

Synchronicity of Patterns

hans opstaele (febr-2010)

Abstract

This text explores *synchronicity*, the chance that patterns occur simultaneous on two different locations.

1 Problem

Synchronicity is defined as a Jungian term, created by the Swiss psychologist Carl Jung, for a meaningful coincidence that has a low probability of being a random or chance event.

'*Synchronizing Signals*' would be a more conventional title for this document, since it is about signals, networks and communication. However, the word '*Synchronizing*' implies that something 'approaches' a certain state, just as a phase-locked-loop approaches synchronicity of a clock. This is, by design of the algorithm impossible, since time is a discrete variable associated with the distance in a network. Instead, this document explores the *probability* that patterns are synchronous and how to (re-)construct synchronicity between locations.

The word '*Patterns*' in the title refers to a periodic signal, but it could be used to refer to geometric patterns¹ or any other representation of a state.

'Synchronicity of Patterns' can be summarized as *maintaining an undefined periodic signal or geometric pattern exactly equal on two different locations in a network*.

1.1 Overview of the Algorithm

Talking about simultaneous events implies certain events that are not simultaneous. There are many parameters to be defined because 'Delays' and 'Signals' are very broad concepts. The text is limited to the following conditions:

1. Delays are represented as a **distance** between communicating nodes in a network.
2. A distance has an associated signal **component** that is sampled at a certain time. For example, symbol x_n is the component x sampled at time n .
3. The two communicating nodes grow apart, or *unfold*, if their state is synchronous. They approach (or *fold*) if synchronicity disappears.
4. The pattern is a period signal with some *invariant properties*. This invariant allows us to validate the desired synchronicity and put a value on its quality.

¹The FUBAR-project translates the concepts that are introduced here to geometric patterns (www.4cko.be)

Items 1, 2 and 3 are discussed in section 2. Item 4 is introduced in section 3. The 'Conclusion' gives a more generalized view of the problem and some future work.

2 Communication on the Network

2.1 The network

The network is constructed starting with two nodes and interleaving each node in every generation (see figure 1). Node 0 is the 'sender' and node 1 is the 'receiver' but, since they are to be synchronous, the difference is sometimes hard as both can interact. Sender and receiver, node 0 and 1, have a relation comparable to the endpoints in a transmission line where both extremes have interference from reflection. To put it in other words, the endpoints are not merely an active sender or passive receiver. This remark aside, the document will stick to the node 0 as the **sender** and node 1 as the **receiver**.

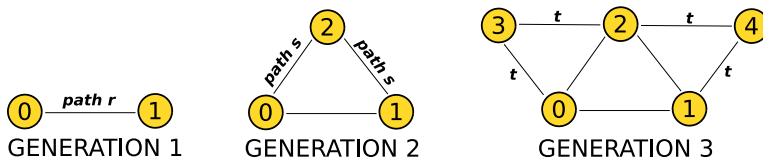


Figure 1: First three generations of the network

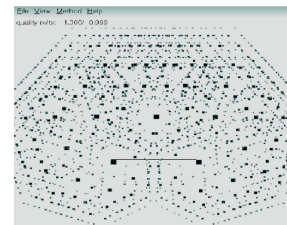


Figure 2: Generation 12

The network has several pathways between two nodes: each paths relates to a different type of signal, called *components* (section 2.3). Every node samples those components. For instance, at time n the following information is sampled:

path r(0:1) The receiver sees sample r_{n-1} when r_n is at the sender. Because of the link, the receiver sees the previous sample.

path s(0:2:1) The receiver sees sample s_{n-2} when s_n is sent by node 0. There are two links to pass between nodes 0 and 1.

path t(0:3:2:4:1) There are four links in this path: the receiver sees t_{n-4} when t_n is sent.

The number of links doubles each generation (1,2,4,8,16,...).

2.2 The dynamic network

The lower pathways can be severed to increase the distance between sender and receiver, This process is called *unfolding*. The algorithm is designed to unfold the network if nodes 0 and 1 are synchronized, thereby creating larger distances (see figure 3).

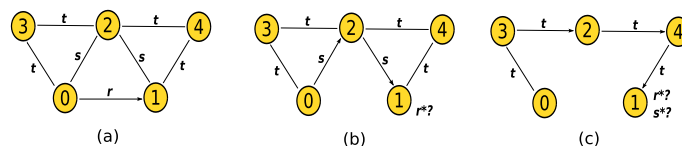


Figure 3: Unfolding to create more distance in the network

Synchronicity is simple in a folded network: a simple linear extrapolation can obtain a good estimate r_n^* at the receiver for sample r_n at the sender. Synchronicity is even easier when the samplerate is very high: the consecutive samples are almost equal at that case and $r_n^* \approx r_{n-1}$.

$$r_n^* = r_{n-1} + \epsilon \quad \text{simple synchronicity}$$

Every extrapolation has, even with very high samplerate, an associated error ϵ . This error builds up with larger distances when the network unfolds.

The presented algorithm is not an extrapolation-method; For this reason, the term 'reconstructed value' is used instead of the more common 'estimate'. This terminology is important to keep the distinction between the purpose of the two methods².

2.3 Signal components on the paths

Each path r, s, t, \dots has another type of signal. As the length of a path increases, the signal is averaged over a longer timescale. This system is based on the *HAAR wavelets* proposed by Alfred Haar (1909).

Path r : (length 1)	r_n (True signal)	
Path s : (length 2)	$s_n = \frac{r_n + r_{n-1}}{2}$	
Path t : (length 4)	$t_n = \frac{s_n + s_{n-2}}{2}$	$= \frac{r_n + r_{n-1} + r_{n-2} + r_{n-3}}{4}$
Path u : (length 8)	$u_n = \frac{t_n + t_{n-4}}{2}$	$= \frac{r_n + r_{n-1} + r_{n-2} + \dots + r_{n-7}}{8}$

2.3.1 Vanishing components

Higher components, u, v, w, x, \dots vanishes for periodic signals. This property is illustrated below. A signal with period 8 is split into its components. It vanishes after the third component:

$$\begin{aligned}
 r &= \{\dots, 0, -1, -2, -1, \mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{1}, \mathbf{0}, -1, -2, -1, 0, 1, 2, 1, \dots\} \\
 s &= \{\dots, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{3}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{3}{2}, -\frac{3}{2}, \dots\} \\
 t &= \{\dots, -1, -\frac{1}{2}, \frac{1}{2}, 1, 1, \frac{1}{2}, -\frac{1}{2}, -1, \dots\} \\
 u &= \{\dots, 0, 0, 0, 0, 0, 0, \dots\}
 \end{aligned}$$

For example: $s_0 = \frac{r_0+r_{-1}}{2} = \frac{0-1}{2} = -\frac{1}{2}$ and $u_0 = \frac{t_0+t_{-4}}{2} = \frac{-1+1}{2} = 0$.

²Validation will also be different for the two methods: synchronicity cannot be cross-validated but uses *invariants* for optimization

3 Patterns and the Reconstruction of Synchronicity

3.1 Using the Vanishing Components

Because a long-term average tends to be stationary, a sufficient high component, say 'x', is assumed to be constant (see figure 4).

$$\text{Stationarity of components } \forall n : x_n \equiv m$$

With this stationarity and reversing the HAAR system (table 3.1), it is possible to reconstruct the real signal r , assuming the constant component x . As shown in figure 4: if the components are settled, it is possible to unfold the network and reconstruct the signal from the higher components.

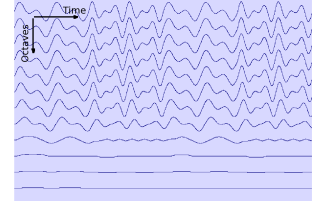


Figure 4: Components of a pattern (top→bottom)

Forward -	Reversed HAAR System
r_n (True signal)	$x_n^* \equiv 0$ (Assumed $\forall n$)
$s_n = \frac{r_n + r_{n-1}}{2}$	$w_n^* = w_{n-32}$
$t_n = \frac{s_n + s_{n-2}}{2}$	$v_n^* = 2w_n - v_{n-16}$
...	...
$x_n = \frac{w_n + w_{n-32}}{2}$	$r_n^* = 2s_n - r_{n-1}$ (Recovered signal)

3.1.1 Invariants and Signal grooming

This section is dedicated to the indicator variable that triggers unfolding: a basis to decide if r_n^* is sufficiently synchronous with r_n at the sender³. The indicator used here is an invariant embedded in the signal. An example of a signal with frequency f and its invariant inv are:

$$x = 2\pi f n \quad ; \quad r_n = \begin{cases} \sin(x) + \sin(\frac{x}{2}) & \text{for symbol '0'} \\ \sin(x) + \sin(\frac{2x}{3}) & \text{for symbol '1'} \end{cases}$$

$$inv_n = (r_n)^2 + \left(\frac{r_n - r_{n-1}}{4\pi f}\right)^2 \quad \text{with} \quad 0 < inv_n < 2.1 + 2\cos\left(\frac{x}{2}\right)$$

The periodic pattern of signal r is created from symbols '0' and '1'. Both functions have the same period 4π (figure 5) and are derived from a fundamental bass $\cos(\frac{x}{2})$.

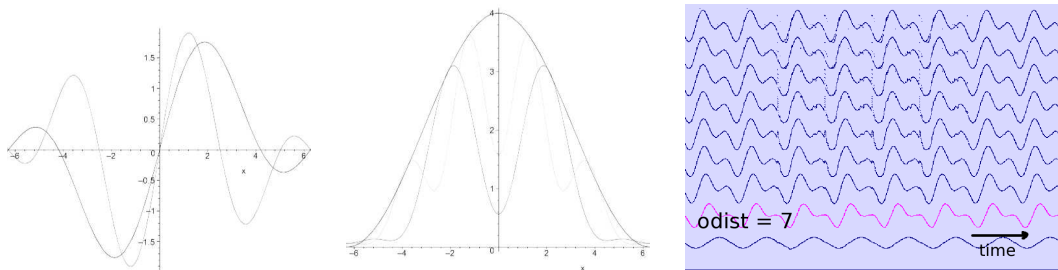


Figure 5: (left) the modulation signal r_n ; (middle) the invariant ; (right) grooming r_n^* with the invariant

³In theory, it is possible to reflect the data and reverse the role of sender and receiver for a cross-validation.

The invariant is derived from two consecutive samples. The distance between inv_n and $2 + 2\cos(\frac{x}{2})$ can be used to detect the current symbol and *groom* the signal at the receiver (figure 5).

To summarize: vanishing components are important to reconstruct r_n^* at node 1, but this is in itself not sufficient: A rounding error from the system or even the unfold operation can provoke an avalanche of errors and break synchronicity.

The signal is corrected, or groomed, using the invariant to provide fault tolerance and avoid errors building up at the receiver.

4 Conclusion and Future Work

The text explores synchronicity in a network: creating an exact duplicate of a pattern on two different locations. Patterns itself are not conventional forms of communication as they cannot be expressed in bits and bytes. They are rather periodic signals or geometric patterns with invariant properties.

The latter is interesting future work: Statistical methods like Kriging provide spatial estimations based on a 'Hypothesis of Stationarity' comparable to the stationarity of components and the invariant used in this text. This could be used to create a visual synchronicity: making two distant places look the same.