BYTE 3: August 1979

The Nature of Robots

PART 3: A CLOSER LOOK AT HUMAN BEHAVIOR

In This BYTE

In the third part of **The Nature of Robots**, William T Powers describes the how and whys of his particular model of human behavior. Mr Powers develops a 2-level control-loop simulation of a 3-muscle system to further the understanding of how our own control system works.

. *Page 94*

About the Author

William T Powers has been exploring the meaning of control theory for studies of human nature since 1953. He spent a number of years (to 1960) in medical physics, and then another 13 (to 1975) as Chief Systems Engineer for the Department of Astronomy at Northwestern University. His occupation has been designing electronic, optical, and mechanical systems for science.

This article appeared in BYTE magazine, volume 4, number 8, AUGUST 1979.
Copyright returned to author. Article recreated by Dag Forssell in 2004.

The Nature of Robots

PART 3: A CLOSER LOOK AT HUMAN BEHAVIOR

Figure, table, and listing numbering continued from part 2.

In part 1 of this series, I demonstrated that the concept of *behavior* is not as clear as certain people would indicate. The patterns that we call behavior result from the convergence of many influences, only a part of which can be attributed to the organism that we say is behaving. Yet the behaving organism varies its own actions so that when the influence of these actions is added to all that is unpredictable, the result is recognizable as patterns of behavior.

In part 2 we observed that a control system controls its input, not its output. It acts on its environment to make its own sensory or perceptual signal match a reference signal received from elsewhere, and to automatically counteract the effects of disturbances. It does not have to sense the cause of the disturbance: it senses the quantity it is controlling, and reacts to deviations of that quantity (or the signal representing it) from a reference level that is set by the reference signal.

The reference signal acts just as an intention ought to act. It specifies some state of affairs that is to be achieved, and serves as a target toward which action always urges the perception of the controlled variable. Under normal circumstances the control system can make its perceptual signal track a changing reference signal, and still oppose the effects of disturbances.

There are two main rules of thumb:

- The reference signal reaching a good control system controls the perceptual signal in that system.
- The actions of the control system vary so as to oppose the effects of disturbances, even if the reference signal remains constant.

Let's see how this control system model applies to one small human subsystem: a spinal *reflex* arc (reflex just means "turned back on itself"). This will lead to some concepts that will be of use to the designers of robots.

The Tendon Reflex

In the early 19th century, Sir Charles Bell established the fact that sensory nerves are separate from motor nerves, and described the "circle of nerves" found in a spinal reflex. A sensory nerve that is part of a spinal reflex arc (we will talk about one that is stimulated by the stretching of a tendon) sends its signal to the spinal cord, and the same cell that receives this signal emits a motor signal that reaches a muscle. When the muscle contracts, it has physical effects that stimulate the same sensory nerve. These are closed loops; the effects of sensory nerves that are stimulated by muscle action affect the same muscle action.

In all such loops that have been discovered, the sense of the feedback is negative. This is true of the tendon reflex. If signals from cells in the spinal cord cause a muscle to contract, the resulting stretch of the tendon stimulates sensors clustered around the tendon. The signals from these sensors reach the same cells in the spinal cord to *inhibit* their firing.

Apparently the materials are present for a control system, but before we discuss this, a digression is necessary.

All or None or Some

One of the most unfortunate accidents to occur in neurology was the discovery that signals in nerves are carried by impulses. The effect was as if the discoverers of electricity had discovered the electron before they had formulated laws of current flow, and thus developed the whole theory of electricity on the basis of collisions between one electron and another electron. As soon as there were instruments to detect nerve signals it was known that the amplitude of an impulse generated by a nerve cell was independent of the source; there was a trigger effect, so that either an impulse was generated, or it was not.

As a result, almost all neurological research has focused on single impulses. The "all-or-none" principle became so firmly entrenched that by the time digital computers arrived on the scene, most people were led off the track. "Aha," they said, "if a nervecell has a threshold that is just high enough, 2 impulses will have to reach it simultaneously to fire it: behold, an AND gate!" Since inhibition (an impulse tending to *reduce* the sensitivity of a nerve cell to an impulse arriving by a different path) can occur, we clearly have the NOT operator, and with the addition of OR (a nerve cell that can be fired by an impulse from any of several paths), we have all of the ingredients for a generalized logic circuit.

There is no longer sufficient reason to believe that the nervous system works in this way. Those who tried to analyze nerve nets as logic devices had to make a lot of assumptions, such as synchronism or clocking, that are incompatible with experimental facts. This more modern understanding was reflected in Dr Ernest Kent's recent BYTE article series, "The Brains of Men and Machines" (January 1978 BYTE, figure 2, page 16). It now seems that single impulses are not a significant unit of information for most neurons. What counts is *frequency of firing.* The sum of frequencies of excitatory and inhibitory impulses reaching a given neuron has an effect on the rate of that neuron's firing so that the output frequency is a function of a set of input frequencies. Most neurons, in other words, compute *analog,* not digital, functions. As we all know, it is perfectly possible to build digital circuitry out of analog components. Digital integrated circuits are all constructed from analog transistors.

Therefore, when I begin to identify components of a control system, as I will do in a moment, the signals will be thought of as *continuously variable frequencies,* not as on/off binary quantities. The *functions* that combine some signals will be functions of continuous variables. While any one neuron behaves as a rather nonlinear device, a collection of neurons performing essentially the same function in parallel yield an overall pleasantly linear input/output relationship, especially if we consider the normal, rather than extreme range of frequencies (zero or saturation rates of firing).

The spinal reflex systems we will now examine involve several hundred—sometimes several thousand—control systems operating in parallel, although they will be drawn as simple control systems. A perceptual signal is really the mean rate of firing in a whole bundle of pathways, all starting from sensors that are measuring the same input (eg: stretch in a tendon). The signal that enters the muscle in this system is a bundle of signals, each exciting 1 or 2 small fibers out of the thousands that make up 1 muscle. Thus, we will be dealing with neural impulses in much the way electronic engineers deal with electrons. In the majority of cases, the number of impulses passing through a cross-section of a bundle of redundant pathways per unit time will be "the signal," just as the number of electrons passing through a cross-section of a conductor per unit time is called "the current."

Level-1 Control System

Figure 13b is a schematic diagram of the tendon reflex. Figure 13a is the diagram of a general control system that I have already shown and discussed earlier. Figure 13a has an input function FNI, a perceptual signal P, a comparator C, a reference signal R, an error signal E, an output quantity O, a feedback function FNF and an input quantity I completing a closed loop. Entering this loop at the same point as the input quantity are the effects of a disturbing quantity D, affected by the disturbance function FND.

Figure 13b contains the same components in the same relationships. The input function is a sensor which emits a signal P, the frequency of which depends continuously on the amount of stretch I of the tendon at the end of the muscle. This signal P travels to the spinal cord, and the local branch enters an *inverter* which is specialized to produce *inhibitory* effects on any neuron it reaches (these actually exist in the spinal cord as Renshaw cells). This inverted copy of the perceptual signal reaches the cell body of a motor neuron C, which also receives an excitatory input from a pathway descending from centers that are higher in the nervous system (the reference signal R).

Figure 13:

(a)

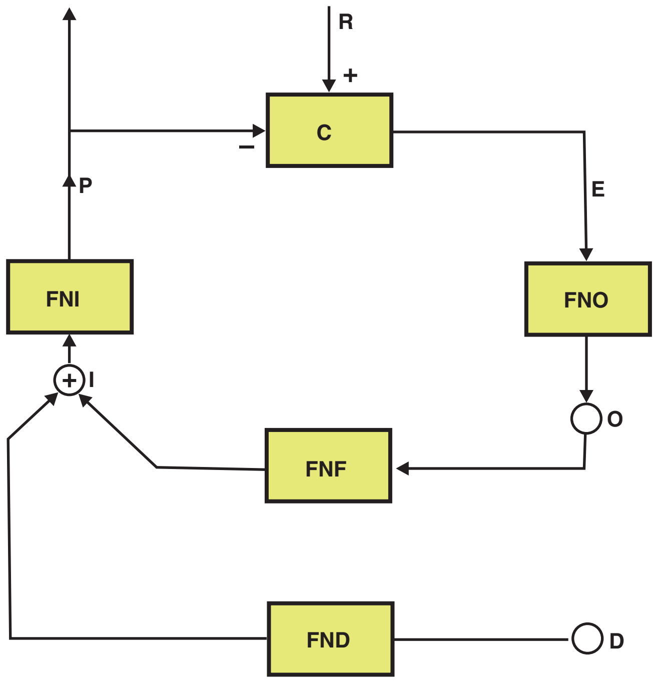

Figure 13a is the standard control-system diagram we have been using in this series.

(b)

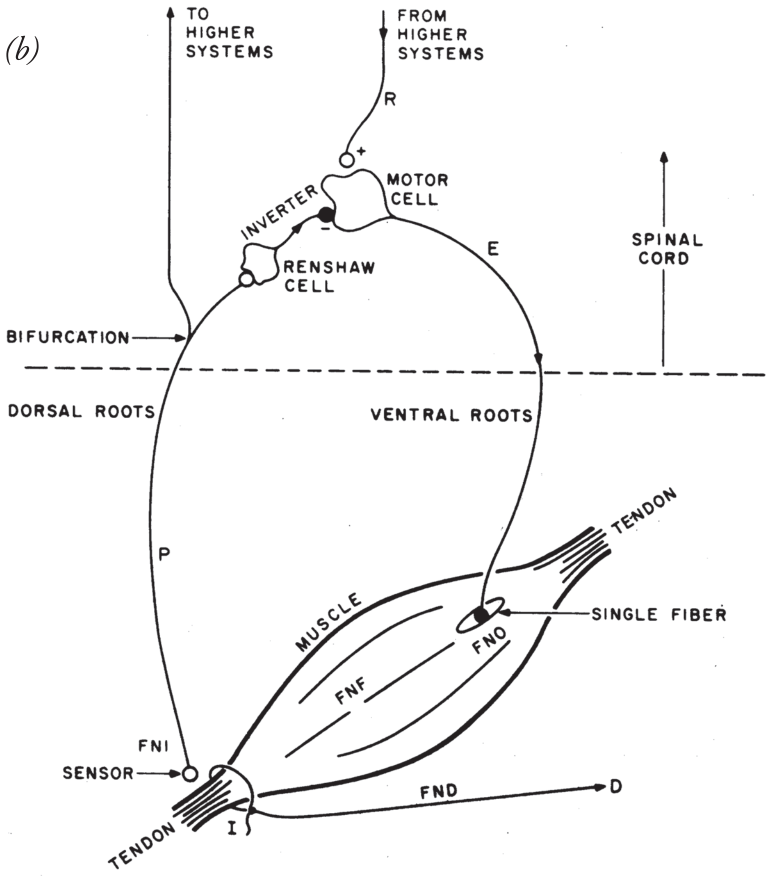

Figure 13b is a spinal reflex arc. FNI is the input function; P, the perceptual signal; C, the comparator; R, the reference signal; E, the error signal; FNO, the output function; 0, the output quantity; FNF, the feedback function; I, the input quantity; FND, the disturbance function; and D; the disturbing quantity. Roots are bundles of nerve fibers entering or leaving the spinal cord. An actual spinal reflex arc may involve several hundred systems like the one in figure 13b, with as many motor cells all operating in parallel. Thus, a signal is a bundle of signals that carry similar information.

The signal emitted by this motor neuron represents the excess of excitation over inhibition, and thus represents the difference between the reference and (inverted) perceptual signal: it is clearly the error signal E. The error signal enters the muscle, where it is converted into an average shortening of the contractile fibers in the muscle FNO. The output quantity O is the net stretch of the connective tissue that links the individual contractile fibers together. The feedback function FNF consists of the mechanical relationships that sum all these individual little forces into one force that will tend to stretch the tendon.

I have shown the disturbance as a string that pulls directly on the tendon. It is rather hard to disturb the tendon control system without dissecting the organism, a procedure that always leaves one wondering whether or not this is the original system. The reflex that is tested with a hammer just under the kneecap is a different one, a muscle-length control system. Artificially stretching the tendon will tend to relax the muscle, since the feedback is inhibitory.

In part 2 I described how control systems work. We now immediately know what this spinal reflex loop does. It maintains the perceptual signal P matching the reference signal R. Since P is a measure of tension in the tendon, we can say that this control system controls the sensed tension, and not the degree of contraction of the muscle. It also varies the amount of contraction in the fibers of the muscle to oppose any extraneous effects that tend to alter the tension in the tendon, either increasing or decreasing it.

We know that muscles are attached to bones, generally across a joint, and that when a muscle changes tension it often changes the angle at the joint that it spans. In this way movements are created and forces are applied to objects, or against gravitational and other forces. However, this little control system knows nothing of that. The only behavior it produces is sensed tension. It controls a neural signal which represents the net force being created by the muscle and any active disturbances. The control system does not know this—it has, after all, only the one kind of sensor. It knows only how much signal it is getting from the outside world, and not even what *kind* of signal this is. It is just an amount. It would need many other sensors and a very intelligent computer in order to know that this amount is measured in units of tension.

First Level of Behavioral Control

Every muscle that is used in voluntary behavior (as opposed to internal or visceral) is involved in a control system like that in figure 13b. There are no exceptions. Thus, there is no way that any higher process in the brain can *directly* produce a muscle tension. The brain can produce a muscle tension only by providing a reference signal which specifies how much tension is to be *sensed.* This does not even determine how tense the muscle will be, for if there is a steady external disturbance working, the muscle will adjust its degree of contraction to compensate for the disturbance. Pull steadily on the tendon, and the muscle will completely relax, even with the presence of a nonzero reference signal. Inject Novocain into the perceptual pathway, and the muscle may go into a violent spasm because it is trying to create a perceptual signal. The brain cannot command the muscles to contract. It can only tell level-1 control systems how much tension to sense. It is up to those control systems to do what is necessary to create the demanded signal.

Gray's Anatomy names about 200 muscles, most of which occur in pairs, and many of which consist of numerous subdivisions capable of having different effects. There are perhaps 500 to 800 muscles which can be distinguished on the basis of different directions of effect. Thus, we own 500 to 800 level-1 control systems. Every human action must be performed by adjusting the reference signals for these control systems. The behavior of these control systems need not be simulated for the simple reason that this has been done to a sufficient degree in part 2 of this series.

There are actually more level-1 control systems than muscles. For example, every muscle also contains *length* sensors, which are involved in level-1 control systems that govern not force, but something related to the stretching of the muscle itself. Length and force can be controlled quite independently under suitable circumstances; however, we won't be getting into such details here. The main point is that we chew, scratch, talk, walk, run, and swim by using level-1 control systems, and by telling them not what to do, but what to sense.

Higher Levels of Control

We have accounted for all outgoing signals from the brain that are concerned with overt actions (in the sense that all will act on level-1 control systems, although there may be, at level 1, control systems we haven't considered here). We have not, however, accounted for all incoming signals. The nervous system has hundreds of millions of sensory endings, most of which are not involved in level-1 control systems.

You'll notice that in figure 13b the perceptual signal branches. This is a real branch; all level-1 perceptual signals involved in these control systems branch, sending one branch upward. Many of the branches—enough to represent what is going on in all the muscles—continue upward to the next level of organization. The perceptual signals from level-1 input functions that are not parts of control systems do likewise. Thus, we can imagine a higher part of the nervous system that is completely surrounded, with regard to input and output, by level-1 systems and input functions.

The signals going downward from this higher part end up in control systems of the general type shown in figure 13b, controlling sensed tension and a few other simple variables. The signals going upward, the level-1 perceptual signals, all reach the next higher level of organization, which happens to be represented in the brain stem, the cerebellum, and one part of the cerebral cortex.

Imagine a second level of control systems. The input functions of this new layer will not be equipped with sensors; instead, they will receive the perceptual signals generated by level-1 input functions (or in the case of signals involved in level-1 control systems, copies of them, courtesy of the bifurcation of the dorsal roots). These signals, in subsets, are the real-time inputs to level-2 input functions, each of which generates one level-2 perceptual signal. We *define* a level-2 input function in terms of the way a single level-2 perceptual signal depends on some set of level-1 perceptual signals.

It is now clearly possible to construct a level-2 comparator, provide it with a reference signal, and make it generate a level-2 error signal. That error signal can then be wired to the input of a level-2 output function, and copies of the output of that FNO can be fanned out to serve as *reference signals for level-1 control systems.*

In fact, we can construct as many level-2 control systems as we like, until we run out of neurons that are located where the level-1 perceptual signals terminate and the level-1 reference signals originate. All out-going signals that are further inward will be accounted for; they will be level-2 reference signals. (If you can figure out why they can't be level-1 reference signals, bypassing level 2, you are beginning to understand control theory. Hint: Level-1 reference signals are adjusted by level-2 systems: what happens if an *arbitrary* signal is added to the output of a level-2 system?)

Some level-1 perceptual signals may be combined to produce level-2 perceptual signals, without involving the new perceptual signals in any level-2 control system. Perceptual signals that *are* involved in level-2 control systems branch, just as their counterparts at level 1 do: one of the branches heads further inward and upward in the brain. We can now repeat the process of going from the first to the second level of control. Clearly, a third level of control systems can be constructed, then a fourth, and so on, until we run out of brain and find ourselves looking at the inside surface of the skull.

This is my model of the brain. It will be discussed in greater detail in the next article of this series. At present we will develop a clearer understanding of the relationship between one level of control and the next higher level of control through the use of BASIC. As you will see, the relationship has some rather amazing and challenging properties.

Two-Level Control Hierarchy

We are going to model a very elementary 2-level control system. I won't attempt to model a *real* human system because it would get too complicated. The imaginary system will consist of 3 level-1 control systems, each controlling sensed force (just as in the tendon reflex system) and 3 level-2 systems, each controlling a separate *aspect* of the forces controlled by level-1 systems.

The 3 muscles will be laid out in a plane, one end of each being joined at a common central point, and the other being anchored to a point in the plane. If the angles between the muscles are equal, they will form a Y. We will assume that the common connection does not move; the muscles will apply a force there but, as in the case of flying a stick-controlled airplane, any movement will be negligible. This allows us to ignore some complex interactions between the muscles. Those interactions would not interfere with control, but would make the model very complicated. In simulating a control organization, it is always the simulation of the *environment* that creates complexities. The geometric interactions between the muscles are properties of the world in which these control systems live, not of the control systems proper.

There will be 3 level-1 control systems, 1 for each muscle. Each will sense the force being generated by its own muscle. Each will have a loop gain of 10, and a slowing factor of 0.07 (see part 2 for discussion of these properties).

There will also be 3 level-2 control systems. One will use the 3 muscles to control a force in the X direction (left and right), another will control a force in the Y direction (up and down), and the third will control the sum of the 3 forces, this sum corresponding to what physiologists call "muscle tone." We will see why there is such a thing as muscle tone (the steady mutually cancelling tension that is always there in muscles). Each level-2 control system will have a loop gain of 50, and a slowing factor of 0.01.

I hope that this arrangement looks a little amazing. Here we have 3 muscles spaced at roughly 120-degree intervals around a common point. No one muscle pulls in either the X or the Y direction. To pull in the X direction, all 3 muscles must alter their tensions. To pull in the Y direction, all 3 must alter their tensions. To vary the muscle tone all 3 must once more alter their tensions. We will be able to set reference values for these 3 variables at the same time, throw in a disturbance of arbitrary size and direction to boot, and there will be no interference among the systems that cannot be easily taken care of. Each level-2 force-controlling system will be able to keep its perceptual signal matched to any reference signal, while the others do the same thing at the same time.

It may add interest to know that the outputs from the level-2 systems to the level-1 systems will not be accurately weighted: the only choice will be whether or not a given level-2 output reaches a given level-1 comparator after multiplication by 1, 0, or –1. All 3 level-2 outputs will reach and be added together in all 3 level-1 comparators. The neat separation of X, Y, and tone control is not accomplished by carefully balancing the amount of output sent to each level-1 system. Only the crudest adjustment has to be made on the output side, essentially the choice between positive and negative feedback, with negative always being chosen.

We now come to what is perhaps the most fundamental concept of this theory of brain function. The organization which determines that an X vector, a Y vector, and a tone or scalar force will be controlled is found in the *input* functions, not in the output functions. The organization of behavior is determined by the *perceptual,* not the motor organization of the brain. By the time we finish this installment you will see exactly how that happens.

Setting Up the Model

Let us start by looking at a typical control system of unspecified level in a hierarchy of control systems. This system will receive multiple input signals from lower-level systems and multiple reference signals from higher-level systems. It will emit just 1 output signal (we will assume that the only need for an explicit output function is to provide error amplification and to smooth; otherwise the error signal could be used directly as the output signal). Figure 14 shows this typical system.

Perceptual Inputs from Lower Levels

The input function will now be a little too complicated to be represented as a BASIC function since we need a set of weighting factors so that each input can be assigned a weight before summing all of the inputs together. The easiest way to deal with weighting factors for a generalized system is to use a matrix that contains all of the factors for all of the levels. For the input function we designate the matrix as S (for sensory) and write it as:

S(L,J,K),
where: L = level
J = system at that level
K = weight of Kth signal from level L –1.

The perceptual signal for this Jth system at the Lth level will be designated P(L,J). The perceptual signal can thus be written as the sum of contributions (weighted) from some set of lower-level systems, a weighting of 0 in the S matrix meaning absence of a connection:

$$P(L,J) = \sum_{K=0}^{N(L-1)-1} S(L,J,K) \times P(L-1,K)$$

where N(L–1) is the number of systems in the next lower level.

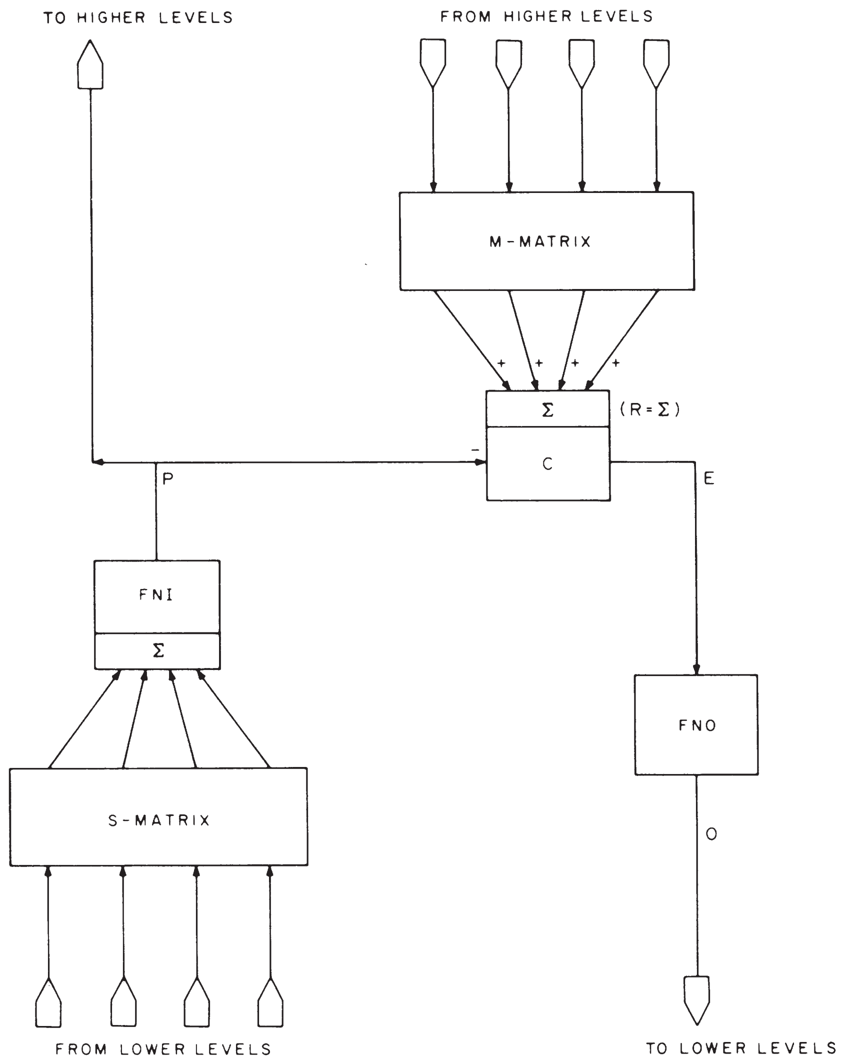

Figure 14: A typical control system in the middle of a hierarchy of control systems. This system receives multiple reference signals, given a positive or a negative sign by an appropriate entry in the M matrix (no other weighting). The sum of these reference signals is the effective reference signal. The system also receives multiple input signals which are copies of perceptual signals in lower-order systems. These signals are given quantitative weightings by the S matrix and summed in the input function FNI of the system to create this system's perceptual signal P. A duplicate of the perceptual signal travels upward to higher-level systems.

The perceptual signal is subtracted from the effective reference signal (or vice versa), and the remainder is emitted by the comparator C as the error signal. The error signal is amplified and smoothed by the output function FNO with the result being emitted to lower-level systems as the output signal O.

Reference Inputs from Higher Levels

A similar operation is performed to calculate the net reference signal R(L,J). A matrix M(L,J,K) is used to select a connection factor (1, 0, or −1) for each output of a higher-level system; the net reference signal is the sum of all the outputs of the higher-level systems, each multiplied by its appropriate factor. A 0, of course, means no connection.

The M matrix is filled by looking at the sign of the corresponding entry in the S matrix for the *next higher level.*

To understand how this correspondence is figured, think of the second index in the matrix as the *destination* of the signal, and the third index as the *source.*

Suppose that we wanted to fill in the M matrix for 1 level of systems. An entry will be −1 if the corresponding S matrix entry of the next higher level is negative, 0 if the S matrix entry is 0, and 1 if the S matrix entry is positive. But *which* is the entry in the S matrix for level L+1 corresponding to M(L,J,K)?

The answer is simple: M(L,J,K) corresponds to S(L+1,K,J). The source and destination indices are simply interchanged. If a higher-level system gives a negative weight (of any amount) to the perceptual signal from a given lower-level system, it sends a copy of its output to the comparator of *the same lower-level system* with a negative (inhibitory) sign. A negative connection factor means that the output of this higher-level system will subtract from the contributions of other higher-level systems to the lower-level *net reference signal.*

Thus, once the S matrix for the next higher level has been filled in, we can calculate the entries in the M matrix:

$M(L,J,K) = SGN(S(L+1,K,J))$

where SGN is the *Sign* function that generates the appropriate 1, 0, or –1.

You may choose to skip these procedures and simply spell out each connection one at a time. My thought in using a general solution is not merely to save lines of program, but to point the way toward expanding the simulation both horizontally (adding more systems at each level) and vertically (adding more levels).

The reference signal for level L, system J, is found by summing over the outputs of all systems of level L+1, multiplying the output from each higher-level system by the appropriate connection factor from the M matrix:

$$R(L,J) = \sum_{K=0}^{N(L+1)-1} M(L,J,K) \times O(L+1,K)$$

To complete this general model we need only calculate the error signal E and the output signal O. The required slowing factor and the error sensitivity are put in the output function.

$E(L,J) = R(L,J) - P(L,J)$

$$O(L,J) = O(L,J) + K(L) \times (G(L) \times E(L,J) - O(L,J))$$

where K(L) is the slowing factor for all systems of level L (see part 2), and G(L) is the error sensitivity for all systems of level L.

Top and Bottom of the Model

We do not have a complete control system at the top of this hierarchy where we will be injecting reference signals for the highest complete level. Therefore we designate those signals as (in this case) O(3,I), output signals from 3 imaginary level-3 systems (us) indexed by I = 0 (X force), 1 (Y force), or 2 (tone). The M matrix for level 2 is set up so that M(2,I,I) is 1, I running from 0 to 2; this establishes connections from each level-3 output to 1 corresponding level-2 reference input. All other entries are left at 0 (my North Star BASIC zeros arrays when they are first dimensioned).

At the bottom, the output signals O(1,I) are supposed to create muscle tensions that affect 3 input quantities; the amount of stretch in the tendon attached to each muscle. To avoid treating a special case, we will designate these input quantities as "level 0 perceptual signals," P(0,I). The value of each input quantity is found by adding the magnitude of the corresponding output to the component of a disturbance that acts along the length of the associated muscle. The value of the input quantity P(0,I) represents the net stretch in a tendon created by the muscle contraction and this component of the disturbance as they act together.

The level-1 S matrix simply connects each input quantity, multiplied by 1, to its respective input function. Thus, we set S(0,I,I) = 1, for I = 0, 1, and 2. All other entries in this matrix are 0.

The geometry of the muscles is adjustable. Since setting up this geometry is the opening phase of the BASIC program, we will take a quick run through this program and discuss the muscle setup. See figure 15 to help visualize how everything works. Figure 16 is the same system, more closely representing the organization of the brain.

Figure 15 appears on the next page, page 10

Figure 15: *The 2-level hierarchy simulated in this article. Three level-1 systems each control the amount of tension in 1 muscle, as represented by the 3 level-1 perceptual signals. Copies of these 3 perceptual signals reach all 3 level-2 systems, where they are weighted and summed so as to represent the X component of muscle force (P(2,0)), the Y component of muscle force (P(2,1)), and total muscle force or muscle tone (P(2,2)).*

Each second level system sends an amplified and smoothed version of its error signal as an output signal to all 3 lower-level systems. Each output signal splits into 3 identical branches, 1 for each level-1 system. When a branch reaches a level-1 comparator, it may be connected directly or through an inverter before being summed with other reference inputs. There is no other weighting of output signals. If necessary, an inverter is used to preserve negative feedback for a particular path.

Each level-1 system amplifies and smooths its error signal to make an output signal reaching just 1 muscle. A higher-level system determines the reference signals for X, Y, and total force. These are specified by the operator of the simulator. All systems correct their own errors simultaneously.

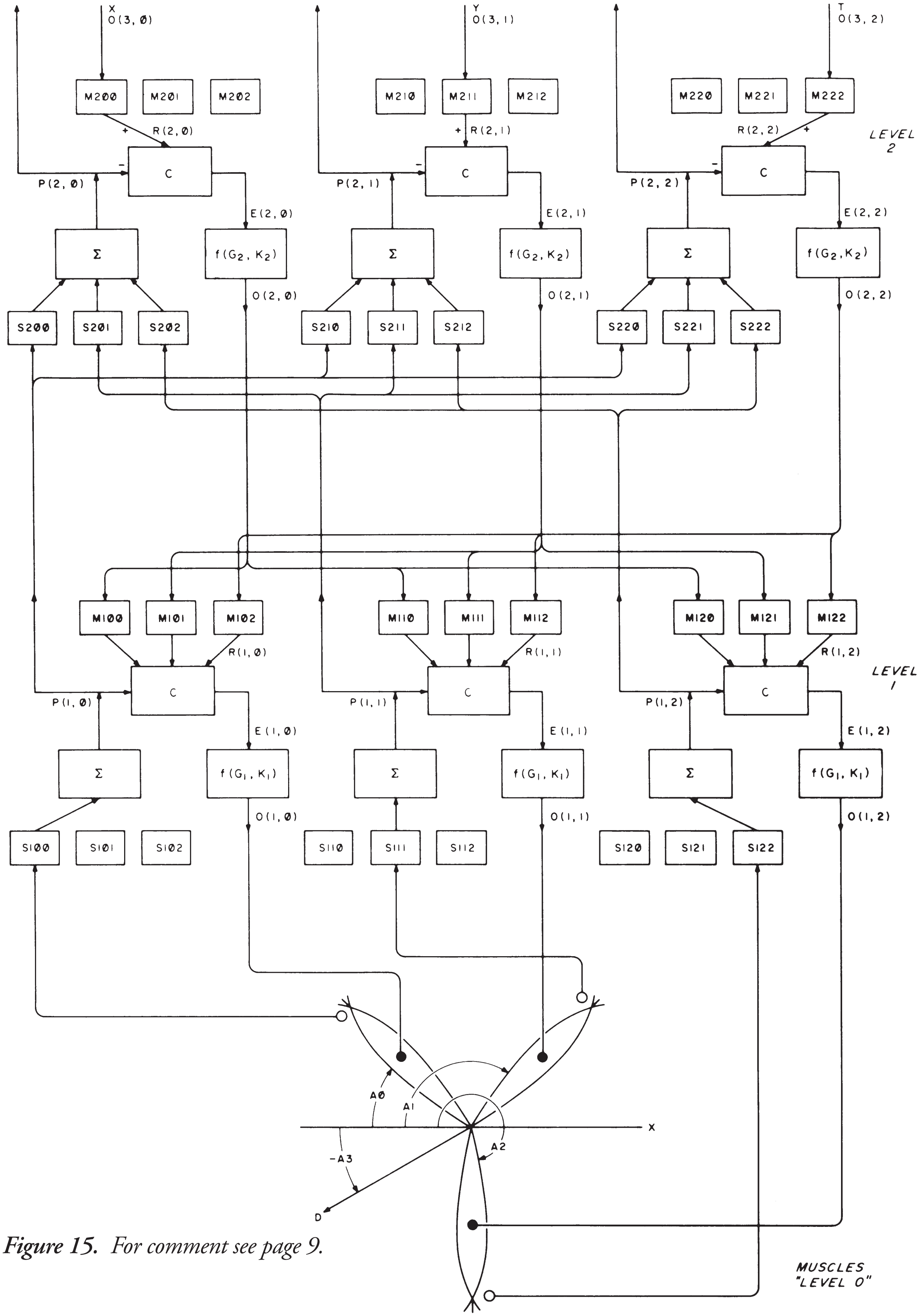

Figure 15. *For comment see page 9.*

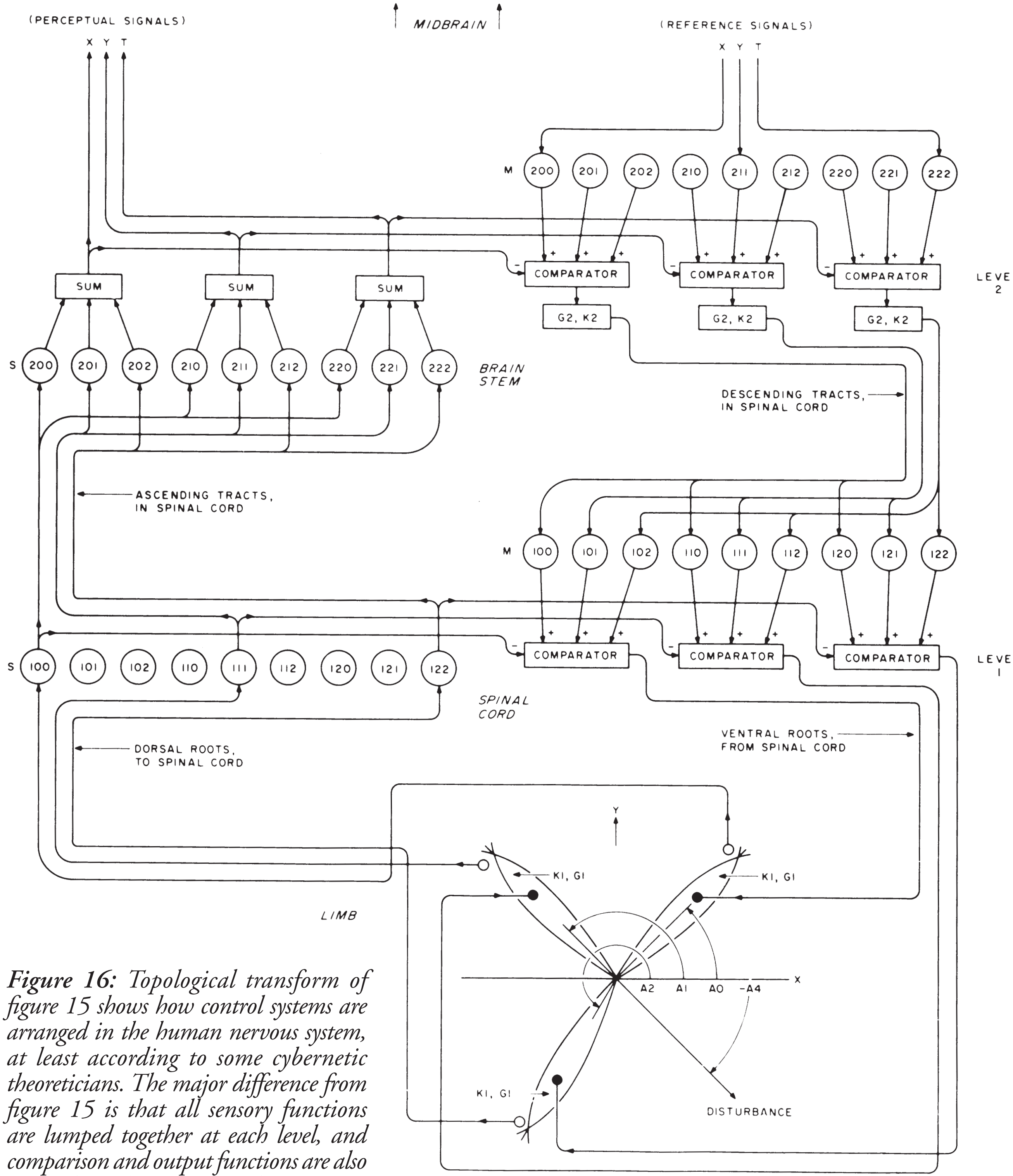

Figure 16: *Topological transform of figure 15 shows how control systems are arranged in the human nervous system, at least according to some cybernetic theoreticians. The major difference from figure 15 is that all sensory functions are lumped together at each level, and comparison and output functions are also lumped together. The S and M matrices are represented in a nervous system as synaptic connections, the weighting of which is determined by the number of branches (from one to hundreds) that form just as a nerve fiber reaches the next cell body.*

The sign of a weighting is determined by whether or not a Renshaw cell (specialized to produce inhibition) is interposed. A collection of comparators and output functions is called a motor nucleus. For level 2 and higher, the branches of perceptual signals that cross over and enter a motor nucleus are called collaterals.

Listing 3: *For comment, see page 13*

```
1  DIM P(2,2),R(2,2),E(2,2),O(3,2),S(3,2,2),M(2,2,2),A(3),K(2)
2  DIM G(2)
3  G(1)=10\ K(1)=.07\ G(2)=50\ K(2)=.01
4  P=3.1415927/180
5  GOSUB 99\ REM         (SET UP MUSCLE GEOMETRY)
6  REM              ******************************************
7  REM              SET UP SENSORY WEIGHTINGS
8  REM              ******************************************
9  FOR I=0 TO 2
10  S(1,I,I)=1
11  S(2,0,I)= COS(A(I))
12  S(2,1,I)= SIN(A(I))
13  S(2,2,I)=1
14  S(3,I,I)=1
15  NEXT I
16  REM             ***************************************
17  REM             SET UP MOTOR WEIGHTINGS
18  REM             ***************************************
19  FOR L=1,TO 2
20  FOR I=0 TO 2
21  FOR J=0 TO 2
22  M(L,I,J)=SGN(S(L+1,J,I))
23  NEXT J\ NEXT I\ NEXT L
24  GOSUB 109\ REM      (SET UP REFERENCE SIGNALS)
25  GOSUB 116\ REM      (SET UP DISTURBANCE)
26  REM             ******************************************
27  REM             CALCULATE SYSTEM BEHAVIOR
28  REM             ******************************************
29  !\FOR Q=1 TO 5
30  FOR J3=0 TO 1
31  L=2\ GOSUB 50\ REM CALCULATE SYSTEMS AT LEVEL L
32  FOR J2=0 TO 1
33  L=1\ GOSUB 50
34  FOR I=0 TO 2
35  P(0,I)=0(1,I)+D*COS(A(I)–A(3))
36  NEXT I\ NEXT J2\ NEXT J3
37  GOSUB 69\ REM        (PRINT TABLE OF VALUES)
38  NEXT Q
39  !"(A)NGLE? (R)EFS? (D)IST? (C)ONT? (P)RINT MATRICES? "
40  INPUT "",A$
41  IF A$<>"A" THEN 42\ GOSUB 102\ GOTO 29
42  IF A$<>"R" THEN 43\ GOSUB 109\ GOTO 29
43  IF A$<>"D" THEN 44\ GOSUB 116\ GOTO 29
44  IF A$<>"C" THEN 45\ GOTO 29
45  IF A$<>"P" THEN 46\ GOTO 76
46  !" ???? "\ !\ GOTO 39
47  REM             **************************************************
48  REM             CALCULATIONS FOR LEVEL L SYSTEMS
49  REM             **************************************************
50  FOR J=0 TO 2
51  V=0
52  FOR K=0 TO 2
53  V=V+P(L–1,,K)*S(L,J,K)
54  NEXT K
55  IF L=1 AND V<O THEN V=0
56  P(L,J)=V\ V=0
57  FOR K=0 TO 2
58  V=V+O(L+1,K)*M(L,J,K)
59  NEXT K
60  R(L,J)=V\ V=O (L,J )
```

```
61  E(L,J)=R(L,J)–P(L,J)
62  V=V+K(L)*(G(L)*E(L,J)–V)
63  IF L=1 AND V<0 THEN O(L,J)=0 ELSE O(L,J)=V
64  NEXT J
65  RETURN
66  REM             *************************************
67  REM             DATA LISTING SUBROUTINE
68  REM             *************************************
69  !\! "ITERATION # ",%2I,Q,"       ----------------------------"
70  FOR J=2 TO 1 STEP –1,
71  !\! "LEVEL ",%2I,J,%#7F2
72  FOR I=0 TO 2\!"          ",R(J,I),"             ",\ NEXT I
73  !\FOR I=0 TO 2\!" ",P(J,I),"       ",O(J,I)," ",\ NEXT I
74  !\ NEXT J
75  !\ RETURN
76  !\!"SENSORY MATRIX"\ !
77  FOR L=1 TO 2
78  !"LEVEL ",%1 I,L
79  FOP J=0 TO 2
80 !"   ",
81  FOR K=0 TO 2
82  !%6F2,S (L,J,K),
83  NEXT K
84  NEXT J
85  !
86  NEXT L
87  !\! "MOTOR MATRIX"\!
88  FOR L=1 TO 2
89  !"LEVEL ",%1I,L
90  FOR J=0 TO 2
91  !"   ",
92 FOR K=0 TO 2
93  ! %6F2,M(L,J,K),
94  NEXT K
95  NEXT J
96  !
97  NEXT L
98  !\ GOTO 39
99  REM             *************************************
100  REM            SET UP MUSCLE GEOMETRY
101  REM            *************************************
102  !\! "MUSCLE ANGLES:"
103  INPUT1 "#1 \ ",A(0)\ INPUT1 " #2\ ",A(1)\ INPUT1 " #3\ ",A (2)
104  A(0)=A(0)*P\ A(1)=A(1)*P\ A(2)=A(2)*P
105  RETURN
106  REM            ****************************************
107  REM            SET UP REFERENCE SIGNALS
108  REM            ****************************************
109  !\!"REFERENCE SIGNALS:"
110  INPUT1 "X: ",O(3,0)\ INPUT1 " Y: ",O(3,1)
111  INPUT1 " TONE: ",O(3,2)
112  RETURN
113  REM            ******************************************
114  REM            SET UP DISTURBANCE & ANGLE
115  REM            ******************************************
116  !\! "DISTURBANCE:"
117  INPUT1 "MAGNITUDE: ",D\ INPUT1 " ANGLE: ",A(3)
118  A(3)=A(3)*P
119  RETURN
READY
```

Listing 3: *(previous page) North Star BASIC simulation of a 3-muscle system. The muscles have 3 operations they are to perform: movement in the X direction, movement in the Y direction, and tone control. A sample run of the simulator is shown in listing 4. The exclamation point is used as an abbreviation for the PRINT statement.*

Listing 4: *(this page) A sample session with the simulator in listing 3. When the simulator is initialized, the user is allowed to set up several values: the 3 muscle angles, the reference signals, and the disturbance magnitude and angle. For each iteration the values for level 1 and level 2 are output in the following form. First the reference signal for the particular muscle is printed. The perceptual signal is printed on the next line, just to the left of the reference signal, and the output signal is printed to the right. This is repeated for every muscle.*

```
RUN
MUSCLE ANGLES:
#1\  30  #2\  150  #3\  270 REFERENCE SIGNALS:
X: –30 Y: 40 TONE: 175 DISTURBANCE:
MAGNITUDE: 0 ANGLE: 0


   PERCEPTUAL      REFERENCE        OUTPUT
     SIGNAL          SIGNAL         SIGNAL

ITERATION # 1 ------------------------------
LEVEL 2
     -30.00          40.00          175.00
 -18.19  –20.76  38.50   20.55  187.25  80.50

LEVEL 1
      80.29          121.81          39.19
  74.52   73.35  109.52 110.46  37.83   36.14

ITERATION # 2  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -32.12  –19.13  45.65   10.29  163.72  61.33

LEVEL 1
      52.49          90.75           31.91
  47.36   47.64  82.67   82.54  27.25   28.61

ITERATION # 3  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -29.56  –18.68  37.28   12.56  177.48  67.63

LEVEL 1
      61.51          98.87           36.40
  55.96   55.93  89.92   89.89  33.67   33.22
```

```
ITERATION # 4  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -29.54  –18.83  40.19   12.57  172.81  65.13

LEVEL 1
      58.87          96.52           33.73
  53.51   53.52  87.72   87.74  30.57   30.64


DISTURBANCE:
MAGNITUDE: 40   ANGLE: 135


ITERATION # 1  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -72.05    2.40  82.15 –8.75    173.67  65.75

LEVEL 1
      59.40          54.60           76.90
  52.56   63.30  63.87   16.98  57.11   93.27

ITERATION # 2  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -15.37  -16.51  51.01   50.17  130.58  65.55

LEVEL 1
      66.48          98.91           56.14
  59.89   69.94  90.08   54.05  55.95   50.56

ITERATION # 3  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -31.36  -17.15  49.55   10.51  167.41  64.63

LEVEL 1
      58.05          95.26           37.01
  55.07   65.55  87.97   48.88  59.19   58.95

ITERATION # 4  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -29.97  -16.56  37.45   11.18  175.04  66.18

LEVEL 1
      61.10          93.65           38.75
  54.44   64.95  38.54   49.97  35.97   61.01

ITERATION # 5  ------------------------------

LEVEL 2
     -30.00          40.00          175.00
 -29.55  -13.39  39.87   11.75  173.88  64.95

LEVEL 1
      60.31          93.10           36.81
  53.94   64.55  88.17   49.55  30.93   59.18
```

The Simulator

Muscle angles. After the dimension statements and the statements that set slowing factors and error sensitivities for each level have been called, the program calls a subroutine that asks for the angle at which each of the 3 muscles is to be set (in degrees). You can use 30, 150, and 270 degrees (for equal spacing). There is nothing to prevent the choice of any angles you like, although you should draw a diagram to determine the effect on the system. It is hard to create a force in a direction in which there is no component of force from any muscle.

Sensory weightings. Lines 9 to 15 organize the perceptions of this system, and thus organize its behavior. For values of I from 0 to 2, all 3 levels of sensory matrix are set up. You can now see how X and Y forces are sensed. The weights for level 2, system 0, correspond to the *cosine* of the angle between the positive X axis and the angle of each muscle. Those for level 2, system 1, correspond to the sine of the same angles. Each input function is weighting the perceptual signals from the muscles according to the *component* of force that is aligned with the direction being sensed. The tone system, level 2, system 2 adds the signals together to yield a total-force signal.

Motor weightings. Lines 19 to 23 use the already entered values of the S matrices to create the connection matrix M. The sign function selects the sign that will preserve negative feedback.

Highest-level reference signals. In line 24, the program calls a subroutine that asks for 3 reference signals: one designating the amount of X force, another designating the amount of Y force, and a third designating the sum of forces, or muscle tone. Positive or negative numbers are allowed. A real nervous system cannot handle negative frequencies, but the same effect can be created by suitable use of inverters so that one (positive) frequency means a positive quantity and another (also positive) frequency means a negative quantity. In reality there would be 6 systems of level 2 in this 4-quadrant system.

I have set up level 1 to behave realistically like a muscle control system; neither negative signals nor negative forces can be produced.

Disturbance. At line 25, the program calls a subroutine which asks for the amount and direction of a constant disturbance. A disturbance might be created by seizing the place where the 3 muscles join, moving it, and holding it in the new position. Despite the fact that the control systems are neither detecting nor controlling position, arbitrary movement of this junction in space will stretch or relax the muscles, creating changes of force due to the spring constants of the muscles. Therefore it is reasonable to suppose that a force disturbance can be created, one which projects into the direction of each muscle according to the cosine of the angle between the disturbance vector and the axis of the muscle.

Calculating the behavior. Lines 29 through 37 call a subroutine that actually does the calculation of signals in all 6 control systems. You will notice 3 nested FOR-NEXT loops. The outer 2 loops cause the lower-level system to iterate twice for every iteration of the higher-level system. This proves to be an exceedingly useful, easy way to stabilize the 2-level system. (I have also tried this with a 3-level system, and it worked just as well.) I have no formal rationale for why this works; informally, it seems to be a good idea to let the lower-level system correct most of its error before the higher-level systems take their own errors seriously.

The inner loop, line 35, simply calculates the values of the input quantities for the level-1 systems, using the angles of the muscles and of the disturbance. This is, in effect, the simulation of the environment (the muscles are in the environment of a neural control system).

At line 37 a routine is called which prints out the signals for all systems: the reference signal on 1 line, the perceptual signal to the lower left of it, and the output signal to the lower right for each system. Line 38 closes the iteration loop; 5 iterations are called for.

Lines 39 through 46 ask what action is to be taken after 5 iterations.

Calculation subroutine. Lines 50 to 65 calculate the signals for each system. The V that occurs here and there is simply a way to reduce the number of times a subscript has to be calculated. The perceptual signal is calculated first, then the reference signal, the error signal, and the output signal, for each system of level L. The level is set at lines 31 and 33 by the calling program. Line 62 contains the slowing routine which appeared in part 2. Lines 55 and 63 determine whether or not level 1 is being calculated; if it is, the perceptual and output signals are prevented from going negative.

Data listing subroutine. This subroutine is called after every complete iteration of both levels. It prints only the perceptual signal, reference signal, and output signal from the 3 systems at each level.

Running the Program

After the RUN command is given, the program asks for all adjustable parameters and then does 5 iterations, printing out the values of all signals each time. It then issues a prompting message, the answer to which determines what happens next. The C command means do 5 more iterations. The P command causes the sensory and motor matrices to be printed out. To get an idea of the time scale on which human level-1 and level-2 systems work, imagine that each iteration takes about 1/20 of a second. (If you are looking for mental exercise, you might adapt the plotter from part 2 to show the variables in this simulation.)

What the Simulator Shows

There has always been a problem in conventional models of the brain that have to do with coordinated actions. The standard description is that something high in the brain thinks of a general command like "push!" and sends the equivalent signals downward toward lower systems. Those lower systems receive the general commands, and *elaborate* on them, turning them into more detailed commands at every step. At the lowest level, all of the detailed commands converge into the *final common pathway,* the relatively few channels running from the spinal cord to the muscles. There, at last, the neural signals are turned into tensions that create motions that create behavior.

The problem that nobody has ever been able to figure out is how a simple general command gets turned into specific commands *that will have effects that satisfy the general command.* Unfortunately, neurology is full of sentences that sound like explanations but are really restatements of the effect that is to be explained. When such sentences are uttered, they create the impression that the problem has been solved and needs no further investigation.

The simulator described here shows a different way for commands to get turned into actions. The command that specifies an X force doesn't simply get partitioned among the muscles. It is a request for a *perception,* not a command to *act.* The system receiving this request perceives the X force through a *convergent,* not *a divergent* network. A divergent network cannot be treated as a function; a convergent network can. When the perceived X force matches the reference X force, the cause of the perception must be in one of the states that will, in fact, create that component of force in the X direction. There is an infinity of different muscle tensions that could create the same component of force. If I were not also specifying 2 *other* functions of force, there would be no way to predict the exact muscle tensions that would exist when the X control system experienced zero error.

Since we are specifying 3 functions of 3 variables, and setting reference levels for the value of each function, there is only one state of the muscles that will allow zero error in all 3 systems at once. What we have done, in fact, is set up an analog computer for the simultaneous solution of 3 equations in 3 variables.

This simulator shows that the reference signals for the lower-level systems do not correspond to any one output from a higher-level system. Nevertheless, the perceptual signal sensed by each higher-level system matches the corresponding reference signal. The higher systems each sense a different function of the set of lower-level perceptual signals. Independent control is possible only because the functions represent independent dimensions of variation of the lower-level world.

In the environment of this 2-level system, there is no such thing as X force, Y force, or tone. There are simply 3 tendons in various states of tension. I have *created* the idea of these 3 forces, by designing input functions that will sense them. I could have made one system that would sense force along a set of curved lines representing direction, and another that would sense force along a different set of curved lines crossing the first set; a coordinate system without any straight lines in it. This would result if the sensors were nonlinear, as we know they are. It would have made no difference, except for the fact that there would not have been a simple label like X force to assign as a meaning for the perceptual signals. It would still be possible to specify 3 reference signals and thus set the 3 perceptual signals to specific values, thereby creating a specific state of tension in all 3 tendons that would automatically resist disturbances. The *way* in which the external situation is represented is almost immaterial, as long as 3 reasonably independent perceptual functions are created. There is no coordinate system in the outside world. The behaving system makes up one of its own.

If there were sensors on each muscle to detect muscle length as well as force, we could add 3 more control systems at level 1, and 3 more independent aspects of the external world to control at level 2.

In fact, there *are* muscle-length sensors, and I am working on several models that take them into account.

If you now imagine 500 to 800 muscles involved with at least twice as many level-1 control systems (length and force surely; rate of change highly likely), you will begin to perceive the richness of the world in which level-2 systems exist. Add to this the millions of sensors for heat, cold, vibration, joint angle, light, sound, taste, smell, hunger, pain, illness, angular acceleration, joint compression, and so on, and you might begin to glimpse the complexity of the real system we are modeling. Since perceptions that arise from sources other than direct effects of muscles exist in large numbers, there can clearly be far more level-2 systems than level-1 systems, although the number of level-2 systems that can act independently at the same time is limited by the total number of comparators available at level 1.

Perhaps you can now see why this approach to a model of a human being (rudimentary as it is at this point) has some powerful implications for the building of robots. I suggest a formal distinction between a *robot* (an imitation of a living system) and an *automaton* (a device which automatically produces complex actions). An automaton is designed to create preselected *movements;* a robot is designed to control preselected *perceptions* (its own). In order for an automaton to produce precise and repeatable behavior, it must be built so strongly that normal disturbances cannot alter its movements, or it must be protected from disturbances that might interfere with its movements. In order for a robot to create, for itself, precise and repeatable perceptions (and thus precise and repeatable *consequences* of behavior), it need only perceive precisely, have a sufficiently high error sensitivity, and be capable of producing forces as large as the largest disturbances that might reasonably occur.

There is much more that can be said about the general relationship of one level of control to another, but this installment has raised enough points to ponder. To prepare for part 4, you should run this simulator and observe what happens to all of the variables in it. Try keeping the disturbance constant in magnitude and rotating its angle; try altering the muscle angles; change line 3 to use different error sensitivities (G(x)) and slowing factors (K(x)). Use the C command for longer iterations, and convince yourself that a steady state has really been reached. See what happens if the muscle tone isn't set high enough (there is a very good reason for muscle tone control). Do a series of iterations with slowly changing reference signals, and plot muscle tension against each reference signal. Get the feel of this small extract of the whole human hierarchy because in part 4 we will widen the field of view to include everything, and we will begin to look at some experiments with human subjects. These experiments will be noninvasive, nondestructive—more like video games than science—but far more useful than the games. ■