

NVIZ: An Integrated Environment for Simulation, Visualization and Analysis of Spinal Neuronal Dynamics

Kalpathi Subramanian, Paul Okey and Matthew Miller

Department of Computer Science, The University of North Carolina at Charlotte, Charlotte, North Carolina, USA

David Bashor

Department of Biology, The University of North Carolina at Charlotte, Charlotte, North Carolina, USA

In this article we describe *NVIZ*, an interactive environment for system level investigation of neural function. *NVIZ* combines simulation, visualization and analysis in a single software system that encompasses five biological levels of organization, from ion channels to generated behavior. Once a simulation has been completed, cell-level output data can be visualized as height fields (presenting data from an entire population in a compact representation), or within an anatomical model of cat spinal cord. *NVIZ* provides a set of numerical analysis tools such as histograms, and a variety of 2D plots that permit both population and cell level analysis within the software system. Movement of a visualized limb segment is generated from activity of motoneurons, using a novel new algorithm called *Net Neural Drive*. The *linked* visualizations in *NVIZ* provide a powerful means to comprehend neuronal activity generated in complex models involving thousands of cells. The ability to design or modify neural circuits involving multiple populations, followed by simulation, visualization and analysis promotes rapid experimentation and the ability to digest massive amounts of time-varying simulation data. *NVIZ* is highly *scalable*, in terms of the number of populations, neurons, and limb segments. Using interactive tools, the visualization can be easily customized to focus on neural activity of interest. We demonstrate the application of *NVIZ* to understanding locomotion of a single limb joint, using a central pattern generator model.

Journal of Imaging Science and Technology 49: 505–519 (2005)

Introduction

Tools for system level investigation of nervous system function include imaging techniques (functional MRI, PET), MEG, EEG and large-scale simulation. An understanding of how the nervous system produces movement requires the spatial resolution of functional MR imaging and the temporal resolution of EEG. Simulation can be used to satisfy these competing demands. The variety of behaviors the nervous system may produce, coupled with its interneuronal complexity, makes it difficult to conceptualize the relationship between circuit and dynamics. Movement is not represented in a simple way in any region of the nervous system.^{1,2} Although much of the experimental data on spinal neural function comes from studies of single cells, observable behaviors are produced by coordinated interactions among neuronal populations. The population view leads us in the direction of seeing how individual neurons interact to form collective units of function. Large-scale simulation helps us understand how the spinal cord produces movement by bridging the gap between neurophysiological data collected one cell at a time and the population view, which shows how individual neurons interact to form collective units.

Simulation of large-scale neural models results in the generation of significant amounts of time-varying data for analysis and interpretation. While computational expense and resources were a major issue in the past, the availability of powerful desktop workstations has made it possible to handle not only more realistic models of neurons (containing thousands of compartments and multiple channels), but also networks of neurons. Over the years, a number of software systems have been developed to model and analyze neurons and networks of neurons, as illustrated by the survey in Ref. 3. Existing packages such as AXONTREE,⁴ NEURON⁵ and NEMOSYS⁶ model and analyze single neurons, while GENESIS⁷ and NODUS⁸ can accommodate networks of neurons. Most of these packages support 2D graph plots, and in some instances (such as AXONTREE), 3D neuron structure representations. However, the number of system variables, neuron populations and scalability to multiple muscles or joints makes analysis tasks laborious, and has thus motivated our use of interactive visualization tools⁹ that can best deal with the large spatio-temporal datasets.

Many experiments involve manipulation of spatially organized inputs, such as stimulation of a particular sensory or motor nerve. Anatomical representation of such inputs provides spatial context and facilitates relating simulation to experimental results. This was the motivation to add anatomical model representations within *NVIZ*, which also makes it easier to communicate

Original manuscript received June 10, 2004

Corresponding Author: K. Subramanian, krs@unc.edu

©2005, IS&T—The Society for Imaging Science and Technology

with colleagues.¹⁰ We are aware of only one other dynamic spatio-temporal visualization of spinal motor activity, the recent work of Yakovenko et al.¹¹; in that work, locations of 27 populations of hind limb motoneurons were digitized using published data (36 sections of Fig. 28 in Ref. 12). Activity of cells during locomotion was animated using EMG data, generated from the literature of normal cat locomotion. Note that cells in their visualizations were animated from “output data” (EMG) rather than from a model of the central pattern generator, as is the case in *NVIZ*.

A considerable body of work has contributed to understanding the spatial topography of brain function.^{13–15} Accurate localization of brain function can help understand the flow of information among the different cortical regions, fusion of information,¹⁶ and thus illuminate the complex relationships between structure and function. In many of these studies, neural tracers were injected into specific brain regions followed by analysis of the labeled elements.^{16,17} The anatomical component of the present study used multiple 2D slices from Ref. 12, containing locations of labeled motoneurons of cat spinal cord. From this dataset, we assembled the three-dimensional representation of the outline of spinal grey matter and discarded locations of individual neurons. Instead we used their dataset to construct bounding boxes within the visualized cord to contain randomly placed cells. 3D reconstruction from 2D images brings up additional issues, such as accurate registration, and, for across-animal comparisons, use of common coordinate systems.¹⁸

Just as accurate visualization of brain activity aids understanding of brain function topography, we expect that visualization of spinal cord anatomy will aid understanding the spinal cord function. It should be noted that data for constructing the cell and population level spinal visualizations (unlike the brain) can be assembled, at least in part, from physiological data sets, because of the known strict segmental, i.e., sequential, arrangement of sensory inputs and motor outputs in the spinal cord.

In this article, we describe the development and application of *NVIZ*, a software system that combines simulation, visualization and analysis within a single computing platform for efficient interactive analysis and understanding of neuronal dynamics in cat spinal cord. The large scale model simulator developed by Bashor¹⁹ forms the core of *NVIZ*. The interactive environment has been expanded from our multilevel visualization work presented in Ref. 9. *NVIZ* now supports the following new capabilities, (1) anatomical model reconstruction of cat spinal cord, with neurons located within and controlled by the simulation, (2) ball and stick views for limb movement, (3) statistical analysis functions (referenced, but not described in Ref. 9), and (4) linking the new anatomical capabilities with the previously existing visualizations. In summary, *NVIZ* provides a powerful combination of 2D and 3D visualization tools that are tightly coupled to simulation and analysis capabilities, all controlled by the same clock. Our system provides an efficient means to explore and analyze large-scale neural models, and currently represents five levels of biological organization, from ion channels to generated behavior.

Methods

NVIZ System Design

Figure 1 illustrates the major components of *NVIZ*. The following sections describe each of the components of this

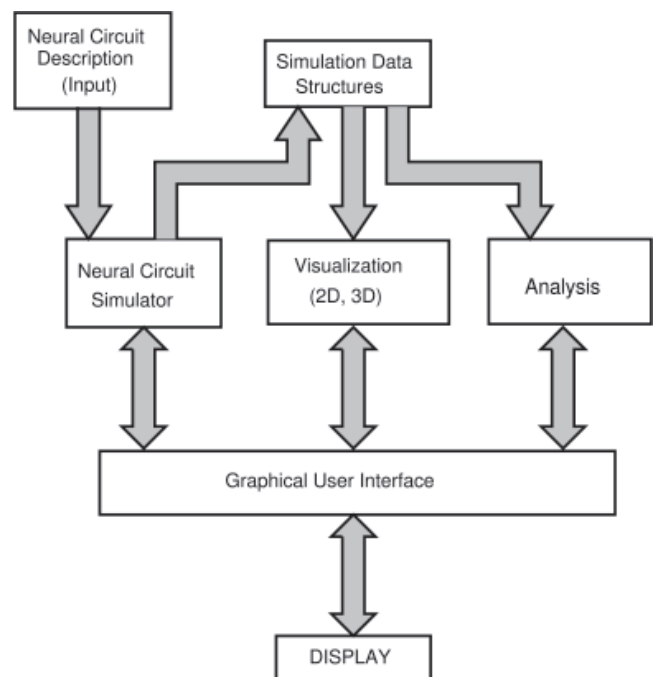


Figure 1. *NVIZ* System Architecture.

diagram. Input to *NVIZ* is a neural circuit description (in the form of a data file) that includes cellular and synaptic biophysical properties and connection patterns among populations. The neural circuit simulator has been adapted from the work of Bashor.¹⁹ That work was based on MacGregor’s²⁰ SYSTM20 population descriptions, PTNRN10 cell algorithms, and SYSTM11 synaptic mechanisms. The present work uses a more complicated cell algorithm, PTNRN21, a two-compartment model, including multiple ionic currents. Data resulting from the simulator are stored in simulation data structures. The interactive visualization and analysis sub-systems use the simulation data. A graphical user interface provides interactive access and control to all the simulation, visualization and analysis features of *NVIZ*. The interface allows easy switching between the different views (height fields or anatomical) supported in *NVIZ*, as well as interactive selection of cell/population level analysis, geometry layout, and view manipulation.

Neural Circuit Simulator

Although the neural circuit simulator is a general-purpose tool, it has been configured in the present application for investigating the spinal motor system. That is, the choice of cell types, interneuronal populations and interconnections are specific to the cord. Other regions of the nervous system could be represented as well, by our system, with appropriate choices of cells, synapses, and population interconnections. Thus, it combines a system-level perspective and the single-unit perspective common in electrophysiological investigations. Motivated by Getting,²¹ the model limits the number of unconstrained variables by specifying first cell input/output characteristics, then synaptic properties, and finally, network connections and appropriate driving inputs. Individual cells (neurons) are grouped into functionally different populations. A circuit diagram, such as that shown in Fig. 2, defines how populations are related to each other.

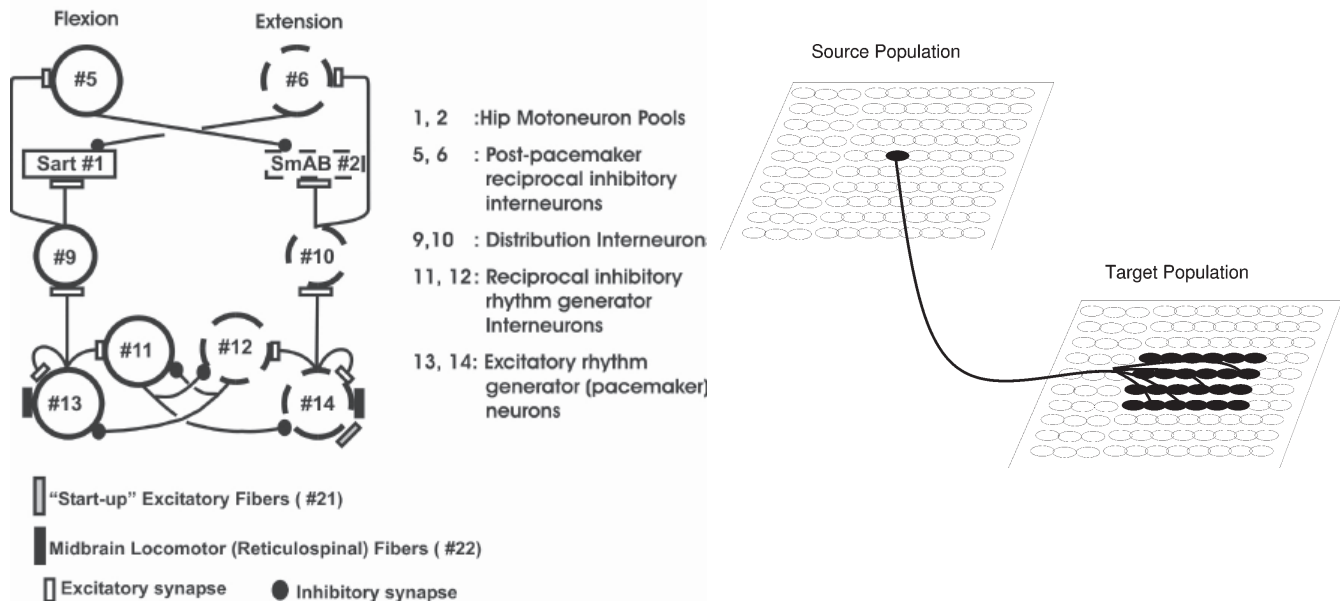


Figure 2. (Left) Central Pattern Generator Circuit for Walking (Right) Synaptic connections from source to target population.

Figure 2 is a hypothetical central pattern generator for walking,²² consisting of 12 populations (10 cell populations and 2 fiber populations). Populations at the bottom of the diagram, 13 and 14, are composed of pacemaker cells. Populations 5, 6 and 9–12 are composed of generic interneurons, and populations 1 and 2 are motoneurons. Each cell type has its own biophysical properties. Excitatory connections are shown by open rectangles, while inhibitory connections are shown by shaded circles and represent tens to hundreds of terminals per sending cell. Fiber populations provide input to the circuit, as indicated by filled and grey rectangles. Operation of this circuit is explained more fully in the Results section. Each population nominally consists of 100 cells, laid out in a 10×10 square grid. To avoid edge effects, a wrap-around algorithm joining the left and right edges, and the top and bottom edges, makes this layout a circular torus. Synaptic connections are defined as the number of terminals and synaptic type from each cell in a source population to cells in a target population. In addition there is a synaptic strength parameter and parameters describing the distribution of the terminals over the receiving population. Figure 2 (right) illustrates a sending cell projecting its terminals to a restricted portion of the target population. System level parameters include the number of populations, number of elements in each population, width and height of the population (which helps define cell location) and the number of synaptic types. A synaptic type is defined by an equilibrium potential and decay time constant. For further details of the connection algorithm and synaptic types, refer to Ref. 19.

The neuron model is based on the potential (E) across two unit-area membrane patches, one representing the neuron soma, the other the dendrite, MacGregor's PTNRN^{21,20,23}. The principal properties of the neuron model are membrane time constant, parameters used to determine the refractoriness following a spike, time-varying threshold, and accommodation properties describing long-term response to input. Details of the two

compartment algorithm and the five neuron types currently in use are described in Ref. 24. The five types of neurons are motoneurons, Renshaw interneurons, generic interneurons, pacemaker interneurons and plateau-producing neurons. Three of these types are illustrated in this study (see Fig. 9 and 10).

The length of a simulation is user-specified; for the walking model, simulation was run for 3-5 sec. of real-time, in steps of 1 ms. At each time step, we recorded the cell soma membrane potential, threshold potential, potassium conductance, spike production and the identity of each firing cell. In the dendrite we recorded dendritic potential, calcium conductance, calcium concentration and potassium conductance. In NVIZ, all of these system variables are held in data structures (memory) for the simulation interval of interest.

Visualization

The types of visualization supported in NVIZ range from single neuron to population views,⁹ to anatomical model reconstructions. Both 2D and 3D visualizations are supported, and more importantly, are *linked*, to facilitate easy switching from one to the other. An important design consideration was to view dynamically changing properties of neuron and system variables.

Dynamic Height Field Views

Height fields were the earliest form of visualization used in NVIZ.⁹ Height fields, in combination with animation, permit system variables to be viewed as a function of time. By spatially distributing multiple interacting populations, it is then possible to visually assess variability among the cells in individual populations, as well as temporal relationships among the different neuron populations. Figure 3 illustrates a height field visualization of the central pattern generator circuit of Fig. 2. 10 cell populations representing a total of 1000 cells are shown. For each population of neurons, three height fields (in gray, green and tan) were defined, and represented geometrically

as a 10×10 grid. They correspond to three cell state variables, membrane potential (E), threshold potential (TH) and spike production (S). With the use of our two compartment model, the membrane potential can represent either the soma or dendritic potential. Alternately, using a pulldown menu, the user can choose to display calcium conductance and calcium concentration, along with spiking. Each neuron was represented by the same grid node in each height field. Thus, as the visualized function values change over the grid, the mesh vertices rise and fall to reflect the changes in the function.

The green field is membrane potential, the excitability level of each cell, updated at each millisecond. The membrane potential of each cell varies with (a) synaptic conductance changes (input) arriving at each time step from other cells or fibers, and (b) time since the last spike in that cell. Thus, the level at which the green field is located represents the probability of spike production. The roughness of the field represents the extent to which cells in the population are in different states of readiness to produce a spike, i.e., the roughness represents the amount of fragmentation in the population. The tan field represents the threshold, that is, the readiness for spike production; when the membrane potential is equal to or greater than the threshold potential, a spike is produced. The threshold for firing depends on the time since the last spike in that cell.²⁴ When the green field touches or crosses the tan field, we know that a spike has occurred. However, spike occurrence is such an important event that it has its own representation, as the gray (lowest) field, an arrangement which most rapidly conveys this information.

Figure 3A shows a particular frame of an animation (time step 2.268 s), with population 2 fragmented, while cells in population 1 are at about the same potential level, but are neither active nor fragmented. The activity in population 2 is generated by activity in population 10 (see Fig. 2), which in turn is produced by activity in population 14, as can be seen in the second column of height fields in Fig. 3A. Other populations that are active in this half cycle are populations 6 and 12, shown in the fourth column.

By default, all populations are laid out in a uniformly spaced rectangular grid. Ideally, all of the populations are visible, without any occlusion. With larger numbers of populations, occlusion can be minimized by utilizing the interactive viewing capabilities (rotate, pan, and zoom) of *NVIZ*. In Fig. 3A, we see the 10 populations laid out in three rows, four populations per row, with the last row containing two populations. *NVIZ* allows the user to customize the population layout by specifying the number of rows and columns, the inter-population spacing along the two dimensions, and selecting the populations to be included in a view, from the list on the left of the window. In Fig. 3A, we selected and included the 10 cell populations corresponding to the central pattern generator circuit of Fig. 2. *NVIZ* also provides a picking capability (the Pick On/Off button below the scrollable list in Fig. 3A); this permits the populations to be individually selected and moved to produce customized views, which can be saved and restored for later use.

In order to make interactive measurements of system variables, a measuring scale is provided with each population. The measuring scale is centered with the starting position of the potential field and its extent is determined by the range of values reached during a

simulation. In Fig. 3B, the view zooms into populations 5 and 11, and their scales have been turned on. In this view, one can see that the threshold potential level is different for the two populations and at about 17 and 12 mV, while the membrane potentials are at about 11 and -5 mV, respectively. The physiological conclusion is that while populations 5 and 11 are hyperpolarized during this phase, cells in population 11 are hyperpolarized to a much greater extent.

Anatomical Model Views

Visualizations constructed with height fields facilitate neural circuit design and experimentation; however, they tend to be abstract, as they lack spatial relationships with respect to the spinal cord. This makes communication with colleagues difficult. In addition, there is no visual representation of the motor output. Thus, we enhanced *NVIZ* to support an anatomically realistic model of the spinal cord, with neurons located at their approximately known locations, and receiving input from the neural circuit simulator. Since our interest in the present work is study of hind limb locomotion, we restricted our anatomical model to the lumbar region of the cat spinal cord. Thus, we digitized the drawings of spinal grey matter from Fig. 28 of Vanderhorst and Holstege,¹² the first 18 of these slices are shown in Fig. 4. Mirror images of these half-slices were then appended to the original slices (see Fig. 2,¹⁰) to form full slices. Using the AC3D²⁵ modeling tool, these “full” image slices were imported as 2D textures and their outlines extracted. Polyline segments were used as the initial representation; each outline had 80 line segments.

In order to construct a smooth model, Kochanek-Bartels interpolating splines²⁶ were used to resample the polylines, as well as create additional slices between the original slices. In this manner the original model of 81×36 voxels (first dimension forms a closed loop in 2D) was resampled to a size of 243×360 voxels. The model is then scaled to the known physical dimensions of cat spinal cord, roughly 3 mm wide, 2.24 mm high and 35.1 mm deep, in this example.

Figure 5 shows several views of the constructed cat spinal cord. The top left panel is a dorsal anatomic view (from the “back” of the animal), with the front of the lumbar enlargement toward the top of the panel. The bottom panel is a view from above and to the right of the cord; front of the enlargement is to the right. It can be seen that there are bumps in the longitudinal views, which are likely registration artifacts of the acquisition process. The butterfly shaped cross-sectional view in the top right panel shows two neuron populations in the interior.

Our illustration focuses on two particular motoneuron populations, *Sartorius* and *Semimembranosus*. The approximate spatial extents of these two populations were estimated from the work of Vanderhorst and Holstege.¹² *Sartorius* neurons extend anatomically from slice 4 through 12 (in red, Fig. 7C), and, when active, cause flexion of the hip. *Semimembranosus* neurons extend from slice 17 through 22 (in green, Fig. 7A), and when active, cause extension of the hip. Two oriented bounding boxes were created within the cord,^{15,18} and 100 small colored cubes, representing neurons, were placed randomly within the box. Each neuron has two states, active or inactive. In the visualizations, the neurons are either fully opaque (active) or transparent (inactive), as can be seen in the top panels of Fig. 7. Labels on the cord indicate the position of spinal roots (L5, L6,

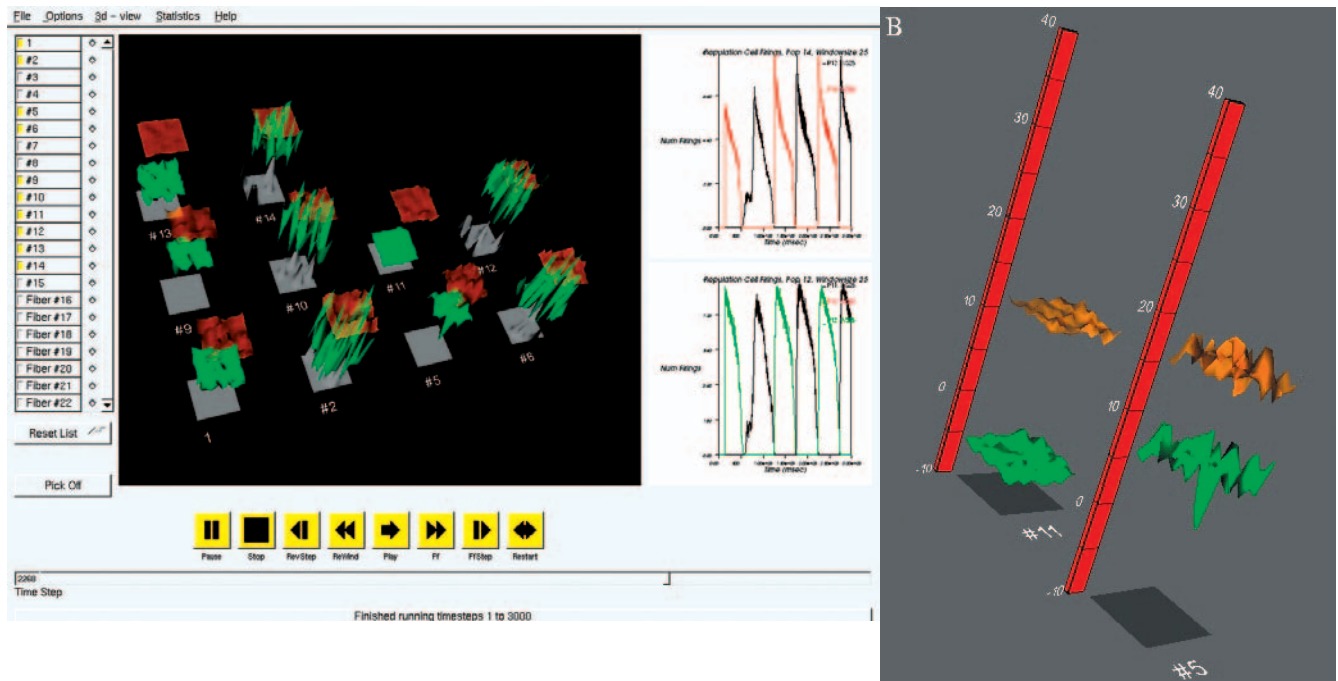


Figure 3. Height Field View (A) 10 cell populations corresponding to the central pattern generator circuit of Fig. 2, time step 2268, (B) Interactive Measurement of threshold and membrane potential for populations 5 and 11.

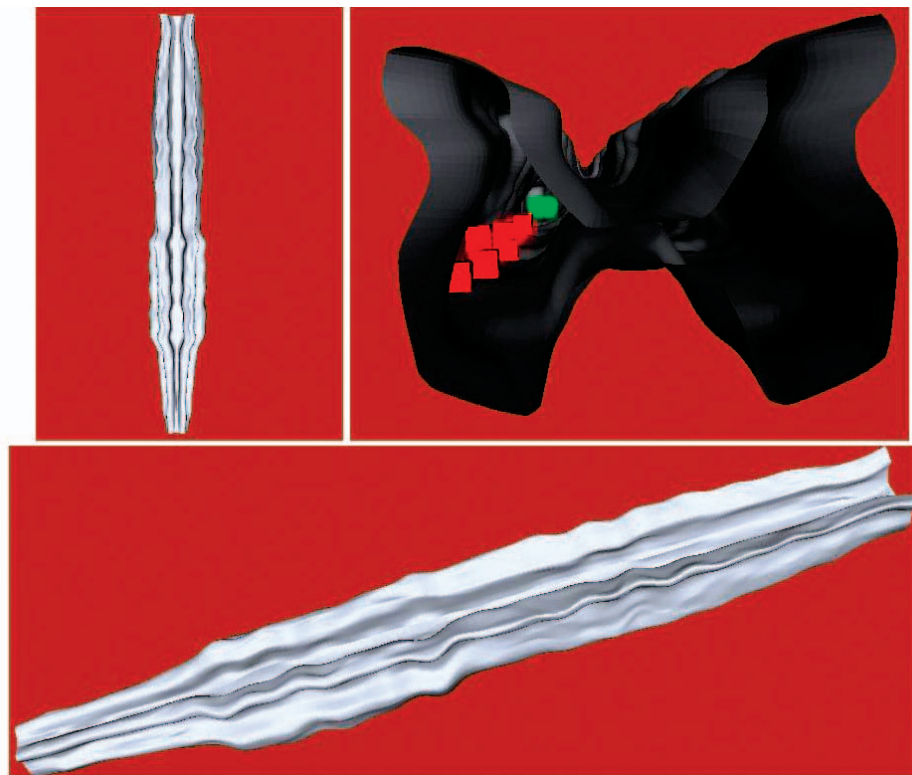


Figure 5. 3D Spinal Cord Model. Example views of reconstructed cat spinal cord model and neurons (colored cubes) located within.

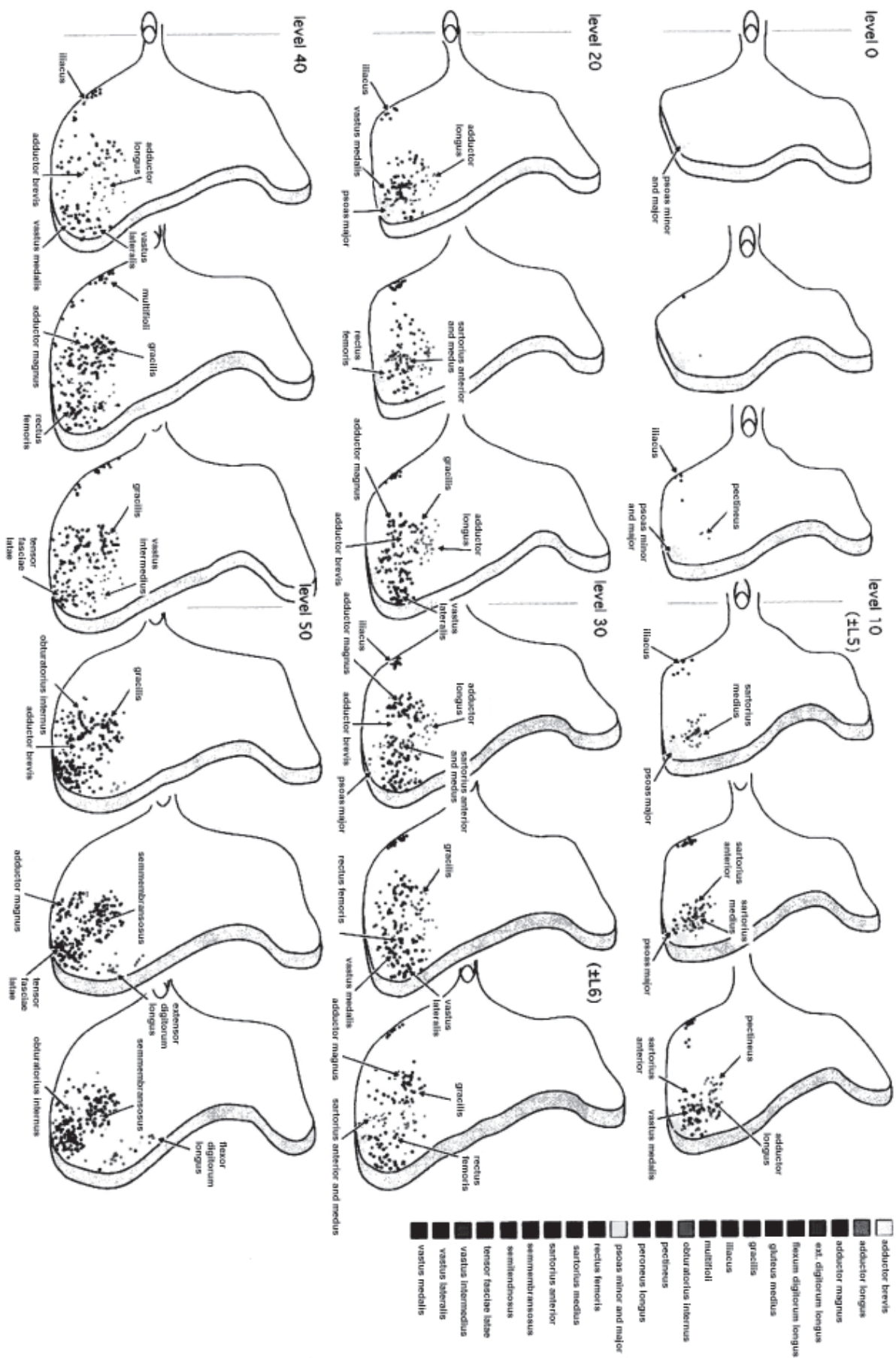


Fig. 28. Summary overview, showing the location of the motoneuronal cell groups of all muscles of groups I–X.

Vanderhorst and Holstege (1997) J. Comp. Neurol. 382: 46-76

Figure 4. Original data for spinal cord reconstruction, slices 1-18, reproduced from Ref. 32.

Algorithm 1: Population Interval Histogram Procedure

Input: population identifier (pop_i), thresholds (TH_{low}, TH_{high}), histogram bin size (bsize)
Output: population interval histogram (pih), interval statistics

```
{
  num bins = (THhigh - THlow) / bsize
  for each time-step t {
    for each cell c in popi {
      if popi → Is Active (c) {
        if c has not previously fired
          record spike timestep t for cell c // first spike for cell c
        else {
          isize = inter-spike interval // t - c0s previous spike time step
          record spike time step t for cell c
          if (isize < THlow)
            pih[0] = pih[0] + 1
          else if (isize > THhigh)
            pih[num bins - 1] = pih[num bins - 1] + 1
          else {
            bin = isize / bsize
            pih[bin] = pih[bin] + 1
          }
        }
      }
    }
  }
  calculate interval statistics: mean, standard deviation, min, max, etc.
}
```

Algorithm 2: Population Cell Firings Procedure

Input: population identifier (pop_i), moving window size (mws)
Output: count of Cells firing at each time step (cell counts[])

```
{
  for each time-step t {
    cell counts[t] = 0
    for each cell c in popi {
      if (ActiveNeuron[t]!CellActive (c, popi))
        cell counts[t] = cell counts[t] + 1
    }
  }
  if (mws > 1) //moving window size > 1
    for (t = mws; t < MaxTimeSteps; t = t + 1) {
      sum cnts = 0
      for (k = t - mws; k < t; k = k + 1) {
        sum cnts = sum cnts + cell counts[k]
      }
      cell counts[t - mws] = sum counts / mws
    }
  Output cell counts[0..MaxTimeSteps-mws-1]
}
```

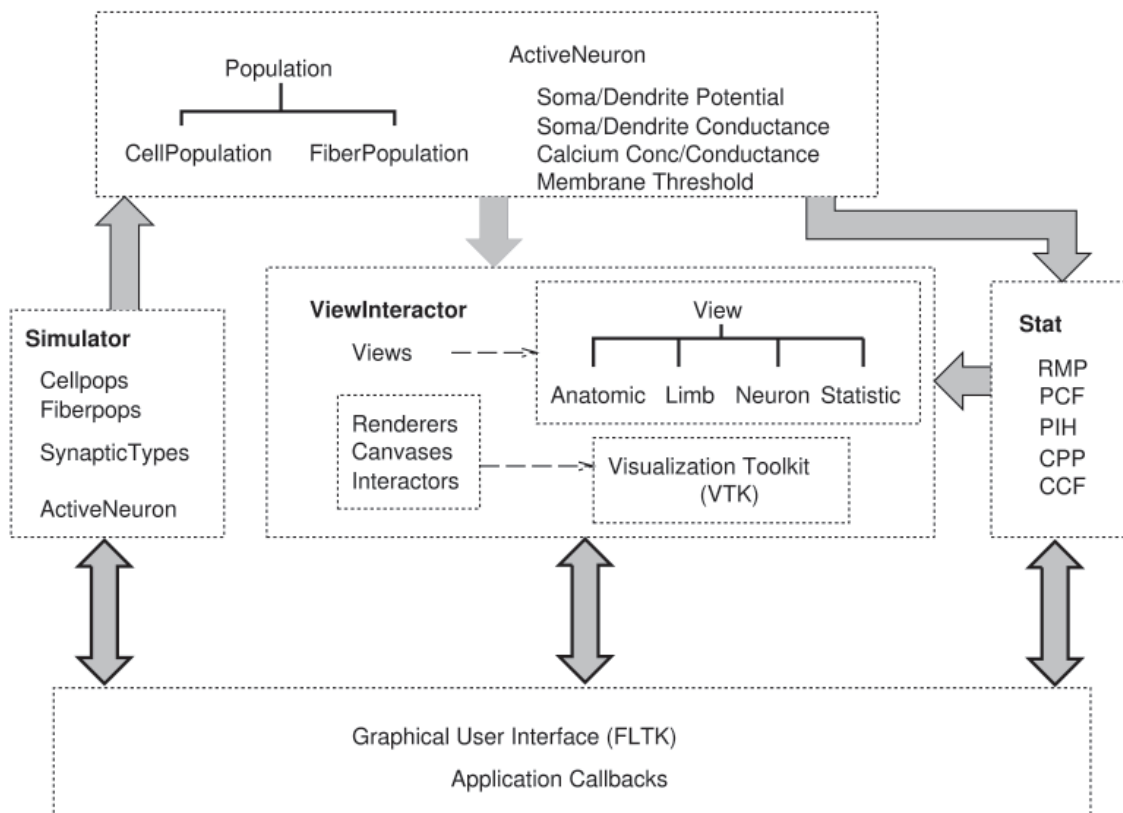


Figure 6. NVIZ Software Architecture.

L7) and relative lengths (as percent) of lumbar enlargement (Level 20, etc.)

Neural Output Views

In addition to the height fields, NVIZ supports three additional views to look at neural output, as illustrated in the panels of Fig. 7, viz., (1) a time-varying output of neural activity of populations represented in the

anatomical model, as seen in the upper parts of the 4 panels of Fig. 7, and (2) limb movement view, seen in lower middle part of each panel, (3) time varying graph of flexor and extensor motoneuron pool output, as smoothed curves of numbers of cells active at each time step. These are discussed in detail in later sections. The time-varying graph view complements the anatomical views in providing more precise temporal information, while the

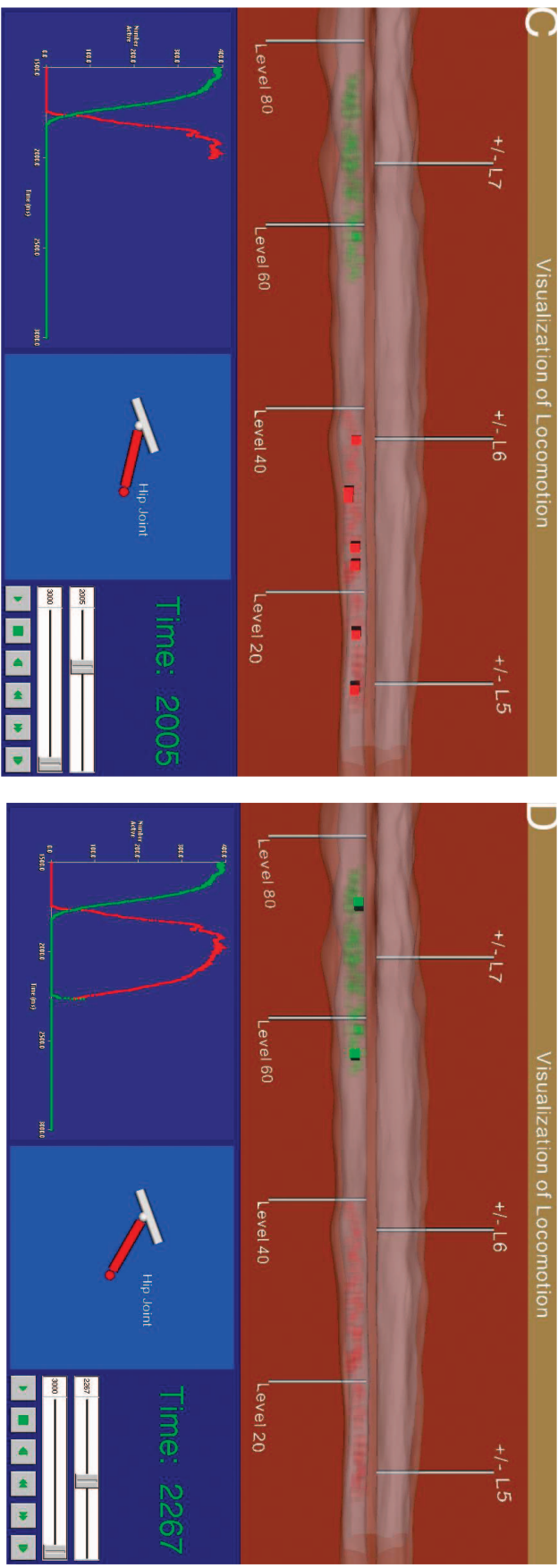
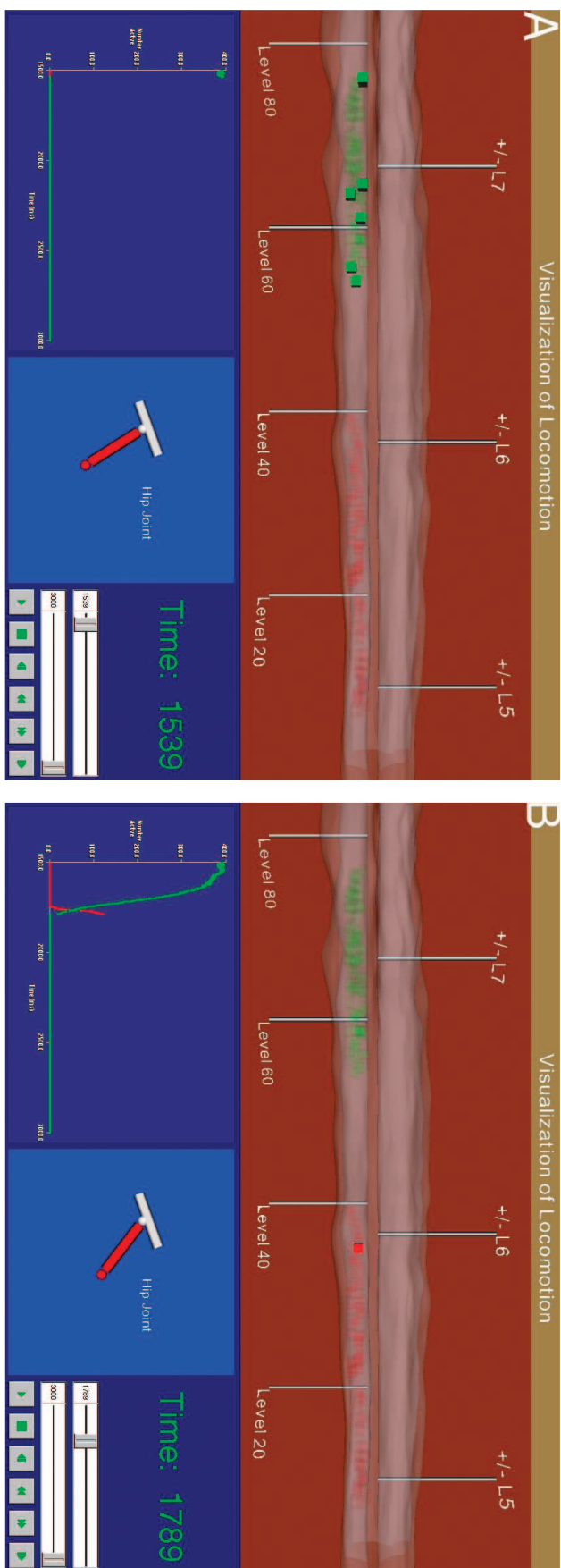


Figure 7. Hip Movement Cycle. (A) Peak of extension phase, limb is extended, about 6 cells active per ms. (B) Transition from extension to flexion, flexor output (red curve) is increasing, thigh elevated. (C) Maximum flexion achieved, thigh lifted, flexor output maximum. (D) Transition from flexion to extension, extensor motor neuron output (green curve) increasing.

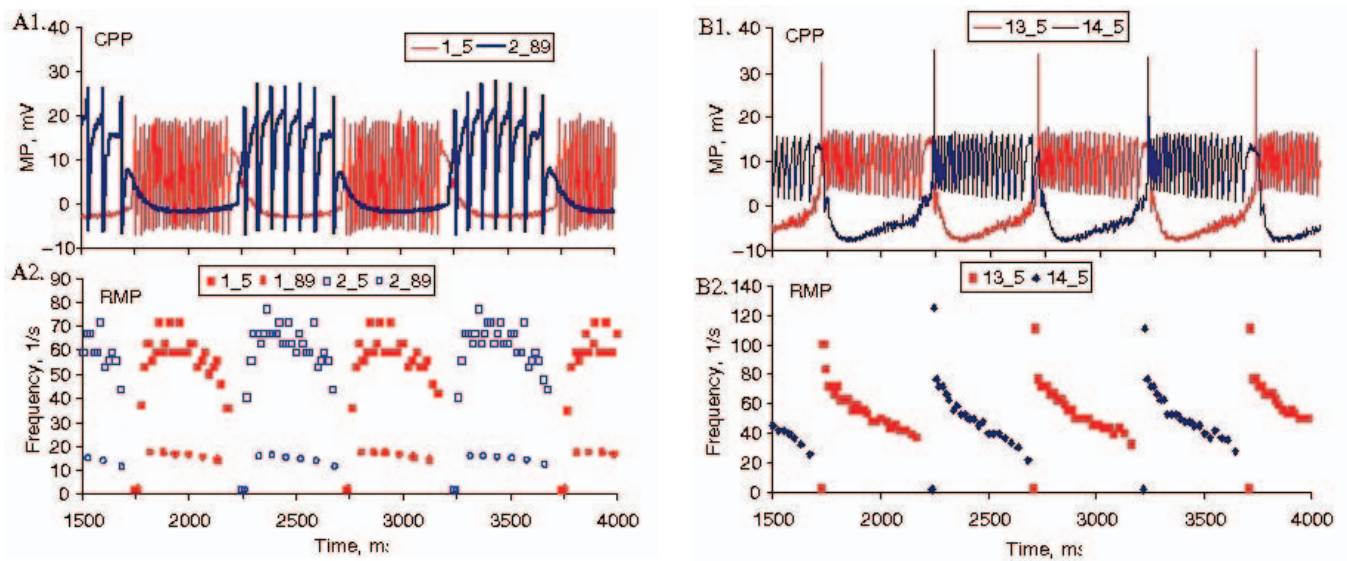


Figure 8. Cell firing patterns of two of the three cell types in the simulation. Panels A1,A2 show records from two extremes of motoneuron response, while B1,B2 are from pacemaker cells. Panels A1 and B1 show cell membrane potential plots (CPP), while panels A2 and B2 show rate meter plots (RMP). Note that panel A2 presents firing rate data for four cells, while A1 shows only two.

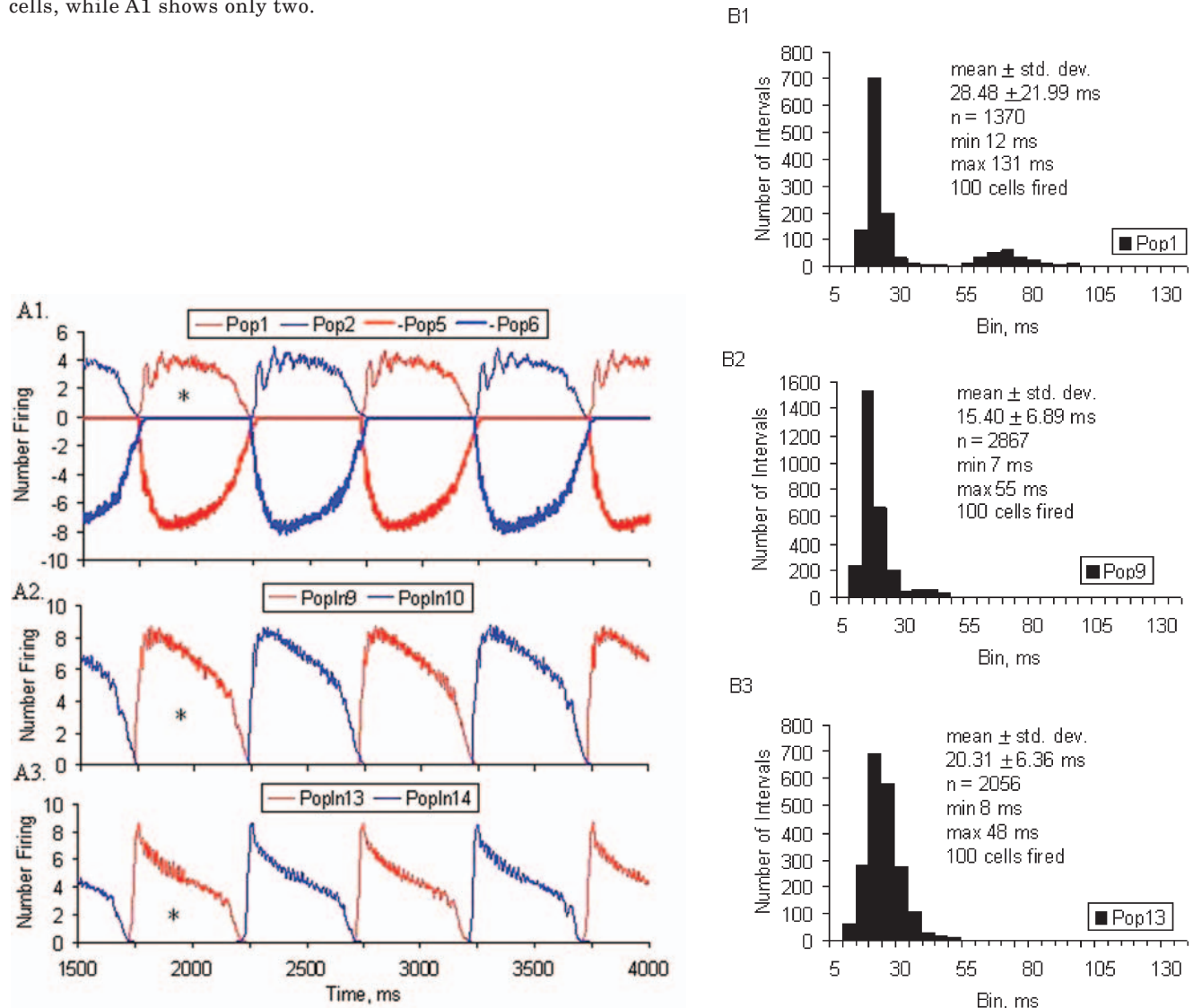


Figure 9. Population response measures. Panels A1,A2,A3 show population cell firing data for six populations in the reference experiment. In panel A1, negative numbers are a plotting convenience only. Panels B1,B2,B3 show population interval histogram data for a single cycle (marked with asterisks) from three of the populations. Downward (negative) plots in A1 are a viewing convenience.

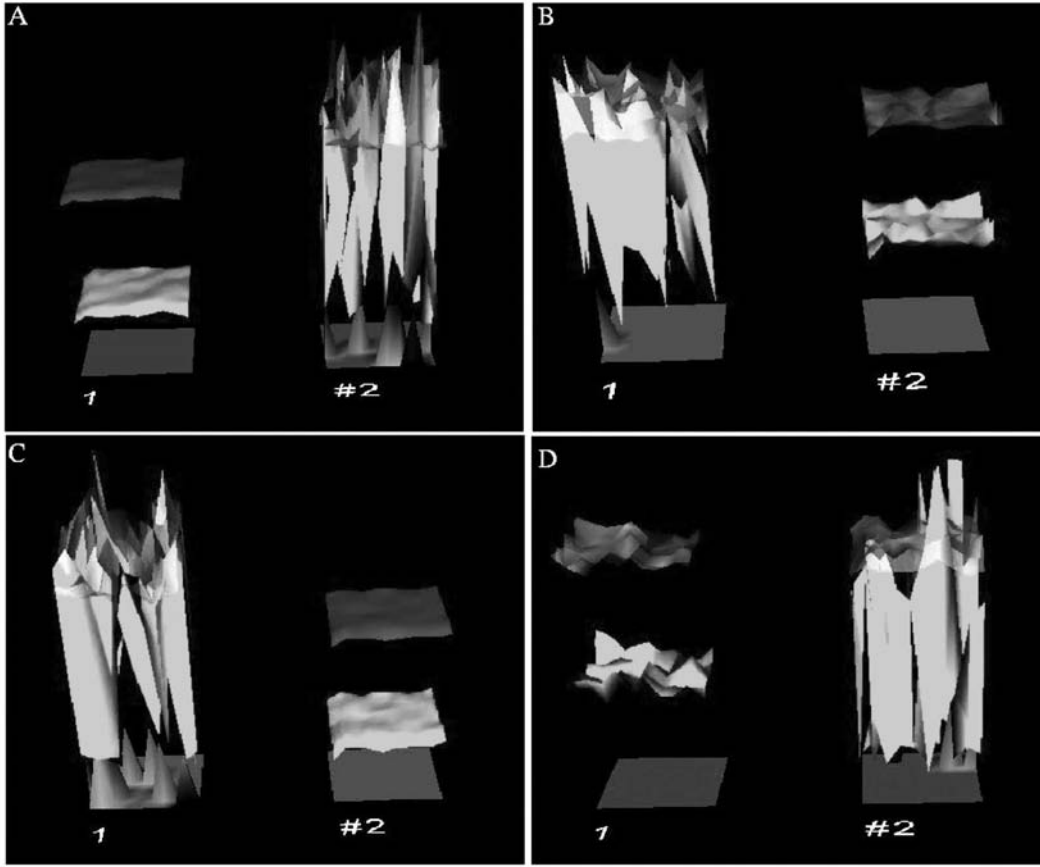


Figure 10. Hip Movement Cycle: Height Field Views (A) Peak extension phase, population 2 (extensors) active; (B) Transition from extension to flexion, population 1 (flexors) active; (C) Maximum flexion; (D) Transition from flexion to extension.

limb movement provides an orientation to the phase, i.e., flexion vs. extension.

Analysis Tools

NVIZ supports a set of numerical tools for analyzing results of a simulation, to compute cell and population characteristics. Currently ten procedures are supported, (1) Population Interval Histogram (PIH), (2) Population Cell Firings (PCF), (3) Rate Meter Plot (RMP), (4) Cell Potential Plot (CPP), (5) Cell Correlation Function (CCF), (6-7) soma and dendrite potassium conductance, (8) dendrite calcium conductance, (9) dendrite calcium concentration and (10) dendrite potential. Functions 6 through 10 are variations of procedure 4 (CPP).

All of these functions may be instantiated from the graphical interface to analyze a simulation. Alternately, a batch file may be specified that contains multiple calls to a particular function for analyzing properties of multiple populations or cells. These functions access and process the data recorded within the simulation data structures.

Population Interval Histogram (PIH)

This function enables the interval between successive cell firings in a population to be recorded and displayed in the form of a histogram. The function expects 4 parameters: the population identifier, pop_i , histogram bin size, $bsize$, and a pair of thresholds TH_{low} and TH_{high} between which the histogram is divided into bins of size $bsize$, given by

$$num_bins = (TH_{high} - TH_{low})/bsize \quad (1)$$

All spike trains from active cells in the input population are scanned and intervals between spikes ($isize$) are tabulated. All intervals below TH_{low} and above TH_{high} are lumped together into two bins, kept at each end of the histogram. Interval sizes between the two thresholds are assigned to the appropriate histogram bin, as shown in Algorithm 1. The procedure only counts intervals that occur after the first cell firing. Thus a cell has to fire twice to produce an interval. In addition, the procedure reports the minimum and maximum interval sizes, mean, variance and standard deviation of the intervals, the total number of intervals and number of cells that produced one or more spikes during this part of the simulation. Algorithm 1 illustrates the PIH procedure and Fig. 9B illustrates an example, discussed in the Results section.

Population Cell Firings (PCF)

The Population Cell Firing (PCF) function reports the number of cells that were active within a population at each time step. As the simulation data structures explicitly store a count of the active cells within each population at every time step, this is extracted by traversing the active cell list of the input population. If this is found, then the number of active cells is output, else there is no activity in the population.

As the PCF plot tends to be quite volatile, an option is provided to compute a “moving window” average of the plot. For this, a window size (in ms) is specified. The total number of cells active within this window is averaged over the window and then plotted. The PCF

procedure is outlined in Algorithm 2. This is one of the most frequently used functions, and is roughly equivalent to the electroneurograms recorded from spinal ventral roots. Examples are shown in Figs. 7 (2D graphs), 3 and 9.

Rate Meter Plot (RMP)

The rate meter plot describes the firing frequency of a cell in a population as a function of the simulation time. A population identifier and a cell identifier (within this population) are specified by the user. These are used to search the simulation data structures for the cell's firing intervals. The cell's firing intervals are calculated in a manner similar to the PIH function, except that all of the intervals are from a single cell. The reciprocal of the interval (suitably scaled to spikes per sec.) is reported as the frequency of cell firing. These values are plotted at the time of second spike. In neurophysiology, this plot is also called instantaneous frequency plot. In addition, the procedure reports the minimum and maximum intervals, minimum and maximum frequencies, mean interval and frequency, and the variance and standard deviation of the intervals. See Fig. 8 for examples.

Cell Potential Plot (CPP)

This is a plot of the membrane potential of a specific cell with respect to time. As we retain the membrane potential of every cell for the period of the simulation, generating these data simply requires a table lookup at each time step to retrieve the cell potential. See Fig. 8 for examples.

Cell Correlation Function (CCF)

This function determines the correlation between the activity of two cells, within the same or different populations. When the two cells are the same, we obtain the auto-correlation function. Input to this function involves specifying the two cells of interest and the populations to which they belong. To compute the correlation between any two cells, the firing times of both cells are recorded for the interval of the simulation that is of interest, $c_1(t)$ and $c_2(t)$. Cross-correlation is then computed as follows:

$$c_1(t) \circ c_2(t) = \frac{1}{M} \sum_{m=0}^{M-1} c_1(m)c_2(t+m) \quad (2)$$

for $t = 0, 1, 2, \dots, M-1$. This is equivalent to sliding $c_2(t)$ under $c_1(t)$ and integrating the product of the two functions, for each value of the displacement t .

To determine $c_1(t)$ and $c_2(t)$ from the simulation data structures, we search the firing cell list at each time step and create two lists, containing the time steps at which the two cells are active. For each translate of $c_2(t)$, the correlation is determined by integrating the point-wise product of the two spike functions. The time steps at which both cells are active contribute to the sum. As the two lists are sorted in order of increasing time, this procedure is equivalent to merging two sorted lists (which can be performed in time proportional to the sum of the list sizes). We maintain both $c_1(t)$ and $c_2(t)$ as dynamically allocated lists of time steps (keeping only those times the cells are active); as most cells are generally active for only a small fraction of the entire simulation period, this leads to an efficient procedure, in terms of both memory and computation time.

Implementation

NVIZ has been implemented in C++ on Unix workstations (SGI/Linux).^{*} All visualizations were generated using the Visualization Toolkit²⁷ and embedded within the FLTK graphical interface.²⁸ All features of NVIZ are available from the graphical interface. Once the application is started, the first step is to read in a neural circuit description (currently from a disk file) followed by specification of the simulation interval, to start the simulation. Once the simulation is completed, the generated data stored in simulation data structures are used for visualization and analysis.

Figure 6 shows the software architecture of NVIZ. The Simulator class consists of neuron populations (cell and fiber populations), synaptic types, and the ActiveNeuron data structure, which holds the data generated by the simulation. The neuron populations are defined by a generic Population class from which cell and fiber populations are derived. The ActiveNeuron data structure is a list, and contains for each time step all system variables computed in the two compartment model, including the membrane potentials and conductances for the soma and dendrite, calcium concentration and conductance, and the membrane threshold.

The Visualization modules are housed in the ViewInteractor class. Four views are currently supported: Anatomic, Limb, Neuron and Statistic. Each of these classes is derived from the View class. The anatomical view displays the spinal cord model, with up to four neuron populations (can be selectively turned on/off). The Limb view is specific to the hip movement application.

The Graph view is used by the analysis functions and may be either static 2D plots or dynamic (time-varying) 2D plots, used by the anatomical view. The ViewInteractor class also contains a number of renderers, canvases and interactor objects that provide the infrastructure for the 2D and 3D visualization, which is implemented by calls to the Visualization Toolkit.²⁷

The analysis functions are implemented within the Stat class and include the Rate Meter Plot (RMP), Population Cell Firings (PCF), Population Interval Histogram (PIH), Cell Potential Plot (CPP) and Cell Correlation Function (CCF). Each of these functions is available from the interface and has dialog widgets to interactively analyze at the population or cell level. Multiple cell or population characteristics may be displayed in the accompanying plots, and support is provided for batch level analysis.

All user interactions with the the three major components of NVIZ, viz., Simulator, ViewInteractor and Stat, are implemented using the GUI class, via application callback functions, using the FLTK Toolkit.²⁸ NVIZ also provides a number of additional features for interaction and analysis (not shown in Fig. 6). The height field view permits a user to specify the populations to be viewed and supports simple rectangular layouts. Measurement scales, which can be toggled on or off, are provided close to each population, to rapidly convey the magnitude of the presented variables. The anatomical view permits the selection of populations to be included in the view. Finally, simulation results can be reviewed (for both anatomical and height field views) via a VCR style interface (see bottom of panel in Fig. 3, and lower right of the panels of Fig. 7), which permits animating the simulation, or analyzing step by step. A time scale bar

^{*}NVIZ uses open source tools that can easily be ported to a Microsoft Windows platform as well.

provides a direct means to move to a specific point in the simulation.

Results

Example Application: Hip Joint Movement Cycle

We illustrate the use of NVIZ with a single hip joint walking application.^{10,22,29} The experiment presented is based on the circuit diagram shown in Fig. 2, a “half-center”, or “reciprocal inhibitory” model of pattern and rhythm formation in the spinal cord of cat or human. The model includes multiple hypotheses:

- Pacemaker neurons comprising populations 13 (arbitrarily called the “flexor rhythm generator”) and 14 (“extensor rhythm generator”) control the rate of locomotor rhythm.
- Alternation of pacemaker activity is controlled through a pair of inhibitory non-pacemaking interneuron populations (11 and 12) arranged such that when cells in 14 fire, they activate cells in 12, which in turn prevent cells in 13 from firing. The burst of activity in 14 is self-terminating, so when cells in 13 stop getting inhibition to oppose the continuing excitation from the fibers (population 22), they fire and the half cycle is repeated. Thus, alternation is the natural property of this part of the network.
- The spinal central pattern generator network for control of hip muscles of one leg consists of four portions: (1) pacemaker kernel (interneuron populations 11-13) is responsible for producing the inherent alternating rhythm of locomotion, through a coupled pacemaker reciprocal inhibitory network, (2) distribution interneurons (populations 9, 10) control the intensity of motor output, by exciting motoneurons (populations 1 and 2). (3) postpacemaker interneurons (populations 5 and 6), produce reciprocal inhibition to the “inactive” population and (4) output elements are motoneuron populations (1, hip flexor sartorius, labeled SART, and 2, hip extensor semimembranosus and anterior biceps, labeled SMAB).

No assumptions incorporated in the model are contradicted by existing data in the literature, so we may view this as a hypothetical spinal central pattern generator for the alternating hip flexion and extension occurring during walking. Activity in the circuit is caused by a fiber population (grey rectangle labeled 22) consisting of 100 fibers acting on two populations (13 and 14) of pacemaker type neurons. The pacemaker cells respond by producing a self-terminating burst of spikes (see Fig. 8, Panel B1). At the beginning of the experiment, a “start-up” fiber population (grey rectangle labeled 21) activates population 14, so that the experiment always starts in the same “phase”, in this case, in “extension”.

Joint Motion: Net Neural Drive

The neural output from the simulator is used to illustrate the motion of a hip joint using a simple ball and stick (limb) display. Motoneuronal pool output is represented as an Activity Index (AI).¹⁹ Activity index is defined as the number of motoneurons firing one or more spikes over a time interval, (say 100 ms), multiplied by their mean frequency of firing, over that interval. Thus, the flexor and extensor activity indices for a given interval are represented as

$$AI_f(t) = N_f(t) \cdot V_f(t) \quad (3)$$

$$AI_e(t) = N_e(t) \cdot V_e(t) \quad (4)$$

where $N_f(t)$, $N_e(t)$ are the number of cells active over Δt for the flexor and extensors, and $V_f(t)$, $V_e(t)$ are the spiking frequencies. The activity index represents “total synaptic effect” of a pool of motoneurons on its muscle. In practice, two procedures were used to calculate the activity index. First, mean population frequency of firing is relatively constant in networks using the Hill-Kernell-MacGregor algorithm, and so V_f and V_e can be safely assumed to be constant for all times during a “step cycle” when cells are active. Second, number of cells active is smoothed across multiple time steps, usually 100 ms, using the PCF analysis function of NVIZ, giving us a time series of active cells. The time series data are multiplied by the mean frequency to give the activity indices as time series, $AI_f(t)$ and $AI_e(t)$. We assume that the activity index is proportional to motor pool force output. Force is proportional to angular acceleration, so, we must integrate neural output twice to get angular displacement. The two smoothing operations (on frequency and cell number) can be viewed as integrations.

From the Activity index, we compute *NND*, the *Net Neural Drive*, as

$$NND(t) = AI_e(t) - AI_f(t) \quad (5)$$

Positive values of *NND* represent extension, and negative values, flexion. Net Neural Drive is thus a new way to specify joint position from neuronal signal. We may think of joint movement, in the simplest case of a single opposing pair of muscles, as depending on two neural control signals (the activity indices, $AI_f(t)$, $AI_e(t)$), one for flexion and one for extension. Downstream from the neural control signals are the force-producing muscle cells, and biomechanical linkages providing the lever arms through which force is applied to bone. Net neural drive collapses all of these steps into a single metric, representing the angular displacement.

To compute the joint angle from *NND*, we scale it to a fractional variable, neural fraction, $nf(t)$, by a linear mapping between two chosen joint angle values, JA_{min} and JA_{max} , the minimum and maximum joint angles, using the following parametric form:

$$JA(nf(t)) = JA_{min} + nf(t) * (JA_{max} - JA_{min}) \quad (6)$$

with

$$nf(t) \in (0.0, 1.0)$$

The limb is animated between $JA_{min}(100^\circ)$ and $JA_{max}(130^\circ)$ in a 2D plane, as can be seen from the panels of Fig. 7.

Simulation Experiment

Motoneuronal outputs were recorded in a representative simulation on the walking circuit of Fig. 2. The simulation was run for 3000 ms. Figures 7 and 10 show motoneuronal outputs (respectively anatomical and height field views) at four different time steps, corresponding to a cycle of extension and flexion.

At time step 1539 (panel A, Fig. 7), we see the peak of the extension phase. The graph view (lower left part of the panel) shows about 6 cells active per ms and the limb visualization (lower middle part of the panel) is in an extended (standing) position. The top part of the panel shows the anatomical location of active motoneurons at this time in the simulation. Six cells in the extensor population are opaque. At time step 1789 (panel B), we

see the transition from extension to flexion. The graph view indicates that the output of the flexor population (red curve) is increasing; the limb visualization shows the thigh elevated. Cells active at this time step are shown in the upper part of the panel. One cell in the flexor population is opaque. At time step 2005 (panel C), maximum flexion (lifting) of the thigh has been reached. The graph view indicates the point of maximum activity of the flexor population, and the limb view indicates maximum flexion. Finally, time step 2267 (panel D) shows the transition from flexion to extension. The graph view (lower left part of panel D) shows the number of extensor motoneurons (green curve) increasing.

The height field views of the hip movement cycle in Fig. 10 illustrate the flexor and extensor output (numbered 1 and 2 in our circuit) at the same four time steps. Panel A shows the peak extension phase, with population 2 highly active, panel B shows the transition to flexion (decreasing extension and increasing flexor activity), panel C illustrates maximum flexion and panel D shows transition to extension. The height field views permits us to simply look at the individual populations and their state variables in isolation, while the anatomical views provide spatial cues to the neurons in the spinal cord.

Population and Cell Level Analysis

Figure 9 illustrates NVIZ population-level analysis measures for this experiment. Figure 9A shows PCF time series graphs (smoothed with a 25 ms moving box filter) of flexor and extensor pacemaker (13, 14, panel A3), distribution interneurons (9, 10, panel A2), post-pacemaker interneurons (5, 6, panel A1) and motoneuron (1, 2, panel A1) populations. The graphs are best read from bottom to top, as a reflection of the arrangement of the circuit diagram (Fig. 2), where the “generating” populations are at the bottom. Note the differences in shape of curves in the three panels, with pacemakers (13, 14, panel A3) sharply peaked, distribution interneurons (9, 10, panel A2) with a more rounded profile, and motoneurons (1, 2, panel A1) even more rounded. The difference in shape reflects both cell properties and operation of the circuit as activity flows from pacemaker populations through distribution interneurons to motoneurons. PCF curves for populations 5 and 6 are included in panel A1, plotted downward as a convenience. Quantification of an individual cycle (marked with an asterisk) using the PIH function is presented in Fig. 9B. Several things can be noted from the histograms and accompanying data. First, a population producing the largest PCF curves (9, panel A2) generates the largest number of spikes ($n = 2867$, panel B2), and has the shortest mean interspike interval (15.4 ms) of the three populations shown. Second, each of the three populations (pacemakers 13, generic interneurons 9, and motoneurons 1) has a characteristic histogram profile, which is correlated with its biophysical properties, and the level of synaptic drive it is receiving. Note that the maximum number of intervals in panel B2 is twice that in panels B1 and B3.

Figure 8 shows cell potential profile (CPP) graphs and closely-related rate meter plot (RMP) graphs for a small number of cells, to illustrate capabilities of NVIZ at this level of analysis. Panel A1 shows CPP graphs for motoneuron cell 1_5 (thin curve), a “heavily-driven” cells in population 1, and cell 2_89 (thick curve), a “lightly-driven” cell. Panel A2 shows the corresponding rate meter plot; note that cell 1_5 (filled squares) fires at a fairly high frequency (around 60 Hz), but that the frequency

gradually increases and decreases over the cycle. A companion cell 1_89 (filled circles), with a lower more regular firing frequency, is also represented in the RMP, but not in the CPP graph. Visual inspection of Panel A1 indicates that cell 2_89 is firing at a much lower frequency than cell 1_5; the RMP graph quantifies the difference. It is interesting to note that the PIH data for population 1 (Fig. 9, Panel B1) indicate that the standard deviation of interspike intervals in population 1 approaches the mean. Examples of pacemaker population 13 CPP graphs are shown in Fig. 8, panel B1, with the corresponding rate meter plots in panel B2. Note the sharply-peaked profile of firing frequency exhibited by this cell type. Compare this cell-level firing profile with the shape of the PCF graph in Fig. 9, panel A3. From this comparison, we can draw the conclusion that the “peaked” form of the PCF graph arises mainly from the “peaked” form of the firing frequency of individual cells in the population. Resting membrane potential of MacGregor type neurons is arbitrarily set to 0 mV in the absence of input, in contrast to “real” neurons, where membrane potential is typically about -70 mV. In addition, spike potential in the MacGregor type neuron is not directly represented in the CPP plots. These examples illustrate the power of having multiple analysis functions available within the simulation environment.

Discussion

The overall design goal of NVIZ was to use interactive visualization and animation to facilitate efficient simulation and analysis of large numbers of neuron populations *within a single integrated environment*. Simulation and analysis are indeed performed within a tightly coupled environment, assisted by 2D and 3D visualizations. In combination with the numerical analysis subsystem, this integrated environment promotes efficient experiment design and hypotheses testing, and incorporates the necessary means to handle models of arbitrarily large size.

NVIZ supports dynamic height field and anatomical spinal cord representations to view neuronal activity in 3D. Though somewhat abstract, animations of height field views, by virtue of their simple (matrix) geometry, provide the ability to visualize critical state variables of every neuron in the simulation, and, in addition, promote easy comparisons among populations. Approximate measurement of cell variables is also facilitated by scales provided adjacent to the height fields. As the individual populations are separated in the visualization, occlusion is not an issue. With larger numbers of populations, occlusion is possible, but usually can be resolved by interactively manipulating the layout with the pick function and interactive viewing tools. The layout features permit scalability to larger models and simulations, by allowing users to focus on selected populations, neurons and system variables of interest. This feature becomes increasingly important for scaling the system to larger models.

Viewing dynamic neuron activity in the context of an anatomical representation contributes to the physical realism of the visualizations, and promotes communication among researchers. The only other work we know of that projects a dynamic data presentation on an anatomical representation of the spinal cord is the work of Yakovenko et al.¹¹ In that work, locations of 27 populations of hind limb motoneurons were digitized from the same data set used here (36 sections of Fig. 28 in Ref. 12). Activity of cells during locomotion was animated using an extensive library of electromyographic (EMG) recordings (muscle activation

patterns) they generated from the literature of normal cat locomotion. The EMG record of a single step, i.e., flexion-extension cycle, was divided into 100 piecewise-constant segments, each corresponding to about 7 ms of real time. Their visualization generated two main findings, (1) greater extensor activity than flexor, a reasonable result, given that there are a greater total number of extensor motoneurons than flexor motoneurons, and (2) the visualization of a front-to-back wave of neural activity in flexion and extension of a single step was not necessarily expected, since some motoneuron pools are distributed throughout the entire visualized region. Note that cells in their visualization were animated from "output data", i.e., EMG, of cats walking on a treadmill, rather than from a model of the central pattern generator, as in *NVIZ*. In other words, *NVIZ* generates movement from hypotheses constructed from first principles. The hypotheses concern biophysical properties of cells, synapses and population connections, as stated in the Results section.

In summary, the different visualizations in *NVIZ* provide the means to comprehend and analyze complex spatiotemporal data, arising from realistic large-scale simulations of cat spinal cord. The visualization tools were created in two windows, each containing 2D and 3D components. The first window, shown in Fig. 3, contains the height field view and most of the numerical analysis modules. The second window, shown in Fig. 7, contains the 3D anatomical view, dynamic time-varying 2D plot of cell firings, and representation of limb movement. Both windows contain VCR controls and time stamps, and access the same underlying data structures.

Conclusions

We have described the goals, design and implementation of *NVIZ*, a new system that integrates simulation, visualization and analysis within a highly interactive environment for understanding complex neuronal dynamics of the spinal cord. *NVIZ* encompasses five biological levels of organization: ion channels, neurons, populations, Net Neural Drive (in lieu of musculoskeletal activation), and observed behavior. We have demonstrated many of the features of *NVIZ* (circuit design, simulation, visualization and analysis) using a single hip joint movement experiment. Notably, the inclusion of an anatomical framework for the data was found to be a powerful means to communicate experimental results. Limb movement was generated by a novel algorithm, *Net Neural Drive*, which circumvents the computational complexity of biomechanical representation.

The use of modern interactive visualization and animation tools and access to the features of *NVIZ* from a graphical user interface makes it an attractive environment for analyzing large scale neuron models of the spinal cord. In the future, we hope to address the following avenues for development and improvement of *NVIZ*:

- Interactive graphical specification and editing of neural circuit descriptions.
- Inclusion of additional interneuronal populations in the anatomical visualization.
- Extension of *NVIZ* to larger models: multiple joints containing larger numbers of populations.
- Validation of the Net Neural Drive algorithm against animal and human joint angle data.
- Expansion the two compartment neuron model to an arbitrary number of dendrites and compartments, with appropriate visual representations.
- Optimization of the software for larger models. ▲

Acknowledgment. A number of students have contributed to different parts of *NVIZ* in earlier versions, including William Hasty,³⁰ James Harrison,³¹ Chris Driggers,³² and Haiping Pan.³³ Anatomical model was constructed with the help of Fig. 28 in Ref. 12. DB thanks Larry Jordan, Dave McCrea and Katinka Stecina for helpful discussions. This work was supported in part by a grant from the University of North Carolina at Charlotte.

References

1. G. M. Edelman, *Neural Darwinism: The Theory of Neuronal Group Selection*. Basic Books, New York, NY, 1987.
2. E. E. Fetz, "Are movement parameters recognizably coded in the activity of single neurons?", *Behavioral Brain Science*, **15**: 679–690 (1992).
3. E. de Schutter, "A consumer guide to neuronal modeling software". *Trends in Neurosciences*, **15**(11):462–464 (1992).
4. Y. Manor, J. Gonczarowski, and I. Segev, "Propagation of action potentials along complex axonal trees, model and implementation". *Biophysics Journal*, **60**, 1411–1423 (1991).
5. M. L. Hines and N. T. Carnevale, "The neuron simulation environment". *Neural Computation*, **9**(6), 1179–1209 (1997).
6. F. E. Theunissen F. H. Eeckman and J. P. Miller, "NeMoSys: An approach to realistic neural simulation". In *Neural Systems: Analysis and Modeling*, F. H. Eeckman, Ed., (Kluwer Academic, Boston, MA, 1993) pp. 121–136.
7. D. Beeman and J. M. Bower, *The Book of Genesis*. (Springer Verlag, 1993).
8. E. De Schutter, "NODUS, A User Friendly Neuron Simulator for Macintosh Computers". In *Neural Systems: Analysis and Modeling*, F. H. Eeckman, Ed., (Kluwer Academic, Boston, MA, 1993).
9. K. R. Subramanian, D. P. Bashor, W. V. Hasty, and S. M. Merkel, "Multi-level visualization of spinal reflex circuit simulations". *IEEE Comp. Graphics Appl.* **17**(3) (1997).
10. K. R. Subramanian, D. P. Bashor, M. T. Miller, and J. A. Foster, "Visualizing the spinal neuronal dynamics of locomotion". *Proc. SPIE*, **5295**, 286 – 295 (2004).
11. S. Yakovenko, V. Mushahwar, V. Vanderhorst, G. Holstege, and A. Prochazka, "Spatiotemporal activation of lumbosacral motoneurons in the locomotor step cycle". *J. Neurophysiology*, **87**(3), 1542–1553 (2001).
12. V. G. Vanderhorst and G. Holstege, "Organization of lumbosacral motoneuronal cell groups innervating hindlimb, pelvic floor, and axial muscles in the cat". *J. Comparative Neurology*, **382**, 46–76 (1997).
13. G. A. Ascoli, Ed., *Computational Neuroanatomy: Principles and Methods*. (Humana Press, Totowa, NJ, 2002).
14. J. G. Bjaalie, Ed., "Three-dimensional computer reconstructions in neuroanatomy". In *Quantitative Methods in Neuroanatomy*, M. G. Stewart, Ed., (John Wiley and Sons, New York, NY, 1992) pp. 249–293.
15. J. G. Bjaalie, "Localization in the brain: new solutions emerging". *Nature*, **3**, 322–325, April 2002.
16. T. B. Leergaard, K. D. Alloway, T. A. T. Pham, I. Bolstad, Z. S. Hoffer, C. Petterson, and J. G. Bjaalie, "Three-dimensional topography of corticospinal projections from rat sensorimotor cortex: comparisons with corticostriatal projections reveal diverse integrative organization". *J. Comparative Neurology*, **478**, 306–322 (2004).
17. T. B. Leergaard and J. G. Bjaalie, "Architecture of sensory map transformations", in *Computational Neuroanatomy: Principles and Methods*, G. A. Ascoli, Ed., (Humana Press, Totowa, NJ, 2002) pp. 199–217.
18. A. Brevik, T. B. Leergaard, M. Svanevik, and J. G. Bjaalie, "Three-dimensional computerized atlas of the rat brain stem precerebellar system: approaches for mapping, visualization, and comparison of spatial distribution data". *Anatomy and Embryology*, **204**(4), 319–332 (2001).
19. D. P. Bashor, A large-scale model of some spinal reflex circuits. *Biological Cybernetics*, **78**, 147–157 (1998).
20. R. J. MacGregor, *Neural and Brain Modeling* (Academic Press, NY, 1987).
21. P. A. Getting, "Reconstruction of small neural networks", in *Methods in Neuronal Modeling*, C. Koch and I. Segev, Eds., (MIT Press, Cambridge, 1989).
22. D. P. Bashor, "One leg walking: A model of rhythm and pattern generation in cat lumbar spinal cord", *Society for Neuroscience Abstracts*, **26**, 747.5 (2000).
23. R. J. MacGregor and R. M. Oliver, A model for repetitive firing in neurons. *Kybernetik*, **16**, 53–64 (1974).
24. D. P. Bashor, "Hill-Kernell-MacGregor two-compartment neuron models for use in spinal network" *Biological Cybernetics*, 2004. Submitted.
25. A. Colbourne. Ac3d. <http://www.fltk.org>.
26. D. H. U. Kochanek and R. H. Bartels, "Interpolating splines with local tension, continuity and bias control". *Comp. Graphics*, **18**(3), 33–41 (1984).
27. W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. 3rd ed., (Prentice Hall Inc., Clifton Park, NJ, 2002).

28. B. Spitzak, *The fast light toolkit*. <http://www.fltk.org>.
29. D. Bashor, Y. Dai, D. Kriellaars, and L. Jordan, "Pattern generators for muscles crossing more than one joint", in *Annals of the New York Academy of Sciences*, **680**, 444–447 (1998).
30. W. V. Hasty, *An integrated graphical environment for the simulation and analysis of spinal motor circuits*. Senior Thesis (The University of North Carolina at Charlotte, Charlotte, NC, 1996).
31. J. A. Harrison, *SPSIM: An integrated visual simulation environment for analyzing spinal reflex circuits*. Senior Thesis, (The University of North Carolina at Charlotte, Charlotte, NC, 1997).
32. C. Driggers, *Visualization tools*. Senior Thesis, (The University of North Carolina at Charlotte, Charlotte, NC, 2001).
33. H. Pan, *Improvements to NVIZ*. Senior Thesis, (The University of North Carolina at Charlotte, Charlotte, NC, 2003).