



Biomolecular implementation of linear I/O systems

K. Oishi E. Klavins

Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA 98195, USA
E-mail: koishi@uw.edu

Abstract: Linear I/O systems are a fundamental tool in systems theory, and have been used to design complex circuits and control systems in a variety of settings. Here we present a principled design method for implementing arbitrary linear I/O systems with biochemical reactions. This method relies on two levels of abstraction: first, an implementation of linear I/O systems using idealised chemical reactions, and second, an approximate implementation of the ideal chemical reactions with enzyme-free, entropy-driven DNA reactions. The ideal linear dynamics are shown to be closely approximated by the chemical reactions model and the DNA implementation. We illustrate the approach with integration, gain and summation as well as with the ubiquitous robust proportional-integral controller.

1 Introduction

Synthetic biochemical reaction networks with desired dynamic behaviours are difficult to design because of inherent non-linearities and substantial uncertainties in reaction mechanisms. Yet natural systems abound in which reliable behaviour is obtained from the composition of enormous numbers of molecular subsystems. It seems clear that some kind of design theory for biochemical reaction networks ought to be obtainable.

In engineering, the design of dynamic systems from unreliable or poorly modelled basic components is not a new problem. The field of control systems engineering is focused on the task of designing dynamic I/O systems that interact with and augment the behaviour of unknown or poorly modelled dynamic modules. In this paper, we explore the applicability of a standard design theory, linear I/O systems, to the design of predictable and robust synthetic biochemical systems. Linear I/O systems are present in almost any engineered device, from stereo amplifiers to aircraft autopilots, and can be implemented electronically, mechanically and in software. It has been shown that chemical reaction networks can approximate arbitrary polynomial ordinary differential equations [1]. However, prior work in this area lacked a practical, modular, design methodology for implementing a specified ODE, and did not address the implementation of more abstract I/O systems. In this paper, we show that linear I/O systems can also be implemented with chemical reactions and, in particular, with enzyme-free entropy-driven DNA reactions.

Broadly speaking, there exists two types of synthetic biochemical devices: *in vivo* and *in vitro*. A variety of dynamic devices have been constructed *in vivo*, including toggle switches [2], oscillators [3, 4] and band pass filters [5, 6]. However, these examples represent one-of-a-kind

systems and not general design methodologies for biochemical systems. The absence of predictive models for complex systems designed from modular components suggests synthetic biology *in vivo* may suffer from a lack of a large library of well-characterised parts. In contrast, systems synthesised from DNA *in vitro* [7–10] are becoming substantially more complex and reliable. With DNA, well-understood models of hybridisation and strand displacement are available to design and predict molecular interactions [11–13]. Recently, it was shown that any physically realistic abstract chemical reaction can be well-approximated by an appropriately designed DNA strand-displacement reaction [14]. Furthermore, similar systems have been demonstrated experimentally: DNA and RNA have been used to design a variety of devices including catalysts and amplifiers [9], logic gates [7] and detectors [15].

This paper is based on and inspired by the above results in designing predictable DNA devices, although in principle other biochemical implementations of linear I/O systems could be constructed. Specifically, we show that any linear I/O system can be built from the composition of three types of reactions, namely catalysis, degradation and annihilation. We then show how to implement these reactions with DNA devices, but of course, any other programmable system of molecules could also be used.

A biomolecular device can be abstracted to an I/O device taking an input signal, such as a time varying concentration of some chemical species, and producing an output signal. The I/O systems abstraction allows for the composition of devices into systems of interacting sub-systems: the output of one device is ‘wired’ to the input of another.

Formally, an I/O system is specified by an input space, output space, an internal state and a mapping that describes the output signal generated given an input signal and an initial internal state. Often an I/O system corresponds to a modular part of some physical system. In equations, we

write an I/O system in state space [16]

$$\dot{x} = f(x, u) \quad (1)$$

$$y = g(x, u) \quad (2)$$

where $u: \mathbb{R} \rightarrow \mathbb{R}^n$ is an input signal, $y: \mathbb{R} \rightarrow \mathbb{R}^m$ is an output signal, $x: \mathbb{R} \rightarrow \mathbb{R}^p$ is the internal state of the system and $n, m, p \in \mathbb{Z}^+$. Note that signals such as u are functions of time; however, for simplicity we will write u instead of $u(t)$.

I/O systems may be composed in parallel or in series, to obtain a composite I/O system, by combining output signals or by setting the output signal of one system to be the input signal for another system. Composition can be represented graphically in a block diagram by drawing a block for each subsystem and representing the connections with directed edges, as illustrated in Fig. 1a.

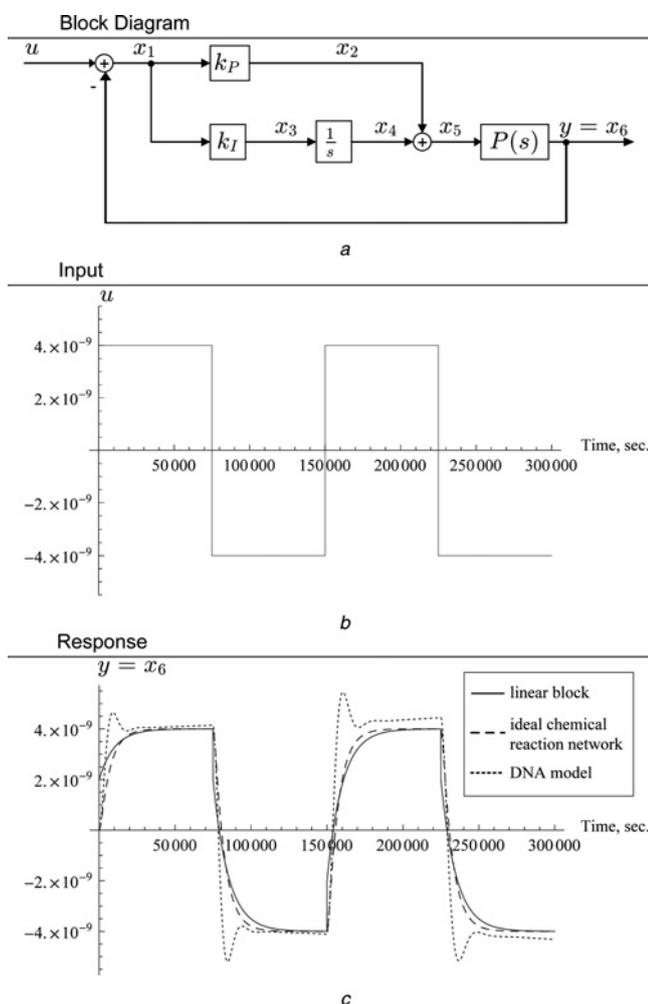


Fig. 1 PI controller block diagram and behaviour

a Block diagram, for a PI controller. The signal u is an input, y is an output signal and x_1, \dots, x_6 are internal signals. The negative sign next to the edge going into the left summation block means that the output of the summation is $x_1 = u - y$. The PI controller is a feedback system that tracks an input signal over a large class of plants $P(s)$. Here the plant $P(s)$ is implemented with reactions (29) and (30)

b Input signal driving the PI controller. The input signal u is a square wave

c Output trajectories for the ideal PI controller as well as the PI controller implemented with ideal chemical reactions and the DNA model. The steady-state error observed in the DNA model of the PI controller is a result of the sequestration of signal molecule y^\pm in intermediate reaction species involved in the left summation block

Of particular interest are controllers. A controller is an I/O system built around a particular system to be controlled (called the ‘plant’). For example, the controller illustrated in Fig. 1a is a proportional-integral (PI) controller built around the plant $P(s)$ with control input u and output y . Often the plant corresponds to a module with unknown or poorly modelled dynamics. The controller is designed to produce an output from the plant that is a desired function of the control input u . Controlled systems can be categorised into open-loop and closed-loop controllers. In an open-loop system the input to the plant does not depend on its output. In a closed-loop system (such as in Fig. 1a) the input to the plant is a function of its output, involving a comparison between the input to the controller and the output of the plant. Although open-loop systems are generally easier to analyse than closed-loop systems, feedback allows the engineer to design systems that are robust to modelling uncertainty and errors such as exogenous disturbances and retroactivity [17, 18].

The most well-understood and useful class of I/O systems are those constructed from linear subsystems. The ubiquitous PI control scheme is composed of linear subsystems. Linear systems are trivially composable – the serial composition of two linear systems is again linear, as are the parallel composition and sum. The analysis of a large linear system is no more difficult than the analysis of its smaller components. Furthermore, although the class of linear systems may appear limiting, essentially all physical systems are approximately linear near their desired operating regions. For this reason linear controllers are used to control a large class of non-linear plants.

There are two fundamental representations of linear systems, the state space representation and the frequency space representation. Both representations are useful and complementary in design and analysis. A typical single-input single-output linear system can be represented in state space by

$$\dot{x} = Ax + Bu \quad (3)$$

$$y = Cx + Du \quad (4)$$

Where $u, y: \mathbb{R} \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, and A, B, C, D are appropriately sized real matrices. The frequency space representation of this system can be found by taking the Laplace transform of (3)–(4).

$$sX(s) = AX(s) + BU(s) \quad (5)$$

$$Y(s) = CX(s) + DU(s) \quad (6)$$

The frequency space representation is only defined for linear systems. However, in frequency space, linear systems can be analysed based on their I/O behaviour, abstracting away the internal state x through the transfer function

$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (7)$$

Two plants may have different state space representations, yet have identical transfer functions. The transfer function reduces composition and comparison of dynamical systems based on their I/O behaviour to simple algebra.

The atomic components of linear I/O systems are linear zero- and first-order systems. In particular, signal splitting, integration, gain and summation (see Fig. 2) form the basis

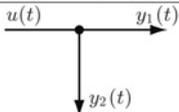
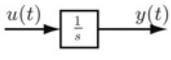
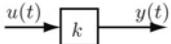
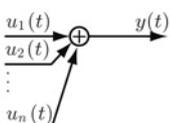
Component Type	Block Diagram	State Space Equations	Transfer Function
Signal Splitting		$\dot{x}(t) = 0$ $y_1(t) = u(t)$ $y_2(t) = u(t)$	$\frac{Y_1(s)}{U(s)} = 1$
Integration		$\dot{x}(t) = u(t)$ $y(t) = x(t)$	$\frac{Y(s)}{U(s)} = \frac{1}{s}$
Gain		$\dot{x}(t) = 0$ $y(t) = ku(t)$	$\frac{Y(s)}{U(s)} = k$
Summation		$\dot{x}(t) = 0$ $y(t) = \sum_{i=1}^n u_i(t)$	$Y(s) = \sum_{i=1}^n U_i(s)$

Fig. 2 Primitive components of continuous time linear I/O systems represented as a block diagram, state space equations and frequency space equations

of all linear systems. To apply the results of linear systems to synthetic biology, our task is to implement these basic primitives and describe how they can be physically composed to obtain any arbitrary linear system.

2 Results

We show how to realise linear I/O systems at two levels of abstraction. First, we describe an intermediate representation that uses simple, idealised chemical reactions. This level of abstraction requires that we instantiate the notions of a signal and of integration, gain and summation blocks. Second, we describe an implementation of the intermediate representation in which the ideal chemical reactions are approximated by enzyme-free DNA devices. The DNA implementation is one of many possible implementations that are suggested by the form of the intermediate representation. For example, one might also imagine implementations with transcriptional switches [8] or MAPK cascades [19–21], to name a couple of possibilities. Throughout this paper we use the PI controller [16] illustrated in Fig. 1a both as a design objective and as a running example. It should be clear, however, that our approach allows for the implementation of any finite dimensional linear system with chemical reactions and with DNA devices.

2.1 Signals represented as chemical concentrations

A natural representation of a signal within a block diagram might be via the time-varying concentration of a particular chemical species. However, concentrations can only be non-negative, whereas signals in arbitrary linear systems take on positive and negative values. Therefore we represent a signal u by the difference in concentration between two particular chemical species. Specifically, for each signal u we introduce the chemical species u^+ and u^- .

Remark 1: We have overloaded the symbols u^+ and u^- so that they represent both time varying concentrations and the names of a particular chemical species. It should be clear from the context which usage is intended.

The species u^+ and u^- are referred to as the positive and negative components of the signal u , respectively,

and the actual value of u equals the difference between its components

$$u = u^+ - u^- \tag{8}$$

One implication of this scheme is that no signal value has a unique representation. For example, $u^+ = 100 \text{ M}$ and $u^- = 100 \text{ M}$ represents the same signal value as $u^+ = 0 \text{ M}$ and $u^- = 1 \text{ M}$. We refer to the unique representation of u where $u^+ = 0 \text{ M}$ or $u^- = 0 \text{ M}$ as the minimal representation of u . Forcing signals to be represented minimally would be more efficient in an actual implementation of this scheme. To accomplish this forcing, the implementations below include reactions in which the positive and negative components of a signal annihilate each other, as in (11).

To allow for blocks to be ‘wired’ arbitrarily, it should be that a block has no retroactive effect on its input signals [17]. For example, in electrical implementations of I/O systems, we require that devices draw almost no current from their input signal sources, so that the meaning of a signal is not changed by the devices using it. One way to satisfy this notion is to require that signals act as catalysts for the reactions implementing blocks to which they are wired. In the sequel we show how integration, gain and summation blocks can be approximated using a minimal set of reaction types: catalysis, degradation and annihilation reactions, given in (9), (10), (11), respectively.



2.2 Integration

An integration block takes as input a signal $u(t)$ and produces the output signal $y(t) = \int_0^t u(\tau) d\tau + y(0)$ with $t \in \mathbb{R}$. The transfer function of an integration block is $1/s$, as shown in Fig. 2. A chemical reaction network that implements this block is as follows



Where $\alpha, \gamma, \eta \in \mathbb{R}^+$. The block consists of two catalysis reactions (12) and an annihilation reaction (13).

Remark 2: We collapsed two reactions, $u^+ \xrightarrow{\alpha} u^+ + y^+$ and $u^- \xrightarrow{\alpha} u^- + y^-$, into reaction (12). We use this notation with both \pm and \mp superscripts for brevity.

The catalysis of u^+ and u^- at rate α and annihilation of y^+ and y^- results in the following mass action equations

$$\dot{u}^+ = \dot{u}^- = 0 \quad (14)$$

$$\dot{y}^+ = \alpha u^+ - \eta y^+ y^- \quad (15)$$

$$\dot{y}^- = \alpha u^- - \eta y^+ y^- \quad (16)$$

$$\dot{y} = \dot{y}^+ - \dot{y}^- = \alpha u \quad (17)$$

Note that the bimolecular annihilation reaction drives the concentration of chemical species y^+ and y^- towards the minimal representation of the signal y , and causes the dynamics of y^+ and y^- to be non-linear. However, the signal dynamics, $y = y^+ - y^-$, remain linear owing to the symmetry between y^+ and y^- .

2.3 Gain and summation

Gain and summation blocks produce output signals that are linear combinations of their inputs. A gain block takes as input a single signal $u(t)$ and produces the output signal $y(t) = ku(t)$ where $k \in \mathbb{R}$. A summation block takes as input the signals $\{u_i(t)\}_{i=1}^n$, and produces the output signal $y(t) = \sum_{i=1}^n u_i(t)$. The transfer functions for gain and summation are $Y(s)/U(s) = k$ and $Y(s) = \sum_{j=1}^n U_j(s)$, respectively. The following chemical reaction network that outputs a linear combination of its input signals implements both gain and summation

$$u_i^\pm \xrightarrow{\gamma k_i} u_i^\pm + y^\pm \quad (18)$$

$$y^\pm \xrightarrow{\gamma} \emptyset \quad (19)$$

$$y^+ + y^- \xrightarrow{\eta} \emptyset \quad (20)$$

Where $k_i, \gamma, \eta \in \mathbb{R}^+$ for $i \in \{1, 2, \dots, n\}$. In the special case $n = 1$, this chemical representation approximates the gain block in Fig. 2 for $k \geq 0$. For $n > 1$ this chemical representation approximates the summation block in Fig. 2. The chemical reaction network consists of $2n$ catalysis reactions (18), two degradation reactions (19) and one annihilation reaction (20). The chemical reaction network gives us the following mass action equations

$$\dot{u}_i^+ = \dot{u}_i^- = 0 \quad (21)$$

$$\dot{y}^+ = \gamma \left(\sum_{i=1}^n k_i u_i^+ - y^+ \right) - \eta y^+ y^- \quad (22)$$

$$\dot{y}^- = \gamma \left(\sum_{i=1}^n k_i u_i^- - y^- \right) - \eta y^+ y^- \quad (23)$$

$$\dot{y} = \gamma \left(\sum_{i=1}^n k_i u_i - y \right) \quad (24)$$

For a constant inputs u_i , the steady state value of y is

$$\lim_{t \rightarrow \infty} y(t) = \sum_{i=1}^n k_i u_i \quad (25)$$

The chemical representation can be extended to allow negative multiplicative weights. For $k_i < 0$, the catalysis reactions (18) are replaced with

$$u_i^\pm \xrightarrow{\gamma k_i} u_i^\pm + y^\mp \quad (26)$$

As before, the annihilation reaction drives the concentration of chemical species y^+ and y^- towards a minimal representation of the signal y without affecting the dynamics of the signal y .

2.4 Any linear I/O system can be approximated with ideal chemical reactions

Although the dynamics of the chemical representations are not equivalent to the dynamics of integration, gain and summation, the chemical representations can be used to approximate the dynamics of integration, gain and summation. As shown in the prequel, the signal dynamics of the chemical representations are linear systems, which means that we can compare the I/O behaviour of the chemical implementations to ideal integration, gain and summation, through their transfer functions. In particular, as shown in the SI Text, increasing the rate parameter γ for gain and summation has the effect of speeding transient dynamics. This results in a closer time-response to ideal gain and summation given a step input. We regard a chemical representation of a linear I/O system as an approximation of an ideal linear I/O system if the transfer functions of the two systems can be made equivalent in the limit as the rate parameter γ goes to infinity.

Equation (17) shows that in the chemical implementation of integration, the output signal y is the integral of αu . This results in the transfer function

$$\frac{Y(s)}{U(s)} = \frac{\alpha}{s} \quad (27)$$

To produce the integration dynamics described in Fig. 2, we can set the rate $\alpha = 1$ or compose this system in series with a chemical implementation of gain where $k = \alpha^{-1}$. The equivalent dynamics of the chemical representation where $\alpha = 1$ and an integration block are illustrated in Fig. 3a.

The transfer functions for the chemical representations of gain and summation are computed from (24)

$$Y(s) = \frac{\gamma}{s + \gamma} \sum_{i=1}^n k_i U_i(s) \quad (28)$$

The result is a first-order stable linear system. Shown in Figs. 3b and c, the chemical representation tracks the behaviour of the gain block ($n = 1$) illustrated in Fig. 2 and weighted summation ($n > 1$) with zero steady-state error for step input signal $u(t)$ and square wave input signals $u_1(t)$ and $u_2(t)$. Increasing γ the I/O behaviour of this system can be made arbitrarily close to the I/O

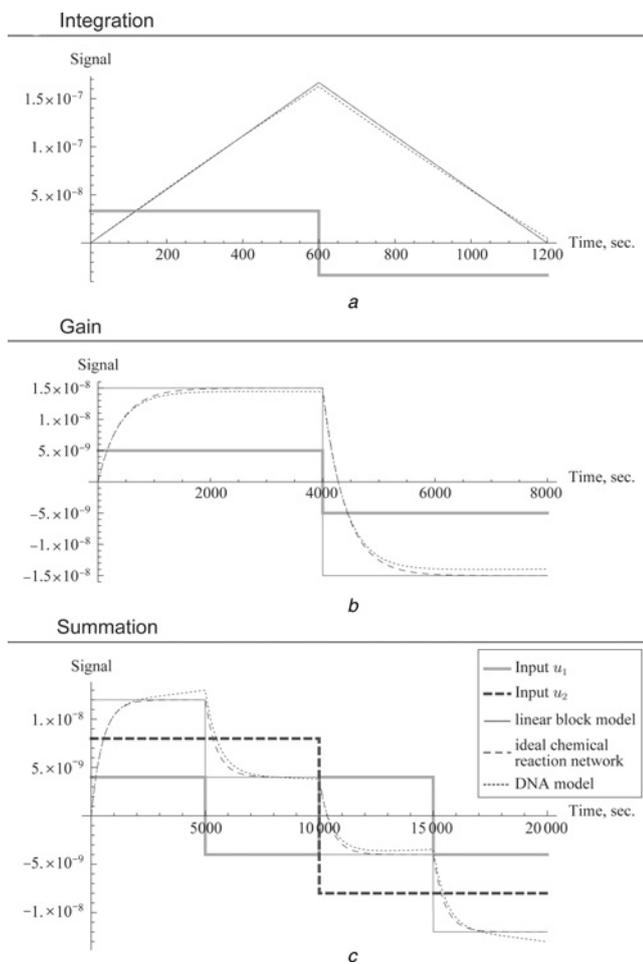


Fig. 3 Step response of the linear block model, chemical reaction representation, and DNA model of integration, gain and summation blocks

a Integration block: The linear block model follows the trajectory $y(t) = \int_0^t u_1(t)$. The ideal chemical reaction representation follows this trajectory precisely. The DNA model drifts from the ideal chemical reaction trajectory as molecular fuel species are consumed

b Gain block: The linear block model follows the trajectory $y(t) = 3u_1(t)$. The chemical reaction representation produces the correct steady-state output. As with integration, the DNA model closely follows the ideal chemical reaction trajectory, but drifts as fuel species are consumed

c Summation block: The linear block model follows the trajectory $y(t) = u_1(t) + u_2(t)$. Given inputs u_1 and u_2 the output should consist of four monotonically decreasing steps. The chemical reaction representation follows each step in steady-state. As before, the DNA model drifts from the ideal chemical reaction representation as fuel species are consumed. For each system the input u_1 is a square wave

behaviour of a weighted summation

$$\lim_{\gamma \rightarrow \infty} Y(s) = \sum_{i=1}^n k_i U_i(s)$$

To produce the summation dynamics described in Fig. 2, set $k_i = 1$ for $i = 1, \dots, n$. A similar result follows for the negative weight extension where reactions (18) are replaced with reactions (26) for some set $i \in I \subseteq \{1, \dots, n\}$.

Because any finite dimensional linear system can be decomposed into integration, gain and summation blocks the chemical reaction representation of linear I/O systems can be used to approximate any finite dimensional linear I/O system. It follows directly that as γ goes to infinity, the

transfer function of an implemented chemical system approaches that of the ideal specification.

Theorem 1: All finite-dimensional continuous time linear systems can be approximated with catalysis and degradation reactions.

Two details of the ideal chemical reaction model may be particularly concerning when implementing with biomolecules: first, the rate γ will be physically constrained to some (very large) finite value. Second, it may be impossible to precisely match reaction rates between chemical reactions as specified, for example, by the reaction pairs (18) and (19). In simulation, realistic values of γ can produce a time-response close to the specified system for step inputs, as illustrated in Figs. 1c and 3 for integration, gain and summation components as well as the more complicated PI controller. If reaction rates involving positive and negative components do not match, then for fast annihilation reaction rates, η , the signal dynamics of the resulting system can be shown to be close to those of a related linear switch system. Details are discussed in the SI Text.

2.5 Optimising the chemical representation

The approximation of a linear system by a chemical reaction network can be made arbitrarily good by tuning the rate parameter γ . However, as mentioned, in practice it is not possible to assign arbitrary values to γ . A time response closer to the ideal system may be obtained by exploiting other parameters and decreasing the order of the chemical reaction model. For example, a gain composed with an integrator as illustrated in Fig. 4a is a first-order system. The unoptimised chemical representation (Fig. 4b) is a second-order system, however this system can be implemented precisely using an optimised first-order weighted integration reaction (Fig. 4c). Illustrated in Fig. 4d, given a step input, the unoptimised representation of this system results in steady-state error (which decreases as γ increases), while the optimised representation replicates the dynamics of the ideal I/O system exactly.

In general two other optimisations can be made in the chemical representation: First, the weighted summation of n signals may be approximated by the second-order representation of n summation and gain reaction sets, or a first-order weighted summation reaction network. Second, the weighted or unweighted summation of n integrated signals may be approximated by second or third-order representation of integration composed with gain (in the weighted case) and summation reaction sets, or implemented precisely by n weighted integration reaction sets, all of which share the same output species. In addition to reducing the overall order and output error of the approximation, these optimisations minimise the total number of species needed to implement a linear I/O system with catalysis, degradation and annihilation reactions. We take advantage of these optimisations in the following example.

2.6 Example 1: chemical representation of a PI controller

To illustrate our method, we construct the PI controller shown in Fig. 1a. The PI controller is an example of a closed loop linear system designed to drive the output of the plant P to a desired set point u . The key feature of the PI controller is

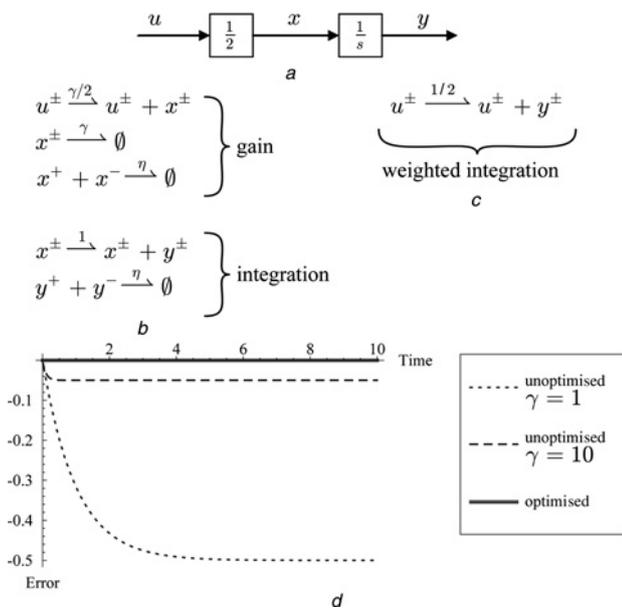


Fig. 4 Approximation error for optimised and unoptimised ideal chemical reaction representations of the I/O system $\dot{x} = (1/2)u$, $y = x$

a Block diagram representing the ideal weighted integration system
 b Unoptimised chemical reaction representation. This representation consists of three pairs of signal species, a gain block and an integration block. The signal dynamics resulting from mass action kinetics is a second-order linear system
 c Optimised chemical reaction representation. This representation consists of two pairs of signal species and a single weighted integration block. The signal dynamics resulting from mass action kinetics is a first-order linear system that matches the dynamics of the ideal system
 d Error trajectory for the signal y given $u(0) = u^+(0) - u^-(0) = 1$ and $x(0) = y(0) = 0$. The unoptimised chemical reaction representation of the weighted integration system results in some non-zero steady-state error which decreases monotonically as γ increases. The Optimised chemical reaction representation results in zero steady state error

that it tracks any step input u with zero steady-state error for a large class of plants. Such a device could be useful, for example, in regulating a fuel species driving a variety of downstream devices. Suppose the plant P is realised by leaky expression and chemical devices that produce and consume the signal species x_5^\pm according to the reactions

$$\emptyset \xrightarrow{\gamma\delta_1} x_5^\pm \quad (29)$$

$$x_5^\pm \xrightarrow{\gamma\delta_2} \emptyset \quad (30)$$

for $\delta_1, \delta_2 \in \mathbb{R}^+$. As derived in the SI Text, the effect of (29)–(30) can be modelled as the plant $P(s) = (1 + \delta_2)^{-1}$. Note that δ_1 does not appear in this expression as a result of the symmetric effect (29) has on molecular species x_5^+ and x_5^- .

In general, there are two steps to compiling a biochemical controller from a block diagram. First, signals in the block diagram are enumerated, and it is determined which signals correspond to chemical species. Second, primitive blocks are instantiated by sets of chemical reactions. The PI controller in Fig. 1a consists of signals $u, y, x_1, \dots, x_5, x_6$ where u is the input signal and $y = x_6$ is the output signal. Using optimisations from the prequel, we approximate the PI controller using species $u^\pm, x_1^\pm, x_4^\pm, x_5^\pm$, in the chemical reactions outlined in Fig. 5. The signal dynamics of the chemical realisation is a second-order linear approximation

Component	Ideal Chemical Reactions		
Plant	\emptyset	$\xrightarrow{\gamma\delta_1}$	x_5^\pm
$x_5 \rightarrow P(s) \rightarrow x_6$	x_5^\pm	$\xrightarrow{\gamma\delta_2}$	\emptyset
Summation	x_6^\pm	$=$	x_5^\pm
$u \rightarrow \oplus \rightarrow x_1$	u^\pm	$\xrightarrow{\gamma}$	$x_1^\pm + x_1^\mp$
$x_6 \rightarrow \ominus \rightarrow \oplus$	x_5^\pm	$\xrightarrow{\gamma}$	$x_5^\pm + x_1^\mp$
Weighted Integration	$x_1^\pm + x_1^\mp$	$\xrightarrow{\eta}$	\emptyset
$x_1 \rightarrow k_I \rightarrow \frac{1}{s} \rightarrow x_4$	x_1^\pm	$\xrightarrow{k_I}$	$x_1^\pm + x_4^\pm$
Weighted Summation	$x_4^\pm + x_4^\mp$	$\xrightarrow{\eta}$	\emptyset
$x_1 \rightarrow k_P \rightarrow \oplus$	x_1^\pm	$\xrightarrow{\gamma k_P}$	$x_1^\pm + x_5^\pm$
$x_4 \rightarrow \oplus \rightarrow x_5$	x_4^\pm	$\xrightarrow{\gamma}$	$x_4^\pm + x_5^\pm$
	x_5^\pm	$\xrightarrow{\gamma}$	\emptyset
	$x_5^\pm + x_5^\mp$	$\xrightarrow{\eta}$	\emptyset

Fig. 5 PI controller from Fig. 1a implemented in ideal chemical reactions

of the first-order PI controller with the following signal dynamics

$$\dot{x}_1 = \gamma(u - x_5 - x_1) \quad (31)$$

$$\dot{x}_4 = k_I x_1 \quad (32)$$

$$\dot{x}_5 = \gamma((k_P x_1 + x_4 + \delta_1) - (1 + \delta_2)x_5) \quad (33)$$

As shown in the SI Text, and demonstrated in Fig. 1, this controller produces the stable output $\lim_{t \rightarrow \infty} y(t) = u(t)$ for any step input $u(t)$ when $\delta_1, \delta_2 \in \mathbb{R}^+$. Additionally, as $\gamma \rightarrow \infty$, the transient dynamics of the chemical controller approach those of the ideal PI controller.

2.7 DNA implementation

The ideal chemical reactions representation of arbitrary linear systems is a useful template which can be used to guide the implementation of arbitrary linear systems in physical substrates. Choosing a particular biomolecular implementation forces us to consider physical constraints such as limited supplies of energy-storing fuel molecules and finite maximum reaction rates.

In enzyme-free DNA reactions, energy can be stored in metastable DNA molecules [7, 9, 10, 22]. Chemical reactions are driven by the transformation of metastable fuel into non-reactive waste products, and the reaction pathways are programmed through the sequence of nucleic acids and the availability of single-stranded binding sites, or ‘toeholds’. Toehold-mediated branch migration and strand displacement reactions are quantitatively well-understood [11–13], and many complex devices have been constructed around this design principle [7–10]. Recently it was shown that a DNA-based schema built on these design principles may approximate a large class of unimolecular and bimolecular chemical reactions [14]. Here we illustrate and simulate a DNA-based implementation of catalysis, degradation and annihilation (Fig. 6), as well as the PI controller introduced in Example 1, implemented using the schema of Soloveichik *et al.* Details of the implementation, simulations and conditions under which the implementation may be considered valid are given in the SI Text.

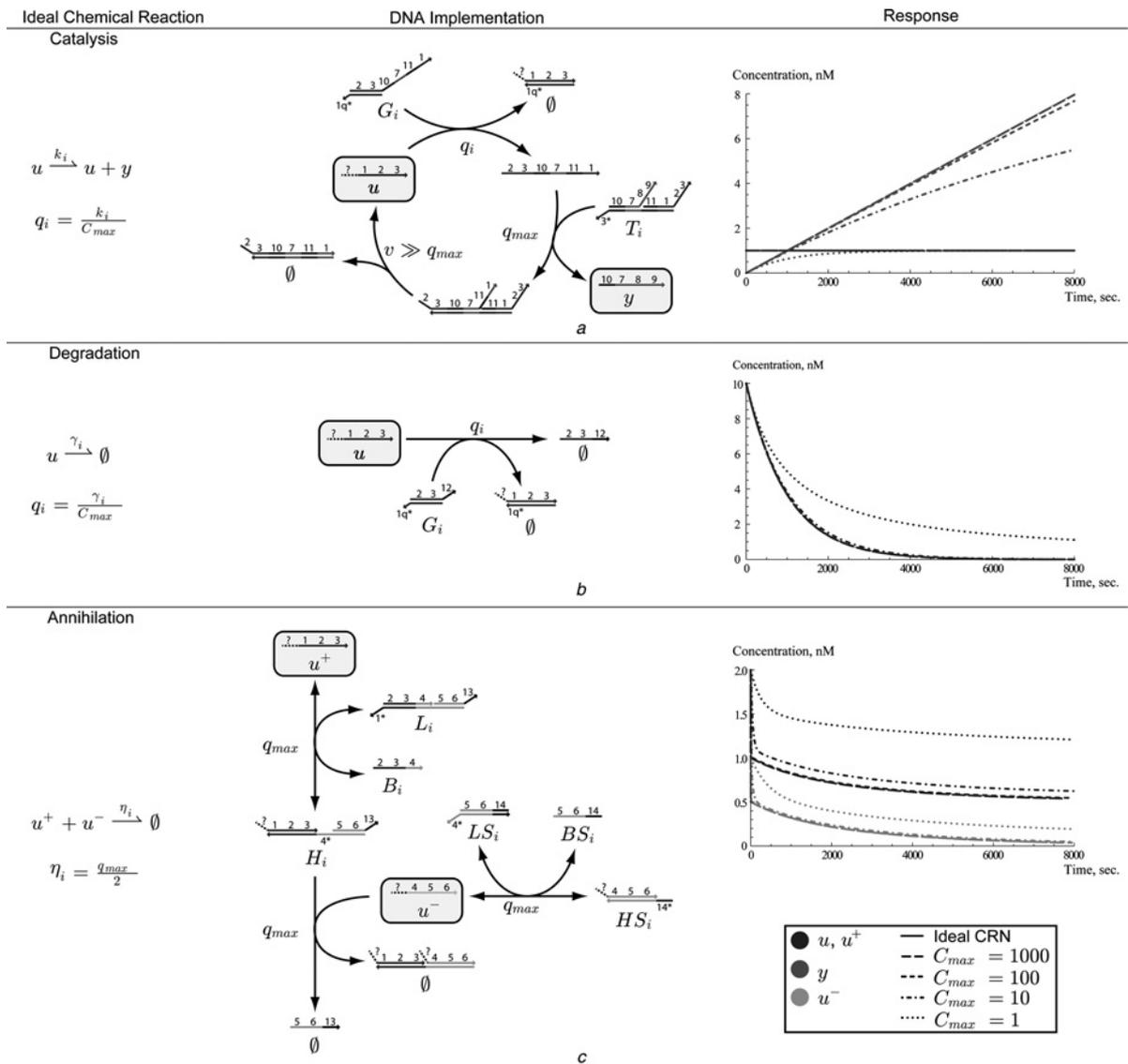


Fig. 6 Ideal chemical reaction, DNA implementation and signal response for

- a Catalysis
- b Degradation
- c Annihilation reactions

Domain 1_q is a subset of the domain 1. The initial concentration of fuel species G_i, T_i, L_i, B_i, LS_i and BS_i are set to $C_{max} = 1, 10, 100, 1000$ nM. For the catalysis and degradation response, $u(0) = 1$ nM. For the annihilation response, $u^+(0) = 1\zeta$ nM, $u^-(0) = 0.5\zeta$ nm, where $\zeta = 2$ is a scaling factor that attenuates for the initial fast transient where u^+ and u^- are sequestered in intermediate species. All other initial concentrations are set to zero

Following the notation from Soloveichik *et al.* [14], in this implementation of chemical reaction primitives for linear I/O systems we set q_{max} to be the maximum strand displacement rate and assume $q_1 \ll q_{max}$. All reactions are entropy-driven with potential energy stored in fuel species G_i, T_i, L_i, B_i, LS_i and BS_i . Fuel species are assumed to appear in initial concentration C_{max} . Numbers label domains, which are unique sequences of nucleotides. In this parameterising scheme, two labelled domains are complementary if and only if their labels are x and x^* , respectively. Of note is the domain 1_q^* , which denotes a subsequence of the domain 1^* with length tuned to the reaction rate q_i .

Signal molecules in a given linear system correspond to single stranded DNA made uniquely addressable via a sequence of three domains on the 5' end. Signal molecules catalyse the degradation of fuel species into non-interacting waste molecules. As with any entropy-driven system the trivial steady-state occurs where the fuel species have degraded into non-reactive waste. However, in order to

approximate the ideal chemical reactions, we limit our interest to the regime where the concentration of fuel species is much greater than the concentration of signal molecules. As shown in Fig. 6, the dynamics of the DNA system approach those of the ideal chemical reaction as C_{max} increases.

Fig. 6a illustrates the production of signal molecule y and degradation of fuels G_i and T_i catalysed by the signal molecule u . For large C_{max} the concentration $y(t)$ is approximately the integral of the concentration $u(t)$. Fig. 6b shows the degradation of fuel species G_i driven by the presence of signal molecule u . Again, for large C_{max} the concentration trajectory $u(t)$ is approximately exponential decay. Lastly, Fig. 6c shows the degradation of fuel species L_i and LS_i driven by the presence of both u^+ and u^- molecules. Note that the dynamics of this reaction can be separated into fast dynamics where u^+ and u^- are sequestered in intermediate species through their reactions with L_i and LS_i , respectively, and slow dynamics where u^-

degrades into waste through its interaction with the intermediate species that sequestered u^+ . Since it is the slow dynamics that approximate the ideal annihilation reaction, the initial concentration of u^+ and u^- must be scaled to attenuate for sequestering effect of the fast dynamics. That is, the initial concentration of all unregulated signal molecules must be scaled by a factor of two in order to approximate the ideal chemical reaction network. Again, the approximation of ideal annihilation improves for large C_{\max} . Simulations of integration, gain and summation blocks scaled to realistic parameters are given in Fig. 3. Details of the implementation and simulations are given in the SI Text.

2.8 Example 2: DNA PI controller

Utilising the DNA implementation of catalysis, degradation and annihilation, we again simulate a realisation of the PI controller illustrated in Fig. 1a. As with the ideal chemical reaction representation, the plant P is realised by the chemical reactions (29) and (30). Note that as before, this disturbance models leaky expression of a signal molecule, and downstream load on the output signal. This realisation is based on the optimised chemical representation shown in Fig. 5. The step response of the DNA implementation of the PI controller in the face of the plant P are shown in Fig. 1c. Note that as discussed earlier, the unregulated input u was scaled in order to attenuate the sequestering effect of the annihilation reaction. Note that the output trajectory of the DNA implementation of the PI controller closely matches the output trajectory of the ideal chemical reaction implementation discussed in Example 1 at the beginning of the simulation, tracking the input u with near zero steady-state error. The drift away from zero steady-state error as time increases is owing to the consumption of finite fuel molecules modelled in the DNA implementation. Details of the implementation and simulation are given in the SI Text.

3 Conclusions

We have demonstrated a method for realising arbitrary linear I/O systems at two levels of abstraction: (i) an intermediate chemical reaction representation, and (ii) a proposed DNA implementation. The intermediate chemical reaction representation of linear I/O systems provides a template for implementation using arbitrary biomolecules. Notably our construction of linear I/O systems relies on only three types of chemical reactions: catalysis, degradation and annihilation.

Although we have explored in some depth an implementation of linear I/O systems using DNA, one could imagine an implementation of these reactions using a variety of different substrates: gene regulatory networks, MAPK cascades or some combination of these systems [4–6, 19–21]. The DNA implementation provides a specific avenue for composing existing DNA devices, and case study with which to examine experimental considerations such as finite chemical concentrations, realistic reaction rates, unmodelled chemical reactions and data collection. We are currently working on an experimental implementation of the ideas in this paper using DNA, and plan to report on this effort soon.

The results here are representative of a new focus on abstraction in chemical systems. In building a design theory for chemistry, chemical reactions networks are usually the

most natural intermediate representation – the middle of the ‘hourglass’ [23]. Many different high level languages and formalisms have been and can likely be compiled to chemical reactions, and chemical reactions themselves (as an abstract specification) can be implemented with a variety of low level molecular mechanisms.

4 Materials and methods

All simulations are performed in Mathematica, Version 7.0.1.0 [24], with numerical solver NDSolve. Mathematica files are available upon request. Specific details of the ideal chemical realisation and DNA implementation of linear I/O primitives as well as the PI controller, including chemical reaction network models and reaction rates, are given in the SI Text. Analysis of the class of chemical plants (29) and (30) as well the effect of the PI controller on that class of plants are also discussed in the SI Text.

5 Acknowledgments

We thank D. Soloveichik, G. Seelig, D. Zhang and J. Bishop for useful discussions. This work is supported by NSF Grant #0832773: The Molecular Programming Project.

6 References

- Klonowski, W.: ‘Simplifying principles for chemical and enzyme reaction kinetics’, *Biophys. Chem.*, 1983, **18**, (2), pp. 73–87
- Gardner, T.S., Cantor, C.R., Collins, J.J.: ‘Construction of a genetic toggle switch in *Escherichia coli*’, *Nature*, 2000, **403**, (6767), pp. 339–342
- Elowitz, M.B., Leibler, S.: ‘A synthetic oscillatory network of transcriptional regulators’, *Nature*, 2000, **403**, (6767), pp. 335–338
- Kim, J.: ‘In vitro synthetic transcriptional networks’. PhD thesis, California Institute of Technology, 2007
- Sohka, T., Heins, R.A., Phelan, R.M., Greisler, J.M., Townsend, C.A., Ostermeier, M.: ‘An externally tunable bacterial band-pass filter’, *Proc. Natl. Acad. Sci. USA*, 2009, **106**, (25), pp. 10135–10140
- Basu, S., Gerchman, Y., Collins, C.H., Arnold, F.H., Weiss, R.: ‘A synthetic multicellular system for programmed pattern formation’, *Nature*, 2005, **434**, (7037), pp. 1130–1134
- Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E.: ‘Enzyme-free nucleic acid logic circuits’, *Science*, 2006, **314**, (5805), pp. 1585–1588
- Kim, J., White, K.S., Winfree, E.: ‘Construction of an in vitro bistable circuit from synthetic transcriptional switches’, *Mol. Syst. Biol.*, 2006, **2**
- Zhang, D.Y., Turberfield, A.J., Yurke, B., Winfree, E.: ‘Engineering entropy-driven reactions and networks catalyzed by DNA’, *Science*, 2007, **318**, (5853), pp. 1121–1125
- Yurke, B., Turberfield, A.J., Mills, A.P., Simmel, F.C., Neumann, J.L.: ‘A DNA-fuelled molecular machine made of DNA’, *Nature*, 2000, **406**, (6796), pp. 605–608
- Zhang, D.Y., Winfree, E.: ‘Control of DNA strand displacement kinetics using toehold exchange’, *J. Am. Chem. Soc.*, 2009, **131**, (47), pp. 17303–17314
- Green, C., Tibbetts, C.: ‘Reassociation rate limited displacement of DNA strands by branch migration’, *Nucleic Acids Res.*, 1981, **9**, (8), pp. 1905–1918
- Panyutin, I.G., Hsieh, P.: ‘Formation of a single base mismatch impedes spontaneous DNA branch migration’, *J. Mol. Biol.*, 1993, **230**, (2), pp. 413–424
- Soloveichik, D., Seelig, G., Winfree, E.: ‘DNA as a universal substrate for chemical kinetics’, *Proc. Natl. Acad. Sci.*, 2010, **107**, (12), pp. 5393–5398
- Schulman, R., Winfree, E.: ‘Programmable control of nucleation for algorithmic selfassembly’, *DNA Computing 10*, 2005, (*LNCS*, **3384**), pp. 319–328
- Dorf, R.C., Bishop, R.H.: ‘Modern control systems’ (Prentice Hall, 2008, 11th edn.)
- Del Vecchio, D., Ninfa, A.J., Sontag, E.D.: ‘Modular cell biology: retroactivity and insulation’, *Mol. Syst. Biol.*, 2008, **4**

- 18 Dullerud, G.E., Paganini, F.: 'A course in robust control theory' (Springer, 2000, 1st edn.)
- 19 Park, S., Zarrinpar, A., Lim, W.A.: 'Rewiring MAP kinase pathways using alternative scaffold assembly mechanisms', *Science*, 2003, **299**, (5609), pp. 1061–1064
- 20 Bashor, C.J., Helman, N.C., Yan, S., Lim, W.A.: 'Using engineered scaffold interactions to reshape MAP Kinase pathway signaling dynamics', *Science*, 2008, **319**, (5869), pp. 1539–1543
- 21 Grünberg, R., Serrano, L.: 'Strategies for protein synthetic biology', *Nucleic Acids Res.*, 2010, **38**, (8), pp. 2663–2675
- 22 Yin, P., Choi, H.M.T., Calvert, C.R., Pierce, N.A.: 'Programming biomolecular self-assembly pathways', *Nature*, 2008, **451**, (7176), pp. 318–322
- 23 Doyle, J., Csete, M.: 'Rules of engagement', *Nature*, 2007, **446**, (7138), p. 8600
- 24 <http://www.mathematica.com>, accessed 2010