



Article submitted to journal

Subject Areas:

DNA Computing, Molecular computing

Keywords:

Shadow cancellation, leaks, DNA strand displacement, dynamical systems

Author for correspondence:

Corresponding author

e-mail: rajivteja.nagipogu@duke.edu

Leak-resilient Enzyme-free Nucleic Acid Dynamical Systems through Shadow Cancellation

Rajiv Teja Nagipogu¹, John H. Reif¹

¹Department of Computer Science, Duke University, Durham, NC 27708, United States

DNA strand displacement (DSD) emerged as a prominent reaction motif for engineering various nanoscale molecular computation devices. However, these reaction systems are affected by noisy interactions known as leaks, which significantly affect their scale and sensitivity by interacting with the circuit downstream. This cascading effect is more visible in circuits with complex dynamics, such as feedback loops, as the leak would be subject to nonlinear amplification. Static design-level techniques are primarily preventive and are, therefore, inadequate as they cannot control the amplification once a leak materializes in the circuit. Tackling this issue, our group previously introduced a dynamic leak-elimination technique known as ‘shadow cancellation’ and demonstrated its effectiveness in a cross-catalytic amplifier circuit. Our contributions to this work are as follows. First, we propose a semi-automated and heuristic technique that makes shadow cancellation accessible to arbitrary strand displacement circuits. Then, we use domain-level simulations to show that the shadow cancellation methodology can improve the dynamics of the DSD circuits of four quintessential dynamical behaviors, including autocatalytic amplification, oscillations, consensus, and control. Further, through various probing experiments, we evaluate the robustness of the overall methodology to the underlying practical considerations involved in the shadow cancellation-enabled circuit design. Our work provides a path toward implementing robust and durable dynamical behaviors using strand displacement reactions.

1. Introduction

Intricate networks of biochemical interactions drive the biological processes responsible for the survival and functioning of an organism. Reproducing the dynamics of such sophisticated processes *in vitro* could enable molecular-scale sensing and actuation [1], with applications such as molecular diagnostics [2, 3], and smart therapeutics [4]. The development of such systems requires the construction of modular chemical building blocks that are robust to noise. We differentiate between two qualitatively different approaches to solving this problem. First is the “synthetic biology” approach, which involves carefully studying constituent reaction networks of a process and utilizing this knowledge to develop increasingly complex synthetic chemical systems [5]. In contrast, the field of DNA Computing (also known as molecular computing) takes an “engineering” approach, attempting to build coarse functional approximations of said systems with more emphasis on implementation. The ‘DNA’ in DNA Computing refers to using DNA-DNA reactions for simulating chemical dynamics.

DNA is an excellent substrate for simulating chemical behavior, as the specificity inherent in its base pair hybridization leads to programmable reaction pathways. In a prototypical construction of a DSD circuit, chemical behavior, specified as an abstract reaction system known as a chemical reaction network (CRN) [6–8], is systematically translated into a set of DNA reactions. In this regard, a reaction motif known as DNA strand displacement (DSD; strand displacement) [9, 10] has become prominent due to its design simplicity and flexibility to tune reaction rates. In a typical strand displacement reaction, a single-stranded (ssDNA) *invader* displaces another ssDNA *incumbent* from a partially double-stranded (dsDNA) *gate complex*. The reaction proceeds with the invader first latching onto a single-stranded overhang in the gate complex known as the *toehold* and reversibly exchanging the base pairs with the incumbent. Numerous strand displacement circuits have been engineered for a variety of molecular devices, including logic gates [11–13], analog computation [14], dynamical systems [15–21], and neural network computation [22, 23].

Despite this versatility, strand displacement circuits suffer from intrinsic background noise known as *leaks*, the undesired spurious displacement events due to erroneous interactions among the substrates. One of the dominant leak pathways is *fraying*, where the base pairs at the helix edges unbind spontaneously, opening a short toehold, which then attracts an invading trigger, causing strand displacement. Leaks adversely affect the sensitivity and scale of the circuit as the leaked strands interact with the circuit downstream, resulting in undesired behaviors. This cascading effect is much more severe in circuits with dynamical behaviors, such as feedback loops, as the leaked strands undergo exponential amplification, causing the circuit to degrade rapidly (for example, the “Rock-Paper-Scissors” oscillator in [17]).

Much work has been done to understand the leak pathways and develop design principles that reduce their occurrence [24]. Sequence-level modifications include short clamp domains [17, 25, 26] at the helix edges to prevent fraying. Domain-level modifications include using long domains and incorporating redundancy such that the leak pathway is preceded by multiple low-probability events [27]. Circuit-level modifications include localizing the interacting strands on a DNA origami to reduce unwanted side reactions. While these modifications effectively diminish the probability of a leak event, they cannot curtail the leak amplification once it materializes in the circuit.

Tackling this issue, Song *et al.* [16] proposed a dynamic leak-eliminating strategy known as *shadow cancellation*. This method works by installing an orthogonal *shadow circuit* alongside the original *primary circuit* so that the leaks from the two circuits cancel each other quickly. To accomplish this, the inputs to the shadow circuit are zeroed out such that the signal activity in the shadow circuit is primarily due to leaks. By designing the shadow circuit to have a near-identical rate profile and leak characteristics as the primary circuit, the activity in the shadow circuit could be interpreted as the leak activity in the primary circuit. Finally, a fast cancellation mechanism ensures that any leak generated in the two circuits is quickly sequestered so that the activity in the shadow circuit remains close to zero (and, by extension, the leak activity in the primary circuit, thereby restoring the circuit to normalcy).

The rest of the paper is organized as follows. Section 2 overviews the React-Produce framework that translates our CRNs into a DSD reaction system. We discuss the construction of the DSD circuits for general unimolecular and bimolecular reactions in this framework and utilize them as templates for later constructions. We will also overview the leak mechanisms native to this framework. Then, we describe a heuristic methodology for automatically constructing the shadow circuit, given only the primary circuit’s description (domains, domain arrangement, and nucleotide sequences). Section 3 presents the domain-level simulation results of our target dynamical systems. Finally, in Section 4, we demonstrate that the shadow cancellation methodology is adequately robust to the practicalities of our assumptions regarding the methodology through various probing experiments.

2. Methods

(a) CRN to DSD translation

Several translation schemes have been proposed to convert arbitrary CRNs into a DSD reaction system [13, 17, 21, 28, 29]. In this work, we use the *React-Produce* framework proposed by Srinivas *et al.* [17] to model the DSD circuits of our dynamical systems. This framework models each formal reaction of the CRN into a two-phase pipeline—React and Produce. *React* phase reactions consume the reactant signal strands, whereas the *Produce* phase reactions release

the product strands. This framework employs three kinds of DNA substrate molecules to facilitate the conversion: (i) signal strands (ssDNA), (ii) gate complexes (dsDNA), and (iii) auxiliary strands (ssDNA). The *signal strands* directly map to the formal CRN species, whereas the *gate complexes* and the *auxiliary strands* collectively represent the fuel species that mediate the interactions among the signal strands by providing raw material and the necessary free energy required to drive the reaction.

At the domain level, each DNA substrate in this framework is constructed using four classes of domains: (i) the first toehold (fX; 7 nt), (ii) the migration domain (mX; 15 nt), (iii) the second toehold (sX; 7 nt), and (iv) the history domain (hXi; 15 nt), where X is a placeholder for a formal species of the CRN. The first three domains (fX, sX, and mX) uniquely represent a formal signal species (X), whereas the history domain can vary (signified by the *i* in hXi) among the signal strands of the same species.

(i) Bimolecular reaction design in the React-Produce framework

Here, we describe the React-Produce implementation of a general bimolecular reaction (BIMOL) ($U + V \xrightarrow{k} X + Y$). As this translation serves as an overarching template for the rest of our constructions, we discuss it in more detail and present the other constructions as modifications of this translation. We refer the reader to the original work [17] for an exhaustive description of the framework.

First, the CRN of the BIMOL reaction is rewritten using signal strands as $U_i + V_j \xrightarrow{k} X_n + Y_o$, where the signal strands (U_i , V_j , X_n , and Y_o) directly represent the formal species U, V, X, and Y respectively. The reaction serves as a condensed version of the final DSD circuit, implying that the signal strands U_i and V_j are consumed during the React phase and the signal strands X_n and Y_o are released during the Produce phase. Additionally, the circuit consists of two gate complexes (React_{UVX_n} and $\text{Produce}_{VX_nY_o}$), one each for the React and Produce phases, and two auxiliary strands Back_{UV} and Helper_{Y_o} .

In the first reaction of the React phase (Figure 1A), the first reactant (U_i) reversibly reacts with the React complex (React_{UVX_n}) to release the auxiliary Backward strand (Back_{UV}) and an intermediate React complex (ReactInt_{UVX_n}). Since the fuel species React_{UVX_n} and Back_{UV} are in excess, ReactInt_{UVX_n} quickly approaches a pseudo-equilibrium with a concentration proportional to the concentration of U_i . In the second (and final) reaction of the React phase (Figure 1B), the second reactant strand (V_j) reacts with ReactInt_{UVX_n} to release a “Flux” strand (Flux_{VX_n}). The release of the Flux strand initiates the Produce phase. In the first reaction of the Produce phase (Figure 1C), Flux_{VX_n} strand reversibly binds to the Produce complex ($\text{Produce}_{VX_nY_o}$) and releases the first product strand (X_n) and the intermediate Produce complex ($\text{ProduceInt}_{VX_nY_o}$). In the second (and final) reaction (Figure 1D) of the Produce phase, the auxiliary “Helper”

strand (Helper_{Y_o}) binds to the toehold (fY) and displaces the second product strand (Y_o), completing the circuit.

Applying the law of mass action kinetics to the BIMOL DSD circuit, we obtain the following approximate rate law for the two product strands X_n and Y_n

$$\frac{d[Y_o]}{dt} = \frac{k_{1f}}{k_{1b}} k_2 [U_i][V_j] \quad (2.1)$$

$$\frac{d[X_n]}{dt} = \frac{k_{1f}}{k_{1b}} k_2 [U_i][V_j] \quad (2.2)$$

This rate law is approximate in that we make the following simplifying assumptions in its derivation: (i) the fuel species are at a much higher concentration than the signal strands, and their concentrations stay relatively the same for the entirety of the experiment, and (ii) the reaction intermediates are present at lower concentrations than the signal strands and are at pseudoequilibrium. Note that the rate equations (2.2) and (2.1) agree with the ideal bimolecular rate law for $U + V \xrightarrow{k} X + Y$ ($\frac{dY}{dt} = \frac{dX}{dt} = k[U][V]$), when the rate constant of the ideal reaction $k = \frac{k_{1f}}{k_{1b}} k_2$. A detailed derivation of the rate law for the BIMOL DSD circuit is provided in Supplementary Information (A.1).

(ii) Unimolecular reaction design in the React-Produce framework

The DSD circuit of the general unimolecular reaction (UNIMOL; $U \xrightarrow{k} X + Y$) is similar to the BIMOL DSD circuit, except an auxiliary Helper strand (Helper_V) is used in the place of the second reactant (V_j) signal strand (also suggested by [17]).

Here, we describe the DSD circuit for an elementary unimolecular reaction ($U \xrightarrow{k} X + Y$), hereafter referred to as UNIMOL. The circuit is constructed similarly to the BIMOL DSD circuit, except for using an auxiliary Helper strand instead of the absent second reactant (as suggested by [17]). Figure S1 in Supplementary Information depicts the four DSD reactions in the UNIMOL DSD circuit.

Applying the law of mass action kinetics to the UNIMOL circuit, we obtain the following approximate rate equations for the two product strands:

$$\frac{d[X_n]}{dt} = \frac{k_2 k_{1f}}{10 k_{1b} + k_2} [U_i] \quad \frac{dX}{dt} = k[U] \quad (2.3)$$

$$\frac{d[Y_o]}{dt} = \frac{k_2 k_{1f}}{10 k_{1b} + k_2} [U_i] \quad \frac{dY}{dt} = k[U] \quad (2.4)$$

These equations match closely with the ideal unimolecular rate law ($\frac{dX}{dt} = \frac{dY}{dt} = k[U]$ for $U \xrightarrow{k} X + Y$) with rate constant $k = \frac{k_2 k_{1f}}{10 k_{1b} + k_2}$. A detailed derivation of the DSD rate law is provided in Supplementary Information (A.3).

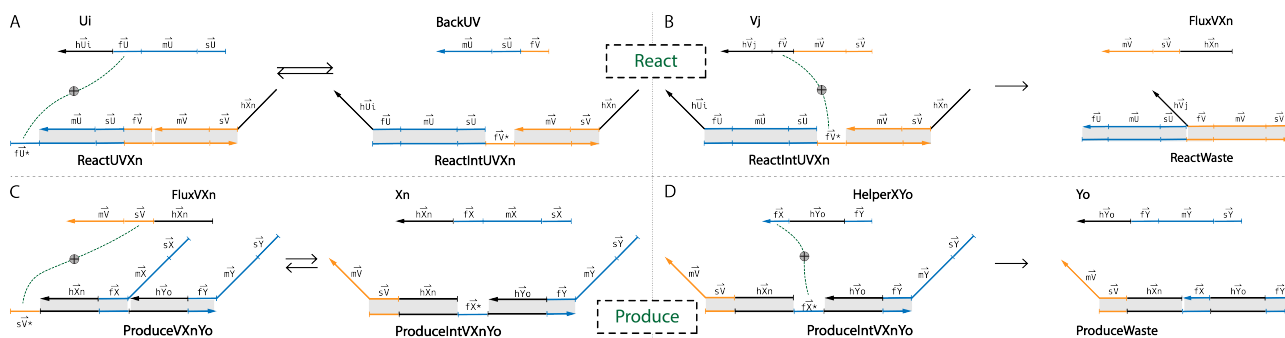


Figure 1: Domain-level reactions of the bimolecular autocatalytic amplifier ($U_i + V_j \rightarrow X_n + Y_o$) implemented in the “React-Produce” framework. **A** and **B** are the reactions in the *react* stage and **C** and **D** are the reactions in the *produce* stage. (A) The first reaction of the “React” stage. The signal strand C_j reacts reversibly with the “React” complex $\text{React}_{\text{CBC}_j}$ to produce the intermediate complex $\text{ReactInt}_{\text{CBC}_j}$. (B) The second reaction is the “React” stage. The second input Br reacts with $\text{ReactInt}_{\text{CBC}_j}$ to release the ‘flux’ strand $\text{Flux}_{\text{BC}_j}$. (C) The first reaction of the “Produce” stage. $\text{Flux}_{\text{BC}_j}$ reacts with the “Produce” complex $\text{Produce}_{\text{BC}_j\text{C}_k}$ and releases C_j and the intermediate complex $\text{ProduceInt}_{\text{BC}_j\text{C}_k}$. (D) Second reaction of the “Produce” stage. The ‘helper’ strand $\text{Helper}_{\text{CCK}}$ reacts with $\text{ProduceInt}_{\text{BC}_j\text{C}_k}$ to release the second signal strand C_k .

(b) Mechanistic modeling of leak

Two types of leaks are commonly observed in strand displacement systems: (i) initial leak and (ii) gradual leak. *Initial leak* refers to the signal activity observed in the circuit immediately after the addition of the gate complexes (with signal strands yet to be added). This leak is primarily attributed to weak annealing of the gate complexes and happens only at the beginning of the experiment. Common solutions to curtail this leak include using ultra-pure strands or “thresholding” complexes that absorb the leaked signal. *Gradual leak*, on the other hand, is systemic and persists for the entirety of the experiment. Reynaldo *et al.* [30] analyzed two possible gradual leak pathways: (i) toeless strand displacement (toeless-SD) and (ii) dissociation. The *toeless-SD* is primarily due to fraying at the helix edges, which opens up a short toehold that attracts an invader, resulting in the undesired displacement of the incumbent strand. In contrast, the *dissociation* pathway refers to the slow but independent detachment of an incumbent strand from the gate complex. Among the two, we only utilize the *toeless-SD* pathway in our leak modeling as it was experimentally shown to be several orders of magnitude faster than the dissociation pathway [30].

Srinivas *et al.* [17] identified three *toeless-SD* leak pathways in the React-Produce framework of the BIMOL DSD circuit: (i) React-second input leak, (ii) React-Produce leak, and (iii) Produce-Helper leak. *React-second input* leak (Figure 2A) occurs due to fraying at the nick of the React complex ($\text{React}_{\text{UVX}_n}$), opening it to attack from the second input strand (V_j via the fV domain), resulting in the displacement of the Flux strand (V_j). *React-Produce* leak (Figure 2B) occurs due to fraying at the “blunt” end of the React complex ($\text{React}_{\text{UVX}_n}$), opening it to attack from the open toehold (sV^*) of the Produce complex ($\text{Produce}_{\text{VX}_n\text{Y}_o}$), again displacing the $\text{Flux}_{\text{VX}_n}$ strand. *Produce-Helper* leak (Figure 2C) occurs due to fraying at the nick of the Produce ($\text{Produce}_{\text{VX}_n\text{Y}_o}$) complex, opening it to

an attack from the Helper strand (HelperY_o), resulting in the displacement of the second product strand (Y_o). Following the original work [17], we only include the Produce-Helper leak in our leak modeling, as its magnitude is significantly higher than the other two leaks due to the presence of two high-concentration fuel species. While the React-Produce leak also involves two fuel species, its magnitude is lower in the presence of clamps at the helix ends, which was shown to diminish the rate of ‘blunt-end’ strand displacement significantly [26].

While choosing the rate constant of the Produce-Helper leak mechanism, we considered the experimentally observed values of 10 /M/s by Srinivas *et al.* [17] and 40 /M/s by Reynaldo *et al.* [30] for the *toeless-SD* pathway with a 16 nt probe at 32.5 °C. We set our Produce-Helper rate constant to be 20 /M/s.

(c) Shadow cancellation

(i) Our automated design for the shadow circuit

There are three primary requirements for the effective functioning of shadow cancellation [16]: (i) the primary and shadow circuits should have similar rate profiles; (ii) both circuits should have similar leak characteristics, and (iii) the cancellation reactions should be faster than the fastest strand displacement reaction in the circuit. Song *et al.* [16] fulfilled these requirements by carefully selecting from a database [13] of strand sequences with known kinetic properties. However, the ad-hoc nature of this process imposes a significant bottleneck for adapting shadow cancellation to arbitrary strand displacement circuits. Here, we present a more general and automated strategy to construct the shadow circuit, given only the strand sequences and the positional arrangement of domains in the primary circuit.

(1) The shadow circuit is designed to be identical in structure to the primary circuit, i.e., there is a one-to-one

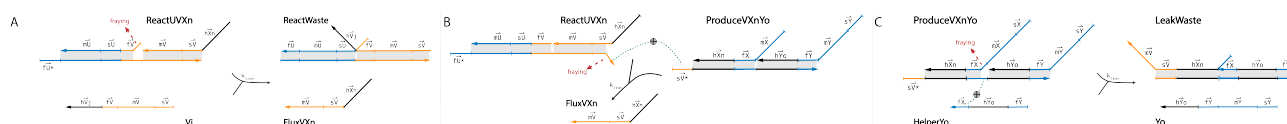


Figure 2: (A) The *React-second input* leak reaction in the bimolecular autocatalytic amplifier circuit. A short toehold is generated due to fraying at the nick in the $\text{React}_{\text{CBCj}}$ complex, opening it to attack from the second input Br , which displaces the flux strand Flux_{BCj} . (B) The *Produce-Helper* leak reaction in the bimolecular autocatalytic amplifier circuit. A short toehold is generated due to fraying at the nick of the $\text{Produce}_{\text{BCjCk}}$ complex, opening it to attack from the helper strand $\text{Helper}_{\text{Ck}}$, which displaces Ck . (C) The *React-Produce* leak reaction.

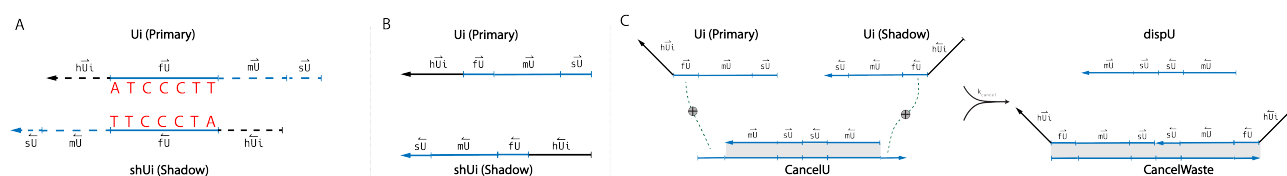


Figure 3: (a) The sequence of each domain in the shadow circuit is constructed by mirroring the sequence of the corresponding domain in the primary circuit. Additionally, the domains are designed to be non-palindromic to ensure non-orthogonality among the two circuits. (b) Since the cancellation process requires the complementary strands of the primary and shadow circuits to attack the cancel complex from opposite ends, a shadow strand is constructed by reversing the domain order of the corresponding primary circuit strand. (c) Cancellation mechanism where the strands Cj (from the primary circuit) and sCj (from shadow circuit) cooperatively hybridize to the CancelC complex and cancel each other [31]. In no particular order, Cj and sCj hybridize to toeholds at the opposite ends of the CancelC complex, displacing the dispC strand.

correspondence between the domains, signal strands, and fuel species of the two circuits.

(2) The DNA sequence for each shadow circuit domain is obtained by simply reversing the domain sequence of its primary circuit counterpart. Figure 3A depicts this transformation between the primary circuit domain (fU) and the shadow circuit domain fU . Note that $\rightarrow$ on top of the domain name indicates a primary circuit domain, whereas $\leftarrow$ indicates the corresponding shadow circuit domain.

(3) The shadow circuit species are constructed by applying the following transformation to the corresponding primary circuit species. First, the individual domains of the substrate are substituted by their shadow circuit counterparts. Then, the domain order of each constituent strand of the substrate is reversed. Figure 3B illustrates this transformation, where the shadow circuit strand shUi (bottom) is constructed by replacing the domains and reversing their order in the primary circuit strand Ui (top). Note that the reversal of the domain order is essential for the cancellation mechanism discussed in later sections.

(4) Finally, the circuits are modified if necessary to ensure no cross-talk between the primary and shadow circuits, except through the cancellation mechanism.

(ii) Cancellation mechanism

The cancellation mechanism in shadow cancellation is enabled through a class of DNA gates known as ‘cooperative hybridization complexes’ [31], referred to as the *cancellation complexes*. These complexes enable

‘annihilative’ reactions between two signal strands by sequestering them conjunctively (only when both strands are simultaneously present). Figure 3C depicts a cancellation reaction between the primary circuit strand Ui and the shadow circuit strand shUi facilitated by the cancellation complex CancelU . The cancellation complex contains toeholds on either end, one each for the strands to be sequestered. In the associated cancellation reaction, Ui and shUi latch onto their respective toeholds, eventually displacing the top strand (dispU). Note that the strand displacement will not go through without one of the two strands. In our domain-level simulations, we utilize the experimental rate constants reported by Zhang *et al.* [31].

There are, however, side effects to using the cancellation mechanism. While a strand is not ‘canceled’ in the absence of its counterpart, it is temporarily sequestered from the circuit. Since the primary circuit will have a higher signal activity than the shadow circuit (inputs of the shadow circuit are zeroed out), the primary circuit strands will be temporarily sequestered in the cancellation complexes, suppressing the primary circuit activity. This issue is known as toehold occlusion. In the later sections, we will discuss how it affects the circuit design and dynamics in greater detail.

3. Leak-resilient nucleic acid dynamical systems

Dynamical behavior is a salient feature of many biological processes. Examples include the chemical oscillators in

circadian rhythms [32, 33], autocatalytic reaction networks in metabolism [34], consensus protocols in bacterial *quorum sensing* [35, 36], and feedback control systems that regulate molecular concentrations inside a cell [37]. Robust and durable implementations of such sophisticated processes *in vitro* are expected to lead to novel diagnostic [2], biomedical [38, 39], and biosensing applications [40, 41].

In this section, we present the domain-level simulation results of our target dynamical systems using Peppercorn [42], a domain-level strand displacement reaction simulator. Peppercorn takes as input the domain lengths and their positional arrangement in the substrates and outputs a set of all possible strand displacement reactions along with their estimated rate constants. The circuit dynamics are then estimated by solving the ordinary differential equation (ODE) system resulting from the combined mass action kinetics of the corresponding DSD circuit. Our target dynamical systems are as follows: (a) a Unimolecular autocatalytic amplifier, (b) a Bimolecular autocatalytic amplifier, (b) a Rock-Paper-Scissors oscillator, (c) a Molecular consensus protocol, and (d) a Proportional Integral (PI) feedback controller. We simulate each dynamical system in four different settings that incrementally move toward a shadow cancellation-enabled circuit (we refer to them collectively as *simulation settings*): (i) Vanilla, (ii) Leaky, (iii) VanillaOccluded, and (iv) LeakyCancel. The *Vanilla* setting describes the dynamics of a ‘pure’ DSD circuit, i.e., a circuit without leaks. This circuit refers to the reaction set generated by Peppercorn. The *Leaky* setting describes the dynamics of a leak-affected DSD circuit. However, since Peppercorn cannot currently model leak pathways, this circuit is constructed by manually adding the leak reactions (specifically, the Produce-Helper leak reactions; see (b)) to the Vanilla circuit. *LeakyCancel* setting describes the dynamics of a shadow cancellation-enabled DSD circuit. This circuit combines the Leaky primary circuit, the Leaky shadow circuit with certain inputs zeroed out, and the cancellation mechanism. However, the presence of the cancellation complexes introduces undesired side effects into the circuit as these complexes, by nature, temporarily sequester the signal strands in the absence of their counterpart strands, suppressing the circuit activity (intuitively, the primary circuit’s activity as it dominates that of the shadow circuit). This phenomenon is referred to as *toehold occlusion*. Due to this, the LeakyCancel setting can no longer be fairly compared with the Vanilla setting. To overcome this limitation, we devised the *VanillaOccluded* setting, which models the dynamics of a DSD circuit in the presence of toehold occlusion. The associated DSD circuit is constructed by adding the cancellation complexes to the Vanilla DSD circuit. Since the shadow circuit is absent, cancellation reactions are not possible, and the only interactions between the primary circuit and the cancellation complexes are the reversible ‘occlusion’ reactions. We provide a more rigorous analysis of the effects of toehold occlusion on the circuit design and its dynamics in the later sections.

(a) Leak-resilient unimolecular autocatalytic amplifier

The unimolecular autocatalytic amplifier (UNIAMP) exponentially amplifies a signal until all the available resources are exhausted. Its formal CRN consists of a single unimolecular reaction $C \rightarrow C + C$, where C represents the amplified signal. We rewrite the reaction using DNA signal strands: $C_j \rightarrow C_j + C_k$. Note that the signal strands C_j and C_k are instances of the same formal species C , differing only in their history domains. Since the amplifier CRN consists of a single unimolecular reaction, its DSD circuit is constructed by modifying the UNIMOL DSD circuit such that C_j is consumed during the React phase, whereas C_j and C_k are released (in that order) during the Produce phase.

Figure 4 depicts the dynamics of the UNIAMP DSD circuit in the four simulation settings. Figure 4A represents the Vanilla dynamics. Here, the signal curve of C (representing the combined concentration of C_j and C_k) traces an exponential S-shaped curve with an initial exponential region, followed by an inflection region and culminating in a saturation region, where it reaches a steady state concentration close to the initial concentration of the fuel species. The inflection region begins when the signal concentration is comparable to that of the fuel species, and the saturation region is reached once the fuel species are depleted. Figure 4B juxtaposes the Leaky (solid) and Vanilla (dashdot) dynamics. Here, the excess C_k strands generated due to the Produce-Helper leak amplify exponentially, leading to steeper amplification and faster saturation. Figure 4C juxtaposes the VanillaOccluded (solid) and Vanilla (dashdot) dynamics. We notice that the former signal curve is suppressed compared to the latter due to toehold occlusion. Finally, Figure 4D juxtaposes the LeakyCancel (solid) and VanillaOccluded dynamics (dashdot). We observe that the LeakyCancel dynamics closely follow the VanillaOccluded dynamics, demonstrating the efficacy of shadow cancellation in restoring a leak-affected UNIAMP DSD circuit (Figure 4B) to normality. Further, the shadow signal (dotted) is restricted to a very low concentration, indicating that the leak activity in the circuit is effectively subdued.

(b) Leak-resilient bimolecular autocatalytic amplifier

The bimolecular autocatalytic amplifier (BIAMP) given by a reaction of the form $C + B \xrightarrow{k} C + C$ exponentially amplifies a signal C at the expense of a ‘fuel’ signal B . The effective CRN of the DSD circuit in terms of the signal strands is given by $C_j + Br \xrightarrow{k} C_j + C_k$, where the collective concentration of the strands C_j and C_k represents C and concentration of the strand Br represents B . Consequently, the BIAMP DSD circuit is constructed by modifying the BIMOL circuit such that C_j and Br are consumed during the React phase (similar to U_i and V_j), whereas C_j and C_k are released during the Produce phase (similar to X_n and Y_o).

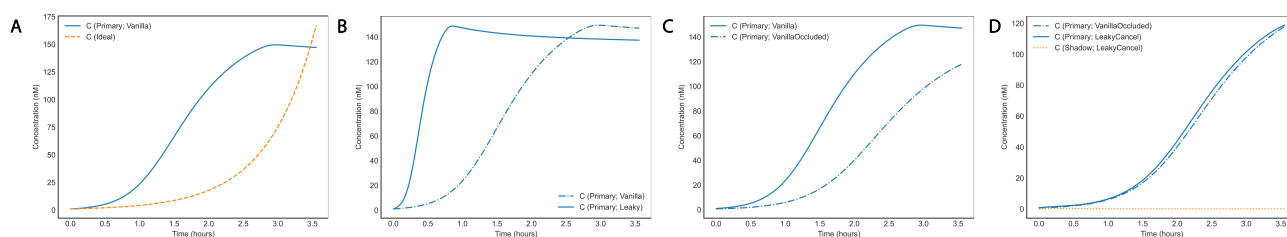


Figure 4: “React-Produce” simulations of the UNIAMP circuit in three different settings: (a) UNIAMP in the absence of leaks. (b) UNIAMP in the presence of leaks but no shadow cancellation. (c) UNIAMP in the presence of leaks and with shadow cancellation.

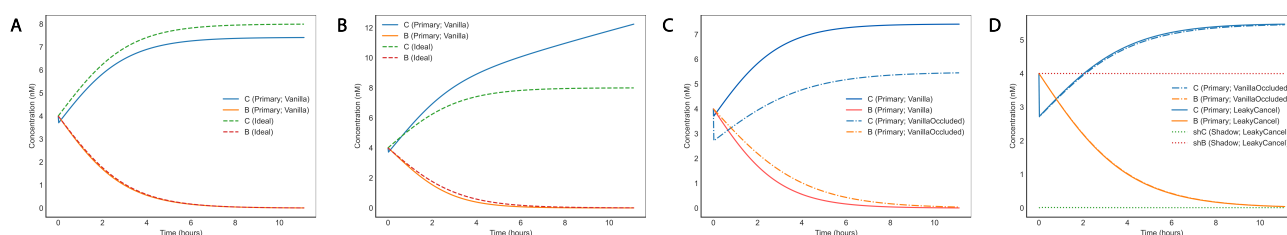


Figure 5: “React-Produce” simulations of the BIAMP circuit in three different settings: (a) BIAMP in the absence of leaks. (b) BIAMP in the presence of leaks but no shadow cancellation. (c) BIAMP in the presence of leaks and with shadow cancellation.

Figure 5 depicts the Peppercorn simulations of the BIAMP DSD circuit. In its Vanilla setting (Figure 5A), the signal C grows exponentially, reaching a steady state concentration equal to the sum of initial concentrations of C and B, whereas B depletes to zero. In the Leaky setting (Figure 5B), the Produce-Helper leak results in the excessive release of the Ck strand, causing the amplifying signal to grow beyond the expected maximum concentration, even after the exhaustion of the fuel signal. In the VanillaOccluded setting (Figure 5C), the recorded signal activity is lower than in the Vanilla setting due to the sequestering of the amplifying signal strands by the cancellation complexes. In the LeakyCancel setting (Figure 5D), the activation of the cancellation reactions subdues the leak activity in the circuit and restores the circuit dynamics to normalcy (similar to dynamics in the VanillaOccluded setting).

(c) A leak-resilient Rock-Paper-Scissors oscillator

The Rock-Paper-Scissors (RPS) oscillator is an autocatalytic system in which three competing species engage in a cyclic dominance pattern, with the dominating species alternating periodically. Figure 6C depicts the dynamics of an ideal RPS oscillator, where the three signals trace sinusoidal-like paths. The cyclic precedence among the three species is reminiscent of the precedence rules of the zero-sum game ‘rock, paper, scissors’ (Figure 6A; arrows indicate the precedence rules; textitrock beats *scissors*, *scissors* beats *paper*, and *paper* beats *rock*). The formal CRN for this system

is a combination of three BIAMP reactions, as shown in Figure 6B. The precedence rules are indirectly encoded into these reactions by setting the dominant species as the amplifying signal and the other as the fuel signal. For example, in constituent BIAMP reaction, $C + B \rightarrow C + C$ represents that the amplifying signal (C) precedes the fuel signal (B). Oscillations of this kind are observed in the population dynamics of many coexisting biological systems [43] and predator-prey ecological models [44].

The DSD circuit for the RPS oscillator is constructed by linearly combining the DSD circuits of the three constituent BIAMP reactions. To our knowledge, the most recent DNA implementation of an RPS oscillator is by Srinivas *et al.* [17] and is designed using the React-Produce framework. However, due to excessive leaks, their circuit sustained oscillations for only a few cycles (This is not a criticism of the work as their goal was to demonstrate oscillatory behavior in enzyme-free nucleic acid circuits). Following this work, we adapt their circuit design, rate constants, fuel concentrations, and leak rates in our domain-level simulations. To ensure the validity of our construction, we compared the dynamics of our domain-level simulation with their mechanistic model in Figure S3. While both plots trace similar curves, they differ in two aspects: (i) We ignore the initial leak in our simulations as it is outside the scope of this work, whereas they model it based on experimentally observed values, and (ii) They explicitly model toehold occlusion by the fuel gate complexes, whereas it is implicitly handled in our simulations by Peppercorn. We provide a detailed account of our design choices in Supplementary Information (A.2).

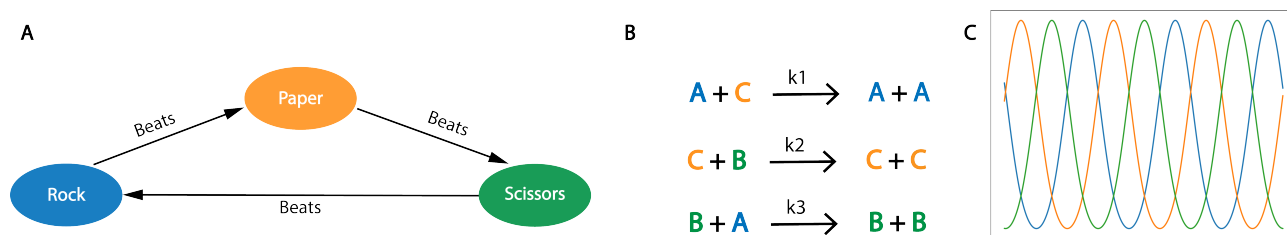


Figure 6: (a) Schematic of the Rock-Paper-scissors game. (b) CRN of the Rock-Paper-Scissors oscillator. The reactions of the CRN mimic the rules of the game illustrated through the autocatalytic consumption of the losing species by the winning species. (c) Dynamics of the ideal RPS oscillator depicting the cyclic dominance pattern.

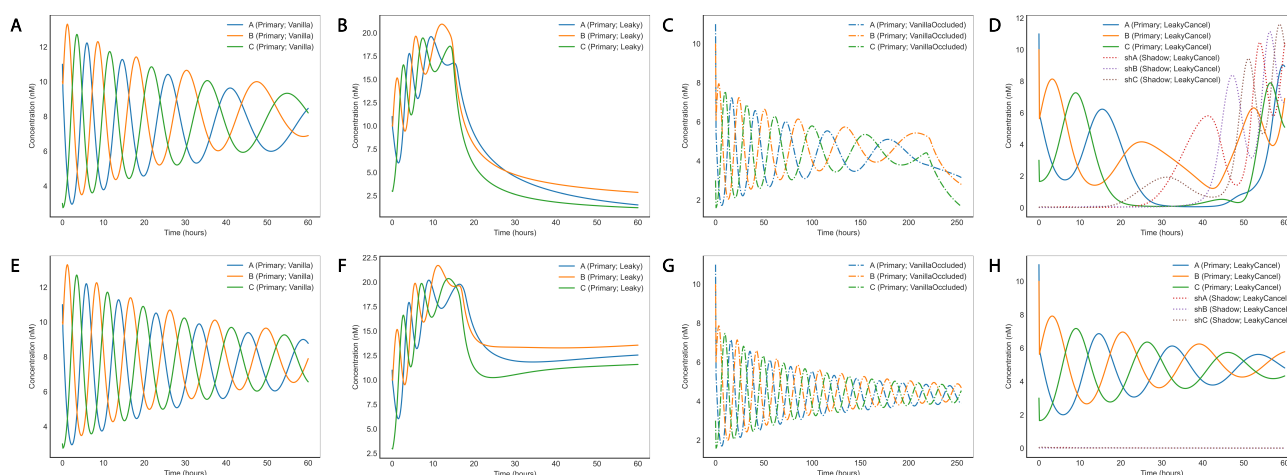


Figure 7: RPS Peppercorn simulations

Figure 7 depicts the behavior of the RPS DSD circuit in different simulation settings. Figure 7A shows the Vanilla oscillator dynamics, where the circuit sustains oscillations (14 cycles in 60 hours). The amplitude of the oscillations decreases with time due to the consumption of the fuel species. Figure 7B shows the Leaky oscillator dynamics. Here, the release of the excess signal strands due to the Produce-Helper leak and their subsequent amplification due to the autocatalytic nature of the circuit causes the circuit to dissipate rapidly (6 cycles and oscillations sustained for only 10 hours). Figure 7C juxtaposes the VanillaOccluded (dashdot) and Vanilla (solid) dynamics. We observe that the circuit activity is suppressed in the former setting compared to the latter, as evident from the lower amplitude and the initial dip in the signal concentrations due to toehold occlusion. Figure 7D shows the LeakyCancel dynamics. However, the primary circuit's activity (solid) collapses abruptly, followed by an increased shadow circuit's activity, while the oscillations in the primary circuit are subdued. We hypothesize that this counterintuitive behavior is due to an interplay of the following factors: (a) the unequal consumption of the fuel species in the primary and shadow circuits—since the primary circuit has higher activity (by design) than the shadow circuit (maintained close to zero), the fuel species in the former are consumed faster, and (b) the depletion of the cancellation complexes. As the

concentration of cancellation complexes drops below a threshold, the cancellation process can no longer nullify the leak activity in the circuits. At this juncture, the shadow circuit has a higher fuel species concentration than the primary circuit and becomes the dominant circuit. As a result, the activity in the primary circuit is actively suppressed, causing the oscillations to die out. Excess cancellation complexes are not a viable solution, as they are strong sinks of signal strands and can effectively nullify the circuit activity through toehold occlusion. Using large quantities of fuel species (to counteract the unequal consumption) is not viable either, as the leaks in such a setting can become unwieldy for shadow cancellation (leak rates become comparable to the strand displacement rate).

We propose *buffering* the fuel species and the cancellation complexes to tackle the above two issues simultaneously. Lakin *et al.* [23] demonstrated an adaptive buffering system, where the gate complexes are in excess in an inactive form and can be activated dynamically with the help of 'activating' strands. In this work, however, we adopt a simple buffering scheme, where the fuel species and the cancellation complexes are introduced into the circuit at a constant rate chosen by trial and error as if permeating through a one-way semi-permeable membrane.

Figures 7E-H represent the dynamics of the simulation settings in the 'buffered' RPS oscillator setup. Figure 7E

shows the buffered Vanilla dynamics. This is similar to its unbuffered counterpart (Figure 7A), except that the oscillations cease in the latter once the fuel species are consumed. Figure 7F shows the Leaky dynamics and illustrates that buffering cannot mitigate leaks, as the leaked strands can still undergo exponential amplification. Figure 7G represents the VanillaOccluded dynamics. Similar to the Vanilla setting, this differs from the unbuffered case in that the latter ceases after all the fuel species are consumed. Note that the time range is deliberately extended to show the difference between the buffered and unbuffered setup. Figure 7H depicts the buffered LeakyCancel dynamics, where the oscillations are restored to normality by buffering the fuel species and the cancellation complexes. Shadow cancellation, in conjunction with buffering, helps restore the circuit dynamics in the following ways: (a) By preventing the depletion of the fuel species and the cancellation complexes, it ensures that the activity in the shadow circuit doesn't overpower the primary circuit; and (b) The circuit wouldn't require the presence of high concentrations of the fuel species or cancellation complexes, which could lead to excessive leaks or excessive signal suppression respectively.

(d) A leak-resilient Two-molecule consensus system

Consensus protocols enable the agents in a multi-agent environment to agree on a decision, an observation, or a data value. Our target dynamical system is a chemical consensus protocol, where the species with a higher initial concentration (majority species; the rest are referred to as minority species) is chosen as the leader. An example protocol for achieving consensus between two different molecular species is shown in Figure 8A, where the leader (*green*) is elected by transforming the minority species (*red*) into majority species (*green*), until only majority species remain. Figure 8B depicts the CRN for this protocol, consisting of two BIAMP reactions and one bimolecular transduction reaction. Here, A and B are the 'competing' species, with Y acting as the 'moderator' species. At the steady state, the majority species goes to a final concentration equal to the sum of the initial concentrations of all three species, whereas the minority and moderator species go to zero (Figure 8C). Since each constituent reaction is bimolecular, the overall DSD for this consensus system is constructed by combining the BIMOL DSD circuits of the individual reactions. Developing biochemical consensus protocols robust to leaks could enable applications that require collaborative decision-making at the molecular scale [45–48].

The first DSD implementation of the two-molecule consensus algorithm discussed above was by Chen *et al.* [21] that used the 'Join-Fork' framework [29]. In this work, we reimplement the circuit using the React-Produce framework [17] for consistency and comparison. Figure 9 illustrates the behavior of the consensus circuit under the four simulation settings. In all these settings, the initial concentrations of

the species A, B, and Y are set to 3 nM, 8 nM, and 5 nM, respectively, with B being the majority species. Figure 9A represents the Vanilla setting, where at steady state, only B prevails with a final concentration of 20 nM (3 + 8 + 5). In the Leaky setting (Figure 9B), the concentration of the majority species grows beyond the expected maximum concentration (20 nM) until the exhaustion of the fuel species. Figure 9C represents the shadow cancellation-enabled consensus circuit, where the majority concentration is shown to be stabilized. Note that the final concentration of most species is less than 20 nM due to toehold occlusion of the signal species by cancellation complexes.

(e) A leak-resilient PI feedback controller

A Proportional Integral (PI) feedback controller is a catalytic I/O system that retroactively corrects the error in the output signal of a noise-affected system. Our target PI Controller (Figure 10a) is adapted from Yordanov *et al.* [49] (which in turn was adapted from the result by Oishi and Klavins [18] that a PI Controller can be constructed solely from catalysis, annihilation, and degradation reactions). The dynamical system contains two interacting components: *Controller* (C in Figure 10a) and *Plant* (P in Figure 10a). The Plant is inherently faulty and produces an output signal (Y) instead of its intended 'reference' state (*r*). The Controller then retroactively generates 'gradient-like' corrections to the Plant's input (*v*) so that the subsequent calculations of Y converge to *r*. Feedback control is an essential mechanism in the cellular feedback machinery that regulates the molecular concentrations in a cell [50]. For this reason, synthetic biochemical controllers could lead to novel biomedical, biosensing, and therapeutic applications [4, 21, 49].

Figure 10b represents the schematic of the combined CRN of the Controller and the Plant adapted from [49]. Each signal species in the CRN (*E*, *X*, *V*, *Y*, *R*, *L*) models their signal counterpart (*e*, *x*, *v*, *y*, *r*, *load*) in the Controller-Plant schematic (Figure 10a). Additionally, the $\pm$ in the superscripts of the formal signal species indicate the *dual-rail* notation, where a signal's value is modeled as the concentration difference between two complementary formal species. The Controller CRN is constructed solely using Catalysis, Annihilation, and Degradation reactions [18], whereas the Plant CRN consists of Produce, Consume, and Load reactions. We will now briefly discuss the construction of DSD circuits for each reaction and utilize them as components to construct the Controller-Plant DSD circuit.

Note that Oishi and Klavins [18] constructed their DSD circuit using the so-called *four-domain* framework introduced by [28]. Although both frameworks are conceptually similar, we reimplement the controller in the React-Produce framework to maintain consistency and exploit its simple and well-studied leak pathways [17]. The DSD implementation of the PI Controller system differs qualitatively from the rest of the dynamical systems in several aspects: (a) The complexity in the system arises from multiple interacting components rather than from an

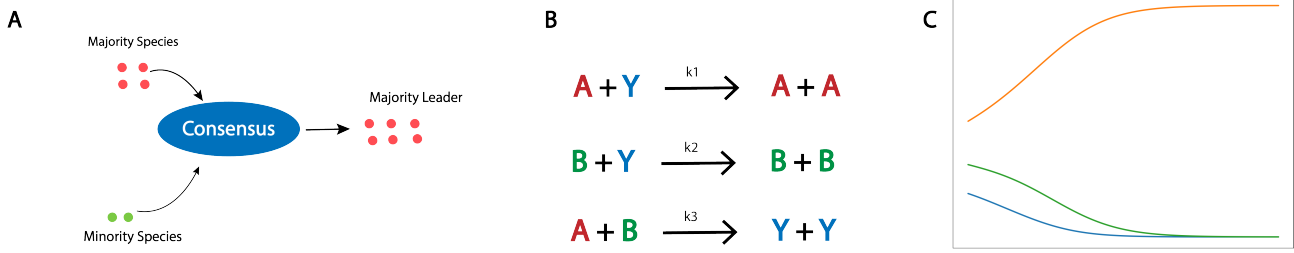


Figure 8: (a) Schematic of the two-molecule consensus system where the majority leader is elected by the conversion of the minority species (green) into the majority species (red) (b) CRN of the two-molecule consensus system. The CRN is made up of three autocatalytic reactions. A and B are the competing species, with Y as the supporting species. (c) Dynamics of an ideal consensus system. The initial concentrations of A, B, and Y are set to 7 nM, 8 nM, and 5 nM, respectively. Since B has the initial majority, at a steady state, only B prevails with a concentration equal to the sum of initial concentrations of all the species, i.e., 20 nM.

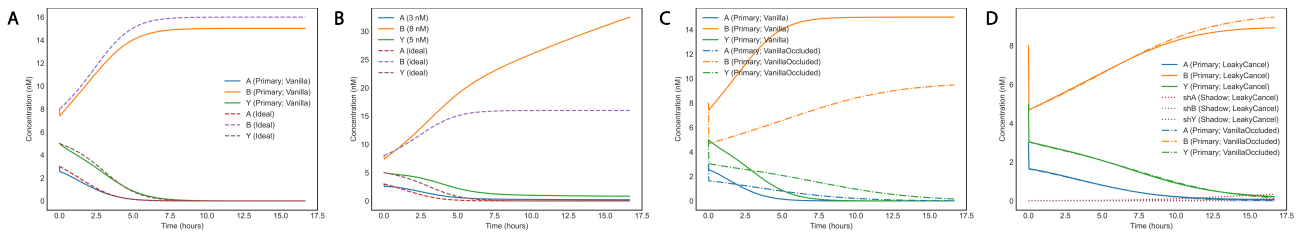


Figure 9: Consensus Results

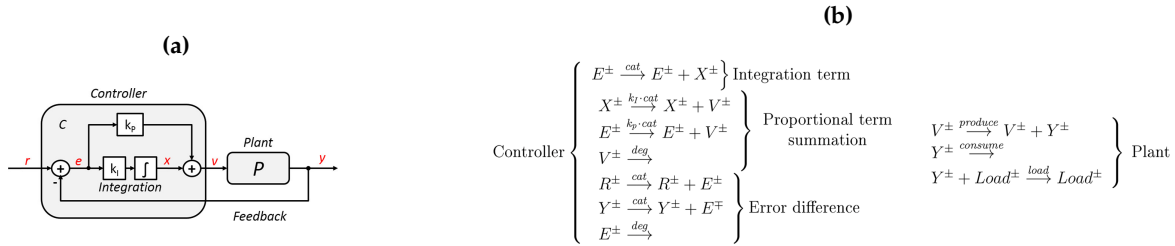


Figure 10: (a) Schematic of a Feedback control system (adopted from [19]) composed of a physical plant P and a controller C . e is the error signal given by the difference between the reference signal r and the plant output Y . The controller automatically adjusts the plant input v to minimize the error according to the tuning parameters k_I and k_P . (b) CRN of the PI feedback controller adopted from Yordanov *et al.* [19], composed using catalysis, annihilation, and degradation reactions. The $\pm$ indicates the species being in the dual-rail format.

autocatalytic behavior; and (b) The system imposes semi-hard constraints on the effective reaction rate constants, which require finer control over the circuit parameters viz. strand displacement rate constants, fuel species concentrations. Now, we'll describe the construction of the DSD circuits for each of the five reaction classes required to create the Controller and Plant and then discuss the construction of the PI Controller from these submodules. We will interchange the abstract signal species and its corresponding strand representation to simplify the description, disambiguating the two whenever necessary. *Catalysis* reactions of the PI Controller are unimolecular reactions of the form $E \xrightarrow{k_{cat}} E + X$. The DSD circuit for these reactions is constructed by modifying the UNIMOL

circuit such that E is consumed during the React phase and E and X are released during the Produce phase. Figure 11A juxtaposes the dynamics of an ideal catalysis reaction (dashed) with its React-Produce implementation (solid). *Degradation* reactions are unimolecular reactions of the form $V \xrightarrow{k_{deg}} \Phi$. Their DSD circuit is implemented by modifying the UNIMOL circuit to consume V during the React phase. We forgo the Produce phase for this reaction by considering the Flux strand released at the end of the React phase to be the Φ species. Figure 11B juxtaposes the dynamics of an ideal degradation reaction (dashed) with its React-Produce implementation (solid). *Annihilation* reactions are bimolecular reactions of the form $X^+ + X^- \xrightarrow{k_{ann}} \Phi$. These reactions represent the

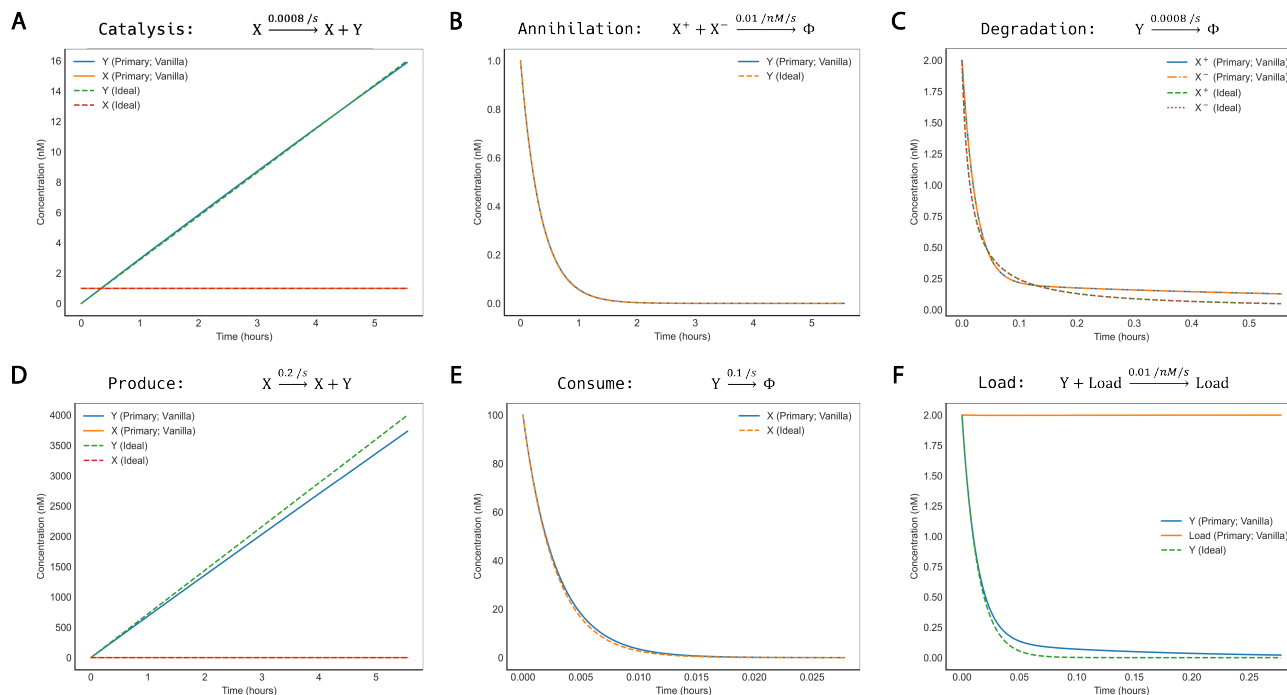


Figure 11: In all the figures, *dashes* represent the ideal reaction dynamics and *solid* lines represent the DSD circuit dynamics.

(a) Catalytic amplifier circuit $E_p \xrightarrow{k_{cat}} E_p + X_p$ implemented by modifying the BIAMP circuit. (b) Degradation reaction $V_p \xrightarrow{k_{deg}} \Phi$ implemented by modifying the UNIAMP circuit. (c) Annihilation reaction $X_p + X_m \xrightarrow{k_{ann}} \Phi$ implemented using two BIAMP circuits ($X_p + X_m \xrightarrow{k_{ann}/2} \Phi$ and $X_m + X_p \xrightarrow{k_{ann}/2} \Phi$) to ensure equal consumption of the two species.

disjunctive canceling reactions among the positive and negative parity dual-rail species of all the signals (e.g., X^+ and X^- are the two species for signal X). Two reactants are consumed in this reaction, and no products are released. Therefore, the DSD circuit for this reaction is constructed using the BIMOL circuit's React phase. The reactants X^+ and X^- are consumed during the React phase, and the Flux strand released at the end of the phase is considered the waste species (Φ). However, in the BIMOL circuit, the second reactant is consumed slower than the first. To equalize their consumption rates, annihilation is implemented using DSD circuits for the following two reactions: $X_p + X_m \xrightarrow{k_{ann}/2} \Phi$ and $X_m + X_p \xrightarrow{k_{ann}/2} \Phi$, where the reactant order is flipped in the second reaction. Figure 11C juxtaposes the dynamics of an ideal annihilation reaction (dashed) with the React-Produce implementation (solid).

Produce reactions of the Plant are unimolecular catalytic reactions of the form $V \xrightarrow{produce} V + Y$. However, their rate constant ($produce = 0.2 /s$) is out of range for a strand displacement implementation for extended periods. To preserve the reaction dynamics, we refurbish (buffer) its fuel species into the reaction mix at a constant rate, similar to that of the RPS oscillator. Figure 11D juxtaposes the ideal dynamics of the Produce reaction with its Buffered DSD implementation.

Consume reactions of the Plant are unimolecular reactions of the form $Y \xrightarrow{consume} \Phi$. While the reaction looks similar

to the Degradation reaction, the rate constant (*consume*) is out of range for a strand displacement implementation. Therefore, we implement buffering, similar to the Produce reaction, to maintain the reaction dynamics for longer periods. Figure 11E juxtaposes the ideal dynamics of a Consume reaction with its React-Produce implementation. *Load* reactions of the Plant are bimolecular reactions of the form $Y + L \xrightarrow{load} L$, where the second reactant (L) catalytically consumes the first (Y). Unlike the previous two reactions, Load reactions do not require buffering. Their DSD circuits are constructed by modifying the BIMOL circuit such that the reactants Y and L are consumed during the React phase in that order, and L is released during the Produce phase (not to be confused with the Produce reaction of the controller). Figure 11F juxtaposes the dynamics of an ideal Load reaction (*dashed*) with its React-Produce implementation (*solid*).

The PI Controller combines multiple interacting submodules with independently diverse behaviors. Therefore, constructing a functional feedback control system requires careful tuning of the circuit parameters, i.e., rate constants of the strand displacement reactions and the associated fuel concentrations. In this work, we construct a simplified PI Control circuit with a constant reference state (R) and a constant Load (L). The Vanilla PI Control circuit is constructed by simply combining the DSD circuits of its CRN reactions. Figure 12A depicts the dynamics of this circuit, where the concentration of the output species

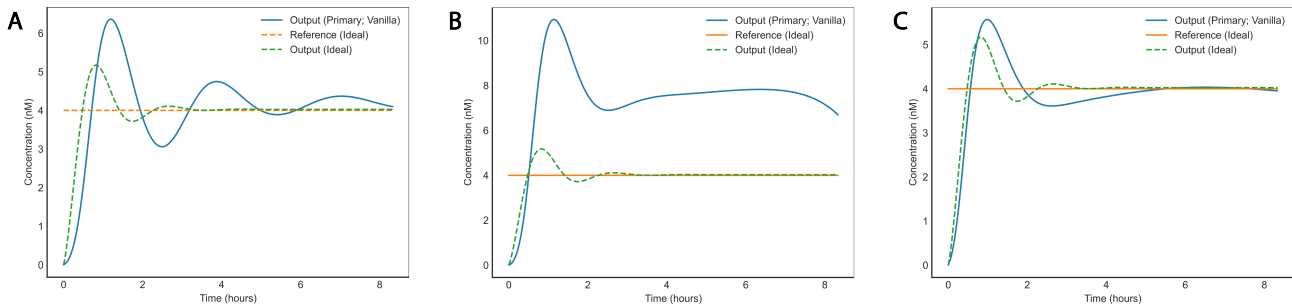


Figure 12: Each row contains the “React-Produce” simulations of the three different settings in this order from left to right – *Column 1* Pure DSD circuit without leaks. *Column 2* DSD circuit has leaks but no shadow cancellation. (iii) *Column 3* DSD circuit with leaks and shadow cancellations. Note that simulations of all the dynamical systems except the PI controller are domain-level simulations using Peppercorn. In contrast, those of the PI controller are ODE simulations using the *Catalyst.jl* Julia package (**a, b, c**) RPS oscillator; (**d, e, f**) Consensus protocol; (**g, h, i**) PI feedback controller

(Y) oscillates with damping amplitude around the reference state (R) before collapsing onto it. For the Leaky setting, our leak model consists of the leak reactions from the Controller submodules with the possibility of a Produce-Helper leak (mainly the Catalysis reactions). It is important to note that this simplifying assumption regarding the leak doesn't imply that some submodules are leak-free but was done to demonstrate the improvements provided by shadow cancellation without overly complicating the circuit. Figure 12B depicts the dynamics of the PI Control circuit in this setting. We observe that Y moves away from R (instead of oscillating around it), rendering the circuit unusable. Figure 12C represents the dynamics of the LeakyCancel setting of the PI Control circuit, where shadow cancellation restores the circuit to normalcy (similar to the Vanilla setting).

4. Discussion

So far, the proposed shadow cancellation methodology makes several idealizing assumptions. We state them again here for clarity:

- The primary and shadow circuits should run at a similar rate.
- The primary and shadow circuits should have similar leak characteristics.
- The cancellation reactions should be much faster than the fastest strand displacement reaction in the circuit.

In this section, we discuss several practical considerations regarding these assumptions and show that shadow cancellation is adequately robust to their deviations from ideality.

(a) Shadow circuit out-of-phase with Primary circuit

The first assumption stipulates that the primary and shadow circuits should possess a near-identical rate profile. A faster primary circuit leads to uncontrollable

amplification due to excess leak activity, whereas a slower primary circuit leads to signal suppression due to excessive cancellation [16]. This poses a significant bottleneck to the wider adoption of shadow cancellation as it is hard to reverse engineer DNA sequences that fit a predefined kinetic description. Even our heuristic method (described in §(c)) cannot reliably guarantee similar rates despite preserving the toehold sequences (which were shown to be crucial in determining the speed of a strand displacement reaction [9, 51]). We refer to this discrepancy as the shadow circuit being “out-of-phase” with the primary circuit.

Here, we examine the out-of-phase dynamics of two of our dynamical systems: (i) BIAMP, an end-point system, and (ii) RPS oscillator, a non-equilibrium system, chosen for their qualitative diversity. To simulate the out-of-phase behavior, we freeze the rate constants of the primary circuit and manually perturb the rate constants of the shadow circuit. Without loss of generality, we assume that the shadow circuit runs faster (has higher rate constants) than the primary circuit. We simulate the two dynamical systems in four different settings (referred to as the *perturbation settings*): (i) shadow = $1.5 \times$ primary, (ii) shadow = $2 \times$ primary, (iii) shadow = $5 \times$ primary, and (iv) shadow = $10 \times$ primary, where each setting indicates the relative speed of the shadow circuit to the primary circuit. For example, the ‘shadow = $5 \times$ primary’ setting represents that the rate constant of each strand displacement reaction in the shadow circuit is five times higher than the rate constant of the corresponding primary circuit reaction. Note that we do not apply this perturbation to the leak rate constants and assume that both circuits leak at an identical rate, as fraying, not branch migration, is the rate-limiting step of the pertinent leak pathways. Later in this section, we will discuss the effects of dissimilar rate constants between the two circuits. To keep the perturbed rate constants within the possible range for strand displacement, we set the primary circuit rate constants to be of the order 10^{-4} /nM/s and adjust the shadow circuit's rate constants according to the perturbation setting.

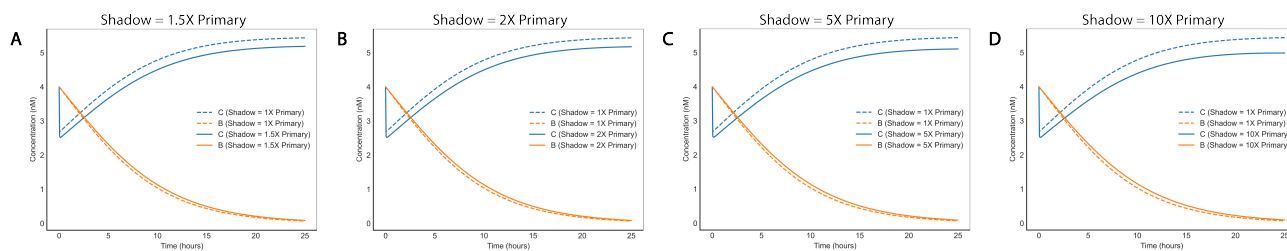


Figure 13: Dynamics of the shadow cancellation-enabled BIAMP circuit when the shadow circuit is out-of-phase with the primary circuit under different perturbation settings.

(i) Results of perturbation experiments

The approximate bimolecular rate law specified by (2.2) and (2.1) suggests that in an $p\times$ perturbation setting, the overall reaction rate (represented by the rate of formation of the products) increases by a factor of p . Figure 13 juxtaposes the dynamics of the unperturbed BIAMP circuit (dashed) with the perturbed BIAMP circuit in each of the four perturbation settings. We observe that the circuit dynamics are preserved even at $10\times$ perturbation (Figure 13D).

Parallely, we utilize the buffering-enabled RPS circuit for our perturbation experiments. In keeping with the original construction process, the perturbed RPS DSD circuit combines the individual perturbed BIAMP DSD circuits of its three constituent reactions. Figure 14 depicts the dynamics of the perturbed RPS circuit in each of the four perturbation settings. The oscillations are sustained even in the $10\times$ perturbation setting (Figure 14D).

The BIAMP and RPS circuits are shown to be adequately robust to the shadow circuit being ‘out-of-phase’ with the primary circuit, waiving the requirement that the two circuits run at a similar rate for shadow cancellation to be effective. To estimate the role of shadow cancellation in this observation, we measure the leak activity in the circuit both in the presence and absence of shadow cancellation. For this, we utilize the overall signal activity of the shadow circuit (with its inputs zeroed out) as a proxy for the leak activity in the primary circuit. Since the shadow circuit runs faster than the primary circuit, the overall activity in the shadow circuit acts as an upper bound on the primary circuit’s leak activity. Figures 15A and 15B juxtapose the aforementioned leak activity in the presence and absence of shadow cancellation in the BIAMP and RPS circuits. In both cases, the leak activity grows uncontrollably without shadow cancellation, whereas it is subdued close to zero in its presence. This is because the cancellation complexes block the leak activity from proliferating by quickly sequestering the leaked strands, thereby preventing them from interacting with their parent circuit. Figures 15C and 15D juxtapose the dynamics of the primary and shadow circuits in the presence of shadow cancellation for the $10\times$ perturbation setting. As the leak activity is closer to zero, the circuit remains stable.

(b) Shadow circuit leaks at a different rate

The ‘out-of-phase simulations in 4§(a) assume that the primary and shadow circuits leak at an identical rate. However, since fraying is the rate-limiting step of a toeless-SD pathway [30, 51], the difference in the nucleotide sequences at the helix edges cause the leak rates in the two circuits to differ. In this regard, we assume that the shadow circuit leaks at a higher rate than the primary circuit, as the leaks would grow uncontrollably otherwise. We simulate this difference in the leak rates as follows. The upper and lower bounds for the leak are set to 1×10^{-8} /nM/s and 8×10^{-8} /nM/s. These values are adapted from the experimental observations from Reynaldo *et al.* [30], who reported a rate of 79 /M/s for toeless-SD at 37.5°C. The leak (Produce-Helper leak) rate constant for the primary circuit ($k_{leak}^{primary}$) is frozen at 1×10^{-8} /nM/s and the shadow circuit’s leak rate constant (k_{leak}^{shadow}) in the increments of 1×10^{-8} . We refer to them collectively as the *leak perturbation* settings.

Figure 16 illustrates the effects of differing leak rates on the circuit dynamics in the BIAMP and RPS circuits. Note that since the shadow circuit leaks at a higher rate, some of the legitimate signals in the primary circuit will also be ‘canceled’ via the cancellation complexes in addition to its leak activity. To partially neutralize the excess cancellation, we scale down the concentrations of the shadow circuit proportionally to the increase in the leak rate. This proportionality is decided by trial and error.

Figure 16A shows the signal dynamics under different perturbation settings, and Figure 16B shows the corresponding leak activity. We observe that shadow cancellation, albeit with the concentration adjustment, can preserve the circuit dynamics, even when the shadow circuit leaks eight times faster than the primary circuit. This is corroborated by the similar leak activity (Figure 16B) and similar consumption of cancellation complexes (Figure 16C), which is stoichiometrically equivalent to the amount of ‘canceled’ leak. Figures 16D, 16E, 16F, and 16G illustrate the RPS circuit dynamics in the leak perturbation settings. Like the BIAMP circuit, the oscillations are sustained in all the settings.

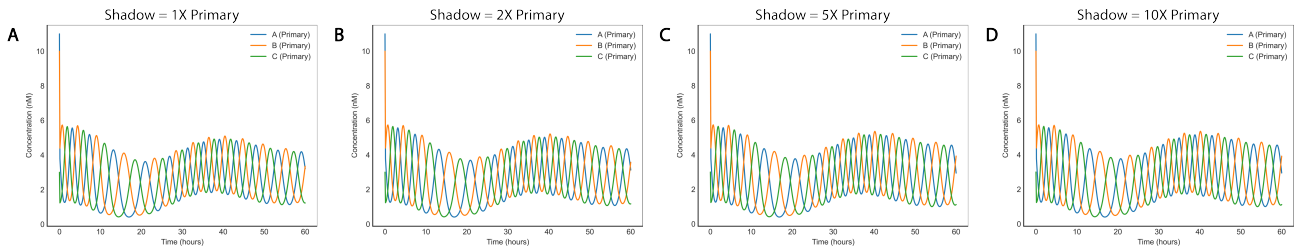


Figure 14: Peppercorn simulations of perturbing the rate constants of the shadow circuit concerning the primary circuit. (a) Shadow = $1.5 \times$ Primary (b) Shadow = $2 \times$ Primary (c) Shadow = $5 \times$ Primary (d) Shadow = $10 \times$ Primary

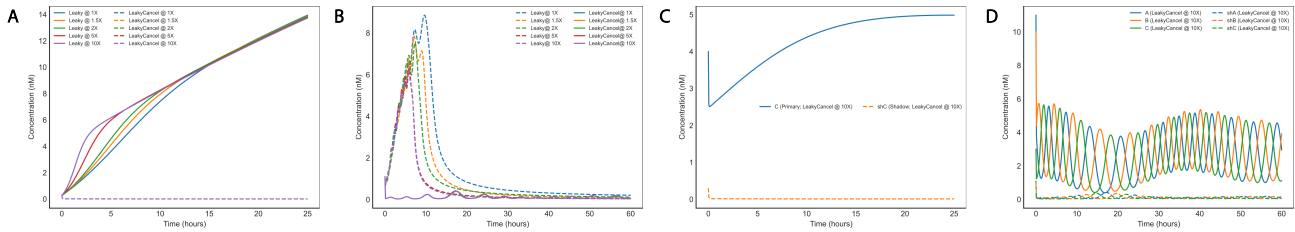


Figure 15: Dynamics of the shadow cancellation-enabled BIAMP circuit when the shadow circuit is out-of-phase with the primary circuit under different perturbation settings.

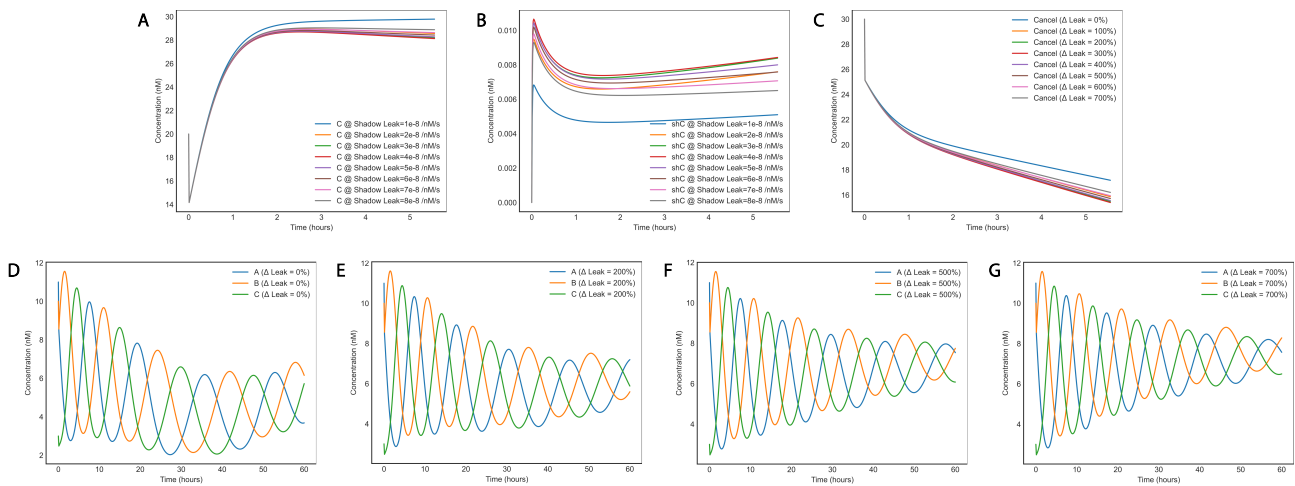


Figure 16: Results of Perturbing the leak rate. We observe that doubling the leak rate constant would not sustain the oscillations in the RPS oscillator.

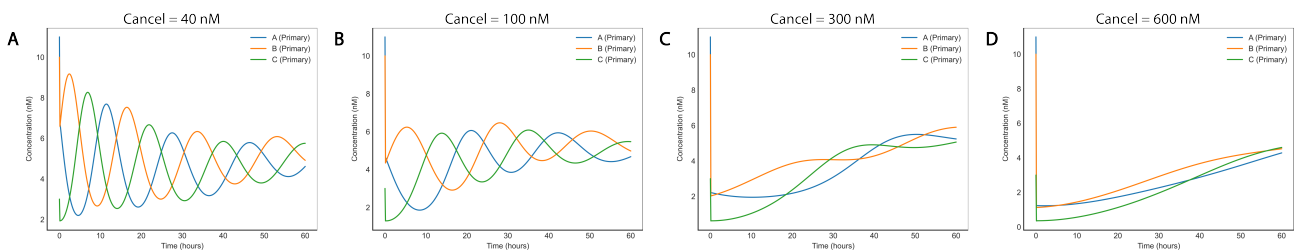


Figure 17: Effects of toehold occlusion by cancellation complexes in the RPS oscillator.

(i) Fast cancellation reactions and Toehold occlusion

Another requirement of the shadow cancellation methodology is that the cancellation reactions should be much faster than the fastest strand displacement reaction in the circuit so that the leaked strands are swiftly removed before they can interact with their parent circuit. Increasing the concentration of the cancellation complexes is not a viable strategy due to a side effect known as *toehold occlusion*. Due to the trimolecular nature of the cancellation mechanism, the absence of the 'cancellation' counterpart signal (e.g., *shCj* from the shadow circuit and *Cj* from the primary circuit act as mutual cancellation counterparts) leads to the signal being temporarily sequestered by the cancellation complex, leading to the suppression of circuit dynamics. This effect can be observed as a dip in the initial signal concentration (e.g., Figure 7D) and the subsequent underexpression of the signal amplification (e.g., Figure 5C). Figure 17 illustrates the change in the circuit dynamics of the RPS oscillator under different initial concentrations of the cancellation complexes. We observe a huge drop in performance when the cancellation complexes are set to an initial concentration of 300 nM, whereas the circuit can no longer sustain oscillations at a concentration of 600 nM. This is likely because the cancellation complexes completely sequester one or more of the signals, leading to the collapse of circuit behavior.

Srinivas *et al.* [17] reported the signal sequestration by the React complexes and utilized the Backward strand as a fuel species to keep the signal in its single-stranded form. They also suggested weakening the toeholds to avoid excessive sequestering. However, weakening the toeholds could also slow down the strand displacement reactions. In this work, we partially handle this issue by adjusting the cancellation complexes to trade-off between maximizing the leak suppression and minimizing toehold occlusion.

5. Conclusion

Through domain-level DSD simulations, we demonstrate that shadow cancellation (in conjunction with buffering

whenever applicable) can effectively mitigate the ill effects of leaks in nucleic-acid dynamical systems known to leak profusely. We propose a semi-automated heuristic method to construct the shadow circuit given only the DNA sequences of the primary circuit, making the method generalizable to arbitrary strand displacement circuits. The DSD circuits Our target dynamical systems—two autocatalytic amplifiers, a dynamic oscillator, a consensus protocol, and a feedback controller – are constructed using the React-Produce framework proposed by Srinivas *et al.* [17]. We use the designs and experimentally observed rate parameters for the general bimolecular reaction from this work and adapt them to assemble our target dynamical systems. Each dynamical system is simulated in four different settings that describe the circuit behavior under various configurations of leak and cancellation. All our chosen dynamical systems showed significant improvement in the dynamics under shadow cancellation. Since the overall shadow cancellation methodology makes several idealizing assumptions, we studied various practical considerations of these assumptions and showed that shadow cancellation is adequately robust to deviations in these assumptions. Simulating the out-of-phase dynamics (when the shadow circuit runs faster than the primary circuit) showed the circuit was stable even at 10× perturbation. By quickly canceling the leak activity, the cancellation complexes restrict the out-of-phase behavior of the shadow circuit to affect the overall circuit dynamics. Further, oscillations were sustained in the RPS circuit even when the shadow circuit leaked with a rate constant 1.5× than the primary circuit (this equals 0.18 /nM/hr more leak). Finally, we showed that the reduction of the strand displacement speed for shadow cancellation can be mitigated by scaling up the concentrations. We showed that even at 10× scale-up of the overall circuit, the circuit dynamics were sustained and reached a steady state in less than half the time it took for the baseline circuit. The possibility of robustly constructing dynamical systems can lead to novel applications in medical diagnostics, biosensing, smart therapeutics, etc.

Authors' Contributions. Rajiv Teja Nagipogu (Rajiv) and John H. Reif (John) jointly conceived the general shadow cancellation technique. Rajiv performed the simulations and wrote the article. John supervised the work and also edited the article.

Competing Interests. The authors have no competing interests.

Funding. This work is funded by the National Science Foundation under grant no. 1909848 and 2113941 to JR

Acknowledgements. The authors would like to acknowledge the discussion with Dan Fu.

References

- ¹E. Katz, *Biomolecular information processing: from logic systems to smart sensors and actuators* (John Wiley & Sons, 2013).
- ²Q. Ma, M. Zhang, C. Zhang, X. Teng, L. Yang, Y. Tian, J. Wang, D. Han, and W. Tan, "An automated DNA computing platform for rapid etiological diagnostics", *Science Advances* **8**, eade0453 (2022).
- ³X. Song, M. Yang, F. Coulter, Z. Lyski, F. Tafesse, W. Messer, and J. Reif, "Rapid in-home testing of COVID-19 with a simple, inexpensive, lyophilized SARS-CoV-2 molecular test kit", in preparation (2020).
- ⁴S. J. Jones, A. F. Taylor, and P. A. Beales, "Towards feedback-controlled nanomedicines for smart, adaptive delivery", *en, Exp. Biol. Med.* **244**, 283–293 (2019).
- ⁵N. Nandagopal and M. B. Elowitz, "Synthetic biology: integrated gene circuits", *science* **333**, 1244–1248 (2011).
- ⁶M. Feinberg and F. J. M. Horn, "Dynamics of open chemical systems and the algebraic structure of the underlying reaction network", *Chem. Eng. Sci.* **29**, 775–787 (1974).
- ⁷M. Feinberg, *Foundations of Chemical Reaction Network Theory* (Springer International Publishing).
- ⁸F. Horn and R. Jackson, "General mass action kinetics", *Arch. Ration. Mech. Anal.* **47**, 81–116 (1972).
- ⁹D. Y. Zhang and G. Seelig, "Dynamic DNA nanotechnology using strand-displacement reactions", *Nature chemistry* **3**, 103–113 (2011).
- ¹⁰B. Yurke, A. J. Turberfield, A. P. Mills, F. C. Simmel, and J. L. Neumann, "A DNA-fuelled molecular machine made of DNA", *Nature* **406**, 605–608 (2000).
- ¹¹A. Eshra, S. Shah, T. Song, and J. H. Reif, "Renewable DNA Hairpin-Based Logic Circuits", *IEEE Transactions on Nanotechnology* **18**, 252–259 (2019).
- ¹²X. Song, A. Eshra, C. Dwyer, and J. H. Reif, "Renewable DNA Seesaw Logic Circuits Enabled by Photoregulation of Toehold-Mediated Strand Displacement", *RSC Advances* **7**, 28130–28144 (2017).
- ¹³L. Qian and E. Winfree, "A simple DNA gate motif for synthesizing large-scale circuits", *Journal of the Royal Society Interface*, rsif20100729 (2011).
- ¹⁴T. Song, S. Garg, R. Mokhtar, H. Bui, and J. H. Reif, "Analog computation by DNA strand displacement circuits", *ACS synthetic biology* **5**, 898–912 (2016).
- ¹⁵T. Song, S. Garg, R. Mokhtar, H. Bui, and J. H. Reif, "Design and Analysis of Compact DNA Strand Displacement Circuits for Analog Computation Using Autocatalytic Amplifiers", *ACS Synth. Biol.* **7**, 46–53 (2018).
- ¹⁶T. Song, N. Gopalkrishnan, A. Eshra, S. Garg, R. Mokhtar, H. Bui, H. Chandran, and J. H. Reif, "Improving the Performance of DNA Strand Displacement Circuits by Shadow Cancellation", *ACS nano* (2018).
- ¹⁷N. Srinivas, J. Parkin, G. Seelig, E. Winfree, and D. Soloveichik, "Enzyme-free nucleic acid dynamical systems", *bioRxiv*, 138420 (2017).
- ¹⁸K. Oishi and E. Klavins, "Biomolecular implementation of linear I/O systems", *IET Systems Biology* **5**, 252–260 (2011).
- ¹⁹B. Yordanov, J. Kim, R. L. Petersen, A. Shudy, V. V. Kulkarni, and A. Phillips, "Computational design of nucleic acid feedback control circuits", *ACS synthetic biology* **3**, 600–616 (2014).
- ²⁰M. Whitby, L. Cardelli, M. Kwiatkowska, L. Laurenti, M. Tribastone, and M. Tschaikowski, "PID Control of Biochemical Reaction Networks", *IEEE Trans. Automat. Contr.* **67**, 1023–1030 (2022).
- ²¹Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig, "Programmable chemical controllers made from DNA", *Nature nanotechnology* **8**, 755–762 (2013).
- ²²L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades", *Nature* **475**, 368–372 (2011).
- ²³M. R. Lakin and D. Stefanovic, "Supervised learning in adaptive DNA strand displacement networks", *ACS synthetic biology* **5**, 885–897 (2016).
- ²⁴I. Zarubiieva, C. Spaccasassi, V. Kulkarni, and A. Phillips, "Automated Leak Analysis of Nucleic Acid Circuits", *en, ACS Synth. Biol.* **11**, 1931–1948 (2022).
- ²⁵J. SantaLucia Jr, H. T. Allawi, and P. A. Seneviratne, "Improved nearest-neighbor parameters for predicting DNA duplex stability", *en, Biochemistry* **35**, 3555–3562 (1996).
- ²⁶B. Wang, C. Thachuk, A. D. Ellington, and D. Soloveichik, "The Design Space of Strand Displacement Cascades with Toehold-Size Clamps", in *DNA Computing and Molecular Programming* (2017), pages 64–81.
- ²⁷C. Thachuk, E. Winfree, and D. Soloveichik, "Leakless DNA strand displacement systems", in *International Workshop on DNA-Based Computers* (Springer, 2015), pages 133–153.
- ²⁸D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics", *en, Proc. Natl. Acad. Sci. U. S. A.* **107**, 5393–5398 (2010).

- ²⁹L. Cardelli, "Two-domain DNA strand displacement", *Mathematical Structures in Computer Science* **23**, 247–271 (2013).
- ³⁰L. P. Reynaldo, A. V. Vologodskii, B. P. Neri, and V. I. Lyamichev, "The kinetics of oligonucleotide replacements", *en, J. Mol. Biol.* **297**, 511–520 (2000).
- ³¹D. Y. Zhang, "Cooperative hybridization of oligonucleotides", *Journal of the American Chemical Society* **133**, 1077–1086 (2010).
- ³²R. A. Wever, *The circadian system of man: results of experiments under temporal isolation* (Springer Science & Business Media, 2013).
- ³³R. Refinetti and M. Menaker, "The circadian rhythm of body temperature", *Physiology & behavior* **51**, 613–637 (1992).
- ³⁴M. Preiner, J. C. Xavier, A. d. N. Vieira, K. Kleineremanns, J. F. Allen, and W. F. Martin, "Catalysts, autocatalysis and the origin of metabolism", *en, Interface Focus* **9**, 20190072 (2019).
- ³⁵M. B. Miller and B. L. Bassler, "Quorum sensing in bacteria", *Annual Reviews in Microbiology* **55**, 165–199 (2001).
- ³⁶C. Fuqua, S. C. Winans, and E. P. Greenberg, "Census and consensus in bacterial ecosystems: the LuxR-LuxI family of quorum-sensing transcriptional regulators", *Annual review of microbiology* **50**, 727–751 (1996).
- ³⁷W. B. Cannon, "The wisdom of the body", (1939).
- ³⁸S. C. Lenaghan, Y. Wang, N. Xi, T. Fukuda, T. Tarn, W. R. Hamel, and M. Zhang, "Grand challenges in bioengineered nanorobotics for cancer therapy", *en, IEEE Trans. Biomed. Eng.* **60**, 667–673 (2013).
- ³⁹D. A. LaVan, T. McGuire, and R. Langer, "Small-scale systems for in vivo drug delivery", *en, Nat. Biotechnol.* **21**, 1184–1191 (2003).
- ⁴⁰Y. Hua, J. Ma, D. Li, and R. Wang, "DNA-Based Biosensors for the Biochemical Analysis: A Review", *en, Biosensors* **12** (2022).
- ⁴¹F. Hong, X. Chen, Y. Cao, Y. Dong, D. Wu, F. Hu, and N. Gan, "Enzyme- and label-free electrochemical aptasensor for kanamycin detection based on double stir bar-assisted toehold-mediated strand displacement reaction for dual-signal amplification", *en, Biosens. Bioelectron.* **112**, 202–208 (2018).
- ⁴²S. Badelt, C. Grun, K. V. Sarma, B. Wolfe, S. W. Shin, and E. Winfree, "A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures", *en, J. R. Soc. Interface* **17**, 20190866 (2020).
- ⁴³B. Kerr, M. A. Riley, M. W. Feldman, and B. J. M. Bohannan, "Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors", *en, Nature* **418**, 171–174 (2002).
- ⁴⁴A. J. Lotka, "Contribution to the Theory of Periodic Reactions", *J. Phys. Chem.* **14**, 271–274 (1910).
- ⁴⁵L. Cardelli and A. Csikász-Nagy, "The cell cycle switch computes approximate majority", *en, Sci. Rep.* **2**, 656 (2012).
- ⁴⁶D. Angluin, J. Aspnes, and D. Eisenstat, "A simple population protocol for fast robust approximate majority", *en, Distrib. Comput.* **21**, 87–102 (2008).
- ⁴⁷G. Gines, A. S. Zadorin, J.-C. Galas, T. Fujii, A. Estevez-Torres, and Y. Rondelez, "Microscopic agents programmed by DNA circuits", *en, Nat. Nanotechnol.* **12**, 351–359 (2017).
- ⁴⁸J. Li, B. Esteban-Fernández de Ávila, W. Gao, L. Zhang, and J. Wang, "Micro/Nanorobots for Biomedicine: Delivery, Surgery, Sensing, and Detoxification", *en, Sci Robot* **2** (2017).
- ⁴⁹B. Yordanov, J. Kim, R. L. Petersen, A. Shudy, V. V. Kulkarni, and A. Phillips, "Computational design of nucleic acid feedback control circuits", *en, ACS Synth. Biol.* **3**, 600–616 (2014).
- ⁵⁰C. Cosentino and D. Bates, *Feedback control in systems biology* (Crc Press, 2011).
- ⁵¹D. Y. Zhang and E. Winfree, "Control of DNA strand displacement kinetics using toehold exchange", *Journal of the American Chemical Society* **131**, 17303–17314 (2009).