

# Project Report: Agent Based Simulation

Ryan Spangler

December 8, 2009

## 1 Introduction

The problem I have been exploring over the quarter has been: How can a collection of independently behaving yet mutually reactive agents synchronize their firing, given a variety of initial frequencies and phases? Given this question, the simplest basis for a model consists of agents with a phase and a period, some way to perceive the firing of other agents, and rules for adjusting the phase and the period based on the patterns of firing in the agents' immediate vicinity. This was the first model I built, though its limitations led me to embark on another, more ambitious and more comprehensive model based on neural connections and the patterns of excitation and inhibition in an oscillating network.

The idea for the simpler model (which I will refer to as the "sync" model) ended up having been a topic of study in its own right for over fifty years. There were many approaches, and in the framework of basic oscillators with a period and a phase I made a trial of two of the approaches I had encountered in [3]. One was Peskin's model, and the other the Kuramoto model. I discuss these approaches in the "Process" section.

The second model, which I call the "network" model, is a natural outgrowth of the first. The undifferentiated nature of identical oscillators with different periods communicating only through proximity is limited in the range of behavior it is capable of. Mediating the reception of firing through specific links between nodes gives the system as a whole a greater range of behavior. The ideas of using excitatory and inhibitory impulses in particular network configurations stems from ideas in [2].

One of the main questions I was asking was: Under what conditions is synchronization a necessary consequence of the interaction of oscillating

agents? Obviously they must have some means of firing, and some frequency with which to synchronize with the other agents. So really the distinction between a model in which each agent is oscillating randomly and a model where the agents inevitably synchronize (if such a thing is even possible) must lie in the response each agent has to the flashes it sees in the surrounding environment, and the correlation between the flashes it witnesses and the adjustment of its own internal phase and period. If the rules governing this correlation are inaccurate or badly designed, no synchrony will take place. So the question becomes: What rules must an agent follow in order to arrive necessarily to a state of synchronization with its neighbors? This is a difficult question, and the main focus of my endeavors. I was partially successful, and found a simple way to achieve a kind of convergence of frequencies, but was not able to unite the phases of the oscillators. I will cover the range of rules I attempted while working on this problem in the "Process" section.

## 2 Background

Synchronization comes up all the time in nature. In physics the synchronization of waves and matter is known as "entrainment", where natural forces contrive to unite the phases and frequencies of any cycle. In biology cycles and rhythms play a central or supporting role in basically every action the organism takes. The cases of synchrony that are most compelling are those where the mechanism for synchronization are almost inevitable, ie, it takes more energy to avoid synchronization than it does to attain it! The case of the entrainment of the periods of two grandfather clocks in the same room is a great example. No one knew why it happened, but it turns out that energy was passing between the floorboards, almost imperceptibly, but just enough to clash with the swing of the other clock *only if the swing was out of phase*. If the swings were in phase, the mutual effect on the floor boards was in harmony, so no net displacement or force was transmitted. This case and others like it are striking because the eventual synchronization is inevitable, or in the language of dynamics, the state of synchronization is an attractor in the state space. What about synchrony is necessary in this way?

Biological synchrony is astounding for the precise coordination and elaborate mechanisms of control it employs to realize oscillatory synchronization. From the millisecond range periods of neuronal firing to the day long circadian rhythms to the seasonal cycle of plants and trees, cycles appear every-

where in biology. I focus on neural synchrony, but I could have equally well have chosen any workable biological oscillation.

I draw inspiration from Buzsaki's description of the importance of inhibition in attaining neural synchrony. He points out that a network with only excitatory connections could only ever cascade into higher and higher levels of excitation until saturation, an outcome which is inherently uninteresting as all inputs arrive at the same result, a complete homogenous excitation of the entire network. It is the role of inhibition which modulates excitation, sculpts it into a form which has meaning. Inhibition creates the structure of the signal. One example is when a signal triggers multiple neuronal assemblies at once and the assemblies themselves inhibit one another when excited. This results in the assembly with the strongest affinity to the signal repressing the activation of the other assemblies, a kind of "winner-take-all" approach.

This idea of inhibition controlling a signal also has applications in cell biology and gene translation. A particular gene will be triggered for production which controls the expression of some other gene, thereby repressing its target's production. The expression network of a given genome is a complex and tangled affair, but it reduces to the same basic operations of expression and repression playing the roles of excitation and inhibition in a neural network. General network topography and network "motifs" is an idea explored in great detail by Uri Alon in his [1]. In this book, he does an analysis of network motifs as found in biological networks of every scale and timeframe and compares these to the network motifs found in an idealized random network. By determining which motifs occur with far more likelihood than would be given by chance, he is able to identify which motifs have relevance to biological processes.

It is an encouraging idea that an understanding of the dynamics of general networks can lead to insight into a huge variety of different phenomena. There are a staggering number of applications of network based systems and synchronous dynamics. Also, a general understanding of the nature of successful biological distributed and decentralized network systems has proven elusive to this day. It is my aim to tackle the issue head on.

### 3 Process

The process was an adventure of trial and mishap, temporary success and painful disappointment. The tribulations of the sync model led through many attempts naturally into my next conception of synchronization, the network model.

For the sync model, as mentioned before, there were many previous models using the same basic idea of a collection of oscillators with a period and a phase that I encountered in the literature. I picked two of the most plausible ones and made a quick sketch of each in the basic framework I had constructed. One was Peskin's model of a phase curve that arrives fairly quickly somewhere near threshold and perches there, waiting for the next impulse to push it over. In this way, firings trigger a cascade of other firings, lying in wait just below threshold. This was simple, and worked, but seemed not quite as essential of a process as say, the grandfather clock example. The grandfather clock example works because the forces each agent was exerting due to their periodic behavior were precisely those forces that were nudging each phase towards the other's. In this way the forces were *coupled*, and the synchronization was unavoidable. This is the kind of synchronization I want to capture, and this is why I was compelled beyond Peskin's more gimmicky model.

The next model I inspected was the Kuramoto model. This was based in a nonlinear differential equation relating the change in phase of a particular oscillator with the sine of the sum of the differences of its phase with the phases of all of the other oscillators in the system. Because it was the sine, this change is greater if the phases are more divergent. Hence, like the grandfather clocks, the phases change if they are out of phase with one another and remain in cycle if all of the phases line up. Also, the magnitude of the resultant vector represents how aligned all of the vectors were, leading the phase towards a common meeting ground if there is one, and not having much of an effect if all of the phases were different. The shortcoming of this model, in my opinion, is that each oscillator requires knowledge of the phase of all of the other oscillators. This kind of complete knowledge is not realistic in terms of how most oscillators in the world behave. Eventually I moved beyond these simulations of other people's models into the design of my own.

While building the sync model there were a variety of decisions that had to be made. In particular, one of the first was to determine what the sight radius of each agent would be. At first I had a fixed sight radius,

where each agent had the same radius which was controlled by a single slider. This led to a situation where clusters of nodes would have many neighbors, and isolated nodes would have few or even no neighbors. Obviously this was undesirable, as the effect would have to emerge from the integration of its receptivity to the flashes from several other agents at once. Having a single agent alone, or responding to only one other agent, was not sufficient to demonstrate the emergence of synchronization from the interactions of and response to many other agents. Since the agents were fixed in space, I was able to have each agent incrementally expand its personal sight radius until it reached some minimum level of neighbors. This minimum I had controlled by a slider, and under this approach, though each oscillator had a different number of neighbors, some more, some less, all of them had at least a minimum number. Many other factors led to an open-ended phase of tweaking the various parameters to the model. I could almost say that I spent more time tweaking various parameters and seeing what change in the results I would receive than I did actually building the model itself. I would say that setting up the model was establishing an arena in which the effect of the various parameters could be explored.

For the network model, the first challenge I came across was building the network of links between nodes. Building a set of totally random links would leave certain nodes with no links at all! In order to build the network from the ground up I used a modified form of the preferential attachment approach, where a link was established randomly between two nodes weighted towards nodes with a higher number of pre-existing attachments. To get around results where certain nodes lacked any connections at all, I gave each node a minimum number of outgoing connections, restricting the random connection to the terminal of these outgoing connections. This way I achieved a nice balance of in links and out links and maintain a kind of "small-worlds" scale free structure on the link structure of the entire network. The next was finding some way to imbue the network as a whole with some base level of excitation. In order to facilitate this, I arranged a single row of input neurons which themselves contained no in-links, only out-links, and which oscillated at a given set rate of frequency. For now, these inputs fire each at a particular frequency, with no change.

## 4 Summary

As a system, the sync model uses a common, extremely abstracted structure where each agent possesses only a period and a phase. The agents are arrayed randomly throughout the model space, and each agent's neighbors are determined by proximity. The network model extends this by making explicit directional links between various oscillators, and giving the links themselves a strength. In addition, each node is given a sign that indicates whether its influence on the nodes it is connected to is excitatory or inhibitory.

In terms of the rules of the agents, though I tried many methods, the one I settled on was a simple approach where each agent would count the number of flashes that it witnessed between one flash and the next, and based the modification of its frequency on that. My idea was that if an oscillator witnessed many flashes between two of its own flashes, it was probably going slower on average than the oscillators that surrounded it, and if it witnessed only a few or no flashes in this time, it was probably going faster (had a shorter period) than its neighbors. This can be appealed to intuition by considering that if all oscillators were flashing at exactly the same frequency, the number of flashes one agent would witness in between two of its flashes would be equal to the number of neighbors it had. Based on this, any deviation of the flash count from the neighbor count was an indication that it was out of synchrony with the periods of its neighbors. In this way, the count of flashes was compared against the number of neighbors the oscillator possessed. If the flash count was greater than the number of neighbors, the period was reduced by a fixed incremental amount. If the flash count was lower, the period was increased.

Here is the main procedure:

```
to flash
  ifelse neighbor-count > surrounding-flashes
    [ set period period + adjustment-factor ]
    [ set period period - adjustment-factor ]
end
```

The main difference between the sync model and the network model is that in the sync model, each agent responds to the flashes it sees in exactly the same way, depending only on the radius of sight that particular agent has. In this way, the connection pattern between agents in the sync model is spatial

and proximity based. In the network model by contrast, the only connections that exist are explicit links between two nodes. Two nodes may be very close that do not see their respective flashes, and two that are very far away could respond immediately, all depending on the pattern of links between all of the various agents. This gives the network model a greater degree of variability and distinctiveness to their connection structure than is attained in the sync model. The other main distinction of the network model is that it has two main classes of nodes, excitatory and inhibitory, with each having exclusively one or the other affect on the other agents they are linked to. The synchronization of the two subnetworks is considered independently, with the interneuron group (as the inhibitory network is referred to) oscillating at a distinct frequency and phase from the excitatory nodes.

The nodes in the network model work in a similar way to the agents in the sync model (as it did grow out of that model), but there are some important differences in their function and the rules they adhere to. Whereas in the sync model the agent only took stock of its environment and made all of its adjustments each time it fired, in the network model some adjustment to the phase is made each time it receives an impulse from an incoming link. As well as an adjustment to the phase of the node, an adjustment is also made to the weight of the incoming link based on where in its phase the node is when it receives the impulse. In addition, each node has a sign, either 1 or -1, which determines whether the signals it sends along its outgoing links are excitatory or inhibitory. This sign is fixed during the lifetime of the node, each one is born either a positive or a negative.

One of the important features of this network system is that each node is always firing at some rate, and the difference between which nodes are considered "excited" and which are considered "baseline" or "quiet" is one of degree, not of presence. It is that the excited nodes will be firing at a higher frequency than the ones that are repressed, not that the repressed ones will be totally silent. This is in harmony with the nature of biological neurons, who must always be firing to some degree, and distinguish themselves by their enthusiasm.

The network model is also unique in that there are two distinct sets of agents (though they are all of the same breed). One, the basic nodes, are as described. The second, referred to at the end of the "Process" section, are a single row of input nodes which themselves have no incoming links. Each fires at a constant rate unperturbed by any signal or change. Eventually, these inputs will be tied to some kind of environment, with the addition

of an output row as well that affects the same environment. The shared environment could take any form, as long as it had sufficient dynamics to instill activation in the input nodes, and be able to be influenced by the activity of the output nodes. This is discussed further in the "Transfer" section.

## 5 Testing

In order to test the models, I constructed a variety of monitors and graphs in an attempt to understand the dynamics of the system. In each run I would modify a single parameter and see how the behavior would change. Sometimes, the fact that the behavior didn't change at all was a useful data point. Certain parameters were extremely sensitive and would give wildly different results with minor changes, while others seemed to have no effect on the dynamics of the system at all. Others still would have no change until a certain critical value was reached, whereupon everything would change at once. All in all I was very surprised by many of the effects of the parameters and found the monitors and graphs essential in getting a picture of what was going on throughout the system.

In the sync model, I had a plot of how many agents were firing at the same time. In the case of pure chaos this line would appear as a random walk. If synchronization were to occur this would arrange itself into a series of pulses, with spikes occurring during the simultaneous firings and valleys in between. Also, the amplitude would rise as synchrony was emerging until it reached a plateau where most of the agents were firing simultaneously. Another plot that turned out to be useful was a compound graph tracking both the lowest period and the highest period. In a disorganized situation, these values would drift apart from each other as each extreme spilled into the surrounding range. In a case of synchrony, these values would converge and remain constant, signifying the unification of all of the frequencies. Also, there was a histogram of the periods of the various oscillators. In chaos, the periods would be evenly distributed around the space. In synchrony, the periods would cluster around a single value, having a single large peak in the middle. These turned out to be the main useful measures of how well the system was synchronizing.

For the network model the measures are much the same, though since it has not progressed as far as the sync model the system is not yet in place to



have anything to test adequately.

## 6 Transfer

In the sync model, as time went on, I was able to get the periods of a randomly distributed collection of oscillators to converge towards a value, meaning mainly that the difference between the maximum and the minimum periods decreased. They did not decrease towards a stable value however, but increased asymptotically towards some invisible limit. As time goes on, the center drifts less and the extremes draw in, but they do so slowly. Also, the periods converge, but the phases do not. This method allows for frequencies to stabilize towards a single value, but since each agent only keeps track of how many flashes it sees in between its own flashes, it is not paying attention to *where* in its phase these flashes take place. In order for some kind of phase synchrony to occur each agent must take into account where in its phase each flash took place and make an adjustment towards some kind of common ground. In this way the model could benefit from something like the Kuramoto approach, and add together all of the phase vectors (considering phase as a unit circle about the origin with an x and a y component) of the when the surrounding flashes occurred. The direction of the resulting vector shows in what way the phase of the observing agent needs to be corrected, and the magnitude of the vector shows how much agreement the surrounding flashes are in, ie, the magnitude will be greater if more of the surrounding phase vectors are pointing in the same direction. This would allow the oscillators to attain some kind of phase synchrony as well as period synchrony.

The question remains why the periods converge, but not precisely. For this my only explanation is that since all of the periods are constantly adjusting, it is hard for all of them to be at the same period at the same time. However, the histogram of the agents' periods did become more extreme, with more periods at the center of the cluster relatively as time went on.

For the network model, I have hope that the variety of structures that a mix of inhibitory and excitatory impulses possess will lead to a finer grained and more broad class of synchronous phenomena. I want first to attempt to replicate the simplest example of a synchronous circuit based on mutually firing excitatory and inhibitory nodes, and go from there. I will be basing the initial design upon some principles sketched out in "Rhythms of the Brain",

and modify it to some kind of working system.

There has been previous discussion about the input nodes and their patterns of activation. It is my eventual aim to hook these inputs into some kind of signal from the environment so that their frequencies are tied to some kind of internally consistent external phenomenon. This would enable the nodes that are fed by them to generate a persistent signature of the dynamics of the input signal. In general, the way I see this model actually working is to have an input signal which changes at a rate much slower (possibly ten times, or an order of magnitude slower) than the average rate of firing of the nodes. This gives the network as a whole the chance to synchronize with respect to the pattern of excitation generated by the row of input nodes. In order to synchronize, the influence of various links to one another will have to be adjusted, and in particular the inhibitory contribution to the activity of the network will be adjusted to clarify the distinction between which nodes are contributing to the synchronous cluster and which are sitting out, oscillating only at baseline frequency. Once synchronization is established, the input signal will already be shifting to another pattern. In this way the network cycles through a series of coherencies and disruptions to that coherency as the input signal shifts in response to some external phenomenon. During the coherencies, fleeting alliances are made, a minor persistent adjustment towards cooperating in the way the nodes are now cooperating takes place. As the network cycles from state to state, from synchronous coherency to asynchronous coherency, each group of interacting nodes adjusts towards one another in a way which also retains the signature of all of the other coherencies it has contributed to. After time, those patterns which repeat themselves will leave their mark upon the network as a tendency to snap into a corresponding coherency. And since no event happens exactly the same way twice, these mini-alliances will represent flexible building blocks of larger coherencies, able to decompose any novel experience into a new and unique interaction of clusters of organized network subunits. Given such a network, and adding a row of output nodes as well which have the potential to affect the pattern of the input signal through an intermediary shared environment (ie, the output affects the environment that the inputs are sensitive to), it would begin to resemble the relationships of a plausible biological system.

This is my aim of course. I did not have time to take the model to this point, but I intend to continue work on it until I have attained some form as described above.

## 7 Conclusions

The biggest surprise to me turned out to be that the problem of synchronization was such a difficult one. I had assumed I was doing what was the simplest kind of coordination between agents I could think of, and it turned out to be a problem that people had been wrestling with for fifty years. Synchronization is not a simple problem, but though it is challenging it is also worthwhile. I am inspired in the direction I have found towards a fine grained neural signal approach to synchronization.

I am dissatisfied with the performance of the sync model, with its slow convergence and failure to synchronize in terms of phase. I hope to find a mechanism where rather than converge slowly on a result there is more of a phase shift, with things falling quickly into synchrony. The network model holds more promise in this regard, where the potential lies in the ability to transition from one synchronous state to another, if the condition of robust transition to synchrony is achieved.

Netlogo turned out to be a pretty amazing language. I have a background with lisp and scheme, which netlogo resembles in ways, so I found it a natural transition. I found all of the ways in which the language was tuned for handling collections of things to be smooth and pleasant, especially in comparison to some of the more cumbersome languages. Netlogo was obviously designed for expressiveness, and I found many of the things I was trying to do there was already a function for (like finding the distance or direction from one agent to another, selecting all agents in a certain radius, etc). Also, I think it was a good choice to draw from and build upon the basic turtle analogy from classic logo. It offers a helpful perspective on the agents as having an identity, and encourages the programmer to think from the turtles' perspective. This lends itself well to agent based modelling which requires a local understanding and interaction of the agent and its immediate environment to translate to some kind of emergent behavior of the entire system. Thinking and programming from the point of view of the agent made doing very complex things seem simple, or as simple as possible.

All in all, much was learned, many mistakes were made, and a measure of accomplishment was attained. In a way, all I have really done is create a greater deal of work for myself in the future. But now I have some kind of direction, and I look forward to following it.

## References

- [1] Uri Alon. *An Introduction to Systems Biology*. Chapman and Hall, Boca Raton, FL, 2007.
- [2] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, New York, NY, 2006.
- [3] Steven Strogatz. *Sync*. Hyperion, New York, NY, 2004.