PCT and Engineering Control Theory

MULTIDIMENSIONAL CONTROL

Presented at the 19th annual meeting of The Control Systems Group at Marymount College in Los Angeles, July 23-27, 2003, this paper includes a simulation of multiple control systems sharing a common environment. Each control system computes a perceptual variable which is a weighted sum of all of the environmental variables involved, The weights are selected at random. The output of each control system affects all of the same environmental variables through a matrix that is the transpose of the input weight matrix. The result is that each control system can keep its own perceptual signal close to the value of whatever reference signal it is given, independently of all the other systems. This is a "worst case" scenario; an observer seeing only the environmental variables could not tell what variable any of the control systems was controlling. Source code (Delphi) is included.

INTRODUCTION

This paper is an attempt at comparing Perceptual Control Theory (PCT) with approaches that are more direct descendants of engineering control theory and the early concepts of engineering psychology. There are many similarities, of course, since all these approaches deal with closed-loop systems and necessarily are subject to the same laws that apply to closed causal loops. But the approaches differ in more than terminology and emphasis.

In some ways, PCT is rightly considered old-fashioned by some control engineers. This is because it focuses exclusively on negative feedback control systems as they were understood 50 years ago, and says little (outside some very preliminary proposals) about systems that learn and adapt. The counter-argument is that in developing very broad and general concepts of control, those who promote "Optimal control" or "modern control theory" have lost sight of some of the fundamental principles that make negative feedback control a hands-down choice over their way of doing things—if we focus on the context of living organisms. Engineers can build systems with components that have capabilities far beyond what a living system can accomplish with nerve and muscle. It is not unreasonable to think of calculating the actions a machine must produce in order to have a specific effect, even if those calculations require complex mathematical operations carried out at very high speed and with very high accuracy, and even if the actuators with which the deduced actions are carried out must be reliable and accurate to two or three decimal places. It is reasonable for an engineer to build into his control devices precise knowledge about the laws of nature, and complete knowledge of the kinds of perturbations and variations to which the system will be subject.

But to assume that a living organism can produce similarly extensive, precise, and reliable calculations and actions is simply unrealistic. It is unrealistic to assume that organisms come into being equipped with a full engineering education, including not only higher mathematics but practical knowledge about how things work and complete understanding of physical principles. If accurate control by organisms depended on such assumed abilities, we would have to conclude that organisms cannot control anything.

But organisms do control, and they do it very well. In many regards they do it far better than any man-made device has so far been able to do. Of course for simple processes involving simple physical variables, machines can control faster, more accurately, and in more difficult circumstances than any human system could handle. But as soon as we consider more interesting levels of control, machines flunk out completely. They can't even detect the variables that human beings control at higher levels of organization, such as the unique configuration of a face being drawn on canvas, or the happiness in a baby's voice, or the sense of satisfaction one can achieve by singing in tune, or in harmony, or with expressiveness, or at Carnegie Hall. They can't even perceive shapes, so they can't turn a piece of sculpture or a vase of flowers to the best angle for viewing. They can't maneuver a fork to pick up a load of spaghetti. If they could figure out how to put shoes on, they couldn't tie the laces.

What enables human beings and other organisms to do things like these is that they can perceive the world in many ways from the simple and concrete to the general and abstract, ways that no artificial device has so far been able to perceive. They can learn what states of these perceptions are preferable, and they can learn to act on the world in such a way as to bring perceptions to the preferred states and keep them there. And they do this *without* understanding the physics of the environment or of their own bodies, *without* being able to perform complex calculations of the actions required to have specific effects, *without* the ability to make predictions more than a few seconds ahead, or with an accuracy of more than five or ten per cent. They manage to build and drive cars, erect tall buildings, paint eye-deceiving pictures, balance and walk gracefully on two legs, and create countless other states of the world that would not exist without human control actions. Yet they do this using machinery that is crude, even sloppy, by engineering standards.

Obviously they must operate according to some principles that are not visible in the way control theory is normally presented to our view by engineers. PCT is an attempt to find those principles. It is possible that some day a reconciliation will be found between PCT and the styles of control theory in vogue today, but if we want a start on understanding organisms as control systems, PCT will show us the pay dirt immediately, while the engineering approach, in its current incarnations, is simply inappropriate.

CONCEPTUAL AND PRACTICAL DIFFERENCES

Even "classical" control theory has versions that differ importantly from PCT. We will focus on them, then approach some subjects closer to modern control theory. Engineering psychology began in the days of classical control theory, but it adopted some conventions that have resulted in a considerable divergence from the paths taken by PCT (which began to develop in the same decade).

The engineering-psychology view of control

This view of control processes can be illustrated by a traditional diagram of a "pursuit tracking" task. As engineering psychologists first conceived this task in the late 1940s, the person was represented as a "Human Operator" situated between a display and the joystick, as in Fig. 1:

In this task a person sits before a display screen on which is seen a moving target and a cursor that is moved by a joystick that the person manipulates. According to this diagram, what the person sees on the display is the tracking error, the distance between the cursor and the target. The "control" or "control variable" would be U, the state of the arm and muscle that move the joystick. The "output" would be the position of the cursor (fed back to appear on the display) and the "input" would be the position of the target. What the Human Operator perceives would be the tracking error.

The overall purpose of a good control system, as represented under these premises, is to make the "output" match the "input" despite disturbances, noise, and modest changes in environmental parameters (parameters of the "plant"). This is, appropriately

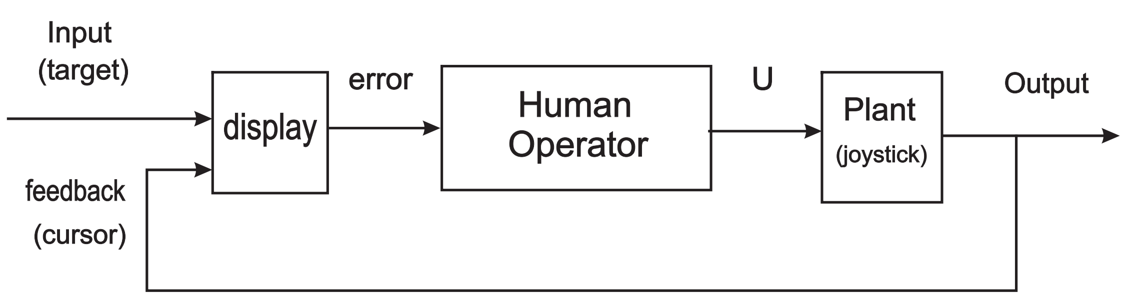

Figure 1. *Conventional diagram of pursuit tracking task, ca. 1950*

to the concerns of engineers, a user-oriented view of a control system. The "output" is the effect that the user wishes to have brought under control, such as a temperature, a position, a speed, a light intensity, or a chemical concentration. The "input" or reference or set-point is the means of adjustment available to the user for setting the desired state of the output. The feedback path is of no direct concern to the user, since it is merely the means for making the output a reliable function of the input, and as Norbert Wiener said, "making performance less dependent on the load." The user doesn't care how the output is caused to be in the desired state—whether by feedback effects inside the control system or simply by careful adjustment of the connections going directly from input to output, or some combination of the two approaches. These would all be classified as "control" processes.

This is very much in line with the general view of "modern control theory."

The PCT view of control

In PCT the same pursuit tracking setup could be drawn in a similar-looking way, but with meanings that are quite different:

If the intended distance is zero (cursor on target), Fig 2. is functionally identical to Fig. 1, and the mathematical description of the whole system would also be the same. However, if the person is trying to keep the cursor a fixed distance to one side of the target, as a hunter aims his sights ahead of a moving target, the intended distance would be nonzero, and if the cursor were then on the target, that would constitute an error. The diagram of Fig. 1 can't represent this case, because it assumes that any nonzero distance between the cursor and target must be an error.

Whatever the setting of the intended-distance signal (this signal is called the "reference signal" in general), the action of the Human Operator will be such as to reduce the distance error to zero: the perceived distance from cursor to target will then match the intended distance.

In this model, the target and cursor positions are both perceived, but now neither one of them tells the Human Operator what relationship between them is to be sought. Those perceptual inputs merely report the current state of the display, without indicating anything special about any one state that might exist. The aspect of the input variables that is to be controlled is determined by the "input function" of the control system, which computes some function of the inputs, here "distance between," and reports it as a perceptual signal. The state of this perceptual signal that will be the aim of control is set by the reference signal inside the Human Operator; it could indicate that the cursor is to be to the left or right of the target

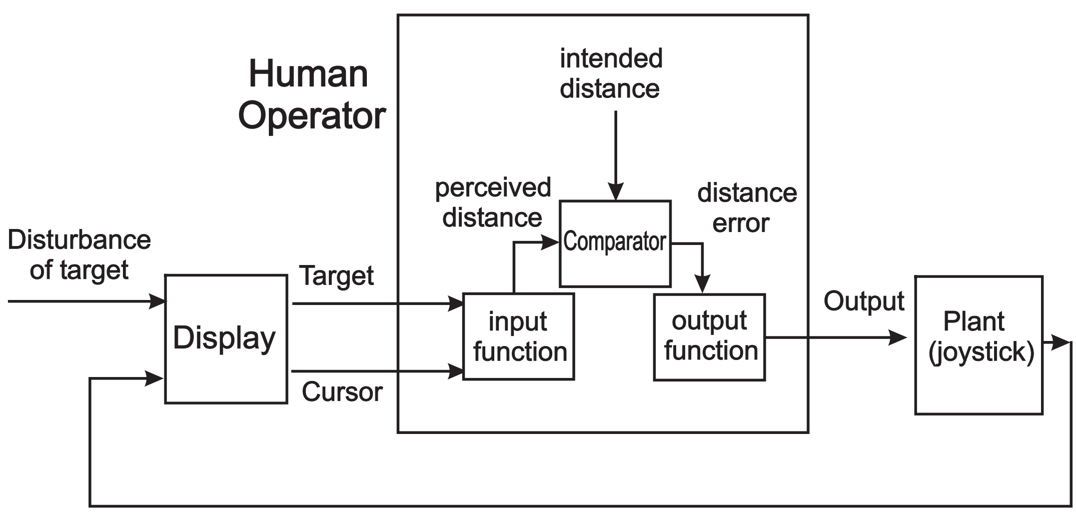

Figure 2. *PCT diagram of a pursuit tracking task. Target and cursor positions are separately perceived. Distance is computed by the "Input Function." The error is the difference between the perceived target-to-cursor distance and the intended target-to-cursor distance which may or may not be zero. "Output" is a measure of the position of the hand that holds the joystick.*

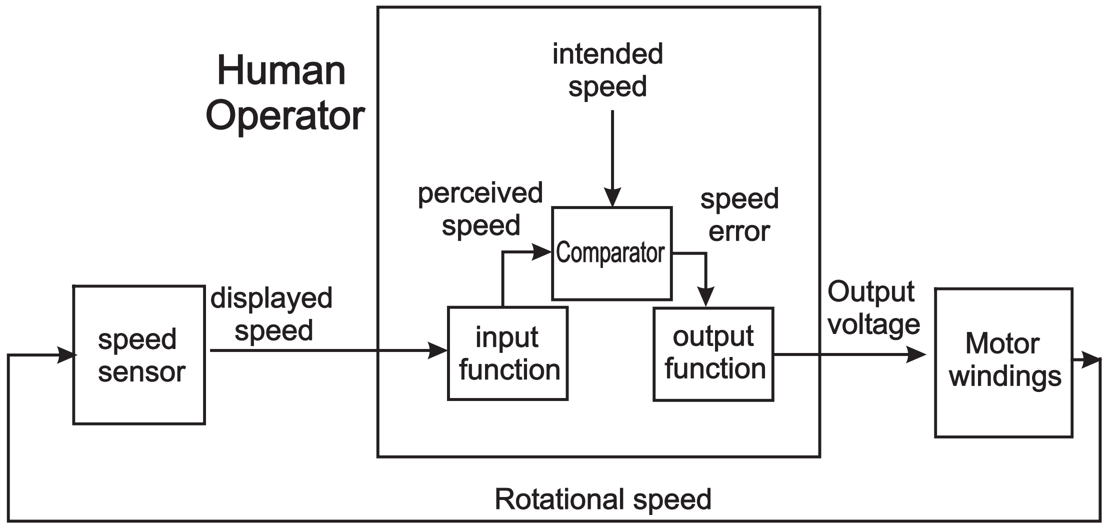

Figure 3. *PCT diagram of a motor speed controller. The Human Operator observes the display on a speed indicator, and adjusts the voltage applied to a motor to make the perceived speed match the intended (reference) speed. Note that the only sensory input is the displayed speed. The intended speed does not come from a sensory input.*

by any amount, on the target, or varying back and forth between left and right (if the reference signal varies between positive and negative values).

In the standard engineering-psychology diagram of a control system, essentially the same as Fig. 1, only one place in the diagram is labeled as an input. But there is another place where an input device must exist, although it carries no label to alert the viewer. That is the place where the feedback arrow branches off of the output arrow. The so-called output is some variable in the environment that is being maintained in a specific state: the variable could be the rotational speed of a motor, and the specific state might be 1000 revolutions per minute. For a feedback signal to exist, however, to be joined at the system's input by a "command" input signal, there must be something that can detect the rotational speed of a motor, represent it somehow, and send the representation to the input of the control system. That would presumably be an electrical signal generated by a sensor of some sort, the signal being proportional to rotational speed but of course not itself being a rotational speed.

In the case of speed control, we are not trying to create a relationship between the rotational speed of a motor and the rotational speed of something else, but simply to keep the rotational speed at a constant 1000 RPM (or whatever speed we desire). In the tracking experiment, there were two perceptual inputs, the target position and the cursor position, and the system controlled a relationship between them. In the speed control system, there is only one perceptual input, and the system simply controls its magnitude. The appropriate PCT diagram for the speed control system would be that of Fig. 3.

In Fig. 3, note that the reference or intended speed does not come from a sensory input. The only sensory input comes from the speed sensor, which reports what the actual speed is at every moment, but does not supply any additional information about what the speed should be. The signal that indicates what the speed should be (when the error has been reduced to zero) must come from somewhere else. That subject will come up later.

Suppose we wanted a control system that would keep the temperature of the motor constant. Fig. 4 shows how it should look:

Fig. 4 should look familiar: it is the speed control diagram with the word "speed" replaced everywhere by the word "temperature" or "temp." Now the Human Operator is reading a displayed motor temperature, and varying the voltage applied to the motor to keep the motor temperature at an intended level. The motor will spin at various speeds while this is happening, but with respect to this control

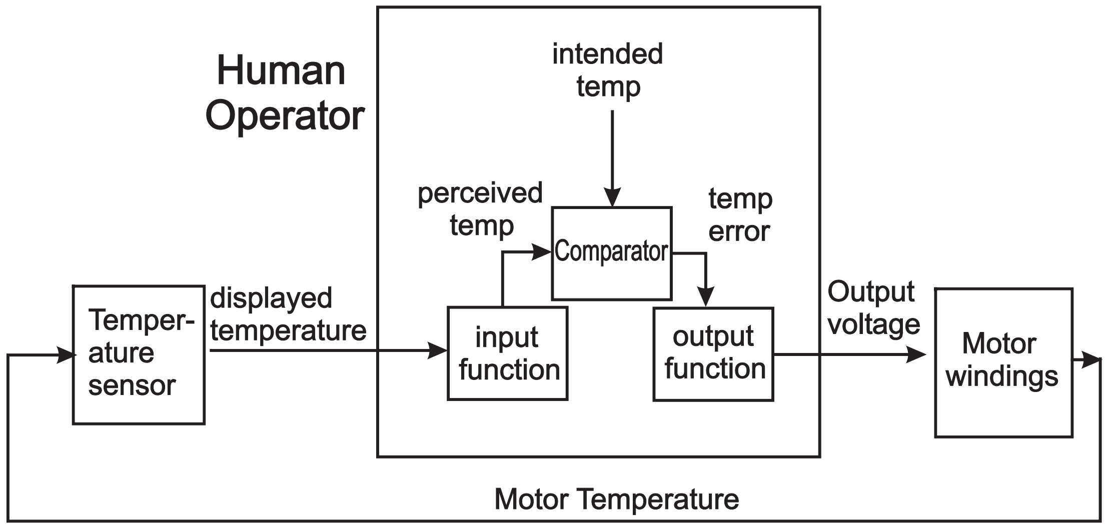

Figure 4. *PCT control system for controlling temperature of a motor. Blank box is comparator*

system, the rotation of the motor shaft is irrelevant (unless it cools the motor). This system perceives and controls only the temperature of the motor; the rotational speed of the motor is uncontrolled. If a fan is turned on and cools the motor somewhat, the voltage will be increased as soon as the temperature has dropped by the minimum needed to detect a temperature drop, and the temperature drop will cease. The same increase in voltage will, of course, cause the motor to spin faster. So the temperature remains controlled very near the desired level, while the speed varies according to how fast the air from the fan is moving and carrying heat away.

Obviously, it is not necessary for the Human Operator to know what actual physical variable is behind the scenes. There is a display which shows a magnitude for the variable being controlled, and an actuator—a knob or pedal or slide-adjustment—that can affect the reading on the display via the environment. The Human Operator can use the actuator to bring the number on the display to any desired reading, and if for any reason it changes, move the actuator to bring the reading back to the same level. It's not necessary to know what caused a change in the reading, either, because regardless of the cause (within reason), moving the actuator the right way will restore the desired value.

Neither is it necessary for the Human Operator to know how the output action works to affect the input. For either the speed or temperature controller, moving the actuator one way makes the motor speed up and its temperature increase; moving it the other way makes the motor go slower and lowers its temperature. The only choices for the actuator are more or less movement of the actuator, one way or the opposite way. The only choices for the perceived effect are that the indication on the sensor gets larger or smaller.

When control is one-dimensional as it is in all these examples, discovering what to do to achieve control requires very little intelligence. There are dynamic considerations, but if the whole problem of control behavior is approached in the right systematic way, the primary problem of dynamics comes down to adjusting the amount of action to be produced by a given amount of error.

"The right way" involves two concepts: controlling one dimension per control system, and stacking up multiple levels of independent systems to use sets of modular control systems as the means of controlling more generalized variables: a multivariable, multiordinal architecture. This approach is specifically appropriate to the analysis of living control systems, in which can be discerned not only multiple processes of control occurring in parallel, but multiple levels of control, stacked physically and organizationally into an extensive and complex hierarchy of control.

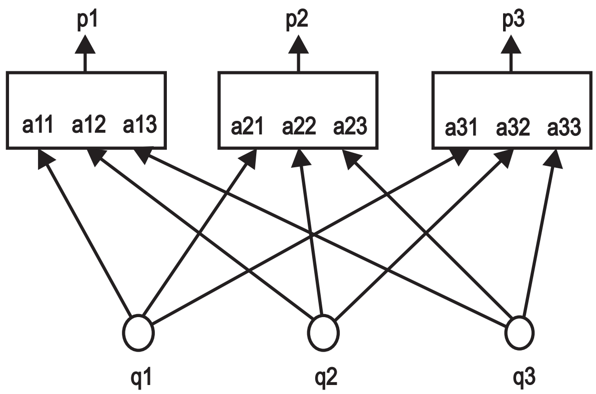

Figure 5. *Three perceptual signals derived simultaneously and in parallel from the values of three environmental variables.*

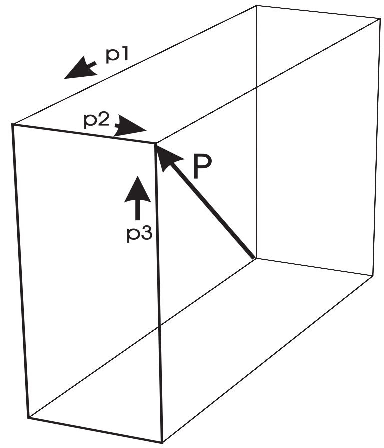

Figure 6. *"Directions" in space with axes parallel to p1, p2, p3. The perceptual signal p1 in Fig. 5 represents the magnitude of P. Control requires that the output weightings be opposite and proportional to a11, a12, and a13 (Fig. 7)*

DIMENSIONS OF CONTROL

The focus now switches to the issue of multidimensional control. PCT is often presented in terms of a single simple control loop in order to show the basic relationships of circular causality. Such presentations have been taken by those following other approaches to mean that PCT considers an organism to be a single control system. This is far from the case, as we will now see.

Multidimensional control, in traditional engineering approaches, is often or even normally treated in terms of the compact mathematical notations of matrix algebra. A set of output variables called **P** (bold-face indicating a vector or matrix) is produced by "premultiplying" a set of input variables **Q** by a matrix **A**, like this:

P = A * Q

Suppose that the matrix **A** consists of a set of coefficients a11 to a33, and **Q** and **P** each consist of three elements q1, q2, q3 and p1, p2, p3. The equation above is then expandable into

```
  P      =          A           *   Q
 |p1|    |a11  a12  a13|        |q1|
 |p2|  = |a21  a22  a23|        |q2|
 |p3|    |a31  a32  a33|        |q3|
```

Expanding even further by the conventions of matrix notation, we find that this matrix equation turns into three simultaneous algebraic equations ("simultaneous" meaning that all three equalities have to hold true at the same time):

```
p1 =  a11q1 + a12q2 + a13q3
p2 =  a21q1 + a22q2 + a23q3
p3 =  a31q1 + a32q2 + a33q3
```

So that is the real meaning of the initial matrix equation above. The matrix equation is a shorthand way of writing three (in this case) simultaneous equations. A model which is described in terms of matrices is not usually meant to imply that the physical system being described actually performs matrix operations using the notation above. What the system actually does is better represented by the set of three equations, for the real system must perform all the additions and multiplications implied by the matrix notation—as, indeed, a computer must do when the programmer .tells it to perform a matrix operation.

Suppose we have an environment with three variables in it, q1, q2, and q3. Suppose, too, that we are going to build three control systems that will each perceive and control something about these three variables. By "something about" it is meant, in this case, that each system will sum sensor readings of the three variables according to a different weighting scheme. We can, in fact, label three perceptual signals p1, p2, and p3, and use the three equations above to show how each signal's magnitude would depend on all three of the environmental variables, q1 q2, and q3. Fig. 5 illustrates what has been said so far: this diagram is a picture of the meaning of the three simultaneous equations above.

The variables q1, q2, and q3 can change independently; any one of them can change without any change in the others. They can thus be represented as measured along three axes of a three-dimensional space, as in Fig. 6. A block with sides equal to the three variables has a diagonal measured from the origin that points in a particular direction, as shown. This diagonal is the resultant of vectors along each axis.

For each of the perceptual signals p1, p2, and p3, we can establish a corresponding reference signal r, giving us r1, r2, and r3. Then we will need a way of determining the errors between r1 and p1, and so forth: comparison processes will then yield three error signals e1, e2, and e3. To close the three control loops, we then must then connect e1, e2, and e3 back to the environmental variables q1, q2, and q3 in such a way that the three error signals will be reduced to zero. If that happened, we would find p1 = r1, p2 = r2, and p3 = r3. This combination of three control systems could then affect the environment so that three perceived aspects of it can be controlled simultaneously to match an equal number of independently adjustable reference signals. More dimensions, obviously, would be possible.

Control requires that each output signal affect all three environmental variables through suitable output weightings in such a way as to oppose any change in p1, p2, and p3. As an example, suppose p1 increases for any reason. The outputs should all change in the direction that will make q1, q2, and q3 change so that when the changes are multiplied by the relevant weights (a11, a12, and a13), the effect on p1 will be to decrease it, an effect opposed to the original change. The same should happen for changes in p2 and p3. This proves to be extraordinarily easy to do. All that is needed is for the output weighting matrix **W** to be the "transpose" of the input matrix **A,** or reasonably close to it (rows of **W** become columns of **A**)**.** Then the values of the variables q1, q2, and q3 are simply the sum of effects from the disturbing variables and the outputs:

$$\mathbf{Q} = \mathbf{W}^*\mathbf{O} + \mathbf{D}$$

With the proper weighting matrix **W**, this multivariable control system will bring each perceptual signal p1, p2, and p3 to a match with its respective reference signal, r1, r2, or r3. This will happen even if the input matrix weightings are selected at random, so the perceptual signals interact (the **W** matrix still has to be set to be the transpose of **A**). It will happen whether there are 3 input variables or 100. When there are very strong interactions among the perceptual signals, meaning that the three directions defined by the three sets of weights are not orthogonal (at right angles to one another), convergence to a final state can be quite slow and the magnitudes required of the three outputs can be enormous. But if the input weights happen to define nearly orthogonal directions, convergence is very rapid and the output magnitudes are at a minimum.

The control of perception

This multidimensional control system controls not the set of variables in the environment, q1, q2, and q3, but the values of three perceptual signals p1, p2, and p3, which are individually brought to matches with their respective reference signals. Each perceptual signal is a weighted sum of all three environmental variables, so there is no one-to-one correspondence between any perceptual signal and any one environmental variable.

The perceptual signals do not represent any single environmental variable, but only something about all three variables taken as a set: an "aspect" of the environment but not (necessarily) a tangible physical thing. It may seem that there would be little point in controlling variables that have no objective existence in the physical world, but that is another difference between PCT and engineering control theory. The engineer has to control something that can be seen and measured in the environment, so the user of the control system will agree that it is being controlled. But in PCT, the only opinion that counts is that of the control system. The only reality the living control system can experience and control is the reality reported by the senses and the neural networks that combine sensory experiences into perceptual signals.

To be sure, organisms would not control perceptions unless doing so proved beneficial to them in some way; either a short-term way or a way conferring fitness to resist forces of natural selection. But this does not imply that each controlled perception must have a known and direct relationship to benefits. For example, consider a human perception that is derived

For Figure 7 see next page.

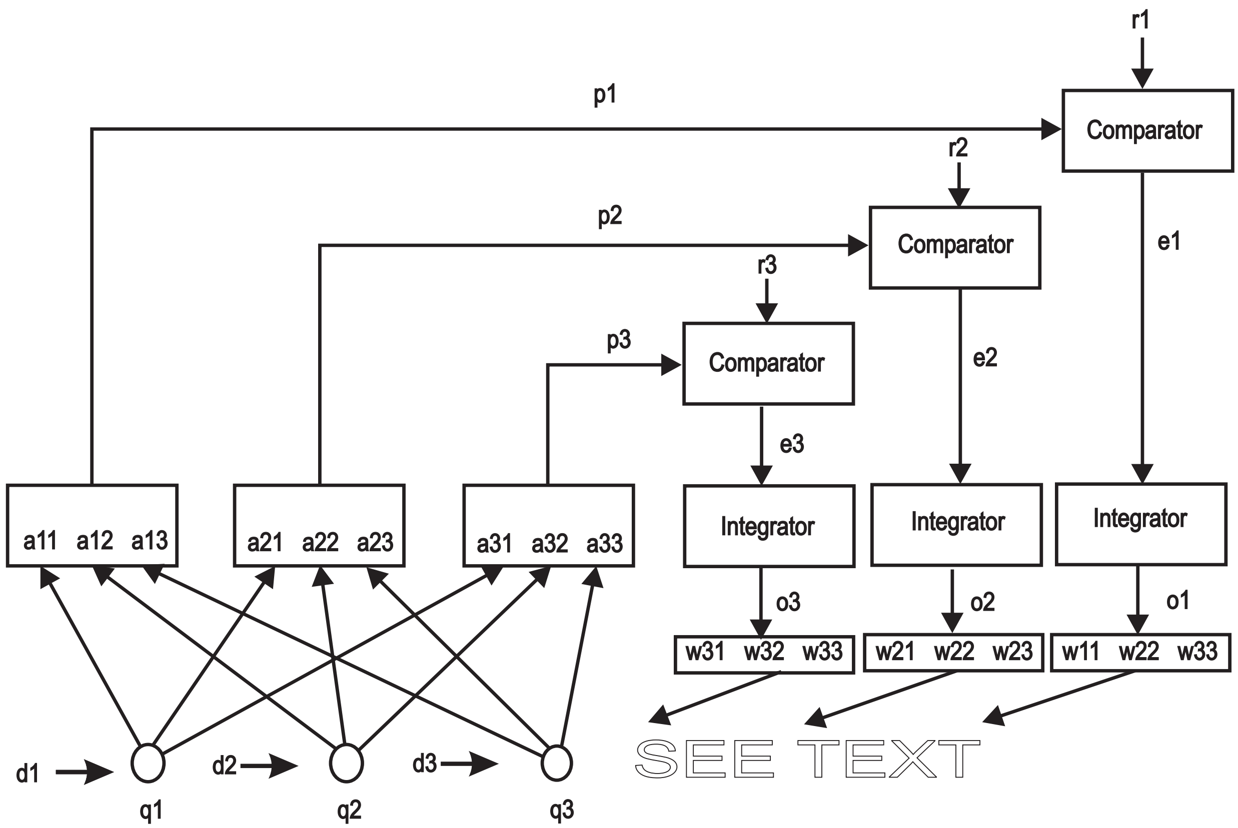

Figure 7. Three interacting control systems. Three perceptions each derived from three environmental variables are controlled relative to three independently-adjustable reference signals. Each error signal is integrated to produce an output signal. The values of the three environmental variables are determined by the sum of the weighted output signals and independent disturbing variables. Thus, for example,

$q1 = w11 * o1 + w21 * o2 + w31 * o3 + d1.$

from two physical input variables: temperature and humidity. In the summertime, weather reports include an estimate of this perception, called the "discomfort index." Human beings experience discomfort when this index rises above a certain level, but it is neither temperature nor humidity that must be high or low. Only a nonlinear weighted sum of temperature and humidity seems to correspond to the critical variable here. That variable has no external physical significance; it is significant only as it is experienced by human beings.

Despite its lack of physical reality, human beings control that variable: if the perception of discomfort is sensed as being above some reference level, people take action to lower it, by lowering the temperature of their environments, or the humidity, or both.

The diagram of Fig. 7 is claimed to be a control system. Would such a system, in fact, control the three perceptual signals in the manner stated? That question can actually be answered, by viewing Fig. 7 as the blueprint for a simulation, the next subject to be taken up.

SIMULATING MULTIDIMENSIONAL CONTROL

In standard engineering design procedures, the equations representing a control system are, where possible, solved by using analytical methods that predict exactly how such a system would behave. But exact analytical or symbolic methods are generally not usable with real systems. The reason is that real systems tend to have nonlinearity and noise, and are subject to perturbations having forms that can't be predicted. So what is normally done is to find idealized descriptive equations that are as close as possible to the forms actually required, but having the useful property that analytical solutions exist for them. While the analytical approach does give exact and general solutions, they are seldom solutions of the equations that come closest to describing the actual physical system under investigation. Advantages of the exactness of symbolic methods may well be lost in the approximations that must be made in setting up tractable, rather than correct, equations, and the relevance of any general theorems found to apply to the mathematical solutions may also be suspect, because, again, the solutions are not solutions of the equations that actually apply. It doesn't take much of a change of form to render an exact general theorem invalid—for example, the Pythagorean Theorem applied to a triangle that is *almost* a right triangle, or that is drawn on a surface that is *almost* flat. Applying the theorem literally can be as misleading as predicting the amount of water a spigot deposits in a bucket having known dimensions and a bottom that is *almost* free of holes.

Fortunately there is an alternative to the analytical equation-solving approach, called "simulation" (also known as "solving numerically"). Simulation begins as the analytical approach does, with representations of relationships among system variables in the form of mathematical equations. It is not necessary, however, to discard the most accurate descriptive equations in favor of approximations that can be solved, because in a simulation equations are never "solved." Instead, they are *evaluated*, a much simpler process.

Consider the equations given for the 3D control system above. In matrix/vector form, they are

P = A*Q
E = R – P
O = k*integral(**E***dt)
Q = W*O + D

The analytical approach would treat these as simultaneous (differential) equations and would result in a general equation for the solution. Unfortunately, in addition to any approximations in the basic equations, the disturbances represented by **D** are of unknown form, so to get a solution it would be necessary to use fictitious forms of **D** that do permit analytical solutions: square or triangular waves, impulses, or sums of a few sine and cosine waves.

In a simulation, however, the above four equations represent all the mathematics we have to deal with. The first equation says **P = A*Q**, which tells us to do three computations starting with initial values used for q1, q2, and q3, and the constants defining matrix **A** (the asterisk * is computerese for an explicit multiplication sign):

p1 = a11*q1 + a12*q2 + a13*q3
p2 = a21*q1 + a22*q2 + a23*q3
p3 = a31*q1 + a32*q2 + a33*q3

The next equations provide three more computations that use the values of p1, p2 and p3 just calculated, plus the values of three reference signals which can be set to arbitrary numbers:

e1 = r1 – p1
e2 = r2 – p2
e3 = r3 – p3

Then come the three integrations:

o1 = k*integral(e1*dt)
o2 = k*integral(e2*dt)
o3 = k*integral(e3*dt)

And finally we have the equations for the q's:

q1 = w11*o1 + w12*o2 + w13*o3 + d1
q2 = w21*o1 + w22*o2 + w23*o3 + d2
q3 = w31*o1 + w32*o2 + w33*o3 + d3

These computations take us from the starting values of q1, q2, and q3 to the next set of values of the same variables a short time later. We can calculate our way around this loop again and again, building up a picture of the way all the variables change through time.

Integration in the digital domain

The "integral" step requires a brief explanation. An integration is a summation, so the output variables o1, o2, and o3 are sums of the values of e1*dt accumulated every time the set of equations is periodically evaluated (every "iteration"). The argument e indicates how fast the sum grows or becomes less negative when e is positive or shrinks or grows negatively when e is negative. The value of dt represents the fraction of a second over which the accumulation takes place, so e*dt is the change in the value of o that takes place over the time interval dt. The actual integration is performed by adding e*dt to the old value of o to obtain the next value of o:

o(new) = o(old) + e*dt

The value of the output o is the sum of all the positive and negative changes of various sizes that have taken place since the Big Bang, or more practically since the last time o was measured.

This is "Euler integration," and for closed-loop systems is accurate enough while being very fast to compute. More exact integration methods exist, but are not usually needed except in simulations of systems without negative feedback,

The fundamental theorem of the calculus states that when any continuous waveform is summed in short segments, the true amount by which the sum changes on each step is approximated by the average of the values at the start and end of the interval times the distance between the samples, which is dt. As the size of dt becomes smaller and a given interval is broken up into smaller and smaller slices, the approximation comes closer and closer to the exact value, becoming equal to the exact value just as the size of dt shrinks to zero and the steps become a continuous curve. The exactness of the value as dt approaches zero is the point of the fundamental theorem. The integral of x*dt is $x^2/2$ — exactly.

In practical terms, we can say that the integral of o changes by the amount e*dt on every iteration of the simulation, to an accuracy that depends on the size of dt. If we make dt represent 1 second, for example, the result will be grossly inaccurate if e is varying significantly twice per second. But with the same speed of variations in e, the sum would be very accurate if we made each iteration represent 0.000001 second—one microsecond. The fundamental theorem of the calculus tells us that we can always pick dt small enough to reduce errors to any size we want (in most cases—mathematicians are quibblers).

Of course it would take much longer to calculate one second's worth of behavior at one million calculations per second than at one calculation per second. It is probably never necessary to use a dt as small as one microsecond in simulating a physical system like an organism. On the other hand, to get a close representation of a real system like the system that controls the position of a human arm, dt must certainly be no larger than 1/30 of a second, and results would be more accurate if dt were reduced to 1/60 second or smaller (conveniently, the rate at which computer screens of moderate resolution are refreshed). The size of dt is chosen so it is as small as necessary to get accurate integrations on the relevant time scale, while not unduly slowing the calculations,

When simulations are done with a real analog computer that employs continuous physical variables, there is no dt to worry about. But simulations are most easily done at present by using digital computers in the way suggested here, emulating a true analog computer by calculating changes in system variables at very short intervals and, for integrations, using values of dt that are short enough to make the inaccuracy verifiably unimportant.

A DEMONSTRATION OF MULTIDIMENSIONAL CONTROL

Accompanying this file is a Delphi program called "MultriControlPrj.exe." This program carries out the simulation described above, with any number of control systems from 3 to 500. Each perceptual signal is a different weighted sum of 3 to 500 environmental variables. The output of each control system affects all 3 to 500 environmental variables through another matrix of weights. The output weighting matrix is the transpose of the input weighting matrix. On each run, the input matrix is filled with a new random assortment of weightings between –100 and 100, and the output matrix is set to the transpose of the input matrix (rows become columns). Each control system is given a randomly selected reference signal between –750 and 750 units.

The screen initially shows the values of all of the perceptual signals (red dots) and reference signals (white circles), Also shown as purple dots are the values of all environmental input quantities. Each control system is represented by these three colors of dots, arranged along one of 3 to 500 vertical axes (zero is the position halfway up the screen). Many of the dots representing perceptual values are initially off the screen.

After sliding the pointer to select the number of control systems, the user can start the program by clicking the start button (the "pattern" radio buttons can be used before and during the run; the default pattern is random). The white reference signals (all but three of them) remain stationary, while the red perceptual signals begin to change, rapidly at first and then more slowly as they come closer to their respective white reference signal values. In three of the systems, the reference signal changes in a sinusoidal pattern between positive and negative 1000 units, so you can see the perceptual signals being made to track their respective reference signals. The tracking lags somewhat since all the control systems use integrations in their output functions. Also, because of mutual influences, the red controlled perceptions do not quite match their white reference circles even when the reference levels are constant.

The purple dots representing the environmental quantities move, too, but they do not come to positions corresponding to either the red perceptual variables or the white reference signals. This makes sense, since no one perceptual signal depends on the value of only one environmental quantity; in fact, each perceptual signal depends on all of the environmental quantities.

Eventually you will see many of the red dots and white circles and most of the purple dots moving slightly as the three reference levels go up and down with their perceptual signals following them. Remember that the purple dots represent Reality, the states of the variables in the environment of the control system, while the red dots represent the variables that are perceived—a very different matter, obviously. In order to control the red dot that tracks a changing reference signal, the associated control system affects all of the purple environmental variables by some positive or negative amount. Some move a lot, some only a little or not at all. Some move the same way the controlled perception changes, some the opposite way. There are small effects on most of the other controlled perceptions, the red dots, but the larger the number of interacting systems, the closer the average effect on any one environmental quantity comes to zero. With 500 systems, the purple dots fall in a much narrower band around zero than when only 30 systems are used. In any case, it would be very hard to see any relationship between the behavior of the changing controlled perception and the behavior of any one environmental variable.

Since the initial perceptual weightings are selected at random, there is no guarantee that the perceptions are independent of each other. Thus there can be considerable conflict between some pairs of control systems. Also, the weightings can add up to a small or a large effect on the perception. For both reasons, some control systems will control more accurately than others. This, indeed, is one basis on which reorganization could occur: the weightings for a given control system's input function could be randomly shuffled until the control error is minimized. This is a promising topic for further investigation.

The main requirement to make this simulation work with any number of control systems is that the output weighting matrix must be the transpose of the input weighting matrix. The question, of course, is how it might come about in a real system that these two sets of weights bear this relationship. We can't expect the weightings to be transferred from the sensory side of the nervous system into the motor side, a process that would require some system that can sense synaptic weightings and copy them in just the right way from one set of neurons into another set some distance away. That sort of operation is easy to program into a computer, but it's difficult to imagine how it could be carried out in and by a nervous system.

One possible alternative is simply that the output weightings are varied until the errors in all the control systems are as small as possible. The "reorganization" scheme of PCT might be able to do that. Or there might be some systematic algorithm that could accomplish it, still without demanding that the nervous system perform higher mathematics. We know that the transpose relationship is the one that will give minimum possible error with the control system design that is used here, so we can judge how well various algorithms work by seeing how closely they manage to bring the output weights to the transpose of the input weights. Investigations of this sort remain to be carried out.

Probably the most interesting implications of this demonstration have to do with epistemology, the relationship of the perceptual representations to quantities in the physical world outside the control systems. As this demonstration is set up, the perceptions are truly random representations of the set of all environmental variables. The fact that control of each perception relative to an independently chosen reference level is possible shows that this arrangement works in the worst possible case, where perception bears no relationship to any actual forms, any "natural kinds," in the environment. Of course if the weightings were such as to produce perceptions corresponding to actual organized entities in the environment, the model would still work the same way, granted that the output matrix comes to resemble the transpose of the input weights.

In the more customary approaches to the problem of multidimensional control, theoreticians have imagined that the brain is somehow able to compute the output signals that would be required to make all the environmental variables match their desired values—and do so prior to taking any action. The mathematical means of doing this is available through calculating the inverse of the output matrix (which of course must be known) to deduce the set of command signals that would cause the environmental quantities to come to the required state.

The present demonstration brings out several problems with this proposition. First, the equipment that the brain would have to possess in order to calculate the inverse matrix is orders of magnitude more complex than what is required for control in the manner shown here. Both approaches require that there be an output matrix, but the PCT approach achieves control without ever computing any inverses.

Second, the customary approach assumes that it is the set of objective variables in the environment that must be controlled, an idea that fails to take into account the difference between the actual state of the environment and its perceived form and state. This epistemological question is in fact begged—it is assumed that whatever calculates the inverses of the output effects somehow knows the identity of the actual variables in the environment, independently of perception. While this may be true (in a manner of speaking) for an engineer building an artificial control system, it is certainly not true of a brain learning to control what it can sense of the world around it. There is no one who can tell the brain what is really out there. It can work out possible world-models, and check its conclusions, only by examining its own perceptions.

An important unanswered question is how a system like this could arrive at a non-random set of input weightings that would provide it with perceptions having objective meaning in the world outside (if indeed that happens). We can envision some sort of repetitive process by which the input weightings are varied, but the critical question is what will stop the variations, and when. What is the criterion that tells the system that one form of input matrix is a little better than a different form? Better for what? For ease of control (least effort)? For greatest independence of one perception from others? For maximizing benefits and minimizing costs to other systems in the body? The possibilities are numerous, and remain to be explored.

HIERARCHIES OF MULTIDIMENSIONAL CONTROL SYSTEMS

It is possible to imagine an indefinitely large set of control systems of the kind described above. The result would be control of a large set of perceptual signals derived from environmental quantities through sensing and weighted summation. There would be a set of reference signals, one for each controlled perception, which can be set to arbitrary values. What sets them, according to PCT, is a set of higher-order control systems (if they are not always set to zero).

Obviously these higher-order control systems experience an "environment" in much the same way the lowest level does: as a set of input quantities. Now the quantities are lower-order perceptual signals which are themselves weighted sums of environmental quantities, but the higher system knows nothing of that. The first-order perceptual signals are the sensory environment of the second-order systems.

Similarly, each higher-order control system acts on its environment by sending output signals into it, just as a lower-order system does. Now, however, those output signals do not reach physical actuators. Instead, they affect the reference signals of the lower systems.

We can now imagine exactly the same sort of situation we had before, with lower-order perceptual signals substituting for the environmental quantities sensed by the lower system, and lower-order reference signals substituting for the places where the outputs of the control system act. We could in fact set up a second layer of control in which the perceptual variables controlled were weighted sums of all the perceptual signals of the next lower order. Each second-order control system would form a perceptual signal through an input matrix, and it would add its contribution to each lower-order reference signals through an output matrix that is the transpose of the input matrix. In principle, then, we would have a second layer of control using the whole first layer as its means of control.

Just as we can have an indefinitely large array of control systems at a single level, so we can have, in principle, an indefinitely large number of orders of control. In both cases, the real system has to tell us where to stop. In the real system, we can be sure, there is neither an infinite number of control systems at a given level, nor an infinite number of levels.

The Perceptron is a multi-layered device in which weighted sums of sensory input signals are combined in multiple layers to produce an ultimate set of output signals. During learning, the weightings are altered on the basis of discrepancies between the actual outputs and the outputs that are somehow decided to be the correct ones. For the most part, Perceptrons in behavioral models are treated as the input part of a stimulus-response system, but there is no reason in principle why they could not be part of a control hierarchy, with closed loops at each level and control being carried out in terms of continuous variables.

However, it is not self-evident that weighted summation is the only possible type of computation through which higher-order perceptions can be derived from lower-order ones. In developing PCT, I spent considerable time looking for what might be hierarchically-related types of controlled perceptions, and eventually arrived at a list of 11 types. Only the first two, intensities and sensations, would seem amenable to being derived through weighted summation. The fourth level, for example, was proposed to be concerned with transitions, which is to say time derivatives, and no amount of weighted summation of scalar variables will produce a signal indicating time derivatives. If that is indeed a level of perception and control, then at the very least the perceptual input function would require the ability to compute derivatives of input signals, or perhaps sums of weighted derivatives. At the sixth level, what appears to be perceived are relationships among lower sets of variables, which is to say the forms of functions exemplified in the way sets of lower perceptions behave together. Weighted summation would not seem even slightly suited to computing that sort of perception. Higher still in the hierarchy we seem to have discrete variables, such as logical variables and category names—nothing at the higher levels of experience would seem a suitable candidate for computation by weighted summation.

It is possible to set up control-system simulations for single levels of perception. Tracking tasks, for example, involve control of a relationship between a target and a cursor that is supposed to stay on it, or near it. However, this requires assuming that a perceptual input function exists and produces a signal corresponding to a measure of the relationship that we can compute in other ways. We can propose that there is a control system that perceives the distance between the cursor and the target, and we can say that the magnitude of this perceptual signal corresponds to the measured or otherwise known distance between these entities in laboratory space—but we can't put into the simulation the mechanism by which this physical situation is turned into a perceptual signal.

The predictive power of such simulations can be quite impressive. We can often match the performance of a simulation to that of a real person within one percent or better over a continuous 60-second experimental run. But there is no way, in most cases, to build a two-level simulation involving either a higher or a lower system while accounting for the way the higher-level perception is derived from the lower. Those dependencies we still have to get from informal observations.

As matters stand now, therefore, we have to conclude that we do not know how perceptions of higher than order two are computed, and our only way of guessing what these perceptions are is through careful observation of subjective experience—which is to say, the only kind of experience there is. Perhaps someone else knows, but I don't.

CONCLUSIONS

The single-level multi-dimensional control system discussed here is just a beginning. The applications, extensions, and further developments of this concept will be manifold. Adding higher orders of control broadens the research and modeling possibilities even more. Not even mentioned here is the likelihood that higher-order systems can perceive aspects of system performance, and adjust not lower reference signals but the parameters of lower control systems to achieve higher performance and tighter control, or to satisfy other criteria. That possibility opens the door to new models of adaptive systems which work in ways very different from those now imagined in fields like "optimal control." PCT has spent most of its life being introduced at an elementary level, primarily to non-technical audiences. There is some reason to think it can now be carried in new directions, if enough people with the right training join the few who are already involved.