Chapter 5: Low Level Details of the Grasping Model

In this chapter, the region-level model presented in the previous chapter is expanded to a neural level of detail. We first introduce a computational structure, referred to as a μ-schema, that serves as an intermediary between the schema level of analysis and the neural level of implementation. Making use of the μ-schema mechanism, we turn to the implementation of the model, and discuss 1) how information is represented within each region, and 2) how individual computing units are connected in micro-circuits to implement particular aspects of the neural dynamics.

In addition, the basal ganglia are implicated as a mechanism that is responsible for the general management of schemas. In particular, the basal ganglia are involved in this model in three distinct ways:

1. Arbitration between competing schemas in F5 and AIP.
2. Managing the phasing that takes place within F5.
3. Providing a mechanism through which the selection of a grasp may be biased as a function of either task requirements or abstract stimuli.

μ-schemas: An Intermediate Computational Mechanism

Up to this point in our modeling discussion, we have thought of the various components of our motor program as abstract entities (schemas) that exist within a layer (or set of layers), but we have not yet significantly addressed the issue of how these entities might be implemented at the neural level. The schema level of analysis is useful because we have been able to treat different programs, such as those for power grasp and precision pinch, as discrete and encapsulated entities. This allows us to assume essentially independent implementation and execution of these entities. Combined with the concepts of schema priming, cooperation/competition, and activation, this level of analysis significantly simplifies the implementation and our understanding of the resulting system.

In moving down to the neural level of analysis, we would like to maintain some notions analogous to the high-level computational concepts that we used at the schema level. However, two factors conspire against this. First of all, the discreteness assumption begins to break down. We have already seen in the analysis of the Sakata data how an individual AIP cell might participate in several distinct grasps. In fact, the level of participation is graded: for some grasps, the cell responds more strongly than others. The second difficulty is that since we have assumed that an appropriate level of the neural model is the leaky-integrator (Arbib, 1989), we necessarily lose the specialization of the various inputs that we had at the schema level (i.e., priming, competition/cooperation, and activation).

In order to address these difficulties, we introduce the notion of a μ-schema (pronounced "micro-schema"), which is a computational unit that acts as an intermediary between schemas and neurons. Physically, an individual μ-schema is composed of a set number of leaky-integrator neurons, each with a specialized function. A schema is then constructed by gathering together a set of μ-schemas. However, it is possible for two schemas to share a single μ-schema (we would refer to these schemas as *overlapping*). In addition, a μ-schema's participation in a schema can be graded. In other words, its contribution to a schema's output may range continuously from significant to not at all.

A μ-schema differs from the concept of a sub-schema (already provided by schema theory) in that sub-schemas are themselves schemas. μ-schemas, on the other hand, are considered atomic elements and cannot be decomposed further without losing the properties of the μ-schema.

In the remainder of this section, we first show the μ-schema implementation using a group of neurons, referred to as a *column*, and then show how columns within the different regions connect together to implement the neural programs described in Chapter 5.

The Columnar Structure

The columnar structure of the μ-schema is designed such that the schema theoretic concepts of priming, mutual cooperation/competition, and activation can still be used in the design and analysis of our neural models. In addition, the column is intended as a generic computing unit; every brain region in our model is made up of a set of these columns (as will be discussed, some configurational differences exist from one region to another).

The column consists of a set of four neural units (Figure 5.1). The primary function of the column is performed by the *signal unit*. This unit fires in response to the presentation of a specific set of *trigger inputs*. Trigger inputs can be, for example, *go signals* (as is the case for F5) or some combination of object features (as in AIP). Activation of the signal unit can cause the *output unit* to fire, which results in the generation of an output from the column. When this happens, we refer to the column as being *active*. The column output, depending upon the brain region, is connected either to other columns (in the same or other layers) or to actuators.

The simple feature detection that is performed by the signal unit is subject to two levels of constraints, without which the column cannot become active. The first constraint is that the column must be *primed*. Priming signals (detected by the *priming unit*) are typically received from high-level regions. For example, when F6 begins to prepare a grasp program, it will prime all the columns within F5. In this architecture, priming signals are considered to be non-specific; the priming of F5 by F6 prepares the system to perform a grasp, but does not necessarily specify the details of the grasp.

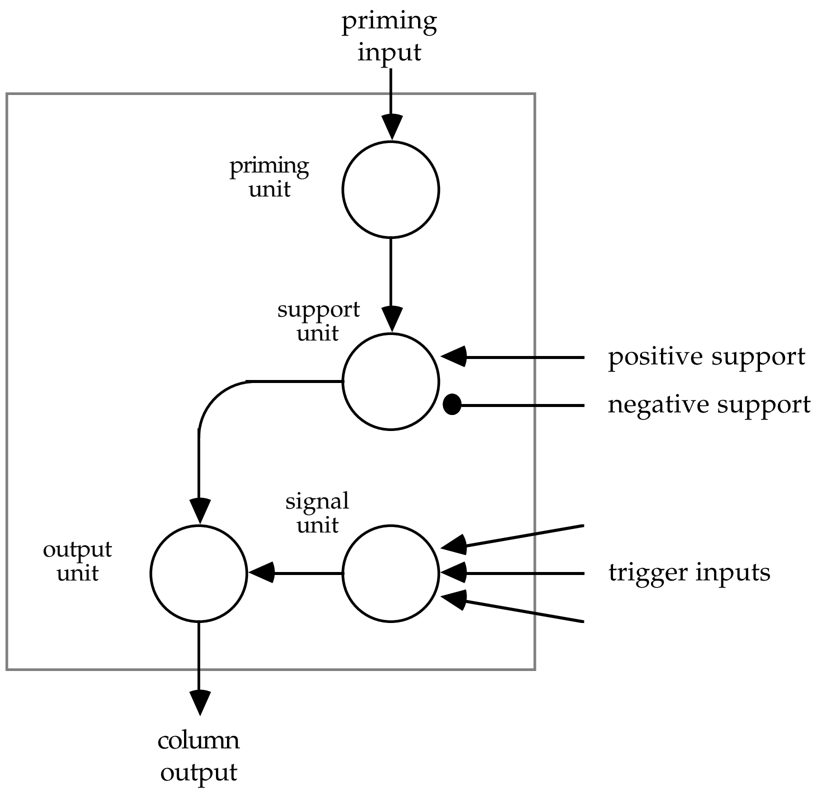

Figure 5.1: Column architecture.

The second level of constraints, received through the *positive/negative support* inputs, are used in two different ways. First, in order to enforce the coactivation of columns that participate in the same schema (e.g., a power grasp), positive connections are established from each column output to the support input of the other columns. Likewise, when two columns are involved in different schemas (e.g., a power and a precision grasp), mutually inhibitory connections are established. Second, when a higher-level region (e.g., area 46) selects a very specific schema within another region (a grasp in F5 as a function of some task constraints), it will establish positive connections to the support input of the columns participating in the schema.

The constraints are considered to be satisfied only when the column is both primed and receiving (at the very least) non-negative support. When this occurs, the column is prepared to become active when the correct trigger input is received.

The cortical column model proposed by (Alexandre, Guyot, Haton, & Burnod, 1991) is a connectionist-level model that attempts to capture the basic computations that are performed by small assemblies of real cortical neurons. This model has a number of interesting similarities to the μ-schema structure presented here. First, the state of the fundamental computing unit (the column) is multi-dimensional; the outputs that are generated by the column are affected by the state of the two internal neurons (although the individual neurons can have only one of three states). Second, the inputs are segregated into two classes, the contribution of which depends upon the entry point into the column. In contrast, a self-organization-based learning algorithm has been proposed that relies on Hebbian-like mechanisms to adjust the column's connectivity with other structures.

Column Dynamics

The equations that describe the dynamics of the column are given below. For computational efficiency reasons, only the output unit is implemented as a leaky-integrator neuron (Arbib, 1989). All other units in the column are implemented in a simple feed-forward manner.

The state of the *priming unit* is computed according to:

$$prime\_mem = prime\_inputs \cdot prime\_weights \tag{5.1}$$

$$prime = \mathrm{Esat}(prime\_mem, p\_base) \tag{5.2}$$

where:

$prime\_mem$ is the membrane potential of the priming unit.

$prime$ is the firing rate of the priming unit.

$prime\_inputs$ is a vector of inputs from other columns.

$prime\_weights$ is a vector of corresponding connection strengths.

p_base is the minimum firing rate when a non-zero priming input is received.

$$\mathrm{Esat}(x, base) = \begin{cases} 1 + \mathrm{base} & 1 \le \mathrm{x} \\ \mathrm{x} + \mathrm{base} & 0 < \mathrm{x} < 1 \\ 0 & x < 0 \end{cases} \tag{5.3}$$

The *support unit* state is computed as follows:

$$support\_mem = prime + support\_inputs \cdot support\_weights \tag{5.4}$$

$$support = \mathrm{Esat}(support\_mem - s\_threshold, s\_base) \tag{5.5}$$

where:

$support\_mem$ is the membrane potential of the support unit

$support$ is the firing rate of the support unit

$support\_inputs$ is a vector of inputs from other columns.

$support\_weights$ is a vector of corresponding connection strengths.

$s\_threshold$ is the unit's threshold.

s_base is the minimum firing rate when positive support is received.

$s\_threshold$ and p_base are chosen such that the support unit can only fire if 1) a priming input is received and 2) the input from the support connections is at least non-negative.

The state of the *signal unit* is computed accordingly:

$$signal\_mem = signal\_inputs \cdot signal\_weights \tag{5.6}$$

$$signal = \mathrm{NSLramp}(signal\_mem - sig\_threshold, sig\_base) \tag{5.7}$$

where:

$signal\_mem$ is the membrane potential of the signal unit.

$signal$ is the firing rate of the signal unit.

$signal\_inputs$ is a vector of inputs from sensory inputs or other columns.

$signal\_weights$ is a vector of corresponding connection strengths.

$sig\_threshold$ is the threshold of the signal unit.

sig_base is the minimum firing rate when a non-zero signal is received.

$$\mathrm{NSLramp}(x, base) = \begin{cases} \mathrm{x + base} & 0 < \mathrm{x} \\ 0 & \mathrm{otherwise} \end{cases} \tag{5.8}$$

The output state of the column is computed according to:

$$\tau \frac{d\ output\_mem}{dt} = -output\_mem + support + signal \tag{5.9}$$

$$output = \mathrm{threshexp}(output\_mem - o\_thresh, o\_gain) \tag{5.10}$$

where:

$output\_mem$ is the membrane potential of the column's output unit.

$output$ is the firing rate of the output unit. It is this value that is connected back into other columns.

o_thresh is the output unit threshold.

$$\mathrm{threshexp}(x, gain) = \begin{cases} x^{gain} & 0 < x \\ 0 & otherwise \end{cases} \tag{5.11}$$

o_gain determines the steepness of the activation curve. In the model presented here, the two values used (depending upon the region) are 0.45 and 1.

o_thresh, sig_base, and s_base are set such that both the support unit and the signal units must both be firing in order for the output unit to fire.

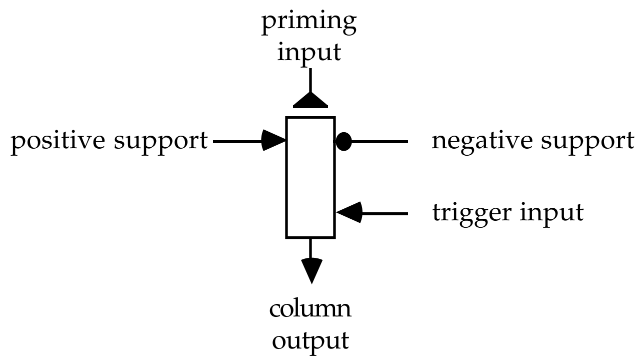

Figure 5.2: Convention used for drawing columns.

Connectivity of Regions

In this section, we define the representation of information that is used in each region and how the columns within the different regions are connected together. In what follows, columns are drawn as rectangles, as shown in Figure 5.2. Priming inputs are drawn as entering the top of the rectangle. Positive and negative support inputs enter from the sides of the upper half of the rectangle; signal inputs from the lower sides. Outputs from the column exit from the bottom.

Object Representations in PIP and IT

The different object classes are represented in PIP by separate populations of neurons (Figure 5.3). Within each population, we use a unit that codes for the general recognition of the shape itself, as well as sub-populations of units to code for the object's parameters. The parametric information is coded using a *population code* (Georgopoulos, Ashe, Smyrnis, & Taira, 1992; Georgopoulos, Kalaska, Caminiti, & Massey, 1982). Each of the neurons responds maximally to a particular parameter value (referred to as a *preferred value*). The activity of a parametric neuron is related to the difference between its preferred value and the actual value being coded (in the case of this model, the activity level is related through a Gaussian function with a specified standard deviation). Note that as it is used here, the term *population code* is not as

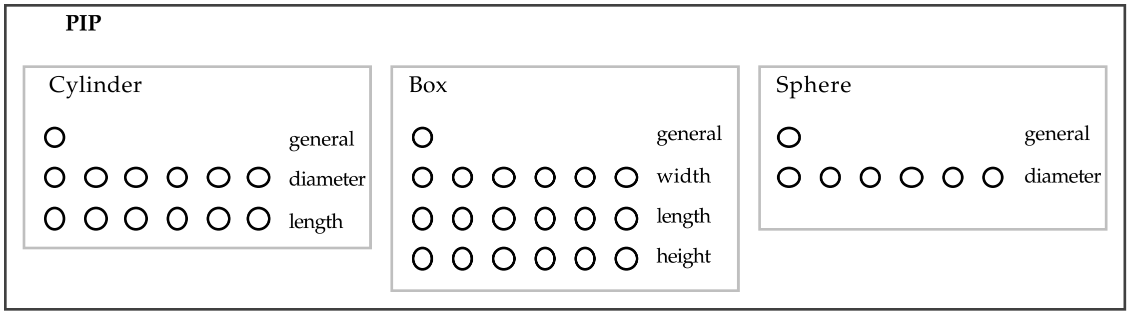

Figure 5.3: Illustration of PIP coding. Every object class is represented by a population of units. Sub-populations of units code for the object parameters.

strict as is used by Georgopoulos. Here, we relax the cosine tuning function requirement (replacing it with a Gaussian), and do not necessarily interpret the coded parameters in a peripersonal coordinate system.

Note also that each fundamental object class maintains its own set of parameter sub-populations. As a result, the diameter of a cylinder and the diameter of a sphere are coded using non-overlapping sets of cells. However, this is not to say that the space of all objects can be partitioned into discrete sets. Rather, we imagine these sets as defining a space of objects; objects that mix properties of two classes will have PIP representation that recruits from both sub-populations.

As is noted in Chapter 5, individual cells in IT appears to capture complex combinations of object features (such as shape and color of the object). This combined with the observation that object representations appear to be rather sparse (Logothetis & Pauls, 1995), implies that relatively different objects are coded by disjoint groups of neurons. In the model, we further simplify the coding by dropping the view-direction dependence (for rotations out of the plane) (Logothetis, et al., 1995), and assign a single unit to each object with which the model is "familiar."

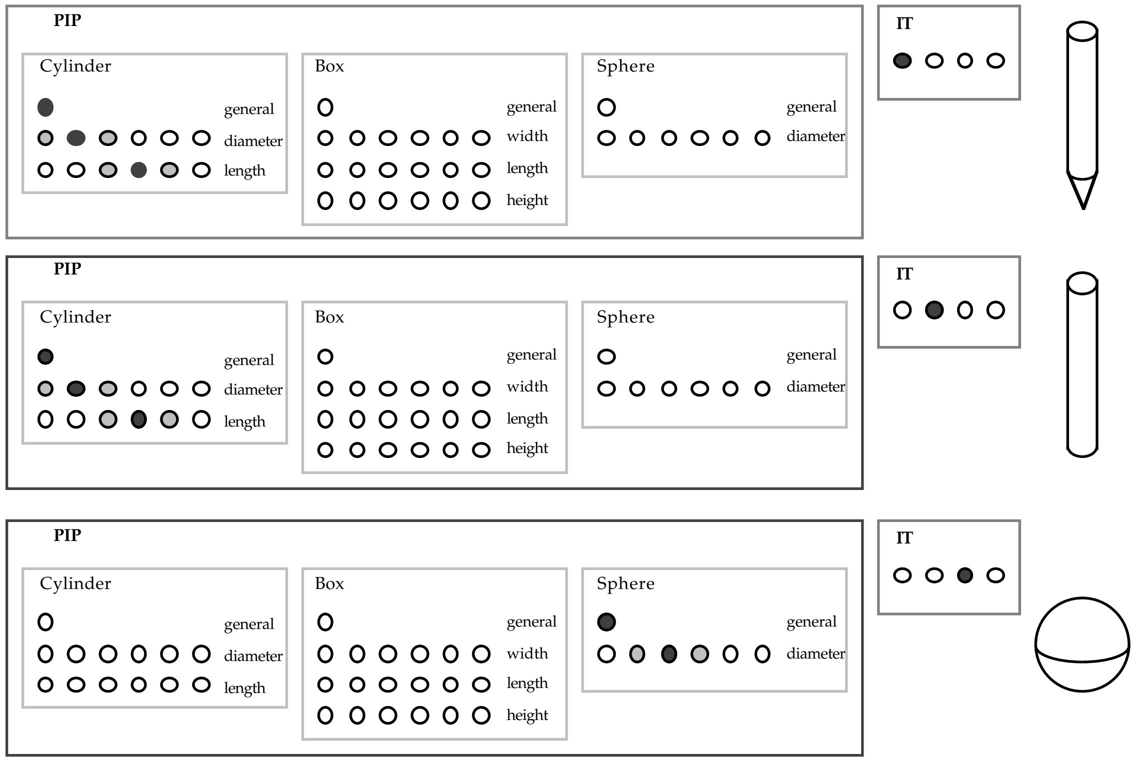

Figure 5.4: Example coding of three objects (pencil, stick, and a ball) in PIP and IT.

These two distinct representations are illustrated in Figure 5.4 for three different (familiar) objects. Note that because the pencil and stick are roughly the same shape and size, they are coded by PIP using identical activity patterns. However, the respective codes are orthogonal in IT. This simplification in the model captures the notion that these objects are *conceptually different* from one another. In the more general case, we would expect that IT would represent conceptual classes of objects in an overlapping manner (e.g., the set of all different mugs). Also, note in the Figure that the medium sphere is coded in a manner that is completely orthogonal (in both PIP and IT) to the stick and pencil.

AIP

In this section, we will first discuss how individual AIP units are described behaviorally. Then, we will show how these behavioral descriptions give rise to projections from PIP and IT, as well as connections that implement cooperation and competition between different AIP cells.

Classes of Neurons in AIP

AIP columns in the model are described according to the behavioral parameters defined in Table 5.1. As we will show below, these parameters determine how individual AIP columns are connected to each other and to columns in other regions. In the following sections, we will outline part of this connection process (for more details, see Appendix A).

Projections from PIP

An affordance derived from PIP maps a very specific object configuration to one possible grasp for that object. Figure 5.5 illustrates two examples of such affordances. Column B (of AIP) is a general precision grasp unit. It is responsible for detecting a cylinder that is *both* narrow and short. The neurons in PIP that code for both of these parameter values project to the signal input of column B. When column B activates in response to the pair of inputs, it generates an output. The resulting signal provides positive support to column A, which is responsible for coding the aperture width for the precision grasp. In this case, the aperture width is derived from the diameter of the cylinder.

Likewise, column C detects wide and short cylinders, while column D captures the appropriate aperture width. Note that the behavioral specification (a precision grip) for columns B and C could be identical. The differentiation into the two different grasps (narrow precision versus wide precision) occurs because a single object configuration does not activate all appropriate AIP columns. Rather, only a randomly selected subset of matching (i.e., appropriate) columns will receive input from a single object configuration. It is important to

visual: [0,1]. Describes the degree to which this is a visual-related cell (Taira, et al., 1990). This parameter determines the probability and strength of connections from PIP and IT.

motor: [0,1]. Describes the degree to which this is a motor-related cell. This parameter controls the probability and strength of connections from F5

phase: {set, extension, flexion}. The phase (AIP) during which the cell begins to fire.

grasp type: {precision, power, side, and finger prehension}. The class of grasp in which this unit will participate.

aperture: either *none* or from the range [0,100mm]. Determines the preferred aperture for the column (if there is a preferred aperture).

Table 5.1: Description of AIP cell behavior.

note, however, that it is still possible for multiple object configurations to activate the same column.

The necessary conditions for connecting an object configuration to a non-specific column (no aperture specificity) are (referring to the behavioral description of the column):

1. An affordance exists that maps the object configuration to a grasp of the column's grasp type.
2. The column is a visual-type unit (the degree of ***visual*** response is non-zero).
3. The column's phasic response is of type *set*.

These conditions are also necessary for establishing a connection from the PIP cell that determines the grasp aperture to an AIP column that codes the aperture width. In both of the above examples, the opposition vector is perpendicular to the axis of the cylinder, so it is the cylinder's diameter that is mapped to the aperture width. If it were the case that the opposition vector were parallel to the cylinder's axis, then the length would determine the grasp aperture.

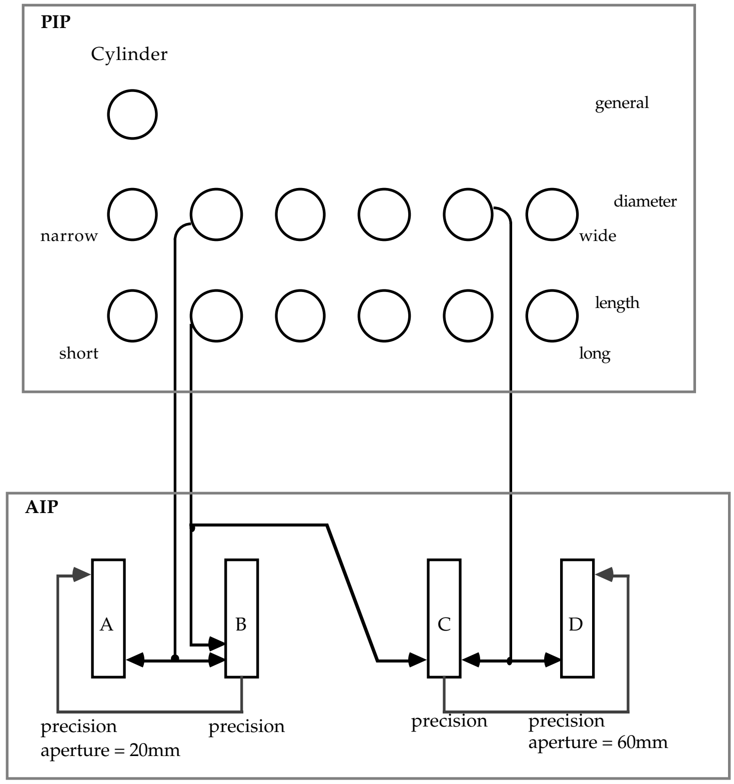

Figure 5.5: Affordance mapping from PIP to AIP. Two affordances are illustrated: narrow and wide cylinders that both map to precision grasps.

Associations from IT

The mapping from object identity in IT to AIP is somewhat simpler than the mapping from PIP, since unique objects (and hence object configurations) already give rise to non-overlapping activity patterns in IT (Figure 5.6). As a result, the identification of the object maps directly to both the grasp type (e.g. column B) and the aperture of grasp (column A). In practice, this projection is important for biasing particular grasp configurations as a function of the specific context presented by the object. However, in the case of A.T., where no information is available from the dorsal stream (PIP), the projection from IT can provide (in some cases) the necessary grasp type and parameters.

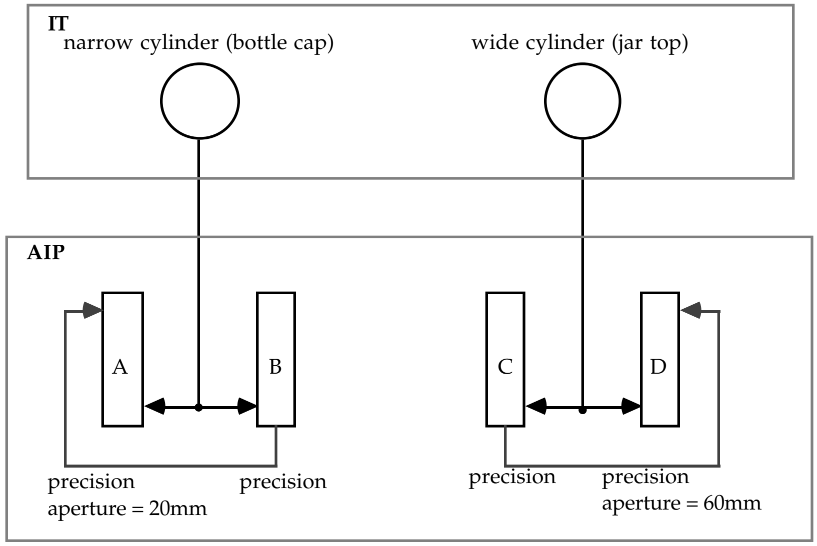

Figure 5.6: Affordance mapping from IT to AIP. The bottle cap activates a precision grasp with a narrow aperture; the jar top maps to a precision grasp with a wide aperture.

Schema Cooperation/Competition Via the Basal Ganglia

The resolution process between different affordances is handled by local interactions in AIP and loops through the Basal Ganglia (as we will see, F5 ultimately is responsible for resolution between the different grasps). Columns that are members of the same schema ensure that they are coactive by exchanging positive supporting connections (e.g., columns B and C in Figure 5.7). Note that columns B and A participate in slightly different schema instances (aperture of 20mm as opposed to 25mm, respectively). However, because they are similar enough, a small positive supporting connection is established. This connectivity between columns of similar schema instances supports the formation of a population code for the representation of the parameters (in this case, the aperture width).

Competition between different schemas is implemented through an inhibitory loop through the Basal Ganglia (BG). All AIP cells project to this region of the BG, which returns an inhibitory signal to all columns in AIP (not all connections are shown in the figure). This mechanism is similar to the maximum selector circuit of (Didday, 1976). The indirect

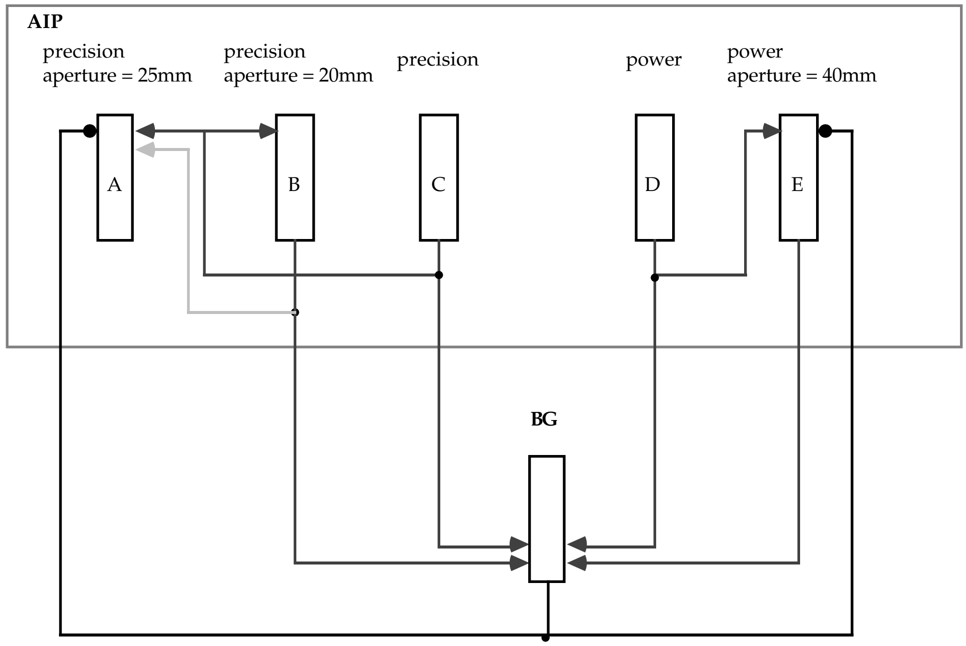

Figure 5.7: Cooperation and competition in AIP. Cooperating columns in AIP exchange positive supporting connections. Columns participating in different schemas inhibit one-another via a loop through the Basal Ganglia.

inhibitory signal due to the activation of column C upon columns A and B is counteracted by the direct excitatory signals.

F5

In this section, we detail at the column level the interactions that give rise to grasp schema selection, competition/cooperation, and sequencing within F5.

Representation of Grasp

As with AIP, each column in F5 is assigned a behavioral description. An F5 behavioral description consists of a small set of *grasp descriptions*, each of which are specified as described in Table 5.2. By allowing the behavior of a single column to be determined by multiple grasp descriptions, we are able to address the observations made by Rizzolatti that individual F5 cells can participate by varying degrees in different grasps (Rizzolatti, 1995). However, in the

grasp type: {precision, power, side, and finger prehension}. The class of grasp in which this column will participate.

aperture: either *none* or from the range [0,100mm]. Determines the preferred aperture for the column (if there is a preferred aperture).

grasp response: [0,1]. Degree of participation of this column when executing the specified grasp type.

phase response [{set, extension, flexion, hold, release}]: [0,1]. A vector that determines the relative response level (from phase to phase) of the column during execution of the specified grasp. Typically only a small subset of the vector elements are non-zero (1-2).

Table 5.2: F5 grasp descriptor. Description of F5 cell behavior in the model is done by specifying one or more grasp descriptors.

discussion that follows, the scheme of connectivity will make reference to an F5 behavioral description as if it consisted of a single grasp description.

Note that at this time, no data has been reported on F5 aperture size encoding. This aspect is necessarily included so that the model is able to preshape and grasp appropriately for objects of various sizes. The choice of the population encoding scheme (Georgopoulos, et al., 1992) has been made as a generalization of observations made in AIP and PIP (Murata, et al., 1993; Sakata, 1994). We will return to this point in the following Chapter.

Schema Priming by F6 and Cooperation/Competition Via the BG

As discussed in Chapter 5, the setup and execution of the grasping program is initiated by cells in F6. This is implemented in the model by global priming connections from specific F6 cells (which are active for the duration of the reaching/grasping/holding task) to all columns in F4 and F5 (Figure 5.8). The priming signal serves to prepare all columns in the F5 layer (and hence all grasping schemas) for possible participation in the coming motor program, and does not specify or bias particular grasps. Biases for specific grasp schemas are received from AIP and area 46, and as will be discussed below, provide excitatory supporting inputs into subsets of F5 columns.

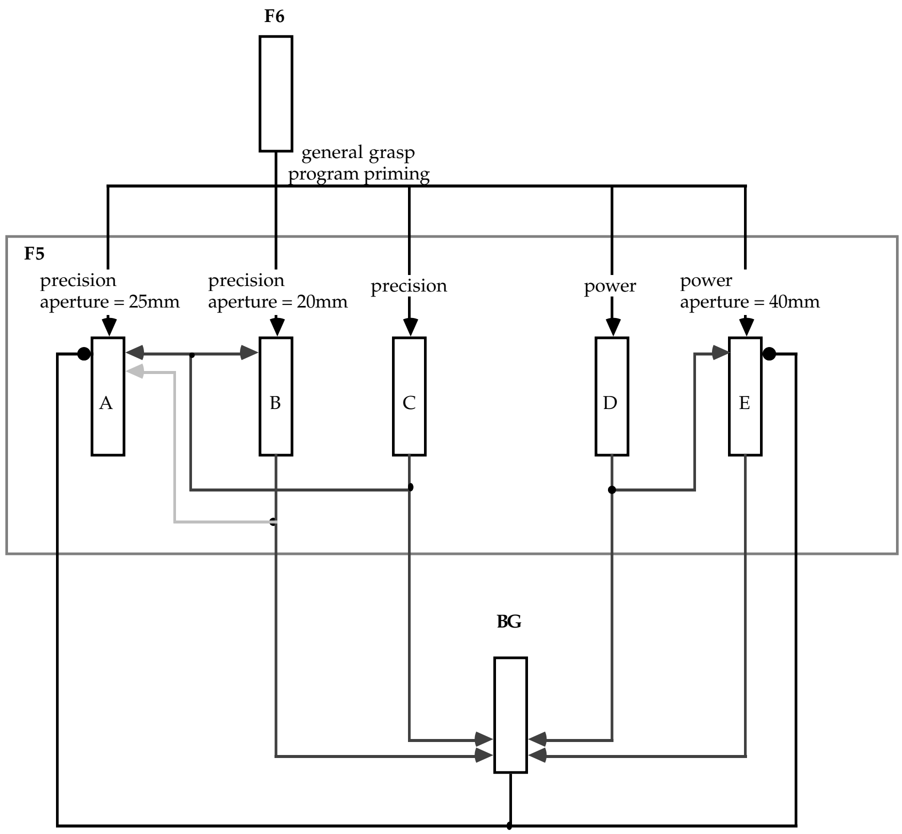

Figure 5.8: Priming of the general grasping schema, and selection of a specific grasp to execute. The selection process involves a cooperation/competition mechanism. Cooperation is implemented through local excitatory support between columns; competition is implemented via an inhibitory loop through the Basal Ganglia (BG).

Once biasing information is made available from AIP and area 46, one responsibility of F5 is to select a single grasp for execution. This is accomplished through a cooperation/competition mechanism similar to that which is found in AIP. Competition is implemented through an inhibitory loop involving BG (providing inhibitory connections to the support inputs of the F5 columns). Excitatory cross-connections between columns that participate in the same schema (also impinging on the support inputs of the columns) encourage these columns to be coactive. As in AIP, columns participating in similar schema instances (e.g., columns A and B) may also support one-another. Unlike AIP, however, excitatory supporting connections are limited to columns that participate in overlapping phases of the grasping program.

Task Specific Biasing of the Selected Grasp

The task that the system is preparing to execute after grasping the target object can (and often does) affect the selection of the grasp. In the model, high-level task information is stored in area F6. Task-related biases are imposed upon F5 via connections through the BG (Figure 5.9). This biasing is done at the level of different classes of grasp (e.g., precision pinch versus power grasp), and does not include parametric information. Connections from F6 provide positive support to columns in F5 that are involved in the *set* phase of the grasping program. In addition, connections are limited to those columns in F5 that do not specify movement parameters (e.g., aperture). Hence, the model relies on visual input to fill in the parameters. We imagine that in tasks involving movement in the dark (requiring the subject to remember the object to be grasped), area 46 would provide both grasp and parametric information.

Abstract instruction stimuli, as processed by area F2 can also be used to bias the selection of the type of grasp.

Interaction between F5 and AIP

Figure 5.10 illustrates the relationship between columns (in AIP) belonging to a single affordance, and the columns in F5 that represent the corresponding grasp schema. The columns in AIP as a group are responsible for continually providing excitatory supporting inputs to the F5 columns. This implements an active memory that ensures that once selected the entire grasp program (involving all phases of movement) will be executed.

In the Figure, the AIP columns (A-D) differ in the type of signal input that they receive. Column A does not receive any recurrent inputs from F5, and hence is considered a visual-related unit (visual inputs from PIP and IT not shown). Columns B-D receive some amount of recurrent activity from F5, and are thus considered motor-related units (or at least have a motor-related

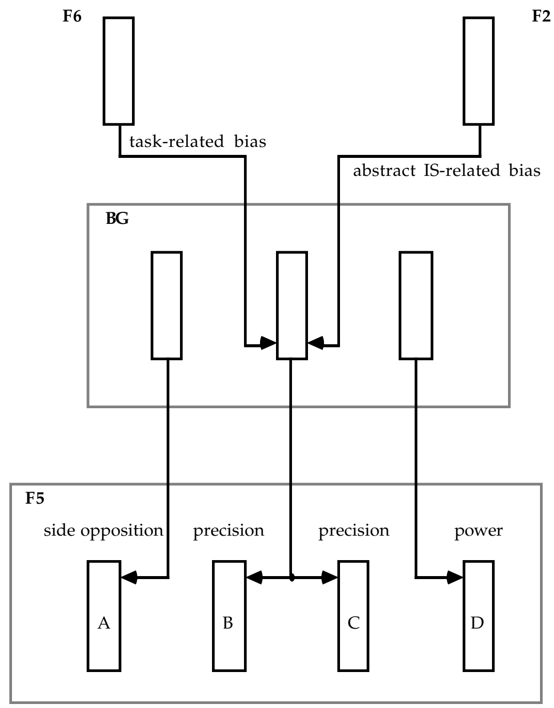

Figure 5.9: F6 and F2 biasing of the grasp schema selection process.

component). However, note that each of these three columns receive inputs from different subsets of the F5 columns. The subset is determined by the phase entry of the AIP column's behavioral description. Specifically, column B (which is a set-related unit) receives projections from the F5 set, extension, flexion, and hold units. As a result of these connections, this AIP cell will begin to fire (in response to recurrent motor signals) during the set phase of the task.

In contrast, column C (an extension-related unit) only receives recurrent activity from the F5 extension, flexion, and hold units, and thus will only become active (disregarding any visual components) once execution of the movement begins. The F5 flexion and hold units are the only phases to project to column D, which is a flexion-related unit.

Under normal circumstances, the recurrent loop from F5 back to AIP serves to update the AIP state in a feed-forward manner (i.e., AIP receives this information before sensory feedback from the descending motor command can return). However, when a perturbation occurs, this is

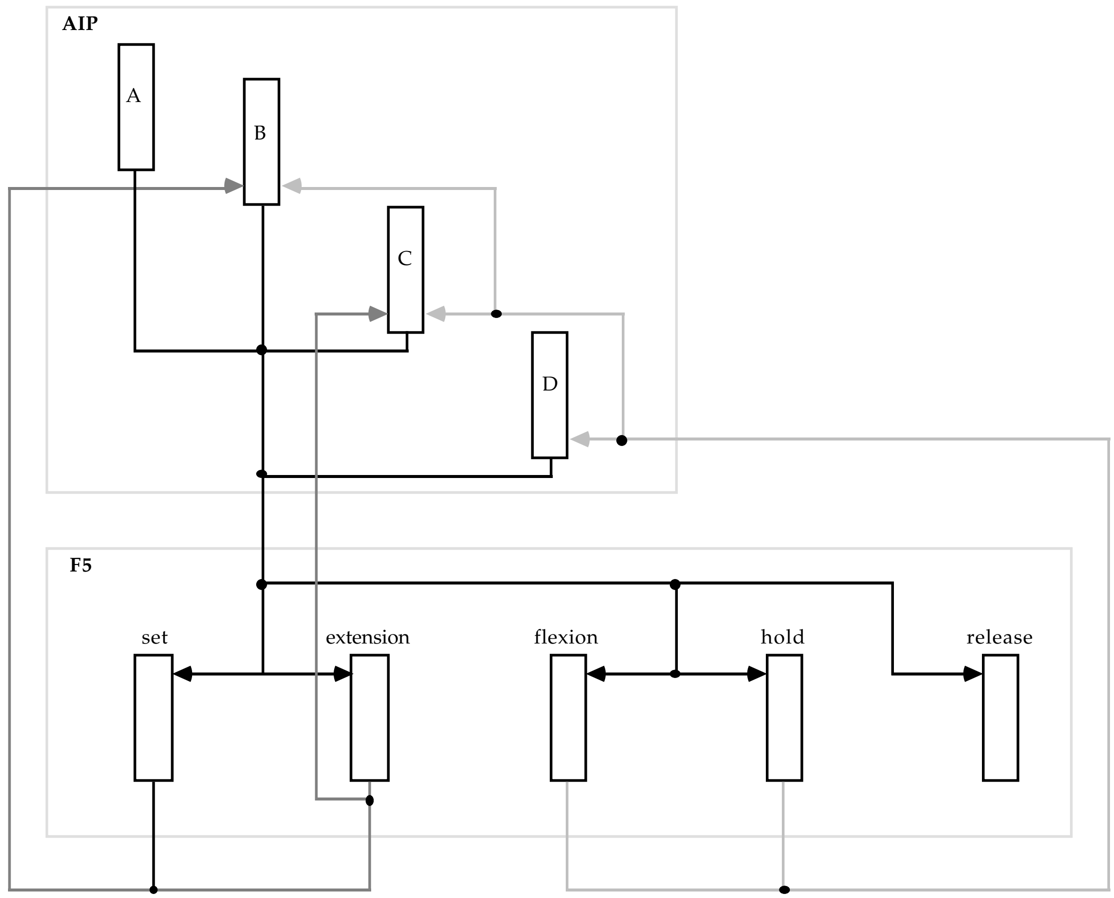

Figure 5.10: Interaction between AIP and F5 during execution of the reaching/grasping/holding task. Column A is visual-related; columns B-D are motor-related.

detected by F5 after some delay. This new state information is then passed up to AIP (and hence in this case, AIP receives feedback information).

The necessary condition that a group of AIP and F5 columns be connected in the manner illustrated in Figure 5.10 is that there must be a match in the grasp that they specify. This implies that not only must the grasp types be the same, but the parameters of the grasp must be similar, if they are specified. For example, AIP columns that do not specify a grasp aperture will only be interconnected with F5 columns that also do not specify an aperture. Likewise, AIP columns that encode an aperture of X will only be connected to F5 columns of an aperture similar

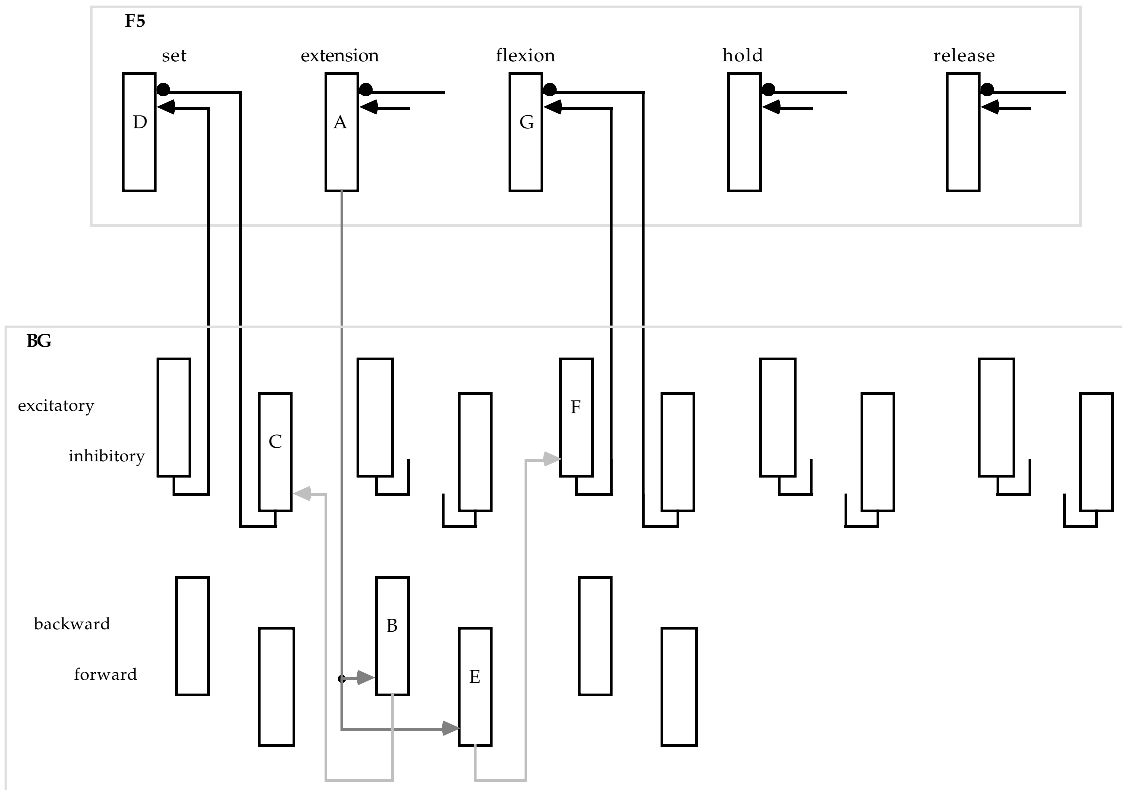

Figure 5.11: F5 interaction with BG implementing the cascade of activity in F5 as execution of the motor program progresses. F5 reports to BG its current phase; BG in turn inhibits the previous phase and prepares the next phase of movement for execution.

to X. The degree of similarity will determine the likelihood and the strength of the interconnections.

Phasing Via BG

In the previous chapter, we demonstrated the functional connectivity that is necessary to implement the cascade of activity that flows through a set of F5 units as execution of the reach/grasp/hold motor program progresses. In this section, we show the details of the implementation of this cascade.

A subregion of BG (separate from the subregions discussed earlier that are responsible for competition and grasp biasing) is used in the model to implement phase-specific connectivity in a manner that is independent of the grasp that is actually being executed (hence we are separating the component schemas from the mechanism that implements the *coordinated control program*). At each stage of the motor program, F5 reports its current phase to the Basal Ganglia. In turn, the BG positively supports all F5 columns that are involved in the next phase of movement. In addition, F5 columns that code for the previous phase receive negative support, causing them to be shunted and enforcing a phase transition.

The set of connections required to implement one step in this process are demonstrated in Figure 5.11. F5 column A is an extension unit. When it becomes active, it sends an excitatory signal to columns B and E (the backward/forward columns, respectively) in the BG. Column B projects to the *inhibitory* column (C) that corresponds to the previous (set) phase of the motor program. This column, in turn inhibits column D, the set-related unit in F5. Consequently, column D is shunted and the phase transition from set to extension is complete. In parallel with this process, the active column E projects to column F (the excitatory unit of the flexion phase), which then provides positive support to F5 column G (flexion unit). The result is that column G is now prepared for activation.

In the above description, there is no functional distinction between the backward/forward columns in BG. The situation, however, becomes more complicated when F5 columns are allowed to participate in multiple phases. Figure 5.12 illustrates one example in which an F5 column (A) is active during both the extension and flexion phases of movement. Because this column first becomes active during the extension phase, it receives positive support from the BG excitatory unit H. When the column becomes active in response to a trigger stimulus (in this case the go signal; connection not shown) it activates the BG backward column B. As in the above case, column B then activates unit C, which in turn shunts F5 column D.

The last phase in which F5 column A participates is the flexion phase. It is therefore responsible for preparing for the hold phase of the program. It accomplishes this by activating the BG forward unit E (which corresponds to the flexion phase). Unit E activates the BG

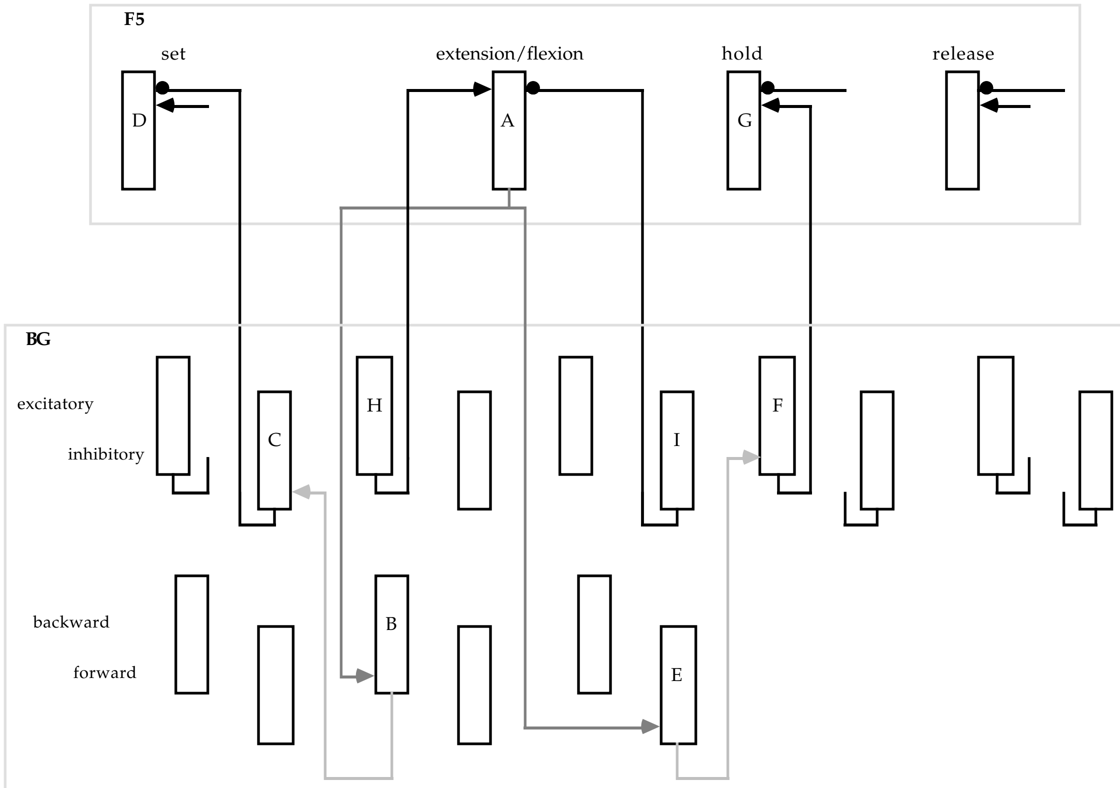

Figure 5.12: BG phasing when an F5 unit is active during both the extension and flexion phases of the reach/grasp/hold program.

excitatory unit F, which then positively supports the F5 hold column (G). Once contact with the target object is established, column G will activate, and (through connections not shown) will indirectly excite BG unit I, which then shunts column A.

It is important to note that all phase transition information is contained within the Basal Ganglia, and that this information is independent of the grasp that is being executed in F5. Specifically, the critical information is stored in the projections from the backward/forward columns to the excitatory/inhibitory columns (we refer to these as the *phasic projections*). We imagine that in the more general case, where the sequence of phases may be different depending upon the task that is being performed, that it would be possible to rearrange these phasic projections. The arrangement of these projections would be specifiable by high-level regions (possibly including F6 and area 46).

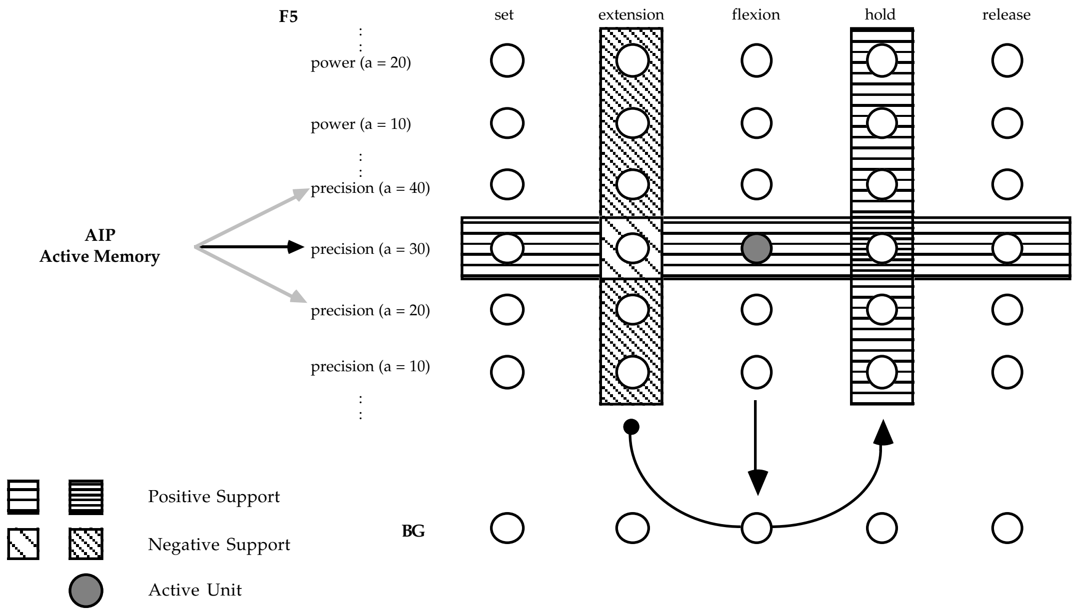

Figure 5.13: Abstract representation of the interaction between the AIP active memory and the BG phasing loops. AIP provides a detailed specification of the motor program; BG is responsible for executing the sequence of steps in the specified program.

Interaction of the AIP Active Memory and the BG Phasing Systems

In the previous sections, we have illustrated how F5 independently interacts with the AIP active memory and the BG phasing systems. In this section, we demonstrate how these two loops interact with one-another in order to execute a sequence of steps belonging to a specific grasping motor program.

Figure 5.13 illustrates at an abstract level the interaction of these two loops. The matrix of cells in F5 represent a set of grasp/hold motor programs; each program is horizontally arrayed. For clarity, we are explicitly representing many possible grasp type and aperture combinations. In addition, we are discretely representing every phase of the programs (phases are aligned vertically).

At the instant depicted in the figure (the flexion phase), the network is executing a precision grasp with an aperture of 30 mm. This state is captured by the activation of the precision/aperture = 30/flexion unit at the center of the F5 matrix. The AIP active memory maintains a record of one of the affordances that corresponds to the precision/aperture=30 grasp. This active memory continually provides positive support to all F5 units in the precision/aperture=30 grasp program (as depicted by the shaded horizontal box). In addition, due to the population coding scheme, this active memory also supports (to a smaller degree) two other precision grasp programs (aperture = 20,40).

At the same time, the Basal Ganglia circuit maintains a representation of the current motor program phase (in this case flexion). It responds by sending an inhibitory projection to all F5 cells that are involved in the extension phase of movement (left shaded column of F5 units). In addition, this BG unit provides positive support to all F5 units that code for the hold phase of movement (right-hand shaded column).

When contact with the target object is finally established, several flexion cells receive the signal input from SII indicating the contact event. Because the F5 precision/aperture=20/hold cell is receiving the highest amount of positive support (and assuming that the object is the one that is expected), the cell very quickly activates and wins the competition with other flexion cells. We then see a phase transition to the flexion phase of the precision/aperture=20 motor program.

In general, the convergence of the detailed motor program, specified by the AIP active memory, and the positive support for the next phase of movement from BG allows for the rapid transition to the next phase of movement when the triggering event is detected. However, note that if an unexpected hold triggering event is detected (for the example described by the figure), the other flexion cells (not corresponding to the current motor program) still receive some degree of positive support from the BG. As a result, not only do we see a phase transition in the F5 matrix, but we also can see a shift in the motor program (to a different row in the matrix). Note that this motor program shift is most likely to happen when the hold triggering event corresponds to a motor program (row in the matrix) that is similar to the currently executing motor program, because it is already receiving a small amount of positive support from the AIP active memory. We will return to this issue in the following chapter.

Grasp Execution

Layer F5 manages the execution of the reach/grasp/hold task through each phase of the motor program. During each phase of movement, F5 recruits 1) a set of motor assemblies in MI that are responsible for actuation of the individual joints; and 2) a set of SII units that monitor the global state of the hand. In this section, we describe in detail the representation of information in each of the key regions, and the neural circuitry that is involved in the low-level execution of the motor program.

SI and SII

The primary somatosensory cortex (SI) represents both the proprioceptive and tactile state of the hand at the level of individual joints and tactile pads, respectively. The proprioceptive information for each joint (the joint position) is represented using a population coding scheme[1] (Figure 5.14). The tactile information, coded in a separate SI channel, represents the force detected by a set of tactile pads located on the fingers and thumb. Force is coded by the firing rate of these units. For each finger pad, one unit increases its firing rate as the force is increased; a second unit reduces its firing rate with the increasing force.

Columns in SII integrate the local state information represented in SI in order to compute an estimate of the global state of the hand. Individual columns in SII receive inputs from many different proprioceptive arrays (column A), and can also integrate tactile input from selected pads (column B). Because these columns have a high threshold, they only respond to very specific configurations of joint position and finger pad state. Hence, they are referred to as

---

[1] Although a more appropriate representation would use firing-rate to encode the joint position (Kandel, et al., 1991), the population coding scheme reduces the number of connections and the computational complexity of the model implementation.

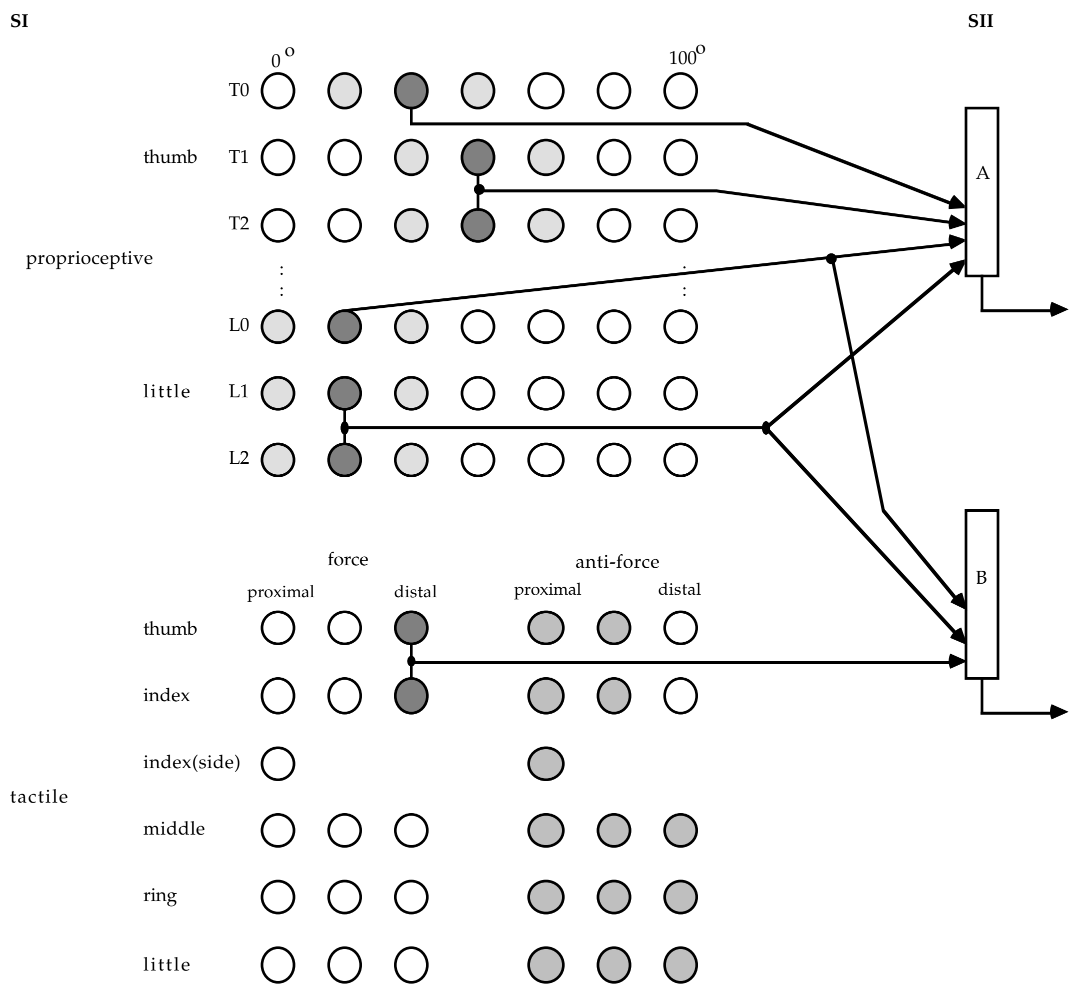

Figure 5.14: Representation of hand state information in the somatosensory cortices (SI and SII). SI carries two channels of information: proprioceptive (or joint position) and tactile. Each individual cell in SI has a small receptive field, and hence only captures a limited view of the hand state. SII columns detect hyperfeatures of SI inputs, thus representing a global view of the hand.

hyperfeature detectors. Column B, for example, combines joint position information with tactile inputs from the distal pads of the thumb and index finger. Hence the column not only is capable of detecting a contact event, but it is a contact event with an object of a certain width.

MI

A column in the primary motor cortex of the model is intended to capture the properties of a set of motor cortex neurons, projecting to both agonist and antagonist muscles that are responsible for actuating a subset of joints in the hand. In the model, we simplify this interaction by assuming that stimulation of a single MI column specifies a joint equilibrium position for the joints to which it is connected (Giszter, Mussaivaldi, & Bizzi, 1993; Shadmehr, 1993; Shadmehr, Mussiavaldi, & Bizzi, 1993). In addition, the degree of stimulation affects the joint stiffness. We use $\hat{\theta}_{ij}$ to denote the contribution that MI unit *i* has on the equilibrium position of joint *j*, and S_{ij} to denote the stiffness contribution ($0 \le S_{ij} \le 1$). When there is no physical connection from unit i to joint j, $S_{ij} = 0$. When multiple MI units are active, their contributions are summed, weighted by their activity level:

$$S_j = \sum_i a_i * S_{ij} \tag{5.12}$$

$$\hat{\theta}_j = \begin{cases} \dfrac{\sum_i a_i * \hat{\theta}_{ij} * S_{ij}}{S_j} & \text{if } S_j > \varepsilon \\ \tilde{\theta}_j & \text{otherwise} \end{cases} \tag{5.13}$$

where:

a_i is the activity level of MI column i.

$\tilde{\theta}_j$ is the default equilibrium position for joint j.

Finally, the joint actuator is modeled as a revolute spring (derived from (Katayama, 1993)):

$$\tau_j = K(S_j)(\hat{\theta}_j - \theta_j) - B(S_j)\dot{\theta}_j \tag{5.14}$$

where:

τ_j is the torque applied to joint j.

$K(\ )$ and $B(\ )$ are linear functions of S_j.

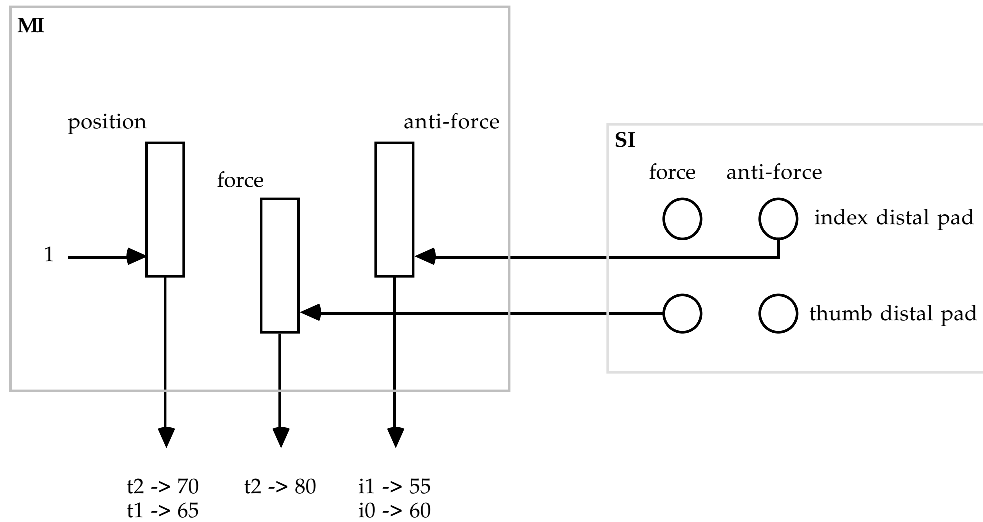

Figure 5.15: Three classes of MI columns. Force responsive columns receive projections from SI units that detect tactile stimulation; anti-force columns are sensitive to the absense of tactile input; position columns are receive a constant input.

MI Activation

At any given instant during the execution of a grasping movement, the currently active F5 columns are responsible for selecting the set of MI columns that will generate the desired hand movement. This selection is done through excitatory projections from the F5 columns to the support inputs of the MI columns. Additional details of this mapping are given below.

MI columns fall into three distinct classes, depending upon the type of signal inputs that they receive. Inspired by observations of (Picard & Smith, 1992b), one class of columns receive inputs from one or more SI tactile units. These columns are referred to as being *force responsive* (Figure 5.15), because they become active when a force is detected in SI. The complement to this set comprise the *anti-force responsive* columns, which receive inputs from the corresponding SI cells. The final class of MI columns are the *position units*. In the model, these columns receive a constant input. Thus, the only condition for their activation is that they receive positive supporting inputs from the F5 layer.

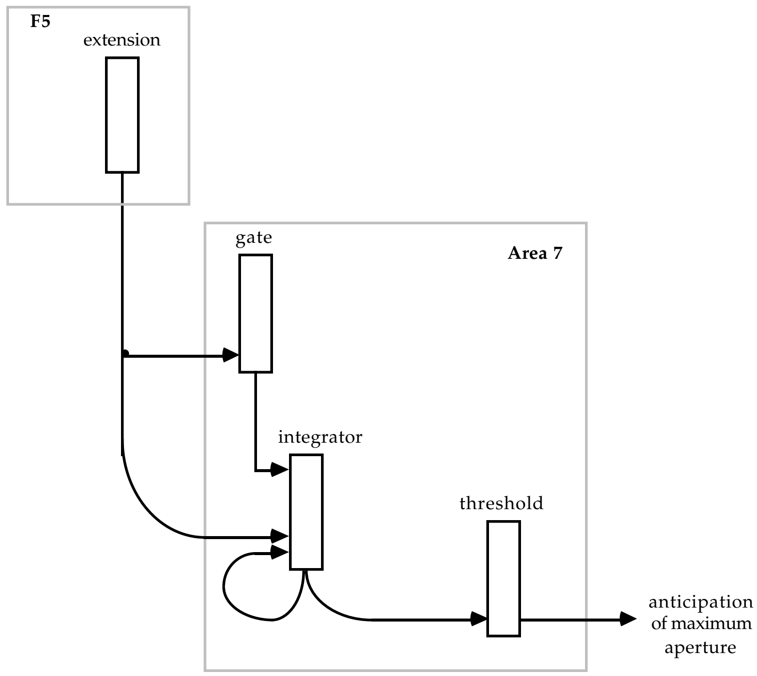

Figure 5.16: Internal model of grasp extension state.

Internal Timing Model of Current Hand Extension

As described thus far, the model relies on sensory feedback to detect the point of maximum desired aperture during the extension phase of movement so that a transition can be made to the flexion phase of movement. However, during fast movements, the delay of this sensory information relative to the point at which it must be used is significant. In addition, deafferented human subjects are still able to perform preshape and enclose movements, although the movements tend to be slower and more variable than in the afferented case (Gentilucci, Toni, Chieffi, & Pavesi, 1994).

This difficulty is approached in the model through the introduction of an internal model of hand state (Hoff & Arbib, 1993). The internal model makes use of descending motor commands (from F5) in order to estimate the point in time at which the maximum aperture is achieved. This is accomplished by tracking the amount of time that F5 spends in the extension phase of movement. A simple implementation of this timing mechanism is demonstrated in Figure 5.16.

Initiation of the extension phase of movement in F5 activates the *gate column* in the internal model (area 7). Positive support to the *integrator column* initiates the integration of the F5 input. Once this integral passes beyond a specified threshold, the internal model generates an output that can be used to trigger the transition from the extension phase of movement to the flexion phase.

Not shown in the figure is a calibration mechanism that utilizes SII inputs to appropriately set the threshold.

Full Execution Circuit

Depending upon the current phase of the motor program, columns in F5 will recruit different configurations of the MI/SI/SII circuitry. Exclusively set-related columns are not involved in the direct generation of movement, and hence do not activate lower-level regions. Extension-related columns, activated by a triggering signal (Go), instantiate a goal position for the hand joints that corresponds to the maximum aperture of the preshape (Figure 5.17). This is accomplished by sending excitatory signals to the support inputs of a selected set of MI columns in the position pool. The relative strengths of these projections are balanced in such a way that the specified joint equilibrium position is the point of maximum aperture. However, a single F5 column will generally only project to a subset of the hand joints (typically the joints of 1-3 fingers). Hence, a population of F5 columns is necessary to specify the goal state of the entire hand.

Besides driving units in MI, the F5 extension columns also positively support a subset of SII columns that will be involved in the detection of the point of maximum aperture. This detection process only examines the proprioceptive state of the hand (position). In addition, extension columns also activate the integration process in Area 7 that will be used to estimate the point at which maximum aperture will be achieved.

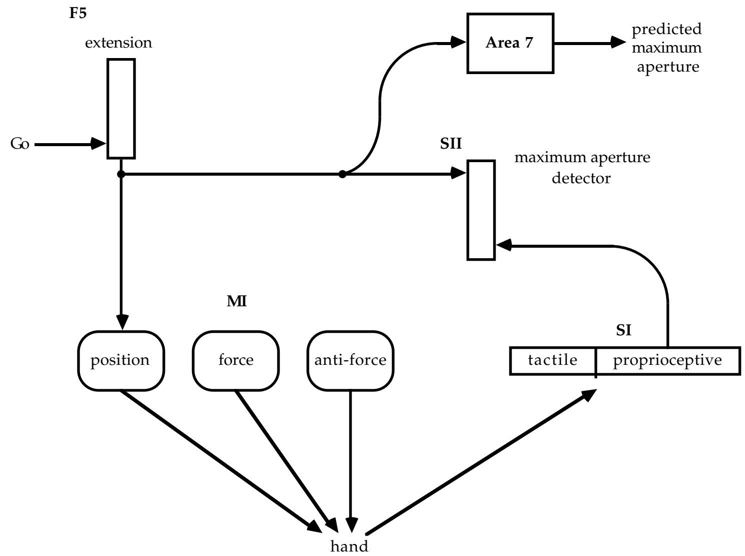

Figure 5.17: The low-level circuit involving F5 extension columns.

F5 flexion columns are activated by the detection of maximum aperture--by either SII or the internal model of area 7 (Figure 5.18). Not only does the flexion column drive the position pool of MI (corresponding to the point of anticipated contact with the target object), but also sets up SI/MI reflex loops (the MI anti-force pool) that will cause the hand to continue to close if contact is not detected at the anticipated position.

The F5 flexion units also provide positive support to SII columns that will detect the combination of hand configuration (proprioceptive state at the point of contact) and the actual contact with the target object (tactile inputs).

The F5 hold columns recruit units from all three MI pools (Figure 5.19). Those joints that are involved in applying a stabilizing force are controlled through a balance of force and anti-force pools. The remaining joints are controlled using the position pool.

Finally, the F5 release columns recruit motor units from the MI force pool to induce release of the object (Figure 20). In addition, SII columns are recruited to detect the point at which the release occurs.

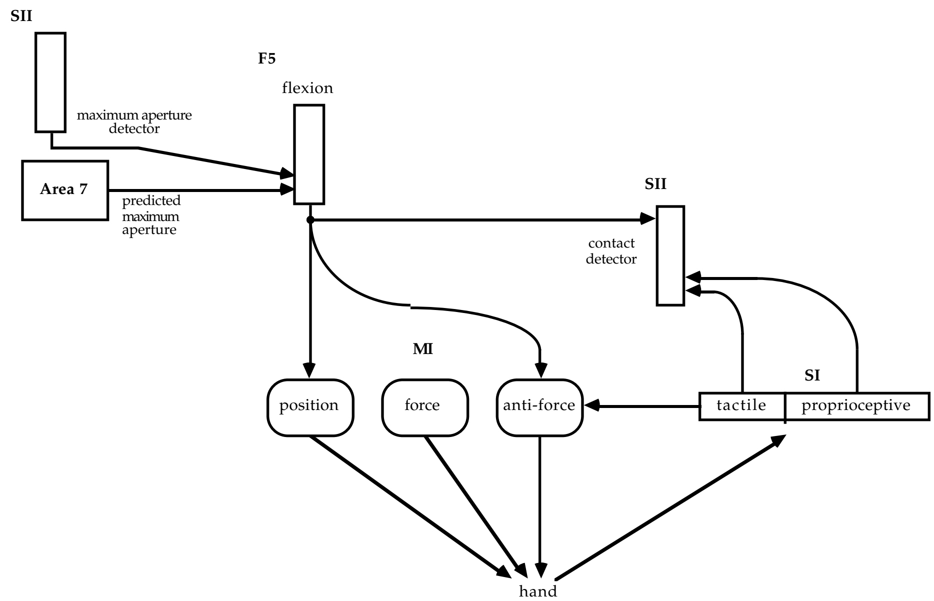

Figure 5.18: Low-level circuit involving flexion columns.

Summary

This chapter has presented a detailed view of the grasping model. First, we introduced the concept of µ-schema as a bridge from the language of schema theory to the neural implementation level. The µ-schema maintains the schema-level operators of priming, cooperation/competition, and activation, while still being implemented using a small group of neurons (referred to as a column). In order to construct a schema, a set of µ-schemas are bound together through the cooperation operator (implemented as excitatory connections between the columns). Different schemas can then interact in a either a cooperative manner (via excitatory cross-connections between the member µ-schemas), or in a competitive manner (through inhibitory cross-connections). The level of participation by an individual µ-schema within a schema, however, can be graded (as determined by its activity level). In addition, two distinct schemas may share a single µ-schema.

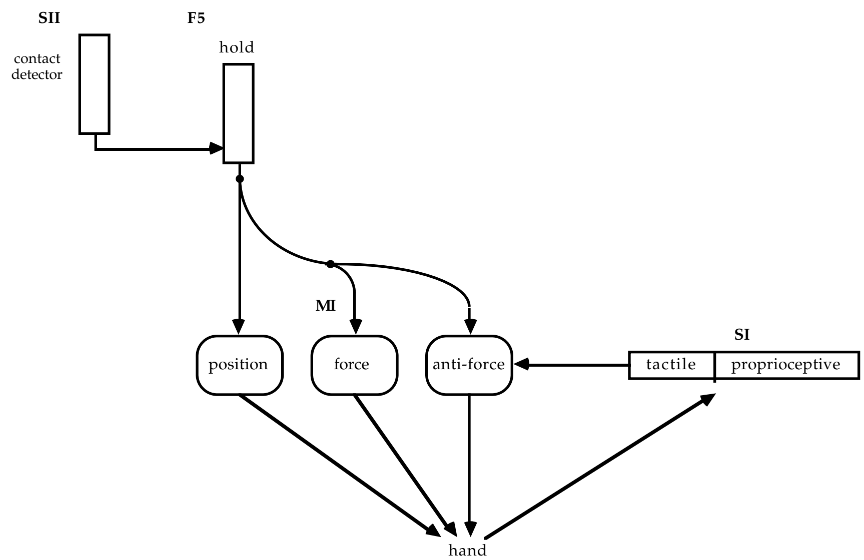

Figure 5.19: Implementation of the hold phase of the motor program.

Second, we demonstrated how information is represented in each layer in the model, and then how micro-circuits of columns involving multiple layers are constructed in order to achieve the desired computations. The primary model components are:

1. PIP represents object shape using non-overlapping pools of neurons. Within these pools, sub-populations of neurons encode the parameters of the object (making use of a population coding scheme).

2. IT represents different familiar objects using non-overlapping representations.

3. PIP and IT associate an object description with a set of affordances in AIP.

4. AIP associates an affordance with a corresponding grasp in F5.

5. During execution of the movement, AIP maintains an *active memory* of that affordance, reinforcing the choice of grasp in F5 in a manner that is independent of the phase of movement. The active memory may be updated by F5 in the face of perturbations.

6. The basal ganglia (BG) provide a mechanism for generating the sequence of phases that are observed in F5. This mechanism provides this functionality in a manner that is independent of the actual grasp that has been chosen.

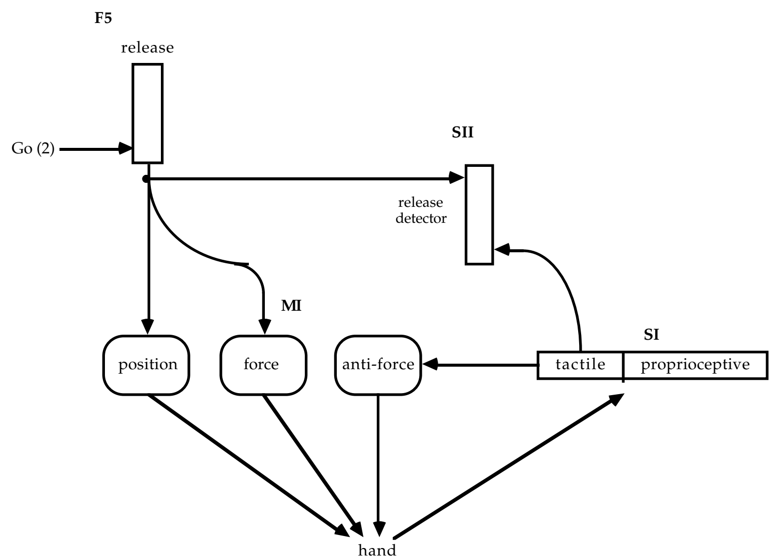

Figure 5.20: Low-level circuit for the release phase of movement.

7. The basal ganglia are also involved in implementing the schema competition mechanisms in F5 and AIP. In addition, task-related biases in the grasp decision process can be introduced by higher level systems (area 46, F6 or F2) via a pathway through the BG.

8. A single MI column specifies equilibrium positions for a subset of the joints of the hand. During specific phases of movement, F5 columns recruit a motor assembly of MI columns.

9. Units in SII provide context-dependent detection of the global hand state. This information is used by F5 to monitor the ongoing grasp so that it may react to unanticipated situations.

10. An internal model of grasp extension state is introduced as a mechanism for initiating the transition from the extension phase of movement to the flexion phase without relying on immediate sensory feedback information.

As we will see in the following Chapter, the functionality hypothesized for several regions (especially F5 and AIP) have several important implications, including:

1. The visual/motor associative process involving AIP affects how objects are encoded at the neural level in AIP.

2. The recurrent connections from F5 to AIP affect the motor-related activity in AIP.

3. The hypothesis that F5 is involved in the selection of a single grasp to execute implies certain patterns of activity when multiple grasps must be considered before a selection is made.