

RESEARCH ARTICLE

Synthesizing Configurable Biochemical Implementation of Linear Systems from Their Transfer Function Specifications

Tai-Yin Chiu^{1,2}, Hui-Ju K. Chiang^{2,3}, Ruei-Yang Huang², Jie-Hong R. Jiang^{2,4*}, François Fages³

1 Department of Physics, National Taiwan University, Taipei, Taiwan, **2** Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, **3** EPI Lifeware, Inria Paris-Rocquencourt, Rocquencourt, France, **4** Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

* jhjiang@ntu.edu.tw



CrossMark
click for updates

OPEN ACCESS

Citation: Chiu T-Y, Chiang H-JK, Huang R-Y, Jiang J-HR, Fages F (2015) Synthesizing Configurable Biochemical Implementation of Linear Systems from Their Transfer Function Specifications. PLoS ONE 10 (9): e0137442. doi:10.1371/journal.pone.0137442

Editor: Jean Peccoud, Virginia Tech, UNITED STATES

Received: April 3, 2015

Accepted: August 16, 2015

Published: September 9, 2015

Copyright: © 2015 Chiu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: The Ministry of Science and Technology, Taiwan, provided support in the form of salaries for authors Tai-Yin Chiu, Hui-Ju Chiang, Ruei-Yang Huang, and Jie-Hong R. Jiang, but did not have any additional role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. The specific roles of these authors are articulated in the "author contributions" section.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

The ability to engineer synthetic systems in the biochemical context is constantly being improved and has a profound societal impact. Linear system design is one of the most pervasive methods applied in control tasks, and its biochemical realization has been proposed by Oishi and Klavins and advanced further in recent years. However, several technical issues remain unsolved. Specifically, the design process is not fully automated from specification at the transfer function level, systems once designed often lack dynamic adaptivity to environmental changes, matching rate constants of reactions is not always possible, and implementation may be approximative and greatly deviate from the specifications. Building upon the work of Oishi and Klavins, this paper overcomes these issues by introducing a design flow that transforms a transfer-function specification of a linear system into a set of chemical reactions, whose input-output response precisely conforms to the specification. This system is implementable using the DNA strand displacement technique. The underlying configurability is embedded into primitive components and template modules, and thus the entire system is adaptive. Simulation of DNA strand displacement implementation confirmed the feasibility and superiority of the proposed synthesis flow.

Introduction

Advancements in synthetic biology have resulted in the development of biochemical systems of increasing complexity that are capable of using living cells as well as cell-free systems. Synthetic biology holds promise for biotechnology, biomedicine, bio-environmental, bioenergy, and other applications. Many computation and control design examples have been demonstrated either *in vivo* or *in vitro*. For instance, oscillators [1], toggle switches [2], logic gates [3], band pass filters [4], and analog circuits [5] have been designed and implemented in living cells, while digital circuits [6], neural networks [7], and switchable memories [8] have been demonstrated in cell-free systems. These developments show a clear growing trend in design

complexity. Indeed, the more sophisticated systems we can build, the better skills we will have to comprehend biology and to build even more sophisticated systems.

The increasing system complexity renders the necessity of design automation tools. Constructing biochemical systems bottom-up using pre-characterized parts in synthetic biology is analogous to designing electronic systems using pre-designed standard modules. While electronic design automation (EDA) has largely enabled system design under exponential capacity growth over the past five decades thanks to the driving force of Moore's law, design automation in synthetic biology may play a similar key role in the construction of complex biochemical systems [9]. Computer-aided modeling, simulation, synthesis, and verification are crucial because a biochemical design is often intended to function in a biological context that is often too complex to be fully characterized.

Depending on its target application, a system is often designed for a specific application rather than as a general purpose computing machine. In control applications, linear systems are pervasive due to their simplicity of design and analysis. Any linear control system can be realized with three primitive components: integration, gain, and summation. Realizing linear control with biochemical reactions has been proposed by Oishi and Klavins [10], who showed that three types of chemical reactions (catalysis, degradation, and annihilation) are sufficient to implement the three primitive components. In principle, any polynomial ordinary differential equation can be approximated by chemical reaction networks (CRNs) [11]. Therefore, any linear control system can be built using CRNs.

To realize CRNs, nucleic acids have been exploited as a universal tool for biomolecular computation [12]. By appropriately changing their nucleotide sequences, the interaction between nucleic acids can be precisely controlled and programmed *in vitro*. Specifically, toehold-mediated or enzyme-free DNA strand displacement (DSD) is a promising approach to perform biological computations. Although the DSD mechanism has been studied since the 1970s [13–17], until recently it was systematically used to build a molecular machine made of DNA and RNA [18]. The DSD method is attractive for several reasons. First, the kinetics of DSD devices can be engineered by controlling toehold binding rates, which may range from 1 to $6 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$ [19–21]. Second, composability [22] can be achieved. In typical implementation schemes, single-stranded DNAs play the role of signals and double-stranded DNAs act as gates, and they work together with the mechanism of toehold-mediated DNA displacement [19]. DNA signals of different domain lengths [23–25] have been proposed. For example, gates can be composed to realize complex systems using a 2-domain signal architecture [25]. Third, the DSD mechanism works autonomously [26] as long as the DNA or RNA fuels are supplied. With these advantages, many biomolecular devices, e.g., [27–29], have been designed using this powerful mechanism. The DSD technology allows computation and interfacing with molecular components in living organisms [30] and shows potential in bio-sensing and control, biomedicine, and other applications.

Despite the advancement of biochemical implementation of linear systems, four challenges remain to be solved. First, the construction of Oishi and Klavins's system requires that the rate constants of the underlying reactions be carefully matched to achieve the intended integration, gain, and summation functions. This requirement imposes substantial practicality restrictions because, in reality, the reaction rates of available reactions can be limited. Thus, not all gain and summation blocks can be realized. Second, once a system is constructed, its function is fixed and cannot be changed without redesign. However, this fixed functionality can be inadequate for a system reacting to its biochemical environment, which is intrinsically stochastic and often full of uncertainty. Thus, designing systems with dynamic adaptation capabilities is crucial, especially in the biochemical context. Third, the CRN implementations of the gain and summation components are approximative. In essence, the transfer functions of the gain and

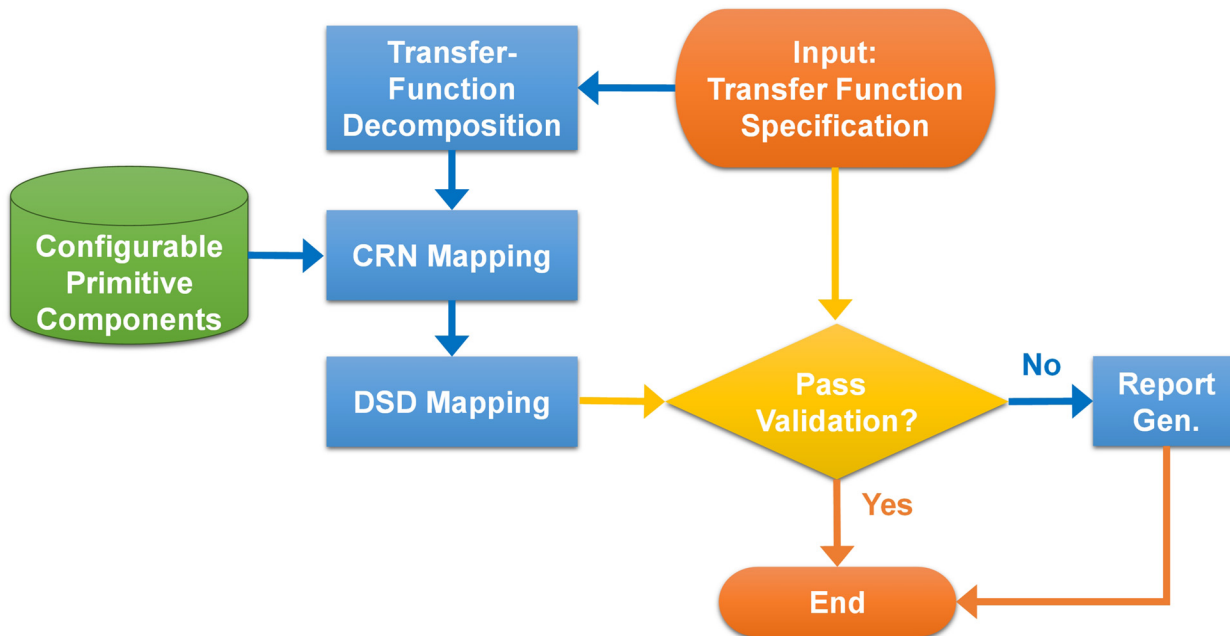


Fig 1. Automated flow of linear system synthesis.

doi:10.1371/journal.pone.0137442.g001

summation components contain extra poles rather than just the ideal scalar and summation. When a complex system is built from these components, these additional poles might cause system behavior that deviates from its specification and can even lead to unwanted instability. Fourth, an automated flow of the synthesis of CRN from the transfer function specification of a linear system is lacking. Although a linear system can be compiled into a CRN with a direct block-by-block conversion from a block diagram, the block diagram may not be available in the first place and the CRN implementation may only approximate the specification. Specifying a linear system using transfer functions can be more natural than using block diagrams. However, converting a transfer function into a block diagram suitable for CRN realization can be a nontrivial optimization task.

In this paper, we tackle the above challenges. To address the first two challenges, we devise a mechanism to make linear control systems configurable by adding auxiliary species to CRN as control knobs. The concentrations of the auxiliary species can be adjusted not only to compensate for reaction rate mismatch but also to reconfigure a control system to dynamically adapt to environmental changes. Hence, implementing linear control systems in biochemistry can be made more practical. To resolve the last two challenges, we apply parallel decomposition on a given transfer function and express it as a summation of elementary modules, and propose a CRN solution that achieves the exact implementation of the elementary modules. The CRN implementation of a transfer function can be further mapped and realized using the DSD method. The proposed method lends itself to an automated design flow (as depicted in Fig 1) where the linear system to be synthesized is specified by a transfer function, which is decomposed for CRN implementation and further mapped into the DSD reactions. Through simulation using Visual DSD [31, 32], we show the feasibility and superiority of our proposed design automation flow for synthesizing a specified linear system into DSD reactions.

Methods and Results

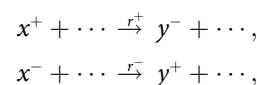
Although synthetic biology shows promising potential in biotechnology applications, it poses grand challenges to complex system design. A typical reactive system may involve sensing, actuation, information processing, and other tasks, while a control unit is one of the fundamental ingredients that constitute a reactive system. In this work, we address the design automation problem of linear control systems that are to be realized by biochemical reactions.

Any linear time-invariant (LTI) system can be fully specified by means of a transfer function, which corresponds to the Laplace transform of the impulse response of the system, with all initial conditions set to zero. In essence, once the impulse response of a linear system is known, the output $y(t)$ of the system with any input $u(t)$ can be characterized using the transfer function. However, for a multiple-input multiple-output (MIMO) linear system, the principle of superposition suggest that the total effect on any output due to all the inputs acting simultaneously is obtained by summing up the output effects due to the individual inputs acting alone. Consequently any MIMO system can be specified by a matrix of transfer functions. For simplicity, in this paper we focus on single-input single-output (SISO) systems without loss of generality.

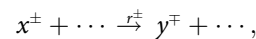
Configurable Primitive Components

A transfer function can be realized by building a network of primitive components consisting of integration, gain, and summation blocks; therefore, implementing an LTI system in biochemical systems amounts to realizing these three basic components with chemical reactions [10]. However, the CRN realizations of these components proposed by Oishi and Klavins [10] are deficient in that rate constants have to be carefully matched and the designed control system cannot be reconfigured. These deficiencies impose a practicality issue that restricts system realization and a flexibility issue that prevents dynamic system adaptation. We can overcome these issues by introducing configurable primitive components as discussed below.

Following Oishi and Klavins [10], we represent a real signal x by the difference ($x^+ - x^-$) between the concentrations of two complementary molecular species x^+ and x^- . In this paper, we do not distinguish notationally between a species and its concentration. Moreover, we abbreviate a pair of chemical reactions of complementary species



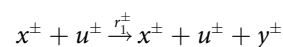
with rate constants r^+ and r^- , to



where the plus and minus signs in the superscripts \pm and \mp are ordered.

Table 1 shows the chemical reactions and transfer functions of the proposed configurable primitive components, which we will elaborate on in the following sections.

Integration Block. An integration block takes a signal $u(t)$ as input, and produces a signal $y(t) = k \int_0^t u(\tau) d\tau + y(0)$, for some constant $k \in \mathbb{R}$, as output. The chemical realization of an integration block for $k \geq 0$ consists of a pair of catalytic reactions



and an annihilation reaction

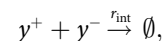


Table 1. Three primitive components, their chemical reactions, and their transfer functions.

Type	Reactions	Transfer Function
Integration	$\begin{cases} x^\pm + u^\pm \xrightarrow{r_1^\pm} x^\pm + u^\pm + y^\pm \\ y^+ + y^- \xrightarrow{r_{int}} \emptyset \end{cases}$	$Y(s) = \frac{k}{s} U(s),$ $k \equiv r_1^+ x^+ = r_1^- x^-$
Gain & Summation	$\begin{cases} x_i^\pm + u_i^\pm \xrightarrow{r_i^\pm} x_i^\pm + u_i^\pm + y^\pm, i = 1, \dots, n \\ z^\pm + y^\pm \xrightarrow{r_0^\pm} z^\pm \\ y^+ + y^- \xrightarrow{r_{gs}} \emptyset \end{cases}$	$Y(s) = \sum_{i=1}^n \frac{k_i}{s + k_0} U_i(s),$ $\begin{cases} k_i \equiv r_i^+ x_i^+ = r_i^- x_i^-, i = 1, \dots, n \\ k_0 \equiv r_0^+ z^+ = r_0^- z^- \end{cases}$

doi:10.1371/journal.pone.0137442.t001

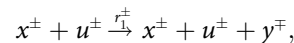
where r_1^+, r_1^-, r_{int} are the rate constants. The reactions differ from those of Oishi and Klavins [10] in that the auxiliary species x^\pm are newly added to the catalytic reactions. Both the auxiliary species x^\pm and the input species u^\pm serve as catalysts.

With the definition that $r_1^+ x^+ = r_1^- x^- \equiv k$, the signal y is exactly the integration of signal u as described by the ordinary differential equation

$$\dot{y} = \dot{y}^+ - \dot{y}^- = r_1^+ x^+ u^+ - r_1^- x^- u^- = ku,$$

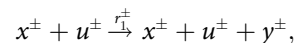
for $\dot{y}^\pm = r_1^\pm x^\pm u^\pm - r_{int} y^+ y^-$. Taking the Laplace transform of the above equation, we obtain the transfer function $\frac{Y}{U} = \frac{k}{s}$.

Because the concentrations of x^+ and x^- can be controlled externally, in theory it is always possible to design a reaction network to meet any required k . For $k < 0$, the reactions are the same except that the catalytic reactions should be modified by reversing the complementary species of the output signal y as

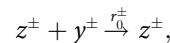


where y^\mp replaces the original y^\pm .

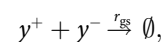
Gain and Summation Blocks. A weighted summation block takes a number of input signals $u_i(t)$, $i = 1, 2, \dots, n$ and produces an output signal $y(t) = \sum_{i=1}^n k_i u_i(t)$ for $k_i \in \mathbb{R}$. A gain block is a special weighted summation block with only one input $u(t)$ that produces output $y(t) = k_1 u(t)$ for $k_1 \in \mathbb{R}$. The gain block with $k_1 \geq 0$ can be realized by one pair of catalytic reactions,



one pair of degradation reactions,



and an annihilation reaction



where x^\pm and z^\pm are auxiliary species. These reactions induce the following kinetic equations

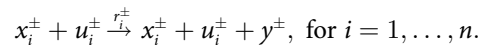
$$\dot{y}^\pm = r_1^\pm x^\pm u^\pm - r_0^\pm z^\pm y^\pm - r_{gs} y^+ y^-.$$

Let $k_1 \equiv r_1^+ x^+ = r_1^- x^-$ and $k_0 \equiv r_0^+ z^+ = r_0^- z^-$. The mass action of y becomes

$$\dot{y} = k_1(u^+ - u^-) - k_0(y^+ - y^-) = k_1 u - k_0 y. \tag{1}$$

When the steady state is reached, the changing rate of y (i.e., \dot{y}) equals zero, which implies $y = \frac{k_1}{k_0} u$. Note that the concentrations of the auxiliary species x^\pm and z^\pm are controlled externally. Therefore, it becomes theoretically possible to meet any required k_1 by tuning the concentrations of these auxiliary species. For $k_1 < 0$, the plus and minus signs of the superscripts of y in the pair of catalytic reactions should be swapped.

For the weighted summation block, the reactions are the same as those for the gain block except that the pair of catalytic reactions becomes



By setting $k_i \equiv r_i^+ x_i^+ = r_i^- x_i^-$ we obtain the equation

$$y = \sum_i \frac{k_i}{k_0} u_i \tag{2}$$

in the steady state. Similarly, if a scaling factor $k_i < 0$, we swap the signs in the superscript of y in the reaction corresponding to input u_i .

It is worth noting that in contrast to the integration block, the weighted summation block only approximates the intended weighted summation when the steady state is not yet reached; this is often the case in practice. This approximation can be clearly seen from its transfer function $\frac{Y}{U} = \sum_i \frac{k_i}{s + k_0}$, rather than the ideal $\frac{Y}{U} = \sum_i \frac{k_i}{k_0}$. Although this approximation may seem to inevitably cause a chemical reaction implementation to deviate from its specification, we will later show methods that can be used to circumvent this imperfection.

Case Study. To assess the benefit of the auxiliary species, we perform a proof-of-concept case study on the mass-spring-damper (MSD) system as shown in Fig 2A. The system can be modeled by the equation

$$M\ddot{x} + b\dot{x} + kx = F.$$

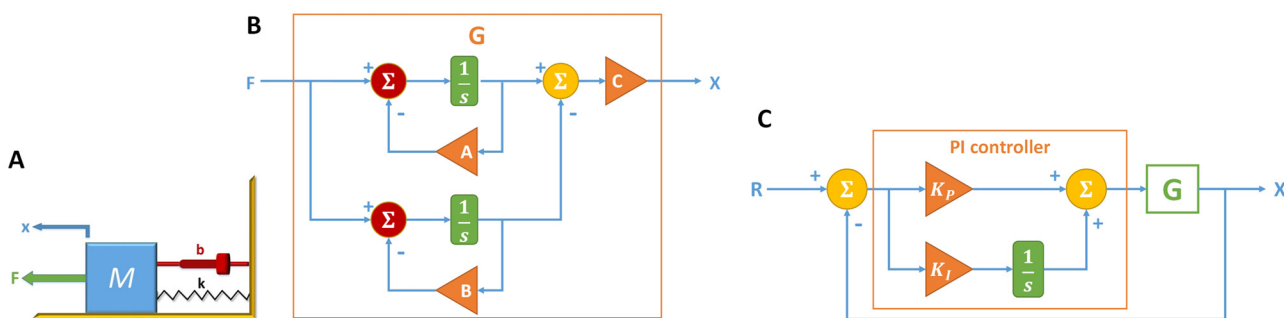


Fig 2. Mass-spring-damper system and its block diagrams. (A) MSD system. (Let $F = 1$ N, $M = 1$ kg, $b = 10$ N s/m, and $k = 20$ N/m.) (B) Block diagram of the MSD model, where the triangular blocks denote gain functions with their corresponding weights, the rectangular blocks denote integrators, and the circle blocks denote mixers for summation and/or subtraction. (Let $A = 2.764$, $B = 7.236$ and $C = 0.2236$.) (C) Block diagram of the PI-controlled MSD model, where G is the plant shown in (B). (Assume the values of K_p and K_t given in Table 2.)

doi:10.1371/journal.pone.0137442.g002

Using $M = 1$ kg, $b = 10$ N s/m, $k = 20$ N/m, and $F = 1$ N, by Laplace transform we derive the transfer function

$$G = \frac{1}{s^2 + 10s + 20} \approx 0.2236 \left(\frac{1}{s + 2.764} - \frac{1}{s + 7.236} \right)$$

The transfer function can be implemented with the block diagram shown in Fig 2B. The proportional-integral (PI) controller to the MSD system is shown in Fig 2C.

The block diagrams are constructed with $k_0 = 10$ for all the summation and gain blocks, with the exception of the summation blocks shown in red and gain blocks A and B with $k_0 = 50$. The values of k_i 's are set to αk_0 , where the values of α are equal to the weights specified in the corresponding gain blocks. Additionally, we assume q_0^+ and q_0^- have the same values as k_0 , and q_1^+ and q_1^- are mismatched to k_1 by 10%, where q_1 and q_0 are the rate constants of the catalytic and degradation reactions, respectively, formulated in the prior work [10].

Fig 3A1, 3A2, 3A3, and 3B show the step, impulse, and sinusoidal responses of the MSD system, and the step responses of the PI-controlled MSD system, respectively. Our method achieves better approximation to the ideal cases than the prior method [10]. One of the advantages of our method is that we can match the weight $\frac{k_1}{k_0}$ by tuning the concentrations of x^\pm and z^\pm . In contrast, no tuning is possible in the prior method [10] to avoid the inexact gain $\frac{q_1}{q_0}$,

where q_1 and q_0 are the rate constants of the catalytic and degradation reactions formulated in the prior work [10], respectively, due to the mismatch of the rate constants. (Note that the biochemical implementations have their own optimal K_P and K_I values, shown in Table 2, to approximate the ideal system.)

Suppose that the spring and damper of the above MSD system are now replaced with new ones for $b = 40$ N s/m and $k = 60$ N/m. Our method can still adapt the PI-controller to the new MSD system without redesigning the PI-controller, whereas the prior method has no such capability. Because we can tune the concentrations of x^\pm and z^\pm in biochemical implementation, it is possible for us to adapt (K_P, K_I) to optimal values (40, 60) for the new PI-controlled MSD system, which is in contrast to the original system (15, 20). Fig 3C compares the results with and without such reconfigurability.

DSD Realization of Configurable Primitive Components

We exploit the DNA strand displacement (DSD) technique [24, 33] as an experimental chassis for our synthesis flow, and map the synthesized CRNs to DSD reactions. The simulation is conducted using the Visual DSD tool [34] for validation.

To implement each primitive component, we map the reactions listed in Table 1 to DSD reactions considering the compatibility among the components. We adopt the two-domain DSD method [25] to compile CRNs into nucleic acid-based chemistry to achieve flexible composability. In this method, any free (i.e., unbound) single-stranded DNA species consist of two domains: one toehold domain and one recognition domain. A toehold may initiate the binding between a single-stranded DNA species and a double-stranded DNA species. The two-domain DSD method supports compositionality, which indicates that a block implemented with DSD reactions can be cascaded with other blocks implemented with DSD reactions. Thus, the output species of one block fits the input species required by its downstream blocks.

As a notational convention, we indicate a domain t to be a toehold by a hat \hat{t} . We also use superscript $*$ to indicate the Watson-Crick complement of a domain DNA sequence (e.g., domain t^* is sequence CTAG if domain t is sequence GATC). For a single-stranded species, we

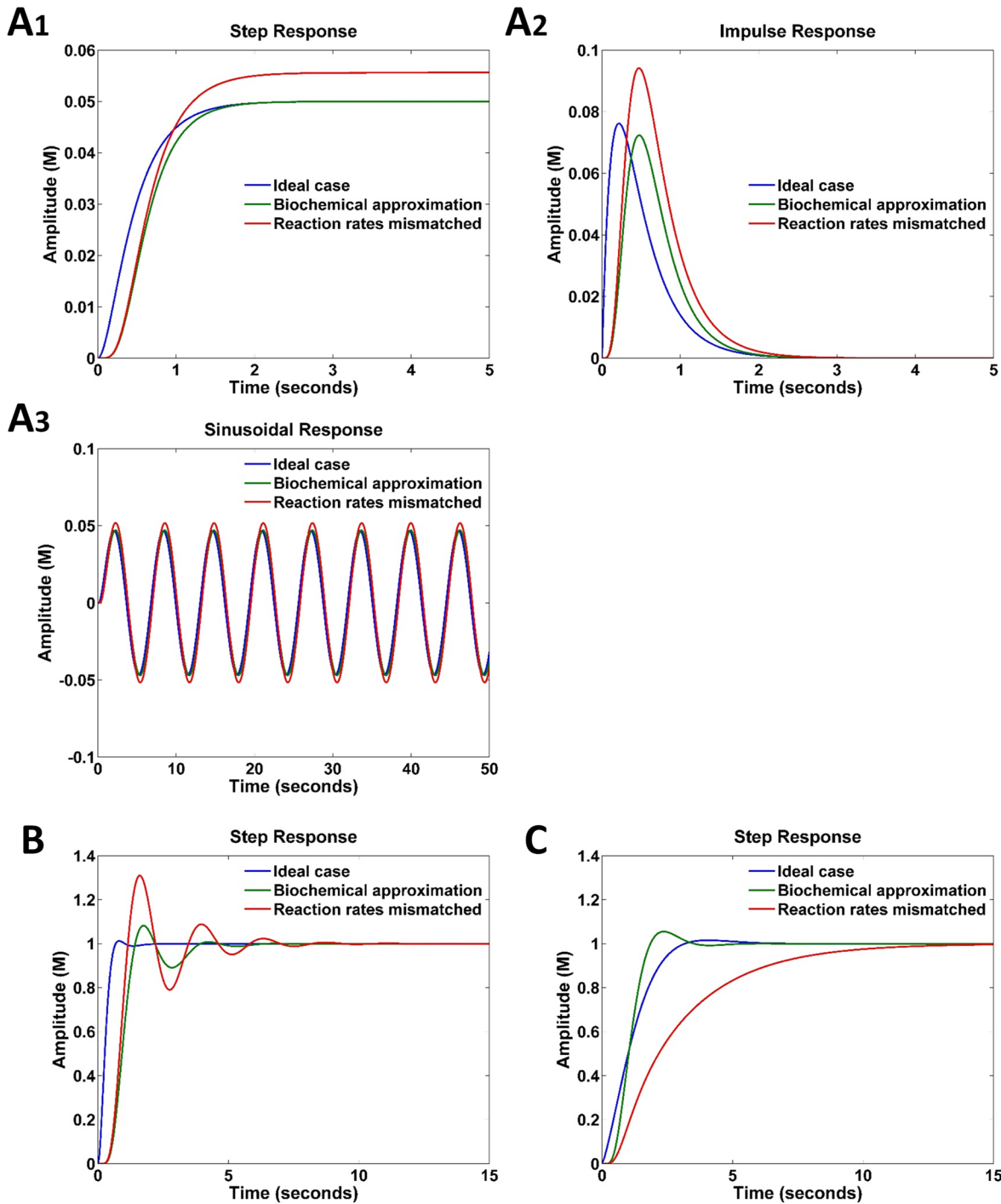


Fig 3. Responses of MSD systems with and without a PI controller. The blue, green, and red curves represent the responses in ideal, configurable biochemical implementation, and nonconfigurable biochemical implementation cases, respectively. (A) Step, impulse, and sinusoidal responses of the MSD. (B) Step responses of PI-controlled MSD. (Assume 10% rate mismatch in the MSD system.) (C) Step responses of PI-controlled MSD, where the MSD undergoes a parameter change with $b = 40$ N s/m and $k = 60$ N/m, respectively, to induce a gain change of $A = 1.561$, $B = 38.44$ and $C = 0.0271$.

doi:10.1371/journal.pone.0137442.g003

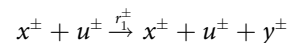
Table 2. Values of (K_p, K_t) in ideal, configurable and non-configurable implementations of the original and new MSD systems.

	Ideal	Config. Imp.	Non-config. Imp.
Original MSD	(30, 70)	(15, 20)	(15, 20)
New MSD	(15, 55)	(40, 60)	(15, 20)

doi:10.1371/journal.pone.0137442.t002

indicate the 5' end using a vertical bar “|” and the 3' end by an angle bracket “)” or “⟨.” We call a single-stranded species of the form |) (5' end on the left and 3' end on the right) and ⟨ | (5' end on the right and 3' end on the left) an upper strand and a lower strand, respectively. A double-stranded species with an upper strand |*t u*) (and a lower strand ⟨*t* u**) is denoted as [|*t u*]. A double-stranded species with one or more dangling (or exposed) toeholds is called a gate, and is used to help transduce one or more species into other species. Double-stranded species without any dangling (exposed) toeholds or single-stranded species containing no toeholds are regarded as waste because they lack the ability to react with other species. The gates and species separately used to construct catalysis, degradation, and annihilation CRNs are called fuel.

To ignore the effects of fuel depletion, we maintain all of the separately used species at constant concentrations. Moreover, we set the binding rates of all toeholds to 0.05 nM⁻¹ s⁻¹, which is consistent with the prior work [35]. We assume infinite unbinding rates [32, 36] to ignore the unproductive reactions [32] that result from the situation where a single-stranded species binds to a gate but does not further displace the strand with which it competes for the same recognition domain. Under these settings, we map the catalytic chemical reactions



to the DSD reactions shown in Fig 4, where the domains highlighted in red and green are toeholds. Reactions $x^+ + u^+ \xrightarrow{r_1^+} x^+ + u^+ + y^+$ and $x^- + u^- \xrightarrow{r_1^-} x^- + u^- + y^-$ can be realized by the same set of DSD reactions except for species renaming (on recognition domains). For brevity, we simply use the same set of species |*t x*) (*catalysis_7*), |*t u*) (*sp_0*), and |*t₂ y*) (*sp10*) to denote the species x^+ , u^+ , and y^+ , respectively, to implement the former reaction and to denote x^- , u^- , and y^- , respectively, for the latter reaction.

The set of DSD reactions outlined in Fig 4 consists of two stages. Gate *catalysis* initiates the first stage of the reactions by reacting with |*t x*). At the end of the first stage, the species |*a t*) will be produced, and this species will react with gate *catalysis_1* to initiate the second stage. During the second stage reactions, the species |*t x*) and |*t u*) will be produced to compensate for their consumption in the first stage.

We note that if the recognition domains of *u* and *x* were exchanged within the seven gates and one waste in the four first stage reactions, the entire set of reactions would still represent an implementation of the intended catalytic chemical reactions. However, the new arrangement of [|*u t*] followed by [|*x t*] would accelerate input |*t u*) (effectively, u^+ or u^- in the catalytic chemical reactions) consumption because the input |*t u*) would directly react with the new *catalysis* gate. This is in contrast to the original *catalysis* structure, where [|*x t*] is followed by [|*u t*]. Therefore, this consumption acceleration of |*t u*) would amplify the bias between the production and consumption rates of |*t u*). Because the concentration of |*t u*) could not remain constant, the input species |*t u*) could not be treated as a catalyst as expected. As a result, the new set of DSD reactions will not be as effective as the example presented in Fig 4.

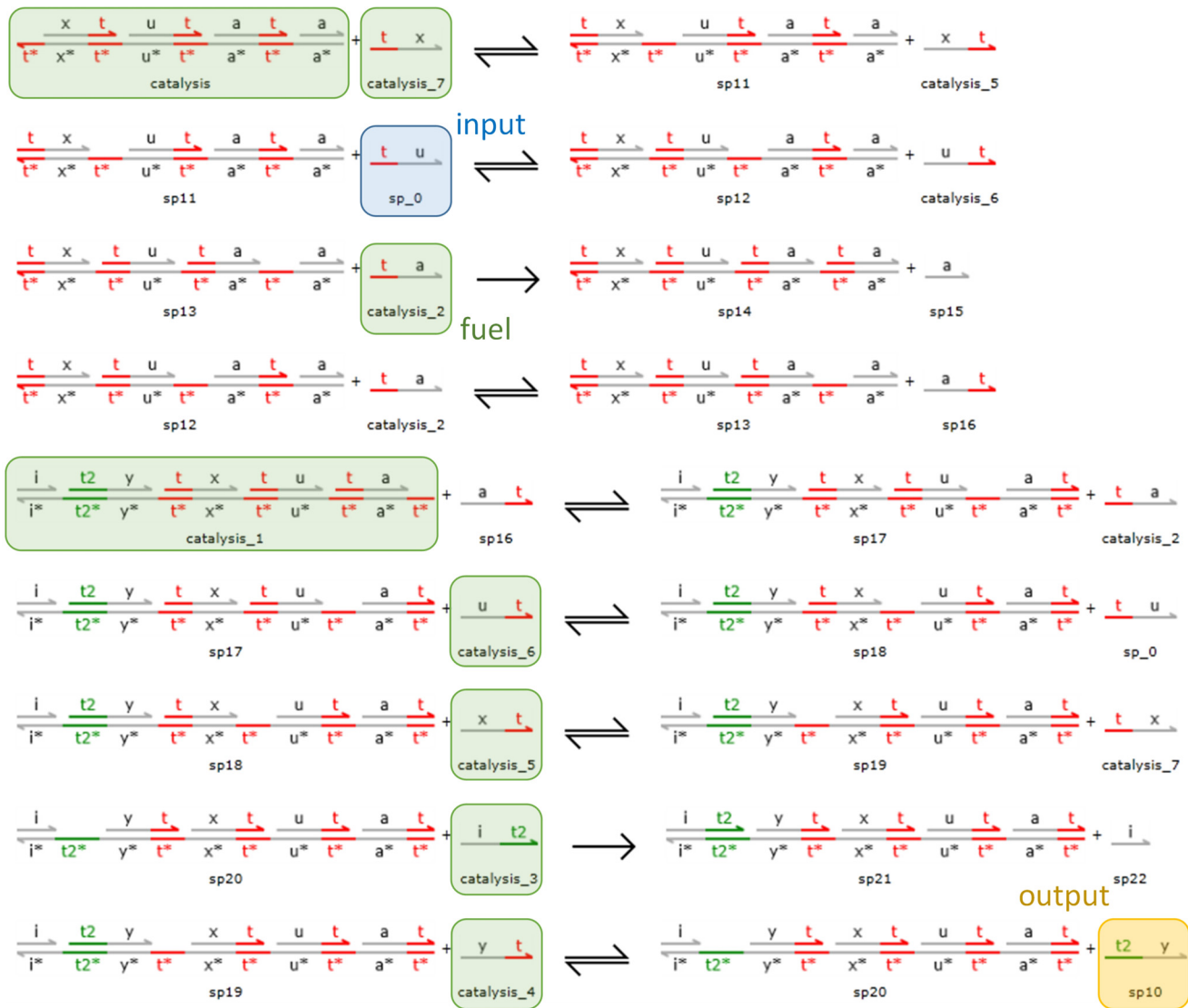


Fig 4. DSD realization of catalytic reactions. The species in green are fuels whose concentrations are fixed to 2000 nM to avoid the effect of fuel depletion. The species *sp_0*, *sp10*, and *catalysis_7* represent the input u^+ or u^- , output y^+ or y^- , and catalyst x^+ or x^- , respectively. Species *catalysis_7* reacts with the gate *catalysis* to begin the first stage of the reactions followed by a series of reactions involving species *sp_0* and *catalysis_2*. At the end of the first stage, species *sp16* is produced and reacts with the gate *catalysis_1* to begin the second stage of the reactions. In this stage, species *catalysis_7* and *sp_0* are yielded to compensate for their consumption in the first stage. Finally, the output species *sp10* is generated.

doi:10.1371/journal.pone.0137442.g004

To calibrate the values of the rate constants r_1^\pm , we conduct an experiment on the catalytic reaction $x^\pm + u^\pm \xrightarrow{r_1^\pm} x^\pm + u^\pm + y^\pm$. Let x^\pm be mapped to $[\hat{t} \ x]$, u^\pm to $[\hat{t} \ u]$ and y^\pm to $[\hat{t}_2 \ y]$ at the concentrations summarized in Table 3. Additionally, we set and maintain the concentrations of all fuel to 2000 nM. However, the concentration of auxiliary species $[\hat{t} \ x]$ can be obtained by the weight $k \equiv r_1^+ x^+ = r_1^- x^-$ after r_1^\pm are determined. Based on the relationship $r_1^+ x^+ u^+ = y^\pm = y_{\text{final}}^\pm / \tau$, where y_{final}^\pm are the concentrations of $[\hat{t}_2 \ y]$ after an elapsed time $\tau = 1000$ seconds, we derive the rate constants r_1^\pm . As seen from Table 3, the reaction rates r_1^\pm hold

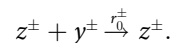
Table 3. Experimental data used for measuring the rate constant of DSD catalysis.

$(\hat{t} u\rangle, \hat{t} x\rangle, \hat{t}_2 y\rangle)$ (nM, nM, nM)	r_1^\pm ($\text{nM}^{-1} \text{s}^{-1}$)
(1, 1, 24.97)	0.0250
(1, 100, 2272)	0.0227
(10, 10, 2469)	0.0247
(100, 1, 2436)	0.0244
(100, 100, 220492)	0.02205
$(10^{-3}, 10^{-3}, 2.49 \times 10^{-3})$	0.0249

doi:10.1371/journal.pone.0137442.t003

almost the same value when the concentrations of $|\hat{t} u\rangle$ and $|\hat{t} x\rangle$ are smaller than 100 nM, but slightly decrease when one of the concentrations approaches 100 nM. Because $|\hat{t} u\rangle$ and $|\hat{t} x\rangle$ will be low concentrations (approximately 10^{-6} to 1 nM) in our later experiments, we set $r_1^\pm = 0.0249 \text{ nM}^{-1} \text{ s}^{-1}$. Fig 5A and 5B show the concentration of $|\hat{t}_2 y\rangle$ over time under the concentration (in nM) settings of $(|\hat{t} x\rangle, |\hat{t} u\rangle)$ at (0.001, 0.001) and (100, 100), respectively. These results suggest that the DSD reactions produce the output y^\pm (i.e., $|\hat{t}_2 y\rangle$) at a constant rate. The linearity of the DSD reactions indicates their adequacy in realizing our proposed configurable primitive components.

Similar to the construction proposed by Yordanov *et al.* when mapping chemical reactions $y^\pm \xrightarrow{q_0} \emptyset$ to DSD reactions [37], we map the degradation chemical reactions



to the DSD reactions shown in Fig 6. For brevity, the DSD reactions shown only correspond to one of the reactions $z^+ + y^+ \xrightarrow{r_0^+} z^+$ and $z^- + y^- \xrightarrow{r_0^-} z^-$. The species $\langle \hat{t}_2^* [y] \rangle$, labeled *deg*, is the only gate or fuel used in this network, while the species $|\hat{t}_2 y\rangle$ represents y^\pm . The products *sp3*

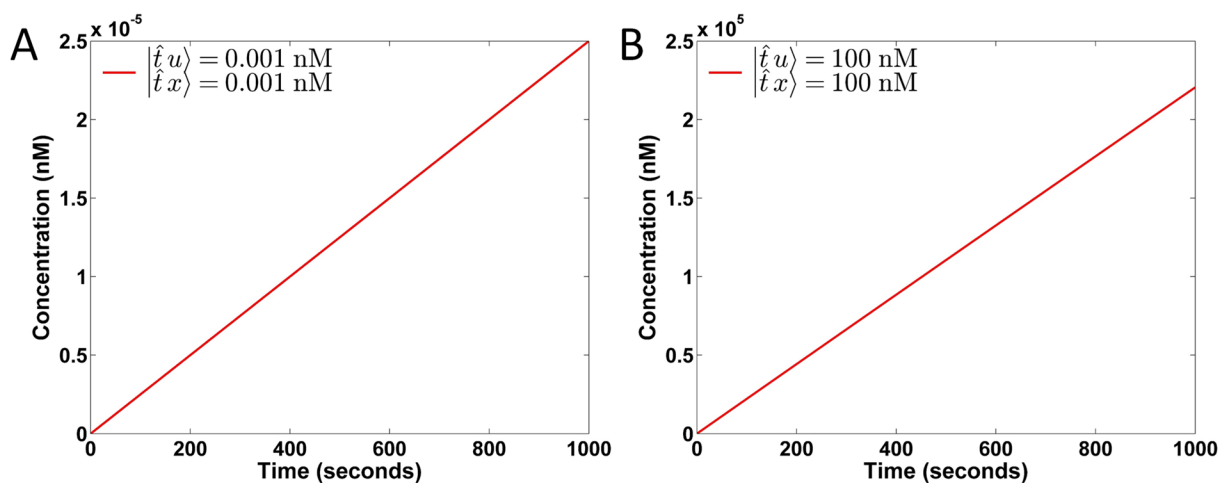


Fig 5. Concentration of $|\hat{t}_2 y\rangle$ over time in DSD realized catalytic reactions. (A) $(|\hat{t} x\rangle, |\hat{t} u\rangle) = (0.001, 0.001)$ in nM and (B) $(|\hat{t} x\rangle, |\hat{t} u\rangle) = (100, 100)$ in nM.

doi:10.1371/journal.pone.0137442.g005

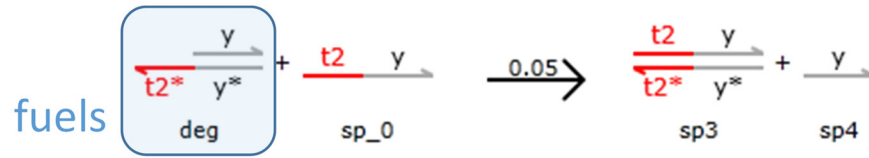
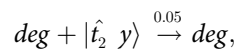


Fig 6. DSD realization of the degradation reaction. Here, *deg* serves as fuel with its concentration maintained at a fixed value and corresponds to catalysts z^\pm . In the reaction, the rate constant equals the toehold binding rate $0.05 \text{ nM}^{-1} \text{ s}^{-1}$.

doi:10.1371/journal.pone.0137442.g006

and *sp4* are waste products. In contrast to the DSD reactions proposed by Yordanov *et al.* [37] where the fuel concentrations are not maintained as constants, we maintain the fuel *deg* at a constant concentration. Therefore, the reaction can be effectively regarded as



which is the same form as the above degradation chemical reactions. Thus, the species *deg* corresponds to z^+ or z^- , and the rate constants r_0^\pm equal $0.05 \text{ nM}^{-1} \text{ s}^{-1}$, which is the binding rate constant of a toehold. By solving the differential equation of the kinetics of y^\pm , we derive that its concentration at time *t* is $y(t)^\pm = y(0)^\pm \exp[-(0.05z^\pm)t]$, where $y(0)^\pm$ are the initial concentrations of $y(t)^\pm$. Fig 7A and 7B plot $y(t)^\pm$ for $z^\pm = 1 \text{ nM}$ and 100 nM , respectively. In each case, there are three curves corresponding to different initial concentrations of $y(0)^\pm = 10, 50$ and 100 nM . These six curves are shown in Fig 7A and 7B and exactly fit the ideal waveforms.

Finally, following the construction scheme of Yordanov *et al.* [37], we map the annihilation chemical reaction



to the DSD reactions provided in Fig 8, where species $|\hat{t}_2 yp\rangle$ and $|\hat{t}_2 ym\rangle$ represent y^+ and y^- ,

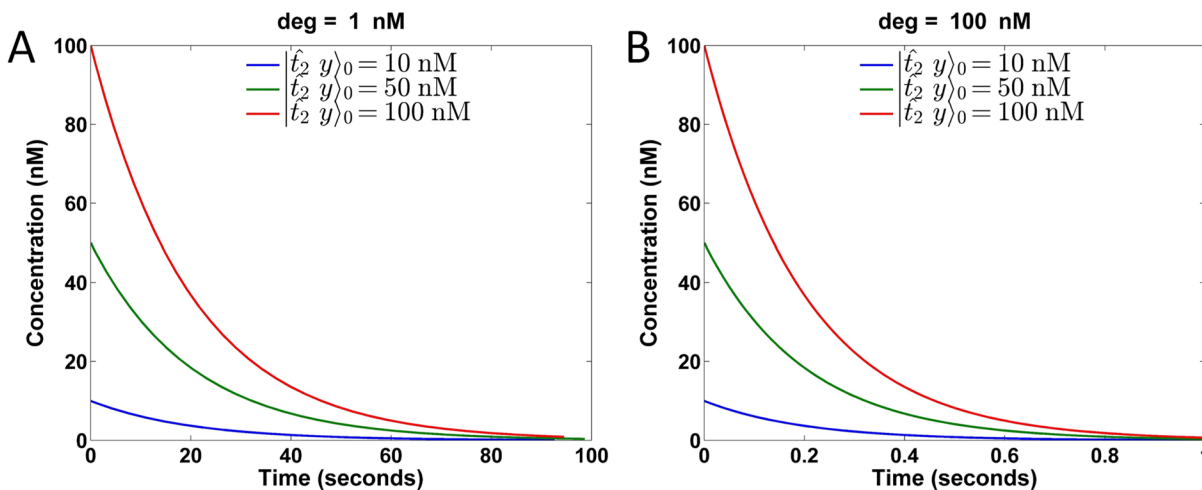


Fig 7. Simulation of DSD realized degradation reactions. (A) Concentration of $y^\pm = |\hat{t}_2 y\rangle$ over time under $deg = z^\pm = 1 \text{ nM}$. (B) Concentration of $y^\pm = |\hat{t}_2 y\rangle$ over time under $deg = z^\pm = 100 \text{ nM}$. The three curves in each case correspond to different initial concentrations $y(0)^\pm = 10, 50,$ and 100 nM .

doi:10.1371/journal.pone.0137442.g007

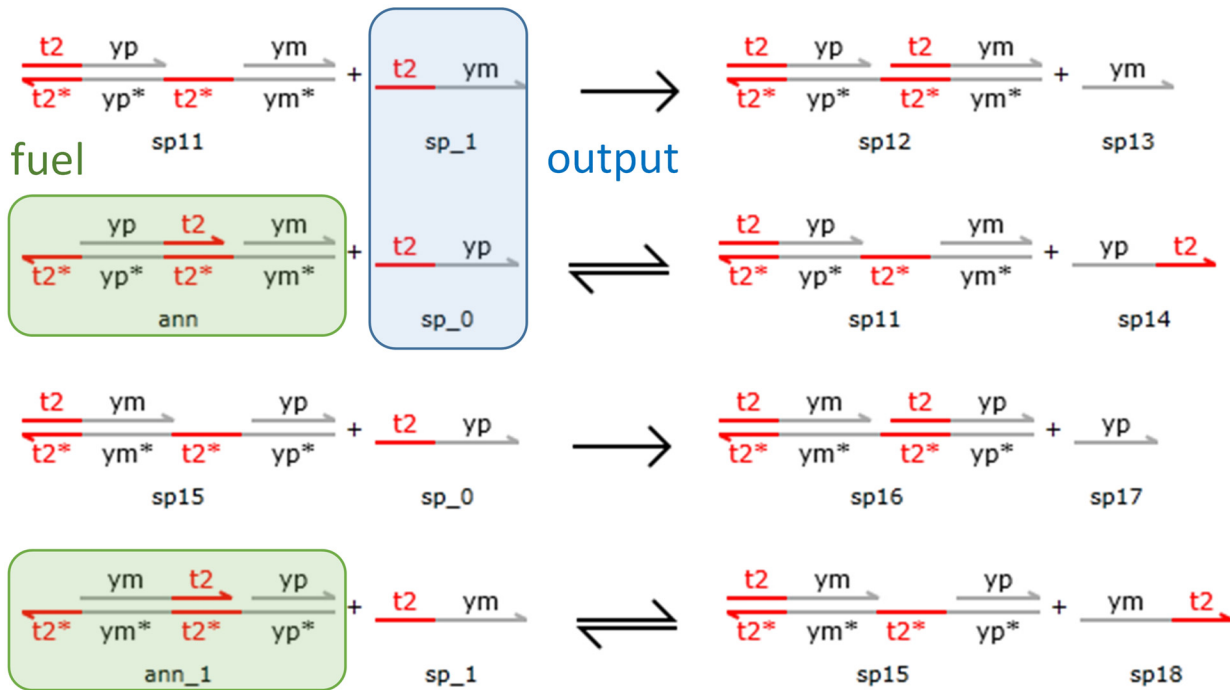


Fig 8. DSD realization of the annihilation reaction. Here, $|t_2, ym\rangle$ and $|t_2, yp\rangle$ correspond to the output species y^+ and y^- , respectively. Two fuels (ann and ann_1) with similar structures are used to ensure that the consumption rates of y^+ and y^- are balanced.

doi:10.1371/journal.pone.0137442.g008

respectively. These species correspond to the output species of the primitive components. Two fuels with similar structures are used to make the entire DSD reaction symmetric and balance the consumption rates of $|t_2, yp\rangle$ and $|t_2, ym\rangle$. As shown in Fig 8, the fuel ann first reacts with $|t_2, yp\rangle$ and then the product $sp11$ reacts with $|t_2, ym\rangle$. In contrast, the fuel ann_1 first reacts with $|t_2, ym\rangle$ followed by the reaction between the product $sp15$ and $|t_2, yp\rangle$. Therefore, the consumption rates of $|t_2, yp\rangle$ and $|t_2, ym\rangle$ are balanced and achieve effective annihilation.

The above DSD reactions for catalysis, degradation, and annihilation form a set of basic elements that enable the construction of primitive components and any proper linear system. We built the proposed configurable primitive components of integration and weighted summation with DSD reactions. The simulation results are shown in Fig 9. Fig 9A shows the output response of the integration component computing $y = \int_0^t u(\tau) d\tau$ with respect to fixed inputs $u^+ = 2$ nM and $u^- = 1$ nM. The concentration of y^- (green curve) remains at approximately 8 nM due to the effect of the annihilation reaction [Eq (3)], while the concentration of y^+ (blue curve) grows to 105 nM after 100 seconds. As expected, the output signal $y = y^+ - y^-$ grows linearly and reaches 97 nM after 100 seconds. Fig 9B shows the simulation results of the weighted summation component computing $y = 2u_1 + u_2$. Two sets of inputs $(u_1^+, u_1^-) = (5, 3)$ and $(u_2^+, u_2^-) = (2, 1)$ in nM are observed. As expected, the component functions correctly with $y = 12 - 7 = 5$ (for $y^+ = 2 \times 5 + 2 = 12$ and $y^- = 2 \times 3 + 1 = 7$) nM.

The above simulation results confirm the successful realization of the integration and weighted summation components using DSD reactions. Because good compatibility is observed among the blocks constructed using the two-domain DSD method, these primitive

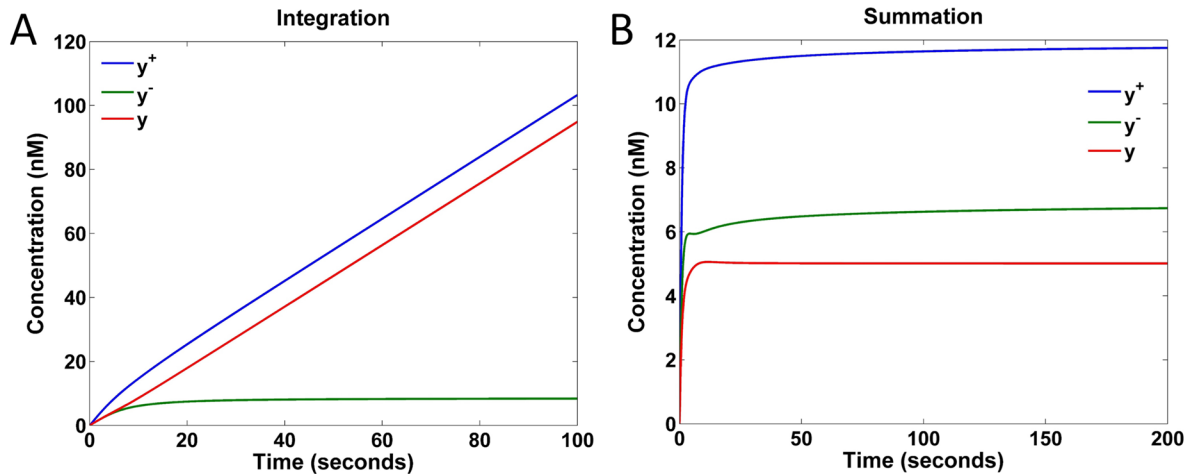


Fig 9. Simulation of DSD realized primitive components. (A) Concentrations of y^+ , y^- , and $y = y^+ - y^-$ of the integration component computing $y = \int_0^t u(\tau) d\tau$ under constant inputs $u^+ = 2$ and $u^- = 1$ nM. (B) Concentrations of y^+ , y^- , and $y = y^+ - y^-$ of the weighted summation component computing $y = 2u_1 + u_2$ under constant inputs $(u_1^+, u_1^-) = (5, 3)$ and $(u_2^+, u_2^-) = (2, 1)$ in nM.

doi:10.1371/journal.pone.0137442.g009

DSD reactions can be composed to realize sophisticated linear systems. Later we will show that any proper transfer functions can be realized using the DSD realized primitive components.

Implementation of Transfer Functions in Biochemistry

Transfer Function Decomposition. The transfer function $G(s)$ of an LTI system can always be written as

$$G = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}, \quad (4)$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of the output $y(t)$ and input $u(t)$ signals, respectively. A transfer function is called **strictly proper**, as we shall assume, if $m < n$ (i.e., the degree of the numerator polynomial is smaller than the denominator polynomial).

Using the fundamental theorem of algebra, $U(s)$ can be rewritten as

$$U(s) = (s + \alpha_1)^{n_1} \dots (s + \alpha_l)^{n_l} (s^2 + \beta_1 s + \gamma_1)^{m_1} \dots (s^2 + \beta_p s + \gamma_p)^{m_p},$$

where α_i , β_i and γ_i are real numbers satisfying $\beta_i^2 - 4\gamma_i < 0$ for stable LTI systems, l and p are non-negative integers, and n_i and m_i are positive integers. By partial fraction expansion, Eq (4) can be factorized as

$$G(s) = \sum_{i=1}^l \sum_{j=1}^{n_i} \frac{A_{ij}}{(s + \alpha_i)^j} + \sum_{i=1}^p \sum_{j=1}^{m_i} \frac{B_{ij}s + C_{ij}}{(s^2 + \beta_i s + \gamma_i)^j},$$

where A_{ij} , B_{ij} , and C_{ij} are real coefficients. Because the terms of $\frac{1}{(s + \alpha_i)^j}$ and $\frac{1}{(s^2 + \beta_i s + \gamma_i)^j}$ can

be realized by cascading j times the elementary modules of $\frac{1}{(s + \alpha_i)}$ and $\frac{1}{(s^2 + \beta_i s + \gamma_i)}$,

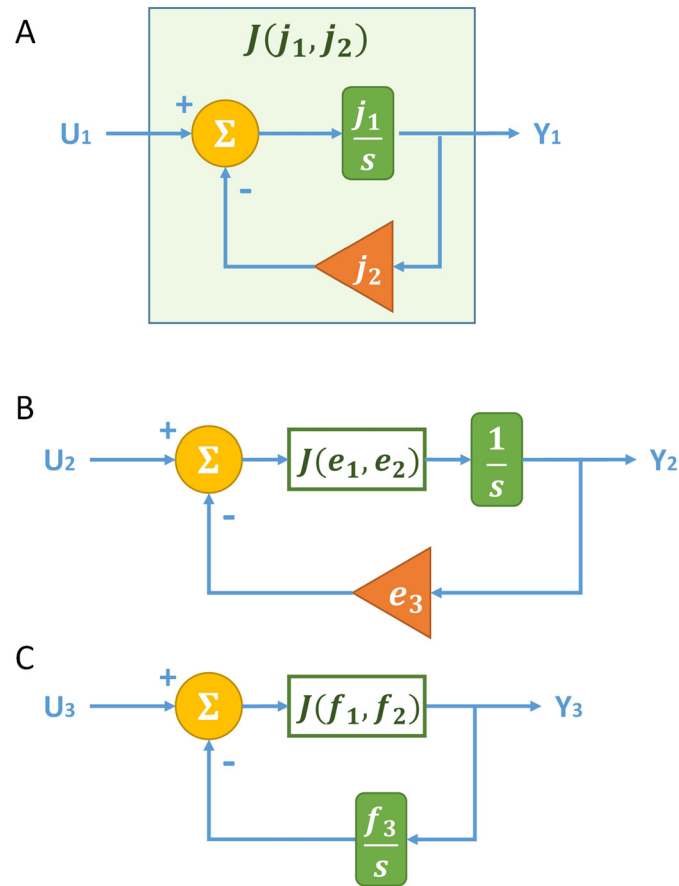


Fig 10. Block diagrams of naive implementation of elementary modules. (A) Block diagram of the degree-(1, 0) module with parametric weights j_1 and j_2 to match the transfer function coefficients. (B) Block diagram of the degree-(2, 0) module with parametric weights e_1 , e_2 , and e_3 to match the transfer function coefficients. (C) Block diagram of the degree-(2, 1) module with parametric weights f_1 , f_2 , and f_3 to match the transfer function coefficients.

doi:10.1371/journal.pone.0137442.g010

respectively, we only need to implement the set

$$\left\{ \frac{D_1}{s + \alpha}, \frac{D_2}{s^2 + \beta s + \gamma}, \frac{D_3 s}{s^2 + \beta s + \gamma} \right\}, \quad (5)$$

of elementary modules, where D_1 , D_2 and D_3 are real numbers. We refer to the three elementary modules as the “degree-(1, 0),” “degree-(2, 0),” and “degree-(2, 1)” modules, respectively, according to the degrees of their denominator and numerator polynomials.

Below we investigate how to implement these three elementary modules to construct any LTI system with CRNs. First, we provide a naive construction to illustrate inexactness, and then present a refined exact solution.

Naive Implementation of Elementary Modules. To build a degree-(1, 0) module $J(j_1, j_2)$ with parameters j_1, j_2 with real values, one might use the negative feedback loop consisting of a forward integration block of $\frac{j_1}{s}$ and a negative feedback of gain j_2 (as shown in Fig 10A). The

transfer function of this realization equals $\frac{j_1}{s + j_1 j_2}$, which exactly represents the target form.

Recall that the introduction of auxiliary species enables the configurability of primitive components, and thus the values of j_1 and j_2 can be set as desired.

To build a degree-(2, 0) module, one has to increase the degree of the denominator polynomial. Therefore, one might try to use a degree-(1, 0) module for a compositional construction (as shown in Fig 10B). Let the degree-(1, 0) module be $J(e_1, e_2)$ fed to a forward integration block of $\frac{1}{s}$, and let the negative feedback have gain e_3 . Then, the transfer function of the block diagram equals $\frac{e_1}{s^2 + e_1 e_2 s + e_1 e_3}$ as desired.

To build a degree-(2, 1) module, one might also try to use a degree-(1, 0) module for compositional construction (as shown in Fig 10C). Let the degree-(1, 0) module be $J(f_1, f_2)$ fed forward to the output, and place an integration block of $\frac{f_3}{s}$ on the negative feedback path. Then, the transfer function of the block diagram equals $\frac{f_1 s}{s^2 + f_1 f_2 s + f_1 f_3}$ as desired.

Although the block diagrams of Fig 10 yield the desired transfer functions of the elementary modules, they are problematic due to the imperfection of the weighted summation components realized by the chemical reactions (with transfer function $\sum_i \frac{k_i}{s + k_0}$ instead of $\sum_i \frac{k_i}{k_0}$ as discussed earlier in the primitive component construction section). The following example demonstrates that a system constructed in this fashion may deviate drastically from the specification and can even be unstable.

Consider the LTI system specified by transfer function

$$G = \frac{3s^4 + 19s^3 + 47s^2 + 51s + 22}{s^5 + 10s^4 + 36s^3 + 67s^2 + 66s + 30}. \quad (6)$$

By partial fraction expansion, G can be expressed as

$$G = \frac{1}{s^2 + 3s + 3} + \frac{s}{s^2 + 2s + 2} + \frac{2}{s + 5}. \quad (7)$$

By defining

$$\begin{cases} G_1 \equiv \frac{1}{s^2 + 3s + 3}, \\ G_2 \equiv \frac{s}{s^2 + 2s + 2}, \\ G_3 \equiv \frac{2}{s + 5}, \end{cases} \quad (8)$$

they correspond to the basic elements in Eq (5) and can be implemented with the block diagrams of Fig 10 by setting $(j_1, j_2) = (2, 2.5)$, $(e_1, e_2, e_3) = (1, 3, 3)$ and $(f_1, f_2, f_3) = (1, 2, 2)$. Assume that $k_0 = a$ for some positive real constant a for all weighted summation primitive components; then, assume that $k_i, i = 1, \dots, n$, of a weighted summation primitive component are determined according to the required weights. The transfer functions after the chemical reaction

realization become

$$\left\{ \begin{array}{l} G_1 \rightarrow \frac{a^2(s+a)}{s^2(s+a)^3 + 3a^2s(s+a) + 3a^3} \equiv G'_1, \\ G_2 \rightarrow \frac{a^2s}{s^2(s+a)^2 + 2a^2s + 2a^2} \equiv G'_2, \\ G_3 \rightarrow \frac{2a(s+a)}{s(s+a)^2 + 5a^2} \equiv G'_3. \end{array} \right. \quad (9)$$

Based on (asymptotic) stability analysis, the implemented system is unstable when a pole of the transfer functions (i.e., a root of the denominator polynomials) falls in the right-half of the s -plane. To avoid instability, the value of a has to be carefully determined. However, choosing a proper value for a can be tedious and sometimes even impossible.

The responses of G'_1 , G'_2 , and G'_3 to the step input with an amplitude of 10^{-6} are shown in Fig 11. Specifically, Fig 11A and 11B show the responses of G'_1 , C and D show the responses of G'_2 , and E and F show the responses of G'_3 . As seen from the plots of Fig 11A, 11C, and 11E, the responses are unstable under the chosen small a values, while for plots of B, D, and F, the responses are stable under the chosen large a values. However, even the stable responses may deviate from the ideal transfer function responses to some extent. When a in G'_1 , G'_2 and G'_3 increases to 6, the maximal concentrations of the auxiliary species used in each case are $6 \times 3/0.0249 \approx 720$, $6 \times 2/0.0249 \approx 480$, and $6 \times 2.5/0.0249 \approx 600$ nM, respectively, where the factors 3, 2 and 2.5 are the weights of the gain blocks in each system. As seen in Fig 11, the naive implementations cannot approximate the ideal responses well even under such high concentrations.

Exact Implementation of Elementary Modules. To prevent the non-ideal effect of the weighted summation components, we present a refined method that exactly implements the three elementary modules with chemical reactions.

The degree-(1, 0) module can be realized directly by the CRN transfer function of the primitive gain block. For the other two modules, we take advantage of the non-ideal form of the weighted summation block and devise their corresponding block diagrams, whose CRN transfer functions match the specification. The block diagrams, normal transfer functions, and CRN transfer functions for the three elementary modules are summarized in Fig 12. The CRN transfer function of the degree-(2, 0) module can be calculated from

$$\left\{ \begin{array}{l} X_1 = \frac{k_1}{s+k_0}U - \frac{k_2}{s+k_0}Y, \\ Y = \frac{k_3}{s}X_1. \end{array} \right. \quad (10)$$

Therefore, the CRN transfer function $\frac{Y}{U}$ equals

$$\frac{Y}{U} = \frac{k_1k_3}{s^2 + k_0s + k_2k_3}.$$

Note that if the gain and summation blocks are ideal, the normal transfer function should be $\frac{k_1k_3}{k_0s + k_2k_3}$. Similarly, the CRN transfer function of the degree-(2, 1) module can be derived as

$$\frac{Y}{U} = \frac{k_1s}{s^2 + k_0s + k_2k_3}.$$

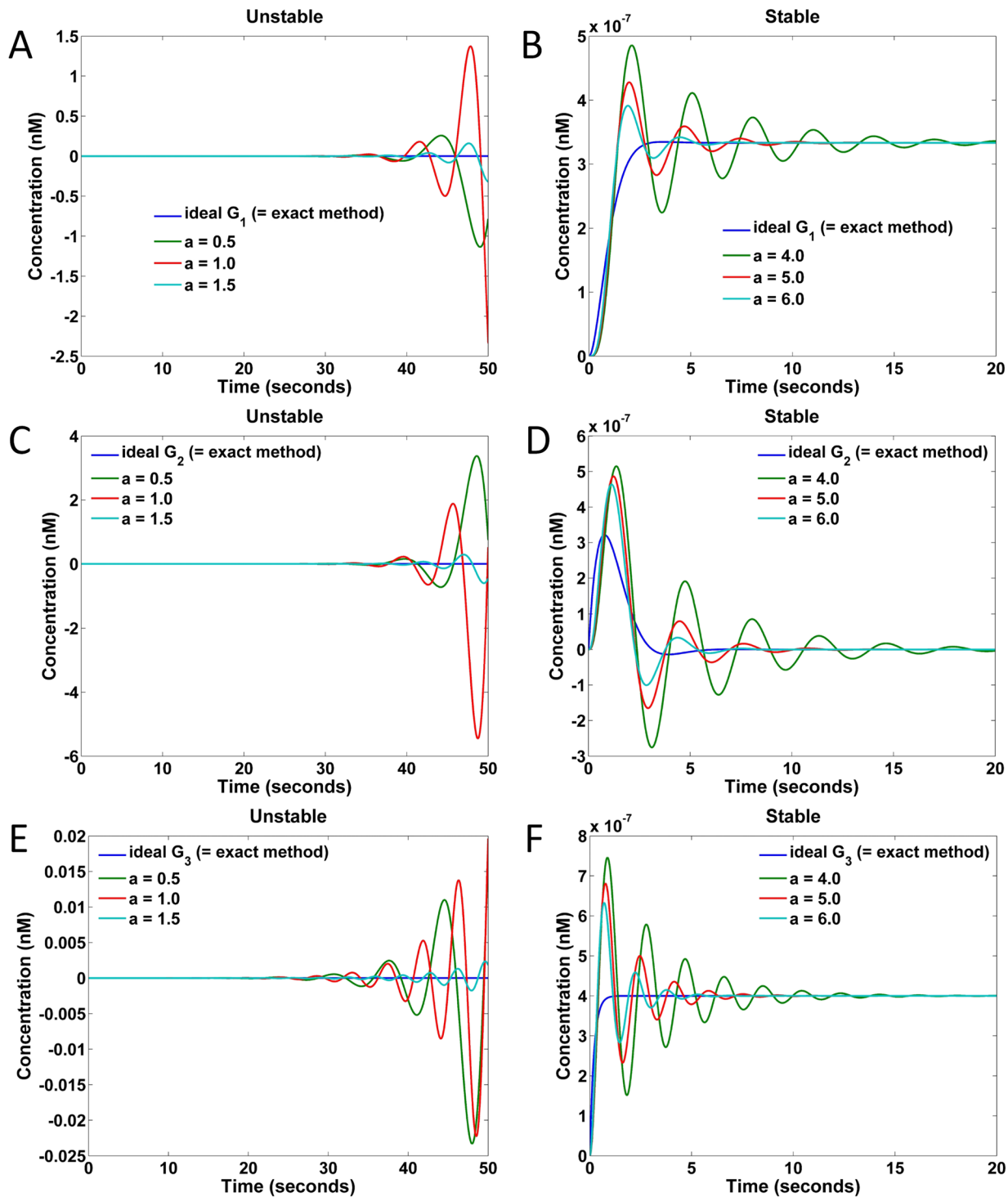


Fig 11. Responses of naive implementations to a step input of amplitude 10^{-6} nM under different values of parameter a . (A) Responses of G_1 under $a = 0.5, 1.0, 1.5$. (B) Responses of G_1 under $a = 4.0, 5.0, 6.0$. (C) Responses of G_2 under $a = 0.5, 1.0, 1.5$. (D) Responses of G_2 under $a = 4.0, 5.0, 6.0$. (E) Responses of G_3 under $a = 0.5, 1.0, 1.5$. (F) Responses of G_3 under $a = 4.0, 5.0, 6.0$.

doi:10.1371/journal.pone.0137442.g011

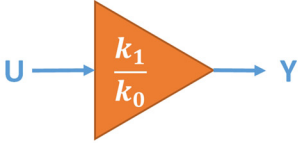
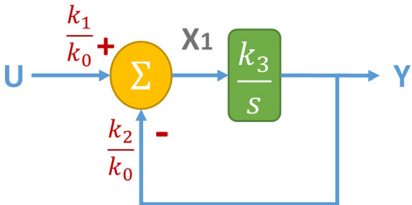
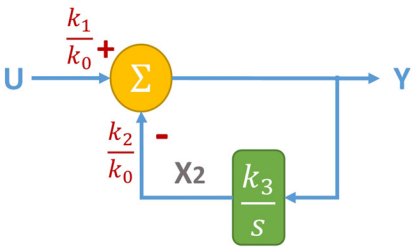
Block Diagram	Normal Transfer Function	CRN Transfer Function
	$\frac{k_1}{k_0}$	$\frac{k_1}{s + k_0}$
	$\frac{k_1 k_3}{k_0 s + k_2 k_3}$	$\frac{k_1 k_3}{s^2 + k_0 s + k_2 k_3}$
	$\frac{k_1 s}{k_0 s + k_2 k_3}$	$\frac{k_1 s}{s^2 + k_0 s + k_2 k_3}$

Fig 12. Block diagrams and their transfer functions for the exact implementation of the elementary modules. The first, second, and third rows in the table correspond to the implementations of the degree-(1, 0), degree-(2, 0), and degree-(2, 1) modules, respectively. The first, second, and third columns in the table show the block diagrams, normal transfer functions, and CRN transfer functions, respectively. A normal transfer function is obtained directly from its block diagram, whereas a CRN transfer function is obtained when the gain and summation blocks in the block diagram are realized using the gain and summation CRNs in Table 1.

doi:10.1371/journal.pone.0137442.g012

As a result, the responses can be made exact through the control of auxiliary species concentrations.

There are two special cases that need attention. First, if the coefficient β vanishes in the denominators of the transfer functions $\frac{D_2}{s^2 + \beta s + \gamma}$ and $\frac{D_3 s}{s^2 + \beta s + \gamma}$ of the degree-(2, 0) and degree-(2, 1) modules, respectively, we have to make $k_0 = 0$ in the CRN transfer functions.

Therefore, we need to remove the degradation reactions $z^\pm + y^\pm \xrightarrow{r_0^\pm} z^\pm$ from the summation blocks. Second, for the negative parameter k_1 or k_2 in Fig 12, as mentioned previously the plus and minus signs of the superscripts of y in the pair of catalytic reactions of the weighted summation CRN should be swapped; for negative k_3 , the superscript signs of y in the pair of catalytic reactions of the integration CRN should be swapped. Notice that k_0 can always be non-negative for stable linear systems, whose poles should all be on the left-half of the s -plane. Therefore, with these modifications our method is sufficient to implement any stable linear system.

DSD Realization of Transfer Functions

We map the naive and exact implementations of the transfer function specification of Eq (6) into DSD reactions for comparison. In the simulation, we assume the rate constants $r_0^\pm = 0.05 \text{ nM}^{-1} \text{ s}^{-1}$ and $r_i^\pm = 0.0249 \text{ nM}^{-1} \text{ s}^{-1}$ for $i = 1, \dots, n$.

The responses of DSD that realized naive implementation of the transfer function specification are shown in Fig 13. Specifically, Fig 13A and 13B show the results realizing the transfer function G_1 in Eq (8), C and D show those realizing G_2 , and E and F show those realizing G_3 . In the plots of Fig 13A, 13C, and 13E, the realized systems are unstable under the chosen small a values, while in the plots of B, D, and F the realized systems are stable under the chosen large a values. These results are consistent with the CRN responses of Eq (9) in Fig 11.

The maximum k_i 's in the DSD realizations of G'_1 , G'_2 , and G'_3 are $6.0 \times 3 = 18.0$, $6.0 \times 2 = 12.0$, and $6.0 \times 2.5 = 15.0$, respectively. Because the rate constants r_i^\pm , $i = 1, \dots, n$ are set to $0.0249 \text{ nM}^{-1} \text{ s}^{-1}$, the corresponding concentrations of the catalysts x^\pm in the gain and summation blocks in Fig 10 of G'_1 , G'_2 , and G'_3 are approximately $18.0/0.0249 \approx 720$, $12.0/0.0249 \approx 480$, and $15.0/0.0249 \approx 600 \text{ nM}$, respectively. Not only are such high concentrations undesirable, but it is also difficult to find a proper a value to fit the ideal response.

To compare the naive and exact implementations of transfer function specifications under DSD realization, the responses of the targeted transfer functions G_1 , G_2 and G_3 are plotted in Fig 14A, 14B, and 14C, respectively. The red curves in Fig 14A, 14B, and 14C correspond to the naive implementations with $a = 6$; the involved concentrations of catalysts are approximately 720, 480, and 600 nM, respectively. The species with the largest concentrations used in the exact implementations of G_1 and G_2 are both the catalysts x^\pm in the integration CRNs. Because the weights of the integration blocks in the exact implementations of G_1 and G_2 are both equal to 2.0, the corresponding concentrations of x^\pm in the catalytic reactions are approximately 80 nM. In contrast, in the exact implementation of G_3 the catalysts z^\pm in the gain block (where k_0 equals 5.0) have the highest concentration ($5.0/0.05 = 100 \text{ nM}$). Compared to the naive implementations involving concentrations as large as 720 nM, the exact implementations require much lower concentrations, and yet fit the ideal responses much better.

For the overall transfer function $G = G_1 + G_2 + G_3$, the responses of the DSD reactions realized via the naive and exact implementations are compared in Fig 15 against the ideal response. There are two approaches combining the subsystems G_1 , G_2 , and G_3 . One approach is to use a summation block to sum up the outputs Y_1 , Y_2 , and Y_3 of G_1 , G_2 , and G_3 , respectively; the other approach is to keep Y_1 , Y_2 , and Y_3 intact without summation by assuming that the output species of Y_1 , Y_2 , and Y_3 share the same piece of the DNA segment (rather than the entire DNA segment) that corresponds to the intended final product. The result of the former approach is shown in Fig 15A and the latter approach is shown in Fig 15B, which corresponds to the direct superposition of individual G_1 , G_2 , and G_3 responses. We observe that summing Y_1 , Y_2 , and Y_3 with an additional summation block may result in slight distortions in comparison with direct superposition. In either case, the exact implementation fits the ideal response much better than the naive implementation.

Discussion

Rate Matching and Configurability

The CRN of the integration block proposed by Oishi and Klavins contains the catalytic reactions $u^\pm \xrightarrow{q^\pm} u^\pm + y^\pm$. The integration is made possible under the assumption that $q_c^+ = q_c^- \equiv w_{\text{int}}$. Accordingly $\dot{y} = w_{\text{int}} \dot{u}$ (i.e., output y is a weighted integration of input u). However, the CRNs of the gain block contain one pair of additional degradation reactions

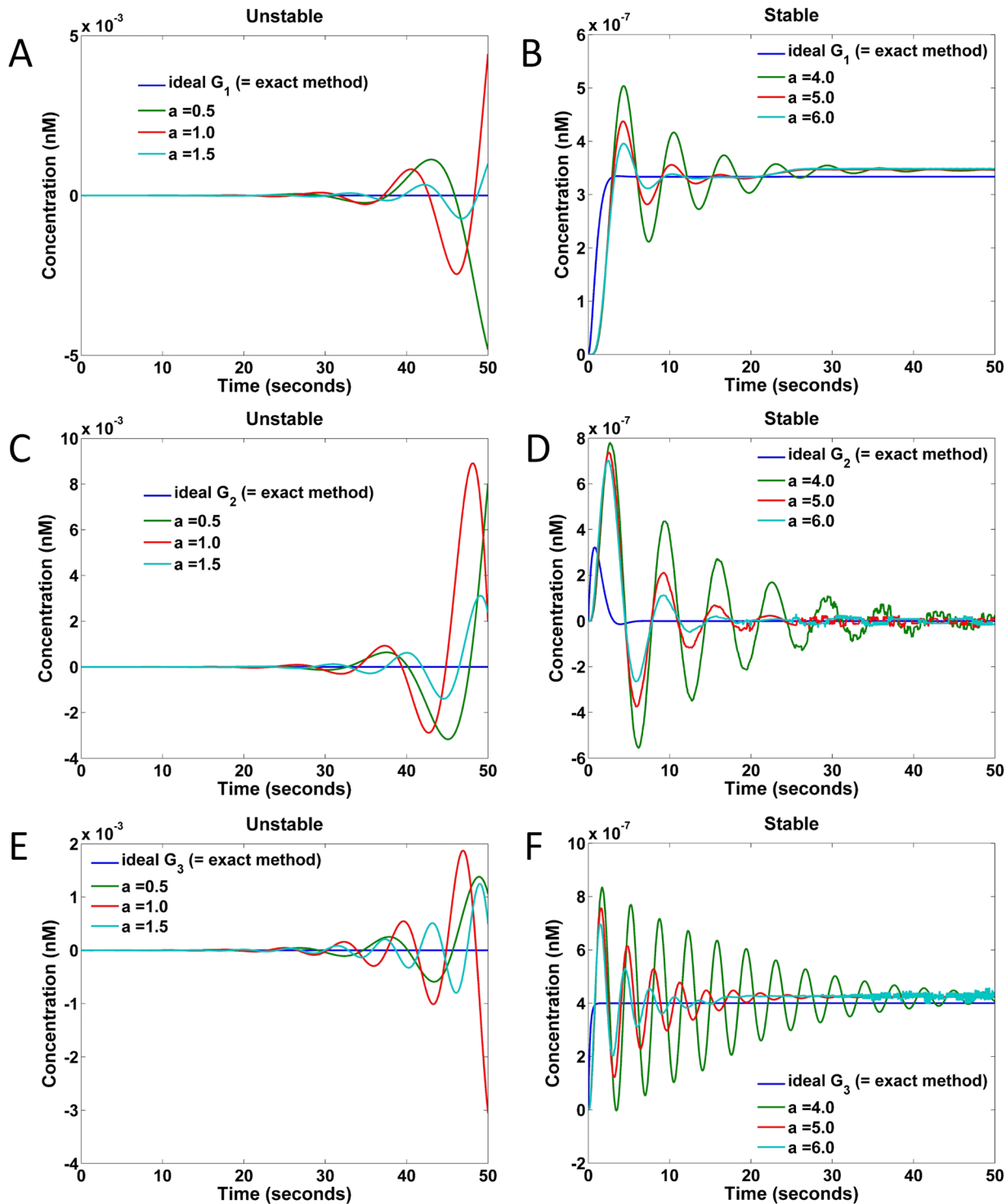


Fig 13. Responses of DSD realized naive implementations to a step input of amplitude 10^{-6} nM under different values of parameter a . (A) Responses of G_1 under $a = 0.5, 1.0, 1.5$. (B) Responses of G_1 under $a = 4.0, 5.0, 6.0$. (C) Responses of G_2 under $a = 0.5, 1.0, 1.5$. (D) Responses of G_2 under $a = 4.0, 5.0, 6.0$. (E) Responses of G_3 under $a = 0.5, 1.0, 1.5$. (F) Responses of G_3 under $a = 4.0, 5.0, 6.0$.

doi:10.1371/journal.pone.0137442.g013

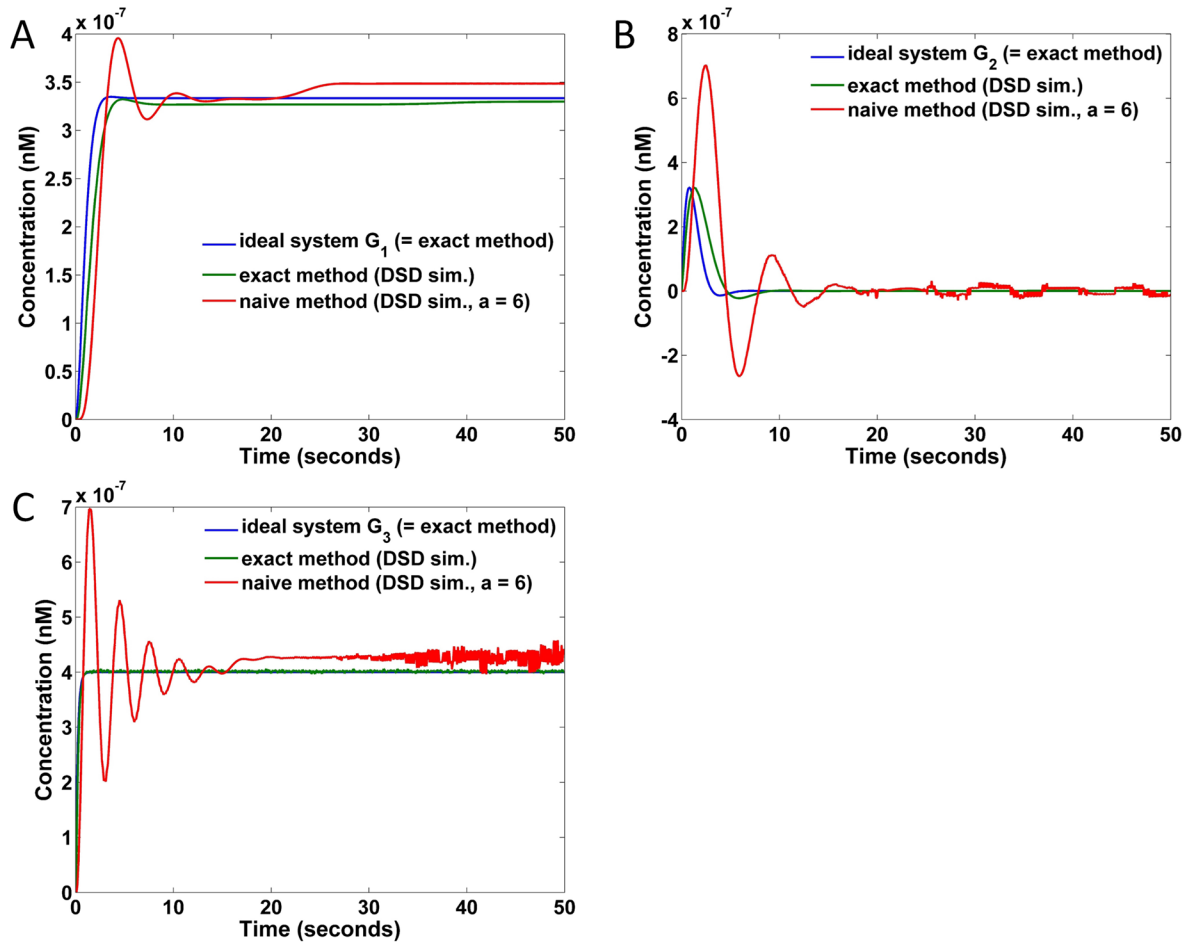


Fig 14. Response comparisons on DSD realized systems under the naive and exact implementations of transfer functions G_1 , G_2 , and G_3 . (A) Responses of G_1 implementations. (B) Responses of G_2 implementations. (C) Responses of G_3 implementations. The curves in red corresponds to naive implementations (under the parameter setting $a = 6$ in G_i), while the curves in green correspond to exact implementations.

doi:10.1371/journal.pone.0137442.g014

$y^\pm \xrightarrow{q_d^\pm} \emptyset$. The gain function with weight w_{gn} is realized by assuming $\frac{q_c^+}{q_d^+} = \frac{q_c^-}{q_d^-} = w_{gn}$. Thus, not only $q_c^+ = q_c^-$ but also $q_d^+ = q_d^- = q_c^+ / w_{gn}$ must be satisfied.

To cope with the problem of matching q_c^+ and q_c^- in the integration CRN, Oishi and Klavins provided a solution by maintaining a large rate constant q_{ua} in the annihilation reaction $u^+ + u^- \xrightarrow{q_{ua}} \emptyset$ so that q_c^+ and q_c^- do not need to be exactly the same, as long as they are close enough to w_{int} . The reason is that one of the concentrations of u^+ and u^- would approach zero under fast annihilation, and thus \dot{y} equals either $w_{int} u^+$ or $w_{int} u^-$. Because q_c^+ and q_c^- are close to w_{int} , the integration can be reasonably approximated. Similarly, the matching problem in the gain CRN is overcome by asserting not only a large q_{ua} , but also a large q_{ya} in $y^+ + y^- \xrightarrow{q_{ya}} \emptyset$. Thus, by requiring that q_c^\pm are close enough and the ratios $\frac{q_c^+}{q_d^+}$ and $\frac{q_c^-}{q_d^-}$ are close to w_{gn} , the gain function can be well approximated.

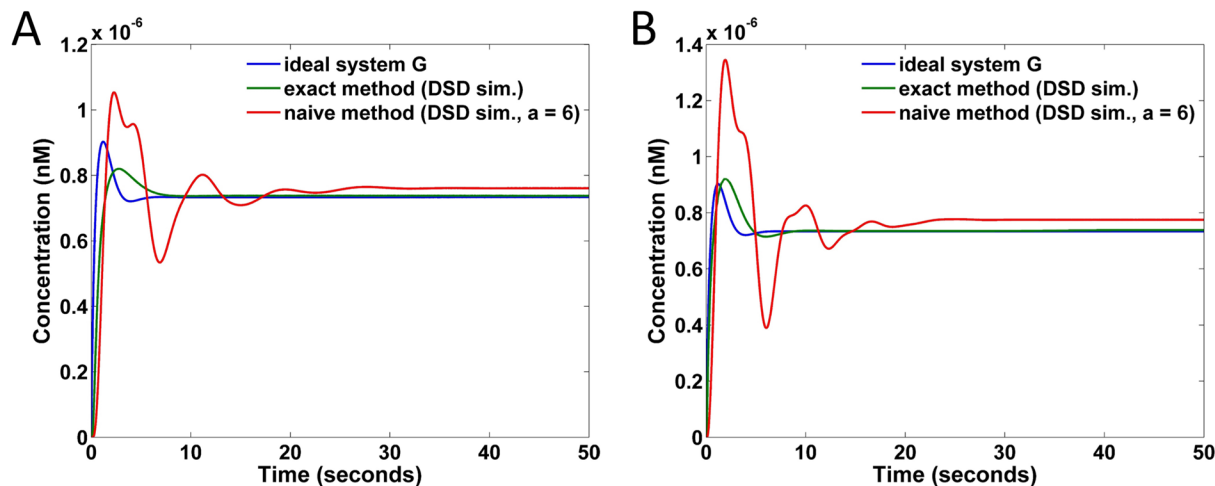


Fig 15. Response comparisons on DSD realized systems for the naive and exact implementations of transfer function G. (A) G implemented by a summation block adding G_1 , G_2 , and G_3 , assuming $a = 6$ for this summation block in both the naive and exact methods. (B) G implemented by superposing G_1 , G_2 , and G_3 , assuming that the DSD output species of G_1 , G_2 , and G_3 have a common sub-domain that is recognized as the output of G.

doi:10.1371/journal.pone.0137442.g015

However, the above assumptions of fast annihilations $u^+ + u^- \xrightarrow{q_{ua}} \emptyset$ (and $y^+ + y^- \xrightarrow{q_{ya}} \emptyset$) may not be easily satisfied, especially when these reactions are achieved by composing multiple reactions (e.g., in the DSD realization). Two undesirable situations might occur in composite reactions even if the rate constant q_{ua} is high. One is that u^+ and u^- both degrade, but the species with the lower concentration converges to a non-zero concentration value due to the restoration of the reactants by other reactions. The other is that although the species with the lower concentration vanishes, the remaining species converges to a concentration value less than $|u^+ - u^-|$ because some fuels can react with the remaining species. The approximations may be unsound when these situations occur. Although the effect of inappropriate assumptions might not be serious for a single primitive component, for a complex system composed of many primitive components the approximation would be crude due to error accumulations.

In contrast, because our primitive components are configurable due to the addition of the auxiliary species, we can match not only the reaction rates but also the required weights by tuning the concentrations of auxiliary species. Another advantage provided by the use of auxiliary species is the ability for dynamic system adaptation. As discussed in the case study of the PI-controller, the parameters of a PI-controller may be modified by simply tuning the concentrations of auxiliary species to react to environmental changes. Thus, by using auxiliary species we may fulfill a new system with the same set of reactions by tuning only the concentrations of the auxiliary species. In contrast, without auxiliary species the system would have to be redesigned by finding new catalysts and degradations to match the new weights and rate constants.

Transfer Function Decomposition

A transfer function can be decomposed in many different ways. Two common choices are through parallel decomposition ($G = G_1 + G_2$) and serial decomposition ($G = G_1 \cdot G_2$). In this paper, we adopt the former for the following reasons. First, given a strictly proper transfer function it is not always possible to decompose it as a product of the strictly proper transfer functions. For instance, the transfer function $\frac{2s^2 + 3s + 4}{(s + 1)(s + 2)(s + 3)}$ cannot be decomposed serially

into a system involving the cascade of two strictly proper subsystems. Because only proper systems where the degree of the numerator polynomial does not exceed that of the denominator polynomial are physically realizable, the only hope for implementing the system physically is to have a product of bi-proper subsystems, where the degrees of the numerator and denominator polynomials are the same. For example, it can be decomposed as $\frac{2s^2 + 3s + 4}{(s + 1)(s + 2)} \cdot \frac{1}{(s + 3)}$. However, if we implement the system with CRNs or the primitives introduced previously, the bi-proper subsystem might be an approximation due to $\frac{2s^2 + 3s + 4}{(s + 1)(s + 2)} = 2 - \frac{3s}{s^2 + 3s + 2}$, whose realization might involve a gain and a summation block, both of which are approximations. Second, by cascading subsystems using serial decomposition a system may be susceptible to error accumulation due to realization approximation. In contrast, parallel decomposition avoids the issue of error accumulation and amplification. Third, a strictly proper transfer function can always be decomposed as a summation of strictly proper transfer functions, which can be modeled by the three elementary modules described in Eq (5). Furthermore, we show that the three elementary modules allow exact CRN implementation using our configurable primitive components. For these reasons, parallel decomposition may be preferable to serial decomposition.

Nevertheless, applying parallel decomposition exclusively may not always be the best choice. The reasons are twofold. First, parallel decomposition requires one final summation block to sum up all elementary modules. This summation block, when it is implemented with a CRN, is approximative. Second, a hybrid decomposition strategy may achieve more effective implementation than applying a parallel decomposition. For instance, consider the system of transfer function $T = \frac{3s^2 + 8s + 6}{(s + 1)(s + 2)(s^2 + 2s + 2)}$. If only parallel decomposition is applied, then T is decomposed into $\frac{2}{s^2 + 2s + 2} + \frac{1}{s + 1} - \frac{1}{s + 2}$. Nevertheless, if serial decomposition is also exploited, then $\frac{1}{s + 1} - \frac{1}{s + 2}$ can be rewritten as $\frac{1}{(s + 1)(s + 2)}$, which can be implemented by cascading two gain blocks with $(k_0, k_1) = (1, 1)$ and $(2, 1)$, respectively. Therefore, with this rewriting a final summation block only needs to sum up the outputs of the two subsystems rather than three subsystems. Furthermore, the hybrid parallel and serial decomposition strategy may be more cost effective than using parallel decomposition alone. Future work should exploit a good hybrid decomposition strategy and remove the approximative effect due to the final summation block in parallel decomposition.

DSD Reaction Rates

In this paper, we realize all CRNs with DSD reactions. Typically, the rate constants in DSD reactions are approximately $10^{-3} \text{ nM}^{-1} \text{ s}^{-1}$ [33]. For example, if one requires the parameter $k_0 = r_0^\pm z^\pm$ in the degradation reaction to be 100 s^{-1} , then the concentrations of z^\pm should be of magnitude 10^5 in nM. However, such a high concentration might be impractical. To alleviate this high concentration requirement, one may try to increase the DSD reaction rates. Zhang *et al.* constructed and characterized DNA catalytic circuits driven by entropic gains [28]. Based on the entropy effects, a variant called the tethered entropy driven catalytic circuits was introduced [35, 38] to shorten the catalytic cycle and improve the reaction kinetics using localized hybridization reactions, which are achieved by tethering the key species in DSD reactions to increase their local concentrations. The influence of tethering is effectively presented in a speed-up factor λ , whose value can be feasibly reach approximately 10^5 [35]. Thus, the toehold

binding rate constants can be scaled up to $\lambda \times 5 \times 10^{-5} \text{ nM}^{-1} \text{ s}^{-1}$, where $5 \times 10^{-5} \text{ nM}^{-1} \text{ s}^{-1}$ is the short toehold binding rate constant in diffusion-based systems [6].

In our DSD simulation, we assume the factor $\lambda = 1000$ is available to increase the toehold binding rate constants to approximately $0.05 \text{ nM}^{-1} \text{ s}^{-1}$. Effectively, the concentrations of the auxiliary species can be reduced by a factor of 25. Under the toehold binding rate increase, a catalysis has a rate constant approximately $0.025 \text{ nM}^{-1} \text{ s}^{-1}$ in our interested range of concentrations under the intended system operation. Thus, the kinetics of the products y^\pm will now become $y^\pm = 0.025x^\pm u^\pm$. If x^\pm could be tuned to 1000 nM, then we can have an integration block with weight as high as 25.

Although we have reduced the required concentrations of auxiliary species using the speed-up factor λ , there are still some limitations. First, if β is set to 500 in the elementary modules $\frac{D_2}{s^2 + \beta s + \gamma}$ and $\frac{D_3 s}{s^2 + \beta s + \gamma}$ shown in Fig 12, then we have to prepare the catalysts z^\pm with large concentrations (approximately $500/0.05 = 10000 \text{ nM}$ due to the relation $\beta = r_0^+ z^+ = r_0^- z^-$). In contrast, if $D_3 = 500 = r_1^+ x^+ = r_1^- x^-$, then the concentrations of x^\pm would be $500/0.025 = 20000 \text{ nM}$. Thus, although we have reduced the concentrations of the auxiliary species to increase the range of coefficients in the transfer functions, further reduction may be needed.

Fuel Supply

DSD is a relatively mature technology for reliable *in vitro* construction of biochemical systems, and is more practical than *in vivo* experiments. However, one of the main obstacles for practical DSD experiment is to provide constant supply of fuels [37]. This paper assumes the fuels are large in quantity and ignore the situations where the fuels become deficient. In reality, the fuels are converted to inert waste and one has to continually provide the fuels to the system. Providing stable fuel supply can be challenging in a cellular context. Therefore, it is easier to implement DSD reactions *in vitro* instead of *in vivo*.

Author Contributions

Conceived and designed the experiments: TYC RYH JHJ. Performed the experiments: TYC. Analyzed the data: TYC HJC RYH JHJ FF. Contributed reagents/materials/analysis tools: TYC HJC RYH JHJ FF. Wrote the paper: TYC HJC RYH JHJ FF.

References

1. Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. *Nature*. 2000; 403:335–338. doi: [10.1038/35002125](https://doi.org/10.1038/35002125) PMID: [10659856](https://pubmed.ncbi.nlm.nih.gov/10659856/)
2. Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*. 2000; 403:339–342. doi: [10.1038/35002131](https://doi.org/10.1038/35002131) PMID: [10659857](https://pubmed.ncbi.nlm.nih.gov/10659857/)
3. Tamsir A, Tabor JJ, Voigt CA. Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. *Nature*. 2011; 469:212–215. doi: [10.1038/nature09565](https://doi.org/10.1038/nature09565) PMID: [21150903](https://pubmed.ncbi.nlm.nih.gov/21150903/)
4. Sohka T, Heins RA, Phelan RM, Greisler JM, Townsend CA, Ostermeier M. An externally tunable bacterial band-pass filter. *Proc Natl Acad Sci USA*. 2009; 106:10135–10140. doi: [10.1073/pnas.0901246106](https://doi.org/10.1073/pnas.0901246106) PMID: [19502423](https://pubmed.ncbi.nlm.nih.gov/19502423/)
5. Daniel R, Rubens JR, Sarpeshkar R, Lu TK. Synthetic analog computation in living cells. *Nature*. 2013; 497:619–623. doi: [10.1038/nature12148](https://doi.org/10.1038/nature12148) PMID: [23676681](https://pubmed.ncbi.nlm.nih.gov/23676681/)
6. Qian L, Winfree E. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*. 2011; 332:1196–1201. doi: [10.1126/science.1200520](https://doi.org/10.1126/science.1200520) PMID: [21636773](https://pubmed.ncbi.nlm.nih.gov/21636773/)
7. Qian L, Winfree E, Bruck J. Neural network computation with DNA strand displacement cascades. *Nature*. 2011; 475:368–372. doi: [10.1038/nature10262](https://doi.org/10.1038/nature10262) PMID: [21776082](https://pubmed.ncbi.nlm.nih.gov/21776082/)
8. Padirac A, Fujii T, Rondelez Y. Bottom-up construction of *in vitro* switchable memories. *Proc Natl Acad Sci USA*. 2012; 109:E3212–E3220. doi: [10.1073/pnas.1212069109](https://doi.org/10.1073/pnas.1212069109) PMID: [23112180](https://pubmed.ncbi.nlm.nih.gov/23112180/)

9. Densmore D, Hassoun S. Design automation for synthetic biological systems. *IEEE Design & Test of Computers*. 2012; 29:7–20. doi: [10.1109/MDT.2012.2193370](https://doi.org/10.1109/MDT.2012.2193370)
10. Oishi K, Klavins E. Biomolecular implementation of linear I/O systems. *IET Syst Biol*. 2011; 5:252–260. doi: [10.1049/iet-syb.2010.0056](https://doi.org/10.1049/iet-syb.2010.0056) PMID: [21823756](https://pubmed.ncbi.nlm.nih.gov/21823756/)
11. Klonowski W. Simplifying principles for chemical and enzyme reaction kinetics. *Biophys Chem*. 1983; 18:73–87. doi: [10.1016/0301-4622\(83\)85001-7](https://doi.org/10.1016/0301-4622(83)85001-7) PMID: [6626688](https://pubmed.ncbi.nlm.nih.gov/6626688/)
12. Padirac A, Fujii T, Rondelez Y. Nucleic acids for the rational design of reaction circuits. *Curr Opin Biotechnol*. 2013; 24:575–580. doi: [10.1016/j.copbio.2012.11.011](https://doi.org/10.1016/j.copbio.2012.11.011) PMID: [23265857](https://pubmed.ncbi.nlm.nih.gov/23265857/)
13. Holloman WK, Wiegand R, Hoessli C, Radding CM. Uptake of homologous single-stranded fragments by superhelical DNA: A possible mechanism for initiation of genetic recombination. *Proc Natl Acad Sci USA*. 1975; 72:2394–2398. doi: [10.1073/pnas.72.6.2394](https://doi.org/10.1073/pnas.72.6.2394) PMID: [1094467](https://pubmed.ncbi.nlm.nih.gov/1094467/)
14. Green C, Tibbetts C. Reassociation rate limited displacement of DNA strands by branch migration. *Nucleic Acids Res*. 1981; 9:1905–1918. doi: [10.1093/nar/9.8.1905](https://doi.org/10.1093/nar/9.8.1905) PMID: [6264399](https://pubmed.ncbi.nlm.nih.gov/6264399/)
15. Robinson BH, Seeman NC. Simulation of double-stranded branch point migration. *Biophys J*. 1987; 51:611–626. doi: [10.1016/S0006-3495\(87\)83386-6](https://doi.org/10.1016/S0006-3495(87)83386-6) PMID: [3580487](https://pubmed.ncbi.nlm.nih.gov/3580487/)
16. Weinstock PH, Wetmur JG. Branch capture reactions: Effect of recipient structure. *Nucleic Acids Res*. 1990; 18:4207–4213. doi: [10.1093/nar/18.14.4207](https://doi.org/10.1093/nar/18.14.4207) PMID: [2198537](https://pubmed.ncbi.nlm.nih.gov/2198537/)
17. Panyutin IG, Hsieh P. The kinetics of spontaneous DNA branch migration. *Proc Natl Acad Sci USA*. 1994; 91:2021–2025. doi: [10.1073/pnas.91.6.2021](https://doi.org/10.1073/pnas.91.6.2021) PMID: [8134343](https://pubmed.ncbi.nlm.nih.gov/8134343/)
18. Yurke B, Turberfield AJ, Mills AP, Simmel FC, Neumann JL. A DNA-fuelled molecular machine made of DNA. *Nature*. 2000; 406:605–608. doi: [10.1038/35020524](https://doi.org/10.1038/35020524) PMID: [10949296](https://pubmed.ncbi.nlm.nih.gov/10949296/)
19. Yurke B, Mills AP. Using DNA to power nanostructures. *Genetic Programming and Evolvable Machines*. 2003; 4:111–122. doi: [10.1023/A:1023928811651](https://doi.org/10.1023/A:1023928811651)
20. Li Q, Luan G, Guo Q, Liang J. A new class of homogeneous nucleic acid probe based on specific displacement hybridization. *Nucl Acids Res*. 2002; 30:e5. doi: [10.1093/nar/30.2.e5](https://doi.org/10.1093/nar/30.2.e5) PMID: [11788731](https://pubmed.ncbi.nlm.nih.gov/11788731/)
21. Zhang DY, Winfree E. Control of DNA strand displacement kinetics using toehold exchange. *J Am Chem Soc*. 2009; 131:17303–17314. doi: [10.1021/ja906987s](https://doi.org/10.1021/ja906987s) PMID: [19894722](https://pubmed.ncbi.nlm.nih.gov/19894722/)
22. Gyorgy A, Del Vecchio D. Modular composition of gene transcription networks. *PLoS Computational Biology*. 2014; 10:e1003486. doi: [10.1371/journal.pcbi.1003486](https://doi.org/10.1371/journal.pcbi.1003486) PMID: [24626132](https://pubmed.ncbi.nlm.nih.gov/24626132/)
23. Cardelli L. Strand algebras for DNA computing. In: Deaton R, Suyama A, editors. *DNA Computing and Molecular Programming*. vol. 5877 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg; 2009. p. 12–24. Available from: http://dx.doi.org/10.1007/978-3-642-10604-0_2
24. Soloveichik D, Seelig G, Winfree E. DNA as a universal substrate for chemical kinetics. *Proc Natl Acad Sci USA*. 2010; 107:5393–5398. doi: [10.1073/pnas.0909380107](https://doi.org/10.1073/pnas.0909380107) PMID: [20203007](https://pubmed.ncbi.nlm.nih.gov/20203007/)
25. Cardelli L. Two-domain DNA strand displacement. *Mathematical Structures in Computer Science*. 2013; 23:247–271. doi: [10.1017/S0960129512000102](https://doi.org/10.1017/S0960129512000102)
26. Green SJ, Lubrich D, Turberfield AJ. DNA hairpins: Fuel for autonomous DNA devices. *Biophysical Journal*. 2006; 91:2966–2975. doi: [10.1529/biophysj.106.084681](https://doi.org/10.1529/biophysj.106.084681) PMID: [16861269](https://pubmed.ncbi.nlm.nih.gov/16861269/)
27. Seelig G, Soloveichik D, Zhang DY, Winfree E. Enzyme-free nucleic acid logic circuits. *Science*. 2006; 314:1585–1588. doi: [10.1126/science.1132493](https://doi.org/10.1126/science.1132493) PMID: [17158324](https://pubmed.ncbi.nlm.nih.gov/17158324/)
28. Zhang DY, Turberfield AJ, Yurke B, Winfree E. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*. 2007; 318:1121–1125. doi: [10.1126/science.1148532](https://doi.org/10.1126/science.1148532) PMID: [18006742](https://pubmed.ncbi.nlm.nih.gov/18006742/)
29. Bath J, Turberfield AJ. DNA nanomachines. *Nat Nanotechnol*. 2007; 2:275–284. doi: [10.1038/nnano.2007.104](https://doi.org/10.1038/nnano.2007.104) PMID: [18654284](https://pubmed.ncbi.nlm.nih.gov/18654284/)
30. Amir Y, Ben-Ishay E, Levner D, Ittah S, Abu-Horowitz A, Bachelet I. Universal computing by DNA origami robots in a living animal. *Nat Nanotechnol*. 2014; 9:353–357. doi: [10.1038/nnano.2014.58](https://doi.org/10.1038/nnano.2014.58) PMID: [24705510](https://pubmed.ncbi.nlm.nih.gov/24705510/)
31. Phillips A, Cardelli L. A programming language for composable DNA circuits. *J R Soc Interface*. 2009; 6:419–436. doi: [10.1098/rsif.2009.0072.focus](https://doi.org/10.1098/rsif.2009.0072.focus)
32. Lakin MR, Youssef S, Cardelli L, Phillips A. Abstractions for DNA circuit design. *J R Soc Interface*. 2012; 9:470–486. doi: [10.1098/rsif.2011.0343](https://doi.org/10.1098/rsif.2011.0343) PMID: [21775321](https://pubmed.ncbi.nlm.nih.gov/21775321/)
33. Zhang DY, Seelig G. Dynamic DNA nanotechnology using strand-displacement reactions. *Nat Chem*. 2011; 3:103–113. doi: [10.1038/nchem.957](https://doi.org/10.1038/nchem.957) PMID: [21258382](https://pubmed.ncbi.nlm.nih.gov/21258382/)
34. Lakin MR, Youssef S, Polo F, Emmott S, Phillips A. Visual DSD: A design and analysis tool for DNA strand displacement systems. *Bioinformatics*. 2011; 27:3211–3213. doi: [10.1093/bioinformatics/btr543](https://doi.org/10.1093/bioinformatics/btr543) PMID: [21984756](https://pubmed.ncbi.nlm.nih.gov/21984756/)

35. Chandran H, Gopalkrishnan N, Phillips A, Reif J. Localized Hybridization Circuits. In: Cardelli L, Shih W, editors. *DNA Computing and Molecular Programming*. vol. 6937 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg; 2011. p. 64–83. Available from: http://dx.doi.org/10.1007/978-3-642-23638-9_8
36. Lakin MR, Parker D, Cardelli L, Kwiatkowska M, Phillips A. Design and analysis of DNA strand displacement devices using probabilistic model checking. *J R Soc Interface*. 2012; 9:1470–1485. doi: [10.1098/rsif.2011.0800](https://doi.org/10.1098/rsif.2011.0800) PMID: [22219398](https://pubmed.ncbi.nlm.nih.gov/22219398/)
37. Yordanov B, Kim J, Petersen RL, Shudy A, Kulkarni VV, Phillips A. Computational design of nucleic acid feedback control circuits. *ACS Synth Biol*. 2014; 3:600–616. doi: [10.1021/sb400169s](https://doi.org/10.1021/sb400169s) PMID: [25061797](https://pubmed.ncbi.nlm.nih.gov/25061797/)
38. Gopalkrishnan N, Garg S, Chandran H, Reif J. Speeding up DNA circuits using localized hybridization; 2011. In *Foundations of Nanoscience Meeting (FNANO)*.