CORE
Metadata, citation and similar papers at core.ac.uk
Provided by CURVE/open

Biomolecular implementation of nonlinear system theoretic operators

Foo, M, Sawlekar, R, Kim, J, Bates, DG, Stan, GB & Kulkarni, V

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:
Foo, M, Sawlekar, R, Kim, J, Bates, DG, Stan, GB & Kulkarni, V 2017, Biomolecular implementation of nonlinear system theoretic operators. in *2016 European Control Conference, ECC 2016.* 2016 European Control Conference, ECC 2016, Institute of Electrical and Electronics Engineers Inc., pp. 1824-1831
https://dx.doi.org/10.1109/ECC.2016.7810556

DOI 10.1109/ECC.2016.7810556
ISBN 978-1-5090-2960-0
ESBN 978-1-5090-2959-4

Publisher: IEEE

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

Biomolecular implementation of nonlinear system theoretic operators

Mathias Foo[1], Rucha Sawlekar[1], Jongmin Kim[2], Declan G. Bates[1], Guy-Bart Stan[3], and Vishwesh Kulkarni[1]

***Abstract*— Synthesis of biomolecular circuits for controlling molecular-scale processes is an important goal of synthetic biology with a wide range of *in vitro* and *in vivo* applications, including biomass maximization, nanoscale drug delivery, and many others. In this paper, we present new results on how abstract chemical reactions can be used to implement commonly used system theoretic operators such as the polynomial functions, rational functions and Hill-type nonlinearity. We first describe how idealised versions of multi-molecular reactions, catalysis, annihilation, and degradation can be combined to implement these operators. We then show how such chemical reactions can be implemented using enzyme-free, entropy-driven DNA reactions. Our results are illustrated through three applications: (1) implementation of a Stan-Sepulchre oscillator, (2) the computation of the ratio of two signals, and (3) a PI+antiwindup controller for regulating the output of a static nonlinear plant.**

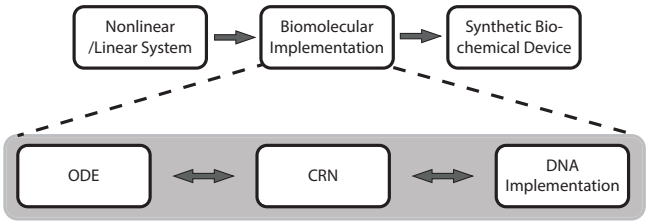

Fig. 1: Synthetic biomolecular devices should ideally have a nonlinear input-output behaviour. In this paper, we present results on how biomolecular implementations of nonlinear operators can be realised by first converting the ordinary differential equations (ODEs) into their equivalent chemical reaction networks (CRNs) and then obtaining a DNA implementation of these CRNs.

I. INTRODUCTION

Design of biomolecular circuits for *in situ* monitoring and control is an important goal of synthetic biology, with numerous potential applications ranging from metabolic production of biomaterials to the design of "smart" therapeutics capable of diagnosis and treatment. So far, several synthetic devices have been designed and implemented *in vivo* using protein expression and gene regulation mechanisms: for example, logic gates [1], memory elements [2], oscillators [3], filters [4]-[5] and controllers of cellular differential processes [6]. However, the problem of imparting a programmable robust dynamic behaviour to such synthetic biological circuits has remained open, primarily because a proper understanding of the input-output properties of genetic components is still currently lacking, especially in the context of their interactions with the host cell within which these circuits operate.

Recently, the direct use of nucleic acids for performing computation has emerged as a promising approach for addressing the above problems [7]-[10]. For these type of systems, the sequences of nucleic acid components dictate their interactions through the well-known Watson-Crick base-pairing mechanism, which enables a precise programming of molecular interactions by the choice of relevant sequences. This approach has allowed the implementation of a number of complex circuits based on DNA strand displacement [11], DNA enzyme [12] and RNA enzyme [13], and has been used for the modelling and implementation of various nucleic-acids-based circuits such as feedback controllers [14], predator-prey dynamics [15] and transcriptional oscillators [16].

It is possible to approximate any abstract *chemical reaction network* (CRN) by a set of suitably designed DNA strand displacement reactions [17]. This logic extends well to approximate a set of linear ordinary differential equations (ODEs) by a set of suitably designed DNA strand displacement reactions [10], [18]. This has opened up the possibility of utilising nucleic acid computations for the design and implementation of various types of synthetic biological circuits - the approach is illustrated conceptually in Fig. 1.

In this paper, we build on the framework of Oishi-Klavins, proposed in [19], to present results on how abstract chemical reactions can be used to implement a number of nonlinear system theoretic operators such as polynomial functions, rational functions and Hill-type nonlinearities. It turns out that an elegant mathematical framework on how concentrations of abstract biochemical species should be used to implement several complex computational functions such as the square root, the n-th root, and division of two numbers was already well presented in [20]. Some of the ideas of [20] are found in [19], and hence are reflected in our constructs as well. However, no suggestions are found in [20] on how such operations can be implemented using real world biomolecular species such as DNA, RNA, and enzymes. In contrast, our approach clearly tackles this problem by using the concepts developed in [10] and [19]. We then show

[1]Mathias Foo, Rucha Sawlekar, Declan G. Bates and Vishwesh Kulkarni are with Warwick Integrative Synthetic Biology (WISB), School of Engineering, University of Warwick, Coventry CV4 7AL, UK. M.Foo@warwick.ac.uk,R.Sawlekar@warwick.ac.uk, D.Bates@warwick.ac.uk, V.Kulkarni@warwick.ac.uk

[2]Jongmin Kim is with the Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, Massachusetts 02115, USA. Jongmin.Kim@wyss.harvard.edu

[3]Guy-Bart Stan is with the Department of Bioengineering, Imperial College London, London SW7 2AZ, UK g.stan@imperial.ac.uk

how these abstract chemical reactions can be realised using enzyme-free, entropy-driven DNA reactions.

The class of biochemical circuits that can be implemented by our framework is not limited to the ones covered by the well-known theory of chemical reaction networks, developed in [21]–[24], in that our framework facilitates the implementation of rational functions as well.

The paper is organised as follows. In Section II, the results of [19] on representing linear systems using idealised chemical reactions are summarised. In Section III, we present our main results on how abstract chemical reactions can be used to implement polynomial functions, rational functions and Hill-type nonlinearities. After presenting an overview on DNA implementations in Section IV, we present simulation case studies in Section V to illustrate the main results. The paper is concluded in Section VI and all relevant chemical reactions along with their DNA implementations are summarised in the Appendix.

II. Background Results: Gain, Summation, Integration

Our notation follows the notation used in [19] and [10]: for example, we represent a bidirectional, i.e., a reversible bimolecular chemical reaction as

$$X_1 + X_2 \underset{\delta_2}{\overset{\delta_1}{\rightleftharpoons}} X_3 + X_4,$$

where X_i are chemical species with X_1 and X_2 being the reactants and X_3 and X_4 being the products. Here, δ_1 denotes the forward reaction rate and δ_2 denotes the backward reaction rate. A unimolecular reaction features only one reactant whereas a multimolecular reaction features two or more reactants. Degradation of a chemical species X at rate K into a waste or an inert form is denoted as $X \xrightarrow{K} \emptyset$.

Whereas signals in systems theory can take both positive and negative values, biomolecular concentrations can only take non-negative values. Hence, following the approach in [19] and [10], we represent a signal, x as the difference in concentration of two chemical species, x^+ and x^-. Here, x^+ and x^- are respectively the positive and negative components of x such that $x = x^+ - x^-$. In practice, x^+ and x^- can be realised as single strand DNA molecules, as illustrated in [10].

In [19], results on how to represent elementary system theoretic operations such as gain, summation and integration using idealised abstract chemical reactions are obtained and it is shown that only three types of elementary chemical reactions, namely, catalysis, annihilation and degradation are needed for such representations. In [10], this set of elementary chemical reactions is further reduced to only two. We now summarise their main results and refer the interested reader to [10] and [19] for the complete background theory.

Lemma 1: [Scalar gain K]
Let $x_o = Kx_i$ where x_i is the input, x_o is the output and K is the gain. This operation is implemented, at the steady state, using the following set of abstract chemical reactions: $x_i^{\pm} \xrightarrow{\gamma K} x_i^{\pm} + x_o^{\pm}$, $x_o^{\pm} \xrightarrow{\gamma} \emptyset$ and $x_o^+ + x_o^- \xrightarrow{\eta} \emptyset$, where γ and η are the kinetic rates associated with degradation and annihilation respectively. □

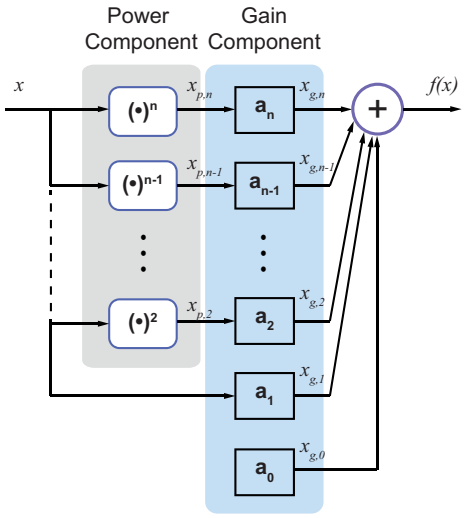

Fig. 2: The input-output system derived in Lemma 5 to compute the univariate polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$. Our result uses intermediate variables $x_{p,i}$ which can be computed using the abstract chemical reactions given by Lemma 4. This implementation requires $11n+3$ abstract chemical reactions, where n is the degree of the polynomial $f(x)$.

Lemma 2: [Summation]
Consider the summation operation $x_o = x_i + x_d$, where x_i and x_d are the inputs and x_o is the output. This operation is implemented, at the steady state, using the following set of abstract chemical reactions: $x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}$, $x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\pm}$, $x_o^{\pm} \xrightarrow{\gamma} \emptyset$ and $x_o^+ + x_o^- \xrightarrow{\eta} \emptyset$. The subtraction operation $x_o = x_i - x_d$ is implemented using the following set of abstract chemical reactions: $x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}$, $x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\mp}$, $x_o^{\pm} \xrightarrow{\gamma} \emptyset$ and $x_o^+ + x_o^- \xrightarrow{\eta} \emptyset$. □

Lemma 3: [Integration]
Consider the integrator $x_o = K \int x_i dt$ where x_i is the input, x_o is the output, and K is the DC gain. It is implemented, at the steady state, using the following set of abstract chemical reactions: $x_i^{\pm} \xrightarrow{K} x_i^{\pm} + x_o^{\pm}$ and $x_o^+ + x_o^- \xrightarrow{\eta} \emptyset$. □

Strictly speaking, each of the equations with superscript $\pm$ and $\mp$ should be written down after decomposing it into its '$+$' and '$-$' individual component - for example, $x_i^{\pm} \xrightarrow{K} x_o^{\pm}$ should be written down as the the set of the following two reactions: $x_i^+ \xrightarrow{K} x_o^+$ and $x_i^- \xrightarrow{K} x_o^-$. However, for brevity, following [19], we will represent such a set of reactions compactly as $x_i^{\pm} \xrightarrow{K} x_o^{\pm}$.

Remark 1: It can be easily proved that the steady states stipulated in Lemmae 1 – 3 exist and that the time to reach the steady states is a function of the reaction rates. For example, the scalar gain K can be realised accurately within 1% error in $5/(\gamma K)$ seconds, the summation operation can be realised within 1% error in $5/\gamma$ seconds, and so on. In general, the time to reach the steady state is inversely

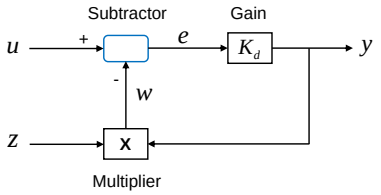

Fig. 3: A block diagram representation of the feedback system $\mathscr{S}_D$ that computes the ratio $y = u/z$ where u and z are biomolecular signals.

proportional to the concerned reaction rates. □

Using mass action kinetics, it follows that the gain operator realised in this manner is described using the following ODE, $\frac{dx_o}{dt} = \gamma(Kx_i - x_o)$. Likewise, the ODEs for the summation and integrator operations are given by $\frac{dx_o}{dt} = \gamma(x_i + x_d - x_o)$ and $\frac{dx_o}{dt} = Kx_i$, respectively.

III. Main Results

A. Polynomials and rational functions

Lemma 4: [Polynomial x^i]
Let $x_{p,i}$ denote the polynomial of degree n defined as $x_{p,i} = x^i$. Then, $x_{p,i}$ is realised through the following set of idealised abstract chemical reactions, which should be implemented via a series of em bimolecular reactions:

$$\underbrace{x^{\pm} + \cdots + x^{\pm}}_{i \text{ times}} \xrightarrow{\gamma_p} \underbrace{x^{\pm} + \cdots + x^{\pm}}_{i \text{ times}} + x^{\pm}_{p,i} \tag{1}$$

$$x^{\pm}_{p,i} \xrightarrow{\gamma_p} \emptyset, \tag{2}$$

$$x^{+}_{p,i} + x^{-}_{p,i} \xrightarrow{\eta} \emptyset, \tag{3}$$

where η is chosen to be arbitrarily large. □

Proof: Using generalised mass-action kinetics, it can be verified that the set of chemical reactions given by (1) to (3) is described using the following ordinary differential equation:

$$\frac{dx^{\pm}_{p,i}}{dt} = \gamma_p \left(\left(\prod_{\ell=1}^{i} x^{\pm} \right) - x^{\pm}_{p,i} \right). \tag{4}$$

Hence, using the final value theorem, it follows that the set of chemical reactions given by (1) to (3) implements the desired function at steady-state with $1/\gamma_p$ as the time constant. □

Lemma 5: [Univariate polynomial]
Let $f(x)$ be the univariate polynomial of degree n defined as

$$f(x) = \sum_{i=0}^{n} a_i x^i. \tag{5}$$

Then, $f(x)$ is realised through the feedforward system illustrated in Fig. 2. □

Proof: The proof follows in a straightforward manner using the proofs of Lemmas 1-4. □

Remark 2: It may be noted that the constant a_0 can be realised as $\emptyset \xrightarrow{a_0^+} x^+_{g,0}$ and $\emptyset \xrightarrow{a_0^-} x^-_{g,0}$ so that the product $x_{g,0}$ approaches the steady state value a_0 with the time constant of $1/a_0$. □

Remark 3: A multivariate polynomial such as, for example, $f(x,y)$ can be realised by extending Lemma 5 as $f(x,y) = f_1(x)f_2(y)$, where $f_1(x)$ and $f_2(y)$ are appropriately chosen univariate polynomials. The multiplication can be realised trivially using the following logic. Suppose we want to compute $z = xy$. Then the required chemical reactions are: $x^{\pm} + y^{\pm} \xrightarrow{\gamma_p} x^{\pm} + y^{\pm} + z^{\pm}$, $z^{\pm} \xrightarrow{\gamma_p} \emptyset$ and $z^{+} + z^{-} \xrightarrow{\eta} \emptyset$. The resulting ODE is $dz/dt = \gamma_p(xy - z)$ so that z approaches the required result xy at steady state, with the time constant being $1/\gamma_p$. Extending this logic to all intermediate variables of interest, the required feedforward circuit to compute the multivariate polynomials is obtained. □

Lemma 6: [Rational function]
Consider the system $\mathscr{S}_D$ shown in Fig. 3. Let the biomolecular signals u and z be its inputs. Then its output y computes the ratio u/z. □

Proof: From Fig. 3, we have $e = u - zy$ and $y = K_d e$. Substituting the former equation into the latter one and rearranging the variables, we get $y = K_d(u - yz) = \frac{K_d u}{1 + K_d z} = \frac{u}{(1/K_d) + z}$. If K_d is chosen large enough, $y \approx u/z$. □

Remark 4: Our implementation of the divider, which is a special case of the rational function, is illustrated in Fig. 3. It comprises a gain, a subtractor, and a multiplier. The corresponding sets of reactions are obtained using Lemmas 1, 2 and Remark 2 of Lemma 5. □

Remark 5: This configuration can be taken a step further to compute the ratio of two polynomials. Let $\hat{u}$ and $\hat{z}$ be the univariate polynomials of individual species. The abstract chemical reactions for both $\hat{u}$ and $\hat{z}$ can be realised using Lemma 5. Then, the ratio of these two polynomials, i.e., $\hat{u}/\hat{z}$ is computed in a similar manner as computing the ratio of u and z using Lemma 6. □

B. Hill-type nonlinearity

Hill-type nonlinearities occur naturally in the activation-inhibition interactions in biological networks and can be represented mathematically as either $N(x) = x^{m_1}/(\alpha + x^{m_2})$ or $N(x) = 1 - \left(x^{m_1}/(\alpha + x^{m_2})\right)$ where m_1 and m_2 are positive integers and $\alpha > 0$. One way to obtain a biomolecular implementation of such nonlinearities is to apply Lemma 6 to derive abstract chemical reactions that implement such a rational function and then obtain a DNA implementation of such chemical reactions. In many instances, however, the main objective is simply to implement a qualitative ultrasensitive input-output behaviour, rather than implement the exact quantitative rational function. In that case, it is preferable to note that the naturally occuring *mitogen activated protein kinase* (MAPK) cascades exhibit such an ultrasensitive response, and can be approximated by a set of chemical reactions of the form:

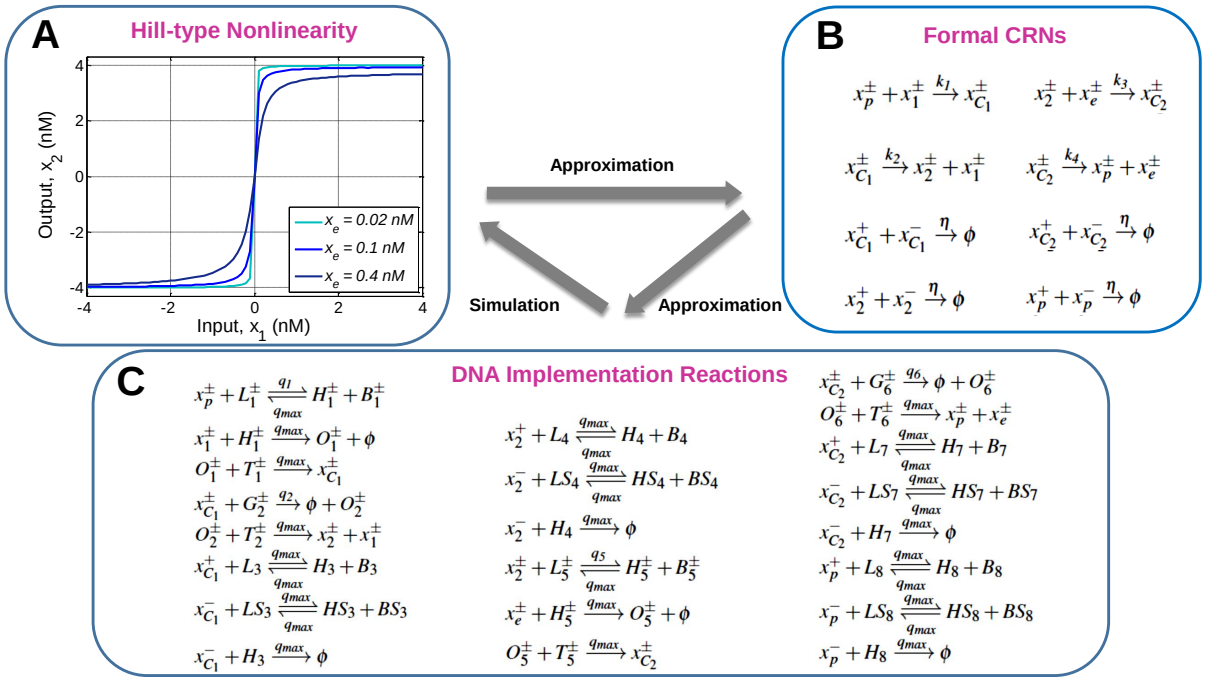

Fig. 4: Our implementation of a Hill-type nonlinearity by mimicking a stage of the MAPK cascades: (A) The desired qualitative ultrasensitive input-output response. (B) This ultrasensitive response can be obtained using these 12 abstract chemical reactions that convert the biomolecular signal x_1 into the biomolecular signal x_2. (C) DNA implementation of the reactions illustrated in (B). The strength of ultrasensitivity can be tuned by varying the concentration of x_e.

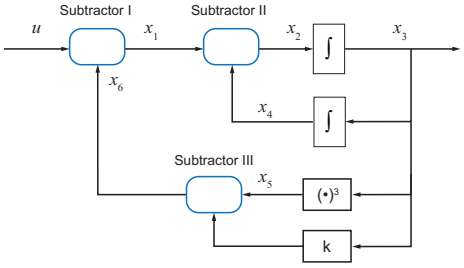

Fig. 5: Our results can be used to build oscillators, such as the Stan-Sepulchre oscillators [30], from scratch using abstract chemical reactions. We present a case study for the shown Lienard system which is a particular case of these oscillators.

$$A_1^{\pm}+A_2^{\pm} \underset{k_1}{\overset{k_2}{\rightleftharpoons}} A_3^{\pm} \xrightarrow{k_3} A_4^{\pm}+A_2^{\pm}$$
$$A_4^{\pm}+A_5^{\pm} \underset{k_4}{\overset{k_5}{\rightleftharpoons}} A_6^{\pm} \xrightarrow{k_6} A_1^{\pm}+A_5^{\pm},$$

where A_i ($i \in \{1,2,\ldots,6\}$) are biomolecular species (see [25] and [26]). This set of chemical reaction is better implemented through the set $\mathscr{S}_N$ of 12 abstract chemical reactions given in Fig. 4(B) which, in turn, can be implemented using 32 DNA implementation reactions, as illustrated in Fig. 4(C). The set $\mathscr{S}_N$ can be represented through the following set of ordinary differential equations [25]:

$$\frac{dx_2^+}{dt} = k_2 x_{C_1}^+ - k_3 x_2^+ x_e^+ \quad \text{and} \quad \frac{dx_2^-}{dt} = k_2 x_{C_1}^- - k_3 x_2^- x_e^-.$$

Now, $x_2 = x_2^+ - x_2^-$. Hence

$$\frac{dx_2}{dt} = k_2\left(x_{C_1}^+ - x_{C_1}^-\right) - k_3\left(x_2^+ x_e^+ - x_2^- x_e^-\right).$$

Without loss of generality, let $(x_2^+ x_e^+) = (x_2 x_e)^+$ and $(x_2^- x_e^-) = (x_2 x_e)^-$, as proposed in [25]. Then it can be verified that $\mathscr{S}_N$ is described by the following set of ODEs:

$$\begin{aligned}
\frac{dx_2}{dt} &= k_2 x_{C_1} - k_3 x_2 x_e,\\
\frac{dx_{C_1}}{dt} &= k_1 x_p x_1 - k_2 x_{C_1},\\
\frac{dx_{C_2}}{dt} &= k_3 x_2 x_e - k_4 x_{C_2}.
\end{aligned}$$

By varying the concentration of x_e, the slope of the Hill-type nonlinearity can be controlled, as is illustrated in Fig. 4(A).

Remark 6: While $\mathscr{S}_N$ can also be realised using Lemmas 4 and 5, the number of these abstract chemical reactions depends on the values of m_1 and m_2 — using the prescription of Lemma 5, a univariate polynomial of degree n is realised using $11n+3$ chemical reactions. Hence if $m_1 = m_2 = 1$, at least 28 reactions are required to realise $\mathscr{S}_N$. To obtain an ultrasensitive response, one typically requires higher values of m_1 and m_2, leading to a correspondingly higher number of chemical reactions, whereas our results in this section

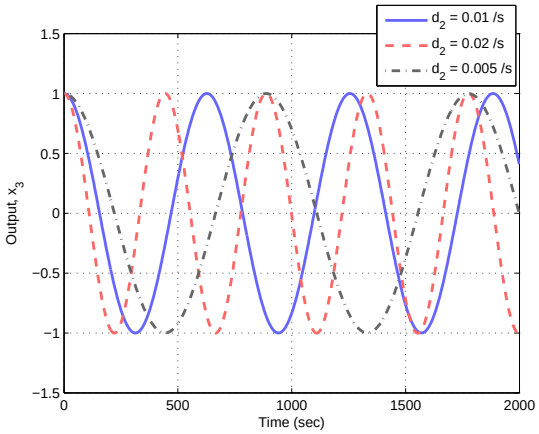

Fig. 6: Programmable oscillations produced by our biomolecular implementation of the Lienard system illustrated in Fig. 5. The period of the oscillations can be tuned by varying the reaction rate d_2.

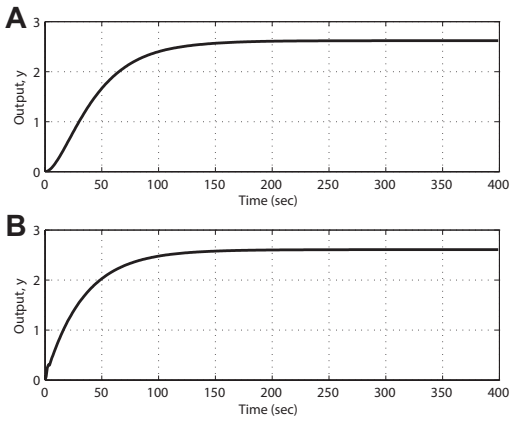

Fig. 7: Computations of univariate polynomials and rational functions using chemical reactions given by Lemma 6. (A) The ratio $y = u/z$ of two scalar-valued signals u and z is computed, where $u = 5.5$nM and $z = 2$nM. (B) The rational function $\frac{u+u^2}{z+z^2}$ is computed, where $u = 3.5$M and $z = 2$M — the values are set abnormally high so that the squared terms do not become vanishingly small.

facilitate such a qualitative input-output response using only 12 abstract chemical reactions — this number is independent of m_1 and m_2. □

IV. DNA IMPLEMENTATION

The framework relating chemical reactions to DNA strand displacement (DSD) has been well established (see e.g. [10], [17], [27]–[29]) and it essentially converts arbitrary chemical reaction network to a DSD model. In [17], the designed DNA-based scheme compile the unimolecular and bimolecular chemical reactions into strand displacement DNA-based chemistry to achieve the desired behaviour of the considered biomolecular system. Here, only the results are presented and interested readers are referred to [17] for details. The DNA implementation for a unimolecular reaction, $r_u : X_1 \xrightarrow{\delta} X_2 + X_3$ and a bimolecular, $r_b : X_1 + X_2 \xrightarrow{\delta} X_3$ are respectively given by Eqns. (6) and (7).

$$X_1 + G_i \xrightarrow{q_i} O_i \text{ and } O_i + T_i \xrightarrow{q_{max}} X_2 + X_3 \tag{6}$$

$$X_1 + L_i \underset{q_{max}}{\overset{q_i}{\rightleftharpoons}} H_i + B_i \ , \ X_2 + H_i \xrightarrow{q_{max}} O_i \text{ and } O_i + T_i \xrightarrow{q_{max}} X_3 \tag{7}$$

where G, O, T, L, H, B are auxiliary species with appropriate initial concentrations C_{max} while $q_i = \delta/C_{max}$ and q_{max} are the partial and maximum strand displacement rate respectively. This approached is followed throughout the paper (see e.g. Fig. 4(C)).

V. SIMULATION RESULTS

A. Implementation of a Stan-Sepulchre oscillator

Since the class of Stan-Sepulchre oscillators has a provably unique globally stable limit cycle [30], we decided to implement it to illustrate our results derived in Section III. We derived the set of abstract chemical reactions to implement the block-diagram shown in Fig. 5 which represents a Lienard system, which is a special case of the Stan-Sepulchre oscillators. The choice of k determines the bifurcation property of the system. We set $k = 1$.

This system comprises three subtractors, two integrators and a power component. Table I summarises the DNA implementation, CRNs and ODEs for the Lienard system. The resulting oscillations are shown in Fig. 6. In this simulation, we set $d_1 = 0.01$ /s, $d_3 = 1000$ /s, $k_{s1} = 0.003$ /s and $k_{s3} = 0.003$ /s. The frequency of oscillation can be tuned by varying the reaction rate d_2.

B. Implementation of a rational function

We next illustrate how to compute a ratio of two biomolecular signals u and z. Let $u = 5.5$nM and $z = 2$nM. The simulation result of $y = u/z$ is shown in Fig. 7(A). Table II summarises the DNA implementation, chemical reactions, and ODEs needed to implement this divider. Here, $K_d = \gamma_1 E_d = 10,000$, $\gamma_1 = 0.001$ /s, $\gamma_2 = 1$ /s and $k_{s1} = 0.1$ /s.

Next, we compute the ratio of two polynomials $y = \hat{u}/\hat{z}$, where $\hat{u} = u + u^2$ and $\hat{z} = z + z^2$. Let $u = 3.5$M and $z = 2$M — the values are set abnormally high so that u^2 and z^2 do not become vanishingly small. The simulation result is shown in Fig. 7(B). Table III summarises the DNA implementation, CRNs and ODEs for $\hat{u}$ and $\hat{z}$. In computing $\hat{u}$ and $\hat{z}$, $\gamma_1 = 1$ /s and $k_{s1} = 0.015$ /s. For subtractor, $k_{s2} = 1$ /s, gain, $K_d = \gamma_2 E_d = 10,000$, $\gamma_2 = 400$ /s and multiplier, $\gamma_3 = 0.03$ /s.

C. Implementation of PI+anti-windup controller for regulating the output of a static plant

The ultrasensitive response of a Hill-type nonlinearity well approximates the actuator saturation in many real-world applications. We now show how a biomolecular implementation of a PI+anti-windup controller can be implemented to counter such actuator saturations. The set of chemical reactions and its DNA implementation to realise a PI controller

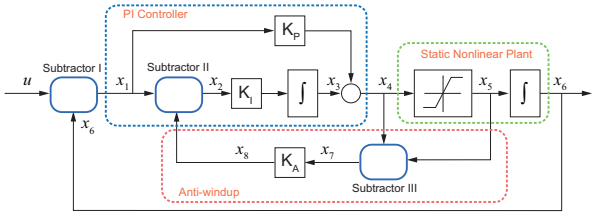

Fig. 8: Using our results on biomolecular implementation of saturation nonlinearities, a nonlinear PI+antiwindup controller can be synthesised rather than the linear PI controller synthesised in [10]. Simulation results for this feedback system are given in Fig. 9.

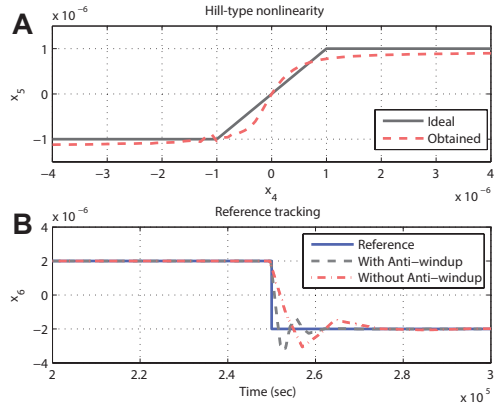

Fig. 9: (A) The Hill-type nonlinearity synthesised by our approach. (B) By adding the anti-windup to the PI controller, the tracking response becomes significantly faster.

has been derived in [19] and [10] respectively. Here, we extend those results by incorporating an anti-windup scheme in the presence of input saturation characterised by the Hill-type nonlinearity. A block diagram of the closed-loop system in which a static nonlinear plant is regulated by such a PI+anti-windup controller is shown in Fig. 8.

This system features three subtractors, two gain components, two integrators, one summation, and a Hill-type nonlinearity. Table IV shows the DNA implementation, chemical reactions, and ODEs for the saturation nonlinearity. The anti-windup comprises a subtractor and a gain. For the PI controller, the set of chemical reactions and its DNA implementation has been derived in [19] and [10]. Fig. 9(A) illustrates the Hill-type nonlinearity obtained with $k_1 = 400$ /mM/s, $k_2 = 0.004$ /s, $k_3 = 1350$ /mM/s, $k_4 = 0.0015$ /s and $x_e = 0.1$ nM. This yields an input saturation between $\pm 1 \times 10^{-6}$ /s. The ideal saturation curve and the realised saturation curve are shown in Fig. 9(A).

As shown in Fig. 9(B), the PI controller is sluggish in tracking the direction change in the reference signal whereas the PI+antiwindup controller is faster in tracking the reference signal, and also reduces the settling time from 26,000 seconds to 12,000 seconds.

VI. CONCLUSIONS

We have presented results on how abstract chemical reactions can be used to implement a number of nonlinear system theoretic operators such as multivariate polynomials, rational functions, and Hill-type nonlinearities. These results extend the architecture established for linear dynamic systems in [19]. We have shown how a combination of three elementary abstract idealised reactions, viz., catalysis, annihilation, and degradation can be used to realize these functions and have translated these chemical reactions into enzyme-free, entropy-driven DNA reactions. We have illustrated these results through three applications: (1) the Stan-Sepulchre oscillator, (2) computation of the ratio of two biomolecular signals and polynomials, and (3) regulation of a static nonlinear plant using a PI+anti-windup controller. We intend to follow the approach of [10], which uses [17], [27]-[29], to obtain the DNA strand displacement, genelet, and DNA Toolbox implementations of the results derived in this manuscript.

ACKNOWLEDGMENT

We gratefully acknowledge the financial support from EPSRC and BBSRC via research grants BB/M017982/1, the EPSRC Fellowship EP/M002187/1, and from the School of Engineering of the University of Warwick. We thank the reviewers for pointing out additional references.

REFERENCES

[1] A. Tamsir, J.J. Tabor, and C.A. Voigt, "Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'", *Nature*, vol. 469, pp. 212-215, 2011.

[2] T.S. Gardner, C.R. Cantor, and J.J. Collins, "Construction of a genetic toggle switch in *Escherichia coli*", *Nature*, vol. 403, pp. 339-342, 2000.

[3] M. Elowitz, and S. Leibler, "A synthetic oscillatory network of transcriptional regulator", *Nature*, vol. 403, pp. 335-338, 2000.

[4] S. Basu, Y. Gerchman, C.H. Collins, F.H. Arnold, and R. Weiss, "A synthetic multicellular system for programmed pattern formation", *Nature*, vol. 434, pp. 1130-1134, 2005.

[5] T. Sohka, R.A. Heins, R.M. Phelan, J.M. Greisler, C.A. Townsend, and M. Ostermeier, "An externally tunable bacteria band-pass filter", *Proceedings of National Academy of Science, USA*, vol. 106, no.25, pp. 10135-10140, 2009.

[6] K.E. Galloway, E. Franco, and C.D. Smolke, "Dynamically reshaping signaling networks to program cell fate via genetic controllers", *Science*, vol. 341, no. 6152, pp. 1235005, 2013.

[7] G. Seelig, D. Soloveichik, D.Y. Zhang, and E. Winfree, "Enzyme-free nucleic acid logic circuits", *Science*, vol. 314, no. 5805, pp. 1585-1588, 2006.

[8] D.Y. Zhang, A.J. Turberfield. B. Yurke, and E. Winfree, "Engineering entropy-driven reactions and networks catalyzed by DNA", *Science*, vol. 318, no. 5853, pp. 1121-1125, 2007.

[9] A. Padirac, T. Fujii, and Y. Rondelez, "Nucleic acids for the rational design of reaction circuits", *Current Opinion of Biotechnology*, vol. 24, issue 4, pp. 575-580, 2013.

[10] B. Yordanov, J. Kim, R.L. Petersen, A. Shudy, V.V. Kulkarni, and A. Philips, "Computational design of nucleic acid feedback control circuits", *ACS Synthetic Biology*, vol. 3, pp. 600-616, 2014.

[11] D.Y. Zhang, and G. Seelig, "Dynamic DNA nanotechnology using strand-displacement reactions", *Nature Chemistry*, vol.3, pp. 103-113, 2011.

[12] K. Montagne, R. Plasson, Y. Sakai, T. Fujii, and Y. Rondelez, "Programming an *in vitro* DNA oscillator using a molecular networking strategy", *Molecular Systems Biology*, vol. 7, 466, 2011.

[13] J. Kim, and E. Winfree, "Synthetic *in vitro* transcriptional oscillators", *Molecular Systems Biology*, vol. 7, 465, 2011.

[14] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig, "Programmable chemical controllers made from DNA", *Nature Nanotechnology*, vol. 8, pp. 755-762, 2013.

[15] T. Fujii, and Y. Rondelez, "Predator-prey molecular ecosystems", *ACS Nano*, vol. 7, no. 1, pp. 27-34, 2013.

[16] M. Weitz, J. Kim, K. Kapsner, E. Winfree, E. Franco, and F.C. Simmel, "Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator", *Nature Chemistry*, vol. 6, pp. 295-302, 2014.

[17] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal subtrate for chemical kinetics", *Proceedings of National Academy of Science, USA*, vol. 107, no. 12, pp. 5393-5398, 2010.

[18] W. Klonowski, "Simplifying principles for chemical and enzyme reaction kinetics", *Biophysical Chemistry*, vol. 18, no. 2, pp. 73-87, 1983.

[19] K. Oishi, and E. Klavins, "Biomolecular implementation of linear I/O systems", *IET Systems Biology*, vol. 5, issue 4, pp. 252-260, 2011.

[20] H. Buisman, H. ten Eikelder, P. Hilbers, and A. Liekens, "Computing algebraic functions with biochemical reaction networks", *Artificial Life*, vol. 15, no. 1, pp. 5-19, 2009.

[21] M. Feinberg. "Chemical reaction network structure and the stability of complex isothermal reactors I. The deficiency zero and deficiency one theorems", *Chemical Engineering Science*, vol. 42, no. 10, pp. 2229-2268, 1987.

[22] M. Feinberg. "Chemical reaction network structure and the stability of complex isothermal reactors II. Multiple steady states for networks of deficiency one", *Chemical Engineering Science*, vol. 43, no. 1, pp. 1-25, 1988.

[23] N. Barkai and S. Leibler. "Robustness in simple biochemical networks", *Nature*, vol. 387, pp. 913-917, 1997.

[24] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell. "The segment polarity network is a robust developmental module", *Nature*, vol. 406, pp. 188-192, 2000.

[25] R. Sawlekar, F. Montefusco, V. Kulkarni and D.G. Bates, "Biomolecular implementation of a Quasi Sliding Mode feedback controller based on DNA strand displacement reactions", *Proceedings of the 37th IEEE Engineering in Medicine and Biology Conference*, Milan, Italy, 2015.

[26] C. Gomez-Uribe, G.C. Verghese, and L.A. Mirny, "Operating regimes of signal cycles: statics, dynamics and noise filtering", *PLoS Computational Biology*, vol. 3, issue 12, pp. e246, 2007.

[27] F. Horn, and R. Jackson, "General mass action kinetics", *Archive for Rational Mechanics and Analysis*, vol. 47, no. 2, pp. 81, 1972.

[28] C. Thachuk, "Logically and physically reversible natural computing: a tutorial." *Reversible Computation. Springer Berlin Heidelberg*, pp. 247-262, 2013.

[29] D.Y. Zhang, "Towards domain-based sequence design for DNA strand displacement reactions." *DNA Computing and Molecular Programming. Springer Berlin Heidelberg*, pp. 162-175, 2011.

[30] G.-B. Stan, and R. Sepulchre, "Analysis of interconnected oscillators by dissipativity theory", *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 256-270, 2007.

APPENDIX I
CHEMICAL REACTIONS AND DNA IMPLEMENTATIONS

In this section, we note down all relevant sets of chemical reactions along with their DNA implementations for the case studies presented in Section IV.

DNA Implementation	Formal CRNs	ODEs
Stan-Sepulchre Oscillator		
Subtractor I		
$u^{\pm}+G_1^{\pm} \xrightarrow{q_1} \emptyset + O_1^{\pm}$ $O_1^{\pm}+T_1^{\pm} \xrightarrow{q_{max}} u^{\pm}+x_1^{\pm}$	$u^{\pm} \xrightarrow{k_{s1}} u^{\pm}+x_1^{\pm}$	
$x_6^{\pm}+G_2^{\pm} \xrightarrow{q_2} \emptyset + O_2^{\pm}$ $O_2^{\pm}+T_2^{\pm} \xrightarrow{q_{max}} x_6^{\pm}+x_1^{\mp}$	$x_6^{\pm} \xrightarrow{k_{s1}} x_6^{\pm}+x_1^{\mp}$	
$x_1^{\pm}+G_3^{\pm} \xrightarrow{q_3} \emptyset$	$x_1^{\pm} \xrightarrow{k_{s1}} \emptyset$	$\frac{dx_1}{dt} = k_{s1}(u - x_6 - x_1)$
$x_1^{+}+L_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_1+B_1$ $x_1^{-}+LS_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_1+BS_1$ $x_1^{-}+H_1 \xrightarrow{q_{max}} \emptyset$	$x_1^{+}+x_1^{-} \xrightarrow{\eta} \emptyset$	
Subtractor II		
$x_1^{\pm}+G_4^{\pm} \xrightarrow{q_4} \emptyset + O_4^{\pm}$ $O_4^{\pm}+T_4^{\pm} \xrightarrow{q_{max}} x_1^{\pm}+x_2^{\pm}$	$x_1^{\pm} \xrightarrow{d_1} x_1^{\pm}+x_2^{\pm}$	
$x_4^{\pm}+G_5^{\pm} \xrightarrow{q_5} \emptyset + O_5^{\pm}$ $O_5^{\pm}+T_5^{\pm} \xrightarrow{q_{max}} x_4^{\pm}+x_2^{\mp}$	$x_4^{\pm} \xrightarrow{d_2} x_4^{\pm}+x_2^{\mp}$	
$x_2^{\pm}+G_6^{\pm} \xrightarrow{q_6} \emptyset$	$x_2^{\pm} \xrightarrow{d_3} \emptyset$	$\frac{dx_2}{dt} = d_1x_1 - d_2x_4 - d_3x_2$
$x_2^{+}+L_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_2+B_2$ $x_2^{-}+LS_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_2+BS_2$ $x_2^{-}+H_2 \xrightarrow{q_{max}} \emptyset$	$x_2^{+}+x_2^{-} \xrightarrow{\eta} \emptyset$	
Forward Path Integrator		
$x_2^{\pm}+G_7^{\pm} \xrightarrow{q_7} \emptyset + O_7^{\pm}$ $O_7^{\pm}+T_7^{\pm} \xrightarrow{q_{max}} x_2^{\pm}+x_3^{\pm}$	$x_2^{\pm} \xrightarrow{1} x_2^{\pm}+x_3^{\pm}$	
$x_3^{+}+L_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_3+B_3$ $x_3^{-}+LS_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_3+BS_3$ $x_3^{-}+H_3 \xrightarrow{q_{max}} \emptyset$	$x_3^{+}+x_3^{-} \xrightarrow{\eta} \emptyset$	$\frac{dx_3}{dt} = x_2$
Feedback Path Integrator		
$x_3^{\pm}+G_8^{\pm} \xrightarrow{q_8} \emptyset + O_8^{\pm}$ $O_8^{\pm}+T_8^{\pm} \xrightarrow{q_{max}} x_3^{\pm}+x_4^{\pm}$	$x_3^{\pm} \xrightarrow{1} x_3^{\pm}+x_4^{\pm}$	
$x_4^{+}+L_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_4+B_4$ $x_4^{-}+LS_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_4+BS_4$ $x_4^{-}+H_4 \xrightarrow{q_{max}} \emptyset$	$x_4^{+}+x_4^{-} \xrightarrow{\eta} \emptyset$	$\frac{dx_4}{dt} = x_3$
Cubic Component		
$3x_3^{\pm}+G_9^{\pm} \xrightarrow{q_9} \emptyset + O_9^{\pm}$ $O_9^{\pm}+T_9^{\pm} \xrightarrow{q_{max}} 3x_3^{\pm}+x_5^{\pm}$	$3x_3^{\pm} \xrightarrow{\gamma_1} 3x_3^{\pm}+x_5^{\pm}$	
$x_5^{\pm}+G_{10}^{\pm} \xrightarrow{q_{10}} \emptyset$	$x_5^{\pm} \xrightarrow{\gamma_1} \emptyset$	
$x_5^{+}+L_5 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_5+B_5$ $x_5^{-}+LS_5 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_5+BS_5$ $x_5^{-}+H_5 \xrightarrow{q_{max}} \emptyset$	$x_5^{+}+x_5^{-} \xrightarrow{\eta} \emptyset$	$\frac{dx_5}{dt} = \gamma_1(x_3^3 - x_5)$
Subtractor III		
$x_5^{\pm}+G_{11}^{\pm} \xrightarrow{q_{11}} \emptyset + O_{11}^{\pm}$ $O_{11}^{\pm}+T_{11}^{\pm} \xrightarrow{q_{max}} x_5^{\pm}+x_6^{\pm}$	$x_5^{\pm} \xrightarrow{k_{s3}} x_5^{\pm}+x_6^{\pm}$	
$x_3^{\pm}+G_{12}^{\pm} \xrightarrow{q_{12}} \emptyset + O_{12}^{\pm}$ $O_{12}^{\pm}+T_{12}^{\pm} \xrightarrow{q_{max}} x_3^{\pm}+x_6^{\mp}$	$x_3^{\pm} \xrightarrow{k_{s3}} x_3^{\pm}+x_6^{\mp}$	
$x_6^{\pm}+G_{13}^{\pm} \xrightarrow{q_{13}} \emptyset$	$x_6^{\pm} \xrightarrow{k_{s3}} \emptyset$	$\frac{dx_6}{dt} = k_{s3}(x_5 - x_3 - x_6)$
$x_6^{+}+L_6 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_6+B_6$ $x_6^{-}+LS_6 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_6+BS_6$ $x_6^{-}+H_6 \xrightarrow{q_{max}} \emptyset$	$x_6^{+}+x_6^{-} \xrightarrow{\eta} \emptyset$	

TABLE I: DNA Implementation, CRNs and the corresponding ODEs for the implementation of Stan-Sepulchre oscillator. $\emptyset$ indicates the absence of products or waste. Here, $k_{s1} = 0.003$ /s, $d_1 = 0.01$ /s. $d_3 = 1000$ /s, $\gamma_1 = 1$ /s, $k_{s3} = 0.003$ /s and $d_2 = 0.005, 0.01, 0.02$ /s. $C_{max} = 1$ μM, $q_{max} = 1$ MM/s and $q_1 - q_3 = k_{s1}/C_{max}$, $q_4 = d_1/C_{max}$, $q_5 = d_2/C_{max}$, $q_6 = d_3/C_{max}$, $q_7 - q_8 = 1/C_{max}$, $q_9 - q_{10} = \gamma_1/C_{max}$, $q_{11} - q_{13} = k_{s3}/C_{max}$.

DNA Implementation	Formal CRNs	ODEs
Divider: Ratio of two species		
Subtractor		
$u^\pm + G_1^\pm \xrightarrow{q_1} \emptyset + O_1^\pm$ $O_1^\pm + T_1^\pm \xrightarrow{q_{max}} u^\pm + e^\pm$	$u^\pm \xrightarrow{k_{s1}} u^\pm + e^\pm$	$\frac{de}{dt} = k_{s1}(u - w - e)$
$e^\pm + G_2^\pm \xrightarrow{q_2} \emptyset + O_2^\pm$ $O_2^\pm + T_2^\pm \xrightarrow{q_{max}} w^\pm + e^\mp$	$e^\pm \xrightarrow{k_{s1}} e^\pm + e^\mp$	
$e^\pm + G_3^\pm \xrightarrow{q_3} \emptyset$	$e^\pm \xrightarrow{k_{s1}} \emptyset$	
$e^+ + L_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_1 + B_1$ $e^- + LS_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_1 + BS_1$ $e^- + H_1 \xrightarrow{q_{max}} \emptyset$	$e^+ + e^- \xrightarrow{\eta} \emptyset$	
Forward Path Gain		
$e^\pm + G_4^\pm \xrightarrow{q_4} \emptyset + O_4^\pm$ $O_4^\pm + T_4^\pm \xrightarrow{q_{max}} e^\pm + y^\pm$	$e^\pm \xrightarrow{\gamma_1 E_d} e^\pm + y^\pm$	$\frac{dy}{dt} = \gamma_1(E_d e - y)$
$y^\pm + G_5^\pm \xrightarrow{q_5} \emptyset$	$y^\pm \xrightarrow{\gamma_1} \emptyset$	
$y^+ + L_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_2 + B_2$ $y^- + LS_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_2 + BS_2$ $y^- + H_2 \xrightarrow{q_{max}} \emptyset$	$y^+ + y^- \xrightarrow{\eta} \emptyset$	
Multiplier		
$y^\pm + L_3^\pm \underset{q_{max}}{\overset{q_{3'}}{\rightleftharpoons}} H_3^\pm + B_3^\pm$ $z^\pm + H_3^\pm \xrightarrow{q_{max}} O_5^\pm$ $O_5^\pm + T_5^\pm \xrightarrow{q_{max}} w^\pm$	$y^\pm + z^\pm \xrightarrow{\gamma_2} w^\pm$	$\frac{dw}{dt} = \gamma_2(yz - w)$
$w^\pm + G_6^\pm \xrightarrow{q_6} \emptyset$	$w^\pm \xrightarrow{\gamma_2} \emptyset$	
$w^+ + L_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_4 + B_4$ $w^- + LS_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_4 + BS_4$ $w^- + H_4 \xrightarrow{q_{max}} \emptyset$	$w^+ + w^- \xrightarrow{\eta} \emptyset$	

TABLE II: DNA Implementation, CRNs and the corresponding ODEs for the implementation of divider. Here, $k_{s1} = 0.1$ /s, $\gamma_1 = 0.001$ /s, $K_d = \gamma_1 E_d = 10,000$, $\gamma_2 = 1$ /s. $C_{max} = 1$ μM, $q_{max} = 1$ MM/s, $q_1 - q_3 = k_{s1}/C_{max}$, $q_4 = \gamma_1 E_d / C_{max}$, $q_5 = \gamma_1 / C_{max}$, $q_{3'} = q_6 = \gamma_2 / C_{max}$.

DNA Implementation	Formal CRNs	ODEs
Computing $(\cdot) + (\cdot)^2$		
Quadratic Component		
$2x^\pm + G_1^\pm \xrightarrow{q_1} \emptyset + O_1^\pm$ $O_1^\pm + T_1^\pm \xrightarrow{q_{max}} 2x^\pm + x_{p,2}^\pm$	$2x^\pm \xrightarrow{\gamma_1} 2x^\pm + x_{p,2}^\pm$	$\frac{dx_{p,2}}{dt} = \gamma_1(x^2 - x_{p,2})$
$x_{p,2}^\pm + G_2^\pm \xrightarrow{q_2} \emptyset$	$x_{p,2}^\pm \xrightarrow{\gamma_1} \emptyset$	
$x_{p,2}^+ + L_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_1 + B_1$ $x_{p,2}^- + LS_1 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_1 + BS_1$ $x_{p,2}^- + H_1 \xrightarrow{q_{max}} \emptyset$	$x_{p,2}^+ + x_{p,2}^- \xrightarrow{\eta} \emptyset$	
Summation		
$x_{p,2}^\pm + G_3^\pm \xrightarrow{q_3} \emptyset + O_3^\pm$ $O_3^\pm + T_3^\pm \xrightarrow{q_{max}} x_{g,2}^\pm + y^\pm$	$x_{p,2}^\pm \xrightarrow{k_{s1}} x_{p,1}^\pm + y^\pm$	$\frac{dy}{dt} = k_{s1}(x_{p,2} + x_{p,1} - y)$
$x_{g,1}^\pm + G_4^\pm \xrightarrow{q_4} \emptyset + O_4^\pm$ $O_4^\pm + T_4^\pm \xrightarrow{q_{max}} x_{p,1}^\pm + y^\pm$	$x_{g,1}^\pm \xrightarrow{k_{s1}} x_{g,1}^\pm + y^\pm$	
$y^\pm + G_5^\pm \xrightarrow{q_5} \emptyset$	$y^\pm \xrightarrow{k_{s1}} \emptyset$	
$y^+ + L_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_2 + B_2$ $y^- + LS_2 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_2 + BS_2$ $y^- + H_2 \xrightarrow{q_{max}} \emptyset$	$y^+ + y^- \xrightarrow{\eta} \emptyset$	

TABLE III: DNA Implementation, CRNs and the corresponding ODEs for computing $\hat{u} = u + u^2$ and $\hat{z} = z + z^2$. Here, $\gamma_1 = 1$ /s, $k_{s1} = 0.015$ /s. $C_{max} = 1$ μM, $q_{max} = 1$ MM/s, $q_1 - q_2 = \gamma_1 / C_{max}$, $q_3 - q_5 = k_{s1}/C_{max}$.

DNA Implementation	Formal CRNs	ODEs
Hill-type nonlinearity		
$x_p^\pm + L_1^\pm \underset{q_{max}}{\overset{q_1}{\rightleftharpoons}} H_1^\pm + B_1^\pm$ $x_4^\pm + H_1^\pm \xrightarrow{q_{max}} \emptyset + O_1^\pm$ $O_1^\pm + T_1^\pm \xrightarrow{q_{max}} x_{c1}^\pm$	$x_p^\pm + x_4^\pm \xrightarrow{k_1} x_{c1}^\pm$	$\frac{dx_5}{dt} = k_2 x_{c1} - k_3 x_5 x_e$ $\frac{dx_{c1}}{dt} = k_1 x_p x_4 - k_2 x_{c1}$ $\frac{dx_{c2}}{dt} = k_3 x_5 x_e - k_4 x_{c2}$
$x_{c1}^\pm + G_2^\pm \xrightarrow{q_2} \emptyset + O_2^\pm$ $O_2^\pm + T_2^\pm \xrightarrow{q_{max}} x_4^\pm + x_5^\pm$	$x_{c1}^\pm \xrightarrow{k_2} x_4^\pm + x_5^\pm$	
$x_{c1}^+ + L_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_3 + T_3$ $x_{c1}^- + LS_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_3 + BS_3$ $x_{c1}^- + H_3 \xrightarrow{q_{max}} \emptyset$	$x_{c1}^+ + x_{c1}^- \xrightarrow{\eta} \emptyset$	
$x_5^+ + L_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_4 + T_4$ $x_5^- + LS_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_4 + BS_4$ $x_5^- + H_4 \xrightarrow{q_{max}} \emptyset$	$x_5^+ + x_5^- \xrightarrow{\eta} \emptyset$	
$x_5^\pm + L_5^\pm \underset{q_{max}}{\overset{q_5}{\rightleftharpoons}} H_5^\pm + B_5^\pm$ $x_e^\pm + H_5^\pm \xrightarrow{q_{max}} \emptyset + O_5^\pm$ $O_5^\pm + T_5^\pm \xrightarrow{q_{max}} x_c^\pm$	$x_5^\pm + x_e^\pm \xrightarrow{k_3} x_{c2}^\pm$	
$x_{c2}^\pm + G_6^\pm \xrightarrow{q_6} \emptyset + O_6^\pm$ $O_6^\pm + T_6^\pm \xrightarrow{q_{max}} x_p^\pm + x_e^\pm$	$x_{c2}^\pm \xrightarrow{k_4} x_p^\pm + x_e^\pm$	
$x_{c2}^+ + L_6 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_6 + B_6$ $x_{c2}^- LS_6 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_6 + BS_6$ $x_{c2}^- + H_6 \xrightarrow{q_{max}} \emptyset$	$x_{c2}^+ + x_{c2}^- \xrightarrow{\eta} \emptyset$	
$x_p^+ + L_7 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_7 + B_7$ $x_p^- + LS_7 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_7 + BS_7$ $x_p^- + H_7 \xrightarrow{q_{max}} \emptyset$	$x_p^+ + x_p^- \xrightarrow{\eta} \emptyset$	
Anti-windup gain		
$x_7^\pm + G_7^\pm \xrightarrow{q_7} \emptyset + O_7^\pm$ $O_7^\pm + T_7^\pm \xrightarrow{q_{max}} x_7^\pm + x_8^\pm$	$x_7^\pm \xrightarrow{\gamma_2 K_A} x_7^\pm + x_8^\pm$	$\frac{dx_8}{dt} = \gamma_2(K_A x_7 - x_8)$
$x_8^\pm + G_8^\pm \xrightarrow{q_8} \emptyset$	$x_8^\pm \xrightarrow{\gamma_2} \emptyset$	
$x_8^+ + L_8 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_8 + B_8$ $x_8^- + LS_8 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_8 + BS_8$ $x_8^- + H_8 \xrightarrow{q_{max}} \emptyset$	$x_8^+ + x_8^- \xrightarrow{\eta} \emptyset$	

TABLE IV: DNA Implementation, CRNs and the corresponding ODEs for the implementation of Hill-type nonlinearity and anti-windup gain. Here, $k_1 = 400$ /mM/s, $k_2 = 0.004$ /s, $k_3 = 1350$ /mM/s, $k_4 = 0.0015$ /s, $x_e = 0.1$ nM, $\gamma_2 = 0.9$ /s, $K_A = 156$. $C_{max} = 1$ μM, $q_{max} = 1$ MM/s, $q_1 = k_1/C_{max}$, $q_2 = k_2/C_{max}$, $q_5 = k_3/C_{max}$, $q_6 = k_4/C_{max}$, $q_7 = \gamma_2 K_A / C_{max}$, $q_8 = \gamma_2 / C_{max}$.