Dynamical Models in Psychology
Psychology 465A
Dr. Lawrence Ward

1. Statical vs Dynamical Models

1.1 Statical Stevens's Law: $R = c \cdot S^m$

1.2 Dynamical version: $R_i = cS_i^m$

1.3 In 1.2 above the *i* index is a surrogate for time

2. Dynamical models: models of *change* over *time*

2.1 Differential equation is simplest:

$$\frac{dx}{dt} = at$$

Says rate of change of *x* is a linear function of time, *t*. Say start at $t = 0$ with $a = 0.001$, so $a \cdot 0 = 0$. At $t = 1000$, $dx/dt = 0.001 \cdot 1000 = 1$. At $t = 10{,}000$, $dx/dt = 0.001 \cdot 10000 = 10$, etc. so *x* is increasing faster and faster as time goes by. How do we write this, i.e., solve the equation? We integrate both sides with respect to *t* (inverse of differentiation) and obtain

$$x = a\frac{t^2}{2}$$

Exercise: Draw a graph of this equation for various values of *a*. This is an explicit function of time.

2.2 An implicit function of time:

$$\frac{dx}{dt} = ax$$

With solution

$$x = e^{at}$$

Exercise: draw a graph of this equation for various values of *a*.

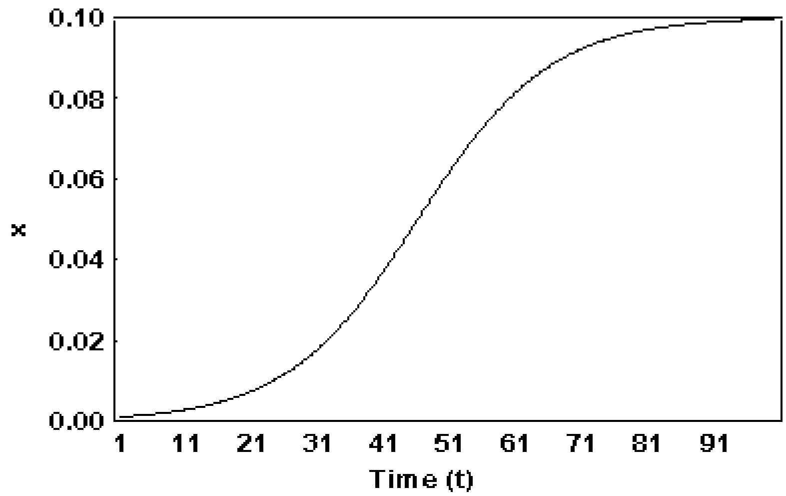
2.3 A more interesting dynamical model:

$$\frac{dx}{dt} = ax - bx^2$$

This is the equation in 2.2 with feedback ($-bx^2$) added. The solution is

$$x = \frac{c}{1 + (c / x_0 - 1)\exp[-a(t - t_0)]}$$

And a graph of this solution looks like this:



This has been used to model non-renewable resources such as petroleum.

2.4 The logistic difference equation: $x_{n+1} = ax_n - ax_n^2$

In a different form: $\frac{\Delta x}{\Delta t} = x(a - 1) - ax^2$

Resemblance to differential equation in 2.3 is clear. This equation leads to chaos!

3. Chaos theory

3.1 **Chaos: 1:** *the confused unorganized state existing before the creation of distinct forms,* **2:** *complete disorder* (Merriam-Webster, 1971, p. 81).

3.2 Chaos theory is about nonlinear dynamical systems and their maps. One of the most important goals of the theory of dynamical systems is "...*to understand the eventual or asymptotic behavior of an iterative process*." (Devaney, 1989, p. 17) The dynamical system could be specified by a system of nonlinear differential equations in time, in which case understanding the system involves solving the equations, a very difficult, often impossible undertaking. Alternatively, the system could be specified as the iteration of a discrete function, in which case we want to understand what happens to the value of the function as we repeatedly iterate it. This is the case that we are most concerned with in this chapter. Since this is not something we are used to doing with functions, I will tell you a bit more about iteration before we go on into chaos.

4. Iteration of a function.

To iterate a function $f(x)$ means to calculate the value of the function for some starting value, x, and then to put that value, called $f^1(x)$, back into the function in place of the initial x value, so we have $f^2(x) = f(f^1(x))$, $f^3(x) = f(f^2(x)) = f(f(f^1(x)))$, and so forth. (Notice that the superscripts do not have their usual meaning of multiplication here, but rather refer to the number of times a function has been applied to the starting value of the independent variable.) These points are called the *forward orbits* of the *map f(x)*. "Map" is just another word for function, since functions map a set of numbers into another set in the sense of connecting pairs of numbers. For example, what happens to the function $f(x) = x^2$ if we start with $x = 2$? Well, $2^2 = 4$, $4^2 = 16$, $16^2 = 256$, and so forth. Clearly the value of the function increases without limit (and it will do so for any starting value greater than 1 or, after the initial iteration, for any starting value less than -1), an example of an "exploding" process. Nonetheless, this outcome is extremely sensitive to the starting value. If we start with $x = 0.1$, the process dives toward an asymptote of 0 ($0.1^2 = 0.01$, $0.01^2 = 0.0001$, $0.0001^2 = 0.00000001$, etc.). We could call this a "collapsing" process. The same outcome will occur for any number between -1 and 1 except for 0. If $x = 0$, since $0^2 = 0$, the system will stay at 0. Similarly, if $x = 1$, then the system stays right at 1, since $1^2 = 1$. Any such points, for which $f(x) = x$, are called *fixed points* of $f(x)$. Finally, for $x = -1$, the first iteration will produce $(-1)^2 = 1$ and any subsequent iterations will then stay at 1. In this case, the behavior of the function as it is repeatedly iterated is clear and easy to see. If we knew that our dynamical system was described by this function we would simply have to know the starting value to be able to predict how it would behave, as summarized in Table 4.1. Notice that the graph of $f(x) = x^2$, although useful in many contexts, is not particularly useful here. What we are interested in is the behavior of the function under iteration, i.e., as it takes on new values at discrete time points in the future or in the past.

Table 4.1. Behavior of $f(x) = x^2$ as it is repeatedly iterated from a starting value of x
.

Starting x	< -1	-1	-1< x <0	0	0 < x < 1	1	>1
Behavior	explodes	1	collapses	0	collapses	1	explodes

As interesting as the function $f(x) = x^2$ is, it does not display the really interesting and useful behavior of periodicity or of chaos. A slight modification will take care of periodicity. Consider the function $f(x) = x^2-1$. This map has fixed points at $(1\pm\sqrt{5})/2$; for example $((1+\sqrt{5})/2)^2 - 1 = (1+\sqrt{5})/2)$, although that is not obvious (go ahead, do the calculations for yourself). More interestingly, the points 0 and -1 are *periodic points* of the function, and they form a *periodic orbit*. This means that the value of the function cycles between these points forever if it ever gets to one or the other from some other point. To see this, start with $x = 0$. Then, $x^2-1 = 0^2 -1 = -1$. Now, we have to calculate the function for $x = -1$: $x^2-1 = (-1)^2 -1 = 1 -1 = 0$. We are back at 0. And so it goes, the function simply flips back and forth between 0 and -1 forever, a periodic orbit of period 2 (takes two iterations to come back to the starting point).

5. Chaos

But what about chaos? How do we get chaos? Surprisingly (for those who have never experienced it, at least), a further simple modification seems to do the trick. Consider the function $f(x) = 4x - 4x^2$ (the logistic difference equation with $a = 4$). I iterated this function for a starting value of 0.3. Here are the first 13 iterates (according to my 12-digit-precision calculator): 0.3, 0.84, 0.5376, 0.99434496, 0.02249224209, 0.0879453645, 0.32084391, 0.871612381, 0.4476169529, 0.989024066, 0.004342185149, 0.166145577, 0.554164898, 0.9882646555 And so it goes, the numbers just keep jumping around like that, seemingly at random, never repeating (although there is a larger pattern, which we will discuss shortly). This seems more like chaos according to the dictionary - a disordered list of numbers with no apparent pattern. But it isn't! Chaos can never be displayed in a list of numbers written with finite precision. That is because any finite precision calculation of the values of any function must eventually repeat a number, since only a finite number of different numbers can be represented, and from that point on it is periodic - the same as the last time that particular number occurred. To see this informally, consider making the above calculations with only 2-digit precision, that is, we simply drop, or round off if you prefer - it doesn't matter- any digits higher than the two to the right of the decimal point (since all numbers will be between 0 and 1 inclusive). So, starting with 0.30 again, we get 0.30, 0.84, 0.54, 0.99, 0.04, 0.15, 0.51, 1.00, 0.00, 0.00, 0.00, 0.00, forever. Oops! A fixed point! If we add a few more digits of precision we can get periodic behavior, a cycling between, say, 9 or 10 or even 13 periodic points. But no matter how many digits of precision we add, rounding (or dropping) always occurs and we end up in periodic behavior, although it can look pretty random indeed, even with only 12-digit precision. So how do we know this is chaos? And what is chaos anyway, if we can't ever see it?

As you might have expected, chaos is an abstract mathematical concept that can only be approximated in our real, quantized world. Here is one respectable mathematical definition, slightly modified from Devaney (1989): Let V be a set. A function f that maps V into itself is said to be *chaotic* on V if (1) f has sensitive dependence on initial conditions, (2) f is topologically transitive, and (3) periodic points are dense in V. To take the easy one first, *dense* periodic points means that there is a very large number of periodic points everywhere in V, that is, that in any small neighborhood near a particular point in V, there must be a very large number of periodic points (the word *infinite* springs to mind, but is not precisely what is meant here). This is so that the function can return to that neighborhood again and again without ever repeating exactly- otherwise it would be strictly "periodic" - possibly with a very long period. This might sound like an oxymoron, a nonperiodic function with a lot of periodic points, and it is! To have so many periodic points that they can never all be reached, and none repeated, is the ultimate in periodicity - the function is so intensely periodic that it isn't periodic any more, it has become chaotic! In a sense, chaos is the asymptote of periodicity as the number of periodic points increases without limit. *Topologically transitive* means that a map will eventually move under iteration from one arbitrarily small neighborhood of phase space to any other, so that the system can't be decomposed into disjoint sets that are invariant under the map, i.e., it can't become trapped in some small neighborhood, like a fixed point, forever. Finally, and probably most importantly, *sensitive dependence* means that there exist points in phase space arbitrarily close to every point x that eventually diverge from x by at least some amount δ under iteration of f. All points need not diverge, but at least one in every neighborhood must do so. This means that the map defies numerical computation - the smallest roundoff error becomes magnified by iteration and the calculated orbit bears no resemblance to the real one. Moreover, this means that precise prediction of the future behavior of such an equation is impossible if that prediction depends on numerical calculation. Thus, the behavior appears to be "random" or "stochastic." Chapter 24 discusses the relationships between chaos and randomness.

Sensitive dependence is also probably the easiest and most "empirical" attribute of chaos, since we can see it happen even in limited-precision calculations. For example, consider what happens in our iteration of $f(x) = 4x - 4x^2$. Let's start at the same place but on the fourth iteration make a small change. Instead of using 0.5376, let's use 0.5377, a difference of only 0.0001. The sequence continues: 0.5377, 0.99431484, 0.02261135582, 0.0884003296, 0.311342845, 0.857633911, 0.48839194147, 0.999461012, 0.00215479033, 0.00860058883, 0.00341064748, 0.1317728928, 0.045763519. Now compare the two sequences. They do stay close together for a while, about the first 8 iterations. But then they diverge drastically! Where the original sequence had 0.166145577, 0.554164898, 0.9882646555 the new one has 0.00860058883, 0.00341064748, 0.1317728928. The orbits are now far from each other; the difference on the 13th iteration (0.9882646555 - 0.1317728928 = 0.856491627) is nearly 10,000 times as large as the original difference, although this difference is arbitrary and will fluctuate up and down as the sequences are iterated further. The point is that the two sequences are now apparently unrelated, although they began very close together, and unless they happen to both converge on the same number at some future time, they will continue to fluctuate independently of each other.

What kinds of functions have this behavior and why are they interesting? Certain nonlinear differential equations, nonlinear difference equations, and iterated nonlinear functions (which are similar to difference equations) are the culprits. This is interesting because such equations describe dynamical systems, but unfortunate because the impossibility of precise prediction means that we must think of new uses for such descriptions. They can be used to predict the general form of the system's behavior (its attractors - see next section), but not its precise state (cf., Kellert, 1993). It is also interesting because, and this can defy imagination, such functions are completely deterministic! That is, there is a precisely stateable algorithm for calculating the next value of any iteration on such a function, albeit that the calculation must be made from less than full precision. We are not sampling from a probability distribution here, we are calculating a function value. And yet the function behaves so strangely, even with limited precision calculations (say anything over 10 digits), that precise prediction is impossible, and the result looks random. Some have argued that this means that we can explain the unpredictability and apparent randomness of some behavior of the universe (and of humans), and yet salvage

Einstein's belief that God does not play at dice, by modeling that behavior with such equations. Since they are deterministic they are not truly random, pace Einstein, and yet they defy our puny human attempts at prediction, forcing us into elaborate subterfuges like quantum physics to deal with the apparent randomness.

6. A chaotic model of human memory

Clayton and Frey (1996) presented a nonlinear deterministic model of the dynamics within each trial of a prototypical memory task. They adapted their model from models of animal populations competing in the same niche that use equations similar to the logistic difference equation. Clayton and Frey's model looks like this:

$$x_{t+\Delta t} = x_t + \left(1 - \frac{x_t + y_t}{K}\right) \cdot r_x \cdot x_t - c \cdot x_t \cdot y_t \qquad (6.1)$$

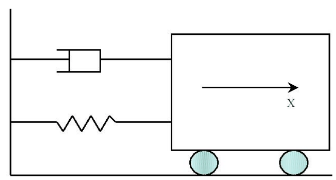
$$y_{t+\Delta t} = y_t + \left(1 - \frac{x_t + y_t}{K}\right) \cdot r_y \cdot y_t - c \cdot x_t \cdot y_t \qquad (6.2)$$

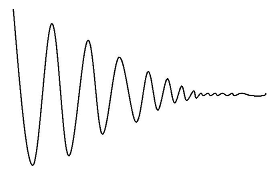
The first part of each equation is analogous to the logistic difference equation, and the last, subtractive, term has been added to introduce competition between the two processes. In this model the main variables, *X* (whose dynamics are described by Equation 6.1) and *Y* (whose dynamics are described by Equation 6.2), represent the dispositions to respond correctly and incorrectly, respectively, in some memory task. For example, they could be the disposition to say that a particular word either had or had not been seen previously in a study list, when the word in fact had been present in the study list. Clearly these are competitive dispositions. In Equations 6.1 and 6.2, $x_{t+\Delta t}$ and $y_{t+\Delta t}$ represent the values of the variables *X* and *Y* at time $t + \Delta t$, where Δt is an interval after time *t*. The two variables increase over the interval Δt proportionally to their own values at time *t* and to their growth rates, r_x and r_y ($1 < r_x , r_y < 3$). *K* ($\approx$7)is a constant that limits the growth of each disposition to the capacity of working memory, and the constant *c* ($0<c<1$) represents directly the competition between *X* and *Y* so that each process is diminished in proportion to the strength of the other. The model shows interesting dynamics, in particular temporary dominance by one process before the other, with a higher growth rate, takes over (and presumably determines the ultimate response), and chaotic behavior by the winning process after it dominates.

7. Taxonomy of Oscillators

Oscillators are systems that change in some more or less regular way. This makes them ideal both for measuring time and for describing processes that evolve in time. A good example is a pendulum like that in a "grandfather clock," which swings back and forth and keeps reasonably good time (has a roughly constant period).

7.1 Damped, unforced, linear oscillator: Example is a little goldmine cart sort of object with mass *m* (gold plus cart) and frictionless wheels attached to a goldmine wall via a spring with stiffness *s*, as pictured below.





Notice that there is another little object attached in parallel to the spring - that is the "dashpot damper," a container filled with oil and a piston that tends to act against the force exerted by the spring on the cart. Imagine we have attached a mule to the cart and we make the mule pull the cart away from the wall to which the spring is attached. When the mule begins to complain that the load is getting heavy, we release the cart. What happens? Of course you already guessed that the cart will rush back towards the mine wall until the spring is compressed somewhat, and then will be pushed away again, and so forth, *oscillating* back and forth until it finally slows to rest, its oscillations *damped out* by the dashpot damper, which provides a force opposing both inward and outward movements (right side of figure). Differential equation for this oscillator:

$$m\frac{d^2x}{dt^2} + r\frac{dx}{dt} + sx = 0 \qquad (7.1)$$

where x is position, t is time, m and s are mass and spring stiffness, as mentioned, and r represents the amount of damping provided by the dashpot. Typically the equation is divided by the mass, m, to put it in standard form:

$$\frac{d^2x}{dt^2} + b\frac{dx}{dt} + cx = 0 \qquad (7.2)$$

where $b = r/m$ and $c = s/m$.

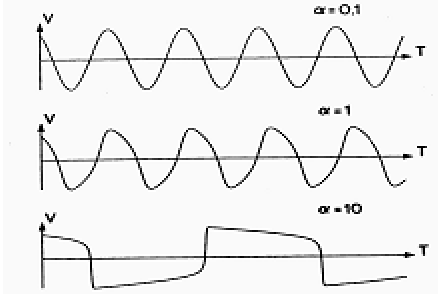
7.2 Oscillators in general can be categorized with respect to three attributes: linearity, damping, and forcing. Differential equations are, among other mathematical properties, either *linear* or *nonlinear*, although the nonlinearity can enter in various ways and have various consequences. Linear differential equations have all terms of either degree one or degree zero (degree refers to the exponents of the derivatives). Equation 7.2 is linear since the derivatives in the various terms all have degree one (don't confuse the notation for the first, second, etc.

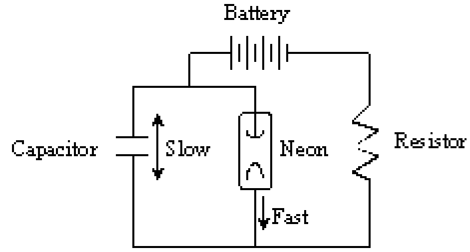
derivative with the exponent of the derivative - the first term in the Equation 7.2 represents the second derivative to the first power!). Squaring *dx/dt*, among other possibilities, would make the equation nonlinear. Linear differential equations exhibit some very special properties (symmetries) that often allow them to be solved exactly, whereas nonlinear equations lack such symmetries and are very difficult to work with. The real world is always nonlinear but often can be approximated usefully by linear equations. *Damping* refers to a force opposing the "natural" oscillation of the system. Without the dashpot damper the system portrayed in the above figure would continue its sinusoidal oscillations forever at the same amplitude. The damping in Equation 7.2 is linear since the damping term, *b(dx/dt)*, is proportional to the velocity of the cart, (*dx/dt*). If it were proportional to the square of the velocity the damping would be nonlinear. Oscillators in the real world are always damped; that is they are not perfectly efficient - they lose, or dissipate, energy. Sometimes the damping can be negligible over the time period of interest, but usually it is not. Finally, *forcing* refers to energy inputs from outside the system of interest. If our mule pulled at and pushed at the cart repeatedly even as it oscillated under the influence of the spring (which stores and then releases the energy of the initial displacement), the oscillation would be forced. In the real world, oscillations are always forced because no system can be completely isolated from the rest of the universe. The forcing can, however, sometimes be so small that it can be neglected. The oscillators with the most interesting behavior are both damped and forced. The interaction of damping and forcing with the basic structure of a system can generates complex dynamics - often chaotic - and a complicated phase space picture, showing qualitatively different behaviors for different parameter regimes.

7.3 The van der Pol Relaxation Oscillator: an unforced, damped, nonlinear oscillator,

$$\frac{d^2v}{dt^2} + \alpha(v^2 - 1)\frac{dv}{dt} + \omega^2 v = 0 \qquad (7.3)$$

where the variable *v* could stand for the voltage across a resistance in an electrical circuit as in the right side of the figure below. Notice that the damping term (the middle term on the left) is what makes the equation nonlinear: because of the function $\alpha(v^2-1)$ multiplying the first derivative (*dv/dt*) this term is of third degree. The relationship between α and ω is crucial to the behavior of the oscillator. When α << ω, the oscillator behaves like a linear oscillator with very small amount of damping, generating a very slowly decaying sine wave, as portrayed in the top of the left side of the figure below. When α ≈ ω, the oscillatory behavior becomes less sinusoidal but is still fairly regular (middle). When, however, α >>ω, the oscillation approaches a square wave (bottom). Notice in the bottom tracing in the figure that each sudden transition (nearly vertical trace) is preceded by a considerable time period during which voltage changes slowly.





Van der Pol (1926) called this a “relaxation oscillation” because each slow change followed by a quick jump resembles the buildup and release of charge in a capacitor with capacitance *C* and “relaxation time” $\tau = RC$ (where *R* is the resistence of the circuit), the time it takes for the capacitor to discharge. In this system relaxation, or the time taken for the “memory” of the previous state to decay, is the important feature, not the restoring force ($\omega^2 v$). A relaxation oscillator is highly prone to phase locking with an external driving frequency (forcing), even one that does not correspond to the natural frequency of the unforced oscillations, while maintaining a relatively constant oscillation amplitude. These properties make such oscillators ideal systems for control of a system that should produce a response of fixed amplitude but variable frequency, for example the heart or a neuron.

Relaxation oscillators can even give rise to chaos! In another famous paper, van der Pol and van der Mark (1927) reported that the neon tube oscillator they were to use later as a model of the heart had a remarkable property. If a forcing oscillator was added to the circuit in series with the neon tube, the nonlinearity of the relaxation oscillator made the whole circuit produce oscillations at frequencies that were equal to the forcing frequency divided by various whole numbers (2,3,4,5,6...), the whole number increasing as the parallel capacitance was increased. This turned out to be a very useful property for electronic engineering. But they also noted an anomaly in this circuit: at certain values of the capacitance “Often an irregular noise is heard in the telephone receivers before the frequency jumps to the next lower value.” (p. 364) This noise was chaos (Kellert, 1993, also noted this). Note that the equation describing this circuit is Equation 7.3 with a simple sinusoidal forcing term added ($E_0 \sin\omega t$), a completely deterministic equation.

Relaxation times in physical systems correspond to “memory” in cognitive systems. That is, a “relaxation” is a time period during which the influence of previous events on current behavior gradually declines. In a time series, this is measured by correlations between current behavior and behavior at various lags. Human behavior is characterized by time series that generate rhythms, orders, sequences, and so forth, even chaos. The relaxation oscillation is a formal construct, represented by the behavior of equations like Equation 7.3, that could be used to model the ubiquitous temporal correlations found in human cognitive and other behavior. Of course such relaxations can happen at several scales in human behavior, and so coupled systems of relaxation equations would be required, making the task of analysis very difficult (even the analysis of Equation 7.2 is very difficult).

7.4 Noisy Oscillators in the Brain

Neurons display relaxation time behavior par excellence: they fire an action potential (spike) when the voltage across the cell membrane in the trigger zone of the soma (around the base of the axon that leaves the soma) exceeds a threshold, and then “gradually” build back up again, usually under the influence of input from other neurons, until the threshold is crossed again and then they fire another spike, and so forth. Hodgkin and Huxley (1952) developed an influential mathematical model of this process based on studies of action potentials in the squid giant axon. The model consists of four first-order differential equations that describe the flow of ions through the cell membrane arising from several physiological mechanisms (e.g., the sodium pump, passive diffusion). The equations are not easy to solve or analyze, and they have been the subject of much study, much as have Einstein's equations of general relativity, even though they do not produce all of the phenomena shown by vertebrate neurons (for example, they don't show adaptation to a constant stimulus - Yamada, Koch & Adams, 1998). Richard FitzHugh was one of those who worked on the analysis of the Hodgkin-Huxley equations (e.g., FitzHugh, 1960). He first tried describing their behavior using phase space methods, such as those discussed earlier in this book, combined with reducing the systems of equations to a more manageable size by holding one or more of the variables of state constant, i.e., assuming that their derivatives were equal to zero. He found that approach inadequate to describe how trains of impulses occur, but noticed that neural action potentials are actually relaxation oscillations similar to those described by van der Pol and van der Mark (1928) for the heartbeat. In an influential paper FitzHugh (1961) described a model of a neuron based on a special case of Equation 7.3 ($\omega = 1$) plus some additional terms, including a forcing input, and showed how it mimicked the behavior of the Hodgkin-Huxley equations. The resulting equations were studied further by Nagumo, Arimoto and Yoshizawa (1962) and have come to be called the FitzHugh-Nagumo model (although FitzHugh, obviously a modest man, called it the Bonhoeffer-van der Pol model).

The FitzHugh-Nagumo equations (in one popular form, at least) look like this:

$$\frac{dv}{dt} = \alpha(w + v - v^3 / 3 + z) \qquad (7.4)$$

$$\frac{dw}{dt} = -(v - a + bw) / \alpha \qquad (7.5)$$

where v is a voltage across the neuronal membrane that represents a fast, excitation process, w is a voltage that represents a slower, recovery process, z is the input stimulus intensity (the forcing term representing input from other neurons), α is the damping constant (greater than $\omega = 1$ so that the system shows relaxation oscillations), and a and b are constants that affect the recovery rate. As I stated earlier, FitzHugh (1961) showed that this system mimicked the Hodgkin-Huxley equations and thus that the neural action potential is a form of relaxation oscillation.

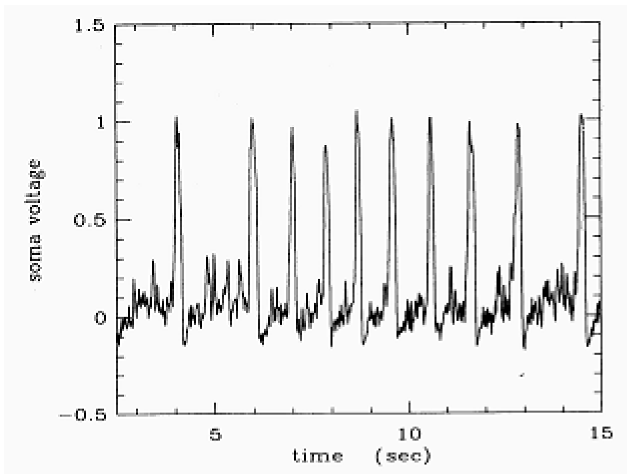
Since 1961 the FitzHugh-Nagumo model of the neuron has been used in many applications in biophysics, including especially in the study of stochastic resonance in excitable systems (see Chapter 22). In this latter application, stochastic (noise) forcing is added to the "fast" Equation 7.4 and deterministic forcing is added to the "slow" Equation 7.5 (Longtin, 1993). This is consistent with the usual case that the noise is at a much faster time scale (comparable to spike duration) than is the signal (comparable to recovery duration). Longtin's (1993) equations look like this:

$$\varepsilon \frac{dv}{dt} = v(v - a)(1 - v) - w + \xi(t) \qquad (7.6)$$

$$\frac{dw}{dt} = v - cw - b - r\sin(\beta t) \qquad (7.7)$$

where $\xi(t)$ is the noise forcing and $r \sin(\beta t)$ is the deterministic forcing (replacing the z in

Equation 31.4). There are also a few other changes in Equations 31.6 and 31.7 from Equations 31.4 and 31.5 but don't worry about them, they don't change the essence of the process. Longtin (1993) showed that these equations behave much like neurons (see figure below). In further papers he has shown how the equations' behavior can be synchronized to periodic forcing (Longtin, 1995a) and demonstrate stochastic phase locking (Longtin, 1995b).



Many more examples of the relevance of relaxation oscillators to dynamical cognitive science can be adduced. I am suggesting that systems of coupled relaxation oscillators (neurons) at many scales are sufficient to produce rhythm, timing, serial order, and many of the other phenomena of the unfolding of human behavior in time, including the recurrence of thoughts analyzed by Crovitz (1970) and the stream of consciousness described by James (1890). Indeed, modeling such phenomena with relaxation oscillators allows us to understand the sense in which such dynamical systems create their own time - the system itself is oscillating, and oscillations are the only means we have of measuring time. In a deep sense oscillations *are* time. Models based on relaxation oscillators would explain why the behavior emerges "when" it does, and why there are correlations between behaviors at many scales, including possibly chaotic behavior. Stochastically-driven and deterministically-driven relaxation oscillators comprise stochastic processes, and some of the modern tools of mathematics and physics can be brought to bear on their analysis. Such models should be particularly well-suited to describing the phenomena of synchronization of neurons and temporal coding of information that are thought to underlie much neural activity (e.g., Singer, 1993, 1994) and also possibly form the basis of human consciousness (e.g., Tononi and Edelman, 1998 - see Chapter 35). Already the influence of relaxation oscillators in neural modeling is pervasive (e.g., Koch & Segev, 1998). The promise is that modeling human cognitive phenomena with relaxation oscillators will allow the existing and future neural models to be coupled to behavioral models to bring dynamical modeling to a new level of complexity and usefulness.