Dynamics and Cognition

16

Timothy van Gelder
1996

What is cognition? Contemporary orthodoxy maintains that it is computation: the mind is a special kind of computer, and cognitive processes are internal manipulations of symbolic representations. This broad idea has dominated the philosophy and the rhetoric of cognitive science—and even, to a large extent, its practice—ever since the field emerged from the post-war cybernetic melee. It has provided the general framework for much of the most well-developed and insightful research into the nature of mental operation. Yet, over the last decade or more, the computational vision has lost much of its lustre. Although work within it continues, a variety of difficulties and limitations have become increasingly apparent, and researchers throughout cognitive science have been casting about for other ways to understand cognition. As a result, under the broad umbrella of cognitive science, there are now many research programs which, one way or another, stand opposed to the traditional computational approach; these include connectionism, neurocomputational approaches, ecological psychology, situated robotics, and artificial life.

Is any alternative conception of the nature of cognition emerging from these programs? More generally, is there any real alternative to understanding cognition as computation? One of the most persuasive considerations favoring the computational conception has been the so-called *What-else-could-it-be?* argument. As Allen Newell put it,

> although a small chance exists that we will see a new paradigm emerge for mind, it seems unlikely to me. Basically, there do not seem to be any viable alternatives. This position is not surprising. In lots of sciences we end up where there are no major alternatives around to the particular theories we have. Then, all the interesting kinds of scientific action occur inside the major view. It seems to me that we are getting rather close to that situation with respect to the computational theory of mind. (1990, p. 5)

The central claim of this paper is that there is indeed a viable alternative. Rather than computers, cognitive systems may be *dynamical* systems; rather than computation, cognitive processes may be state-space evolution within these very different kinds of systems. If correct, this effectively disarms the *What-else-could-it-be?* argument, and advances the broader project of evaluating competing hypotheses concerning the nature of cognition. Note that these aims do not require establishing that the dynamical hypothesis is *true*. All they require is describing and motivating it sufficiently to show that it does in fact amount to a genuine alternative conception of cognition—one that is viable as a serious and fruitful avenue of research, as far as we can now tell.

A helpful way to introduce the dynamical conception is via a somewhat unusual detour through the early industrial revolution in England, circa 1788.

1 The governing problem

A central engineering challenge for the industrial revolution was to find a source of power that was reliable, smooth and uniform. In the latter half of the eighteenth century, this had become the problem of translating the oscillating action of a steam piston into the rotating motion of a flywheel. In one of history's most significant technological achievements, Scottish engineer James Watt designed and patented a gearing system for a rotary steam engine. Steam power was no longer limited to pumping; it could be applied to any machinery that could be driven by a flywheel. The cotton industry was particularly eager to replace its horses and water wheels with the new engines. However, high-quality spinning and weaving required that the source of power be highly uniform—that is, there should be little or no variation in the speed of rotation of the main driving flywheel. This is a problem, since the speed of the flywheel is affected both by the pressure of the steam from the boilers, and by the total workload being placed on the engine, and these are constantly fluctuating.

It was clear enough how the speed of the flywheel had to be regulated. In the pipe carrying steam from the boiler to the piston there was a throttle valve. The pressure in the piston chamber, and so the speed of the wheel, could be adjusted by turning this valve. To keep engine speed uniform the throttle valve would have to be turned, at just the right time and by just the right amount, to cope with changes in boiler pressure and workload. How was this to be done? The most obvious solution was to employ a human mechanic to turn the valve as necessary. However, this had several drawbacks: mechanics required wages, and were often unable to react sufficiently swiftly and accurately. The industrial revolution thus confronted a second engineering challenge: to design a device that could *automatically* adjust the throttle valve so as to maintain uniform flywheel speed despite changes in steam pressure or workload. Such a device is known as a *governor*.

Difficult engineering problems are often best approached by breaking the overall task down into simpler subtasks, continuing the process of decomposition until one can see how to construct devices that can directly implement the various component tasks. In the case of the governing problem, the relevant decomposition seems clear. A change need only be made to the throttle valve if the flywheel is not currently running at the correct speed. Therefore, the first subtask must be to measure the speed of the wheel, and the second must be to calculate whether there is any discrepancy between the desired speed and the actual speed. If there is no discrepancy, no change is needed, for the moment at least. If there *is* a discrepancy, then the governor must determine by how much the throttle valve should be adjusted to bring the speed of the wheel to the desired level. This will depend, of course, on the current steam pressure, and so the governor must measure the current steam pressure and then on that basis calculate how much to adjust the valve. Finally, the valve must actually be adjusted. This overall sequence of subtasks must be carried out often enough to keep the speed of the wheel sufficiently close to the desired speed.

A device able to solve the governing problem would have to carry out these various subtasks repeatedly in the correct order. So we could think of it as obeying the following algorithm:

```
(1) Begin:
      (i) Measure the speed of the flywheel;
     (ii) Compare the actual speed against the desired speed.
(2) If there is no discrepancy, return to step 1; otherwise:
      (i) Measure the current steam pressure;
     (ii) Calculate the desired alteration in steam pressure;
    (iii) Calculate the necessary throttle-valve adjustment;
     (iv) Make the throttle-valve adjustment.
(3) Return to step 1.
```

There must be some physical device capable of actually carrying out each of these subtasks. So we can think of the governor as incorporating a tachometer (for measuring the speed of the wheel), a device for calculating the speed discrepancy, a steam-pressure meter, a device for calculating the throttle-valve adjustment, a throttle-valve adjuster, and some kind of central executive to handle sequencing of operations. This conceptual breakdown of the governing task may even correspond to the governor's actual composition; that is, each subtask may be implemented by a distinct physical component. The engineering problem would then reduce to the (presumably much simpler) one of constructing the various components and hooking them together so that the whole system functions in a coherent fashion.

Now, as obvious as this approach now seems, it was not the way the governing problem was actually solved. For one thing, it presupposes devices that can swiftly perform some fairly complex calculations; and, for another, it presupposes transducers that can transform physical conditions into symbolic arguments for these calculations, and then transform the results back into physical adjustments. Both are well beyond the capabilities of anything available in the eighteenth century.

The real solution, adapted by Watt from existing windmill technology, was much more direct and elegant. It consisted of a vertical spindle geared into the main flywheel so that it rotated at a speed directly dependent upon that of the flywheel itself (see figure 16.1). Attached to the spindle by hinges were two arms, and on the end of each arm was a metal ball. As the spindle turned, centrifugal force drove the balls outwards and hence upwards. By a clever arrangement, this arm motion was linked directly to the throttle valve. The result was that, as the speed of the main wheel increased, the arms rose, closing the valve and restricting the flow of steam; as the speed decreased, the arms fell, opening the valve and allowing more steam to flow. The engine adopted a constant speed, maintained with extraordinary swiftness and smoothness in the presence of large fluctuations in pressure and load.

It is worth emphasizing how remarkably well the centrifugal governor actually performed its task. This device was not just an engineering hack employed because computer technology was unavailable. In 1858, *Scientific American* claimed that an American variant of the basic centrifugal governor, "if not absolutely perfect in its action, is so nearly so, as to leave in our opinion nothing further to be desired".

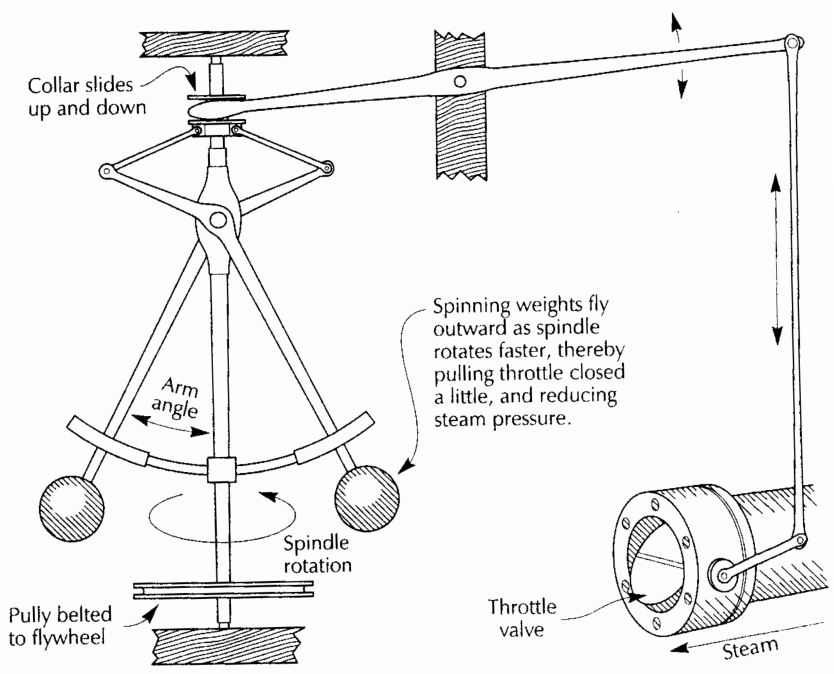
Collar slides up and down
Arm angle
Spinning weights fly outward as spindle rotates faster, thereby pulling throttle closed a little, and reducing steam pressure.
Spindle rotation
Pully belted to flywheel
Throttle valve
Steam

Figure 16.1: The Watt centrifugal governor for controlling the speed of a steam engine. (Drawing adapted from Farey 1827.)

But why should any of this be of any interest in the philosophy of cognitive science? The answer may become apparent as we examine a little more closely some of the differences between the two governors.

2 Two kinds of governor

The two governors described in the previous section are patently different in construction, yet they both solve the same control problem, and we can assume (for purposes of discussion) that they both solve it sufficiently well. Does it follow that, deep down, they are really the same kind of device, despite superficial differences in construction? Or are they deeply different, despite their similarity in overt performance?

It is natural to think of the first governor as a *computational* device; one which, as part of its operation *computes* some result—namely, the desired change in throttle-valve angle. Closer attention reveals that there is in fact a complex group of properties working together here, a group whose elements are worth teasing apart.

Perhaps the most central of the computational governor's distinctive properties is its dependence on *representation.* Every aspect of its operation, as outlined above, deals with representations in some manner or other. The very first thing it does is measure its environment (the engine) to obtain a symbolic representation of current engine speed. It then performs a series of operations on this and other representations, resulting in an output representation, a symbolic specification of the alteration to be made in the throttle valve. This final representation then causes the valve-adjusting mechanism to make the corresponding change.

This is why it is appropriately described as *computational* (now in a somewhat narrower sense): it literally computes the desired change in the throttle valve by manipulating symbols according to a schedule of rules. Those symbols, in the context of the device and its situation, have meaning; and the success of the governor in its task is owed to its symbol manipulations being in systematic accord with those meanings. The manipulations are discrete operations which necessarily occur in a determinate *sequence*; for example, the appropriate change in the throttle valve can only be calculated after the discrepancy, if any, between the current and desired speeds has been calculated. At the highest level, the whole device operates in a *cyclic* fashion: it first measures (or "perceives") its environment; it then internally computes an appropriate change in the throttle valve; and then it effects this change ("acts" on its environment). After the change has been made and given time to affect engine speed, the governor runs through whole the cycle again and again, repeatedly.

Finally, notice that the governor is *homuncular* in construction. Homuncularity is a special kind of breakdown of a system into parts or components, each of which is responsible for a particular subtask. Homuncular components are ones that, like departments or committees within bureaucracies, interact by communicating (that is, by passing meaningful messages). Obviously, the representational and computational nature of the governor is essential to its homuncular construction: if the system as a whole did not operate by manipulating representations, it would not be possible for its components to interact by communicating.

These properties—representation, computation, sequential and cyclic operation, and homuncularity—form a mutually interdependent cluster; a device with any of them will typically possess the others. Now, the Watt centrifugal governor does not exhibit this cluster of properties as a whole, nor any of them individually. As obvious as this may seem, it deserves some discussion and argument, since it often meets resistance, and a few useful insights can be gained along the way.

There is a common intuition to the effect that the angle at which the arms are swinging represents the current speed of the engine, and that it is because the quantities are related in this way that the governor is able to control that speed. This intuition is misleading, however: the concept of representation gets no real explanatory grip in this situation. Serious explanations of how governors work—ranging from a mid-nineteenth-century mechanic's manual for constructing them, through Maxwell's original dynamical analysis (see below), to contemporary mathematical treatments—never in fact traffic in representational talk. Why not?

The heart of the matter is this. At all times, the speed of the engine influences the angle of the arms. Yet the arms are directly connected to the throttle valve, which controls the flow of steam to the engine. Thus, at all times, the angle of the arms is also influencing the speed of the engine. The quantities are thus simultaneously determining the shapes of each other's changes. There is nothing mysterious about this relationship; it is quite amenable to mathematical description. However, it is much more subtle and complex than the standard concept of representation—very roughly, one thing "standing in" for another—can handle. In order to describe the relationship between arm angle and engine speed, we need a framework that is *more powerful,* with respect to this kind of situation, than talk of representations. That framework is the mathematical language of dynamics; and, in that language, the two quantities are said to be *coupled.* The real problem with describing the governor as a representational device, then, is that the relation of representing—something standing in for something else—is just too *simple* to capture the actual interaction between the centrifugal governor and the engine.

If the centrifugal governor is not representational, then it cannot be computational, at least in the specific sense that its processing cannot be a matter of the rule-governed manipulation of symbolic representations. Its noncomputational nature can also be established in another way. Not only are there no representations to be manipulated, there are also no distinct manipulations that might count as computational operations—no discrete, identifiable steps in which one representation could get transformed into another. Rather, the system's entire operation is smooth and continuous; there is no possibility of nonarbitrarily dividing its changes over time into distinct manipulations, and no point in trying to do so. From this, it follows that the centrifugal governor is not *sequential* and not *cyclic* in its operation in anything like the manner of the computational governor. Since there are no distinct processing steps, there can be no sequence in which those steps occur. There is never any one operation that must occur before another one can take place. Consequently, there is nothing cyclical about its operation. The device has, to be sure, an "input" end (where the spindle is driven by the engine) and an "output" end (the connection to the throttle valve). But the centrifugal governor does not follow a repetitive sequence in which it first takes a measurement, then computes a throttle-valve change, then makes that adjustment, and then starts over with another measurement, and so on. Rather, input, internal activity, and output are all happening continuously and at the very same time, much as a radio is producing music at the same time as its antenna is receiving signals.

The fact that the centrifugal governor is not sequential or cyclic in any respect points to yet another deep difference between the two kinds of governor. There is an important sense in which *time does not matter* in the operation of the computational governor. Of course, inasmuch as the device must control the engine speed adequately, its internal operations must be fast *enough*; moreover, they must happen in the right order. Beyond these minimal adequacy constraints, however, there is nothing that dictates *when* each internal operation should take place, *how long* each should take to carry out, or *how long* should elapse between them. There are only pragmatic implementation considerations: which algorithms to use, what kind of hardware to use to run the algorithms, and so forth. The timing of the internal operations is thus essentially *arbitrary* relative to that of any wider course of events. It is as if the wheel said to the governing system: "go away and figure out how much to change the valve to keep me spinning at 100 rpm. I don't care how you do it, how many steps you take, or how long you take over each step, as long as you report back within (say) 10 ms."

In the centrifugal governor, by contrast, there is simply nothing that is temporally unconstrained in this way. There are no occurrences whose timing or velocity or acceleration is arbitrary relative to the operation of the engine. All behavior in the centrifugal governor happens in the very same real time frame as both the speed and changes in speed of the flywheel. We can sum up the point this way: the two kinds of governor differ fundamentally in their *temporality*, and the temporality of the centrifugal governor is essentially that of the engine itself.

Finally, it need hardly be labored that the centrifugal governor is not a homuncular system. It has parts, to be sure, and its overall behavior is the direct result of the organized interaction of those parts. The difference is that those parts are not modules interacting by communicating; they are not like little bureaucratic agents passing representations among themselves as the system achieves the overall task.

3 Conceptual frameworks

In the previous section I argued that the differences in nature between the two governors run much deeper than the obvious differences in mechanical construction. Not surprisingly, these differences in nature are reflected in the kinds of conceptual tools that we must bring to bear if we wish to understand their operation. That is, the two different governors require very different conceptual frameworks in order to understand how they function *as governors*—that is, how they manage to control their environments.

In the case of the computational governor, the behavior is captured in all relevant detail by an algorithm, and the general conceptual framework we bring to bear is that of mainstream computer science. Computer scientists are typically concerned with what can be achieved by stringing together, in an appropriate order, some set of basic operations: either how best to string them together to achieve some particular goal (programming, theory of algorithms), or what is achievable in principle in this manner (computation theory). So we understand the computational governor as a device capable of carrying out some set of basic operations (measurings, subtractings, and so on), and whose sophisticated overall behavior results from nothing more than the complex sequencing of these basic operations. Note that there is a direct correspondence between elements of the governor (the basic processing steps it goes through) and elements of the algorithm that describes its operation (the basic instructions).

The Watt centrifugal governor, by contrast, cannot be understood in this way at all. There is nothing in it for any algorithm to latch onto. Very different conceptual tools have always been used instead. The terms in which it was described above, and indeed by Watt and his peers, are straightforwardly mechanical: rotations, spindles, levers, displacements, forces. Last century, more precise and powerful descriptions became available, but these also have nothing to do with computer science. In 1868, the physicist James Clerk Maxwell made a pioneering extension of the mathematical tools of *dynamics* to regulating and governing devices (Maxwell 1868). The general approach he established has been standard ever since. Though familiar to physicists and control engineers, it is less so to most cognitive scientists and philosophers of mind, and hence is worth describing in a little detail.

The key feature of the governor's behavior is the angle at which the arms are hanging, for this angle determines how much the throttle valve is opened or closed. Therefore, in order to understand the behavior of the governor we need to understand the basic principles governing how arm angle changes over time. Obviously, the arm angle depends on the speed of the engine; hence we need to understand change in arm angle as a function of engine speed. If we suppose for the moment that the link between the governor and the throttle valve is disconnected, then this change is given by the differential equation

$$\frac{d^2\theta}{dt^2} = (n\omega)^2 \cos\theta \sin\theta - \frac{g}{l}\sin\theta - r\frac{d\theta}{dt}$$

where θ is the angle of the arms, n is a gearing constant, ω is the speed of the engine, g is a constant for gravity, l is the length of the arms, and r is a constant of friction at the hinges (Beltrami 1987, p. 163). This nonlinear, second-order differential equation tells us the instantaneous *acceleration* in arm angle, as a function of what the current arm angle happens to be (designated by the *state variable* θ), how fast the arm angle is currently changing (the derivative of θ with respect to time, $d\theta/dt$), and the current engine speed (ω). In other words, the equation tells us how change in arm angle is changing, depending on the current arm angle, the way it is changing already, and the engine speed. Note that in the system defined by this equation, change over time occurs only in arm angle, θ (and its derivatives). The other quantities (ω, n, g, l, and r) are assumed to stay fixed, and are called *parameters.* The particular values at which the parameters are fixed determine the precise shape of the change in θ. For this reason, the parameter settings are said to fix the *dynamics* of the system.

This differential equation is perfectly general and highly succinct: it is a way of describing how the governor behaves for any arm angle and engine speed. That generality and succinctness come at a price, however. If we happen to know what the current arm angle is, how fast it is changing, and what the engine speed is, then from this equation all we

can figure out is the current instantaneous acceleration. If we want to know at what angle the arms will be in a half-second, for example, we need to find a *solution* to the general equation—that is, an equation which tells us what values θ takes as a function of time. There are of course any number of such solutions, corresponding to all the different behavioral trajectories that the governor might exhibit; but these solutions often have important general properties in common. Thus, as long as the parameters stay within certain bounds, the arms will always eventually settle into a particular angle of equilibrium for that engine speed; that angle is known as a *point attractor.*

Thus far I have been discussing the governor without taking into account its effect on the engine, and thereby indirectly on itself. Here, the situation gets a little more complicated, but the same mathematical tools apply. Suppose we think of the steam engine itself as a dynamical system governed by a set of differential equations, one of which gives us some derivative of engine speed as a function of current engine speed and a number of other variables and parameters:

$$\frac{d^n\omega}{dt^n} = F(\omega, \ldots, \tau, \ldots)$$

One of these parameters is the current setting of the throttle valve, τ, which depends directly on the governor arm angle, θ. We can thus think of θ as a parameter of the engine system, just as engine speed ω is a parameter of the governor system. (Alternatively, we can think of the governor and steam engine as comprising a single dynamical system in which both arm angle and engine speed are state variables.) This *coupling* relationship is particularly interesting and subtle. Changing a parameter of a dynamical system changes its total dynamics (that is, the way its state variables change their values depending on their current values, across the full range of values they may take). Thus, any change in engine speed, no matter how small, changes not the state of the governor directly, but rather the way the state of the governor *changes,* and any change in arm angle changes not the speed of the engine directly, but the way the speed of the engine changes. Again, however, the overall system (engine and governor coupled together) settles quickly into a point attractor; that is, engine speed and arm angle remain constant—which is exactly the desired situation. Indeed, the remarkable thing about this coupled system is that under a wide variety of conditions it always settles swiftly into states at which the engine is running at a particular speed.

In this discussion, two very broad, closely related sets of conceptual resources have (in a very modest way) been brought into play. The first is *dynamical modeling,* that branch of applied mathematics which attempts to describe change in real-world systems by describing the states of the system numerically and then writing equations which capture how these numerical states change over time. The second set of resources is *dynamical systems theory,* the general study of dynamical systems considered as abstract mathematical structures. Roughly speaking, dynamical modeling attempts to understand natural phenomena as the behavior of real-world realizations of abstract dynamical systems, whereas dynamical systems theory studies the abstract systems themselves. There is no sharp distinction between these two sets of resources, and for our purposes they can be lumped together under the general heading of *dynamics.*

4 Morals

This discussion of the governing task suggests a number of closely related lessons for cognitive science. First, various different kinds of systems, fundamentally different in nature and requiring very different conceptual tools for their understanding, can subserve sophisticated tasks—including interacting with a changing environment—which may initially appear to demand that the system have knowledge of, and reason about, its environment. Second, in any given case, our sense that a specific cognitive task *must* be subserved by a (generically) computational system *may* be due to deceptively compelling preconceptions about how systems solving complex tasks must work. It may be that the basically computational shape of most mainstream models of cognition results not so much from the nature of cognition itself as it does from the shape of the conceptual equipment that cognitive scientists typically bring to the study of cognition. Third, cognitive systems may in fact be *dynamical* systems, and cognition the behavior of some (noncomputational) dynamical system. Perhaps, that is, cognitive systems are more relevantly similar to the centrifugal governor than they are either to the computational governor, or to that more famous exemplar of the broad category of computational systems, the Turing machine.

In what follows, this third suggestion will be developed into a specifically dynamical conception of cognition via an explication of the key concept of *dynamical system.* An example will then illustrate how even "high level" cognitive performances may be intelligible in thoroughly dynamical terms. The final section will briefly defend the viability of the dynamical conception as a research program in contemporary cognitive science.

5 Three kinds of system

What are dynamical systems? How do they differ not only from computers, but also from connectionist networks—hitherto the main competition for computational models in cognitive science?

Begin with the concept of a *system.* The term 'system' is often used very loosely, designating pretty much any complex thing we wish to talk about (for example, a roulette betting system). For current purposes, however, systems are best defined more tightly as sets of *variables* (things, aspects, features, and the like) which change over time, such that the way any one variable *changes* at a given time depends on the *states* of other variables in the system at that time. Taken together, the states of all the variables make up the state of the system as a whole. Systems can be affected by external factors as well; these are commonly known as *parameters* when they are relatively fixed and influence only the way the variables interact, and as *inputs* when they are occasional and set the actual states of some variables directly.

Systems can be classified in many different ways. The most useful classifications are those that are neither too wide nor too narrow. For example, sometimes computers are taken very broadly as systems that *compute,* dynamical systems as systems that *change,* and connectionist networks as just a species of dynamical system. However, such wide definitions wash out the very contrasts that are most important for understanding what is going on in cognitive science. In what follows, informal, but more restrictive, specifications will be adopted as guides: computers are *symbol manipulators,* dynamical systems are *sets of coupled magnitudes,* and connectionist systems are *networks of neural units.* The differences among these ideas can be articulated by focusing on four points of contrast: the kinds of variables involved, the ways states change, the tools for describing the changes, and more general features that lend each kind of system its distinctive character (see table 16.1).

Thus, computers (in the relevant sense) always have digital variables.[1] For a variable to be digital, there must be some set of discrete values, such that at any relevant time the variable has unambiguously taken on one or another of those values. Thus a memory location (bit)

	Computational systems	Dynamical systems	Connectionist systems
Informal description	Symbol manipulators	Sets of coupled magnitudes	Networks of neural units
Classic exemplars	Turing machine; LISP machine	Solar system; Watt governor	Perceptron; Hopfield net
Kinds of variable	Digital—often syntactical	Quantitative—states and rates	Quantitative—activation levels
Changes in states	Discrete steps (sequential)	Interdependent in "real" time	Propagated interaction
Tools for description	Transition rules ("programs")	Differential equations	Weighted-sum equations
General character	Interpretable as representations	Coupled—with environment too	Homogenous & high-dimensional

Table 16.1: Differences among kinds of systems.

in an ordinary electronic computer is on or off; an abacus rod has a definite number of beads on each end; a Turing-tape square is either empty or occupied by a '1'; and so on. Variables in a dynamical system, by contrast, are not essentially digital (or not); rather, the important thing is that they be *quantities*—that is, variables for which it makes sense to talk about *amounts* or about *distances*[2] between values. Amounts and distances are subject to measurement: the use of some standard "yardstick" for systematically assigning numbers to values and differences. Thus the height of a falling object can be measured in meters, and the distance between any two heights determined by subtraction to yield a distance fallen. This contrasts with computers, in which there is a critical *difference* but no relevant *distance* between values of a variable (such as being empty as opposed to being occupied by a '1'). Since the variables of dynamical systems are quantities, a little mathematics allows us to talk of distances between total states. Hence the state *set* of a dynamical system is, in an interesting sense, a *space*, within which any state is a *position*, and any behavior a *trajectory*. These in turn clear the way for other important and powerful dynamical notions, such as *attractor, bifurcation, stability, and equilibrium*.

Connectionist networks have quantities as variables, so they also differ in this respect from computers. How do they differ from dynamical systems? An essential feature of connectionist networks is that their

variables are modeled, in a very generic way, on biological neurons; consequently, they exhibit a distinctively "neural" form of interactive change. Each variable has a certain *activity level* (its value), and can be influenced by a certain subset of other variables. This influence, thought of as flowing or propagating along a "connection", is modulated by a parameter, known as a *weight*. The way in which units in connectionist networks change their activity values is specified by a simple function (usually just summation) of the modulated activities of all the units by which they are influenced.[3]

Now, dynamical systems *can* change state in this neural fashion, but they need not (consider the centrifugal governor, which has no connections or weights). Rather, what really makes change *dynamical*, in a strong sense, is an orthogonal requirement: it happens *in "real" time*. What does this mean?

Obviously, any system that changes at all, changes "in" time in *some* sense. But consider an abstract Turing machine, a mathematical entity standing outside the actual time of everyday events. This machine has states, and it "changes" from one state to the next; but there is no sense in which it *spends* time in any state, or *takes* time to change state. "Time" here is nothing more than an ordered series of discrete points (t_1, t_2, ...). These points have no *duration*; nothing *elapses*. The integers are a convenient way to index these time points, since they have a familiar order. But this use can be misleading, since it falsely suggests that there are *amounts* of time involved. Practical considerations aside, one might just as well use proper names, ordered alphabetically, as labels for points of time.

Now, *real* (actual, everyday, worldly) time has two obvious properties that mere orders lack. First, real time is at least dense (between any two points of time there is another one); and second, real time is a *quantity* (there are *amounts* of time and *distances* between times). These give rise to a distinctive sense in which a process can happen *in* time (real or otherwise). The system must be in some state or other at every point of time; and so, if time is dense, the system's states and state changes must themselves be densely ordered in time. A system that is in time in this sense is potentially always changing. Further, if time is a quantity, we can relate happenings in the system in terms amounts of time; we can talk for example of how long they take, and (if the variables are quantities) of the *rate* of change. This latter fact is particularly important. For, if both time and system variables are continuous, we can talk of instantaneous rates of change, accelerations, and so on, and thus of systems in which rates of change depend on the current states, and even *current rates of change,* of the system variables (for example, the solar system and the centrifugal governor). In order to describe such systems, we need mathematical tools that can relate rates of change in variables to those variables themselves; that is, we need *differential equations.*

But doesn't change in *any* actual system—including computers—happen in real time, and thus *in* time in the relevant sense? Yes and no. Consider classical computation and complexity theory—the study of what computers as such can do. This theory is founded on the idea that details of *timing* don't matter; time is measured simply in steps or operations. But the theory carries over in its entirety to concrete, physical computers such as my Macintosh. This is to say that, in understanding the behavior of ordinary computers *as computers,* we can abstract away from the dense and quantitative nature of real time. From this point of view, they are only incidentally in time; changing the timing details would not affect what they are computing in any way. By contrast, one could never understand the behavior of the solar system while ignoring its timing. This is one of the most important differences between computers and systems that are genuinely *dynamical* in the current sense.[4]

What can be said more positively about state changes in a computer? Well, the variables are digital, and so any state change must be from one digital configuration to another. This means that transitions are *essentially* discrete: there is no theoretically relevant time between any time and the next, and no theoretically relevant state between any state and the next. These properties are reflected in the nature of the rules which describe the behavior of computers. These rules ("programs") always specify what the *next* state is, usually by specifying a discrete *operation* which transforms the current state into the next state. Further, the rules are always expressed in terms of digital properties of variables: for example, change a square from *empty* to *occupied by a '1'.*

So far, computers have been characterized in terms of the nature of their variables, their state changes, and how these state changes are specified—effectively, as *automatic formal systems* (Haugeland 1985). Yet nothing could count as a computer, in the full sense, without computing. In the most general terms, computing requires a computer, an external domain, and a systematic correspondence between the two such that states and transitions of the former *make sense* in relation to

the latter. In other words, computers are those automatic formal systems whose structure supports a systematic and sensible correspondence with some domain (such as arithmetic, baseball, or whatever). Note that the digital nature of computers characteristically supports computing of a more particular kind: namely, that in which the domain itself has a clear, well-ordered structure. Relevant states of the system are structured configurations of tokens interpretable as *symbolic representations* of the domain; and state changes amount to *inferences* from one symbolic representation to another.

Now, it is clearly not essential to dynamical systems that they be systematically and sensibly interpretable with respect to some external domain. Despite the best efforts of astrologers, there is no good way to interpret the motions of the planets with regard to any other concerns. But this is not to say that dynamical systems *cannot* be interpreted; sometimes they can, and this may enable them to be understood as exhibiting cognitive functions. But any interpretation, if there is one, is always after the fact; it is no part of the dynamical system as such. A system is dynamical in virtue of *other* properties. The nature of their variables and state-changes have already been discussed, but—as in the case of computers—there is more to the story. Much of the unique flavor of dynamical systems is captured by the idea of *coupling.* As explained above, two variables are coupled when the way each *changes* at any given time depends directly on the way the other *is* at that time. In other words, coupled variables simultaneously, interdependently coevolve, just like arm angle and engine speed in the centrifugal governor. Genuinely dynamical systems exhibit high degrees of coupling; every variable is changing all the time, and all pairs of variables are, either directly or indirectly, mutually determining the shapes of each other's changes. For example, in the solar system, the position and momentum of every massive body is constantly changing, and every variable influences every other one.

In a computer, by contrast, at each step most variables remain unchanged; and the changes that do occur are influenced by at most a few other values. Interestingly, this is also a point of contrast between connectionist networks and dynamical systems. Some networks (for example, fully recurrent networks) are dynamical in our sense; but others—such as archetypal three-layer feed-forward networks (generalized perceptrons)—exhibit no coupling at all.[5] What distinguishes connectionist networks, apart from their basically neural interaction, is that they are typically *high-dimensional* and *homogeneous.* The former property is nothing more than having a relatively[6] large number of variables; the latter is having all variables change in basically the same way. Standard mathematical specifications of connectionist networks involve just a single equation schema with indexes for variables and parameters; this form of description is made possible by homogeneity, and necessary by high-dimensionality.

This completes our brief tour of computers, dynamical systems, and connectionist networks as categories of systems. Two points are worth noting before moving on. First, the aim has been to capture the core idea in each case, rather than to provide sets of conditions which provide precise, rigid and mutually exclusive boundaries. Second, there are many different notions of computer, dynamical system, and so on, that are useful for different purposes. Those offered here are not intended to be better or more correct in general, but, at best, more useful for the philosophy of cognitive science.

6 Three conceptions of cognition

The essence of the dynamical conception of cognition is the idea that cognitive systems are dynamical systems, and cognition the behavior of such systems. The distinctions drawn in the preceding section now combine with the earlier discussion of governors to yield a more precise elaboration of this idea. Both the dynamical and the computational conceptions of cognition turn out to comprise clusters of mutually compatible and constraining commitments with three layers. The core in each case is a specific empirical hypothesis concerning the kind of system that natural cognitive systems are. Wrapped around this core are two further commitments, one concerning the "cognitive level" properties of cognitive systems, and the other concerning the kinds of conceptual tools that are most appropriate for the study of cognition. Thus, the dynamical and computational conceptions both constitute richly textured visions of the nature of cognition.

Thus, in the computational vision, cognitive systems are computers (digital, rule-governed, interpretable systems) with a modular internal structure; they interact with their environments in a cyclic process that begins with input transducers producing symbolic representations in response to the environment, continues with sequential internal computations over symbolic structures, and ends with output transducers affecting the environment in response to symbolic specifications. Each internal operation is algorithmically specified and takes place in the

system's own arbitrary time frame; the whole process can be considered independently of the body and the environment except insofar as they deliver occasional inputs and receive outputs. Since the cognitive system is a computer that works by sequential transformations of symbolic representations, its most revealing descriptions are those using the conceptual apparatus of mainstream computer science. In short, the computational vision sees people as computational governors writ large.

This contrasts at every level with the dynamical vision, in which people bear deeper similarities to the *centrifugal* governor. Cognitive systems are taken to consist of sets of coupled quantities evolving in real time. These quantities may be abstract "cognitive" features (see the example below) or they may be aspects of the body or even of the environment. At a higher level, cognitive systems are understood to be complexes of continuous, ongoing, mutually constraining *changes*. The fundamental mode of interaction with the environment is not to represent it, or even to exchange inputs and outputs with it; rather, the relation is better understood via the technical notion of coupling. To be sure, cognition can, in sophisticated cases, involve representation and sequential processing; but such phenomena are best understood as emerging from a dynamical substrate, rather than as constituting the basic level of cognitive performance. As complexes of continuous, ongoing change, cognitive systems are best understood using the very same tools that have proven so effective for such processes elsewhere in science: dynamical modeling and dynamical systems theory.

Where does connectionism fit into all of this? Somewhere awkwardly in the middle. Some connectionist networks are thoroughly dynamical; but others, such as layered feed-forward networks, are configured to behave more in the cyclic and sequential fashion of computational systems. Not surprisingly, when trying to understand their systems, connectionists sometimes borrow from computer science, sometimes from dynamics, and sometimes from other fields such as statistics. Connectionist networks sometimes transform static input representations into static output representations; other times, they settle dynamically into attractors, bifurcate, and so on. In short, connectionism may be kind of half-way house between two conceptions of cognition each of which has a greater theoretical integrity on its own. Of course, it might turn out that understanding cognition really does require an eclectic mix of ingredients from several conceptual frameworks. Alternatively, it may be that connectionism is an unstable mongrel, little more than a temporary phase in the transition from generically computational to generically dynamical approaches to the study of cognition.

7 An example of dynamical research

At this stage, an example may help convey an intuitive sense of how the dynamical approach, just specified in very abstract terms, can yield real insights into the nature of cognition. Consider the process of coming to make a decision among a variety of options, each of which has attractions and drawbacks. This is surely a high-level cognitive task, if anything is. Psychologists have done countless experimental studies of how people choose, and have produced almost as many mathematical models to describe and explain that behavior. The dominant approach in modeling stems from the classic expected-utility theory and statistical decision theory, as originally developed by von Neumann and Morgenstern (1944/80). The basic idea is that an agent makes a decision by selecting the option that has the highest expected utility, which is calculated in turn by combining some formal measure of the utility of each possible outcome with the probability that it will eventuate if that option is chosen. Much of the work within this framework is mathematically elegant and provides a useful account of optimal reasoning strategies. As an account of the *actual* decisions people reach, however, classical utility theory is seriously flawed; human subjects typically deviate from its recommendations in a variety of ways. As a result, many theories proposing variations on the classical core have been developed—typically relaxing certain of its standard assumptions, with varying degrees of success in matching actual human choice behavior. Nevertheless, virtually all such theories remain subject to some further drawbacks:

- They do not incorporate any account of the underlying *motivations* which give rise to the utility that an object or outcome holds at a given time.
- They conceive of the utilities themselves as static values, and can offer no good account of how and why they might change over time, or why actual preferences are often inconsistent and inconstant.
- They offer no serious account of the deliberation *process*, with its attendant vacillations, inconsistencies and distress; and they have nothing to say about the relationships that have been uncovered between time spent deliberating and the choices eventually made.

Curiously, these drawbacks appear to have a common theme; they all concern, one way or another, *temporal* aspects of decision making. It is worth asking whether they arise because of some deep structural feature inherent in the whole framework which conceptualizes decision-making behavior in terms of calculating expected utilities.

Notice that utility-theory based accounts of human decision making ("utility theories") are deeply akin to the computational solution to the governing task. That is, if we take such accounts as not just describing the *outcome* of decision making behavior, but also as a guide to the structures and processes that *generate* the behavior, then there are basic structural similarities to the computational governor. Thus, utility theories are straightforwardly computational; they are based on static representations of options, utilities, probabilities, and the like, and processing is the algorithmically specifiable internal manipulation of these representations to obtain a final representation of the choice to be made. Consequently, utility theories are strictly sequential; they presuppose some initial temporal stage at which the relevant information about options, likelihoods, and so on, is acquired; a second stage in which expected utilities are calculated; and a third stage at which the choice is effected in actual behavior. And, like the computational governor, they are essentially atemporal; there are no inherent constraints on the timing of the various internal operations with respect to each other or changes in the environment.

What we have, in other words, is a model of human cognition which, on the one hand, instantiates the same deep structure as the computational governor, and on the other, seems structurally incapable of accounting for certain essentially temporal dimensions of decision making behavior. At this stage, we might ask: what general *kind* of model of decision making behavior we would get if, rather, we took the *centrifugal* governor as a prototype? It would be a model with a relatively small number of continuous variables influencing each other in real time. It's behavior would be defined by low-dimensional nonlinear differential equations. And it would be a model in which the agent and the choice environment, like the governor and the engine, are tightly coupled.

It would, in short, be rather like the "motivational oscillatory theory" (MOT) model described by mathematical psychologist James Townsend (Townsend 1992). MOT enables modeling of various

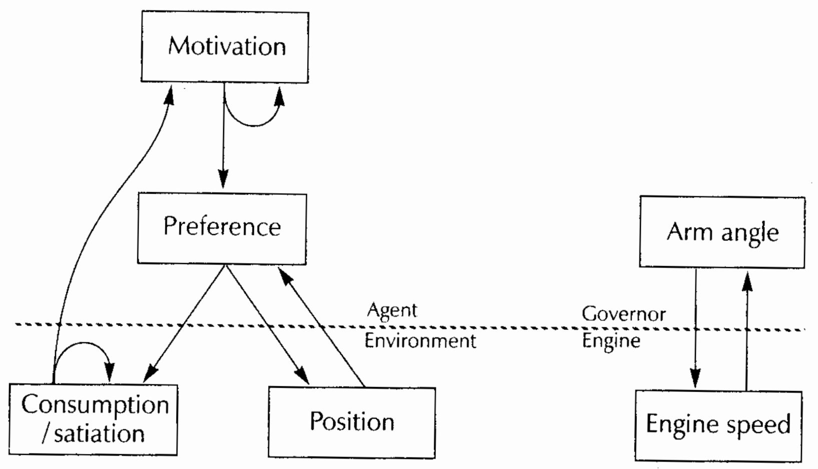

Figure 16.2: The MOT model of decision making compared to the centrifugal governor. Boxes represent variables and arrows represent influence. In each case, coupled variables evolve continuously in time, and the system spans a kind of agent/world divide. The MOT system is significantly more complex than the governor, but still very simple compared with many dynamical models of cognition. Note that these diagrams should not be interpreted in a connectionist fashion; the lines are not connections and do not have weights. (The MOT diagram is adapted from Townsend 1992).

qualitative properties of the kind of periodic behavior that occurs when circumstances offer the possibility of satiating desires arising from more or less permanent motivations. An obvious example is regular eating in response to recurring natural hunger. It is built around the idea that in such situations, your underlying motivation, transitory desires with regard to the object, distance from the object, and consumption of it are continuously evolving and affecting each other in real time; for example, if your desire for food is high and you are far from it, you will move toward it, which influences your satiation and so your desire. The framework thus includes variables for the current state of motivation, satiation, preference, and action (movement), and a set of differential equations that describe how these variables change over time as a function of the current state of the system.[7]

MOT stands to utility theories as the centrifugal governor does to the computational governor. In MOT, cognition is not symbol manipulation, but rather state-space evolution in a dynamical system that is in certain key respects rather like the centrifugal governor. It is a

system that demands dynamical tools in its analysis. MOT produces behavior which, if one squints while looking at it, seems like decision making—after all, the agent will make the move that offers the most reward, which in this case means moving toward food if hungry enough. There is, however, a sense in which this is decision making without decisions, for there never are, in the model, any discrete internal occurrences that one could reasonably characterize as decisions. In this approach, decision making is better thought of as the behavior of an agent under the influence of the pushes and pulls that emanate from desirable outcomes, undesirable outcomes, and internal desires and motivations; in a quasi-gravitational way, these forces act on the agent with strengths varying as a function of distance.

The MOT model is a special case of a more general dynamical framework that Townsend and Jerome Busemeyer (1993) call "decision field theory". That framework, too complex to describe succinctly (an overview is provided in Busemeyer and Townsend 1995), faithfully models a wide range of behavior more easily recognizable as decision making, as studied within the traditional research paradigm. Indeed, their claim is that decision field theory "covers a broader range of phenomena in greater detail" than do classical utility theories, and even goes beyond them by explaining in a natural way several important paradoxes of decision making. The important point is that the general decision field theory works on the same fundamental dynamical principles as MOT. There is thus no question that at least certain aspects of human high-level cognitive functioning can be modeled effectively using dynamical systems of the kind that can be highlighted by reference to the centrifugal governor.

8 Is the dynamical conception viable?

In order soundly to refute a What-else-could-it-be? argument, a proposed alternative must be viable—that is, plausible enough that it is reasonably deemed an open empirical question whether the orthodox approach or the alternative is ultimately more promising.

One measure of the viability of an approach is whether valuable research can be carried out within its terms. On this measure, the dynamical approach is certainly in good health. Dynamical models have been or are being developed for a very wide range of aspects of cognitive functioning, from (so-called) "low level" or "peripheral" aspects such as perception and motor control, to (so-called) "central" or "higher" aspects such as language and decision-making, through to related areas such as psychiatry and social psychology. As already mentioned, a good deal of connectionist work falls under the dynamical banner, and this work alone would qualify the dynamical approach as worth taking seriously. However there are now also *non*connectionist dynamical models of numerous aspects of cognition, and their ranks are swelling. Further, in a number of fields under the broader umbrella of cognitive science, dynamics provides the dominant formal framework within which particular theories and models are developed; these include neural modeling, autonomous agent ("animat") research, ecological psychology and, increasingly, developmental psychology.[8]

Of course, it is quite possible for a research program to flourish, even though, for deep reasons, it will eventually prove inadequate—either in general or in particular respects. (Remember behaviorism.) So, in evaluating the plausibility of an alternative, we should also consider whether any known *general* considerations either support it or—perhaps more importantly—undermine it. Many general considerations have been raised in favor of the computational conception of cognition; and, given the stark contrasts, these might appear to argue *against* the dynamical alternative. It isn't possible to address all (or even any) such arguments adequately here; but I will briefly comment on one of the most powerful—not, however, to refute it, but rather to reveal something of the potential of the dynamical approach.

Cognition is often distinguished from other kinds of complex natural processes (such as thunderstorms or digestion) by pointing out that it depends on *knowledge.* One challenge for cognitive scientists is to understand how a physical system might exhibit such dependence in its behavior. The usual approach is to suppose that the system contains internal structures that *encode* or *represent* the knowledge. Further, it is often thought that the best way to encode or represent knowledge is to use *symbolic* representations, manipulated by some computational system. Thus, insofar as the dynamical approach abjures representation completely, or offers some less powerful representational substitute, it may seem doomed.

But, while the centrifugal governor is clearly nonrepresentational, and while (as argued above) representation figures in a natural cluster of deep features that are jointly characteristic of computational models, in fact there is nothing to prevent dynamical systems from incorporating some form of representation. Indeed, an exciting feature of the dynamical approach is that it offers opportunities for dramatically

reconceiving the nature of representation in cognitive systems, even within a broadly noncomputational framework. A common strategy in dynamical modeling is to assign representational significance to some or all of the state variables or parameters (see, for example, the Townsend and Busemeyer decision-field-theory model mentioned above, or consider a connectionist network in which units stand for features of the domain).

Though representations of this kind may be exactly what is needed for *some* cognitive modeling purposes, they don't have the kind of combinatorial structure that is often thought necessary for *other* aspects of high-level cognition. However, within the conceptual repertoire of dynamics there is a vast range of entities and structures which might be harnessed into representational roles; individual state-variables and parameters are merely the very simplest of them. For example, it is known how to construct representational schemes in which complex contents (such as linguistic structures) are assigned in a recursive manner to points in the state-space of a dynamical system, such that the representations form a fractal structure of potentially infinite depth, and such that the behavior of the system can be seen as transforming representations in ways that respect the represented structure. Yet even these methods are doing little more than dipping a toe into the pool of possibilities. Representations can be trajectories or attractors of various kinds, trajectories obtained by the sequential chaining of attractors, even such exotica as transformations of attractor arrangements as a system's control parameters change (Petitot 1995).

Dynamicists are actively exploring how these and other representational possibilities might be incorporated into cognitive models, without buying the rest of the computational worldview. Consequently, while the dynamical approach is certainly a very long way from having actual solutions to most concrete problems of knowledge representation, it clearly holds sufficient promise to maintain its current viability as an alternative.

What positive reasons are there to think that the dynamical approach is actually on the right track? Again, space does not allow serious treatment of the arguments, but some are at least worth mentioning. In practice, an important part of the appeal of the dynamical approach is that it brings to the study of cognition tools that have proved extraordinarily successful in many other areas of science. But is there anything about *cognition,* in particular, that suggests that it might best be understood dynamically?

One central fact about natural cognitive processes is that they always happen *in real time,* which means not merely that, like any physical process (including ordinary digital computation), they occupy some extent of actual time, but that details of *timing*—durations, rates, rhythms, and so on—are critical to how they operate in real bodies and environments. As we saw above, dynamics is all about how processes happen in real time, whereas timing details are in a deep sense extrinsic to computational systems. Cognition also has other general features for which a dynamical approach appears well-suited. For example, it is a kind of complex behavioral organization which is emergent from the local interactions of very large numbers of (relatively) simple and homogenous elements. It is pervaded by both continuous and discrete forms of change. At every level, it involves multiple, simultaneous, interacting processes. Dynamics is a natural framework for developing theories that account for such features. Further, the systems within which cognition takes place (the brain, the body, the environment) demand dynamical tools for their description. A dynamical account of cognition promises to minimize difficulties in understanding how cognitive systems are real biological systems in constant, intimate, interactive dependence on their surroundings.[9]

A final way to underpin the viability of the dynamical conception is to place it and the computational conception in broad historical perspective. Computationalism, as cognitive science orthodoxy, amounts to a sophisticated instantiation of the basic outlines of a generically Cartesian picture of the nature of mind. The prior grip that this picture has on how most people think about mind and cognition makes the computational conception seem intuitively attractive. This would be unobjectionable if the Cartesian conception itself were basically sound. However, the upshot of philosophical evaluation of the Cartesian framework over the last three centuries, and especially in this century, is that it seriously misconceives mind and its place in nature.

Cognitive scientists tend to suppose that the primary respect in which Descartes was wrong about mind was in subscribing to an interactionist dualism: the doctrine that mind and body are two distinct substances that causally interact with one another. However, already by the eighteenth century the inadequacy of this particular aspect of Cartesianism had been exposed (Berkeley 1710/1977; Leibniz 1714/1977) and thoroughgoing brain-based materialism espoused (Hobbes 1651/1962; La Mettrie 1748/1912). Some of the greatest achievements of twentieth-century philosophy of mind have been to expose various other, more subtle, pervasive and pernicious epistemological and ontological misconceptions inherent in the Cartesian picture. These misconceptions are very often retained even when substance dualism is rejected in favor of some brain-based materialism, such as functionalism in its various guises.

Among the most important of these anti-Cartesian movements is one spearheaded by Ryle in Anglo-American philosophy (Ryle 1949/84) and Heidegger in continental philosophy (Heidegger 1927/62; Dreyfus 1991). Its target has been the generically Cartesian idea that mind is an inner realm of representations and processes, and that mind conceived this way is what causally explains intelligent behavior. This movement comprises three interrelated components. The first is a relocating of mind. The Cartesian tradition is mistaken in supposing that mind is an inner realm or entity of any sort, whether mental substance, brain states, or whatever. Ontologically, mind is much more a matter of what we *do* within environmental and social possibilities and bounds. Twentieth-century anti-Cartesianism thus draws mind *out*—in particular outside the skull. That aspect of mind that remains inside, and *is* the causal basis of our behavior, is *cognition.*

The second component is a reconceiving of our fundamental relationship to the world around us. In the Cartesian framework, the basic relation of mind to the world is one of representing it and thinking about it, with occasional "peripheral" interaction via perceiving and acting. It has been known since Berkeley that this framework has fundamental epistemological problems. But only more recently has it been shown that escaping these problems means reconceiving the human agent as essentially embedded in and skillfully coping with a changing world, and that representing and thinking about the world is secondary to and dependent upon such embeddedness (Guignon 1983).

The third component is an attack on the supposition that the kind of behavior we exhibit (such that we *are* embedded in our world and can be said to have minds) could ever be causally explained utilizing only the generically Cartesian resources of representations, rules, procedures, algorithms, and so on. A fundamental Cartesian mistake is to suppose, as Ryle variously put it, that practice is accounted for by theory, that knowledge *how* is explained in terms of knowledge *that,* or that skill is a matter of thought. In other words, not only is mind not to be found inside the skull, but also cognition, the inner causal basis of intelligent behavior, is not itself to be explained in terms of the basic entities of the general Cartesian conception.

My concern here is not to substantiate these claims or the post-Cartesian conception of the person to which they point (see, for example, Dreyfus 1972/92); it is simply to make the computational conception of cognition seem less than inevitable by casting doubt upon the philosophical framework within which it thrives. Orthodox computational cognitive science has absorbed some of the important lessons of seventeenth-century reactions to Cartesianism, but so far has remained largely oblivious to the more radical twentieth-century critiques. If, however, if we *begin* with a thoroughly post-Cartesian approach, the dynamical account of cognition will, in many ways, be immediately attractive. The post-Cartesian conception rejects the model of mind as an atemporal representer, and, like the dynamical approach to cognition, emphasizes instead ongoing, real-time interaction of situated agents with a changing world. The post-Cartesian agent is essentially temporal, since its most basic relationship to the world is real-time skillful coping; the dynamical framework is therefore a natural choice since it builds time in right from the start. The post-Cartesian agent manages to cope with the world without necessarily representing it. A dynamical approach suggests how this might be possible by showing how the internal operation of a system interacting with an external world can be so subtle and complex as to *defy* description in representational terms—how, in other words, cognition can *transcend* representation. In short, from the philosophical perspective that has managed to overcome the deep structures of the Cartesian world-view, the dynamical approach looks distinctly appealing; the Watt governor is preferable to the Turing machine as an archetype for models of cognition.

Notes

1. Here I am using the term 'computer' to refer specifically to digital computers rather than the wider class that includes so-called "analog computers". It is the narrower class which lies at the heart of the mainstream computational conception of cognition.
2. *Distance* is captured mathematically by a *metric*—a function that maps pairs of values of a variable onto real numbers in a way that satisfies certain familiar constraints. Any *quantity*, as such, will have a nontrivial metric associated with it. (An example of a trivial metric would be the same/different "metric", a function that returns 1 if two values are different and 0 if they are the same. Applied to sequences, this kind of metric is known as "Hamming distance". It is a useful function for some purposes, but is not a metric in the relevant sense.)

3. Hence the ubiquitous "sigma" (Σ) term in connectionist equations.
4. The general field of dynamical systems theory studies many systems that are defined in terms of discrete maps. Some of these are just discretized versions of continuous systems that are *in* time in a full-blooded sense. Others, however, are not (including some that exhibit chaotic behavior). These systems are dynamical, but only in a wider sense than is used here. They bear interesting similarities both to dynamical systems and to computers.
5. The presence of a connection between two units is not enough for coupling in the full sense. Genuine coupling requires bidirectional connections and simultaneous update.
6. Relative to what? There are many ways to cash this out, but for nonlinear systems, "large" is roughly *so many that we find it hard to understand the behavior of the system.*
7. The equations, with rough and partial translations into English, are:

$$\frac{dm}{dt} = M - m - c$$

(The change in motivation depends on how the current levels of motivation and consumption compare with some standard level of motivation, M.)

$$\frac{dz}{dt} = m \times \left[\frac{1}{z_1^2 + z_2^2 + a} + 1 \right]$$

(The change in *preference* for a goal depends on current motivation and distance from the object of preference.)

$$\frac{dc}{dt} = (x + C - c) \times \left[\frac{b}{z_1^2 + z_2^2 + r} + 1 \right]$$

(The change in consumption depends on the level of preference, the level of consumption, and the distance from the object of preference.

$$\frac{dz_1}{dt} = -(x \cdot z_1) \qquad \frac{dz_2}{dt} = -(x \cdot z_2)$$

(Movement toward or away from the object depends on the current level of preference for it.)

8. Rather than cite individual examples, I merely list here some overviews or collections which the interested reader can use as a bridge into the extensive realm of dynamical research on cognition. Kelso (1995) is both a manifesto for the dynamical approach and an accessible encapsulation of one powerful research program. A representative sampling of current research is contained in Port and van Gelder (1995), which also contains guides to a much larger literature. An excellent illustration of the power and scope of dynamical research, in a neural network guise, is Grossberg (1988). Serra and Zanarini (1990) present an overview of a variety of dynamical systems approaches in artificial intelligence research. For the role of dynamics in developmental psychology, see Smith and Thelen (1993) and Thelen and Smith (1993).

9. For more detailed treatment of these arguments, see van Gelder and Port (1995).