Biomolecular implementation of nonlinear system theoretic operators

Mathias Foo¹, Rucha Sawlekar¹, Jongmin Kim², Declan G. Bates¹, Guy-Bart Stan³, and Vishwesh Kulkarni¹

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.

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 utilizing 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
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

$$X_1 + X_2 \xrightarrow{\delta_1} X_3 + X_4,$$

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, $\delta_1$ denotes the forward reaction rate and $\delta_2$ 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:

$$x_i^+ \xrightarrow{K} x_i^+ + x_o^+, \quad x_i^- \xrightarrow{\emptyset} \emptyset \quad \text{and} \quad x_o^+ + x_o^- \xrightarrow{\emptyset} \emptyset,$$

where $K$ is the gain. This operation is implemented, at the steady state, using the following set of abstract chemical reactions:

**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:

$$x_i^+ + x_d^+ \xrightarrow{\emptyset} (x_i^+ + x_o^+), \quad x_i^- + x_d^- \xrightarrow{\emptyset} (x_i^- + x_o^-), \quad x_o^+ + x_o^- \xrightarrow{\emptyset} \emptyset.$$

**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:

$$x_i^+ \xrightarrow{K} x_i^+ + x_o^+, \quad x_i^- \xrightarrow{\emptyset} x_d^- + x_o^-, \quad x_o^+ + x_o^- \xrightarrow{\emptyset} \emptyset.$$
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, \( \frac{dx}{dt} = \gamma(Kx_i - x_o) \). Likewise, the ODEs for the summation and integrator operations are given by \( \frac{dx}{dt} = \gamma(x_i + x_d - x_o) \) and \( \frac{dx}{dt} = Kx_i \), respectively.

III. MAIN RESULTS

A. Polynomials and rational functions

**Lemma 4:** [Polynomial \( x' \)]

Let \( x_{p,i} \) denote the polynomial of degree \( n \) defined as \( x_{p,i} = x' \). 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:

\[
\begin{align*}
\underbrace{x_i^+ + \cdots + x_i^+}_{\text{i times}} & \rightarrow \underbrace{x_i^+ + \cdots + x_i^+ + x_i^+}_{\text{i times}}, \\
\underbrace{x_{p,i}^-}_{\text{i times}} & \rightarrow \emptyset, \\
\underbrace{x_{p,i}^+ + x_{p,i}^-}_{\text{i times}} & \rightarrow \emptyset
\end{align*}
\]

(1)

where \( \eta \) 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:

\[
\frac{dx_{p,i}}{dt} = \gamma_p \left( \prod_{i=1}^{n} x_i^+ - x_{p,i}^- \right).
\]

(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.

**Remark 2:** It may be noted that the constant \( a_0 \) can be realised as \( 0 \rightarrow x_{g,0}^+ \) and \( 0 \rightarrow x_{g,0}^- \) 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: \( x^+ + y^+ \rightarrow z^+ + z^+ + z^- + z^- \rightarrow \emptyset \) and \( z^+ + z^- \rightarrow \emptyset \). The resulting ODE is \( \frac{dz}{dt} = \gamma_p(xy - z) \) 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 5:** [Rational function]

Consider the system \( \mathcal{F}_D \) 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 \( e = u - cy \) and \( y = K_d e \). Substituting the former equation into the latter one and rearranging the variables, we get \( y = K_d(u - yz) = \frac{K_d u}{1 + K_d z} \). 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 \( N(x) = 1 - (x^{m_1}/(\alpha + x^{m_2})) \) 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 occurring 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.

![Diagram](image)

### Ordinary Differential Equations

\[
\begin{align*}
\frac{dx_1}{dt} &= k_2 x_1^+ - k_3 x_2^+ x_e^- & \frac{dx_2}{dt} &= k_2 x_1^- - k_3 x_2^- x_e^+ \\
\end{align*}
\]

Now, \( x_2 = x_2^+ - x_2^- \). Hence

\[
\frac{dx_2}{dt} = k_2 (x_1^+ - x_1^-) - k_3 (x_2^+ x_e^- - x_2^- x_e^+).
\]

Without loss of generality, let \((x_1^+ x_e^-) = (x_2 x_e)^+\) and \((x_2^- x_e^+) = (x_2 x_e)^-\), as proposed in [25]. Then it can be verified that \( S_N \) is described by the following set of ODEs:

\[
\begin{align*}
\frac{dx_1}{dt} &= k_2 x_1^- - k_3 x_2 x_e, \\
\frac{dx_1}{dt} &= k_1 x_p x_1 - k_2 x_1, \\
\frac{dx_2}{dt} &= k_3 x_2 x_e - k_4 x_2.
\end{align*}
\]

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 \( S_N \) 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 \( S_N \). 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 $d_2$.

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 an unimolecular reaction, $r_u: X_1 \xrightarrow{\delta_1} X_2 + X_3$ and a bimolecular, $r_b: X_1 + X_2 \xrightarrow{\delta_2} X_3$ are respectively given by Eqns. (6) and (7).

\[
X_1 + G_1 \xrightarrow{q_1} O_1 \quad \text{and} \quad O_1 + T_1 \xrightarrow{q_{\text{max}}} X_2 + X_3
\]

\[
X_1 + L_1 \xrightarrow{q_1} H_1 + B_1 \quad \text{and} \quad X_2 + H_2 \xrightarrow{q_{\text{max}}} O_1 \quad \text{and} \quad O_1 + T_1 \xrightarrow{q_{\text{max}}} X_3
\]

where $G$, $O$, $T$, $L$, $H$, $B$ are auxiliary species with appropriate initial concentrations $C_{\text{max}}$ while $q_1 = \delta/C_{\text{max}}$ and $q_{\text{max}}$ are the partial and maximum strand displacement rate respectively. This approach is followed throughout the paper (see e.g. Fig. 4C).

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 $d_1 = 0.01$ $\text{s}$, $d_3 = 1000$ $\text{s}$, $k_{b1} = 0.003$ $\text{s}$ and $k_{b3} = 0.003$ $\text{s}$. 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 $u = 5.5 \text{nM}$ and $z = 2 \text{nM}$. The simulation result of $y = u/z$ is shown in Fig. 7A. Table II summarises the DNA implementation, chemical reactions, and ODEs needed to implement this divider. Here, $K_d = \gamma_1 E_d = 10,000$, $\gamma_1 = 0.001$ $\text{s}$, $\gamma_2 = 1$ $\text{s}$ and $k_{b1} = 0.1$ $\text{s}$.

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 $u = 3.5 \text{M}$ and $z = 2 \text{M}$ — 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. 7B. Table III summarises the DNA implementation, CRNs and ODEs for $\hat{u}$ and $\hat{z}$. In computing $\hat{u}$ and $\hat{z}$, $\gamma_1 = 1$ $\text{s}$ and $k_{b1} = 0.015$ $\text{s}$. For subtractor, $k_{s2} = 1$ $\text{s}$, gain, $K_d = \gamma_2 E_d = 10,000$, $\gamma_1 = 400$ $\text{s}$ and multiplier, $\gamma_3 = 0.03$ $\text{s}$.

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
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-Sepulcreh 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

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.

<table>
<thead>
<tr>
<th>DNA Implementation</th>
<th>Formal CRNs</th>
<th>ODEs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stan-Sepulchre Oscillator</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtractor I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$w^+ + G_{1}^{+} \xrightarrow{k_{1}} s_{5} \xrightarrow{k_{6}} + O_{1}^{+}$</td>
<td>$dx_{5} \xrightarrow{k_{6}} = k_{3}(u - s_{6} - s_{1})$</td>
<td>$\frac{dw}{dt} = r_{3}(u - s_{6} - s_{1})$</td>
</tr>
<tr>
<td>$O_{1}^{+} + T_{1}^{+} \xrightarrow{k_{2}} s_{5} \xrightarrow{k_{6}} + O_{1}^{+}$</td>
<td>$dx_{5} \xrightarrow{k_{6}} = k_{3}(u - s_{6} - s_{1})$</td>
<td>$\frac{dw}{dt} = r_{3}(u - s_{6} - s_{1})$</td>
</tr>
<tr>
<td>$S_{5}^{+} + G_{1}^{+} \xrightarrow{k_{7}} \theta \xrightarrow{k_{13}} + O_{1}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{5}^{+} + s_{5} \xrightarrow{k_{6}} + x_{5}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtractor II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{5}^{+} + G_{1}^{+} \xrightarrow{k_{13}} \theta \xrightarrow{k_{13}} + O_{1}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$O_{1}^{+} + T_{1}^{+} \xrightarrow{k_{2}} s_{5} \xrightarrow{k_{6}} + O_{1}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{5}^{+} + s_{5} \xrightarrow{k_{6}} + x_{5}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subtractor III</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{5}^{+} + G_{1}^{+} \xrightarrow{k_{13}} \theta \xrightarrow{k_{13}} + O_{1}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$O_{1}^{+} + T_{1}^{+} \xrightarrow{k_{2}} s_{5} \xrightarrow{k_{6}} + O_{1}^{+}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{5}^{+} + s_{5} \xrightarrow{k_{6}} + x_{5}^{+}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE I:** DNA Implementation, CRNs and the corresponding ODEs for the implementation of Stan-Sepulchre oscillator. $\theta$ indicates the absence of products or waste. Here, $k_{3} = 0.003 \text{ /s}$, $d_{1} = 0.01 \text{ /s}$, $d_{3} = 1000 \text{ /s}$, $\gamma_{1} = 1 \text{ /s}$, $k_{3} = 0.003 \text{ /s}$ and $d_{2} = 0.005, 0.01, 0.02 \text{ /s}$. $C_{\max} = 1 \text{ µM}$, $q_{\max} = 1 \text{ MM/s}$ and $q_{1} - q_{3} = k_{4}/C_{\max}$, $q_{4} = d_{1}/C_{\max}$, $q_{5} = d_{2}/C_{\max}$, $q_{6} = d_{3}/C_{\max}$, $q_{7} = d_{4}/C_{\max}$, $q_{8} = 1/C_{\max}$, $q_{9} - q_{10} = \gamma_{1}/C_{\max}$, $q_{11} - q_{13} = k_{3}/C_{\max}$. 

**REFERENCES**


\textbf{Table II: DNA Implementation, CRNs and the corresponding ODEs for the implementation of divider.} Here, \(k_{a1} = 0.1 \text{ /s}, \gamma_1 = 0.001 \text{ /s}, K_d = \gamma_1 E_d = 10,000, \gamma_2 = 1 \text{ /s}, C_{\text{max}} = 1 \mu\text{M}, q_{\text{max}} = 1 \text{ MM/s}, q_1 - q_3 = k_{a1}/C_{\text{max}}, q_4 = \gamma_1 E_d/C_{\text{max}}, q_5 = q_1/C_{\text{max}}, q_3 - q_5 = \gamma_2/C_{\text{max}}.

\textbf{Table III: DNA Implementation, CRNs and the corresponding ODEs for computing } \ddot{u} = u + u^2 \text{ and } \ddot{z} = z + z^2. \text{ Here, } \gamma_1 = 1 \text{ /s}, k_{a1} = 0.015 \text{ /s}, C_{\text{max}} = 1 \mu\text{M}, q_{\text{max}} = 1 \text{ MM/s}, q_1 - q_2 = \gamma_1/C_{\text{max}}, q_3 - q_5 = k_{a1}/C_{\text{max}}.

\textbf{Table IV: DNA Implementation, CRNs and the corresponding ODEs for the implementation of Hill-type nonlinearity and anti-windup gain.} Here, \(k_1 = 400 \text{ /mM/s}, k_2 = 0.004 \text{ /s}, k_3 = 1350 \text{ /mM/s}, k_4 = 0.0015 \text{ /s}, x_0 = 0.1 \text{ nM}, \gamma_1 = 0.9 \text{ /s}, K_A = 156. C_{\text{max}} = 1 \mu\text{M}, q_{\text{max}} = 1 \text{ MM/s}, q_1 = k_1/C_{\text{max}}, q_2 = k_2/C_{\text{max}}, q_5 = k_3/C_{\text{max}}, q_6 = k_4/C_{\text{max}}, q_7 = \gamma_2 K_A/C_{\text{max}}, q_8 = \gamma_2/C_{\text{max}}.