from the command line. This was added to support the SIMCORE/OSPARC environment. Input and output files need to reside in different folders.

Release notes for simulator version 1.0.21
January 16, 2020

This release fixes some connector line problems in the simbuild graphical editor.

Bug Fixes

1. It breaks things if an incomplete connector is saved as part of a model. There were several places where it was possible to do this. A check and a warning dialog has been added to stop this from happening. You must either delete the incomplete connector or finish connecting it before, for example, saving a file.

Which brings us to:

2. If you deleted an incomplete connector, when you moved the node it was attached to, the program crashed. This has been fixed.

3. Another way to create an incomplete connector is to edit an existing one and move the start or end point to the edge of an object's hot-spot (dotted lines). There is a case that is hard to duplicate where it is possible to move the end a pixel or two outside the hot-spot. When loading a model that has instances of this, the program moves the line, or adds a new one, to ensure it is inside the hot-zone. There was an error in this code that sometimes failed to move the point into the hot-zone. This has been fixed. When this happens, a warning is printed in the info box in simbuild and the model is marked as "needs to be saved." The user should save the corrected model in these cases.

Release notes for simulator version 1.0.20
January 6, 2020

New Features

1. The user's manual has been brought up to date. In the simbuild Help menu, selecting "View Manual" displays this manual. Selecting "View Older Manual" displays the newsned manual. The newsned program assumes that the user will edit text files in situations where using the GUI is awkward, or does not support an operation. The updated user's manual deletes a lot of content that assumed this. Users wishing to edit the .snd, .sim, .ols, and other files in a text editor should consult the older manual.

2. The Linux programs use a plotting program plotmv. This is a twenty year old program that has not been maintained since 1995. The default is gnuplot, which is now the only option. A Windows version of this is available, but it takes more work, such as porting Linux shell scripts to Windows powershell scripts. This work remains to be done.

Bug Fixes

1. When using simbuild's File -> Save Sim File, the corresponding .ols file was not saved. Now it is.

2. Each run of the simulator creates / overwrites several files. A few of these used the launch number as part of the file name. Some did not and would be over-written by the next launch, resulting in unpredictable behavior. Now all of the files generated by each launch have the launch number as part of the file name.

3. Related to this, there was a bug in a Linux script where each launch would over-write the power spectrum file.

4. When using the network connection to send simrun results directly to simviewer, and if simrun completes a simulation faster than simviewer can display it, some of the output results are never displayed. The wave file method does not have this issue. The reason for this is that data written to files are buffered and when the file is closed, the data is written to the file even if the application terminates. Network sockets buffer the data, but any pending data is discarded when the socket is closed or the application terminates. Simrun now sends an End Of Data packet to simviewer at the end of a simulation and then waits for simviewer to send an acknowledgment that is has received this packet. In effect, it tells simrun it has all the results and that it is okay for it to terminate.

This could also occur when running more complicated simulations for a very long time.

Release notes for simulator version 1.0.19
November 25, 2019

New Features

1. In the Launch Window, Create smr File has been added to the left panel. If selected and if there are rows in the bdt table, a .smr file will be created that CED's Spike2 program can read. The bdt file for Launch #0 is always spawn0.bdt. The corresponding .smr file is spawn0.smr. In spike2, if you select View -> File Information, the date the .smr was created and the .snd file used are in the information dialog box. The simulator assigns channel numbers starting at 101. It assigns all of the cell populations first, then the fiber populations. The .smr file also contains the original cell or fiber populations from the model.

On Windows, if the user is viewing the .smr file in Spike2, Windows will not allow the simulator to create a new file because the file is in use. The user should close the file before starting a new run. The simbuild program will display a dialog box warning the user.

2. In support of this, you can sort the bdt table in the Launch Window by clicking on any header. The sort will list cells first, then fibers. The population numbers and member are then sorted in either ascending order or descending order. You can rearrange the rows by hand. If you list Fibers before Cells, the display order in scope and Spike2 will not correspond to the visual order. The program does not automatically sort the table when new items are added.

3. Add a Find menu item and a single choice, Find. Selecting this pops up a list that contains the cell and fiber populations and the Comment text. Clicking on the Find button ensures that the population is visible and selects it. The parameter page for the population is also displayed.

4. Add the current Launch number selected in the Launch Window to ChangeLog files.

5. A command line program that converts existing bdt or edt files to smr files has been included in the package. The Linux version is edt2spike2, the Windows version is edt2spike2.exe. To run it, open a command line window. On Windows you should use the cd command to change to the program's directory. The Windows command is:

```
    edt2spike2 -n c:/path/to/yourfile/somefile.bdt
```

If the bdt file is in the current directory, you do not need to type in the drive or path. The program will create a .smr file that spike2 can read.

Bug Fixes

1. Toggling the Add Axon/Synapse button from the On to Off state did not change the internal state, so the drawing state remained "drawing a connector." Fixed.

2. Fixed a bug where some lung row values in the Launch Window were being cleared out, which caused everything after that row to fail to plot.

Release notes for simulator version 1.0.18
October 24, 2019

1. The vertical time markers were not correctly displayed. For .bdt tick counts, there were twice as many markers as there should be. Selecting a marker space 1000 milliseconds apart displayed markers every 500 milliseconds. The fix takes the tick size into account, so the marker spacing is correct for .edt, .bdt, or custom tick counts.

2. The infinite loop bug fixed in version 1.0.16 occurred in another place in the code. Fixed.

3. Simviewer has a new scaling option. It is called Anti-clip Scale. If this is selected, the Vertical Gain Scaling Factor is used to scale all of the row plots. If the value of a row plot would over or under shoot the row's box, the gain for that row is adjusted to not let this happen. The alternative is to let the plots expand into neighboring boxes, which is the default, or to clip the plot so parts of it are not displayed, which has not been implemented. This and the Auto Scale option are mutually exclusive.

4. The Users Manual has been been updated, but additional work remains to be done.

Release notes for simulator version 1.0.17
October 11, 2019

1. New functionality: All communication between the simbuild, simrun, and simviewer programs are via the network interface. Simbuild still writes spawn[0-14].txt and spawn[0-14].sim files to the local disk, but sends the contents via the network. This is in anticipation of how the SIM-core interfaces between simulation software might be structured. On possible configuration is to use Docker containers. The typical interface between a service (such as simrun) in a container is via a network connection. It is also possible to share files between containers, but it can get complicated with multiple users. But, in the absence of specific information, who knows? In principle , we could run simbuild and simviewer on one computer and run simrun on a more powerful system which could be running under a different operating system. I made a version that did this to see if it works, and it does. The controls required to do this have been hidden in this version. It takes quite a bit of system administration work to configure the systems.

2. Enhancement: When selecting a cell or fiber population, the population number is displayed next to the optional group name on the parameter pages.

3. Bug fix: When selecting a line that has other connector lines underneath it, a popup dialog box is display so you can select one of the connectors. You can click on the column headers to sort the rows. This was broken and how now been fixed.

4. Bug fix: If you selected a connector and change the type, this was not flagged as a change and you would not be prompted to save or undo the change if you selected another connector. This has been fixed. This had the side-effect that the color of the connector was shown (incorrectly) as the color of the new type. The (correct) color of the current type was not displayed until you selected the connector a second time.

5. Bug fix: A line of code in the build_network function was included inside a debug block. The effect is that it would not execute unless debugging was turned on. The effect of this is dependent on the model. In some cases, it would never execute even when correctly include. Since it allocate memory and returned a pointer, I would expect simrun to crash or produce random simulation results. All copies of version 1.0.16 should be deleted from your systems.

Release notes for simulator version 1.0.16
September 26, 2019

NOTE1: Starting with this usfsim release, all release notes are contained in a single file. Versions will be in most-recent-first order.

NOTE2: The last formal release was version 1.0.11. There were several internal releases that were used for troubleshooting some program defects specific to the Windows version.

1. An electric stimulation fiber type has been added (abbreviated e-stim.) To use this, add a fiber population to the model, then select the new fiber item in the drawing. The Fiber Parameters page now has two options, Normal and Electric Stimulation. Some of the parameters are the same as the Normal type. The parameters specific the e-stim are:

> Deterministic or Fuzzy: If deterministic, e-stim is applied at exactly the same time as determined by the frequency. If fuzzy, e-stim is applied at a random time centered on the deterministic time.
>
> Frequency: How often the stim is applied, in Hz.
>
> Fuzzy Frequency Width: If fuzzy, the stim will be randomly applied sometime in the millisecond interval centered around the deterministic time.

Note that there is not a Fibers In Population parameter. The e-stim fiber type has a hard-wired population of one.

2. Units have been added to many of the spinner controls.

3. Several parameters that were set in tick counts are now set as milliseconds or seconds. The tick counts represent time values, but the actual value will vary if you change the tick interval. If you want to have a start time at one second and select a tick value of 0.5 ms, the start tick count is (doing math in head) 2000 ticks. If you change the tick value to 0.1, 2000 ticks is no longer one second. The presentation of these parameters is now in floating point seconds, so, for example, if you want a start time of 150 milliseconds, you would enter 0.150 seconds. These are converted back into tick values when the simulation is run. The program knows that older model files have the values in tick counts, and converts them to time values using the tick value and the step interval. The file will be updated to the current format when it is saved.

4. You could not view the ChangeLog.txt file if a model has not been loaded. Now you can, but this is only the case for a file named ChangeLog.txt. If you are using the model file name as a basename for a change log file, you still have to load the model first. Of course, these are text files, so can be viewed in any text editor.

5. By default, the launcher window is always on top of the simbuild window. If you want to look at the simbuild window, you have to move it out of the way. There is now an option in the Options menu to toggle this, so you can click on the simbuild window and have it be on top of the launcher window. The windows are independent in this mode, so you have to bring the Launcher window to the foreground by hand. When in the always-on-top mode, bringing the simbuild window to the foreground also brings the launcher window to the foreground. Clicking on the Open Launcher button in the simbuild window will always bring the launcher window to the foreground.

6. The program remembers the last parameters for a cell or fiber population. When you add a new cell or fiber population, these will be the defaults. Previously, the default was always a fixed set of values.

7. When adding an item to the bdt table in the launcher window, a random member of the population is generated. If the number has already been selected, a different random number is generated. If all of the members of the population have already been used, the program hung in an infinite loop. Fixed.

8. The internal tables used when creating and editing the launcher bdt and simviewer tables were not cleared out before copying the current values to the tables. This resulted in deleted rows still being sent to the simrun program. Fixed.

9. Fiber populations have Time to Begin and Time to End parameters. If you do not want the firing to end, you had to enter a value that was larger than the simulation time. If you increased the simulation time, you may exceed the End value. Now, if you select an end value of -1, either explicitly or by using the down arrow to pick the next value less than zero, the program will display "Never". Regardless of the value of the simulation run time, the value for the end time sent to the simrun program is guaranteed to be larger.

10. Several issues that were Windows specific have been fixed. Many of these had to do with using the wrong character in path names. Linux expects a '/' character, Windows expects a '\' character.

11. It is possible when moving the connector end points in a drawing to have the coordinates outside of the dotted circle or rectangle that represents the object's "hot zone." When loading a model where this has occurred, the program failed to make the final connection to the object and the connector was in a state where it could not be selected. If this occurs, the program now moves the endpoint to be inside of the "hot zone." It prints a warning message and marks the drawing as changed. The user will be prompted to save the file if they want to run a simulation, load a new file, or close the program. The user should save the adjusted drawing so this does not happen again.

12. Some additional popup tooltip help has been added to some of the controls to make the program more self-documenting.

13. I am working on updating the user's manual. A lot of the GUI-specific information is no longer accurate and there are conceptual changes as well. The current work-in-progress has been included in the package. It is *usfsim.pdf.* For Linux systems it is located in /usr/local/share/doc/usfsim. For Windows systems, it is located in the directory where the .zip file was unzipped.

14. If network errors are encountered in the Windows version, popup dialog boxes are displayed to communicate possible sources of the errors.

Release notes for simulator version 1.0.13
September 6, 2019

This is an in-progress version that has new software that may assist in diagnosing problems running on Windows. There are some pop-up dialog boxes and text that is printed in the text output controls in simbuild and simviewer.

Screenshots of the dialog box and a copy and past operation of the text in the text controls can be emailed to dshuman@health.usf.edu.

The new electric stim fiber type is not completely implemented, so leave that alone.

Release notes for simulator version 1.0.11
August 14, 2019

In addition to some bug fixes and cosmetic changes, this release has four new features.

1. GROUPS

- There is a new menu, Groups.
- A set of cell and fiber nodes can be selected and then assigned to a group.
- Some (or all) of a model can be in a group.
- A node can only be in one group at a time.
- A node can be removed from a group.
- Connectors belong to a group only if both endpoints are in the group, so the user does not need to select connectors.
- Groups do not nest, that is, a group cannot contain another group.
- There is a page in the Help dialog box that has details of how to manage groups.

2. EXPORT AND IMPORT

- Once you create a group, you can export the group to a .sndlib file.
- The parameters, shared connectors, synapse types, and geometric information is saved in the file.
- The population and synapse numbers are adjusted to start at one.
- You can import one or more .sndlib files into a model.
- Population and synapse numbers are assigned using the next unused numbers.
- The intended use is to create functional subsystems that can be reused in new models.
- Once a group has been imported, the user must make connections between
- it and other nodes in the model.

You can select a subset of a model and export it without assigning it to a group. It makes managing the imported nodes more difficult, so in general, use groups for exporting.

I discovered something during testing that should be mentioned. I loaded a simple model, exported the entire model as a group, then imported it into an empty model, expecting the simulation output to be identical. It was not.

The reason why is that each fiber and cell population has a unique random number seed that is internal to the program that the user cannot change. This should not be confused with the random number seeds that the user can change. If you create model one with fiber pop1 and cell pop1 and create a set of parameters and run a simulation, you will get a result that will be the same each time. However, if you create a new model two by adding in cell pop1 and cell pop2, fiber pop1 and fiber pop2, and then delete cell pop1 and fiber pop1 and create an identical model using fiber pop2 and cell pop2, you will always get the same result for each run, but the results will be slightly different from model one.

The colors of the connectors are likely to be different because the synapse type is what determines the color. The synapse type number in the original model may be different when importing a group. The functionality remains the same, but it may be a bit confusing visually. If the imported types are identical to existing types, the user can change the type.

3. WAVE FILES ARE OPTIONAL

- This version uses network sockets for communication between simbuild, simrun, and simviewer.
- The newsned program wrote the simulation results to a series of wave files.
- This can create thousands of files.
- By default, simrun now sends the contents of the wave files directly to simviewer.
- There is an option in the simbuild Launcher window to select direct communication or wave files.
- If you do not want to view the results again, used direct communication.
- If you want to review the results at a later time, use wave files.

You can run the simviewer program in stand-alone mode to view wave files. By default, when run with no options, it runs in wave file mode using launch number 0. You can use the window controls to navigate around the file system and pick which launch number you want to view.

4. CHANGELOG NAMES

By default all changes are written to a file named ChangeLog.txt. There is an item in the simbuild Options menu to add the basename of the current .snd file to the file name. If you work with several .snd files in the same directory, you can get a change log file specific to each model.