

*BYTE 4: September 1979*

---

## *The Nature of Robots*

### **PART 4: LOOKING FOR CONTROLLED VARIABLES**

## **In This BYTE**

William T Powers brings his discussion of **The Nature of Robots** to a close by applying the previously-discussed techniques and theories in a simple experiment with a human subject.

*Page 96*

#### **North Star Strings**

*The North Star BASIC string expression  $B$(I, J)$  corresponds to  $MID$(B$, I, J)$  in other versions of BASIC.  $B$(I)$  corresponds to  $RIGHT$(B$, I)$ , and  $B$(1, I)$  corresponds to  $LEFT$(B$, 1, I)$ .*

*Figure, table, and listing numbering continued from part 3.*

This article appeared in BYTE magazine, volume 4, number 9, SEPTEMBER 1979.  
Copyright returned to author. Article recreated by Dag Forssell in 2004.

# *The Nature of Robots*

## PART 4: LOOKING FOR CONTROLLED VARIABLES

**I**n this last part of my series of articles, a simple experiment with a human subject will be attempted; an experiment that can be expanded almost indefinitely. All of the principles from the previous parts will be used. Before the experiment starts, note the following main points that have been established:

- The behavior of an organism is not its output, but some consequence of its motor outputs acting together with unpredictable forces or other disturbances.
- For a more or less remote consequence of motor outputs to be repeatable in a disturbance-prone world, the behaving system must sense the consequence, and act to keep it matching some static or dynamic reference condition. By definition, that makes the organism a control system.
- Organisms acting as control systems control what they sense, not what they do.
- What is controlled is what is sensed, even when the sensing involves one or more stages of real-time computations based on primitive sensory signals.
- In a multiple-level control system, the higher levels act by varying the reference signals for lower-level systems. They control perceptions computed from many lower-level perceptions, some or all of which are controlled by the same lower-level systems.
- If there are  $n$  degrees of freedom at one level of control, in principle  $n$  higher-level systems could act independently and simultaneously by sharing the use of the lower-level systems. Any higher-level system acts by sending amplified copies of its error signal to many lower-level systems, each

with the proper sign to achieve a negative feedback effect. Any lower-level system receives a reference signal that is the net effect of superimposed higher-level output signals. This worked for a 2-level system with 3 control systems at each level; there is no limit, in principle, to the number of levels or the number of systems at each level. In practice, there is reason to anticipate finding hundreds of systems at a given level, but no more than 10 or 12 distinct levels in a human being. This will be commented on later.

Abstract models and simulations are fine for conveying general ideas. However, if one does nothing but make models and simulations, it is easy to get involved in the math and engineering, and forget the real thing is there to be seen. Items described in the first 3 articles in this series represent something *real*. Real organisms work much the same way control systems work. They do not work in any of the other ways that have been proposed over the centuries (as far as their behavior is concerned). I am not talking metaphorically. There are excellent reasons to think that when the properties of organisms begin to be investigated in terms of control theory, hard data about the way we are organized will start to accumulate (up to a point, anyway).

The experiment to be described in this article is so simple that it may look elementary. Nevertheless, it is the starting point for a new approach to exploring the organization of human beings. Most new ideas start by looking like old ones, but with a twist that leads in unexpected directions. If you are familiar with tracking experiments, do not be too quick to decide what this is all about.

## Equipment Required

The basic equipment needed to do this experiment is:

- A joystick with 1 degree of freedom (ie: a potentiometer with a stick on the shaft will suffice).
- A reasonably fast analog-to-digital (A/D) converter with 7-bit or more accuracy. My system uses the Cromemco D + 7A, which has 7 analog channels in and 7 out, as well as 1 input and 1 output 8-bit port.
- A memory-mapped display, in which points are plotted on a video screen by depositing appropriate codes in a reserved segment of memory. This, or something equivalent, is essential for creating the moving objects that are involved in the experiment. I use the Polymorphics VTI with the display area in the 1 K bytes of memory starting at hexadecimal location D000. Out of deference to systems that do not have the VTI's graphics capability (however crude), I have used 64 horizontal elements in the alphabetic mode. Higher resolution would be much more desirable, but this much is enough to show the principles well.

If no memory-mapped display is available, but 2 digital-to-analog (D/A) outputs and a triggered oscilloscope are, the display that is needed can be created. Use 1 D/A converter to deflect the trace in the Y direction, and the other (or 1 bit of a digital port) to trigger the sweep. By starting the sweep and then outputting the 3 cursor values in sequence, a 3-segment trace can be created, with the motion of the cursors being up-and-down instead of side-to-side, as in the following program. Lay the oscilloscope on its side if that deviation bothers you.

Systems with built-in graphics under BASIC control, such as Apple, PET, or TRS-80, will probably allow the experiment to be done more simply than how I did it in listing 5 (See page 12). The basic requirement is to be able to read a number from a stored table, add the handle position to it, erase the old cursor, and use the sum to position the new cursor, doing this for 3 cursors at least 4 times per second—the faster the better. (An example of the simulation on the Apple II is shown in listing 6, page 15).)

## Experimental Design

Imagine a display with 3 cursors on it, one above the other. Each cursor can move left and right. The subject looks at this display while holding a control handle. The instructions for the first experiment are very simple: the subject is asked to select 1 of the cursors, and hold it still, somewhere near the center of the screen as accurately as possible for the duration of the run. Engineering psychologists call this “compensatory tracking.” They use it to investigate the limits of speed and accuracy of control in the presence of rapid disturbances of various kinds.

If the handle is held centered, each cursor will be seen to wander back and forth in a pattern that is independent of the other 2 cursors. In this experiment, the disturbances causing this wandering are made very slow and smooth. With even a slight amount of practice, every subject will be able to maintain essentially perfect control. Transfer functions will *not* be measured, nor will the limits of control be tested in the manner traditional in engineering psychology. A subject acting well within the range of normal operations under conditions where the phenomena of control can be clearly seen is desired. The subject selects a visual variable (position of 1 of the cursors), selects a reference level for that variable (a particular position), and maintains the perceived position at the reference position, while disturbances act that tend to move the cursor away from the reference position.

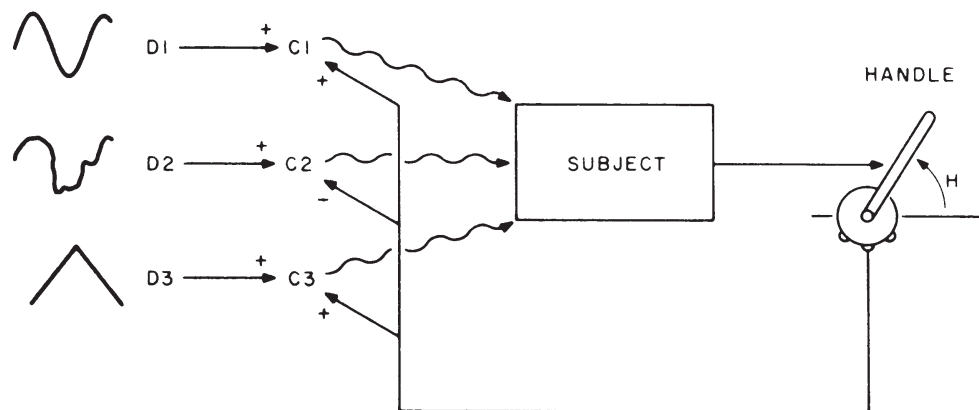
Figure 17 shows the setup in schematic form. The 3 disturbances are labeled D1, D2, and D3. The 3 cursor positions are labeled C1, C2, and C3. The position of the control handle is H. The position of each cursor is determined by the sum of H and one of the Ds. For cursor 2 the effect of the handle is reversed, so the 3 relationships are:

$$C1 = D1 + H$$

$$C2 = D2 - H$$

$$C3 = D3 + H$$

If the subject controls C3 in relation to a reference position of 0 (ie: midscreen), and does so perfectly, then  $0 = D3 + H$ , or  $H = -D3$ . The handle position should be an accurate mirror image of the magnitude of the disturbance D3 at every moment, and the cursor C3 does not move at all. You will find that all subjects, after a little practice, will closely approximate these predictions.



**Figure 17:** Schematic arrangement of experimental setup. Three slow and smooth disturbances are added to a measure of the handle position (with a negative sign for the middle one), to determine the position of 3 corresponding cursors. The subject selects 1 cursor and a reference position for it, and uses the handle to maintain the cursor at that position. A run lasts about 1 minute, and 250 samples of handle position are recorded. For plotting, the cursors are reconstructed from the tables of disturbances and the corresponding records of handle position.

This may seem elementary, obvious, boring and hardly worth the labor of getting the experiment up and running. Do not be deceived; this experiment appears to be simple because it is fundamental. It is fundamental because it can *prove* that all of the life sciences have been using the wrong model. There are also several extensions of the experiment that will show how to get started mapping the whole hierarchy of human control systems. There is no theory and no simulation that carries the impact of *seeing* how a real living control system works; especially when you can understand every detail of what is happening, either as subject or observer. The 3 previous articles in this series have been designed to give the ability to grasp what is happening here. This experiment is designed to give the gut feeling of *knowing*.

### Program Structure

The program in listing 5 is written in North Star BASIC, Version 6, Release 3. It contains a machine-language subroutine for an 8080/Z80 processor which is loaded by the BASIC program at any specified 256-byte memory-address boundary (specify in hexadecimal only the most significant byte of the location of the subroutine).

The *machine-language subroutine* reads in the handle position, adds it with the appropriate sign to the value of a disturbance that is passed to the sub-

routine by the CALL command (in the DE register pair), erases the old cursor, and deposits the new cursor, a rubout, on the screen. Each time the subroutine is called it steps to the next cursor, recycling as necessary. On return from the subroutine, the handle position is passed back to the main program (in the HL registers). The machine-language program is in lines 200 thru 230, expressed as a string of hexadecimal bytes with no punctuation. Thus if your machine is not an 8080/Z80 type, a program can be assembled, the listing copied into these lines, and possibly this program can be made to work with little other modification.

The program asks for the most significant byte of the place where the machine-language subroutine is stored. The loader adjusts memory references by inserting the value of this byte in memory wherever necessary, after the program is loaded (lines 300 thru 330).

The *display area* consists of 1 K bytes of memory starting on any 256-byte boundary. Lines 370 thru 400 ask for the starting location of the memory area devoted to the display, and set up base registers in the machine-language program for the left margin of each cursor's movement. The FILL command is like POKE. If the computer has graphics capability built-in, everything from line 60 thru 400, and the plotting subroutine (later), can be accomplished in a simpler way.

*Disturbance tables* are set up in lines 510 thru 620. The unnecessary use of symbols, instead of constants, is an attempt at acceleration. It still takes a minute to load the 3 disturbance tables, each 250 bytes long. All long tables are *strings*; only 8 bits of accuracy is needed, so by using the CHR\$ and ASC functions, the tables can be stored 1 byte per value instead of 5 bytes per value. Disturbances are in tables because BASIC cannot calculate them fast enough.

Disturbance D1 is a sine wave and D3 is a triangular wave. D2 is a smoothed random disturbance. On reruns, only D2 is reloaded, taking about 20 seconds.

*The experimental run* is controlled by lines 660 thru 780. Lines 660 and 680 lay down 3 arbitrary scales on the screen, while the rest repeatedly call the machine-language subroutine. For each stored value of each disturbance, all 3 cursor positions are computed and plotted, and the handle position is stored in the table H1\$. The inner loop from line 710 to line 770 adjusts the duration of the experimental run; here it is set up so that the disturbances change and a handle position is recorded only every fourth time the display is generated. On my system, this works out so the display is refreshed 16 times per second, and data is sampled and stored 4 times per second. The 2 OUT statements reflect my laziness; I use 2 digital-to-analog outputs to supply the voltage to the potentiometer that measures handle position.

*The data plotting routine* (lines 820 thru 1010) is entered at the end of an experimental run. This routine is set up to plot either on the video screen or on a hard-copy device; it asks for the X and Y dimensions of the plot, which cursor is to be plotted, and which device is to be used. My system is set up so the typewriter is device 2 and the screen is any other device number. If you do not have this ability in your BASIC or system, delete lines 1060 and 1070 (in the subroutine that requests information about the display), and eliminate the "#2," in lines 970 and 990. In North Star BASIC, the exclamation point is short for PRINT.

Only the handle position is stored as data; the cursor positions are reconstructed during plotting from the list of handle positions and the corresponding tables of disturbances.

The plotting scheme is designed to work with any teletypewriter-like device. If you have legitimate graphics, you can rewrite this part and get a more pleasing result.

There are 3 choices for plotting, each associated with cursors C1, C2, and C3. Each plot shows the cursor as a C, the handle position as an H, and the disturbance acting on the cursor as a D. A dot indicates the center of the display when nothing else is there. After each plot is finished, there is a pause; hitting the carriage return will cause the program to ask about the next plot. If the question about the Y dimension of the display is responded to with a 0, the program will reload the random disturbance table and issue a prompt for another experimental run. The old data will be destroyed. Remember, it takes about 20 seconds to reload the random disturbance table. Do not panic if there is a long pause.

At line 1260 there is a utility routine that converts any hexadecimal number up to 10 digits to a decimal number. I used it while writing the program. It calls the conversion subroutine starting at line 1130.

## Running the Experiments

If you possibly can, take the trouble to set this experiment up. Nothing can take the place of actually experiencing yourself as a control system and understanding things that you have taken for granted all your life.

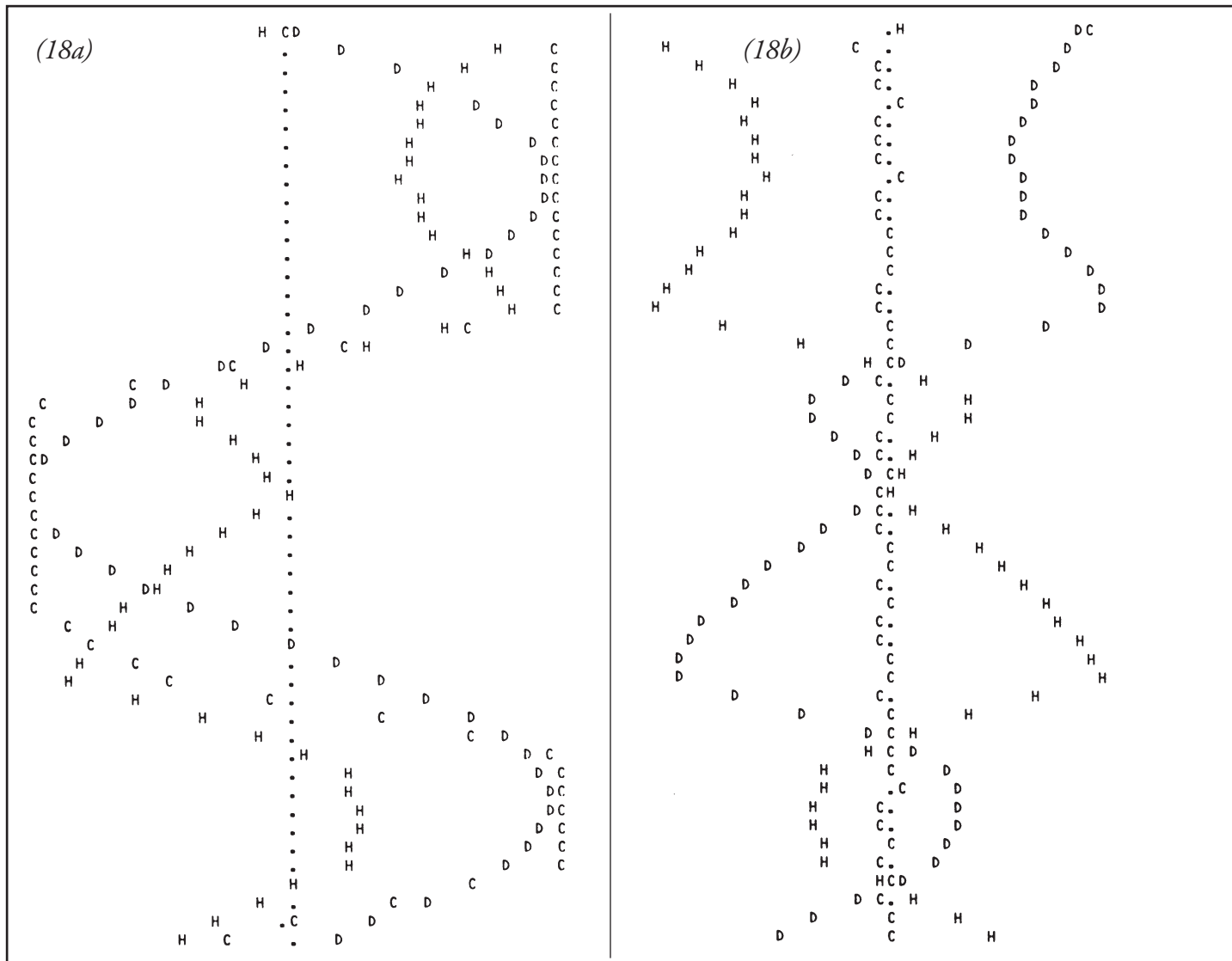
Here is a typical run for the benefit of the many readers who do not have the equipment to do this; the data will then be observed. Here is an old friend, Chip Chad (from part 1 of this series), glaring at the screen and maintaining a choke-hold on the handle, waiting for the experimenter to hit the return key at line 610. The experimenter reaches in and taps the key. The reference scales slide up into place and the 3 cursors pop into view, moving. Chip picks the middle one as most people do the first time, decides to keep it on the middle + mark, and after a few wobbles succeeds.

"So what?" he says.

If learning were being studied, good information could be obtained from this first run. But the plan is to see Chip acting as a *competent* control system, so his first effort is praised and he is given another run (answering the query about Y dimension with a 0). After the second run, the data is plotted for each cursor.

Figure 18 shows the data for each cursor, number 1 on the left, 2 in the middle, and 3 on the right. The 2 end plots are a mess, but the middle plot shows a striking symmetry. The Cs march more or less down the

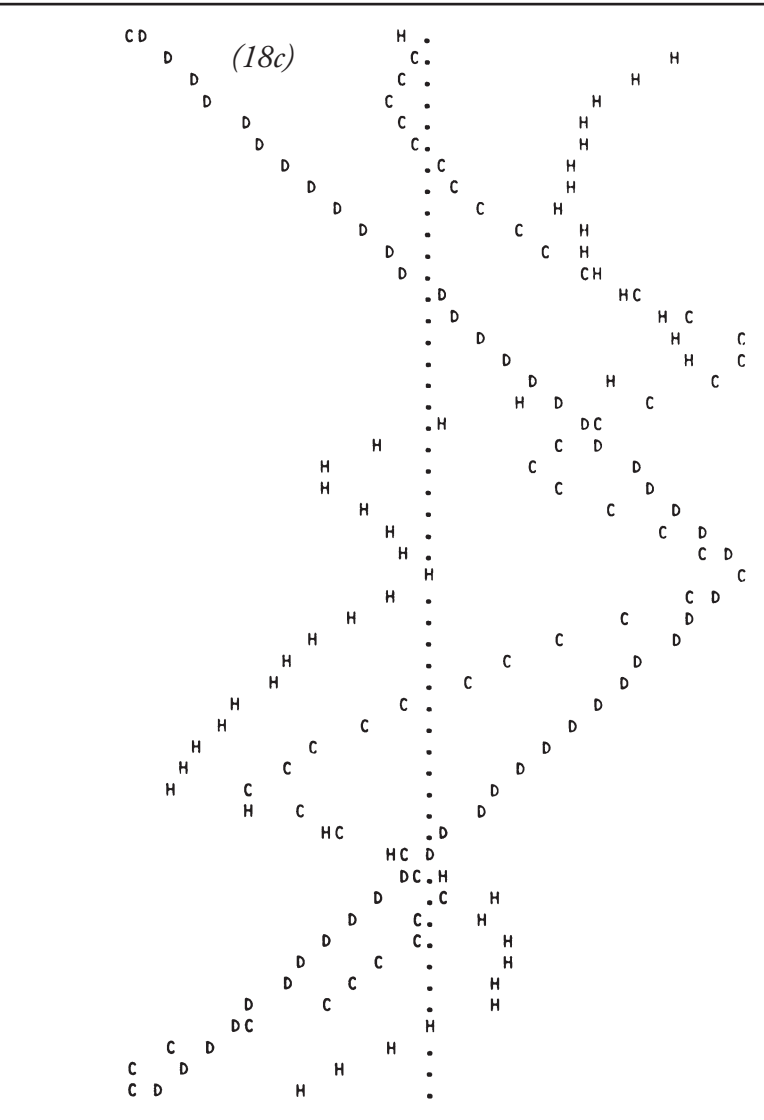
**Figure 18:** A typical run for a practiced subject. In figure 18a is the record for D1, C1, and H. Figure 18b has the record for D2, C2, and H; figure 18c has the record for D3, C3, and H. In figure 18b, the cursor is held near the center, while the handle position is at all times very nearly the mirror image of the disturbance amplitude. It is very easy to decide which cursor was under control.



center of the screen, deviating a little to left and right, but maintaining a constant position on the average. The Ds make a random-looking pattern, and the Hs follow almost the *mirror image* of the D pattern.

Looking carefully at the middle plot, could it be said that the handle position or motion looks like any sort of regular function of the cursor position or motion? There may be *some* relationship, but it certainly is not clear. Probably, nobody would claim

that the large, smooth motions of the handle could be reconstructed *accurately* on the basis of measurements of cursor position (that is, reconstructed roughly or statistically with accuracy, especially if handle *acceleration* is compared with cursor deviation from the average position). The best which could be hoped for would be some statistical relationship (eg: a small signal buried in much noise).



**Figure 19:** Cause and effect paradox. Under the old concept that stimuli cause behavior, the cause and effect chain runs from the disturbance to the cursor, through the subject, to the behavior. However, the correlation of the disturbance and the cursor position is very low, as is the correlation of the cursor position and handle position (for the controlled cursor). This would lead to a prediction of an even lower correlation of disturbance and behavior. In fact, that correlation is normally very high (0.99 or better). Only the control theory analysis of this experiment can explain this otherwise paradoxical situation.

(Note: Figure 19, apparently an illustration of correlations, is nowhere to be seen in the original article. This was most likely an accidental omission, but the illustration is not available now, 25 years later.)

of which might have some relationship to that behavior. Which variable, D or C, would be selected by any statistical test as the most probable cause of the behavior? Of course, D would be selected. In fact, a formal statistical analysis, like those done in every scientific study of behavior, shows D to be the only significant contributor to the behavior, while C, the cursor position, is rejected as an irrelevant variable!

That is a paradox, however, from the traditional point of view. The only way D can affect Chip's behavior is through its effects on C, since all that Chip can sense is the cursor position. The disturbance itself is invisible. If C does not correlate with the behavior, then how can anything that acts exclusively through effects on C correlate any better with behavior? Yet a typical correlation between C and H is around 0.1, while the correlation of H with the corresponding D is typically 0.995. See figure 19.

That is the proof mentioned earlier. The old cause-effect model fails utterly when applied to this situation. The question then is, why have generations of intelligent people believed that behavior is caused by sensory stimulation? The answer is clear: they have been fooled by a monstrous illusion.

The illusion would be easier to see if there was some visible, direct indication of the magnitude of the disturbance. Suppose there were a moving D (or a number that continually reflected the magnitude of D) on the display. Clearly, if Chip managed to control C without that *indication*, he could still do so; he could ignore it and perform as well as ever.

On the other hand, the relationship between the handle position and the magnitude of the invisible disturbance is obvious and quantitative. It is seen that the handle position is the mirror image of the disturbance magnitude with an error of only a few percent of full scale. There is much signal and little noise in that relationship.

Here is the situation. There is 1 measure of Chip's behavior, H. There are 2 variables, D and C, either

However, something has now been added that would mislead a bystander who did not understand control theory.

That bystander could now see 2 variables, both able to affect Chip's senses. Taking the apparent relationships at face value, it would be clear that the *indication* of D was accurately associated with the handle position; while the movements of the cursor, such as they are, show no such association. Furthermore, the variations of D are large and smooth, and there is no observable relationship between D and C. Why should the bystander suspect that C is being affected by D in one way and affected by H in an opposite way? The obvious conclusion is that the variations in D are causing Chip's behavior, while C has nothing to do with his behavior, especially if C does not vary more than the fixed background scales do. If the screen were full of irrelevant cursors, jiggling around slightly, how could the bystander pick C as something of special importance? If BASIC were fast enough, I would have included such irrelevant cursors; the point being made here would then be obvious.

An organism is surrounded by a world full of variables; variables that change within widely diverse ranges. The organism receives many signals from its internal parts, too. In that sort of situation, if the organism is controlling some of the variables, it will react strongly and smoothly to any disturbance tending to alter 1 of the controlled variables. The result is that it will *seem* to be responding directly to the disturbances. There will be no obvious indication that it is controlling anything at all. There is every excuse for even the best of scientists to have observed the relationship between disturbance and behavior, and to have missed the very existence of controlled variables.

The name for such disturbances is *stimuli*. Once in a while, an experimenter must have accidentally picked a real controlled variable to call a stimulus, but the chances are against that. If an attempt is made to manipulate a real controlled variable, the organism will have to be strapped down to keep it from interfering. That is what is done in such cases. If the organism insists on acting like a control system, forcibly break the loop and *make* the organism conform to the theory. As a famous psychologist said, the theme is "Behave, damn it!" It never occurs to such stong-willed individuals that they might have the wrong idea about what is happening.

There is more in this elementary experiment than meets the eye. If all psychologists were to experience it, and try to meet the challenge of explaining these effects using any standard theory, the result would be a total collapse of that science, followed by a rebirth. However, many jobs would be threatened. What has happened instead is that a handful of psychologists has supported this theory, another handful has taken up arms against it, and most have resolutely ignored it.

I suggest that you run this experiment many times with subjects controlling all 3 cursors. Every case will show that mirror-image relationship between D and H and little relationship between C and either D or H. If the previous parts of this series are studied and all the relationships that make up a control system thought about carefully, it will be evident that there is no other explanation for what is going on here. If you get nothing else out of this, you should acquire an intuitive feel for a new theory of how behavior works. You might even begin to understand how to design a robot in a new way.

It is time now to try to fulfill a promise implied in part 1, to show how anyone with a home computing system can make important contributions to this new science of human nature. The best way this can be done is to start with the experiment used, and to show how it can be extended to become a powerful tool for investigating human organization. The main objective will be to introduce the *test for the controlled variable*, the nearest approach I know of to mind reading.

### More Controlled Variables

Once subjects controlling all 3 cursors have been seen, it might seem that the possibilities of this experiment have been exhausted; this is not the case at all. There are controllable variables all over that screen; all of them can be controlled by the same means, movements of the handle in 1 dimension. Discovering them is a good way to get out of the habit of thinking that we simply perceive our environment, and start a new way of thinking: to recognize that we *construct* perceptions, imposing order on our experiences far more than *recognizing* order. As you will see, a controlled variable does not have to be "real" at all.

Here is an example. It is possible to perceive the *relative* position of any of the 2 cursors. The handle affects C2 in a direction opposite to its effects on C1 and C3, so the relative position of C1 and C3 can-



not be controlled because the handle does not affect it. However, it is possible to keep C1 even with C2, or C2 even with C3; in fact, it, is easy. A plot of the results would involve plotting C2–C1 or C3–C2 instead of just C, and D2–D1 or D3–D2 instead of just 1 disturbance. The mirror image relationship with H would be as good as ever. Do not forget that C2–C1 and C3–C2 are *variables*. Any value of the variables can be selected as a reference level (eg: C1 to be 1 inch to the left of C2).

These are examples of *higher-level* controlled variables. If the subject could not perceive the present positions of the cursors, he or she certainly could not perceive their *relative* positions. Relative position is derived from perceptions of individual positions, but not vice versa. In order to *control* relative positions, it is necessary to control (or at least vary) individual positions, but individual positions can be controlled without controlling relative positions. These are the relationships one looks for to map out a *hierarchy* of perception and control.

Other relative perceptions can be controlled. All 3 cursors can be kept lying in a straight line, at least within the range where 1 of them does not fall off the edge of the display and pop up at the other edge. Reducing the amplitude of the disturbances would eliminate that problem. Also, the 3 cursors can be made to form any fixed angle, subject to the same limitation. There may be more static patterns that can be controlled, but I have not thought of any. This is, after all, a simple display.

It is not, however, limited to static conditions. Suppose the subject visualizes a pattern in which 1 cursor moves back and forth slowly between 2 limits. This pattern can easily be maintained, the handle moving just enough to produce it, and enough more to cancel the effects of any of the disturbances. A similar oscillation could be maintained for the *relative* variables. This is a still higher-level variable, a temporal pattern. The subject chooses which temporal pattern to perceive, and what state of that kind of pattern to maintain. Control still requires only the use of the 1-dimensional effect caused by the handle.

There is clearly an infinite range of different temporal patterns, ranging from a simple steady motion in 1 direction to completely arbitrary motions and rhythms. There is an *unlimited* number of potential controlled variables in this simple display. Anything that can be perceived, and that the handle can affect in a systematic way, can be controlled.

For all of these examples of controllable perceptions, it is essential to remember that the disturbances are acting all the time. This is not a matter of producing any particular *behavior*. The cursor cannot be made to move slowly back and forth between fixed limits just by moving the handle slowly back and forth between fixed limits. The handle might be moving the wrong way at many moments, when the disturbance tends to make the cursor move faster than the reference pattern being considered. There is no one-to-one correspondence between handle position or velocity and cursor position and velocity, because of those ever-present disturbances. Regularities of *behavior* are not being looked at here, but regularities of controlled perceptions. If there were a slowly oscillating prism between the display and the subject's eyes, a regular pattern of movement of the cursor on the screen would not be seen. The subject controls the visual image, not the reality. For the higher-level variables, the subject controls some *function* of the visual image (often the controlled variable could not be found, even on the retinas).

One could create displays of far greater complexity, and provide means of affecting the display that have more than 1 degree of freedom to explore a staggering range of possible controlled variables. This is what I suggest be done. The first step in the development of any new science is acquire the facts; here the most needed facts concern *what variables human beings can actually control*. What is needed is a large and simpleminded program of recording the obvious and obscure. What is needed is a body of definitions of variables in *every* sensory mode that people have been able to control. Order and system count much less than sheer volume of data at this point. In fact, an *unsystematic* gathering of data may be the best kind, since it will not be constrained by theories about what people *ought* to be able to control. Anything which can be a way of testing is worth testing at this stage. The possibilities are limited only by the imagination.

We do need some sort of ordering principle—some criterion for judging the reality of any proposed controlled variable. This is where the test appears; here is how it works.

## Test for Controlled Variables

The first thing to remember when investigating a possible controlled variable is that in order for something to be controllable it has to be variable. There is neither the means nor the need to control the existence of the Empire State Building or the planet Jupiter. Not all perceptions are controlled. Some are just disturbances; some are just there.

One might think initially about controlling, for instance, a car. People often speak casually about controlling *things*. But what is meant is controlling *something about* those things. A person cannot really control a car; but under proper circumstances its shape, its color, its price, its speed, its direction, its parking place, its dirtiness, its dangerousness, its desirability, its altitude, or the flatness of its tires can be controlled. A car, after close inspection, proves to be composed entirely of hundreds or even thousands of variables. Together they create “car-ness” in our perceptions. Individually, or in groups, most of them can be affected by one means or another, and can be controlled if it is worth the effort. You can even make the car disappear instantly by closing your eyes. Keep remembering that what is controlled is really a *perception*.

The first step in applying the test for the controlled variable is to define a variable. You do not have to know in advance if it is a controlled variable; you do not even have to know where the supposed control system is. All you have to do is to pick out something that you know is variable and “push” on it.

By push I mean to apply a disturbance that under normal circumstances should have a predictable direction and amount of effect on the variable. If I push hard enough on a life-sized statue, it should tilt in the direction of the push. Perhaps it will topple in that direction according to the simple laws of mechanics.

Having selected a variable and applied a push to it, the next step is to measure the actual effect of the push. I *predict* that pushing on this statue should make it tilt a certain amount in a certain direction. I apply the push and observe the tilt.

If the *actual* effect is far smaller than the *predicted* effect, common sense indicates that something must be pushing back. If the pushing-back is always just enough to cancel any amount or direction of disturbance (within some limits), it can be concluded that the pushing-back is *systematic*. The mirror-image effect that has been observed is what is wanted.

It is necessary to discover *what* is pushing back, and how it is doing the pushing. Perhaps, examining the statue carefully, an iron rod is found supporting its back from its base. In that case, a conclusion is made that there were not enough facts to make a correct prediction of the effects of the push; the bending moment of the rod should have been taken into account. But if no simple explanation for the failure of the prediction is found, one must look further.

Suppose it is discovered that the base of the statue seems to move when pushed. If there is a push to the east, the base tilts to the west moving the center of support east of the center of gravity of the statue, and thus creating a counterforce. Suppose this tilt of the base is found to be always just what is required to offset the effects of the push. It can be concluded that one may be on the trail of a control system.

What has been done is to find out something about the *means* of control, the path by which the output of the control system, if it exists, might be linked to the controlled variable (the angle between the statue’s longitudinal centerline and the vertical). Finding this link is a necessary step in the test.

That step will usually lead to discovering the physical control system. Tracing the wires that work the motors that tilt the base of the statue, you find a black box a few yards away from the statue. That may be the control system, or at least all of it that is not its actuators (which have been found).

There is still one step to be taken. You cannot be completely sure of the nature of the control system until you discover the variable it is really sensing. The situation has been approached with human prejudices; to me, it seems that the controlled variable is the orientation of the statue, a geometric or visual variable. Perhaps that variable is only *related* to the real controlled variable. What must be found now are the *sensors* that the control system is using.

Thinking in visual terms, you might look for a photocell that detects the tilt. Suppose a photocell is found on a stand near the statue. The test calls for breaking this link, preventing the sensing of the statue. The result should be that the effect of the push returns to what would be predicted from mechanical laws. So the photocell is covered and the disturbances are applied again. What happens is that the floodlights illuminating the statue turn on. The statue still resists the push—the photocell was for something else.

By careful searching 4 strain gauges built into the base of the statue are discovered. These provide a signal showing where the center of thrust is, and the wires from the strain gauges run over to that black box. Disconnecting the wires shows that *now* the push succeeds in tilting the statue. As soon as its tilt becomes marked, an angry groundskeeper comes leaping out of the bushes and arrests the experimenter. Aha! You may have discovered *another* control system controlling the state of the statue.

To recapitulate, the test for the controlled variable involves the following steps:

1. Define a variable.
2. Apply various amounts and directions of disturbances directly to the variable.
3. Predict the expected effects of the disturbances, assuming no control system is acting.
4. Measure the actual effect of the disturbances.
5. If the actual effect is essentially the same as the predicted effect, stop. No control system is found.
6. If the actual effect is markedly smaller than the predicted effect, look for the cause of the opposition to the disturbance, and determine that it results from systematic variations in some other variable. If such a cause is found, it may be associated with the output of a control system.
7. Look for a means of *sensing* the controlled variable. If none is found, stop: no control system is proven to exist.
8. If a means of sensing is found, block it, so the variable cannot be sensed. If control is *not* lost, the sensor is not the right one. If no such sensor is found, stop: no control system is proven to exist.
9. If all steps of the test are passed, the variable is a controlled variable, its state is its reference level, and the control system has been identified.

To apply step 8 of the test to our computer experiment, cover the cursor suspected of being controlled with a cardboard strip. Control should be lost. Cover *each* cursor. The covered one will never pass the test. The other steps are easily carried out.

## Concluding Remarks

Now it is up to you. You can test controlled variables involving intensity, sensation, configuration, change, sequence, relationship, strategy, principle, and system concepts having to do with visual, auditory, tactile, kinesthetic, and other senses.

Good luck with the programs, and good hunting for controlled variables. I will be interested to receive word about what people are doing with the information covered in these articles. ■

## BIBLIOGRAPHY

Powers, W T, *Behavior: The Control of Perception*, Aldine Publishing Co, 200 Saw Mill River Rd, Hawthorne NY 10532, 1973.

*Behavior: The Control of Perception* and other books by Bill Powers are now available from <http://www.livingcontrolsystems.com>

*Listing 5: North Star BASIC control-variable simulation. The necessary assembly language routines needed for execution are also given.*

```

10DIMH$(16),D1$(250),D2$(250),D3$(250),H1$(250),B$(82),S$(82)
20 DIM A$(2)
30 H$="0123456789ABCDEF"
40 INPUT "SEED FOR RANDOM GENERATOR (1 - 100) ",A\ Z=RND(A/100)
50 REM *****
60 REM CONVERT 2 HEX DIGITS TO DECIMAL
70 REM *****
80 DEF FNB(A$)
90 U=ASC (A$ (1,1) )\IF U<58 THEN U=U-48 ELSE U=U-55
100 V=ASC(A$(2,2))\IF V<58 THEN V=V-48 ELSE V=V-55
110 RETURN 16*U+V
120 FNEND
130 REM *****
140 REM SET MACHINE-LANGUAGE PROGRAM ORIGIN!
150 REM *****
160 INPUT "MOST SIG. BYTE, SUBROUTINE LOCATION: ",S$
170 GOSUB 1130
180 !"6 SEC TO LOAD SUBROUTINE"
190 M1=256*A0\ M2=M1+9
200 DATA "02000000000000000004BDB19071 F1 F473A0800FE00C21000782F"
210 DATA "3C477881E63F4F2A00003A08003C3CFE06DA2FOOAF"
220 DATA "320800856F5E23563EA0127BE6C0B15F3EAAI2722B73DB19EE80"
230 DATA "6F2600C9"
240 M=M1
250 READ B$\FOR J=1 TO LEN(B$)-1 STEP 2\A=FNB(B$ (J,J+1) )
260 FILL M,A\ M=M+1 \NEXT J\ IF A<>201 THEN 250
270 REM *****
280 REM INSERT RELOCATION BYTES
290 REM *****
300 FILL M1+1,A0\ FILL M1+FNB("12"),A0
310 FILL M1+FNB("17"),A0\ FILL M1+FNB("23"),A0
320 FILL M1+FNB("2D"),A0
330 FILL M1+FNB("26"),A0\ FILL M1+FNB("31"),A0
340 REM *****
350 REM SET LOCATIONS FOR DISPLAY
360 REM *****
370 INPUT "MOST SIG. BYTE, DISPLAY LOCATION: ",S$
380 GOSUB 1130
390 FILL M1+2,FNB("00")\ FILL M1+3,A0\ FILL M1+5,A0+2
400 FILL M1+6,FNB("40")\ FILL M1+7,A0+3
410 REM *****
420 REM LOAD DISTURBANCE TABLES
430 REM *****
440 W=3*3.1415927/250\RO=RND(0)\R1=32*R2=R1
450 !\!\!\A!\!\ [ONE MINUTE TO LOAD DISTURBANCE TABLES]"
460 !\!"THE SCREEN WILL CLEAR AND THREE SCALES WILL APPEAR."
470 !\!"THEN THREE CURSORS WILL APPEAR, ONE FOR EACH SCALE."
480 !\!"PICK ONE CURSOR AND TRY TO HOLD IT IN ONE POSITION"
490 !\!"FOR THE DURATION OF THE RUN, AS EXACTLY AS YOU CAN."
500 !\!" [STAND BY FOR PROMPT]"
510 N1=32\ N2=31\ N3=25\ N4=64\ N5=10
520 FOR J=1 TO 250
530 D1$ (J,J)=CHR$ (N1+N2*SIN(W*J) )
540 D3$ C,J)=CHR$ (64-ABS (J-125) /2)
550 NEXT J
560 !" [LOADING RANDOM DISTURBANCE: STAND BY]"
570 N3=25\FOR J=1 TO 250
580 IF J-N3*INT(J/N3)=0 THEN R0=N4*RND(0)
590 R1=R1+ (R0-R1) /N5\ R2=R2+ (R1-R2) /N5\ D2$ (J,J)=CHR$ (R2)
600 NEXT J

```

```

610 INPUT"READY TO GO: HIT RETURN TO PROCEED. ",A$
620 FOR J=1 TO 16:\NEXT
630 REM *****
640 REM EXPERIMENTAL RUN
650 REM *****
660 !\!\!\GOSUB 680\!\!\!\AGOSUB 680\!\!\!\GOSUB 680\!\
670 GOTO 690
680 FOR I=1 TO 8!"TTTTTTT+",\NEXT I\ RETURN
690 FILL M1+8,4\ REM SYNCH CURSOR COUNTER
700 N1=8\N2=127\N3=128\ FOR J=1 TO 250
710 FOR L=1 TO 4
720 OUT 25,N2\ OUT 26,N3
730 H=CALL (M2,ASC (D1$ (J,J)))
740 H=CALL (M2,ASC (D2$ (J,J)))
750 H=CALL(M2,ASC (D3$ (J,J)))
760 NEXT L
770 H1$ (J,J)=CHR$ (H)
780 NEXT J
790 REM *****
800 REM DATA PLOTTING PROGRAM
810 REM *****
820 GOSUB 1050\ IF Y0<1 THEN 560
830 !"AFTER PLOT, HIT RETURN TO CONTINUE"
840 INPUT"WHICH CURSOR (1,2,3)? ",I
850 IF I=0 THEN 560
860 FOR W=1 TO 250 STEP INT(250/(Y0+1))
870 H= (ASC (H1$ (W,W) )-128)*X0/128
880 ON I GOTO 890,900,910
890 V=ASC(D1$(W,W))-32\ GOTO 920
900 V=ASC(D2$(W,W))-32\ H= -H\ GOTO 920
910 V=ASC (D3$ (W,W))-32
920 V=V*X0/64+1
930 C=V+H+Z0\ IF C<1 THEN C=1\ IF C>X0 THEN C=X0
940 V=V+Z0\IF V<1 THEN V=1\ IF V>X0 THEN V=X0
950 H=H+Z0\ IF H<1 THEN H=1\ IF H>X0 THEN H=X0
960 B$=S$B$(Z0,Z0)
970 !#T0,\B$(V,V)="D"\ B$(H,H)="H"\ B$(C,C)="C"
980 U=0\ IF V>U THEN U=V\IF H>U THEN U=H\IF C>U THEN U=C
990 IF Z0>U THEN U=Z0\B$=B$(1,U)\ !#TO,B$,
1000 NEXT W
1010 INPUT1"" ,A$\ GOTO 820
1020 REM *****
1030 REM SET UP FOR PLOTTING (SUBROUTINE)
1040 REM *****
1050 !\INPUT "Y-DIMENSION OF PLOT (0 = NEW RUN): ",Y0\ Y0=Y0-2
1060 IF Y0<1 THEN RETURN
1070 INPUT "X-DIMENSION OF PLOT (1-72) : ",X0
1080 IF X0>72 THEN 1070\ IF X0<1 THEN 1070\X0=X0-2
1090 INPUT "OUTPUT DEVICE (T OR S)",A$
1100 IF A$="T" THEN T0=2 ELSE T0=0
1110 S$=""\FOR I=1 TO X0\S$=S$+" "\ NEXT I
1120 Z0=INT(X0/2)\ RETURN
1130 REM *****
1140 REM CONVERT HEX IN S$ TO DECIMAL IN A0
1150 REM *****
1160 A0=0\K=1\FOR J=1 TO LEN(S$)-1 \K=K*16\NEXT JK=INT(K+.01)
1170 FOR I = 1 TO LEN(S$)
1180 FOR J=1 TO 16
1190 IF S$ (I,I)=H$ (J,J) THEN EXIT 1220
1200 NEXT J
1210 !"NOT HEX NUMBER"\ EXIT 160
1220 A0 = A0 + K*(J-1)\ K=K/16
1230 NEXT I
1240 RETURN

```

```

1250 REM *****
1260 REM UTILITY, CONVERT HEX TO DECIMAL
1270 REM UP TO TEN HEXADECIMAL DIGITS
1280 REM DO "RUN 1300"
1290 REM *****
1300 DIM H$(16) \HS="0123456789ABCDEF"1 !
1310 INPUT1"HEX= ",S$\GOSUB 1160!"    DECIMAL= ",A0\GOTO 1310

```

---

```

001          *          MACHINE LANGUAGE SUPPORT ROUTINES
002          *
003          ORG      0
004          *
005 0000 0200      ADRO   DBL      ADR1
006 0002 0000      ADR1   DBL      0
007 0004 0000      ADR2   DBL      0
008 0006 0000      ADR3   DBL      0
009 0008 00        COUNT  DATA    0
010 0009 4B        START  MOV      C,E
011 000A DB19      IN      25          GET HANDLE
012 000C 071 F1 F  ARS          DIVIDE BY TWO
013 000F 47        MOV      B,A      SAVE IN B
014 0010 3A0800    LDA      COUNT    CHECK FOR MIDDLE ONE
015 0013 FE02      CPI      2
016 0015 C21C00    JNE      S1
017 0018 78        MOV      A,B      I F MIDDLE ONE NEXT,
018 0019 2F        CMA          MAKE HANDLE NEG.
019 001A 3C        INR      A        (TWO'S COMPL. )
020 001B 47        MOV      B,A
021 001C 78        S1      MOV      A,B
022 001D 81        ADD      C          X=X+HANDLE
023 001E E63F      ANI      :3F      LIMIT TO 63
024 0020 4F        MOV      C,A      SAVE X IN C
025 0021 2A0000    LHLD   ADRO      GET BASE ADDRESS
026 0024 3A0800    LDA      COUNT    GET DISPLACEMENT
027 0027 3C        INR      A
028 0028 3C        INR      A          BUMP TWICE
029 0029 FE06      CPI      6
030 002B DA2F00    JLS      S2          CHECK MODULO 6
031 002E AF        ZAR
032 002F 320800    S2      STA      COUNT
033 0032 85        ADD      L          MAKE ADDRESS FOR
034 0033 6F        MOV      L,A      CURRENT CURSOR.
035 0034 5E        MOV      E,M
036 0035 23        INX      H
037 0036 56        MOV      D,M      DE=OLD SCREEN ADR.
038 0037 3EA0      MVI      A,:AO    LOAD A SPACE
039 0039 12        STAX D    ERASE OLD CURSOR
040 003A 7B        MOV      A,E
041 003B E6C0      ANI      :CO      ZERO DISPLACEMENT
042 003D B1        ORA      C        NEW DISPLACEMENT
043 003E 5F        MOV      E,A      POINTER FIXED
044 003F 3EAA      MVI      A,:AA    LOAD ASTERISK CURSOR
045 0041 12        STAX D    PUT IT ON SCREEN
046 0042 72        MOV      M,D      SAVE CURSOR
047 0043 2B        DCX      H        ADDRESS
048 0044 73        MOV      M,E
049 0045 DB19      IN      25          GET HANDLE AGAIN
050 0047 EE80      XRI      :80      RANGE 0-255
051 0049 6F        MOV      L,A
052 004A 2600      MVI      H,0
053 004C C9        RET

```

```

5  HIMEM: 8192
10  DIM D1 %(250),D2%(250),D3%(250),H1 %(250)
20  INPUT "SEED (0-100): ";A
30  Z = RND (A / 100)
40  REM LOAD DISTURBANCE TABLES
50  W = 3*3.141592654 / 250
60  RO = RND (0);R1 = R2 = 140
65  PRINT : PRINT : PRINT : PRINT : PRINT : PRINT : PRINT : PRINT
70  PRINT : PRINT : PRINT "LOADING DISTURBANCE TABLES"
75  PRINT
80  PRINT "WHEN SCREEN CLEARS, BACKGROUND WILL"
85  PRINT
90  PRINT "APPEAR - THEN THREE CURSORS."
95  PRINT
100 PRINT "PICK ONE CURSOR AND HOLD IT IN"
105 PRINT
110 PRINT "ONE POSITION FOR THE DURATION OF"
115 PRINT
120 PRINT "RUN, AS ACCURATELY AS YOU CAN"
125 PRINT
130 PRINT "STAND BY FOR PROMPT MESSAGE"
140 FOR J = 1 TO 250
150 D1 %(J) = 140 + 130 SIN (W * J)
160 D3%(J) = (125 - ABS (J - 125)) * 270 / 125
170 NEXT J
175 PRINT
180 PRINT "RANDOM DISTURBANCE LOADING: STAND BYE."
185 N3 = 25;R1 = 140;R2 = 140
190 FOR J = 1 TO 250
200 N3 = N3 - 1: IF N3 > 0 THEN 210
205 N3 = 25;R0 = 280 * RND (5)
210 R1 = R1 + (RO - R1) / 05;R2 = R2 + (R1 - R2) / 5
220 D2%(J) = R2
230 NEXT J
240 PRINT : INPUT "HIT RETURN FOR RUN";A$
250 HGR
255 HCOLOR = 3
260 POKE 49234, 0
261 FOR X = 1 TO 280 STEP 10
262 FOR Y = 43 TO 143 STEP 50
263 HPLOT X,Y: HPLLOT X,Y + 14
264 NEXT Y: NEXT X
270 FOR J = 1 TO 250
280 FOR K = 1 TO 4
290 H = PDL (0) - 128
299 HCOLOR = 0: HPLLOT C1 %, 45 TO C1 %,55: HCOLOR = 3
300 C1% = D1 %(J) + H
305 IF C1 % < 0 THEN C1 % = 0
306 IF C1 % > 279 THEN C1 % = 279
307 HPLLOT C1 %,45 TO C1 %,55
309 HCOLOR = 0: HPLLOT C2%,95 TO C2%,105: HCOLOR = 3
310 C2% = D2%(J) - H
315 IF C2% < 0 THEN C2% = 0
316 IF C2% > 279 THEN C2% = 279
317 HPLLOT C2%,95 TO C2%,105
319 HCOLOR= 0: HPLLOT C3%,145 TO C33%,155: HCOLOR= 3
320 C3% = D3%(J) + H
325 IF C3% < 0 THEN C% = 0
326 IF C3% > 279 THEN C3% = 279
327 HPLLOT C3%,145 TO C3%,155
370 NEXT K

```

*Listing 6: A computer such as the Apple II which has high-resolution graphics capabilities greatly simplifies the program originally given in listing 5. This program performs the same operations as the simulation in listing 5. The author acknowledges the assistance of Charles Faso from Computerland of Niles IL in preparing this program.*

```

380 H1 %(J) = H
390 NEXT J
400 HGR
405 POKE 49234, 0
410 FOR J = 1 TO 250
420 Y = 191 - J * 191 / 250
430 U = 88 / 280
435 H = INT (H1 %(J) * U)
440 D1 = INT ((D1 %(J) - 140) * U + 45)
450 D2 = INT ((D2%(J) - 140) * U + 135)
460 D3 = INT ((D3%(J) - 140) * U + 225)
461 C1 = D1 + H:C2 = D2 - H:C3 = D3 + H
462 IF C1 < 0 THEN C1 = 0
463 IF C3 > 278 THEN C3 = 278
480 HCOLOR = 1
490 HPLLOT D1,Y: HPLLOT D2,Y: HPLLOT D3,Y
500 HCOLOR = 2
510 HPLLOT C1,Y: HPLLOT C2,Y: HPLLOT C3,Y
520 HCOLOR = 3
530 HPLLOT H + 45,Y: HPLLOT - H + 135,Y: HPLLOT
      H + 225,Y
540 NEXT J
550 INPUT "";A$
560 TEXT
570 GOTO 180

```