Are asynchronous cellular automata essentially periodic? A conjecture based of the theory of hybrid systems[1]

James NUTARO[2]

Anstract – *Asynchronous cellular automata, synchronous cellular automata, and a special class of hybrid automata are three, essentially interchangeable, forms of a single class of systems. This article expresses these seemingly distinct formalisms as mathematical systems, and from this basis shows that the former are homomorphic simplifications of a particular class of hybrid automata. It is further conjectured that, as a consequence of this relationship, every asynchronous cellular automaton is essentially periodic: like their synchronous counterparts, asynchronous automata appear to settle into a very regular, predictable trajectory if they are observed for a sufficiently long period of time. This conjecture is based on recent work in the theory of differential automata with constant derivatives. The paper concludes with a discussion of problems posed by the hypothesis and of its consequences for discrete event systems that are schedule preserving.*

Keywords – asynchronous cellular automata, hybrid systems, discrete event systems.

1. Research sponsored by Laboratory Directed Research and Development program of Oak Ridge National Laboratory(ORNL), managed by UTBattelle, LLC for the United States Department of Energy under contract no. DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

2. Oak Ridge National Laboratory, Oak Ridge, TN (nutarojj@ornl.gov).

INTRODUCTION

Self-clocked cellular automata are similar to synchronous cellular automata, of which the Game of Life [Gardner, 1970] is best known, but differ in that each cell evolves at its own rate. The rate is expressed as a duration, which can be any real positive number, for the cell's state. A fixed duration is assigned to each cell, and the cell changes state at intervals equal to the duration. A transition rule, which is common to all cells, assigns the transitioning cell a new state that is determined by the current state of itself and its neighbors.

To be concrete, consider a one dimensional, self-clocked cellular automaton with n cells indexed 1, 2, …, n. The neighborhood of cells with index $1 < k < n$ are the cells at k-1 and k+1. The neighbors of the leftmost cell at k=1 are the cells at locations 2 and n; the neighbors of the rightmost cell are at locations n-1 and 1. The subscript k,l is used to denote the left neighbor of cell k and k,r the right. Three pieces of information are maintained for each cell: the time e_k that has elapsed since its last transition, its discrete state q_k, and its duration P_k. The elapsed time begins, and remains, in the interval $[0,P_k]$. The transition rule δ maps the state of a cell and its neighbors into a new state when $e_k=P_k$. A three step procedure is used to simulate the cellular automaton:

1. Find the time t_N until the next transition by finding the smallest P_k-e_k.
2. For each cell add t_N to e_k. If $e_k=P_k$ then calculate $\delta(q_k,q_{k,l},q_{k,r})$ and save the result to ν_k.
3. For each cell, if $e_k=P_k$ then set q_k to ν_k and set e_k to zero.
4. Go to step 1.

Self-clocked cellular automata can simulate asynchronous cellular automata that have a fixed update order. Without loss of generality, consider n cells that are updated from left to right. To simulate this update order, the duration of each cell is n and the elapsed time is initially $e_k=n-k$. The cell at position k has its first update at time k. Subsequent updates occur at times k+n, k+2n, … so that in every n time steps each cell is updated just once and in the desired order.

To illustrate this construction, consider an automaton with three cells. The discrete state of each cell is 0 or 1, the transition rule is

$$\delta(q_k, q_{k,l}, q_{k,r}) = \begin{cases} 0 & \text{if } q_{k,l}=1, q_k=1, q_{k,r}=1 \\ 0 & \text{if } q_{k,l}=1, q_k=1, q_{k,r}=0 \\ 0 & \text{if } q_{k,l}=1, q_k=0, q_{k,r}=1 \\ 1 & \text{if } q_{k,l}=1, q_k=0, q_{k,r}=0 \\ 1 & \text{if } q_{k,l}=0, q_k=1, q_{k,r}=1 \\ 1 & \text{if } q_{k,l}=0, q_k=1, q_{k,r}=0 \\ 1 & \text{if } q_{k,l}=0, q_k=0, q_{k,r}=1 \\ 0 & \text{if } q_{k,l}=0, q_k=0, q_{k,r}=0 \end{cases}$$

and the initial values of the discrete states are $q_1=q_3=0$ and $q_2=1$. Table I shows six time steps of this cellular automaton; in each step, the cell that changes its state is enclosed in a box.

Time	k=1	k=2	k=3	Rule
0	q=0,e=2	q=1,e=1	q=0,e=0	R7
1	q=1,e=0	q=1,e=2	q=0,e=1	R2
2	q=1,e=1	q=0,e=0	q=0,e=2	R7
3	q=1,e=2	q=0,e=1	q=1,e=0	R2
4	q=0,e=0	q=0,e=2	q=1,e=1	R7
5	q=0,e=1	q=1,e=0	q=1,e=2	R2
6	q=0,e=2	q=1,e=1	q=0,e=0	R7

Table I. Six time steps of a self-clocked cellular automaton that simulates a left to right update order.

Self-clocked cellular automata and asynchronous automata with fixed update order are widely used to model multi-agent systems. Applications can be found in fields as diverse as ecology, biology, sociology, and economics. Asynchronous cellular automata are attractive for these applications because they can model asynchronous interactions that occur in the system of interest [Cornforth *et al.*, 2002; Green *et al.*, 2001].

When compared with its synchronously updated counterpart, an asynchronous cellular automaton can exhibit radically different behaviors [Bersini and Detour, 1994; Schönfisch and de Roos, 1999; Stark and Hughes, 2000]. It is surprising then that self-clocked cellular automata seem to share an important property of their synchronous relatives. A synchronous cellular automaton defined over a finite space is periodic because it has a finite number of states. A self-clocked cellular automaton, on the other hand, has an infinite number of potential states because of the elapsed time that is intrinsic to every cell. None the less, it is conjectured here that all self-clocked cellular automata must settle into one of a finite number of limit cycles, the particular choice depending on its initial state.

This conjecture emerges from the theory of hybrid systems, and specifically from the recent work by Matveev and Savkin [2000] that describes the limit cycles of differential automata. The argument begins by showing that self-clocked cellular automata are instances of a class of differential automata that have five critical properties. The theory of hybrid systems shows that these properties are sufficient for the automata to always converge to one of a finite number of limit cycles. The necessary conclusion is that self-clocked cellular automata converge eventually to a limit cycle. Moreover, because self-clocked cellular automata can simulate deterministic asynchronous cellular automata, it is concluded that all deterministic cellular automata – synchronous and asynchronous – are essentially periodic.

1. Brief review of differential automata

Differential automata are finite state automata that have a set of differential equations associated with each discrete state. Discrete events, which change the system's discrete state, occur when the automaton's continuous trajectory encounters an event surface. By changing the discrete state, the event causes the system to select a new set of differential equations which govern its motion away from the interrupting event surface. Figure 1 illustrates the trajectory of a differential automaton with a single continuous variable.

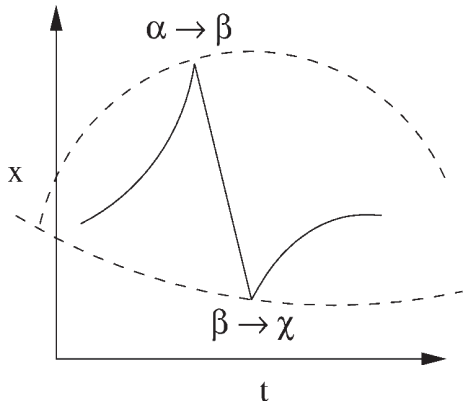


Figure 1. A trajectory of a differential automaton with a single continuous variable × which follows the solid curve, two event surfaces shown as dashed curves, and two discrete transitions $\alpha \rightarrow \beta$ and $\beta \rightarrow \chi$.

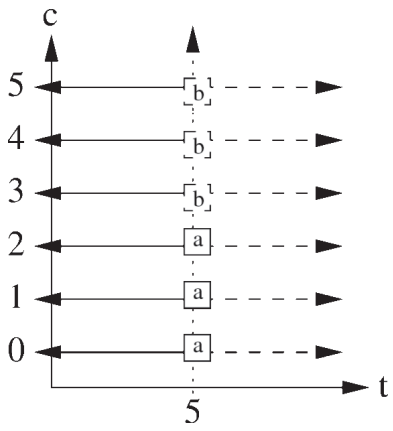


Figure 2. Illustration of the trajectory z that is defined by Equation 4. The value a is indicated with a solid line and b with a broken line.

Differential automata measure time in two dimensions. The first dimension is the real number line, and it measures time by seconds, minutes, and hours. The second dimension counts events that occur at an instant, neatly arranging instantaneous events by the order of their occurrence. Time is modeled by the set $\mathbb{R} \times \mathbb{N}$ and the advance operator $\triangleright$ moves a time (t,c) forward by (t',c') with the rule

$$(t,c) \triangleright (t',c') = \begin{cases} (t+t',0) & \text{if } t' \neq 0 \\ (t,c+c') & \text{if } t'=0 \end{cases} \quad (1)$$

The advance operator is not commutative and it is not associative. Time is ordered first by t and then by c: specifically,

$$(t_1,c_1) < (t_2,c_2) \Leftrightarrow ((t_1 < t_2) \vee (t_1 = t_2 \wedge c_1 < c_2)) \quad (2)$$

$$(t_1,c_1) = (t_2,c_2) \Leftrightarrow (t_1 = t_2 \wedge c_1 = c_2) \quad (3)$$

A trajectory of a hybrid system is a function from $\mathbb{R} \times \mathbb{N}$ to its set of states. The trajectory, by virtue of being a function, has a single value at each point in time; the second dimension of time allows for instantaneous changes in the discrete state of the system. For example, the trajectory

$$z((t,c)) = \begin{cases} a & \text{if } (t,c) \leq (5,2) \\ b & \text{if } (t,c) \geq (5,3) \end{cases} \quad (4)$$

has this property; it is illustrated in Figure 2.

A differential automaton is a system with a finite set Q of discrete states, a set $\mathbb{R}^m$ of continuous state vectors, and two functions that describe its dynamic behavior. The differential function $f : \mathbb{R}^m \times Q \rightarrow \mathbb{R}^m$ describes how the continuous variables evolve between discrete events. The transition function $\Phi : \mathbb{R}^m \times Q \rightarrow Q$ defines the event surfaces and their effect on the discrete state. The continuous state vector × satisfies

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), q) \tag{5}$$

at each instance of real time for which q is constant.

When q changes at time (t,c), its subsequent value at time (t,c+1) is

$$q((t, c+1)) = \Phi(\mathbf{x}((t, c)), q((t, c))) \tag{6}$$

The discrete change in the differential function takes effect at time (t,c)+(0,1)=(t,c+1) and × evolves from its value at the transition. Because × does not change discontinuously, the trajectory ×(t), which satisfies Equation 5 and is a function from $\mathbb{R} \rightarrow \mathbb{R}^m$, is equal to ×((t,c)), a function from $\mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}^m$, for all $\mathbf{c} \in \mathbb{N}$, and so the two technically distinct trajectories can be used interchangeably.

The *total* state transition function Δ_{da} takes the system from an initial state $(x(t_0),q)$ through an interval $[(t_0,c_0),(t_f,c_f))$. Defining first

$$g(\mathbf{x},q,h)=\mathbf{x}+\int_0^h f(\mathbf{x}(t),q)\, dt, \tag{7}$$

the function Δ_{da} is defined recursively by

$$\begin{aligned}
&\Delta_{da}((\mathbf{x},q),[(t,c),(t_f,c_f))) = \\
&\quad 1)\ (\mathbf{x},q) \text{ if } [(t,c),(t_f,c_f)) \text{ is empty} \\
&\quad 2)\ \Delta_{da}((g(\mathbf{x},q,t'-t),q),[(t',c'),(t_f,c_f))) \\
&\qquad \text{if } (\forall \hat{t} \in [t,t'))(\Phi(g(\mathbf{x},q,\hat{t}-t),q)=q) \\
&\qquad \text{where } (t',c') \text{ is the smallest in } [(t,c),(t_f,c_f)] \\
&\qquad \text{such that } \Phi(g(\mathbf{x},q,t'-t),q) \neq q \vee (t',c')=(t_f c_f) \\
&\quad 3)\ \Delta_{da}((\mathbf{x},\Phi(\mathbf{x},q)),[(t,c)+(0,1),(t_f,c_f))) \\
&\qquad \text{if } \Phi(\mathbf{x},q) \neq q
\end{aligned} \tag{8}$$

This abstruse definition bridges a technical, but critical, gap between self-clocked cellular automata and the differential automata that are constructed in the next section; we will return to Equation 8 in Section 3.

2. Modeling self-clocked cellular automata with differential automata

Every self-clocked cellular automaton is a homomorphic image (see, e.g., [Zeigler, 2000]) of a differential automaton whose trajectories converge to a finite number of limit cycles. That self-clocked cellular automata are essentially periodic follows this fact. In this section, the relevant class of differential automata is constructed.

The discrete states of a binary cellular automaton can be numbered by treating its leftmost cell as the most significant bit in a binary number and its rightmost as the least significant. An automaton with n cells has 2^n discrete states. In addition to its binary state, each cell k has a clock τ_k that induces a change of state under two conditions: beginning from zero, τ_k grows until it reaches P_k, τ_k shrinks until it reaches 0. The direction of the clock is the cell's third and final state variable.

The pair (b_k,d_k) is the discrete state of cell k, where $b_k \in \{0,1\}$ is the binary state and $d_k \in \{1,-1\}$ is the direction of the clock; the set of discrete states is $Q=\{0,1\}\times\{1,-1\}$. The clock τ_k begins, and remains, in the interval $[0,P_k]$. The differential automata that models the cell is

$$\frac{d\tau_k}{dt} = d_k \tag{9}$$

$$\Phi_k(\tau_k,(b_k,d_k)) = \begin{cases} (\delta(b_k,b_{k,l},b_{k,r}),-d_k) & \text{if } (\tau_k = 0 \wedge d_k = -1) \\ & \vee(\tau_k = P_k \wedge d_k = 1) \\ (b_k,d_k) & \text{otherwise} \end{cases} \tag{10}$$

A cellular automaton with n cells is a differential automaton with the set of discrete states Q^n and the continuous state vector $\tau=[\tau_1\ \tau_2\ \ldots\ \tau_n]$ for which each component τ_k is in $[0,P_k]$. The dynamic equations for this model are

$$\frac{d\tau}{dt} = [d_1\ d_2\ \ldots\ d_n] \tag{11}$$

$$\Phi(\tau,(b_1,d_1),\ldots,(b_n,d_n)) = ((b'_1,d'_1),\ldots,(b'_n,d'_n)) \tag{12}$$

$$\text{where } (b'_k, d'_k) = \Phi_k(\tau_k, (b_k, d_k)) \tag{13}$$

This differential automaton has five important properties: 1) it is deterministic; every initial state generates a single trajectory, 2) the model is legitimate (*i.e.*, non-Zeno), 3) the vector $d\tau/dt$ is constant between discrete events, 4) Q^n is a finite set, and 5) the event surfaces form hyper-cubes in its phase space. The fourth and fifth conditions confine the continuous trajectories of the cellular automaton to a subset K of $\mathbb{R}^n$. The first, second, and third conditions imply that the discrete trajectories are well behaved: the system is predictable, always moves forward on its real time line, and between events follows straight lines through $\mathbb{R}^n$.

Matveev and Savkin [2000] show that these properties have the following consequences: (i) there exists a limit cycle lying in K, (ii) the number of such cycles is finite, (iii) any limit cycle lying in K is regularly locally asymptotically stable in K, and (iv) any trajectory lying in K regularly converges to one of these limit cycles. Informally, every trajectory of the cellular automaton converges to a periodic trajectory as the real time t goes to infinity. Both the continuous and discrete variables are eventually periodic! Moreover, there are a finite number of these limit cycles, and so they act as distinct equilibrium trajectories for the system.

There are exactly two cellular automata with a single cell, and these give the simplest demonstration of the theory. The cell has a duration P. It is its own left and right neighbor, and so δ is entirely defined by its action on the triples (1,1,1) and (0,0,0); for brevity δ is written as a function of a single value. Two transition rules can be defined:

$$\delta_\alpha(b) = b \text{ and}$$

$$\delta_\beta(b) = \begin{cases} 1 & \text{if } b=0 \\ 0 & \text{if } b=1 \end{cases}$$

Both automata, the first with rule δ_α and the second with rule δ_β, have a pair of limit cycles. These are shown in Figure 3. The event surfaces are lines at t=0 and t=P. Beginning with a direction d=1, the clock moves up to P where a discrete event occurs and causes the direction to change; it then moves to 0 where the direction changes again; and the cell bounces back and forth between these two constraining surfaces. Both automata need two bounces to return to their initial states and so have a period of length 2P.

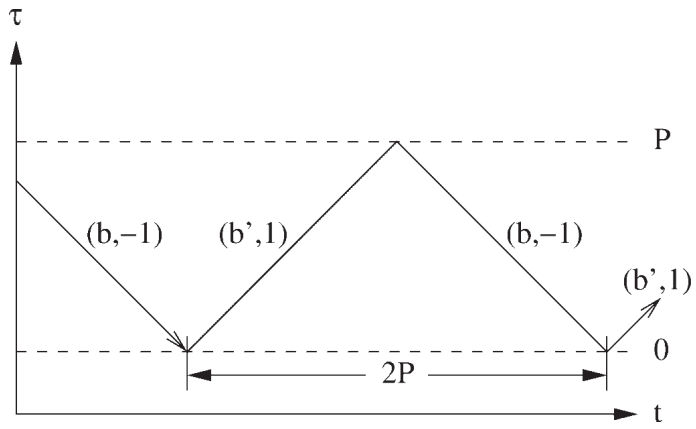


Figure 3. Event surfaces and periodic trajectories of two self-clocked cellular automata each with a single cell; in this drawing, $b'=\delta_\alpha(b)$ and $b=\delta_\alpha(b')$ for the first automaton, and similarly with respect to δ_β for the second automaton.

3. A HOMOMORPHISM FROM DIFFERENTIAL AUTOMATA TO SELF-CLOCKED CELLULAR AUTOMATA

The simulation procedure described to the Introduction can be reformulated as a recursive state transition function. A self-clocked cellular automaton with n cells has the set of states $S = \prod_{k=1}^{n}(\{0,1\} \times \mathbb{R})$, and in the state $s=(\ldots,(q_k,e_k),\ldots)$ each pair (q_k,e_k) describes a single cell. The real time remaining to the next state transition is given by the time advance function

$$ta(s) = \min_{k\in[1,n]} P_k - e_k \qquad (14)$$

The state transition function that takes the system from the state s through an interval $[(t,c),(t_f,c_f))$ is

$$\Delta_{ca}(s,[(t,c),(t_f,c_f)))=$$
$$1)\ s \text{ if } [(t,c),(t_f,c_f)) \text{ is empty}$$
$$2)\ \Delta_{ca}((\ldots,(q_k,e_k+t'-t),\ldots),[(t',c'),(t_f,c_f)))$$
$$\text{if } (\forall k)(e_k<P_k),$$
$$\text{where } (t',c') \text{ is the smallest in } [(t,c),(t_f,c_f)) \quad (15)$$
$$\text{such that } t'-t=ta(s)\vee(t',c')=(t_f,c_f)$$
$$3)\ \Delta_{ca}((\ldots,(q_k',e_k'),\ldots),[(t,c)+(0,1),(t_f,c_f)))$$
$$\text{if } (\exists e_k)(e_k=P_k)$$

$$\text{where } (q_k',e_k')=\begin{cases}(q_k,e_k) & \text{if } e_k<P_k\\ (\delta(q_k,q_{k,l},q_{k,r}),0) & \text{if } e_k=P_k\end{cases} \quad (16)$$

The state transition function Δ_{ca} acts exactly like the iterative procedure from which it is derived: the elapsed times are incremented by the time remaining to next transition; the next state of the active cells is computed; and the process repeats.

With this formalism and the formalism of Equation 8, it can be shown that the self-clocked cellular automaton is a homomorphic image of the differential automaton constructed in Section 2. The state of the differential automaton is mapped into a state of the cellular automaton by

$$H((\ldots,(\tau_k,(b_k,d_k)),\ldots))=(\ldots,(b_k,\varepsilon_k),\ldots) \quad (17)$$

$$\text{where } \varepsilon_k=\begin{cases}P_k-\tau_k & \text{if } d_k=-1\\ \tau_k & \text{if } d_k=1\end{cases} \quad (18)$$

Using Equation 18, the time advance of the cellular automaton can be written

$$ta(H((\ldots,(\tau_k,(b_k,d_k)),\ldots)))=\min_{k\in[1,n]}\frac{1}{2}((P_k-\tau_k)(d_k+1)-\tau_k(d_k-1)) \quad (19)$$

To show that H is a homomorphism, it is enough to consider in turn cases 1, 2, and 3 in Equations 15 and 8. Case 1 is trivial. For cases 2 and 3, first observe that under H, the switching surfaces of the differential automaton define the time advance function of the cellular automaton. Specifically, the event surface that will be encountered next by the differential automaton is defined by

$$\min_{k\in[1,n]}\frac{1}{2}((P_k-\tau_k)(d_k+1)-\tau_k(d_k-1))=0 \quad (20)$$

and the time to reach this surface is ta(H((…,(τ_k,(b_k,d_k)),…))). Therefore, h in Equation 8 and ta(s) in Equation 15 are equal; the hybrid system in state γ and discrete event system in state H(γ) undergo their next events at the same moment.

Keeping now in mind that h=ta(s), consider cases 2 and 3 in turn. For case 2, it is sufficient that simulation over the interval ω=[(0,0),(h,1)) gives

$$H(\Delta_{da}((...,(\tau_k,(b_k,d_k)),...),\omega)) = \Delta_{ca}(H((...,(\tau_k,(b_k,d_k)),...)),\omega) \quad (21)$$

Now consider just the action of H, Δ_{da}, and Δ_{ca} on the single cell k. For the left side of Equation 21

$$H(\Delta_{da}((\tau_k,(b_k,d_k)),\omega)) =$$

$$H((\tau_k + hd_k,(b_k,d_k))) = \begin{cases}(b_k,P_k-\tau_k+h) & \text{if } d_k=-1\\ (b_k,\tau_k+h) & \text{if } d_k=1\end{cases}$$

and for the right side

$$\Delta_{ca}(H((\tau_k,(b_k,d_k))),\omega) = \begin{cases}\Delta_{ca}((b_k,P_k-\tau_k),\omega) & \text{if } d_k=-1\\ \Delta_{ca}((b_k,\tau_k),\omega) & \text{if } d_k=1\end{cases}$$

If d_k=-1 then

$$\Delta_{ca}((b_k,P_k-\tau_k),\omega) = (b_k,P_k-\tau_k+h)$$

and if d_k=1 then

$$\Delta_{ca}((b_k,\tau_k),\omega) = (b_k,\tau_k+h)$$

just as desired.

For case 3, it is sufficient that Equation 21 holds when ω is replaced by the interval ω_ϕ=[(0,0),(0,1)). For each cell there are three cases to consider. Case (i): if 0<τ_k<P_k then the discrete state of the differential automaton does not change. Because h=ta(s), it is also true that e_k<P_k and so the discrete state of the cellular automaton does not change. Therefore Equation 21 holds. Case (ii): If 0=τ_k and d_k=-1 then the differential automaton changes its discrete state and, for H to be a homomorphism, the cellular automaton must as well. Again, because h=ta(s), 0=τ_k and d_k=-1 is equivalent to e_k=P_k. Hence, Equation 21 holds. Case (iii): P_k=τ_k and d_k=1; the argument here is identical to Case (ii). Therefore, H is a homomorphism.

This is the main result. To summarize, the differential automata constructed in Section 2 are essentially periodic; every self-clocked cellular automaton

is a homomorphic image of one of these differential automaton; therefore, every self-clocked cellular automaton is essentially periodic.

4. Discrete event simulation of self-clocked cellular automata

Every self-clocked cellular automaton can be written as a discrete event system in the terms of the Discrete Event System Specification (DEVS) and very efficiently executed using a DEVS simulation engine [Zeigler, 2000]. Each cell is an atomic model with states (b,b_l,b_r,ε) where $b,b_l,b_r \in \{0,1\}$ and $\varepsilon \in [0,P]$. The model's set of input is $\{l,r\}\times\{0,1\}$ where l is the left neighbor, r is the right neighbor, and $\{0,1\}$ is the neighboring state. The set of output is $\{0,1\}$. Letting $q=(b,b_l,b_r,\varepsilon)$, the dynamic behavior of a cell is defined by

$$\delta_{int}(q) = (\delta(b,b_l,b_r),b_l,b_r,0)$$

$$\delta_{ext}(q,e,x^b) = (b,b_l',b_r',\varepsilon+e)$$

$$\text{where } b_l' = \begin{cases} \hat{b} & \text{if } (l,\hat{b}) \in x^b \\ b_l & \text{otherwise} \end{cases}$$

$$\text{and } b_r' = \begin{cases} \hat{b} & \text{if } (r,\hat{b}) \in x^b \\ b_r & \text{otherwise} \end{cases}$$

$$\delta_{con}(q,x^b) = \delta_{ext}(\delta_{int}(q),0,x^b)$$

$$\lambda(q) = \delta(b,b_l,b_r)$$

$$ta(q) = P_k - \varepsilon$$

This model has the binary state b of the cell, the binary states b_l and b_r of its neighbors at their last transition, and the elapsed time ε. When the time advance ta expires, the model produces as output its next binary state, sets b to this new value, and resets the elapsed time. The cell can receive input from its neighbors at any time, and when this occurs it records their binary states and increments δ by the time e that has elapsed since its last event (*i.e.*, change of state by δ_{int}, δ_{ext}, or δ_{con}).

The clock of the differential automata can be recovered by adding the direction d and time t_L of the last event to the state variables of this model. Initially, d=1 and it is multiplied by -1 by the internal transition function. The time t_L of the last event is initially zero, and it is incremented by e in the

external transition function and by the time advance in the internal transition function. The clock τ at any time t is then

$$\tau = \frac{1-d}{2}P + d(\varepsilon + t - t_L)$$

5. Illustrations of the theory

In a left looking cellular automaton, each cell takes its next state from its left neighbor. The transition function is

$$\delta(q, q_l, q_r) = q_l$$

and when P=1 the configuration of the cell space simply translates to the left at each step. Figure 4 shows 125 steps of this model beginning with alternating 1, colored black, and 0, colored white, cells. The periodicity of this model is immediately apparent.

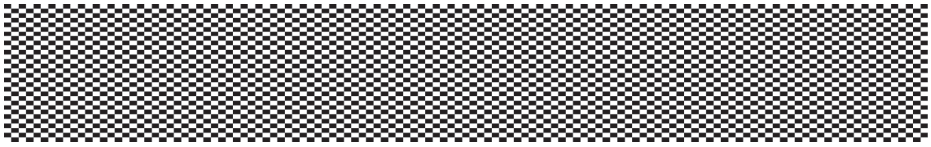


Figure 4. Discrete trajectory of the left looking automaton with P=1. The cells are arranged from bottom to top and time increases from left to right.

Figure 5. Discrete trajectory of the left looking automaton with P selected for each cell at random from 1/2, 1/3, and 1.

Figure 5 shows 125 units of time for the left looking model with P selected at random from 1/2, 1/3, and 1. The trajectory of this automaton initially appears to be irregular, but at the sixth set of bands settles into a recognizable pattern. The continuous phase space of this model has three dimensions: the first for cells with P=1/2, the second for cells with P=1/3, and the third for cells with P=1. Figure 6 shows planar cuts of the phase space, and these clearly depict the periodicity of the clock variables.

This same experiment was repeated with P selected randomly from $\sqrt{2}$, $\sqrt[3]{2}$, and 2. A recognizable pattern takes much longer to appear in this case. Figure 7 shows the discrete trajectory for the first 125 units of time. At first, the behavior of the automaton is quite erratic, but a structure seems ultimately to assert itself. This is illustrated in Figure 8, which shows the discrete trajectory from time 15000 to 15125.

The phase space of this automaton is more intricate than the previous two. The $\sqrt{2} - \sqrt[3]{2}$ plane is shown in Figure 9(a), where a regular pattern of diamonds is quite visible. In fact, there appear to be close spaced nets of diamonds, which produce a distinct pattern. These can be seen in the enlarged view of this plane shown in Figure 9(b). It should be noted, however, that these are not snapshots of a periodic trajectory. Because the clocks of the individual elements lack a common period, their trajectories will completely fill the phase plane.

Conclusions

Small systems, with components having rational durations, move quickly into their intrinsic limit cycles, but these may take a very long time to appear if the phase space of the model has many dimensions or if it has many discrete states. Patterns seen in short-term observations might reflect unique, transient effects or might be part of a long limit cycle.

For automata whose components have irrational durations, however, there is no genuinely periodic trajectory. As the demonstrations above show, the trajectory may appear more or less periodic after some time, and may in fact follow a more or less predictable path indefinitely. That is, it appears to be converging to some periodic trajectory, but if so, the theory developed by Matveev and Savkin [2000] (a brief summary is given in [Matveev and Savkin, 1999]) does not say what that trajectory is, stating only that it exists.

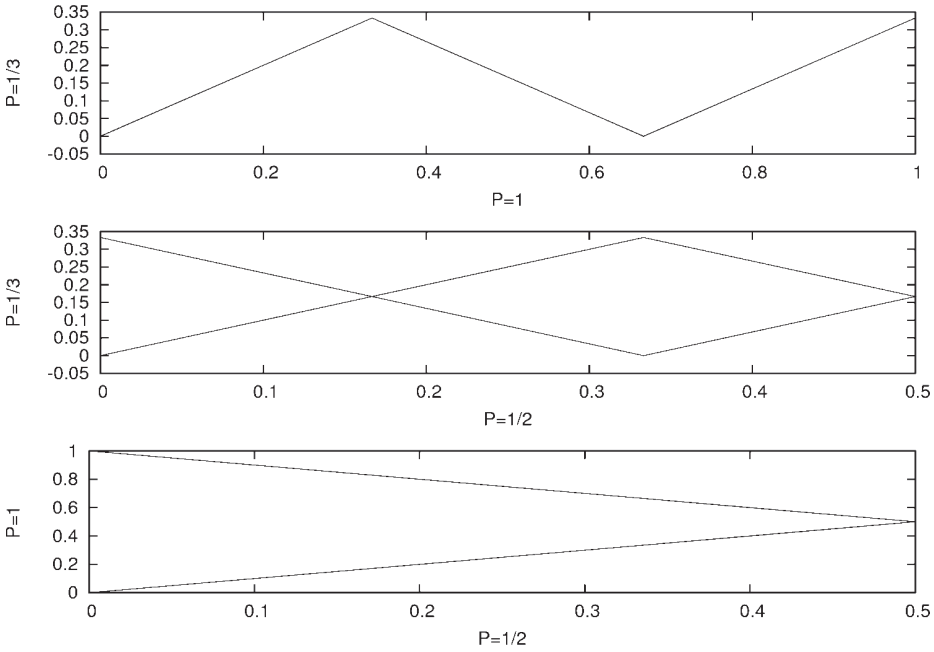
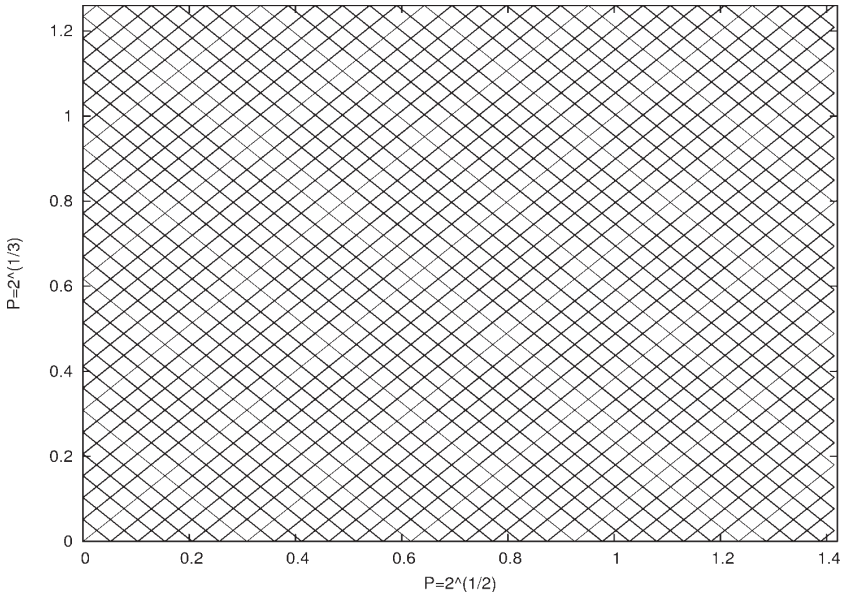


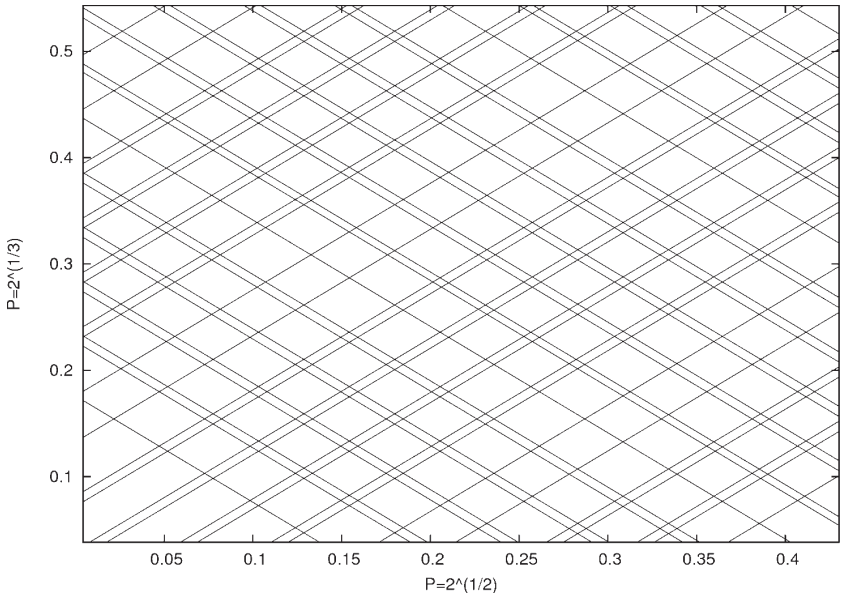
Figure 6. Planar cuts of the phase space corresponding to the discrete trajectory shown in Figure 5.

Figure 7. Discrete trajectory over the real interval [0,125] of the left looking automaton with P selected for each cell at random from $\sqrt{2}$, $\sqrt[3]{2}$, and 2.

Figure 8. Discrete trajectory over the real interval [15000,15125] for the same automaton shown in Figure 7.



9(a) View of the entire phase plane.



9(b) Enlarged view.

Figure 9. The $\sqrt{2} - \sqrt[3]{2}$ phase plane corresponding to the discrete trajectory shown in Figure 8.

This seeming contradiction, between the absence of genuine periodicity and the claim of convergence to a limit cycle, raises three questions:

1. Do the asynchronous automata in fact satisfy the required assumptions, or has some critical point been overlooked?
2. Is an assumption missing from the underlying theory, which if invoked would rule out the asynchronous, cellular automata described above?
3. Is there in fact a limit cycle (defined in [Matveev and Savkin, 2000; 1999] as a class of periodic trajectories) that the system converges too and, if so, what are those trajectories?

With respect to questions 1 and 2, it can be observed that cellular automata are unusual amongst differential automata most often considered in that the real space occupied by distinct, discrete states overlap. This might provide a clue to either the missing assumption (*i.e.*, a positive answer to 2) or the point of divergence from the theory (*i.e.*, a positive answer to 1).

If the conjecture in this paper holds, it has further consequences for the class of Schedule Preserving DEVS (SP DEVS) described by Hwang [Hwang, 2005]. If a SP DEVS has a finite number of discrete states, then in each state the model has a fixed duration. It is likely, therefore, that the same homomorphism developed above for the cellular automata can be extended to this larger class of systems, and so it can be conjectured that all SP DEVS with a finite set of states are essentially periodic. Even if this broad conjecture fails by its encompassing irrational durations, the same conjecture may still hold if only rational durations are considered.

References

[Bersini and Detour, 1994] – Bersini H. and Detour V., Asynchrony induces stability in cellular automata based models. *In: Artificial Life IV*, 1994, pp. 382-387.

[Cornforth *et al.*, 2002] – Cornforth D., Green D., Newth D. and Kirley M., Do artificial ants march in step? Ordered asynchronous processes and modularity in biological systems. *In: Artificial Life VIII*, 2002, pp. 28-32.

[Gardner, 1970] – Gardner M., The fantastic combinations of John Conway's new solitaire game: Life. *Scientific American*, No. 223, 1970, pp. 120-123.

[Green *et al.*, 2001] – Green D., Newth D., Cornforth D. and Kirley M., On evolutionary processes in nature and artificial systems. *In: Proceeding of the Fifth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, 2001, pp. 1-10.

[Hwang, 2005] – Hwang M. H., Tutorial: Verification of Real-time System Based on Schedule-Preserved DEVS. *In: Proceedings of 2005 DEVS Symposium*, San Diego, Apr. 2-8, 2005.

[Matveev and Savkin, 1999] – Matveev A. and Savkin A., Existance and stability of limit cycles in hybrid dynamical systems with constant derivatives. Part 1. General theory. *In: Proceedings of the 38th Conference on Decision & Control*, 1999, pp. 284-289.

[Matveev and Savkin, 2000] – Matveev A. and Savkin A., *Qualitative Theory of Hybrid Dynamical Systems*, Birkhauser. 2000.

[Schönfisch and de Roos, 1999] – Schönfisch B. and de Roos A., Synchronous and asynchronous updating in cellular automata. *Biosystems*, Vol. 51, No. 3, 1999, pp. 123-143.

[Stark and Hughes, 2000] – Stark R. and Hughes W., Asynchronous, irregular automata nets: the path not taken. *Biosystems*, Vol. 55, No. 1-3, 2000, pp. 107-117.

[Zeigler, 2000] – Zeigler B. P., Prauhofer H. and Kim T. G., *Theory of Modeling and Simulation*, Academic Press, 2000 (2nd ed.).