Transient dynamics and their control in time-delay autonomous Boolean ring networks

Johannes Lohmann,[1,2] Otti D'Huys,[1,*] Nicholas D. Haynes,[1] Eckehard Schöll,[2] and Daniel J. Gauthier[1,3,†]
[1]*Department of Physics, Duke University, Durham, North Carolina 27708, USA*
[2]*Institut für Theoretische Physik, Technische Universität Berlin, 10623 Berlin, Germany*
[3]*Department of Physics, The Ohio State University, Columbus, Ohio 43210, USA*
(Received 31 May 2016; revised manuscript received 23 January 2017; published 17 February 2017)

Biochemical systems with switch-like interactions, such as gene regulatory networks, are well modeled by autonomous Boolean networks. Specifically, the topology and logic of gene interactions can be described by systems of continuous piecewise-linear differential equations, enabling analytical predictions of the dynamics of specific networks. However, most models do not account for time delays along links associated with spatial transport, mRNA transcription, and translation. To address this issue, we have developed an experimental test bed to realize a time-delay autonomous Boolean network with three inhibitory nodes, known as a repressilator, and use it to study the dynamics that arise as time delays along the links vary. We observe various nearly periodic oscillatory transient patterns with extremely long lifetime, which emerge in small network motifs due to the delay, and which are distinct from the eventual asymptotically stable periodic attractors. For repeated experiments with a given network, we find that stochastic processes give rise to a broad distribution of transient times with an exponential tail. In some cases, the transients are so long that it is doubtful the attractors will ever be approached in a biological system that has a finite lifetime. To counteract the long transients, we show experimentally that small, occasional perturbations applied to the time delays can force the trajectories to rapidly approach the attractors.

DOI: 10.1103/PhysRevE.95.022211

I. INTRODUCTION

Boolean network models are widely used for describing systems with switch-like interactions in multiple research fields: In the geosciences, they have been used as idealized climate models on a large range of timescales from interannual to paleoclimatic variability [1,2], as well as for earthquake modeling and prediction [3,4]. In biology, Boolean networks have been used to describe neural networks [5,6], and gene regulatory networks (GRNs) [7–11], and are applied to study evolution [12,13] or immune response [14]. Furthermore, Boolean network models find applications in economics [15] and social sciences [16]. In all of these systems, current research focuses on identifying the type and number of coexisting attractors and their real-world interpretation. This is especially the case for GRNs, which are the main focus of this work.

The main idea of GRNs was put forth by Jacob and Monod [17]: They proposed that a transcription factor, which is a protein encoded by a specific gene, can bind to another gene, thereby repressing or enhancing its production rate. As a consequence, a directed network emerges where the nodes are the genes, and links are given by causal interactions via transcription factors. The dynamics of a regulatory network can be approximated by considering genes to be either active or inactive, i.e., producing their target protein or not. This depends in a switch-like manner on the presence of a combination of transcription factors, which motivates the description of GRNs as a Boolean switching network, where each gene evaluates its "on" or "off" state (denoted by "1" or "0") according to a Boolean function of its inputs, i.e., its transcription factors.

Gene expression patterns observed in GRNs are believed to mirror different characteristics of a cell. First, different gene expression patterns correspond to different cell types, as hypothesized by Kauffman [18], to match different coexisting attractors of a Boolean network. Second, a gene expression pattern displays a specific functional state of the cell, such as differentiation (execution of a tissue-specific task), proliferation (cell cycle) and apoptosis (cell death) [19]. Functional state and type of a cell may be determined by separate subsets of the GRN.

When studying the dynamical patterns or attractors of model GRNs [20–22], predictions depend on the choice of models, e.g., the scaling of the number of attractors with respect to system size in random Boolean networks [23]. While a Boolean approximation to sigmoidal switching interactions seems robust in many cases [24–26], several oversimplifications of established models, such as synchronous update of node states, negligence of transmission delays, and homogeneous network elements, can restrict the dynamics or yield artifacts. For example, it is well known that a large number of periodic cycles in Boolean networks become unstable for an autonomous updating scheme [23] and that even very simple Boolean networks can display complex dynamics [27,28]. Nevertheless, long-lasting unstable orbits might still be biologically meaningful if they decay slowly to the asymptotic attractors.

The simplified description of GRNs omits several processes in gene expression that comprise transcription of DNA to mRNA and translation to the target protein. These intermediate steps can be modeled as separate nodes within the GRN [29]. Additionally, intermediate steps in gene expression can cause significant time delay in certain types of genes [30], which we include explicitly in our experimental approach. Time delays along network links in related non-Boolean systems, such as neural networks, have been shown to induce oscillations in systems that would otherwise converge to a

*Now at: Department of Mathematics, Aston University, B7 4ET Birmingham, UK.
†gauthier.51@osu.edu

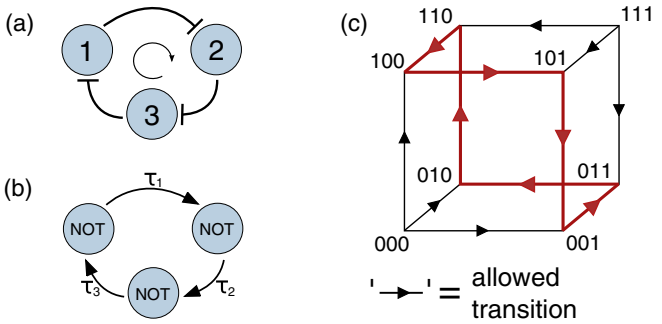

FIG. 1. (a) Ring network of $N = 3$ nodes with inhibitory interactions. (b) Description of panel (a) as time-delay autonomous Boolean network. (c) State transition diagram of the autonomous Boolean network (b) without link delays.

fixed point [31–33]. Transcriptional and translational delays were identified to cause oscillations of the transcription factors in real [34], synthetic [35], and model [36,37] GRNs. In neural ring networks, delay-induced transient oscillations have been reported [38–41], leading to instability that may interfere with information processing. We anticipate that adding delay in models of GRNs will similarly modify the stability of response patterns.

To address the shortcomings of conventional models for representing real-world networks, we use an experimental test bed to realize time-delay autonomous Boolean networks (ABNs) with electronic logic gates on field-programmable gate arrays (FPGAs) [42,43]. With this approach, we also address the question whether the dynamics are robust to noise and heterogeneity as found in any real-world system. The ultrafast timescale of the platform allows us to observe phenomena that are difficult to obtain through numerical simulations of the model because the time-delay model is exceedingly stiff, especially as the time delay increases. Moreover, a general framework to integrate delay equations with stochastic delays has not yet been developed to the best of our knowledge. With this platform, we study small ring networks of inhibitory genes, which exhibit nearly periodic oscillations of gene activity that evolve slowly to an asymptotically stable periodic state. This is a prominent behavior in regulatory systems of organisms [44–48]. Specifically, we study the so-called repressilator [49] network, illustrated in Fig. 1, which is extensively studied as a synthetic network.

The primary purpose of this paper is to show that delays induce transient patterns in ABNs that are distinct from the asymptotic attractor. These transients can be exceedingly long, consisting of millions or billions of oscillations. This suggests that gene regulation patterns inferred from asymptotic study of Boolean network models might not always be relevant on observational timescales. Our experiments incorporate additional features of realistic Boolean models. First, we demonstrate that the transient dynamics are very sensitive to delay heterogeneities and initial conditions. Second, the inherent noise in our system reveals an important role of stochasticity, as we observe long-tailed distributions for the transient durations for longer delay times. Furthermore, we present a method to counteract these super-long transients using small occasional perturbations to one of the time delays. Finally, we reproduce essential experimental observations with a theoretical model based on piecewise-linear delay differential equations.

II. BACKGROUND AND EXPERIMENTAL APPROACH

A. Network dynamics without time delay

A well-established framework describing GRNs as ABNs without link delays is the so-called Glass model, which embeds the logic and topology of Boolean networks in systems of piecewise-linear differential equations [50]. Here, the symbolic dynamics can be described by Boolean network states $(b_1,b_2,...,b_N)$ with $b_i \in \{0,1\}$, which correspond to the current activity of N genes. These states can be depicted as vertices on an N-dimensional hypercube [Fig. 1(b)], and they evolve in time on the directed edges of the hypercube, which are determined by topology and Boolean functions of the network nodes. The attractor on the hypercube corresponds to the asymptotic gene expression pattern of the GRN, which can be static or dynamic, i.e., steady states or cyclic switching patterns.

A ring network with $N = 3$ repressing nodes, modeled by the Boolean NOT function, exhibits a cyclic attractor with the switching sequence $(0,0,1) \to (0,1,1) \to (0,1,0) \to (1,1,0) \to (1,0,0) \to (1,0,1) \to \cdots$ [50]. Without link delays, this attractor is reached immediately after the switching of at most one variable. We test whether these predictions hold as delays are included along the links.

B. Implementation on a field-programmable gate array

We construct experimental repressilator networks with autonomous (unclocked) logic gates performing Boolean NOT operations on a FPGA (Altera Cyclone IV on the Terasic DE2-115 demonstration kit), as described in greater detail in Appendix A. An important characteristic of this experimental system is the response time of a logic gate, which we will need when comparing to the theoretical model described in Sec. V.

Fundamentally, the finite rise and fall time of the output of a logic gate due to a step change in its input gives rise to an effective time delay in its output. In a mathematical model, we can account for this by either choosing to assign a response time to the node and no delay time, or assign a delay time to the node, and take the rise and fall time as instantaneous. Here, we take the three nodes of the repressilator to have a response time of $\tau_r = 0.41 \pm 0.05$ ns, which corresponds to the time the analog voltages take to reach the threshold. This characteristic timescale may be linked to production and degradation rates (lifetimes) of gene products in real GRNs. The network nodes are connected with links that are constructed by connecting a cascade of n_i pairs of NOT gates in series. For these NOT gates, instead of taking all rise and fall times into account, we assign a time delay $\tau_0 = 0.52 \pm 0.05$ ns. Therefore, each network link i has a delay $\tau_i = n_i\tau_0$ due to a discrete number n_i of delaying elements and the networks are thus characterized by the vector $\mathbf{n} = (n_1,n_2,n_3)$. Our initial analysis focuses on equal numbers n of delay elements in each link, corresponding to approximately homogeneous delays.

This delay time is only an estimate because it is subject to different heterogeneities: manufacturing imperfections lead to varying propagation delays of the logic elements and the routing wires in between network nodes and delay lines are not identical at every node. We find that manufacturing imperfections typically give rise to a delay heterogeneity of 50–150 ps in two delay lines with identical number n of delay elements. In contrast, because of asymmetries in the chip layout, we cannot directly measure the heterogeneities due to nonidentical routing wires. By studying different placements of the network on the chip as discussed in Appendix B, we estimate that this heterogeneity is on the order of 100 ps. The heterogeneity is less than 10% of a typical link delay used in our experiment, which ranges from approximately 1 to 3.5 ns.

Because we investigate a time-delay autonomous system, the initial conditions have to be defined for a time interval corresponding to the longest time delay $\tau_{\max}$ in the network. To this end, we first keep the nodes' states fixed for a time much greater than $\tau_{\max}$. Then, the autonomous dynamics are essentially simultaneously released by a signal generated by a clocked register for each node. There is a remaining heterogeneity of roughly 100 ps in the timing of this signal due to the nonzero phase shift of the clock at different positions on the chip and nonidentical wires leading to the different network nodes. As above, this timing difference is no more than 10% of the typical delay used in our experiment.

We make small changes to the time delays by implementing an alternate path within a delay line that includes a logic gate with slightly longer propagation delay than its counterpart in the original path. This allows us to selectively change the heterogeneity of time delays by ∼100 ps. To change the timing heterogeneity of the initial conditions, we use the CAD tool Altera Quartus II as discussed in Appendix B. It allows us to alter the relative positions of the initializing clocked registers for each node, thereby inducing an additional routing delay, which affects the heterogeneity of the timing of the initial conditions by ∼100 ps.

The attractor dynamics are also affected by the presence of stochastic, time-varying fluctuations in the link time delays due to fluctuations in the gate charge on a fast time scale, and on the FPGA chip temperature and supply voltage on longer time scales. To reduce the effects of the change in temperature in the laboratory, we place the FPGA in a small box where the typical variation is $\pm 0.5\,^{\circ}$C. These processes give rise to variation in the delay times of the logic elements on the order of 10 ps over all timescales, which is ∼1% of the typical delay times. These fluctuations are small in comparison to the noisy environment typically encountered in a biological regulatory network. We conjecture, however, that the behavior we observe here will be obtained even with larger noise, albeit with a different timescale for the distribution of transient times.

III. REPRESSILATOR DYNAMICS

A. Transient oscillatory patterns

We observe long nearly periodic transients when we initialize the networks with initial conditions that do not lie on the eventual attractor, i.e., Boolean states (0,0,0) or (1,1,1). In Fig. 2(a), we show the waveform $V_1(t)$ of one node

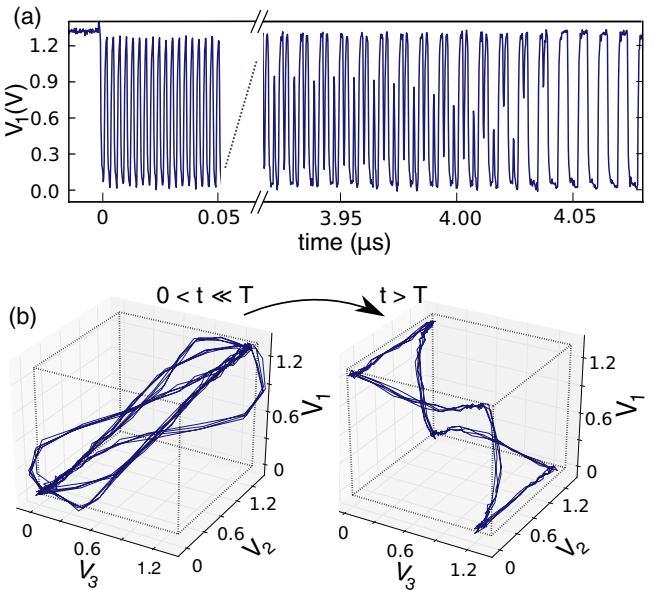

FIG. 2. Temporal evolution of the transient dynamics. (a) Waveform of one node in a network with $\mathbf{n} = (2,2,2)$ delay elements initialized in the (1,1,1) state. The initial nearly periodic in-phase behavior eventually gives way to out-of-phase oscillations after $T \sim 4.05\ \mu$s. (b) Corresponding three-dimensional phase portrait of the trajectory, which evolves from in-phase nearly periodic oscillations (left) to the asymptotic out-of-phase periodic attractor (right).

observed in a network with $\mathbf{n} = (2,2,2)$ after initialization, and after collapse of the transient, which occurs after $T \sim 4.05\ \mu$s, or equivalently approximately 1500 oscillations of each node. Initially, the dynamics of each node is nearly periodic and in-phase with the other nodes with a period of 2.84 ± 0.01 ns. For reference to the theoretical discussion in Sec. V, this period $\sim 2\tau$, where τ is average link delay. The in-phase oscillations are indicated in the three-dimensional phase portrait in Fig. 2(b), where the trajectory evolves nearly on the hypercube diagonal in an apparently closed trajectory, at least on the observed timescale of a few cycles. This trajectory evolves slowly, always remaining nearly periodic. During this transient phase, a cycle edge is catching up to another due to tiny network heterogeneities or initial conditions, eventually leading to the annihilation of a low-high-low transition [at ∼4.02 μs in Fig. 2(a)] then a high-low-high transition (just after ∼4.04 μs), indicating the end of the transient. After this transient collapse, the trajectory is characterized by the stable periodic attractor shown in the right panel of Fig. 2(b) with a period of 8.51 ± 0.01 ns [compare with Fig. 1(b)]. Again, for future reference, this period is $\sim 6\tau$.

B. Scaling and distribution of transient durations

In our experiments, the time to reach the asymptotic attractor is different from trial to trial mainly due to the stochastic variation in the delay due to charge noise fluctuations at the gates, as mentioned in the previous section. Below, we characterize this behavior by measuring the transient duration probability distribution. Especially for long link delays, these times can be exceedingly long. This is an example of

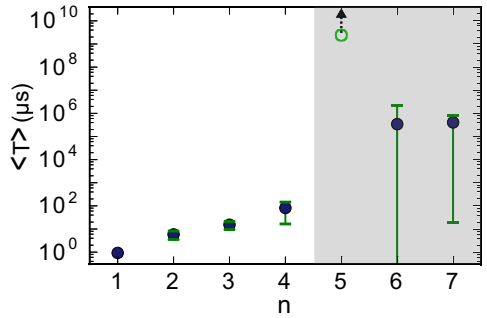

FIG. 3. Mean transient durations for networks with a homogeneous number of n delay elements in each network link, repeatedly initialized in the (0,0,0) state. For $n = 5$, the transients for each experimental run last longer than 40 min at which point the experiment terminated. Here, the data point only signifies the termination point and the arrow indicates that the mean is longer than this value. For each data point in the white region, we collect 100 000 samples. In the gray region, we conduct 30 experimental runs for $n = 5$, 32 000 for $n = 6$, and 94 669 for $n = 7$. The error bars indicate the standard deviation.

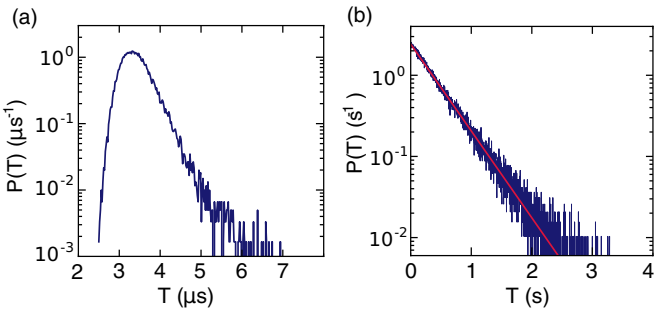

FIG. 4. Probability density function for the transient durations of networks with (a) $\mathbf{n} = (2,2,2)$ and (b) $\mathbf{n} = (7,7,7)$ delay elements. Here, a histogram is generated indicating the number of instances that we observe a transient time within T and $T + \Delta T$, normalized by ΔT and the total number of observed transient times. It approximates the transient time probability density $P(T)$, where $P(T)\Delta T$ is the probability of observing a time interval T when ΔT is small. (b) The mean transient duration $\langle T \rangle$ is 0.41 s and the red line shows an exponential fit [$p(T) \propto e^{-\lambda T}$] to the data with exponent $\lambda = 2.44\ \mathrm{s}^{-1}$.

super-transient behavior, which is encountered in a range of other systems and often have a mean transient time that scales exponentially with a system parameter. This has been observed for spatially extended systems [51–56], networks [38,57,58], and time-delay systems [39]. The observed behavior also shows similarities to stable chaos in coupled map lattices for which the transient time scales with the size of the lattice [59].

We observe a rapid increase of mean transient duration $\langle T \rangle$ with time delay. In Fig. 3, we show the mean transient durations for networks with different numbers $\mathbf{n} = (n,n,n)$ of delay elements in each link, which increases the link time delays. For small delay times (white region of the plot), we observe transients of moderately long duration, i.e., several thousand oscillations.

When the link delays are above ∼2 ns (indicated by the gray region in the figure), a regime of supertransients with typical durations on the order of milliseconds or seconds and beyond is found. Here, the transient durations do not increase monotonically with n. For a network realization with $\mathbf{n} = (5,5,5)$, we find that the transient durations are beyond 40 min for each of 30 experimental runs, at which point we terminate each run. Hence, we can only conclude that the mean transient time exceeds this value. In Fig. 3, this is indicated by an arrow for the data point corresponding to this network. For $\mathbf{n} = (6,6,6)$, the mean lifetimes are roughly 1 s, although there are extremely large variations from run to run. We hypothesize that, in this regime, the observed variation in the mean transient durations are dominated by tiny heterogeneities in link delays and initial conditions due to different routing of wires in the delay lines as we vary n. Further below, we demonstrate that the mean transient durations for fixed n are very sensitive to slight changes in heterogeneity.

From Fig. 3, we furthermore see that the standard deviation increases with delay, which indicates a broadening of the transient duration probability distributions. For small delays, the transient durations are confined to a Gaussian-shaped peak with a short tail, as shown in Fig. 4(a). For longer delays, the distribution of transient durations shows an exponential tail, as displayed in Fig. 4(b). Here, the standard deviation is roughly equal to the mean.

C. Alternative transient patterns

In addition to the in-phase oscillations shown in Fig. 2, we observe transient trajectories in some networks, which correspond to different Boolean switching patterns when depicted on the hypercube. These different transient patterns can coexist for a given network realization and likewise emerge when repeatedly initializing the network. In Fig. 5, we show two different transient trajectories that arise in a network with $\mathbf{n} = (4,4,4)$. While one trajectory represents in-phase oscillations (left), the other corresponds to a switching sequence that visits all states on the hypercube. This is a further indication that the nature and number of coexisting gene expression patterns, which are inferred from our model on observational timescales, differ from the asymptotic state.

D. Sensitivity to heterogeneity

In our experiments with longer link time delays, we observe that the average transient durations $\langle T \rangle$ vary by

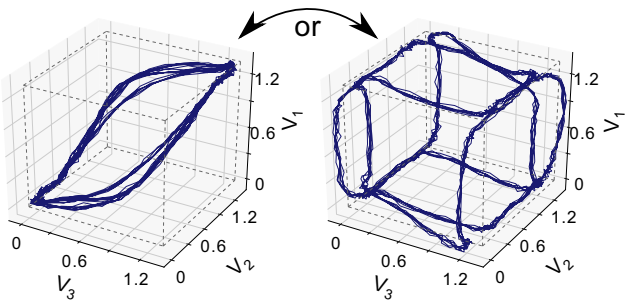

FIG. 5. Phase portrait of two different transient trajectories of a network with $\mathbf{n} = (4,4,4)$ delay elements, both initialized in the (1,1,1) state.

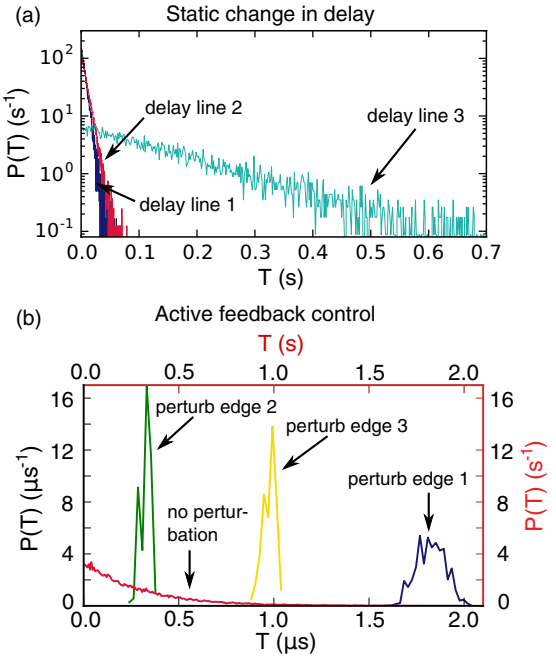

FIG. 6. Transient duration distributions in the presence of heterogeneities and active feedback control. (a) Transient duration distributions in a network with $\mathbf{n} = (7,7,7)$ delay elements when individually altering the timing of the initial conditions of each node by roughly 100 ps. The mean transient durations when delaying nodes 1, 2, and 3 are $\langle T \rangle = 5.93$ ms, $\langle T \rangle = 8.05$ ms, and $\langle T \rangle = 151.46$ ms, respectively. We collected 30 000 samples for each distribution. (b) Transient duration distributions in a network with $\mathbf{n} = (6,6,6)$ delay elements, when perturbing the timing of a specific traveling edge. Note that the time scale for the distribution with no perturbation (red) is seconds.

several orders of magnitude for slightly different on-chip network realizations with the same number of delay elements $\mathbf{n} = (n,n,n)$. We demonstrate that this behavior is a result of changing the fixed or "frozen in" heterogeneities in delay times and initial conditions. We expect these to be on the order of 100 ps, corresponding to $\sim 1/30$ of a typical link delay. We independently alter the heterogeneities of initial conditions or time delays by these small amounts and measure changes in the transient duration distributions. In Fig. 6(a), we show three transient duration distributions obtained by slightly delaying the timing of the initial conditions of one of the nodes as described in Appendix B. The distributions have an exponential tail with means varying by two orders of magnitude. In the experiment with unchanged initial conditions, the mean transient duration is $\langle T \rangle = 0.92$ s. A similar effect is seen as we alter the delay in one of the delay lines by $\sim$100 ps (data not shown here). We conjecture that the larger variation in mean time delay seen in the gray region of Fig. 3 is due to this increased sensitivity to network heterogeneity.

IV. ACTIVE FEEDBACK CONTROL OF TRANSIENT DYNAMICS

It is often important to predict and control the transition of dynamical systems from transient dynamics to a targeted behavior [60]. This has been investigated for technological applications, including turbulent pipe-flows [56] and cascading failures in power grids [61], as well as biological systems, such as ecological models [62] and neural networks [63]. To force our experimental transient dynamics toward the asymptotic attractor, we harness the high sensitivity to small changes of the time delays. This is similar to feedback control methods for chaotic systems, which have been used to stabilize unstable periodic orbits by harnessing the chaotic sensitivity to small, occasional perturbations [64,65].

We apply occasional adjustments to one of the time delays to perturb the timing of selected traveling Boolean transitions. Similar to Blakely *et al.* [66], we detect the signal at one position in the ring and subsequently perturb one of the time delays at a later position. In greater detail, we perturb the timing of a specific traveling Boolean transition (the target edge) every other round trip. This is achieved by implementing delay lines consisting of two alternate paths, which differ in delay time by about 100 ps (less than 10% of a typical link delay time). At a fixed position, before the delay line splits into these two paths, we detect incoming edges and decide whether or not to perturb them. This decision can be made by an asynchronous counter, which counts the number of incoming edges, starting from the target edge as described in greater detail in Appendix C. We prescribe a fixed target value of the counter, reflecting the frequency that we want to perturb the target edge. Whenever this value is reached, we reset the counter and create a short pulse, which leads the delay line to employ the longer path for the target edge. For all other edges, the shorter path is used. For the results presented in Fig. 6(b), we perturbed the target edge once every other round trip.

These targeted perturbations give rise to a rapid decay of the transients for all networks. An example is given in Fig. 6(b) for a network with $\mathbf{n} = (6,6,6)$ delay elements, where, in every delay line, the timing of one of the three traveling edges can be perturbed. When not perturbing any edge, we find an average transient duration of $\langle T \rangle = 0.4$ s, the corresponding distribution has an exponential tail (red curve). Perturbing one of the edges leads to a rapid transient collapse. The transient durations are then confined to a narrow interval around an average that can be as small as $\langle T \rangle = 250$ ns. Thus, the high sensitivity of transient dynamics to heterogeneities can be used to yield a consistently rapid transient collapse when employing targeted perturbations of time delay. Furthermore, the actual time of transient collapse may be controlled by adjusting the application of the occasional perturbations.

V. MATHEMATICAL MODEL OF THE REPRESSILATOR WITH DELAY

To model the dynamics that arise in our experiments, we extend the framework developed by Glass and collaborators [50] by including time delays along the links [67]. We investigate a set of first-order piecewise-linear delay differential equations for a general ring of N nodes,

$$\dot{y}_i(t) = -y_i(t) + F[y_{i-1}(t - \tau_i)], \qquad (1)$$

where i is defined mod N. In this model, the continuous variables $y_i(t)$ relate to the concentration of the gene product of gene i and act as a transcription factor for gene $i + 1$. For the

sake of symmetry, we rescale the dynamics to the interval $y \in [-1,+1]$. The production is regulated by a repressing threshold function,

$$F(y) = \begin{cases} -1 & \text{for} \quad y \geqslant 0 \\ 1 & \text{for} \quad y < 0. \end{cases} \tag{2}$$

Time is scaled in units $\tau_r / \ln 2$, where τ_r is the rise or fall time of the analog signals to the threshold defined in Sec. II B. Each variable is initialized at a time $\tilde{t}_i^0 = \delta_i$, where we choose as initial functions $y_i(t < \tilde{t}_0^i) = 1$, and the link delays τ_i are allowed to have both fixed (frozen in) variation as well as stochastic time-varying behavior.

In the absence of stochastic variation in the delay, but with a small fixed heterogeneity, these differential equations are known to display long transients [39]. Long transient times were also reported in a model for inhibitory ring GRNs that does not incorporate time delays, but instead includes an intermediate step in gene expression, which gives an effective delay time [68].

We begin with a direct analysis of the delay differential Eq. (1) with no delay or initial timing heterogeneities or stochastic effects. We identify the in-phase periodic solution corresponding to the initial transient shown in Fig. 2(a) and find that its period is $P_{\text{in}} = 2(\tau + \ln[2 - e^{-\tau}])$, where τ is the average delay of the three network links. For the asymptotically stable periodic state, we search for an out-of-phase periodic solution and find that the period is $P_{\text{out}} \simeq 6(\tau + \ln[2])$ in the limit as $\tau \to \infty$.

We compare our predictions to the experimental results shown in Fig. 2 for $\mathbf{n} = (2,2,2)$ by converting our expressions back to physical time. We predict that $P_{\text{in}} = 2.79 \pm 0.08$ ns, $P_{\text{out}} = 8.7 \pm 0.24$ ns, where the errors are dominated by our uncertainty in τ_0. The agreement with the experimentally measured values (2.84 ± 0.01 ns and 8.51 ± 0.01 ns, respectively) is good and within our prediction error.

To understand in greater detail the transient dynamics, we continue to ignore the stochastic variation in the delay and follow the approach of Edwards *et al.* [67], where we take advantage of the fact that the system Eq. (1) is piecewise linear. In between switching times $\tilde{t}_i^k$, where $y_i(t)$ changes from a rising to a falling edge and vice versa, the equations can be directly integrated, leading to an exponential evolution toward the ideal Boolean values ± 1,

$$y_i\left(t - \tilde{t}_i^k\right) = \mp 1 + \left(y_i\left(\tilde{t}_i^k\right) \pm 1\right) e^{-(t - \tilde{t}_i^k)}. \tag{3}$$

In the first term, the minus (plus) sign is chosen for a falling (rising) edge. The dynamics is then fully described by the switching times $\tilde{t}_i^k$, and the value of the state variables at these times $y_i(\tilde{t}_i^k)$. Further simplification is possible by defining a new discrete-time variable,

$$\epsilon_i(k) = 1 - \left| y_i\left(\tilde{t}_i^k\right) \right|, \tag{4}$$

which describes the distance of the state variables to the ideal Boolean values $y = \pm 1$ at switching time $\tilde{t}_i^k$. Solving Eq. (3) for $y_i(t) = 0$ leads to a recursive relation,

$$\tilde{t}_i^{k+1} = \tilde{t}_{i-1}^k + \ln\left[2 - \epsilon_{i-1}^k\right] + \tau_i, \tag{5}$$

$$\epsilon_i^{k+1} = \left(2 - \epsilon_i^k\right) e^{-(\tilde{t}_i^{k+1} - \tilde{t}_i^k)}. \tag{6}$$

The initial conditions translate to $\epsilon_i(0) = 0$ and $\tilde{t}_i^0 = \delta_i$.

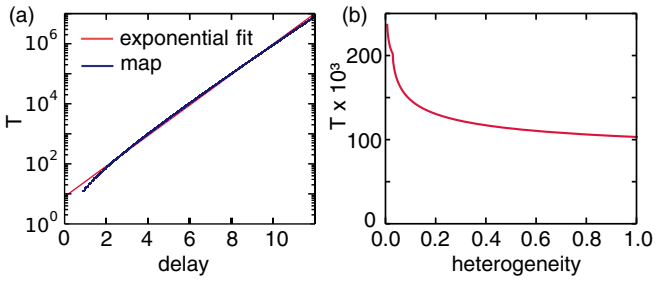

FIG. 7. Transient durations for the map Eqs. (5) and (6) for $N = 3$ with no stochastic fluctuations in the time delay ($\xi = 0$). (a) Transient durations as a function of the mean delay τ for a fixed heterogeneity $\Delta = 0.1$, defined by $\tau_1 = \tau + \Delta/2$, $\tau_2 = \tau$, and $\tau_3 = \tau - \Delta/2$. Differences in initialization timing are not explicitly taken into account, i.e., $\delta_i = 0$. An exponential fit ($T \propto e^{\lambda\tau}$) to the data with exponent $\lambda = 1.17$ is shown in red. (b) Transient durations as a function of delay heterogeneity Δ for a fixed mean delay $\tau = 8$.

The dynamics occur on a fast time scale governed by Eq. (5) related to the nearly periodic in-phase behavior, which drives the dynamics of the slow behavior governed by Eq. (6) and determines the transient duration. We find that the natural logarithm of the absolute value of the Floquet multiplier describing the slow dynamics is given by $3e^{-\tau}/4$ in the limit $\tau \to \infty$, explaining the extremely long transient time.

We next investigate the scaling of the transient durations in this model with delay time and fixed heterogeneity between the time delays. We numerically simulate Eq. (6) until $\epsilon_i(K) > 1$, or, equivalently, $\tilde{t}_i^K < \tilde{t}_i^{K-1}$, meaning that two edges collide and thus the transients terminate. For a mean link delay τ, the duration of the transient can then be estimated as the number of transitions K multiplied by half the initial period oscillations, $T \approx K P_{\text{in}}/2$. For fixed heterogeneity, we find that the transient duration scales approximately exponentially with the mean delay time, as depicted in Fig. 7(a) and in agreement with our estimate based on the Floquet exponent. In Fig. 7(b) we vary the delay heterogeneity for a given mean delay time. We see that the system shows a sensitive dependence on heterogeneity for small differences between initial conditions or link delays, and the mean transient duration diverges for vanishing heterogeneity. Moreover, a timing change of initial conditions has the same effect as fixed heterogeneous time delays. This allows for the possibility that different heterogeneities cancel, yielding super-long transients in agreement with the observations in our experiments.

The reduction of the set of delay differential equations given in Eq. (1) to a map Eqs. (5) and (6) also offers a framework to study the role of stochastic variations in the delay time in a straightforward way. Here we consider the simplest case of a stochastically varying link time delays by modeling the delay term as $\tau_i(k) = \tau_i + \xi_i(k)$, with $\xi_i(k)$ a white Gaussian noise term with zero mean and variance σ^2. The system is then described as

$$\tilde{t}_i^{k+1} = \tilde{t}_{i-1}^k + \ln\left[2 - \epsilon_{i-1}^k\right] + \tau_i + \xi_i(k), \tag{7}$$

$$\epsilon_i^{k+1} = \left(2 - \epsilon_i^k\right) e^{-(\tilde{t}_i^{k+1} - \tilde{t}_i^k)}. \tag{8}$$

We numerically evaluate the map Eqs. (7) and (8) with increasing mean time delays and constant noise strength σ^2.

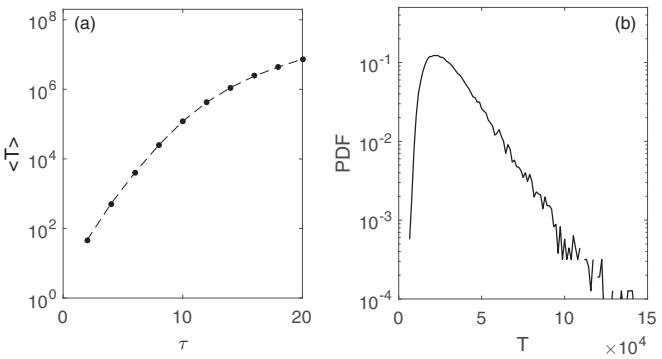

FIG. 8. Transient durations for the map Eq. (6) for $N = 3$. (a) Mean transient duration as a function of the mean delay τ for $\sigma^2 = 10^{-4}$. (b) Probability density function of the transient duration for $\tau = 10$ and $\sigma^2 = 10^{-4}$.

It is seen that the mean transient duration scales exponentially with the mean link delay for the shorter delays, as is illustrated in Fig. 8(a). For longer mean delay, the scaling is subexponential where, we conjecture, the stochastic term in Eq. (7) plays a more important role and reduces the rate of increase but does not suppress the long transient. We confirm this conjecture by increasing the noise strength and find that the break from exponential scaling occurs at shorter mean time delays (data not shown). We also find that the stochastic variation in the link delays leads to a distribution of the transient durations similar to the experimental observations as shown in Fig. 8(b). Here, we see that the distribution is broad with an exponential tail for a long mean time delay. For shorter time delays (not shown), the distribution resembles a gaussian. A more complete investigation of this model, including the role of heterogeneities, size of the ring network and noise strength, is published elsewhere [69].

VI. DISCUSSION

We demonstrate that including time delays in ABN models induces long-lasting transient dynamics that are qualitatively different from the asymptotic attractors. If we transfer this result to GRNs, our findings suggest that the asymptotic gene expression patterns may not always be important for cell fate. Instead, the distinct transient patterns might dominate on observational timescales, which are then linked to different cell types or functions. While transients with hundreds to thousands of oscillations are found for time delays on the order of 1–4 node response times, a dynamical regime of extreme transients emerges at delay times on the order of 5 node response times and beyond. When comparing timescales to real GRNs, one has to consider various processes within gene expression. Some of them have been characterized as time delays, such as mRNA splicing, which can be on the order of the actual gene product lifetime [30,70]. Additionally, there are many more processes that translate into time delay in our model, such as elongation [71], actual mature mRNA production rate, translational delays, and many more. As a result, we believe our studies reflect a biologically plausible parameter regime with regards to time-delay along the network links.

With our experimental platform, we show that the mean transient durations are very sensitive to small changes in the heterogeneities of time delays and initial conditions. Despite this high sensitivity to heterogeneity, our findings are not restricted to time delays with very small heterogeneities. Instead, the qualitative results for the transient dynamics remain unchanged when we implement heterogeneous delays $\mathbf{n} = (n_1,n_2,n_3)$, with $n_1 \neq n_2 \neq n_3$ in the delay lines, assuming that the relative difference of the time delays does not exceed approximately 50%. This robustness is important because heterogeneities in the time delays of real GRNs might be substantial. On the other hand, we demonstrate that applying small occasional perturbations to one of the time delays can indeed rapidly force the transient trajectories toward the asymptotic attractors.

The results presented in this paper are not restricted to the repressilator network. We have carried out the same experiments for repressing ring networks with $N = 2$ [69], representing a toggle-switch [72], and with $N = 4$. Both networks are asymptotically nonoscillatory, but possess two coexisting steady states. We observe the same qualitative transient dynamics: As delays are included along the links, oscillatory transients are induced, which can be extremely long for certain parameter regimes and obey long-tailed distributions. Furthermore, the transient dynamics are sensitive to small changes in heterogeneities and may be controlled with our proposed scheme.

A theoretical model of our experimental ABN using piecewise-linear delay differential equations is consistent with our essential experimental observations. In particular, we predict a sensitive dependence of transient durations on heterogeneities of time delays and initial conditions. For fixed heterogeneities, we furthermore find an exponential scaling with delay time. This is not clearly seen in the experiment due to the fact that we cannot keep the heterogeneities fixed as we vary the time delays. When taking stochastic variation of the time delay into account, this exponential scaling is reproduced for the shorter time delays and saturates at longer delay. Moreover, we recover in our model a broad distribution of transient times with an exponential tail. In other work [69], we further investigate the role of noise and asymmetric thresholds in the model to help understand in more detail the experimental observations.

We conclude that link delays, as often found in biological networks, along with small heterogeneity, give rise to different super-long transient patterns in small network motifs that are distinct from the eventual network attractors. This suggests that these transient patterns, as opposed to the asymptotic dynamical network states, could have a biological relevance. Specifically, the gene expression patterns inferred from GRN models, that are linked to types and functional states of cells, might not be reached on observational time scales. We also show that the observed transient patterns are very sensitive to tiny changes in the heterogeneities of delay, initial conditions, and stochastic behavior, which can be harnessed for controlling the dynamics toward the asymptotic attractors by employing occasional perturbations of the time delays. Future research is needed to investigate the effects of much larger noise than in

our experimental system that is typically found in biological networks.

ACKNOWLEDGMENTS

We gratefully acknowledge discussions of this work with J. Socolar and L. Glass, and the financial support by DAAD and TU Berlin, and the U.S. Army Research Office Grant No. W911NF12-1-0099. This work was partially supported by DFG in the framework of SFB 910.

APPENDIX A: REALIZING THE REPRESSILATOR ON AN FPGA

In this appendix, we present the hardware description code for the repressilator, written in *Verilog* and compiled using Altera's *Quartus II*, and then downloaded to the FPGA. The *Verilog* code is given in Fig. 9.

In line 1, the module **repr_del_het_ff** is declared, which defines the network. It is part of a much larger circuit on the chip, which will not be specified here. This circuit is responsible for starting and stopping the initialization of the network, for detecting the end of the transient, and for sending and receiving data from the computer via a USB chip, which is connected to a general purpose I/O (GPIO) port on the FPGA board. Additionally, the Booleanized dynamics are often written to on-chip RAM in discrete time steps.

```
1    module repr_del_het_ff(
2      dynamics,
3      clock,
4      set_enable,
5      set
6    );
7    input set_enable, clock;
8    input [2:0] set;
9    output [2:0] dynamics;
10   wire [2:0] net, init/*synthesis keep*/;
11
12   parameter n0 = 7;
13   parameter n1 = 7;
14   parameter n2 = 7;
15
16   wire [2*n0-1:0] delay0/*synthesis keep*/;
17   wire [2*n1-1:0] delay1/*synthesis keep*/;
18   wire [2*n2-1:0] delay2/*synthesis keep*/;
19   genvar i;
20   generate
21      for (i=0;i<3;i=i+1)
22         begin : generate_flops
23            DFF init_ff (
24            .d(set_enable),
25            .clk(clock),
26            .q(init[i])
27            )/*synthesis preserve*/;
28         end
29   endgenerate
30   assign delay0[0] = _net[0];
31   assign delay1[0] = _net[1];
32   assign delay2[0] = _net[2];
33
34   generate
35      for (i=0;i<2*n0-1;i=i+1)
36         begin : generate_delay0
37            assign delay0[i+1] = _delay0[i];
38         end
39   endgenerate
40   generate
41      for (i=0;i<2*n1-1;i=i+1)
42         begin : generate_delay1
43            assign delay1[i+1] = _delay1[i];
44         end
45   endgenerate
46   generate
47      for (i=0;i<2*n2-1;i=i+1)
48         begin : generate_delay2
49            assign delay2[i+1] = _delay2[i];
50         end
51   endgenerate
52
53   assign net[0] = (_delay1[2*n1-1]&_init[0])|(set[0]&init[0]);
54   assign net[1] = (_delay2[2*n2-1]&_init[1])|(set[1]&init[1]);
55   assign net[2] = (_delay0[2*n0-1]&_init[2])|(set[2]&init[2]);
56
57   assign dynamics = net;
58
59   endmodule
```

FIG. 9. *Verilog* code defining a ring network with $N = 3$ inhibitory nodes and heterogeneous link time delays.

The module **repr_del_het_ff** has the output port **dynamics**, which is 3 bits wide and connects the dynamics of the three network nodes to other modules of the circuit, or directly to the output buffers leading to the SMA connectors on the FPGA board and an oscilloscope. There are three input ports. The input **clock** feeds the clocked registers used to initialize the network from the on-board clock source. The input **set_enable** is the set signal, which determines when the initial conditions are set, and which is fed to the clocked registers. The input **set** is 3 bits wide and determines the initial conditions of each node. These ports are declared in lines 7–9.

In line 10, two 3 bit wide *wire-type* signals **net** and **init** are declared. *Wire-type* signals are used for either connecting ports of different modules or for the implicit creation of **unclocked** logic gates when they are assigned to a logic expression in an **assign** statement. The wire **net** is used to define the logic gates of the network nodes, while the wire **init** connects the clocked registers with the network nodes. After the *wire* declaration in line 10, the attribute **/*synthesis keep*/** is used to prevent the compiler from removing the declared wires.

In lines 12–14, three parameters **n0**, **n1**, and **n2** are specified, which correspond to the number of delay elements in each delay line. In lines 16–18, the wires defining the logic gates for the three delay lines are declared. Each wire is $2n$ bits wide, corresponding to the $2n$ inverter gates that constitute a delay line of n delay elements.

Lines 19–29 create multiple statements or instantiations in *Verilog* using a loop, which is called a **generate** block. In this environment, three instantiations of **DFF**, a predefined primitive, generate d-type flip-flops, which are the clocked registers initializing the network nodes. A d-type flip-flop assigns its input signal d to its output q whenever a rising edge at its input clk is detected. These three ports are connected to the external inputs **clock** of the clock signal and **set_enable**, which comes from another part of the circuit and determines when the initial conditions are set and released, and the wire-type signal **init**, which is connected to the network nodes. These **generate** blocks are an important tool for creating large, scalable network designs, and one reason why this text-based approach of creating the circuits is powerful.

Lines 30–32 contain three **assign** statements used to create autonomous logic gates from wire-type signals. Here, the first elements of the three delay lines are assigned to the inverse of the network nodes using the symbol "∼", which is one way to implement the Boolean NOT function. In lines 34–51, the remaining logic gates of the delay lines are created in generate blocks by using **assign** statements to assign each new element to the inverse of the previous element in the delay line. This creates three delay lines consisting of cascades of an even number of inverter gates.

Lines 53–55 define the network nodes. In addition to implementing Boolean NOT functions, they also process the signals that initializes the network, which is is done by

formulating a Boolean expression with the operators "∼", "&", and "—" for NOT, AND, and OR, respectively. Thus, the network nodes take on the values of the initial conditions **set** whenever the signal **init**, coming from the clocked registers, is high, while the network nodes invert the last element of the adjacent delay line when **init** is low, corresponding to the free-running autonomous dynamics of the ring network. Finally, in line 57, the output **dynamics** is connected to the network nodes.

APPENDIX B: SPECIFICATION OF ON-CHIP PLACEMENT AND ROUTING

We specify the physical positions of the logic elements on the chip, which is important for controlling the heterogeneity of the link time delays. The logic elements are assembled in so-called logic array blocks (LABs) consisting of 16 elements each, which form a two-dimensional grid. Thus, every logic element has three coordinates: X, Y, and M, where X and Y define the position on the two-dimensional grid, while M defines the position of the element within a LAB. Even M correspond to the combinatorial logic elements, while odd M correspond to the flip-flops that belong to each logic element for synchronous operation. We set the specific location of each network element using a **set_location_assignment** command.

In many cases, it is also important to specify and keep fixed the routing connections that are used in between logic elements, especially the routing channels going from the end of one delay line to the next network node. While this is usually handled by the compiler, it is possible to manually assign a connection to a specific wire by writing a *routing constraint file*, which then constrains the routing channels that can be chosen by the compiler. Furthermore, logic elements in adjacent LABs are connected via so-called *local interconnect* channels which are specified by the keyword **LOCAL_INTERCONNECT:...**, followed by a coordinate of the wire. We constrain these lines for all of our work.

APPENDIX C: SCHEME FOR THE OCCASIONAL PERTURBATION OF TRAVELING EDGES

To greatly shorten the transient dynamics of the repressilator, we occasionally perturb the time-delay in one of the network links whose timing is based on detecting a specific Boolean transition at a fixed location along the network ring. As described below, we develop a method for detecting the transition, which is complicated by the fact that the orientation of the target edge alternates every round trip. The goal of the scheme is to only perturb the target edge every other round trip and to deactivate the perturbation mechanism for the time in between.

Our method is illustrated in Fig. 10. It employs an asynchronous mod-3 counter, which counts the number of edges with the target orientation that pass the measurement point after the target edge has been perturbed. When two edges are counted, the perturbation mechanism is activated again. The asynchronous mod-3 counter is built with two d-type flip-flops that have an asynchronous clear input (denoted by "clr"). We connect the measurement point to the clock input of the first flip-flop. The inverted output of the first flip-flop

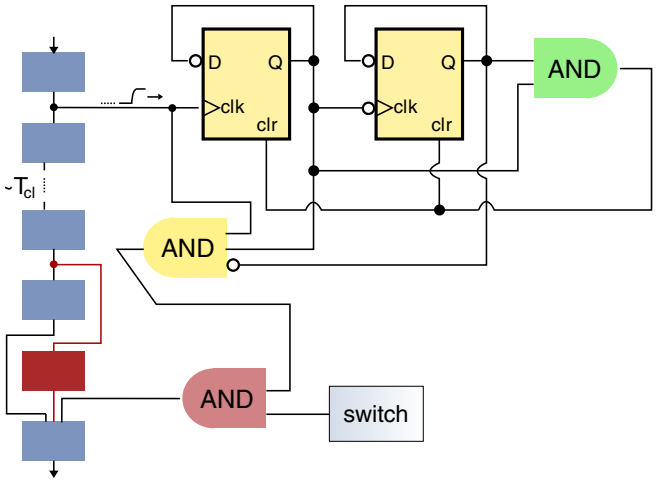

FIG. 10. Scheme for perturbing a single Boolean transition every other round trip using a mod-3 asynchronous counter. The blue rectangles represent a NOT gate an the yellow rectangles represent the d-type flip-flops.

is connected to its own input so that the output toggles in between "0" and "1" at every positive edge it detects at the clock input. Thus, it behaves as the least significant bit of a counter, which counts the positive edges. The inverted output of the first flip-flop is furthermore connected to the clock input of the second flip-flop. In the second flip-flop, we again connect the inverted output to its own input. Thus, the output of the second flip-flop toggles in between "0" and "1" whenever there is a negative edge in the output of the first flip-flop. It therefore behaves like the second least significant bit of a counter, which counts the positive edges at the clock input of the first flip-flop. The combined output of the two flip-flops constitute an asynchronous counter, which can count from 0 to 3. We make it a mod-3 counter by resetting both flip-flops to "0" as soon as a count of three is reached, corresponding to an output of "1" of both flip-flops. This is accomplished by combining the two flip-flops in an AND gate (marked green in Fig. 10) and connecting this gate to the asynchronous clear inputs of the flip-flops.

We let the counter start counting with the target edge so that a count equal to 1, in combination with the detection of the right edge orientation at the measurement point, activates the perturbation mechanism. This is done by combining the counter state with the signal from the measurement point in a 3-input AND gate, whose output is the perturbation signal that selects between the two alternate paths in the delay line. We add another AND gate that can completely deactivate the perturbation mechanism with an on-board switch.

The hardware description code for the control perturbations is largely based on the code given in Fig. 9. Additional code is used to enable or disable the perturbations using a combination of Boolean functions as outlined above. For both control schemes, the propagation time of the signals from the measurement point through the two alternate paths to the multiplexer has to match the control loop latency, i.e., the propagation delay through the logic determining whether a perturbation should be applied. This is an important issue for controlling the repressilator because there is significant

control-loop latency so that we have to measure the edges at an earlier point in the delay line [66]. Ideally, the perturbation signal arrives at the multiplexer one characteristic rise time before the arrival of the signal from the shorter path because we want to avoid perturbing an earlier edge, which might have come close to the target edge.

[1] M. Ghil, A. Mullhaupt, and P. Pestiaux, Clim. Dynam. **2**, 1 (1987).
[2] T. M. H. Wohlleben and A. J. Weaver, Clim. Dynam. **11**, 459 (1995).
[3] I. Zaliapin, V. Keilis-Borok, and M. Ghil, J. Stat. Phys. **111**, 815 (2003).
[4] M. Ghil, I. Zaliapin, and B. Coluzzi, Physica D **237**, 2967 (2008).
[5] K. E. Kürten, J. Phys. A: Math. Gen. **21**, L615 (1988).
[6] S. Bornholdt and T. Rohlf, Phys. Rev. Lett. **84**, 6114 (2000).
[7] R. Albert and H. G. Othmer, J. Theoret. Biol. **223**, 1 (2003).
[8] M. Chaves, R. Albert, and E. D. Sontag, J. Theoret. Biol. **235**, 431 (2005).
[9] A. Chaos, M. Aldana, C. Espinosa-Soto, B. G. P. de León, A. G. Arroyo, and E. R. Alvarez-Buylla, J. Plant Growth Regul. **25**, 278 (2006).
[10] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, Proc. Natl. Acad. Sci. USA **101**, 4781 (2004).
[11] S. A. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, Proc. Natl. Acad. Sci. USA **100**, 14796 (2003).
[12] M. D. Stern, Proc. Natl. Acad. Sci. USA **96**, 10746 (1999).
[13] S. Bornholdt and K. Sneppen, Proc. R. Soc. London B **267**, 2281 (2000).
[14] S. A. Kauffman and E. D. Weinberger, J. Theoret. Biol. **141**, 211 (1989).
[15] B. Coluzzi, M. Ghil, S. Hallegatte, and G. Weisbuch, Int. J. Bif. Chaos **21**, 3511 (2011).
[16] M. Paczuski, K. E. Bassler, and A. Corral, Phys. Rev. Lett. **84**, 3185 (2000).
[17] F. Jacob and J. Monod, J. Mol. Biol. **3**, 318 (1961).
[18] S. A. Kauffman, J. Theoret. Biol. **22**, 437 (1969).
[19] S. Huang and D. E. Ingber, Exp. Cell Res. **261**, 91 (2000).
[20] E. Gehrmann and B. Drossel, Phys. Rev. E **82**, 046120 (2010).
[21] K. Klemm and S. Bornholdt, Proc. Natl. Acad. Sci. USA **102**, 18414 (2005).
[22] F. Ghanbarnejad and K. Klemm, Phys. Rev. Lett. **107**, 188701 (2011).
[23] K. Klemm and S. Bornholdt, Phys. Rev. E **72**, 055101 (2005).
[24] P. Shahrear, L. Glass, R. Wilds, and R. Edwards, Math. Comput. Simul. **110**, 33 (2015).
[25] T. J. Perkins, R. Wilds, and L. Glass, Phil. Trans. R. Soc. A **368**, 4961 (2010).
[26] X. Cheng, M. Sun, and J. E. S. Socolar, J. R. Soc. Interface **10**, 20120574 (2013).
[27] M. Ghil and A. Mullhaupt, J. Stat. Phys. **41**, 125 (1985).
[28] T. Mestl, C. Lemay, and L. Glass, Physica D **98**, 33 (1996).
[29] R. Edwards, A. Machina, G. McGregor, and P. van den Driessche, Bull. Math. Biol. **77**, 953 (2015).
[30] A. Honkela, J. Peltonen, H. Topa, I. Charapitsa, F. Matarese, K. Grote, H. G. Stunnenberg, G. Reid, N. D. Lawrence, and M. Rattray, Proc. Natl. Acad. Sci. USA **112**, 13115 (2015).
[31] K. P. Hadeler and J. Tomiuk, Arch. Rat. Mech. Anal. **65**, 87 (1977).
[32] C. M. Marcus and R. M. Westervelt, Phys. Rev. A **39**, 347 (1989).
[33] V. Flunkert, I. Fischer, and E. E. Schöll, Theme Issue of Phil. Trans. R. Soc. A **371**, 20120465 (2013).
[34] T. Tomida, M. Takekawa, and H. Saito, Nat. Commun. **6**, 8350 (2015).
[35] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, Nature (London) **456**, 516 (2008).
[36] N. A. M. Monk, Curr. Biol. **13**, 1409 (2003).
[37] D. Bratsun, D. Volfson, L. S. Tsimring, and J. Hasty, Proc. Natl. Acad. Sci. USA **102**, 14593 (2005).
[38] Y. Horikawa and H. Kitajima, Physica D **238**, 216 (2009).
[39] K. Pakdaman, C. Grotta-Ragazzo, and C. P. Malta, Phys. Rev. E **58**, 3623 (1998).
[40] P. Baldi and A. F. Atiya, IEEE Trans. Neural Networ. **5**, 612 (1994).
[41] K. L. Babcock and R. M. Westervelt, Physica D **28**, 305 (1987).
[42] D. P. Rosin, D. Rontani, D. J. Gauthier, and E. Schöll, Chaos **23**, 025102 (2013).
[43] D. P. Rosin, *Dynamics of Complex Autonomous Boolean Networks* (Springer, Heidelberg, 2015).
[44] Y. Chen, J. K. Kim, A. J. Hirning, K. Josić, and M. R. Bennett, Science **349**, 986 (2015).
[45] H. Hirata, S. Yoshiura, T. Ohtsuka, Y. Bessho, T. Harada, K. Yoshikawa, and R. Kageyama, Science **298**, 840 (2002).
[46] R. R. Klevecz, J. Bolen, G. Forrest, and D. B. Murray, Proc. Natl. Acad. Sci. USA **101**, 1200 (2004).
[47] G. Tiana and M. H. Jensen, Phil. Trans. R. Soc. A **371**, 20120469 (2013).
[48] S. Pigolotti, S. Krishna, and M. H. Jensen, Proc. Natl. Acad. Sci. USA **104**, 6533 (2007).
[49] M. B. Elowitz and S. Leibler, Nature (London) **403**, 335 (2000).
[50] R. Edwards and L. Glass, Adv. Chem. Phys. **132**, 151 (2005).
[51] T. Tél, Chaos **25**, 097619 (2015).
[52] Y.-C. Lai and T. Tél, *Transient Chaos* (Springer, Berlin, 2010).
[53] A. Wacker, S. Bose, and E. Schöll, Europhys. Lett. **31**, 257 (1995).
[54] T. Tél and Y.-C. Lai, Phys. Rep. **460**, 245 (2008).
[55] J. P. Crutchfield and K. Kaneko, Phys. Rev. Lett. **60**, 2715 (1988).
[56] B. Hof, J. Westerweel, T. M. Schneider, and B. Eckhardt, Nature (London) **443**, 59 (2006).
[57] D. P. Rosin, D. Rontani, N. D. Haynes, E. Schöll, and D. J. Gauthier, Phys. Rev. E **90**, 030902 (2014).
[58] M. Wolfrum and O. E. Omel'chenko, Phys. Rev. E **84**, 015201(R) (2011).
[59] A. Politi and A. Torcini, in *Nonlinear Dynamics and Chaos: Advances and Perspectives*, edited by M. Thiel, J. Kurths, M. C. Romano, and G. Károlyi (Springer, Berlin/Heidelberg, 2010), pp. 103–129.
[60] A. Motter, Chaos **25**, 097621 (2015).
[61] I. Simonsen, L. Buzna, K. Peters, S. Bornholdt, and D. Helbing, Phys. Rev. Lett. **100**, 218701 (2008).
[62] A. Hastings and K. Higgins, Science **263**, 1133 (1994).
[63] M. Rabinovich, R. Huerta, and G. Laurent, Science **321**, 48 (2008).

[64] E. R. Hunt, Phys. Rev. Lett. **67**, 1953 (1991).
[65] R. Roy, T. W. Murphy, Jr., T. D. Maier, Z. Gills, and E. R. Hunt, Phys. Rev. Lett. **68**, 1259 (1992).
[66] J. N. Blakely, L. Illing, and D. J. Gauthier, Phys. Rev. Lett. **92**, 193901 (2004).
[67] R. Edwards, P. van den Driessche, and L. Wang, J. Math. Bio. **55**, 271 (2007).
[68] N. Strelkowa and M. Barahona, Chaos **21**, 023104 (2011).
[69] O. D'Huys, J. Lohmann, N. D. Haynes, and D. J. Gauthier, Chaos **26**, 094810 (2016).
[70] S. Hao and D. Baltimore, Proc. Nat. Acad. Sci. USA **110**, 11934 (2013).
[71] C. G. Danko, N. Hah, X. Luo, A. L. Martins, L. Core, J. T. Lis, A. Siepel, and W. L. Kraus, Mol. Cell **50**, 212 (2013).
[72] T. S. Gardner, C. R. Cantor, and J. J. Collins, Nature (London) **403**, 339 (2000).