# CHAPTER 1: INTRODUCTION TO CNS

CNS, the Cortical Network Simulator, is a software neural systems simulator written by Dr. G.N. Reeke, Jr. with support from The Neurosciences Research Foundation. CNS permits the user to construct models of biologically realistic, behaving neural systems. These systems may comprise multiple interconnected brain areas or nuclei, each of which may contain cells of many different types. Systems constructed with CNS can receive input from real or simulated senses and can provide output to real or simulated motor systems. CNS can be used to simulate purely perceptual as well as behaving systems. It can be used as a neurally based controller for real or virtual "brain-based devices" (BBDs) if compiled to support this feature. CNS is particularly adapted to support models based on G.M. Edelman's theory of neuronal group selection (TNGS). It was used to generate the Darwin III and Darwin IV models at The Neurosciences Institute. Published descriptions of these and other models constructed with CNS are listed in an Appendix.

This introduction will familiarize the prospective user with the main features of CNS with simple working examples. Not all features of CNS are covered here; reference information describing the program in detail is contained in the following chapters of this manual.

## What CNS Can Do

The level of physiological detail in CNS models is intermediate between that of the multicompartment ionic channel model and the "neural network" model. CNS can model a variety of rate-coded and integrate-and-fire cells, including the "simple model" cells proposed by E.M. Izhikevich, "Simple model of spiking neurons," IEEE Trans. Neural Networks 14:1569-1572 (2003) and the "adaptive exponential integrate and fire (aEIF)" model of R. Brette and W. Gerstner "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," J. Neurophysiol. 94:3637-3642 (2005). In brief, cells in CNS can integrate synaptic inputs, some of which may be voltage-dependent, and respond to inputs above a specified threshold in a graded fashion or by emitting single spikes or spike trains. Production of a spike may induce a refractory period, possibly followed by synapse-specific paired pulse facilitation (PPF). Cell responses may be subject to graded depression (habituation). Probe currents or probabilistic noise may be injected into cells. Transmission delays between cells can be simulated. Time constants may be specified to control decay of cell activity or postsynaptic potentials.

The geometry and initial strengths of connections between cells can be specified according to a variety of built-in options, or they can be read from files. Several mechanisms based on the well-known Hebb rule are provided for changing the strengths of synaptic connections. In CNS, this process is called "amplification". Amplification may be made to depend on the magnitude of a "value" signal from some other neural area or an external source. The concept of "value" is an intrinsic part of the TNGS. More information on this topic may be found in K.J. Friston, G. Tononi, G.N. Reeke, Jr., O. Sporns, and G.M. Edelman, "Value-dependent selection in the brain: Simulation in a synthetic neural model," Neurosci. 59:229-243 (1994). Other amplification schemes can be easily added if needed.

Contents

CNS is "batch-oriented": It reads and executes commands from a control file (for old times' sake, these are called "control cards"). The control file contains a description of the networks to be constructed and the stimuli to be applied during the simulation. Once the simulation begins, interactive control is possible by typing commands to the program. Keeping the setup information in an editable file makes it easy to carry out the common development cycle of specify, run, change, and run again. Output from CNS is available in the form of printable text, graphics, and binary files. Text output normally includes a transcript of the control file, a tabular summary of the network parameters, detailed information about individual cells as requested, and statistics. Graphical output is flexible, but typically shows the activity of each cell in the system after each simulation cycle. Graphics can be recorded in a METAFILE for later viewing or for transmission to another computer. The state of the simulated system can be recorded at any time in a SAVENET file, which may be used to restart the run later, possibly with altered parameters. Selected cellular and synaptic parameters can be recorded in a SIMDATA file for offline analysis. A utility program, getgd3, is available for reading data from a SIMDATA file into MATLAB (TM) for easy development of one-off analysis routines.

CNS is written in C and can run on a variety of serial and parallel computers under several different operating systems. Care has been taken to assure that the results are numerically identical on all supported processors. Processors currently supported include SPARC, Intel x86, IBM 370, and Power PC G4. Operating systems currently include UNIX, LINUX, MacOSX, and IBM's VM/CMS, with Microsoft Windows under consideration.

## What CNS Cannot Do

Cells in CNS are positioned on two-dimensional sheets for purposes of plotting their activity, but the program has no concept of the third dimension. Cells can be arranged in rectangular arrays, not hexagonal, and there is no good way to model positional variations in cell density, as occur, for example, in the retina. CNS cannot simulate cellular or neuritic compartments or the dynamics of individual ion channels or ions. It does not currently simulate volume signals, such as diffusible molecules (e.g. nitric oxide). It cannot simulate cell growth. CNS is not designed to simulate common non-biological "neural network" models, such as Hopfield networks or multi-layer perceptrons with back-propagation learning rules. Support for the Darwin IV “NOMAD” platform was removed in V8F.

# CNS Concepts

## Series, Trials, and Cycles

Cycling in CNS occurs in three nested loops, referred to as "trial series" (or simply "series"), "trials", and "cycles". The trial series is the unit of operation over which statistics are collected and printed. It corresponds loosely to a "session" in a laboratory psychology experiment. A trial series consists of one or more trials. The trial is the unit of operation in which a stimulus is presented and remains stationary for one or more cycles of the model. The cycle is the unit of operation, corresponding to a fixed interval of time, in which the responses of all the neural networks in the system are calculated and modification of connection strengths is performed. There may be more than one cycle per trial, corresponding to an experiment in which an animal is paralyzed or otherwise constrained to

view a stimulus for some time, or there may be one cycle per trial, corresponding to unconstrained behavior in which there is a continual interplay between the environment and the responses of the subject. Cycling is controlled by the Group III CYCLE card.

Trial numbers are counted both from the start of each series and from the start of the whole run. The series-relative numbers are used with timers that control when printing, plotting, etc. occurs. The run-relative numbers are used to label trials unambiguously in SIMDATA and REPLAY files. In printed output, graphs, and SAVENET file headers, the series-relative number is given, followed by the run-relative number in parentheses if different (i.e. if not in series 1).

## Regions and Groups

In CNS, neural areas are referred to as "regions". In earlier versions of CNS, regions were known by the TNGS term, "repertoires". Regions are divided into "groups", which are collections of cells that correspond to cortical columns or minicolumns, depending on the model. Groups are constructed by CNS as rectangular arrays of cells and regions are constructed as rectangular arrays of groups.

Regions and groups may contain cells of many different types. No particular relationship is enforced among these cells, except that they share common output and statistical options. All the groups in a region always contain the same number of cells of any given type, but the numbers of cells of different types within a group may differ. The cells of a region are always plotted together in a rectangular panel which may be combined with other regions in a "superposition plot" that shows an entire model system.

CNS groups may play a role in generating intercellular connectivity. For example, connections may be specified to connect only cells that are in the same group, or only cells that are in different groups. Topographic maps between regions connect groups in corresponding geometric positions within their regions, but all the cells within any one group are always deemed to have geometrically indistinguishable positions.

Regions are specified by Group II REGION control cards (REPERTOIRE is synonymous). The user must assign two names to each region, a "long" or "external" name, which is used to identify the region in statistical and graphical output, and a "short" or "internal" name, which is used to refer to it on other control cards that require region names, for example, to specify the source of a class of connections. Long names may contain up to 80 characters. They do not need to be unique and may even be blank. Short names may contain up to 4 characters and must be unique within the model. The size of every region is specified after the names on the REGION card by the numbers of groups along the x and y coordinate directions (*%ngx*, *%ngy*). Plotting and statistical options are specified with the KP or KRP keyword. Options B (make "bubble" plots of cell activity) and A (plot in all trial series) are almost always used. In addition, the width (*%w*), height (*%h*), and location (*%x*, *%y*) of the region's panel in the superposition plot are usually also given. (The present mechanism for locating panels automatically does not work very well.) Optionally, grid lines may be added to delineate clusters of groups (KP=G, *%grids*). Here is a sample REGION card:

```
REGION 'Lateral Geniculate' LGN 32 32, KP=ABG, X=1.0, Y=2.0, W=4.0, H=4.0,
    GRIDS=4
```

Contents

This card sets up a region of 32 x 32 groups. Its external name is "Lateral Geniculate"; its internal name is "LGN". The region will be plotted in every trial in a panel with height and width of 4 inches with the lower left-hand corner located at x=1.0, y=2.0 inches on the screen and with grid lines between every fourth group.

## Cells and Cell Types

The term "cell type" refers to a collection of cells within a region, all of which share common parameters such as response function, decay times, thresholds, and noise levels, and all of which receive connections of the same types and numbers from other cells. (Of course, the individual connections of each cell in a cell type usually differ in source and strength.) Cell types are also known as "layers" because they often correspond to cell layers in a cortical area. As seen by other cells, all cells of a given type are effectively updated at the same instant of time during each simulation cycle.

Cell types are specified by CELLTYPE control cards. On this card, the user must provide a short (4-character) name for the cell type (*%lname*) (long names are not supported), the number of cells of this type to be generated in each group of the current region (*%nel*), and other parameters needed to specify the response function (see next paragraph). Additional parameters may be placed on DECAY, DEPRESSION, AEIF, AUTOSCALE, IZHIKEVICH, NOISE, PHASE, and REFRACTORY PERIOD cards. Plotting and statistical options are specified with the KCTP keyword.

Cells in CNS may have somewhat more complex dynamics than in typical "neural network" applications, including “integrate and fire” spiking, afterhyperpolarization, depression, refractory periods, and paired pulse facilitation. Every cell in a CNS model has a scalar state variable which represents the membrane potential or activity of that cell relative to its resting potential, which is always taken as zero internally. The activity of cell i is denoted by s(i), -256mV <= s < 256mV, 16-bit precision, or s(i,t) to indicate the activity at a particular time, t. Alternatively, a compatibility option is available (COMPAT=C) in which s values are unitless and constrained to the range 0 <= s < 1. It is also possible to calculate s(i) as a profile of activity across 32 (this number is arbitrarily fixed in the CNS program code) subintervals of each simulation cycle. This results in each s(i) being assigned a magnitude and phase. Phase calculations are described in detail later. The value of s(i) is transmitted as input (possibly after a delay, see below) to all cells that are connected postsynaptically to cell i. s(i) is calculated according to a "response function", which has terms corresponding to synaptic inputs, external probe inputs, noise, decay of previous activity, depression, and refractory periods. In rate-coded models, s(i) approximates a sigmoidal response by a piecewise linear (Eqn. 1a) or hyperbolic tangent (Eqn. 1b) function of the total inputs to the cell; spiking cells are described below (see Simple spiking cells, aEIF (Brette-Gerstner) neurons and Izhikevich neurons); other functional forms may be added in future versions of CNS as needed:

(Eqn. 1a) $$s(i) = \{[(A + G + M - Q*Q + V)\mathrm{phi}(Is) + P] + N + W\}\mathrm{phi}(D)$$

(Eqn. 1b) $$s(i) = \{\%thamp*\tanh([(A + G + M - Q*Q + V)\mathrm{phi}(Is) + P] + N) + W\}\mathrm{phi}(D)$$

where:

A = total combined excitatory and hyperpolarizing inhibitory input ("afference") from specific connections (see below) that are not voltage-dependent and from self connection (if any).

G = total combined excitatory and hyperpolarizing inhibitory input from geometric connections (see below).

Contents

M = total combined excitatory and hyperpolarizing inhibitory input from modulatory connections (see below).

Q = total input from specific, geometric, and modulatory connection types that have been designated as contributing "squared" inhibition. The total, Q, is multiplied by (Q/128mV) before being subtracted from the combined excitatory and hyperpolarizing inputs.

V = adjusted excitatory input from voltage-dependent connections (those with `VDOPT=P` or `C` specified). Let *%vdt* be a threshold parameter and *%vdha* be a "half-effectiveness" parameter that can be specified individually for each connection type. Then let Z = W (persistence term) if `VDOPT=P` is specified, Z = (A+G+M) if `VDOPT=C` is specified, or Z = (A+G+M+W) `VDOPT=PC` is specified. Then let Z = Z - *%vdt* or Z = |Z - *%vdt*| if `VDOPT` code `A` is specified. Voltage-dependent input is calculated as for the A term, then multiplied by (1) 0 if Z <= 0, (2) 1.0 - 1.0/cosh(arcsech(0.5)*(Z/*%vdha*)) if `VDOPT` code `H` (for "hyperbolic") is specified, and (3) 0.5*Z/*%vdha* in all other cases.

Is = total combined inhibitory input from specific, geometric, and modulatory connection types designated as "shunting", divided by 128mV.

P = probe current. CNS currently allows for a given constant or ramped probe current to be injected into any of a given list of cells.

N = noise term, which may be drawn from a Gaussian distribution and applied either to all cells or to a randomly selected fraction of the cells of this type. This allows a form of "shot" noise to be generated.

W = decay term = omega*s(i,t-1), where omega is a decay parameter that may be assigned two distinct values. *%omega1* is applied to s(i) in the event of a level 2 (see below) trial reset, while *%omega2* decay is applied in all cycles. Use of a trial reset permits decay of network activity during an intertrial interval (stimulus absent) to be simulated without the need to perform the actual cycles.

D = depression = *%upsd**s(i,t-1) + *%omegad**D(t-1), where *%upsd* = growth coefficient for depression, *%omegad* = decay coefficient for depression. May be thought of as corresponding to the presynaptic habituation observed in *Aplysia* and described in Kandel and Schwartz, p. 1012.

[ ] in the above equation indicates a threshold function which may be designated as of "step" or "knee" type. The terms within the [ ] are ignored unless positive and greater than a positive threshold, *%pt*, or negative and less than a negative threshold, *%nt*. If step type, [x] = 0 if *%nt* < x < *%pt*, otherwise [x] = x. If knee type, [x] = x + *%nt* if x <= *%nt*, [x] = 0 if *%nt* < x < *%pt*, otherwise [x] = x - *%pt*.

phi(x) = sigmoidal function, approximated as phi(x) = 1 - 2*(x**2) + (x**4).

*%thamp* = amplitude multiplier for hyperbolic tangent response. Note that there is no separate scale factor for the tanh argument: the individual connection type scales, probe currents, and mean noise can be used to adjust the argument to any desired scale.

Eqns. 1a and 1b are modified by the effects of simple (nonadaptive) spiking and refractory periods as follows: If the response function is designated as spiking, then the terms in [ ] plus N in Eqn. 1a are replaced by a spike value, *%spike*, when [ ]+N >= (*%st + dst)*, where *%st* is a specified spiking threshold. The value of *%st* can be tailored individually to conditions at each cell by the added dst (for "delta st") term. dst is initially 0 and changes in each cycle according to dst(i,t) = *%upsdst**s(i,t-1) + *%omegadst**dst(i,t-1). In the cycle following a spike, dst is reset to a specified constant "post-spike dst" value, *%psdst*. Additionally, if a refractory period is specified, the cell becomes refractory for *%refrac* cycles each time the [] term exceeds *%st*, whether or not the response function is the spiking type. During the refractory period, all inputs, including noise, are ignored and cell activity decays towards the afterhyperpolarization potential (AHP) according to s(i) = s(i,t-1) - [s(i,t-1) - AHP]*ln(2)/*%refrac*. (If

Contents

the length of the refractory period is just one cycle, s(i) is set to AHP during this cycle. An "abrupt refractory period" option exists which causes s(i) to be set to AHP during all refractory cycles.)

Binary ("on-off") cells can be simulated by making *%st* = *%pt*, so that the cell fires as soon as there is enough input to affect the cell at all. Cells with hysteresis, as described in L. Wang and J. Ross, "Synchronous neural networks of nonlinear threshold elements with hysteresis," Proc. Natl. Acad. Sci. USA 87:988-992 (1990), can be simulated by making *%omega2* = 1.0 so that the cell retains its past state until switched on or off by new excitatory or inhibitory input.

In the following simple example, 8 "V1" cells are placed in each group of the most recently entered region, with *%pt* = 10mV and *%nt* = -10mV and with noise mean 2mV and standard deviation 1mV. The default “KNEE” response function is used:

```
CELLTYPE V1 8 pt=10mV,nt=-10mV
NOISE 2 1
```

Note that in the Int. J. Supercomputer Appl. and Proc. IEEE papers cited above, the N and W terms in Eqn. 1a are erroneously shown as being unaffected by depression. The formulation shown here is correct.

Furthermore, the response function has been changed from that used in versions of CNS prior to V8A in several ways: (1) cell activity and all related quantities are expressed in mV rather than the old dimensionless 0 to 1 scale except when COMPAT=C is specified, (2) activity is now stored internally to 16 bit rather than 8 bit precision (32 bits for adaptive integrate-and-fire cells), (3) squared and shunting inhibition are now applicable to all three connection classes, (4) new parameter VDOPT controls calculations with new options for voltage-dependent connections, (5) inputs from all three connection classes are used in phase calculations, (6) when a spike occurs, the normal inputs are replaced by the spike value, not added to it, (7) each cell can now have an individual spike threshold, *%st*, which varies as described above, (8) refractory periods are triggered by membrane potential, not by accumulation of depression, (9) decay during a refractory period is different as described above, (10) refractory cells can no longer be driven to spike, and (11) the so-called "LTP" used in Darwin III has been removed from CNS. If information on the obsolete features is needed for the interpretation of old CNS runs, please consult the V7C version of this manual.

## Connections and Connection Types

A connection is a simulated synapse between two neurons. A "connection type" is a collection of connections that join cells of one cell type to cells of the same or another type and that share common scales, thresholds, decay methods and rates, and generating rules. Connections may be excitatory, hyperpolarizing, squared inhibitory, or shunting inhibitory. Input from excitatory connections is added to the response of the target cell; input from hyperpolarizing connections is subtracted; input from squared inhibitory connections is squared before being subtracted; and input from shunting inhibitory connections multiplies the input from other connection types as shown in Eqns. 1 above.

Contents

CNS supports three broad classes of connections: specific, geometric, and modulatory. Specific connections are individually constructed and their strengths can vary individually ("learning"); geometric connections are constructed in annular rings suitable for simulating lateral inhibition and their strength does not vary; modulatory connections take as input the average activity of an entire cell layer or an external input. Postsynaptic potentials (A, G, or M terms in Eqns. 1) can be made to decay more slowly than source cell activity; this feature can also be used to simulate typical EPSP decay or even an additional layer of slow inhibitory cells without the need to include them explicitly in the model. A voltage offset may be subtracted from inputs to any type of connection to allow for different zeros on the voltage scale for different cell types or different external inputs. This is referred to as a "reversal potential" in the documentation, because it controls whether the input will have an excitatory (>0) or inhibitory (<0) effect.

Inputs to geometric connections may come from other cells in the simulation. Inputs to specific and modulatory connections may come from other cells in the simulation, or from the built-in environmental simulator, from value schemes, or from externally attached devices. All of these types of inputs are described later. Inputs from cells have the range used for all cells in the simulation (normally -256 to +256 mV, alternatively, 0 to 1 in `COMPAT=C` mode). Inputs from the simulated environment or from value schemes always have the range 0 to 1, but these are mapped onto a range 0 to 128 mV when the mV cell activity scale is in use. Control card options are provided to map inputs from external sources onto either range as needed. A user-written routine ("GetSj") may be called from inside CNS to manipulate input levels in any manner desired.

Specific Connections (represented by A or V in the response function) are individually enumerated in lists stored in computer memory. Each connection has its own specific source cell, designated by l(ij), where l is the number of the source cell, i is an index over target cells, and j is an index over connections; its own connection strength, c(ij), which may be changed independently of all other connections; and possibly other individual variables c0(ij), d(ij), m(ij), r(ij) to be discussed later. Specific connections are generated according to a rule specified by the user, or they can be read from an externally generated file. These connections have a great deal of flexibility, but are the most expensive in terms of both memory and execution time.

A single specific connection type may contain a mixture of excitatory connections and any one of the various types of inhibitory connections. Excitatory and inhibitory inputs are summed separately and the totals are processed separately as described in Eqns. 1. The same number of connections is generated for each target cell reached by a particular connection type, but connections which are generated out-of-bounds are skipped; therefore the total number of connections reaching each cell is not necessarily identical.

The afferent input, A, for cell i, from non-voltage-dependent specific connection types is calculated according to the following equation:

(Eqn. 2) $$A = \mathrm{Sum}(k)\{\%scale(k)*\mathrm{Sum}(j)\{(1+ppf)*c(ij)*[s(l(ij))-\%sjrev(k)-\%et(k)]\}\}$$

where:

k = index that runs over connection types.

*%scale*(k) = scale factor that determines overall strength of connections of type k. The scale factor may be modified dynamically during a run if the AUTOSCALE option (q.v.) is active for the cell type.

Contents

ppf = ppf(ij) = paired-pulse facilitation fraction for j'th connection of this type to cell i, 0 if this feature is not used.

c(ij) = strength of j'th connection of this type to cell i.

s(l(ij)) = activity of source cell connected presynaptically to j'th connection to cell i, i.e. the cell whose number is l(ij). If OPT=I is specified for the connection type, s(j) is replaced by s(j) - <s(j)>, where <s(j)> is a time average of s(j).

*%sjrev*(k) = Reversal potential for connections of type k.

*%et*(k) = "effectiveness threshold", a minimum s value that inputs to connection type k, after subtracting *%sjrev*(k), must exceed in order to have any effect on the postsynaptic cell. This parameter is also known as *%sjmin*. (An option is provided to apply this threshold in "STEP" mode, i.e. the test is applied, but the value is not subtracted when the test is passed. Separate tests for excitatory and inhibitory inputs are also available; see Group II PARAMS card.)

[ ] in the above equation indicates a threshold function, [x] = 0 if x <= 0, otherwise [x] = x. In addition, each term in the sum over k may be constrained to lie in a range between a given minimum (*%mnax*) and maximum (*%mxax*) value and may be subject to decay. Click here for more details.

Specific connections are defined by a CONNTYPE control card placed after the CELLTYPE card that defines the target cells. Because it is often desired to construct dendritic arbors focussed on a particular region of the source layer, generation of l(ij) matrices is specified using two code letters (*%kgen*). The first code indicates the method of generating the first (j=0) connection to each target cell (the location of the arbor), and the second code indicates the method of generating the remaining connections of this type (the form of the arbor). Typical first letter codes include (this is not a complete list):

- U: Select source cells Uniformly from the entire source layer.
- H: Select source cells from a rectangular area in the source layer containing several groups centered on the group in which the target cell is located (a "Hypergroup").
- G: Select source cells from the same Group in which the target cell is located.
- O: Select source cells from any group Other than the one in which the target cell is located.
- T: Select source cells from a region of the source layer corresponding to the location of the target cell in its layer, i.e. form a Topographic map. Code T, unlike codes G or H, can be used when the source and target cells are not in the same region.
- S: Form a topographic mapping from a moving window that Scans the visual input array, i.e. a retina.
- X,Y: Select source cells systematically from the source layer, beginning at a specified cell and incrementing by a specified amount for each subsequent target cell.
- E: Read l(ij) from a CONNLIST file (an External connection list). Connection list files are generally prepared with a program written by the user for the specific situation at hand. The format of the connection list file is given in an Appendix.

Typical second letter codes include:

- I: Each subsequent connection is generated Independently by reapplying the first-connection generating rule.
- A: Each subsequent connection Advances by a constant amount through the source layer.

Contents

- B: Successive connections are generated in the form of a rectangular Box or matrix on the source layer, beginning at the location specified by the first connection.
- C: Successive connections are selected randomly from a rectangular region surrounding the first connection, thus generating a typical "Crow's-foot" shaped arbor.
- Q: Similar to C except that a central area is excluded, thus forming an annulus of connections surrounding the site chosen by the first letter code.
- P: The source layer is Partitioned into *%nc* equal areas and each connection is selected from a different one of these areas. This rule guarantees that no two connections in an arbor have the same source cell (provided the number of connections is not greater than the number of source cells).

The CONNTYPE card accepts a number of parameters which are used with the various connection-generating rules to specify such things as the number of connections per cell (*%nc*), the size of the arbor (*%nrx*, *%nry*), the offset of the first connection from the origin of the source layer (*%offset*, *%xoff*, *%yoff*), and the amount by which generation advances on subsequent connections (*%stride*). Click here for more details.

Connection strengths, c(ij), can be allocated from two through sixteen bits in storage (*%bits*), allowing the user to control the precision of signal transfer within the simulated networks. Connection strengths can be assigned initial values in one of four ways: random (rule R), gradient (rule L), matrix (rule M), or read from a file (rule 2). (1) Random connections are drawn from a normal distribution with specified mean (*%mean*) and standard deviation (*%sigma*). A specified fraction of these connections (*%pp*) are excitatory; the remaining (1 - *%pp*) are inhibitory. (2) Gradient (ramp) connections are assigned c(ij) values according to the position of the source cell in its region in a bilinear gradient with specified values at the four corners of the source layer (*%c00*, *%c01*, *%c10*, *%c11*). There are several options for inverting or rotating the orientation of the map when a new target cell or group is encountered (*%mapopt*). These connections are generally used with motor neurons to provide a crude motor mapping which can be refined by training. (3) Matrix connections are assigned c(ij) values chosen from a set of matrices (*%set*) read from an input file. These are generally used with sensory neurons to provide feature detectors, such as "simple" cells in a model of V1 visual cortex. (4) c(ij) values can be generated by an external program and read from a file. This option can only be used in conjunction with reading l(ij) from the same file, so generating rule EI is required and implied as default with rule 2. Naturally, type L and M connections should only be used when they are consistent with the assumptions of a particular study.

Here is a typical CONNTYPE card that could be used to generate connections with random strengths and random source locations within a crude topographic map:

```
CONNTYPE VC V1 32 RTC MEAN=0.5, SIGMA=0.15, PP=0.75, BITS=6, NRX=12, NRY=12
```

Each connection will originate from a cell of type "V1" in region "VC". Each target cell will have 32 of these connections. Connection strengths are random (code R) with mean 0.5 and standard deviation 0.15. Three-fourths of these connections are excitatory and the remaining one-fourth are inhibitory. Each c(ij) has 6 bits of precision. The source of each arbor of 32 connections is mapped topographically to the input (code T); the connections are distributed randomly (code C) in a 12 x 12 area (NRX, NRY parameters) around the location selected by the topographic mapping.

Contents

Caution: Topographically mapped connections are drawn from a source area specified by the parameters *%nux* and *%nuy*. By default, these are set equal to the size of the entire source array, reduced by (*%nrx*-1) along x and (*%nry*-1) along y if a connection box (code B) is used. If the dimensions (in groups) of the source area are not the same as those of the target region, the program imposes a grid on the source with the dimensions of the target and selects each connection at random from whatever cells lie within the appropriate source grid box, weighting the probability of selecting from each group according to the area of the group that is contained in the grid box. This may result in rows or columns being skipped at random intervals or in cells from the same source group being mapped to different target groups. CNS issues a warning when this condition exists, but it is not an error.

Geometric Connections (represented by G in the response function) are constructed in concentric square rings (circular rings are on the wish list) around the target cell and are generally used to simulate lateral inhibition in neural systems. The source layer must be in a region that has the same shape (*%ngx*, *%ngy*) as the target region. The geometry and connection strengths are implicit (l(ij) and c(ij) values are not explicitly stored), and therefore geometric connections cannot be plastic. Their cost in memory and execution time is intermediate between that of specific and modulatory connections.

The value of the geometric input, G, for cell i is calculated as follows:

(Eqn. 3) $$G = \mathrm{Sum}(k)\{\mathrm{Sum}(r)\{\%beta(k,r)*\mathrm{Sum}(j)[-s(g(r,i,j))-\%gjrev(k)-\%it(k)]\}\}$$

where:

k = index that runs over geometric connection types.

r = index that runs over rings.

*%beta*(k,r) = scale factor for the r'th ring of the k'th geometric connection type. As with specific connections, excitatory (*%beta* < 0) and inhibitory (*%beta* > 0) inputs are summed separately and the totals are processed separately as described in Eqns. 1. Betas can be modified dynamically during a run if the AUTOSCALE option is active for the cell type.

s(g(r,i,j)) = activity of source cell connected to input j. The index number of this cell is g(r,i,j), a function that defines the neighborhood of cell i in terms of surrounding rings. For example, in the following diagram of a portion of a source layer,

```
... 32 33 34 35 36 ...
... 42 43 44 45 46 ...
... 52 53 54 55 56 ...
... 62 63 64 65 66 ...
... 72 73 74 75 76 ...,
```

the first ring surrounding cell 54 would consist of cells 43, 44, 45, 53, 55, 63, 64, and 65, and g(1,54,0) = 43, etc.

*%gjrev*(k) = Reversal potential for geometric connections of type k.

*%it*(k) = "effectiveness threshold", a minimum s value that inputs to connection type k must exceed in order to have any effect. (An option is provided to apply this threshold in “STEP” mode, i.e. the test is applied,

but the value is not subtracted when the test is passed. Separate tests for excitatory and inhibitory inputs are also available; see Group II GCONN card.)

[ ] in the above equation indicates a threshold function, [x] = 0 if x <= 0, otherwise [x] = x. In addition, each term in the sums over k and r may be constrained to have a value no larger than *%mxib*(k,r) and may be subject to decay. Click here for more details.

Geometrical connections are defined by the GCONN control card. This card specifies positionally the source of the connections, the number of rings, and their width. (The innermost ring always contains just a single group.) Each ring may have its own *%beta* and its own *%mxib*. Several "boundary condition" (BC) options are available for specifying the treatment of connections that would otherwise arise outside the source layer. Such connections can be omitted (BC=ZERO) (optionally normalizing the scale of the remaining connections, BC=NORM), connected to the nearest cell on the edge of the source layer (BC=EDGE), reflected back across the boundary to the interior (BC=MIRROR), translated toroidally to the opposite edge (BC=TOROIDAL), or connected to a noise source (BC=NOISE). Switch options can be used to expand the outermost ring at each position to cover the entire source layer (OPT=X) and to prevent cells from inhibiting themselves (OPT=V). Here is a typical GCONN card:

```
GCONN * * 3 2 IT=1.0, BETA=(1.0, 2.0, 4.0), BC=TOR, OPT=V
```

The "* *" indicates that the source of the connections is the current region and cell type. Each cell receives inhibition from cells in three surrounding rings; the annular width of the outer two rings is 2 groups each. Only inputs with s(g) > 1.0mV contribute to the calculation. The mean activity in the three rings is multiplied by 1.0, 2.0, and 4.0, respectively. Rings that extend beyond the edges of the region wrap around toroidally to the opposite edge. Cells do not inhibit themselves.

Modulatory Connections (represented by 'M' in the response function) provide a single input to all cells of a target cell type based on the sum of the activities of all cells of a specified source type. (Modulation can also be connected to external inputs if CNS is compiled to support BBDs.) Modulatory inputs are the cheapest connection type in terms of both memory and execution time because they only need to be calculated once in each time step. The value of the modulatory input, M, is calculated as follows:

(Eqn. 4) $$M = \mathrm{Sum}(k)\{\%mscl(k)*[\mathrm{Sum}(j)[s(j)-\%mjrev(k)-\%mt(k)]/n(k) - \%mto(k)]\}$$

where:

k = index that runs over modulatory connection types.
n(k) = number of cells in source layer indexed by k.
j = index that runs over all cells of source layer indexed by k.
s(j) = activity of source cell j.
*%mjrev*(k) = Reversal potential for modulatory connections of type k.
*%mt*(k) = "modulatory threshold", minimum s value that individual inputs to connection type k must exceed in order to have any effect. (An option is provided to apply this threshold in “STEP” mode, i.e. the test is applied, but the value is not subtracted when the test is passed. Separate tests for excitatory and inhibitory inputs are also available; see Group II MODULATE card.)

Contents

*%mto*(k) = "modulatory threshold-overall", minimum value that the average of the inputs to connection type k must exceed in order to have any effect.

*%mscl*(k) = scale factor for connection type k. Modulatory scales can be modified dynamically during a run if the AUTOSCALE option is active for the cell type.

[ ] in the above equation indicates a threshold function, [x] = 0 if x <= 0, otherwise [x] = x. In addition, M may be subject to decay. Click here for more details.

Modulatory connections are defined by the MODULATE control card. Here is a typical example:

```
MODULATE LGN OFF MT=1.0, MSCL=-10.0
```

With this card, the mean activity of all the "OFF" type cells in the "LGN" region with activity above 1.0mV is multiplied by -10.0 and this (inhibitory) contribution is added to all the cells of the cell type defined by the most recent CELLTYPE card.

Amplification and Value

Methods for changing synaptic strength ("amplification") in CNS are based generally on the TNGS. A great number of available variations on the basic rule permit the user to simulate many biologically realistic as well as purely theoretical learning rules. The "BCM" ("Bienenstock, Cooper, Munro) rule is also available. In its most usual form, the change in connection strength at each time step is given by:

(Eqn. 5) $$c(i,j,t+1) = c(i,j,t) + \%delta(k,r) * phi(c(i,j,t)) * (sbar(i,t-1)-\%mti(k)) * (m(j,t)-\%mtj(k)) * (v(vi,t)-\%mtv(k))$$

where:

c,i,j,k,t,phi have the same meanings as in the equations above, except that phi(c) is set to 1.0 when the sign of the change in c(ij) is different from the sign of c(ij). Other alternatives are available on the AMPLIF card. This factor is included to prevent runaway amplification.

*%delta* = a user-specified rate parameter, which may be positive or negative. This is the product of a general value, *%delta*(k), that may be specified for each connection type, k, and one of up to 8 different values, *%delta*(k,r) that is selected at each iteration according to the value of the rule selector, r. r depends on the signs of the factors (sbar-*%mti*), (m-*%mtj*), and (v-*%mtv*) as explained in "Method of Specifying the Amplification Rule" below.

r = rule selector, an integer, 0 <= r <= 7, used to select the value of *%delta*(k,r) to be used for each amplification event. r is the sum of 1 if (m-*%mtj*) < 0, plus 2 if (sbar-*%mti*) < 0, plus 4 if (v-*%mtv*) < 0. For example, if (v-*%mtv*) >= 0, (sbar-*%mti*) >= 0, and (m-*%mtj*) < 0, r is 1 and rate parameter *%delta*[k,1] is used for this connection.

sbar is just s(i,t-1) unless amplification rule S, T, or Z is specified (see below). Then sbar is the time averaged activity of cell i, calculated according to sbar(i,t) = *%sdamp**s(i,t) + (1 - *%sdamp*)*sbar(i,t-1), where *%sdamp* is a specified damping constant (default: 0.2; *%sdamp* is called "lambda" in some of the publications; DAMPFAC is a synonym for SDAMP on the DECAY control card). Note that the value of s(i,t-1) or sbar(i,t-1) is used--this allows amplification to be carried out before the new value of s(i) has

been calculated for the current time step. This may be changed in a later version of CNS to use s(i,t) or sbar(i,t).

*%mti* = postsynaptic amplification threshold for connection type k. Note that, when all other factors are positive, the change in c(ij) will be positive (potentiation) if s(i) > *%mti* and negative (depression) if s(i) < *%mti*.

m = is just s(j,t) unless amplification rule `S` or `T` is specified (see below). Then m is the concentration of a hypothetical postsynaptic "modifying substance" or "eligibility trace" produced at synapse (i,j) according to m(j,t) = m(j,t-1) + *%upsm**s(j,t) - Min(*%zetam**m(j,t-1),*%mxmp*), where *%upsm* = production rate for m, s(j,t) = presynaptic input to this connection, *%zetam* = decay rate for m, *%mxmp* = maximum decay rate for m. (See L.H. Finkel and G.M. Edelman, "Interactions of synaptic modification rules within populations of neurons," Proc. Natl. Acad. Sci. USA 82:1291-1295 (1985).) (The (m-*%mtj*) term is divided by 128mV if COMPAT=C is in effect.)

*%mtj* = presynaptic amplification threshold for connection type k. Note that, when all other factors are positive, the change in c(ij) will be positive (potentiation) if m(j) > *%mtj* and negative (depression) if m(j) < *%mtj*.

v = level of activity in value scheme *%vi*. The concept of "value" and the method by which it is calculated are described below. Activity in value systems is assumed to modulate the amount of synaptic change at affected synapses by an unspecified heterosynaptic or volume transmission effect.

*%vi* = value index. This is an integer specified by the user which selects any of the value schemes defined in the current run.

*%mtv* = value threshold for amplification. Note that, when all other factors are positive, the change in c(ij) will be positive (potentiation) if v > *%mtv* and negative (depression) if v < *%mtv*.

Alternative Form of the Amplification Rule Employing Combined Presynaptic and Postsynaptic Eligibility Trace. The variable m(j,t) as defined above incorporates only presynaptic information in the eligibility trace for amplification. By specifying amplification rule `U`, the user can instead employ a rule in which m(j,t) becomes m(i,j,t), incorporating a conjunction (product) of both presynaptic and postsynaptic activity at each instant. (This is an adaptation for rate-coded cells of a proposal by E. Izhikevich for spiking cells ("Solving the distal reward problem through linkage of STDP and dopamine signaling", Cerebral Cortex 17:2443-2452 (2007).) When this rule is in effect, Eqn. (5) in modified as follows:

(Eqn. 5a) c(i,j,t+1) = c(i,j,t) + phi(c(i,j,t)) * m(i,j,t) * (v(vi,t)-*%mtv*(k));

(Eqn. 5b) m(i,j,t) = m(i,j,t-1) + *%upsm* * *%delta*(k,r) * (s(i,t-1)-*%mti*(k)) * (s(j,t)-*%mtj*(k)) - Min(*%zetam**m(j,t-1),*%mxmp*)

Variables are as defined above. Because the value, v(vi,t), is not yet known when earlier terms are added into the running value of m(i,j,t) (Eqn. 5b), only the four values of r corresponding to positive value are used to select *%delta*(k,r).

BCM Rule. The BCM rule is described in many references, for example, M.F. Bear, "Mechanism for a sliding synaptic modification threshold", Neuron 15:1-4 (1995). In CNS, the basic form of the BCM rule is combined with features of the TNGS rule described above to give more flexible possibilities for experimentation. In CNS, the general form of the BCM rule is:

Contents

(Eqn. 5c) c(i,j,t+1) = c(i,j,t) + *%delta*(k,r) * phi(c(i,j,t)) *
s(i,t-1) * (s(i,t-1)-thm(i,k)) * (m(j,t)-*%mtj*(k)) * (v(vi,t)-*%mtv*(k));
thm(i,k,t+1) = thm(i,k,t) + *%qdamp* * (s(i,t)^2 - thm(i,k,t))

Variables are as defined above, with the addition of thm, the sliding threshold described in the literature, which is derived from a running average of the squared activity of the cell. In the standard form of the BCM rule, m(j,t) is just the presynaptic activity, s(j,t), and phi, way scales (see below), and the value term are all 1.0.

Method of Specifying the Amplification Rule. Amplification is specified by the AMPLIF (*%delta*, *%mti*, *%mtj*, *%rule*, *%upsm*, *%vi*, *%mtv*, and *%way* parameters) and DECAY (*%mxmp*, *%sdamp*, *%qdamp*, *%zetam*) control cards. If these cards are entered following a CELLTYPE card but before its first CONNTYPE card, they provide defaults for all connection types of that cell type. If entered following a CONNTYPE card, they apply only to that particular connection type. (Some amplification parameters can optionally be entered on PARAMS or THRESHOLDS cards.) A particular amplification rule is specified by a value for the overall rate parameter *%delta*(k) and either four (if value is not involved) or eight (if value is involved) values of the rate multipliers *%delta*(k,r) that are used for the eight possible values of the rule selector, r, as described above. (These are known informally as "ways", as in "there are 8 ways to amplify".) A RULE parameter permits rules to be entered in several fashions, depending on the amount of generality that is required. Possible values of RULE and the method of entering the *%delta* values for each are as follows:

RULE=B ("BCM"): The BCM rule is used. A nonzero value of *%delta* must be entered. Rule option Q along with a *%qpull* parameter described below may be combined with B to pull the thm value higher or lower than what is calculated by the rule; all the other options of the TNGS rule are also available.

RULE=E ("Explicit"): The user wishes to specify the eight rate parameters explicitly. A nonzero *%delta* and a WAY parameter must be given. WAY may take one of two forms: (1) WAY=(*%ws0*, *%ws1*, *%ws2*, ... *%ws7*). In this format, eight individual "way scales", *%ws0*, *%ws1*, etc., are given in parentheses, separated by commas. The overall value *%delta* is multiplied by each of the way scales to give the final amplification factors for the eight ways (values of 'r'). (2) WAY=xxxxxxxx. In this format, the eight way scales are entered in abbreviated form as a string of eight digits, 'xxxxxxxx', where the first 'x' gives *%ws0*, the second 'x' gives *%ws1*, etc. Each 'x' must be either 0 (the corresponding *%ws* is set to 0), 1 (*%ws* is set to 1.0), or 3 (*%ws* is set to -1.0).

RULE=H ("Hebb"): This is equivalent to RULE=E with WAY=10001000, the original Hebb rule. For this and the remaining rules, the given value of *%delta* is multiplied by the eight way scales implied by the rule. An explicit WAY parameter must not be given.

RULE=I ("S(i)" or postsynaptic rule): This is equivalent to RULE=E with WAY=11001000. Amplification occurs only when s(i) (or sbar) > *%mti*.

RULE=J ("S(j)" or presynaptic rule): This is equivalent to RULE=E with WAY=10101000. Amplification occurs only when s(j) (or m) > *%mtj*.

RULE=3 ("3-way" rule): This is equivalent to RULE=E with WAY=11101010. Amplification occurs in 3 of the 4 possible ways (all cases except when both pre- and postsynaptic cells are inactive) when value is above threshold. This is the most commonly used rule in CNS. Since its introduction in the 1982 Darwin II paper, it has been ignored and rediscovered by many other authors, e.g. K.D. Miller, "Synaptic economics: Cooperation and competition in synaptic plasticity," Neuron, 17:371-374 (1996).

Contents

RULE=4 ("4-way" rule): This is equivalent to RULE=E with WAY=11111010. When value is above threshold, amplification is positive when both pre- and postsynaptic cells are active or both cells are inactive and negative otherwise.

Some additional options can be coded in the RULE parameter--see the reference documentation for a full explanation. Among these are:

RULE=C (Supervised Categorization): This rule can be used to train a set of category recognition cells. The value term is included in the amplification rule as with RULE=V, but the value index *%vi* is ignored and instead the value is calculated from the cell number and the class identity 'ign' of the current stimulus sensed with the modality specified on the CONNTYPE card. If the cell number, modulo *%ncs* for the modality, plus 1, equals 'ign', the value is 1, otherwise the value is 0. The value threshold *%mtv* can be set to a value such as 1/*%ncs* so that the cell receives a large positive value when the cell number matches the category, otherwise a small negative value.

RULE=D ("Detailed statistics"): Prints the mean values of (sbar-*%mti*), (m-*%mtj*), and (v-*%mtv*) for the eight way classes. These statistics can be very useful for tuning a run.

RULE=N ("No sign change"): Prevents amplification from changing the sign of c(ij) and allows connection types to be preserved as either exclusively excitatory or exclusively inhibitory. The sign of the connection strength is preserved even when the value goes to zero.

RULE=P: Uses a running average of s(i,t-1) known as qbar as the postsynaptic amplification threshold in place of *%mti*. qbar is calculated the same way as sbar except using an alternative damping factor, *%qdamp*, also specified on the DECAY control card (Default 0.1). In combination with rule S or T, this allows amplification based on the difference between a rapid and a slow running average level of cell activity.

RULE=Q: Similar to rule P except the value of qbar is combined with *%mti* using a "pull factor" *%qpull* according to qbar(used) = (1 - *%qpull*) * qbar(calc) + *%qpull* * *%mti*. This may be combined with rule B.

RULE=R: Uses a running average of s(j) in place of *%mtj*, thus automatically setting the boundary between positive and negative connection strength change at the point of mean input activity. A parameter *%rdamp* is provided to specify how the average <s(j)> is obtained.

RULE=S ("Slow amplification"): Activates the form of Eqn (5) in which m(j,t) is used for presynaptic activity and sbar(i,t-1) for postsynaptic activity. Without RULE=S, s(j,t) is used for presynaptic activity and s(i,t-1) for postsynaptic activity.

RULE=T ("Timed amplification"): Activates a variant of rule S designed to prolong the "memory" of recent activity represented by m(j,t). As with rule S, time-averaged values of s(i) and s(j) are used for amplification. In addition, when m reaches a value *%mxmij*, it remains at *%mxmij* for *%mticks* cycles and then decays abruptly to zero. (*%mticks* is a parameter on the DECAY card, *%mxmij* is on the AMPLIF or PARAMS card.) This option effectively replaces the exponential decay of m(j,t) with a rectangular decay vs. time function.

RULE=U: Amplifies according to Eqns. (5a) and (5b).

RULE=V: The value term (v-*%mtv*) is included in the amplification rule.

RULE=Z: Uses sbar(i,t-1) in Eqn. (5) for postsynaptic activity, but s(j,t) for the presynaptic factor.

Here is an example of an AMPLIF card. This card applies to all connection types of the current cell type:

Contents

```
AMPLIF 3DN 0.33 MTI=10, MTJ=10
```

Rule 3 is used: connections are modified when either or both of the pre- and post-synaptic cells are active. Detailed statistics are provided. Connections are not allowed to change between excitatory and inhibitory. The rate constant *%delta* is 0.33 and the pre- and post-synaptic activity thresholds are both 10mV.

Value. In order for a system to display adaptive behavior, that system must be capable of sensing the consequences of its behavior in the external environment and of modifying its behavior accordingly. According to the TNGS, this requirement is met by the existence of specialized neural regions, known as value systems, whose activity reflects the global evaluation of recent behavior. This activity is assumed to be distributed by diffuse projections throughout large areas of neural tissue, where it may act to modulate the degree of synaptic change in other circuits. In CNS, this modulation is accomplished by multiplying the change in synaptic strength calculated in an otherwise typically Hebbian fashion by the strength of a relevant value signal (see Eqns 5 and 5a).

Activity in value systems may depend on environmental or internal sensors. Value systems may be directly manipulated in conditioning experiments through the provision of rewards. In CNS, many different ways to set up value systems or value schemes are provided by the Group I VALUE card. For convenience, some of these allow value to depend directly on external events in a way that simulates conditional rewards. For example, in reaching experiments, value can be made to depend on the decreasing distance of a hand from an object in the environment. It is also possible to associate a value with each of several objects in a visual stimulus library, such that a CNS value system will respond with the appropriate value when one of those objects is viewed or touched. In the special case of RULE=C, different cells receive different value, allowing a class of supervised-learning algorithms to be simulated. Value is immediate and the information in the next paragraph does not apply.

Value is calculated in three stages, once per trial for environmental value and once per cycle for internal (cell-based) value. First, the selected value scheme is executed to yield a measure of the value of the current situation. Call this number Vc. Next, the value of the recent behavior is evaluated as the scaled change in Vc from the previous time step. A minimum value, *%vmin*, may be substituted for the previous value to give larger penalties for invariant poor responses. Call the result Vb, where Vb = *%vscl**(Vc(t) - Max(*%vmin*,Vc(t-1))). (If absolute rather than differential value is wanted, OPT=A may be coded, in which case Vb = *%vscl**Vc.) Finally, a damped running average of Vb is maintained for use in amplification. Call this value Va, where Va(t) = *%vdamp**Vb(t) + (1-*%vdamp*)*Va(t-1). Va is the quantity plotted on superposition plots (PLOT=S on CYCLE card) and it may also be printed (DP=V on CYCLE card). When value is used as a virtual input to a connection type (*%srcid*=V on CONNTYPE, MODULATE, or NMOD card) with COMPAT=C, the value is restricted to the range 0 <= Va < 1.0. (The parameters *%vmin*, *%vscl*, and *%vdamp* are entered on the VALUE card.)

Introducing value to the sample AMPLIF card given above, we might have:

```
VALUE NFIRE RET FOV BEST=50, HWIDTH=25
... other Group I cards go here ...
REGION ...
CELLTYPE ...
AMPLIF 3DNV 0.33 MTI=10, MTJ=10, VI=1, MTV=0.5
```

Contents

The AMPLIF card specifies (VI=1) that value is taken from the scheme defined on the first VALUE card and that value above 0.5 will be taken as reward. The VALUE card specifies that value is based on the number of cells firing in the the "FOV" cell type of the "RET" region. Value will be 1.0 when 50 cells are firing, falling off to 0 with a half-width of 25 cells, i.e. value will be 0 when either 0 or 100 cells are firing. Click here for a more detailed discussion.

Freezing Amplification. There are several reasons why one might want to freeze connection strength changes during certain cycles of a simulation:

(1) To collect statistics "uncontaminated" by "learning" during performance tests. This can be accomplished by use of the *%frzint*, *%frztrials*, and *%frzcycles* parameters on the CYCLE card. All connection strength changes are globally frozen on every *%frzint*'th series of trials, which comprise *%frztrials* trials and *%frzcycles* cycles per trial, rather than the usual *%ntr* trials and *%ntc* cycles. By default (i.e. when no *%frzint* parameter is entered), the last trial series is frozen and, additionally, an extra "zero'th" frozen series is performed at the beginning of the run to obtain initial performance statistics prior to any training.

(2) To allow responses to stabilize. When more than one cycle is performed for each stimulus presentation (trial), it is usually desirable to delay amplification until the network has had one cycle to respond to each new stimulus, and accordingly, amplification is deferred one cycle in each trial by default. (This delay is essentially an artifact forced by the use of s(i,t-1) rather than s(i,t) in the amplification function. The need for this will be removed in future versions of CNS by calculating s(i,t) prior to amplification and using it in the amplification function.) The number of amplification deferral cycles can be specified by the ADEFER option on the AMPLIFICATION card or globally on the Group I RESPONSE FUNCTION card. Recent versions of CNS automatically set *%adefer* to 0 to get any amplification at all when the number of cycles per trial (*%ntc*) is 1.

Noise

Noise in CNS is a random value which may be added to cellular responses (Group II NOISE card) or to visual inputs (currently valid for monochrome environments only--SNOISE card). The noise is drawn from a Gaussian distribution with specified mean (*%mean*) and standard deviation (*%sigma*). By making the mean large, but applying the noise only to a designated fraction of the cells in each cycle (*%frac*), a form of shot or spike noise can be generated. By making the standard deviation 0 and the fraction 1, a constant bias is added to each cell. It is possible to cause noise levels to be modulated by activity in a designated cell type or external input (NMOD card). This is a computational convenience which can sometimes eliminate the need to create a cell type purely for the purpose of injecting noise into some other cell type.

For example, either of the following NOISE cards:

```
NOISE 10 1 0.05
NOISE MEAN=10mV, SIGMA=1, FRAC=0.05
```

would cause a noise input ('N' in Eqns. 1) selected from a normal distribution with mean 10mV and standard deviation 1mV to be added in each cycle to 5% of the cells in the current cell type.

Contents

Controlling the Simulation

The Group III CYCLE card controls the running of a CNS simulation. The first three parameters on this card, *%nts*, *%ntr*, and *%ntc*, specify the numbers of trial series, trials, and cycles per trial to be performed (these terms have already been defined above). The *%frzint*, *%frztrials*, and *%frzcycles* parameters control insertion of frozen trial series. The PLOT parameter gives overall plotting options. For example:

```
CYCLE 8 500 4 FRZINT=4, FRZCYC=2, PLOT=S
```

This card specifies that 8 trial series of 500 trials each are to be performed. Each trial has 4 cycles of simulation of the neural networks. On every 4'th series, amplification is frozen and only two cycles occur. Plots for all regions are superimposed, not separate.

After each CYCLE card completes, the user has an opportunity to change various stimulation and neural response parameters by entering a new set of Group III control cards. The CHANGE card is particularly useful in this regard, as it permits most of the dynamical network parameters to be varied. After all the changes have been entered, another CYCLE card can be used to continue the simulation.

A CNS cycle does not necessarily correspond to any particular interval in real time, although a time interval can be entered on the CYCLE card which may be used in certain specified circumstances. By choosing appropriate values for growth and decay parameters, the user can make a CNS cycle represent any desired unit time interval. Decay of neural activity (omega parameter in Eqns. 1) is treated specially. Two separate values for omega can be entered on the Group II DECAY card: *%omega2* operates before each cycle as in Eqns. 1, while *%omega1* specifies an additional decay which occurs before each new trial when level 2 trial reset is specified (see below). This allows the user to simulate an intertrial interval during which neural responses are allowed to relax to background levels before presentation of the next stimulus. This is purely a convenience to reduce computation time. (A level 3 trial reset additionally applies decay to depression, sbar, m(ij), etc. variables.)

## Senses and Effectors

Sensory inputs in CNS may be real or simulated. Real inputs come from external devices, such as video cameras (TV or CAMERA control card) or a robot (SENSE card). Camera images can be filtered by specified preprocessor kernels for purposes of smoothing or feature extraction. Simulated inputs are generated by a separate module within CNS, the "environment package". The available simulated senses are vision, touch, and kinesthesia. One or more senses may be treated together as a "modality" for purposes of collecting the categorization statistics discussed later. These statistics can be reported separately for each modality from which a given cell type receives input, directly or indirectly via other cell types.

Contents

Inputs from senses of all kinds are presented to the model neural system in the form of simulated neural activity on a scale of -128 to +128mV (0 to 1 in COMPAT=C mode). These inputs may be attached to ordinary cells via any of the connection types already discussed. Sensory cells are referred to as "virtual" cells because their output is set directly from the corresponding sense value--they have no connections from other cell types and no internal dynamics. Activity of cells in virtual regions can be plotted in superposition plots.

Motor outputs in CNS may also be real or simulated. Real outputs may be connected to robots or other devices via the BBDS ("brain-based-device server") interface, which communicates via network sockets with devices controlled by software written by the user. The simulated effectors are those used in Darwin III: "windows", which correspond to the field of view of an eye, and which may be made to move around in the visual world to simulate visual saccades; and two-dimensional multijointed arms, which are useful for studies of the neural control of reaching and grasping.

Methods of setting up simulated senses and effectors are discussed next. Control card examples may be found in the sample jobs. Methods of working with real sensors and effectors are beyond the scope of this Introduction. Details may be found in Chapter 9.

## Input from the Simulated World

Connection of cells to sensory inputs is accomplished by providing the name of the appropriate sense in place of a region name on a control card that defines a connection type. Thus, senses and ordinary neural regions may be said to share a common "name space". It is best to avoid giving regions the same short name as any of the built-in senses, but in case of conflict, the user-generated name will be used if the name is placed in parentheses and the built-in name if parentheses are not used. Optional interfaces (i.e. those that depend on compile-time options) may additionally use the region names "D1" for Darwin 1 model inputs, and "BBD" for user-defined BBD devices.

Vision. Simulated visual input may be monochrome or colored and is provided in the form of 8-bit luminance levels on a rectangular "input array" (Group I INPUT control card) similar to the raster provided by a video camera. The x and y dimensions of the input array must be powers of two. With colored input, the available 8 bits are allocated 2 to blue, 3 to green, and 3 to red. Images input via the BBD or TV interfaces can use 8-, 16-, or 24-bit color modes. The color sensitivity of each connection type receiving visual input can be specified as one of the three primary colors, corresponding to input from one of the three classes of color-selective retinal cone cells. Random noise can be added to the input array (SNOISE card), but at present this works correctly only with monochrome stimuli. These restrictions on colored virtual input can be overcome, and the available palette increased to 24 bits, by dividing the input array into three separate areas and providing 8-bit stimuli in one of the three primary colors to each area.

To permit simulation of visual saccades, the input array may be viewed (connection generating code S) through one or more "windows" (Group I WINDOW control card) corresponding to retinas. Windows may be made to move across the input array under control of simulated motor output cells. Several windows of different sizes may be "piggybacked" together, permitting simulation of foveal and peripheral vision. Alternatively, the input array or some part of it can be viewed directly.

Contents

Objects may be placed on the input array and may be made to move around it in various ways. Objects may be created by scanning images with an optical scanning device, by manual coding, or by using a bitmap (icon) editor. Scanned images are converted to a format read by CNS (the PIXEL file) by the utility program PIXELPRP. When read into CNS, images generally cover the entire input array and cannot be moved. Smaller objects may be hand coded onto Group III SHAPE cards or drawn with a bitmap or icon editor and converted to SHAPE cards with the i2s utility program provided with CNS. (i2s was written by J. Wray.) SHAPE cards may be read directly into CNS, but multiple shapes are generally collected into LIBRARY files from which the desired shapes can be selected at run time, either randomly or in various fixed sequences. The utility program D3LIBPRP converts SHAPE cards to shape libraries.

No matter which method of encoding is used, stimuli are placed onto the input array during a run by use of the Group III CREATE or SELECT cards. The CREATE card causes a single object to be placed on the input array where it remains for a fixed number of simulation cycles and is then removed. Parameters on the CREATE card allow the particular object and its location and duration to be specified. The SELECT card allows a series of objects to be selected and presented in a repeating cycle. The objects are taken from a specified range of objects in a LIBRARY file. Selection may be in sequential or random order. Random selection may occur with replacement (each choice is independent) or without replacement (a randomly permuted order of the input objects is generated). When the trial series is concluded, the same order can be repeated for the next trial series or a new order can be generated. The maximum number of objects to be placed simultaneously on the input array during a run must be specified on the Group I PLIMITS card.

The boundaries of the input array can be dealt with in various ways during object placement: Smaller boundaries may be set so that no part of an object ever goes out of bounds, or parts of objects that fall out of bounds may be deleted or may be made to wrap around to the opposite edge (toroidal boundary conditions).

Each CREATE or SELECT card carries a "reference number" that allows it to be coupled with a MOVE card that controls movement of the object once it has been placed on the input array. Possible types of movement include linear translation, rotation, oscillation (according to a cosine or square-wave function of time), a combination of linear motion and perpendicular oscillation (boustrophedon) that uniformly covers a rectangular portion of the input array, or random jumps at specified intervals. Boundary conditions like those for object placement may also be specified for object movement, but these are entirely independent and provide additional options, such as the ability to have an object "bounce" back (mirror-reflect) when it reaches an edge of the input array.

Object reference numbers can also be used to refer to objects on TIMER control cards. The largest reference number to be used during a run and the largest number of objects to be placed on the input array at any one time must be specified in advance on the Group I PLIMITS card.

It is possible to have an object react to contact with an arm (see the description of Darwin III in the Proc. IEEE paper cited above). When an object is hit, it recedes to the edge of the input array and disappears. If the object was generated by a SELECT card, a new object is selected and presented a fixed time after the previous object has disappeared.

Contents

Touch. Touch sensors may be provided on any of the segments of a simulated arm and connected to neural inputs (VT virtual sense name). (Usually only the last segment, the "hand", is so equipped.) Touch receptors respond to any object on the input array that is directly under their current location. The magnitude of the touch response ordinarily is determined by the highest "lightness" value on the input array under the touch array. There is an option to make touch "pressure-sensitive", in which case the response is maximal when only one receptor is touched and drops off to half-maximal when all the receptors on the touch array are stimulated together.

Objects on the input array may be placed at different depths (Z coordinate on CREATE or SELECT card). Vision responds to objects at all depths, but touch responds only to near objects. Additionally, touch receptors may be designated as being sensitive only to one particular "color". Either of these methods allows the user to maintain separate visual and touch objects on the input array at the same time.

Kinesthesia. Kinesthesia in CNS provides neural input that is sensitive to the motions of effector organs, either "windows" (eyes undergoing saccades, VW virtual sense name) or arm joints (VJ virtual sense name). Alternatively, Efferenzkopie can be simulated simply by connecting an input of a sensory cell type to the output of a motor neuron cell type. Kinesthesia differs from Efferenzkopie in that it reflects the fact that a commanded muscle motion may fail to occur, perhaps because the eye or joint involved had already reached the limit of its travel in a particular direction.

## Effectors in the Simulated World

Windows. As already mentioned, windows are movable ports through which a part of the input array can be viewed by a model system. The movement of a window is intended to simulate a visual saccade and is ordinarily controlled by output from a designated motor cell region. However, for test purposes, the movement of a window may be tied to the movement of an object on the input array. The controlling object may itself be invisible. Several windows may be "piggybacked" together, permitting simulation of foveal and peripheral vision.

Arms. Arms in CNS exist only in the two-dimensional x,y world of the input array. It is possible to create as many arms as desired (ARM control card), and each arm may have as many joints as desired. One end of every arm is fixed to one of thirteen possible attachment points, which are located at the center of the input array, 45-degree points around the periphery, and points halfway between the center and the edges. There is also an option to attach an arm to a window, which might be useful in models of animal navigation. The free end moves in accord with the joint angle positions commanded by activity in specified motor neuron regions connected to each joint. The movement of each joint can be limited to specified angular ranges (JOINT control card), and the tip of the arm can be constrained to remain always within the input array if desired.

Arms may be given the ability to grip objects on the input array. This is done by specifying AOP=G on the ARM card. A following GRIP card specifies other necessary parameters. Gripping is initiated when an object overlaps the last segment of the arm ("hand") and when activity in a specified value system exceeds a given threshold. In this situation, an internal, not an environmental, value scheme is expected to be defined, reflecting activity in some collection of cells which in the model are responsible for the initiation of gripping. This arrangement is a shortcut to eliminate the need to define and control finger joints in a complicated way to achieve gripping.

Contents

Arms also have an alternative mode of motion known as "canonical trace mode" that was used in Darwin III to simplify the arm's tracing of the edges of objects. In this mode, the arm becomes stiff and movements of the tip are controlled by a single "universal joint" at the point of attachment. It is unlikely that this mode will be needed in new models, however, the facility remains in CNS. Click here for details.

Method for Specifying Motor Arbors. A common method is used in CNS to attach all real or simulated effectors to the motor neuron regions that control them. Each effector is associated with a pair of cell types which move it in opposing directions. These are referred to on the `EFFECTOR` control card as the "excitatory" (`EXCIT` keyword) and "inhibitory" (`INHIB`) cell types, although in general these terms mean only that the "excitatory" cell type moves the effector in an arbitrarily defined positive direction and the "inhibitory" cell type moves it in the opposite direction. Arbitrary lists of cells, known as "motor arbors," may be connected to the excitatory and inhibitory outputs. These arbors may arise in the same or different cell types. If the cell types are different, the cell lists may be the same. The default is to use all the cells of the specified cell type. The sum of the activity in these cells, multiplied by a given scale factor (various keywords are used), determines the amount of motion of the effector at each cycle of the simulation. It is also possible to send the individual cell data to a BBD client.