Programming Dynamic Nonlinear Biomolecular Devices using DNA Strand Displacement Reactions

by

Rucha Anil Usha Sawlekar

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

School of Engineering

September 2016

THE UNIVERSITY OF WARWICK

To my parents, Anil and Usha Sawlekar,

To my grandfather, Jaysingh Pawar,

To my partner Sumeet Satpute,

...and to the fond and loving memory of my uncle, Prof. Sunil Sawlekar.

Acknowledgments

First of all, I express my profound gratitude to my guide Prof. Declan G. Bates for his valuable guidance and unflinching support. I was fortunate to have a supervisor like him, without which it would not have been possible for me to complete the project qualitatively and on time. I also wish to acknowledge with deep gratitude the significant help extended to me by Dr. Vishwesh Kulkarni in this endeavour.

I take this opportunity to thank Dr. Mathias Foo and Dr. Francesco Montefusco, for all the crucial discussions and providing important tips and literature that contributed extensively towards my learning process and academic publications. I am also grateful to all my fabulous friends and lab-mates in our Biomedical Engineering lab for making my time in lab so wonderful and memorable.

During my time at Coventry, I have been fortunate to have fantastic friends who are more like family. Atul Kamath, Anup Das, Nida Siddiqui and Francesco Montefusco have always been there for me and without whose encouragement this journey would have been very difficult. I thank all of them cordially for their friendship, care and our backpacking together!

In conclusion, I thank my family for everything. I thank my late uncle Prof. Sunil Sawlekar who certainly would have been happier and proud to see the completion of my doctoral studies. I thank my granfather Jaysingh Pawar for all his love, blessings and care. I thank my best friend and life partner Sumeet Satpute for all his love, unflinching support and companionship. Whether near or far, it has always been and will be an awesome journey together! Finally, I must acknowledge my incredible parents Anil and Usha Sawlekar for their endless love, efforts and being a warm, close-knit, and super-supportive family. They have loved and supported me constantly, regardless of my faults and flaws. They have been with me through all the good and worse times. Their contribution to my life and career remains incomparable.

Declarations

I declare that the work presented in this thesis is my own except where stated otherwise, and was carried out entirely at the University of Warwick, during the period of September 2013 to September 2016, under the valuable supervision of Prof. Declan G. Bates and significant help from Dr. Vishwesh Kulkarni. The research reported here has not been submitted, either wholly or in part, in this or any other academic institution for admission to a higher degree.

Some parts of the work reported and other work not reported in this thesis have been published. Further parts of this work will be submitted for publication in due course.

Abstract

Recent advances in DNA computing have greatly facilitated the design of biomolecular circuitry based on toehold-mediated *DNA strand displacement* (DSD) reactions. The 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. In this thesis, new results are presented on how *chemical reaction networks* (CRNs) can be used as a programming language to implement commonly used linear and nonlinear system theoretic operators that can be further utilised in combination to form complex biomolecular circuits. Within the same framework, the design of an important class of nonlinear feedback controller, i.e. a *quasi sliding mode* (QSM) feedback controller, is proposed. The closed loop response of the nonlinear QSM controller is shown to outperform a traditional linear *proportional+integrator* (PI) controller by facilitating much faster tracking response dynamics without introducing overshoots in the transient response. The resulting controller is highly modular and is less affected by retroactivity effects than standard linear designs. An important issue to consider in this design process for synthetic circuits is the effect of biological and experimental uncertainties on the functionality and reliability of the overall circuit. In the case of biomolecular feedback control circuits, such uncertainties could lead to a range of adverse effects, including achieving wrong concentration levels, sluggish performance and even instability. In this thesis, the robustness properties of two biomolecular feedback controllers; PI and QSM, subject to uncertainties in the experimentally implemented rates of their underlying chemical reactions, and to variations in accumulative time delays in the process to be controlled, are analysed. The simulation results show that the proposed QSM controller is significantly more robust against investigated uncertainties, highlighting its potential as a practically implementable biomolecular feedback controller for future synthetic biology applications. Finally, the thesis presents new results on the design of biomolecular feedback controllers using the set of chemical reactions underlying covalent modification cycles.

Contents

Acknowledgments iii

Declarations iv

Abstract v

List of Tables ix

List of Figures x

Abbreviations xiii

Chapter 1 Introduction 1

1.1 Research motivation and the context of molecular programming . . 1

1.2 DNA based circuits in synthetic biology 2

1.3 Summary of contributions and thesis organisation 4

Chapter 2 Chemical Reaction Networks and DNA Strand Displacement 9

2.1 Chemical reaction networks 9

2.1.1 Mapping biomolecular reactions to differential equations . . 11

2.1.2 Realising dynamical systems using CRNs 13

2.1.3 Modelling covalent modification cycles 15

2.2 Implementation using toehold mediated DNA strand displacement . 19

2.2.1 The architecture and mechanism 20

2.2.2 Software design tool: Visual DSD 21

Chapter 3 Biomolecular Implementation of Nonlinear System Theoretic Operators **29**

3.1 Background results on linear operators . . . 29

3.1.1 Gain, summation and integration . . . 30

3.2 Nonlinear operators . . . 32

3.2.1 Multiplication, division and polynomial operators . . . 33

3.3 Forming functional circuits using linear/nonlinear operators . . . 39

3.3.1 Fractional exponent . . . 39

3.3.2 Absolute value . . . 41

3.3.3 Logarithm of arbitrary base . . . 45

Chapter 4 Design and Implementation of a Biomolecular Quasi Sliding Mode Controller **49**

4.1 Saturation nonlinearity . . . 50

4.2 Implementing Nonlinear Feedback Controllers . . . 52

4.2.1 Biomolecular Quasi-sliding mode controller . . . 52

4.2.2 Nonlinear process to be controlled . . . 55

4.2.3 Subtractor . . . 56

4.2.4 PI Controller . . . 57

4.3 Performance comparison of the linear and nonlinear controllers . . . 61

4.3.1 Simulation results with modularity . . . 61

4.4 Retroactivity . . . 63

4.4.1 Simulation results with retroactivity . . . 65

4.5 Conclusions . . . 66

Chapter 5 Robustness Analysis of Biomolecular Controllers to Parametric and Time Delay Uncertainties **68**

5.1 Robustness analysis . . . 68

5.2 System description and methodology . . . 69

5.2.1 QSM controller . . . 70

5.2.2 PI controller . . . 71

5.2.3 Process to be controlled . . . 72

5.2.4 Subtractor . . . 73

5.3 Simulation results . . . 73
5.4 Conclusions . . . 77

Chapter 6 Exploiting the Dynamic Properties of a Covalent Modification Cycle for Nonlinear Controller Design **80**
6.1 Designing a covalent modification cycle controller . . . 81
6.1.1 Chemical reactions . . . 81
6.1.2 System of ordinary differential equations . . . 84
6.2 Simulation results . . . 85
6.2.1 Performance analysis of controllers with a linear process . . . 85
6.2.2 Performance analysis of the controllers with a nonlinear process . . . 89
6.3 Flexible input-output mapping improves robustness . . . 92
6.4 Conclusions . . . 95

Chapter 7 Conclusions and Future Work **97**

List of Tables

3.1 DNA Implementation reactions, CRNs and the corresponding ODEs for the implementation of linear and nonlinear system theoretic operators 37

4.1 DNA implementation reactions, CRNs and relevant ODEs for the cloed-loop feedback system components 59
4.2 Process to be controlled — parameter values 62

5.1 Step response characteristics and worst-case parameter ranges for the PI controller. 78
5.2 Step response characteristics and worst-case parameter ranges for the QSM controller. 79

6.1 Step response characteristics and worst-case parameter ranges for the PI and CMC controllers + the linear process. 87
6.2 Step response characteristics and worst-case parameter ranges for PI and CMC controllers + nonlinear process. 90

List of Figures

1.1 Approach to design synthetic biochemical devices illustrated conceptually . . . 3

2.1 DNA reactions $\rightarrow$ chemical reaction (or CRN) $\rightarrow$ ODE . . . 12
2.2 The minimal representation of signal x . . . 14
2.3 The covalent modification cycle and its steady state behaviour of the four regimes . . . 15
2.4 DNA strand displacement reaction mechanism . . . 20
2.5 DNA implementation of catalysis reaction $X_1^{\pm} \rightarrow X_2^{\pm} + X_3^{\pm}$ . . . 24
2.6 DNA implementation of catalysis reaction $X^{\pm} \rightarrow X^{\pm} + Y^{\pm}$ . . . 25
2.7 DNA implementation of bimolecular reaction $X_1^{\pm} + X_2^{\pm} \rightarrow X_3^{\pm}$ . . . 26
2.8 DNA implementation of degradation reaction $X^{\pm} \rightarrow \phi$ . . . 27
2.9 DNA implementation of annihilation reaction $X^{+} + X^{-} \rightarrow \phi$ . . . 28

3.1 A block diagram representing scalar gain . . . 30
3.2 A block diagram representing summation or subtraction . . . 31
3.3 A block diagram representing integration . . . 32
3.4 A block diagram representing multiplication . . . 33
3.5 A block diagram representation of the feedback system $\mathscr{S}_D$ that computes the ratio $y = u/z$ . . . 34
3.6 The input-output system derived to compute the univariate polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$ . . . 35
3.7 Computing fractional exponent using Newton-Raphson block . . . 40
3.8 Computation of $S^{\frac{m}{n}}$ shown with two examples . . . 41
3.9 Computing absolute value using combination of operators . . . 42

3.10 Computing the absolute value using covalent modification cycle regimes . . . 43
3.11 Simulation results for the computation of absolute value . . . 44
3.12 Comparison between ideal and simulated signal-transducing response 45
3.13 Computation of logarithm of arbitrary base . . . 46
3.14 Simulation results of natural logarithm and logarithm of arbitrary base . . . 47

4.1 Input-output characteristic curve producing saturation nonlinearity using: (a) covalent modification cycle, (b) chemical reactions . . . 50
4.2 Saturation nonlinearity using chemical reactions . . . 51
4.3 A prototype embedded biomolecular closed loop feedback control system. . . . 52
4.4 Comparing input-output characteristics of an ideal SMC and QSM controller . . . 54
4.5 Closed loop tracking response of QSM controller with parameters and their nominal values . . . 61
4.6 Closed loop responses comparison of quasi and ideal sliding mode controllers . . . 62
4.7 Closed loop tracking response of PI controller with parameters and their nominal values . . . 63
4.8 Retroactivity effect for a system S . . . 64
4.9 Closed loop tracking response comparison of QSM and PI controller after accounting for retroactivity effects . . . 66

5.1 The biomolecular closed-loop feedback control system with the accumulative process time delay. . . . 70
5.2 Comparing system performance of QSM controller with the PI controller for $\tau = 1000$s . . . 74
5.3 System performance with the PI controller for 20% uncertainty in parameters . . . 75
5.4 System performance with the QSM controller for 20% uncertainty in parameters . . . 75

5.5 System performance with the PI controller for 50% uncertainty in parameters 76
5.6 System performance with the QSM controller for 50% uncertainty in parameters 76

6.1 The signal-transducing mapping resembles the steady-state input-output mapping of a PI controller 81
6.2 A prototype embedded biomolecular closed loop feedback control system with CMC or PI controller. 82
6.3 Simulation results with PI controller + linear process and CMC controller + linear process 88
6.4 Simulation results with PI controller + nonlinear process and CMC controller + nonlinear process 91
6.5 The mapping of steady-state input-output signals of the PI controller and the CMC controller when controlling the linear process . 93
6.6 The mappings of steady-state input-output signals of the PI controller and CMC controller (A) 96

Abbreviations

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
CRISPR	Clustered regularly interspaced short palindromic repeats
CRNs	Chemical reaction networks
DSD	DNA strand displacement
ODEs	Ordinary differential equations
PI	Proportional+integrator
SMC	Sliding mode controller
QSM	Quasi sliding mode controller
CMC	Covalent modification cycle
nt	Nucleotides
ssDNA	Single stranded DNA
mRNA	Messenger RNA
HBB	Haemoglobin beta
LTI	Linear time-invariant
SN	Saturation nonlinearity
MIMO	Multi input multi output

Chapter 1

Introduction

By 'life,' we mean a thing that can nourish itself and grow and decay.

- Aristotle

1.1 Research motivation and the context of molecular programming

The criteria for life, as stated by the brilliant geneticist Norman Horowitz is: *Life possesses the properties of replication, catalysis, and mutability*; the complexity of multicellular organisms starts with a single cell; followed by its division, growth, reproduction, and so on. Thus, living organisms are complex but fascinating phenomena. The development of some of these complex structures performs a particular function - the heart pumps blood through the body, leaves use the energy of sunlight by photosynthesis, fish gills help underwater breathing, etc.

If the development of functions in nature is understood as a program then the underlying chemical reactions can be seen as their programming language. There may exist more than one set of such chemical reactions (combinations of different types of chemical reactions) or more than one set of parameter values - that is able to achieve the desired biomolecular system response. However, one should select a suitable set of chemical reactions having a minimum number of reactions which is preferred or rather feasible for *in vivo/in vitro* implementations. Also, a suitable set of parameter values can be chosen for simulations, keeping in mind that these values are well within the practically achievable physical limits, for eg., DNA or mRNA concentration levels, reaction rates etc.

Now, to treat biomolecular systems and software analogously, one must establish common features. First, any slight alteration in software code can affect the program execution, and eventually the outcome. This is similar to gene mutation that is carried out by altering the nucleotide sequence of DNA so that even a small change in sequence causes a significant difference in growth. This change in reactant and resulting product DNA can be formulated in terms of chemical reactions. For example, developmental mutations in the fruit fly Drosophila melanogaster can result in the growth of a leg out of the head instead of an antenna [1], or genetic disorders in humans can cause autism [2]. Second, programs can be copied and edited to get the desired results. During cell replication, cells duplicate their genetic material to produce two identical daughter cells [3; 4]. Precise and targeted genome editing is possible with a recently developed tool based on the *clustered regularly interspaced short palindromic repeats* (CRISPR) system that relies on the protein Cas9, from bacteria. In *Streptococcus pyogenes*, the re-engineered Cas9 could find and cut the DNA target specified by the guide RNA [5]. Later, this technique was used for genome editing in human cells, for the first time in [6–8].

This ability and possibility to program cells that eventually might be used to design even more complex biochemical systems in the future is the essence of motivation for this thesis. Similar to using software languages in computer coding, the aim is to use a structured set of instructions in order to “program” DNA molecules *in vitro* or *in vivo*.

1.2 DNA based circuits in synthetic biology

Synthetic biology is an interdisciplinary research field at the engineering-biology interface, that encompasses extremely diverse domains such as biotechnology, molecular biology, nanotechnology, biophysics, computer engineering, mathematics and so on. Several of the proposed industrial and biomedical applications of synthetic biology require the ability to precisely and robustly control the behaviour of syn-

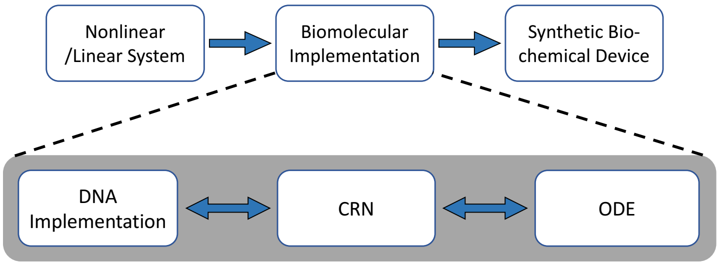


Figure 1.1: Synthetic biomolecular devices should ideally have the capability to produce nonlinear input-output behaviour. The results illustrated in this thesis show how biomolecular implementations of nonlinear operators can be realized by approximating the *DNA strand displacement* (DSD) reactions to the *chemical reaction networks* (CRNs) and then converting into their equivalent *ordinary differential equations* (ODEs).

thetic circuits or devices at a biomolecular level [9; 10]. A fundamental aim of synthetic biology is thus to achieve the capability to design and implement robust embedded biomolecular feedback control circuits [11]. An appropriate modelling and design framework for tackling this problem is provided by *chemical reaction networks* (CRNs), which represent a convenient and concise approach to modelling chemical and biological processes as well as an effective tool for the analysis of their behaviour from both deterministic [12; 13] and stochastic [14] viewpoints. It is possible to approximate any chemical reaction by a set of suitably designed *DNA strand displacement* (DSD) reactions and vice versa [15–17] (discussed precisely further in Chapter 2, Section 2.2.2). The conversion from CRNs to *ordinary differential equations* (ODEs) also works bidirectionally because, applying the mass action law to unimolecular CRNs results in linear ODEs whereas, when applied to bimolecular CRNs, it results in nonlinear ODEs (discussed in Chapter 2, Section 2.1.1). This gives flexibility in understanding the trade-offs between such conversions and helps improving the system design. This approach 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. 1.1.

The direct use of nucleic acids for performing computation has emerged as a promising approach for addressing such problems [17; 18]. The nucleotides sequence of nucleic acid 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 DSD [19], DNA enzymes [20] and RNA enzymes [21], and has been used for the modelling and implementation of various nucleic-acids-based circuits such as feedback controllers [22], predator-prey dynamics [23] and also transcriptional oscillators [24]. So far, several synthetic devices have been designed and implemented *in vivo* using protein expression and gene regulation mechanisms, for example, logic gates [25], memory elements [26], oscillators [27], filters [28; 29] and controllers of cellular differential processes [30].

1.3 Summary of contributions and thesis organisation

The scientific contributions as presented in each chapter in this thesis are summarised below.

Chapter 2 presents the background of CRNs, and explains the DSD mechanism and its underlying kinetics. The mapping of chemical reactions to ODEs, by means of the generalised mass action law is explained, as it is an important step in order to to build a mathematical model that can approximate the dynamic behaviour of the biomolecular system. The methodology adopted for the system design is followed by an overview of the toehold mediated DSD mechanism.

Chapter 3 describes the use of chemical reactions to implement a number of linear and nonlinear system theoretic operators such as gain, integrator, summation, polynomial functions, rational functions, and so on. The signals in the mathematical models considered here are biomolecular, i.e. they represent chemical concentrations of species. Now, being a physical quantity, chemical concentrations cannot be negative. To overcome this difficulty, [31] represented the signal in a mathematical model as a difference between the concentrations of two species labelled as '$+$' and '$-$', and using this approach described how a number of linear operators could be designed and implemented in DNA-based chemistry. Chapter 3 reviews this methodology, and then provides new results on the design of nonlinear system theoretic operators, and their application to form functional circuits.

Previous work on the implementation of feedback controllers using DNA has focussed on the design of linear time-invariant systems only, e.g. the *proportional+integrator* (PI) controllers described in [17; 31; 32]. This approach fails to exploit the inherent potential of biomolecular circuits to implement nonlinear dynamical systems [15; 22; 33], and also requires the use of additional circuitry to overcome the wind-up effects associated with the integrator action. In Chapter 4, the approach of [17; 31; 32] is extended to allow the implementation of nonlinear feedback controllers. The focus is on a well-known type of nonlinear controller called a *sliding mode controller* (SMC), whose strong performance and robustness characteristics have been widely recognised in more traditional control engineering applications [34; 35]. From sliding mode control theory, a perfect SMC can be represented by a relay nonlinearity (see [34; 36; 37]). To avoid a number of theoretical and practical issues with the implementation of such discontinuous switches, in engineering practice SMC's are usually implemented as *quasi sliding mode* (QSM) controllers, i.e. continuous/smooth approximations of the discontinuous SMC. In Chapter 4, it is shown how a set of irreversible chemical reactions can provide a biomolecular implementation of a nonlinear QSM controller. It is also shown how the kinetics of the required chemical reactions can then be implemented as enzyme-free, entropy/enthalpy driven DNA reactions [38], using strand displacement as an elementary computational mechanism. To evaluate the performance properties of the QSM controller, the closed-loop response of the QSM controller is compared to that of a traditional linear PI controller. Feedback control systems usually comprise several individual modules or components connected together to perform a particular function. Although the components behave as expected when designed individually; they can affect each others' behaviour when interconnected. This interesting phenomenon is referred to as *retroactivity* and it has been extensively studied in [39–41]. For the closed-loop simulations of the proposed QSM and PI controllers, the retroactivity is quantified to evaluate its effect on the closed-loop dynamics.

An important requirement for any embedded bimolecular controller is that its design provides robustness to various forms of uncertainty and variability that could arise in its final implementation in DNA. In Chapter 5, the focus is on two important sources of such uncertainty - variability in the reaction rate constants of the chemical reactions underlying the closed-loop control system under consideration, and uncertain time delays in the biomolecular process to be controlled. In practice, experimental biologists are rarely able to specify the reaction rates of chemical reactions exactly, and additionally, as highlighted in [31], unregulated chemical devices or leaky expression can potentially affect production and degradation rates and subsequently alter the behaviour of the designed components. There are also many reasons why time delays may need to be included in CRN models of biomolecular processes, since this avoids cataloging potentially large numbers of intermediate species and their reactions, in favour of describing the dynamic relationships between the concentrations of key species. As a result, fewer concentration variables will generally be required, thus simplifying the overall circuit design problem. Chapter 5 comprises a robustness analysis comparison of the proposed QSM and PI controller, in the presence of uncertainty. The DSD mechanism involves a set of chemical reactions with DNA strands as the input and output. These reactions may exhibit time delays in producing the final product DNA strand, especially if the number of reactions is large. Considering the number of reactions involved in the complete feedback circuit, their accumulative delay is included at the output.

Covalent modification cycles are ubiquitous motifs in cellular signalling. Although such signalling cycles are implemented via a highly concise set of chemical reactions, they have been shown to be capable of producing multiple distinct input-output mapping behaviours such as, ultrasensitive, hyperbolic, signal-transducing and threshold-hyperbolic. Chapter 6 explores more generally how the set of chemical reactions underlying covalent modification cycles can be exploited for the design of synthetic biomolecular feedback controllers with strong performance and robustness properties. The different input-output characteristics of such cycles in their different operating regimes are shown to allow the implementation of different types of feedback controllers. In chapter 6 an overview of the design of a proposed nonlinear *covalent modification cycle* (CMC) controller is provided, and a performance/robustness comparison with a linear PI controller in the presence of parametric uncertainty is described.

Finally, some conclusions and a discussion on the directions for future research in this area are offered in Chapter 7.

The elements of the MatLab simulations codes may be obtained upon request by contacting the author - *rucha.sawlekar@uni.lu*. Parts of chapters 3 to 6 have been published or are currently submitted for publication in a number of research papers, for which the author contributed as leading or co-author:

Publications

Journal papers

1. R. Sawlekar, F. Montefusco, V. Kulkarni, and D. G. Bates, "Implementing nonlinear feedback controllers using DNA strand displacement reactions", in *IEEE Transactions on NanoBioscience*, DOI:10.1109/TNB.2016.2560764, 2016.

2. M. Foo, R. Sawlekar and D.G. Bates, "Exploiting the dynamic properties of covalent modification cycles for the design of synthetic analog biomolecular circuitry", in *Journal of Biological Engineering*, 10.1, DOI:10.1186/s13036-016-0036-1, 2016.

Conference papers

1. 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", in proceedings of the *37th annual IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 949–952, August, 2015.

2. M. Foo, R. Sawlekar, J. Kim, G.-B. Stan, D. G. Bates and V. Kulkarni, "Biomolecular implementation of nonlinear system theoretic operators", in proceedings of the *15th annual IEEE European Control Conference (ECC)*, pages 1824–1831, June, 2016.

3. R. Sawlekar, M. Foo and D. G. Bates, "Robustness analysis of DNA-based biomolecular feedback controllers to parametric and time delay uncertainties", in proceedings of the *12th annual IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 548–551, October, 2016.

Presentations

1. Oral presentation: R. Sawlekar, V. Kulkarni and D. G. Bates, “Implementing nonlinear systems using nucleic acids”, *UK Automatic Control Council, PhD Presentation Showcase*, London, November, 2015. (Prize winner for best presentation)

Chapter 2

Chemical Reaction Networks and DNA Strand Displacement

The chapter begins by presenting the terminology for using *chemical reaction networks* (CRNs) as a programming language in order to design system theoretical operators and circuits. It is followed by a discussion of the methodology for representing species concentrations as *signals*, in order to represent a biomolecular system as a *mathematical model*. Finally, the mechanism of *DNA strand displacement* (DSD) is introduced as a candidate architecture to implement the devices at a molecular level.

2.1 Chemical reaction networks

A system of chemical reactions can be represented mathematically as a CRN [12; 42–44]. A CRN consists of a finite set of reactions, having specified reaction rates, that includes a set of reactants and products. A general set of reversible and irreversible chemical reactions may be written as:

$$X_1 + X_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} X_3 + X_4, \tag{2.1a}$$

$$X_4 \xrightarrow{k_3} X_4 + X_5, \tag{2.1b}$$

$$X_5 \xrightarrow{k_4} \phi. \tag{2.1c}$$

where, X_i denote chemical species. In (2.1a), the quantities on the left hand side of the reaction (X_1 and X_2) are called '*reactants*', and the quantities on the right hand side (X_3 and X_4) are called '*products*'. Based on the number of reactants present, the reactions are either unimolecular (only one reactant), bimolecular (two reactants) or multimolecular (more than two reactants present). Accordingly, (2.1a) is referred to as a bimolecular reaction while (2.1b) and (2.1c) are unimolecular reactions. Furthermore, (2.1a) represents a reversible reaction, i.e. reactants can be re-produced by means of products; with k_1, k_2 denoting the forward and backward reaction rates, respectively. Reactions (2.1b) and (2.1c) are irreversible, with k_3 being the catalysis reaction rate and k_4 a degradation rate that turns X_5 into an inert or waste product (ϕ). Different types of chemical reactions that are used to construct the functional operators in this thesis are explained further in Section 2.2.

The order of the reaction corresponds to the number of molecules reacting with each other simultaneously. Thus, (2.1b) is an example of a first order reaction whereas (2.1a) is a second order reaction. The reaction rates k_1 to k_4 are the number of occurrences of the respective reaction, per unit time, per unit volume, divided by Avogadro's number (6.023×10^{23}) [12]. Also, the units of the reaction rates for first and second order reactions are different. The first order reaction rates k_3 and k_4 have units of reciprocal seconds ($/s$) and second order reaction rates k_1 and k_2 have units ($/M/s$) [4].

A reaction network, $\{\mathcal{S},\mathcal{C},\mathcal{R}\}$ can be specified by the set of species $\mathcal{S}$ that indicates the molecules undergoing the series of chemical reactions; the set of complexes $\mathcal{C}$ that indicates the linear combinations of the used and produced species in the reactions, and by $\mathcal{R}$, the series of reactions taking place. Accordingly, for (2.1), the reaction network is:

$$Species,\ \ \mathcal{S} = \{X_1,\ X_2,\ X_3,\ X_4,\ X_5\}, \tag{2.2a}$$

$$Complexes,\ \ \mathcal{C} = \{X_1+X_2,\ X_3+X_4,\ X_4,\ X_4+X_5,\ X_5,\ \phi\}, \tag{2.2b}$$

$$Reactions,\ \ \mathcal{R} = \{X_1+X_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} X_3+X_4,\ X_4 \xrightarrow{k_3} X_4+X_5,\ X_5 \xrightarrow{k_4} \phi\}. \tag{2.2c}$$

Usually biochemical reactions are characterised by the property referred to as *stoichiometry*, which specifies the reactants and products participating in the reaction and the molar ratios in which they are consumed or produced. Accordingly, the stoichiometric coefficient of a chemical species is positive if it is produced in the forward reaction, and negative otherwise. A stoichiometric matrix representation of the reaction network (2.1) can be given as:

$$
\begin{array}{cc}
 & \begin{array}{cccc} k_1 & k_2 & k_3 & k_4 \end{array} \\
N = & \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{array}{c} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{array}
\end{array}
$$

The stoichiometry matrix for the 5-species 4-reaction scheme has five rows and four columns so that each row belongs to a particular species and each column belongs to a reaction rate. In this way, it determines the proportions in which chemical elements combine or are produced and the weight relations (number of molecules) in any chemical reaction.

2.1.1 Mapping biomolecular reactions to differential equations

As discussed previously, CRNs can be considered both as a programming language and as a mathematical representation of how the molar concentrations of chemical species evolve over time. In other words, CRNs outline how certain species - in the context of this thesis, DNA strands - can react to yield some product species in a finite time. Fig. 2.1 shows the mapping of DNA elementary reactions to CRNs and to *ordinary differential equations* (ODEs), that can be simulated to observe the system dynamics. The key factor in describing the kinetics is to assign a rate function to each of the chemical reactions present in the network and once it is done, one can write a system of differential equations. Note that, for species X_1 and X_2 in (2.1a), both the species are used at the rate k_1 and produced at the rate k_2 in

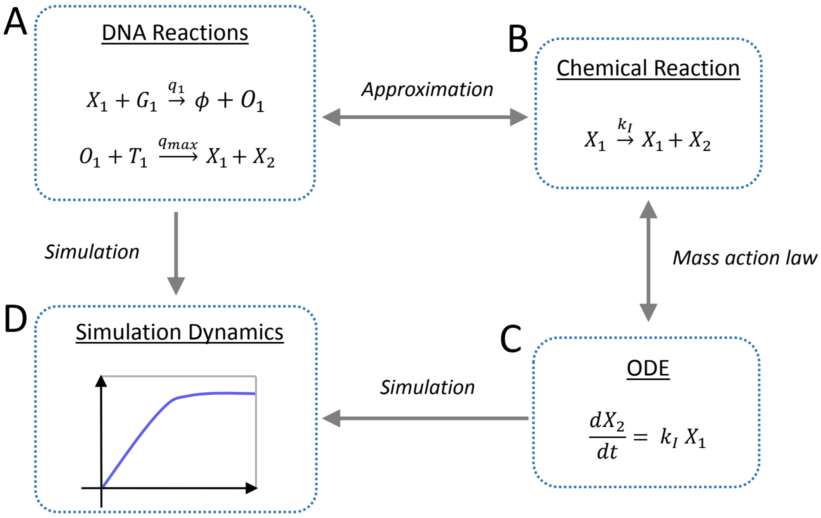


Figure 2.1: DNA reactions → chemical reaction (or CRN) → ODE: (A) Elementary DNA reactions where, X_1, X_2 are signal species and the remaining are the auxiliary species. (B) Chemical reaction can be approximated from DNA reactions and vice versa. (C) Chemical reaction can be approximated by ODE using mass action kinetics. (D) Simulation of ODE to evaluate the behaviour of the DNA reactions.

the reaction. Then, for X_1 and X_2, the ODE can be written as:

$$\frac{dX_1}{dt} = \frac{dX_2}{dt} = -k_1X_1X_2 + k_2X_3X_4, \tag{2.3}$$

where, this formalism of applying reaction rates to the species activities is referred as the *law of mass action* [45]. It is applied to a set of chemical reactions in order to generate a system of ODEs. Now, species X_3 in (2.1a) is produced at the rate k_1 and used at the rate k_2 in the reaction, so that:

$$\frac{dX_3}{dt} = k_1X_1X_2 - k_2X_3X_4, \tag{2.4}$$

By continuing in this way, a system of ordinary differential equations is obtained for each of the species as follows:

$$\frac{dX_1}{dt} = -k_1X_1X_2 + k_2X_3X_4, \tag{2.5a}$$
$$\frac{dX_2}{dt} = -k_1X_1X_2 + k_2X_3X_4, \tag{2.5b}$$
$$\frac{dX_3}{dt} = k_1X_1X_2 - k_2X_3X_4, \tag{2.5c}$$
$$\frac{dX_4}{dt} = k_1X_1X_2 - k_2X_3X_4, \tag{2.5d}$$
$$\frac{dX_5}{dt} = k_3X_4 - k_4X_5. \tag{2.5e}$$

Once the system of ODEs is obtained, the dynamic behaviour of the species can then be simulated and modified by means of their respective concentrations and reaction rates.

2.1.2 Realising dynamical systems using CRNs

In this section, existing results are summarised from [31; 46] to show how different systems are realised in the CRN framework. The notations are adopted from [31] and [15].

Signals and system theory involves input signals that are processed through a mathematical model to generate the output signals. The signals in biological systems are often molecular concentrations. While signals in systems theory can yield both positive and negative values, molecular concentrations, being a physical quantity, can only take non-negative values. Thus, when a negative value of any signal appears in the simulations, it looks contradictory to present the signal as a concentration.

To resolve this difficulty, the approach proposed in [31] is adopted and accordingly any signal x is represented as the difference in concentration of two chemical species, namely x^+ and x^-, such that:

$$x = x^+ - x^-, \qquad \text{or specifically,} \qquad x(t) = x^+(t) - x^-(t) \tag{2.6}$$

where, x^+ and x^- can be physically represented as two individual DNA molecules and their superscripts $+$, $-$ denote only the labels. This way two DNA molecules of the same domain (same nucleotides sequence) can be presented conveniently. Now, x can be treated as a signal in the mathematical model, and physically measured as a concentration that is obtained as a difference in the concentration of two chemical species x^+ and x^-. Based on this method, if the value of x appears negative in simulations - it can now be justified. Note that, relation $x = x^+ + x^-$ is not considered here as only the approach in (2.6) validates the negative values of a signal (or concentration) appearing in the simulations.

The relation in (2.6) is illustrated through an example given in Fig. 2.2

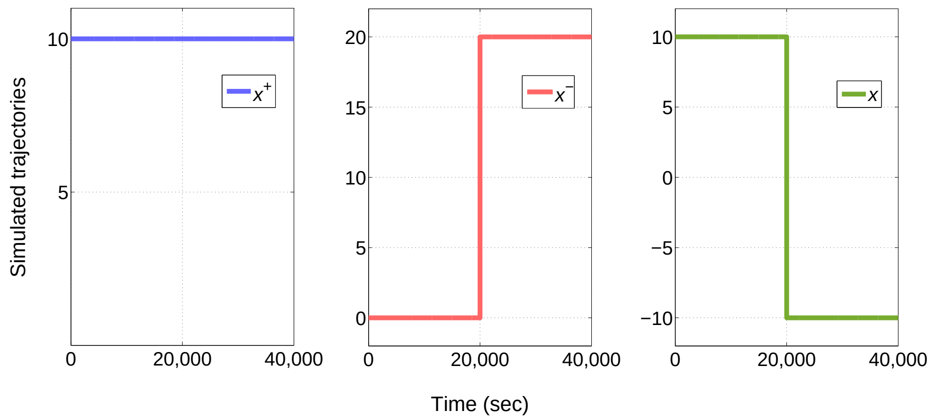


Figure 2.2: The square wave of signal x (right) is modelled by using two instantaneous additions of chemical species at $t = 0$ sec and $t = 20,000$ sec. At time $t = 0$ sec, only DNA strand x^+ is added i.e. $x^+(0) = 10$nM (left). This constitutes the resulting response of x (right) for $t \in [0, 20,000]$ sec, as given in (2.7). At time $t = 20,000$ sec, DNA strand x^- is added i.e. $x^-(20,000) = 20$nM (center). As stated in (2.8), it results in the response of x for $t \in [20,000, 40,000]$ sec.

where, the simulated trajectories represent concentrations of DNA strands x^+, x^- and the resulting value of x over the time, $t = [0, 40,000]$s. In Fig. 2.2, a DNA strand x^+ with a concentration of 10 nM is added initially at time $t = 0$s, in the absence of DNA strand x^- in the solution; then the value of x is given as:

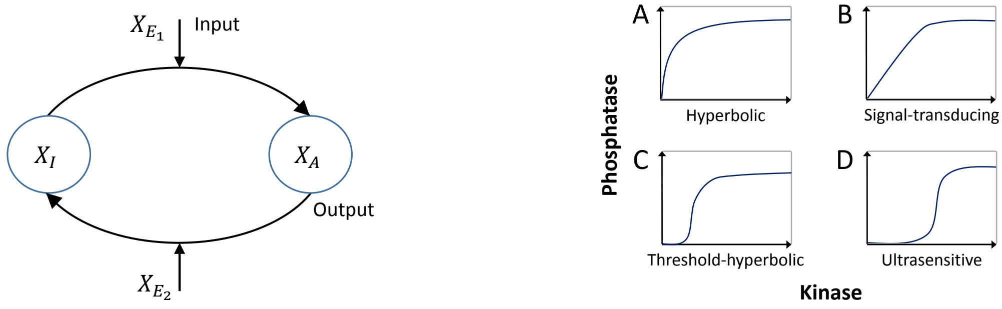
$$x(0) = x^+(0) - x^-(0) = 10 - 0 = 10; \quad \ldots \text{[for t} \in \text{[0, 20,000] s]} \tag{2.7}$$

When the DNA strand x^- is added with a concentration of 20 nM at time $t = 20,000$s, the minimal representation of signal x is given as:

$$\begin{aligned} x(20,000) &= x^+(20,000) - x^-(20,000) \\ &= 10 - 20 = -10; \qquad \ldots \text{[for t} \in \text{[20,000, 40,000] s]} \end{aligned} \tag{2.8}$$

Thus, the simulated trajectory for dynamics of any signal x in Fig. 2.2 follows (2.7) and (2.8), showing how the value of x changes from positive to negative over the time $t = [0, 40,000]$s.

The next section presents an example on how this methodology can be employed in order to derive a mathematical model of a particular biomolecular system known as a covalent modification cycle.



(a) (b)

Figure 2.3: (a) The covalent modification cycle regulated by their phosphorylation/dephosphorylation [59]: inactive (X_I) or active (X_A) proteins can be activated or deactivated by means of the two enzymatic species, namely kinase (X_{E_1}) and phosphatase (X_{E_2}), respectively. (b) Steady state behaviour of the four regimes of the covalent modification cycle.

2.1.3 Modelling covalent modification cycles

A classical example of analog cellular signal processing is signal transduction, which cells rely on for sensing and responding to various internal or external perturbations (see e.g. [47–50]). When perturbations occur, cells perceive them through receptors, which subsequently activate appropriate signalling pathways or cascades in order for cells to communicate with each other and respond accordingly [51–53]. One of the most ubiquitous motifs seen in cell signalling cascades is the cycle of covalent modification (see e.g. [54; 55] and references therein). Examples of this signalling cycle include phosphorylation/dephosphorylation cascades [56], DNA methylation [57] and monoclonal antibodies [58]. The covalent modification cycle is implemented via a highly concise set of chemical reactions, and this set of chemical reactions has been subjected to rigorous mathematical analysis. As shown in [54], under certain conditions, the set of chemical reactions describing the covalent modification cycle can exhibit highly sigmoidal input-output characteristics, generating the so-called ultrasensitive response. In [59], the authors systematically examine the covalent modification cycle in Fig. 2.3(a) and its steady-state responses as in Fig. 2.3(b), to time varying perturbations and demonstrate the existence of two additional types of responses, termed signal-transducing and threshold-hyperbolic.

The chemical reactions from [59] are given by:

$$X_{E_1} + X_I \underset{k_{d_1}}{\overset{k_{a_1}}{\rightleftharpoons}} X_{C_A} \xrightarrow{k_{u_1}} X_A + X_{E_1} \tag{2.9a}$$

$$X_A + X_{E_2} \underset{k_{d_2}}{\overset{k_{a_2}}{\rightleftharpoons}} X_{C_I} \xrightarrow{k_{u_2}} X_I + X_{E_2} \tag{2.9b}$$

where, k_{a_1}, k_{a_2} are the respective substrate-enzyme association rates and k_{d_1}, k_{d_2} are the respective substrate-enzyme dissociation rates. k_{u_1} and k_{u_2} are the forward and backward catalytic reaction rates. This covalent modification cycle in (2.9), operates in the following manner; the two step irreversible reaction (2.9a) is catalysed by the enzyme (kinase) X_{E_1} that reacts with the inactive protein X_I to produce the active protein output, X_A. Similarly, reaction (2.9b) is catalysed by the enzyme (phosphatase) X_{E_2} that reacts with active protein X_A to produce the inactive protein, X_I. X_{C_1} and X_{C_2} are the intermediate products in (2.9a) and (2.9b), respectively that represent the bound concentration of the reactants as, $X_{C_A} = X_{E_1} : X_I$ and $X_{C_I} = X_A : X_{E_2}$.

Now, an individual system of 14 chemical reactions that can generate the identical distinct steady state response regimes as illustrated in Fig. 2.3(b), is given below. Despite the ability to mimic the response regimes, the CRNs given below remain different than the CRNs of a standard covalent modification cycle (2.9), due to the formalism (2.6), adopted from [17; 31]. Strictly speaking, any reaction given below, with superscript $\pm$ and $\mp$ should be decomposed into their individual '$+$' and '$-$' components. For example, any reaction, $X_1^{\pm} + X_2^{\pm} \xrightarrow{k} X_3^{\pm}$ should be read as, $X_1^{+} + X_2^{+} \xrightarrow{k} X_3^{+}$ and $X_1^{-} + X_2^{-} \xrightarrow{k} X_3^{-}$. However, for brevity and to avoid overloading of reactions, they are written in the compact form:

$$X_{in}^{\pm} + X_p^{\pm} \xrightarrow{k_1} X_{C_1}^{+}, \tag{2.10a}$$

$$X_{in}^{\mp} + X_p^{\pm} \xrightarrow{k_1} X_{C_1}^{-}, \tag{2.10b}$$

$$X_{C_1}^{\pm} \xrightarrow{k_2} X_{out}^{\pm} + X_{in}^{\pm}, \tag{2.10c}$$

$$X_{C_1}^{+} + X_{C_1}^{-} \xrightarrow{\eta} \phi, \tag{2.10d}$$

$$X_{out}^{+} + X_{out}^{-} \xrightarrow{\eta} \phi, \tag{2.10e}$$

$$X_{out}^{\pm} + X_e \xrightarrow{k_3} X_{C_2}^{\pm}, \tag{2.10f}$$

$$X_{C_2}^{\pm} \xrightarrow{k_4} X_p^{\pm} + X_e, \tag{2.10g}$$

$$X_{C_2}^{+} + X_{C_2}^{-} \xrightarrow{\eta} \phi, \tag{2.10h}$$

$$X_p^{+} + X_p^{-} \xrightarrow{\eta} \phi. \tag{2.10i}$$

where, k_1, k_3 are binding and k_2, k_4 are the catalytic reaction rates and η is the annihilation rate. Note that, for (2.10a) the product is only $X^+_{C_1}$, rather than $X^{\pm}_{C_1}$ while for (2.10b) the product is only $X^-_{C_1}$. Also, system in (2.9) and its corresponding species are to be treated separately than the system in (2.10) as it can be noted that (2.10) is the modified version of (2.10) according to the adopted methodology, (2.6).

It can be checked that applying the law of mass action to (2.10), in order to obtain the system of ODEs, results in:

For X_{C_1}

$$\frac{dX^+_{C_1}}{dt} = k_1X^+_{in}X^+_p + k_1X^-_{in}X^-_p - k_2X^+_{C_1} - \eta X^+_{C_1}X^-_{C_1},$$
$$\frac{dX^-_{C_1}}{dt} = k_1X^-_{in}X^+_p + k_1X^+_{in}X^-_p - k_2X^-_{C_1} - \eta X^+_{C_1}X^-_{C_1},$$

From (2.6), $x = x^+ - x^-$. Hence:

$$\frac{dX_{C_1}}{dt} = \left[\frac{dX^+_{C_1}}{dt} - \frac{dX^-_{C_1}}{dt}\right],$$

$$\begin{aligned}
&= [k_1X^+_{in}X^+_p + k_1X^-_{in}X^-_p - k_2X^+_{C_1} - \eta X^+_{C_1}X^-_{C_1}]\\
&\quad -[k_1X^-_{in}X^+_p + k_1X^+_{in}X^-_p - k_2X^-_{C_1} - \eta X^+_{C_1}X^-_{C_1}],\\
&= k_1X^+_{in}X^+_p + k_1X^-_{in}X^-_p - k_2X^+_{C_1} - \eta X^+_{C_1}X^-_{C_1}\\
&\quad -k_1X^-_{in}X^+_p - k_1X^+_{in}X^-_p + k_2X^-_{C_1} + \eta X^+_{C_1}X^-_{C_1},\\
&= k_1X^+_{in}(X^+_p - X^-_p) - k_1X^-_{in}(X^+_p - X^-_p)\\
&\quad -k_2(X^+_{C_1} - X^-_{C_1}),\\
&= k_1X^+_{in}X_p - k_1X^-_{in}X_p - k_2X_{C_1},\\
&= k_1X_p(X^+_{in} - X^-_{in}) - k_2X_{C_1},\\
&= k_1X_pX_{in} - k_2X_{C_1},
\end{aligned}$$

$$\boxed{\frac{dX_{C_1}}{dt} = k_1X_{in}X_p - k_2X_{C_1}.}$$

For X_p

$$\frac{dX^+_p}{dt} = -k_1X^+_{in}X^+_p - k_1X^-_{in}X^+_p + k_4X^+_{C_2} - \eta X^+_pX^-_p,$$
$$\frac{dX^-_p}{dt} = -k_1X^-_{in}X^-_p - k_1X^+_{in}X^-_p + k_4X^-_{C_2} - \eta X^+_pX^-_p,$$

From (2.6), $x = x^+ - x^-$. Hence:

$$\frac{dX_p}{dt} = \left[\frac{dX^+_p}{dt} - \frac{dX^-_p}{dt}\right],$$

$$\begin{aligned}
&= [-k_1X^+_{in}X^+_p - k_1X^-_{in}X^+_p + k_4X^+_{C_2} - \eta X^+_pX^-_p]\\
&\quad -[-k_1X^-_{in}X^-_p - k_1X^+_{in}X^-_p + k_4X^-_{C_2} - \eta X^+_pX^-_p],\\
&= -k_1X^+_{in}X^+_p - k_1X^-_{in}X^+_p + k_4X^+_{C_2} - \eta X^+_pX^-_p\\
&\quad +k_1X^-_{in}X^-_p + k_1X^+_{in}X^-_p - k_4X^-_{C_2} - \eta X^+_pX^-_p,\\
&= -k_1X^+_{in}(X^+_p - X^-_p) - k_1X^-_{in}(X^+_p - X^-_p)\\
&\quad +k_4(X^+_{C_2} - X^-_{C_2}),\\
&= -k_1X^+_{in}X_p - k_1X^-_{in}X_p + k_4X_{C_2},\\
&= -k_1X_p(X^+_{in} - X^-_{in}) + k_4X_{C_2},\\
&= -k_1X_pX_{in} + k_4X_{C_2},
\end{aligned}$$

$$\boxed{\frac{dX_p}{dt} = -k_1X_{in}X_p + k_4X_{C_2}.}$$

For X_{out}	For X_{C_2}
$\frac{dX_{out}^+}{dt} = k_2X_{C_1}^+ - k_3X_{out}^+X_e - \eta X_{out}^+X_{out}^-,$ $\frac{dX_{out}^-}{dt} = k_2X_{C_1}^- - k_3X_{out}^-X_e - \eta X_{out}^+X_{out}^-,$	$\frac{dX_{C_2}^+}{dt} = k_3X_{out}^+X_e - k_4X_{C_2}^+ - \eta X_{C_2}^+X_{C_2}^-,$ $\frac{dX_{C_2}^-}{dt} = k_3X_{out}^-X_e - k_4X_{C_2}^- - \eta X_{C_2}^+X_{C_2}^-,$
From (2.6), $x = x^+ - x^-$. Hence:	From (2.6), $x = x^+ - x^-$. Hence:
$\frac{dX_{out}}{dt} = \left[\frac{dX_{out}^+}{dt} - \frac{dX_{out}^-}{dt}\right],$	$\frac{dX_{C_2}}{dt} = \left[\frac{dX_{C_2}^+}{dt} - \frac{dX_{C_2}^-}{dt}\right],$
$= [k_2X_{C_1}^+ - k_3X_{out}^+X_e - \eta X_{out}^+X_{out}^-]$ $-[k_2X_{C_1}^- - k_3X_{out}^-X_e - \eta X_{out}^+X_{out}^-],$ $= k_2X_{C_1}^+ - k_3X_{out}^+X_e - \eta X_{out}^+X_{out}^-$ $-k_2X_{C_1}^- + k_3X_{out}^-X_e + \eta X_{out}^+X_{out}^-,$ $= k_2(X_{C_1}^+ - X_{C_1}^-) - k_3(X_{out}^+ - X_{out}^-)X_e,$ $= k_2X_{C_1} - k_3X_{out}X_e,$	$= [k_3X_{out}^+X_e - k_4X_{C_2}^+ - \eta X_{C_2}^+X_{C_2}^-]$ $-[k_3X_{out}^-X_e - k_4X_{C_2}^- - \eta X_{C_2}^+X_{C_2}^-],$ $= k_3X_{out}^+X_e - k_4X_{C_2}^+ - \eta X_{C_2}^+X_{C_2}^-$ $-k_3X_{out}^-X_e + k_4X_{C_2}^- + \eta X_{C_2}^+X_{C_2}^-,$ $= k_3X_e(X_{out}^+ - X_{out}^-) - k_4(X_{C_2}^+ - X_{C_2}^-),$ $= k_3X_eX_{out} - k_4X_{C_2},$
$\boxed{\frac{dX_{out}}{dt} = k_2X_{C_1} - k_3X_{out}X_e.}$	$\boxed{\frac{dX_{C_2}}{dt} = k_3X_{out}X_e - k_4X_{C_2}.}$

Thus, collectively we get:

$$\frac{dX_{C_1}}{dt} = k_1X_{in}X_p - k_2X_{C_1}, \quad (2.11a) \qquad \frac{dX_p}{dt} = -k_1X_{in}X_p + k_4X_{C_2}, \quad (2.11c)$$

$$\frac{dX_{out}}{dt} = k_2X_{C_1} - k_3X_{out}X_e, \quad (2.11b) \qquad \frac{dX_{C_2}}{dt} = k_3X_{out}X_e - k_4X_{C_2}. \quad (2.11d)$$

as shown in the derivations above, each signal, for example X_{C_1} can be understood as a difference in the concentrations of species $X_{C_1}^+$ and $X_{C_1}^-$. The remaining ODEs throughout the thesis can be derived in a similar manner. The total substrate concentration is expressed as $S_{total} := X_p + X_{out} + X_{C_1} + X_{C_2}$. From (2.11), it can be seen that $(dX_p/dt) + (dX_{out}/dt) + (dX_{C_1}/dt) + (dX_{C_2}/dt) = 0$, at equilibrium and thus S_{total} is assumed to be constant. Through an appropriate choice of reaction rates, one can obtain four distinct operating regimes for (2.11), as shown in Fig. 2.3(b).

2.2 Implementation using toehold mediated DNA strand displacement

DNA encodes genetic information essential for all biological functions such as growth, development, reproduction and so on. DNA is double helical in nature, comprising of the bases adenine (A), thymine (T), guanine (G), cytosine (C) linked by covalent bonds, as proposed by James Watson and Francis Crick in 1953 [60]. The bases are attached to a phosphate group and a deoxyribose sugar, together referred to as *nucleotides* (nt). Two nucleotide strands run opposite to each other and are antiparallel. *Single stranded DNA* (ssDNA) is used as a template, to transcribe *messenger ribonucleic acid* (mRNA) (transcription), which is then translated to specify a sequence of amino acids that forms a protein molecule in a process called translation [4; 61; 62]. Mutation in base pairs can lead to genetic disorders, for example, the mutation in the *haemoglobin* (HBB) gene causes sickle cell anaemia [63; 64].

Over the past few years, synthetic nucleic acids have been used as programmable building blocks for molecular level structures and circuits [19; 65; 66]. In particular, DNA is chosen as the building block in this thesis because its thermodynamics [67–69] and mechanical properties [70; 71] are well understood. Also, the ability to synthesise DNA, and thus RNA and protein is getting relatively faster and cheaper [72; 73]. Another reason is that, to implement a prescribed dynamic behaviour *in vitro*, the chosen molecular structure should offer a way to make the tuning of reaction rate constants simple. It is quite feasible to do this with DSD [74; 75] by changing the length of a particular subsequence of the reacting DNA, referred to as a *toehold*. In Section 2.2.1, the underlying molecular mechanism of toehold mediated DSD is explained, and this mechanism will form the basis of the proposed experimental implementation of the various dynamic devices and circuits described later in this thesis. This mechanism provides a precise control over the reaction kinetics by allowing engineers to program when and where specific actions or steps occur in a molecular device.

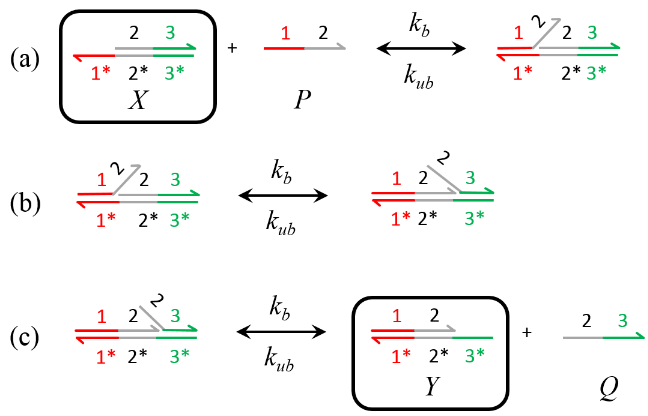


Figure 2.4: DNA strand displacement reaction mechanism: the DNA strands are bonded by Watson-Crick base pairing, denoted by * and arrows indicate a $3^{'}$ end. The basic steps involved are: (a) binding of toehold **1** invader strand P to **1*** of complementary target strand X, (b) branch migration wherein the strand **1-2** partially displaces strand **2-3**, and (c) complete separation of strand **2-3**.

2.2.1 The architecture and mechanism

This section presents an overview of the DSD mechanism, through which the types of chemical reactions used in this thesis may be implemented. Consider the reversible bimolecular reaction:

$$X + P \underset{k_{ub}}{\overset{k_b}{\rightleftharpoons}} Y + Q, \tag{2.12}$$

where, X, P, Y and Q are DNA strands while k_b and k_{ub} are the binding and unbinding rates, respectively. A DSD implementation of this reaction is shown in Fig. 2.4. It begins with an invader strand P binding to the complementary target strand X at the toehold **1*** through Watson-Crick base pairing, denoted by * and arrows indicate a $3^{'}$ end. [60]. Through an intermediate process of *branch migration*, P displaces the evader strand **2-3** from X, thereby producing the partially double stranded product Y that can further react with other DNA complexes using the toehold **3***.

The numbers $1, 2, 3$ and $1^*, 2^*, 3^*$ are referred as domains and denote the set of adjoining nucleotides. If the DNA strands belong to entirely different domains i.e. contain no complementary DNA strands with respect to each other, as is often the case, they do not interact with each other directly and therefore DSD reactions must be mediated by so-called auxiliary DNA species, which must be present in sufficiently large amounts [76]. We assume that the complementary strands react only with each other, although this constraint can be relaxed, as demonstrated in [17]. For the DSD reactions to be fast and thereby reduce mismatches during branch migrations, the toehold domains should be short: for example, of the order of 6–10 nt, where *nt* denotes nucleotides, and the displacement domains should preferably be 20 nt [77]. The reaction rate constants, and consequently the kinetics of the system, are a function of the toehold binding strength and can thus be altered by varying the binding strength and the strand composition [76]. If all the steps and reactions from Fig. 2.4 are considered, the resulting number of ODEs increases. This can be a drawback in designing complex and bigger networks from mathematical modelling point of view. Hence, elementary DNA reactions are approximated into CRNs by excluding auxiliary species as described in [15] (see figures in Section 2.2.2). Corresponding reaction rates are also approximated in terms of initial concentration of auxiliary DNA species (C_{max}), and forward binding reaction rates (q_i and q_{max}).

2.2.2 Software design tool: Visual DSD

The recent accelerated growth in the complexity of the experimental systems implementing DNA devices has led to the frequent use of computing software for the design of such systems. Visual DSD [1] [78; 79] is one such software package, developed by Microsoft Research, that is used for the analysis, simulation and verification of DSD circuits. In Visual DSD, each DNA strand is represented with domains where, the numbers $1,2,3$ and $1^*,2^*,3^*$ denote the set of adjoining nucleotides (see Fig. 2.4). To simulate the DNA reaction, reactant DNA strands are defined in a simple code along with their initial concentrations, toehold presence on the strand (if any) and reaction rate. The evolution of each species concentration over the time can be viewed and also modified by changing these parametes. The simulated reactions may produce different domain composition of the product DNA strands, depending on which domain the toehold presence was defined. If any of the product DNA strands does not have a toehold presence then it might not be able to react further and is thus referred to as inert or waste product (ϕ).

[1]Visual DSD, version v0.14-20140319-34170, Microsoft Research, Microsoft Corporation, 2014.

To implement the linear and nonlinear operators that form the various functional circuits considered in this thesis, four types of chemical reactions are required - catalysis, bimolecular, degradation and annihilation. For each of these chemical reactions, the underlying DSD mechanism is illustrated using Visual DSD in Figs. 2.5 to 2.9. Note that the catalysis reactions (2.15) and (2.18) in Figs. 2.5 and 2.6, respectively, produce different output species depending on the domain composition of the reactant auxiliary species. The auxiliary species considered here are namely, $G_i^{\pm}$, $T_i^{\pm}$, $H_i^{\pm}$, $O_i^{\pm}$, $L_i^{\pm}$, $B_i^{\pm}$, $HS_i^{\pm}$, $LS_i^{\pm}$, $BS_i^{\pm}$. Species $G_i^{\pm}$ and $T_i^{\pm}$, which are partially double stranded DNAs, and single strands of $O_i^{\pm}$, can be observed to have different domain compositions in Fig. 2.5 and Fig. 2.6. As a result, in (2.15) two different products, $X_2^{\pm}$ and $X_3^{\pm}$, are obtained whereas in (2.18) the single species $Y^{\pm}$ is produced.

The domain 1_q^* in Figs. 2.5 to 2.9 denotes the subsequence of domain 1 that may be the same length as 1 but contains some mismatched bases over the displacement domain. The reaction rate of 1_q^* is however tuned to rate q_i [77] and other corresponding reaction rates are set by following the notation from [31] and [15]. Initial concentrations of the auxiliary species $G_{i0}^{\pm}$, $T_{i0}^{\pm}$, $L_{i0}^{\pm}$, $B_{i0}^{\pm}$, $LS_{i0}^{\pm}$, $BS_{i0}^{\pm}$ are set to $C_{max} = 1000$ nM. In Fig. 2.7, which gives the DNA implementation of the bimolecular CRN, the concentrations of $T_i^{\pm}$, $L_i^{\pm}$, $B_i^{\pm}$ remain constant throughout the process [15]. The notion of '*initial concentration*' that is defined in Visual DSD or in *in vitro* / *in vivo* experiments represents a certain amount of molar concentration. Whereas, for the simulations of the associated ODEs in MATLAB [2] [80], the initial condition indicates the initial molar concentration as a reference and is thus set to zero. The MATLAB simulations thus show the dynamic behaviour of the DNA strands that evolve from this reference concentration.

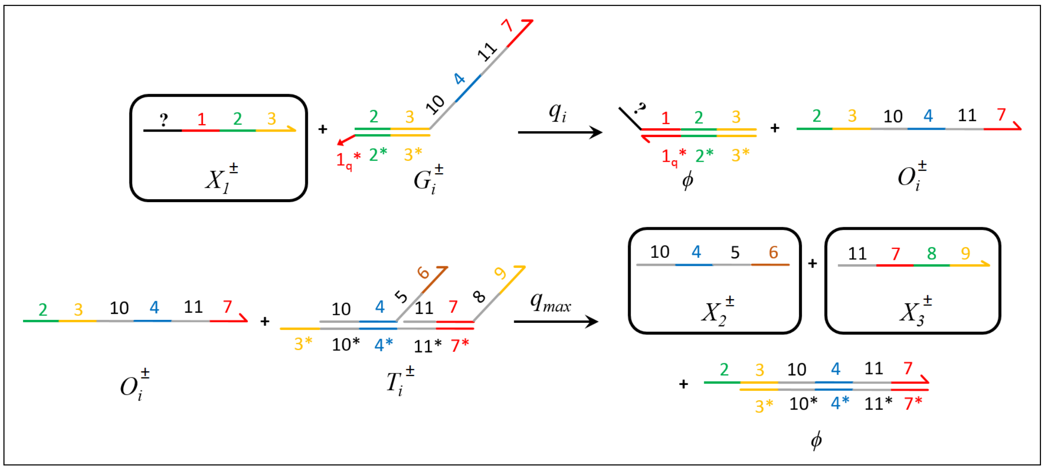
The DSD implementation of the catalysis reactions $X_1^{\pm} \xrightarrow{k_1} X_2^{\pm} + X_3^{\pm}$ and $X^{\pm} \xrightarrow{k_2} X^{\pm} + Y^{\pm}$ is illustrated in Figs. 2.5 and 2.6 . Accordingly, the reactions (2.13), (2.16) initiate with the ssDNA $X_1^{\pm}$ (or $X^{\pm}$) displacing auxiliary species $G_i^{\pm}$ irreversibly at the rate q_i, producing the intermediate complex $O_i^{\pm}$ and waste. Complex $O_i^{\pm}$ on reacting with auxiliary species $T_i^{\pm}$, releases two single stranded products, $X_2^{\pm}$ (or $X^{\pm}$) and $X_3^{\pm}$ (or $Y^{\pm}$). A DNA strand without the presence of a toehold is

[2] MATLAB version 8.3.0.532. Natick, Massachusetts: The MathWorks Inc., 2014.

potentially unable to react further and is thus denoted as an inert or waste product species, indicated as ϕ.

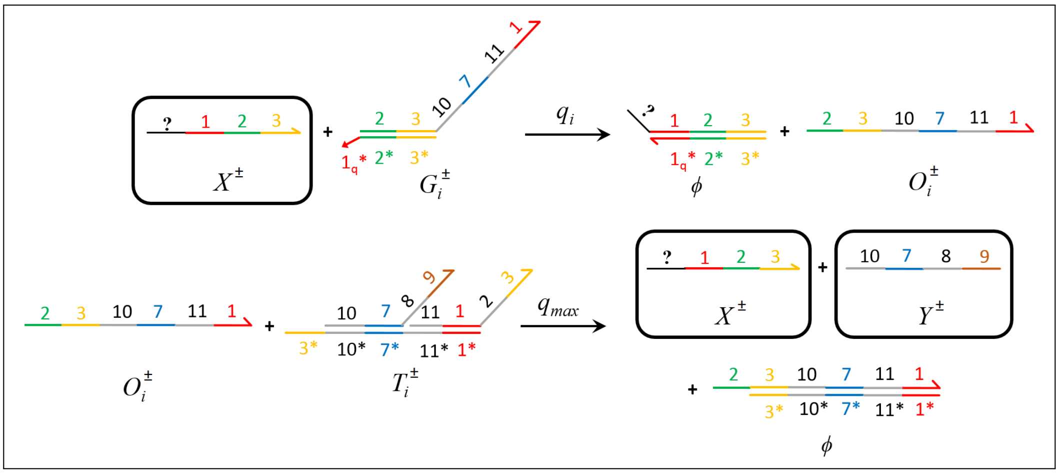
The DSD implementation of the bimolecular reaction (2.22) is shown in Fig 2.7. Here, the reaction begins with single strand $X_1^{\pm}$ reacting reversibly with the auxiliary species $L_i^{\pm}$ to produce activated intermediate complexes $H_i^{\pm}$ and $B_i^{\pm}$. Due to the presence of $X_2^{\pm}$ in the solution with an active toehold, it reacts with complex $H_i^{\pm}$ to release intermediate complex $O_i^{\pm}$ and an inert species ϕ. If $X_2^{\pm}$ is absent then $B_i^{\pm}$ can reversibly displace $H_i^{\pm}$, releasing $X_1^{\pm}$ back into the solution. Complex $O_i^{\pm}$ displaces $T_i^{\pm}$. Hence, the approximated bimolecular reaction given by (2.22) is irreversible and produces ssDNA $X_3^{\pm}$.

The degradation and annihilation reactions are illustrated in Figs. 2.8 and 2.9, respectively. In Fig. 2.8, ssDNA $X^{\pm}$ reacts with partially double stranded species $G_i^{\pm}$ and produces inert species. Due to the absence of a toehold presence, the product species are unable to react further and therefore the reaction is termed as 'degradation'. The annihilation reaction shown in Fig. 2.9 is considered in the reaction network because of the adopted methodology from [31], as explained in Section 2.1.2. For the reaction (2.28), the DSD begins with ssDNA X^+ reacting with L_i reversibly to produce H_i and B_i. A ssDNA X^- reacts with the partially double stranded species LS_i, producing intermediate strands HS_i and BS_i. Further, X^- produces an inert species on reacting with one of the intermediate species, H_i.



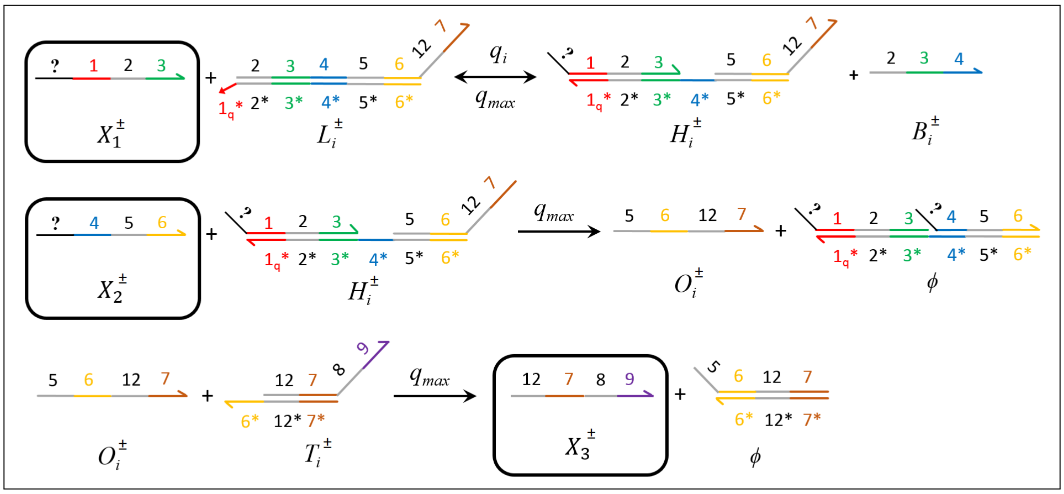
DNA Implementation		CRN	
$X_1^{\pm} + G_i^{\pm} \xrightarrow{q_i} \phi + O_i^{\pm}$ (2.13)			where,
$O_i^{\pm} + T_i^{\pm} \xrightarrow{q_{max}} X_2^{\pm} + X_3^{\pm}$ (2.14)	$\Big\}$	$X_1^{\pm} \xrightarrow{k_1} X_2^{\pm} + X_3^{\pm}$ (2.15)	$\boxed{q_i = \dfrac{k_1}{C_{max}}}$

Figure 2.5: Catalysis reaction $X_1^{\pm} \rightarrow X_2^{\pm} + X_3^{\pm}$. The DNA implementation of catalysis reaction (2.15) with reaction index i and black boxes highlighting the species, $X_1^{\pm}$, $X_2^{\pm}$ and $X_3^{\pm}$. Domain 1_q^* may not entirely complement domain **1** but its toehold domain reaction rate is tuned to q_i. In (2.13), species G_i reacts with $X_1^{\pm}$ and releases $O_i^{\pm}$ along with waste ϕ. $O_i^{\pm}$ on reacting with species $T_i^{\pm}$ produces two single stranded DNAs, $X_2^{\pm}$ and $X_3^{\pm}$. [15]. The question mark appearing on the DNA strands such as $X_1^{\pm}$ and ϕ, indicates the species identifier; as adapted from [31].



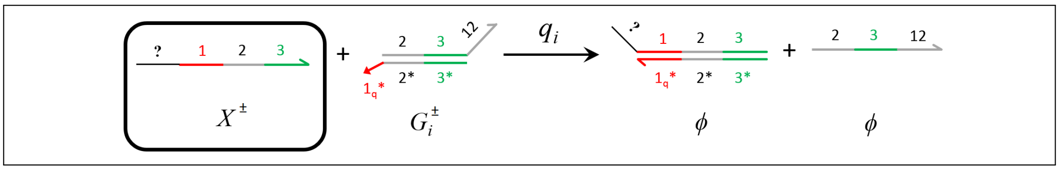
DNA Implementation	**CRN**	
$X^{\pm} + G_i^{\pm} \xrightarrow{q_i} \phi + O_i^{\pm}$ (2.16) $O_i^{\pm} + T_i^{\pm} \xrightarrow{q_{max}} X^{\pm} + Y^{\pm}$ (2.17)	$X^{\pm} \xrightarrow{k_2} X^{\pm} + Y^{\pm}$ (2.18)	where, $\boxed{q_i = \frac{k_2}{C_{max}}}$

Figure 2.6: Catalysis reaction $X^{\pm} \rightarrow X^{\pm} + Y^{\pm}$. The unimolecular catalysis reaction (2.18) is approximated from the DNA implementation with reaction index i. In (2.16), species $G_i^{\pm}$ reacts with $X^{\pm}$ to produce $O_i^{\pm}$ and in (2.17), $O_i^{\pm}$ releases $X^{\pm}$ and $Y^{\pm}$, on reacting with species $T_i^{\pm}$; as adapted from [31]. The strand displacement mechanism resembles to that in Fig. 2.5 but, the nucleotide composition of the product species vary depending on the composition of the auxiliary species involved [15].



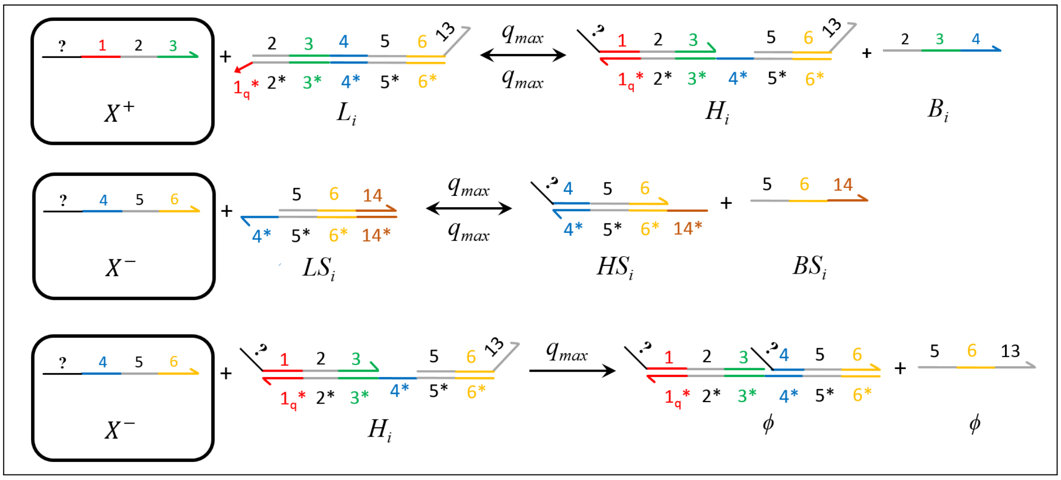
DNA Implementation	CRN
$X_1^{\pm} + L_i^{\pm} \underset{q_{max}}{\overset{q_i}{\rightleftharpoons}} H_i^{\pm} + B_i^{\pm}$ (2.19) $X_2^{\pm} + H_i^{\pm} \xrightarrow{q_{max}} O_i^{\pm} + \phi$ (2.20) $O_i^{\pm} + T_i^{\pm} \xrightarrow{q_{max}} X_3^{\pm}$ (2.21)	$X_1^{\pm} + X_2^{\pm} \xrightarrow{k_3} X_3^{\pm}$ (2.22) where, $\boxed{k_3 = q_i}$

Figure 2.7: Bimolecular reaction $X_1^{\pm} + X_2^{\pm} \rightarrow X_3^{\pm}$: DNA implementation of a bimolecular reaction (2.22) with reaction index i and black boxes highlighting the species, $X_1^{\pm}$, $X_2^{\pm}$, $X_3^{\pm}$. In (2.19) $X_1^{\pm}$ displaces auxiliary species $L_i^{\pm}$ reversibly producing the intermediate complex $H_i^{\pm}$ which reacts with $X_2^{\pm}$ as given in (2.20) producing $O_i^{\pm}$. In (2.21), $X_3^{\pm}$ is produced when $O_i^{\pm}$ irreversibly displaces $T_i^{\pm}$; as adapted from [15].



DNA Implementation	**CRN**
$X^{\pm} + G_i^{\pm} \xrightarrow{q_i} \phi \qquad (2.23)$	$X^{\pm} \xrightarrow{\gamma_i} \phi \qquad (2.24) \qquad$ where, $\boxed{q_i = \dfrac{\gamma_i}{C_{max}}}$

Figure 2.8: Degradation reaction $X^{\pm} \rightarrow \phi$: DNA implementation of species $X^{\pm}$ degradation on reacting with auxiliary species $G_i^{\pm}$. In (2.23), $X^{\pm}$ performs strand displacement on $G_i^{\pm}$ producing inert waste ϕ. (2.24) represents the chemical reaction derived from this DNA strand displacement reaction; as adapted from [31].



DNA Implementation	CRN
$X^+ + L_i \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_i + B_i$ (2.25) $X^- + LS_i \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_i + BS_i$ (2.26) $X^- + H_i \xrightarrow{q_{max}} \phi$ (2.27)	$X^+ + X^- \xrightarrow{\eta_i} \phi$ (2.28) where, $\boxed{\eta_i = \frac{q_{max}}{2}}$

Figure 2.9: Annihilation reaction $X^+ + X^- \rightarrow \phi$: The DSD diagram shows degradation of auxiliary species X^+ and X^- by means of molecules L_i and LS_i. The reaction dynamics are separated into fast and slow time scales such that, X^+ and X^- are sequestered into intermediate species through reaction with L_i and LS_i at a fast reaction rate, while X^- degrades into waste by reacting with H_i at a slower rate. The initial concentrations of X^+ and X^- must be scaled by a factor of 2 (let, ξ = 2, hence, X_0^+ = 1ξ nM and X_0^- = 0.5ξ nM) to attenuate for the sequestering effect of the fast dynamics; as adapted from [31].

Chapter 3

Biomolecular Implementation of Nonlinear System Theoretic Operators

This chapter focusses on the design of linear and nonlinear system theoretic operators using *chemical reaction networks* (CRNs). These CRNs can be approximated in DNA based chemistry and vice-versa thus making the operators implementable using *DNA strand displacement* (DSD) reactions. The chapter begins with a discussion on how to build individual linear and nonlinear operators based on the methodology from [31] and discussed in Chapter 2, Section 2.1.2. Subsequently it is shown how a number of functional circuits can be built using a combination of these individual operators.

3.1 Background results on linear operators

To realise *linear time-invariant* (LTI) systems using CRNs, [31; 81] show different realisations to represent elementary system theoretic operations such as gain, summation and integration. Generally, these mathematical operations are based on the combination of three forms of elementary chemical reactions: catalysis, degradation and annihilation. Note that, while the framework considered in [81] is simpler and also considers nonlinear operators, their proposed framework is not suited for the systems under consideration in this thesis, since it does not allow the computation of negative signals. Thus, the framework proposed by [31] is followed here instead. In this section, the complete background theory based on [31] required to realise linear/nonlinear operators is outlined.

Following [31], a set of reactions is compactly represented as $x_i^{\pm} \xrightarrow{k} x_o^{\pm}$, which represents two reactions: $x_i^{+} \xrightarrow{k} x_o^{+}$ and $x_i^{-} \xrightarrow{k} x_o^{-}$. Strictly speaking, any reaction with superscript $\pm$ and $\mp$ should be decomposed into their individual '$+$' and '$-$' components but for brevity and to avoid overloading of reactions, they are written in the compact form.

Note that, lower-case notations for chemical species are used only for the lemmas in this thesis. Also, the DNA implementation reactions of the functional operators may be reversible but, their corresponding approximated CRNs appear to be irriversible, in order to simplify the mathematical modelling of the CRNs. Now the lemmas to support the implementation of the linear theoretic operators using CRNs are presented:

3.1.1 Gain, summation and integration

Lemma 1 [Scalar gain k]

Let, $x_o = kx_i$ where, x_i and x_o are the input and output as shown in Fig. 3.1, re-

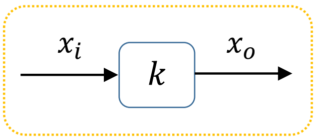


Figure 3.1: A block diagram representing scalar gain

spectively and k represent scalar gain. This operation can be implemented using the following set of chemical reactions:

$$x_i^{\pm} \xrightarrow{\gamma k} x_i^{\pm} + x_o^{\pm}; \qquad x_o^{\pm} \xrightarrow{\gamma} \emptyset; \qquad x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset. \tag{3.1}$$

where, γ and η are the kinetic rates associated with degradation and annihilation reactions respectively.

Proof: *Using generalised mass-action kinetics, the following ODE is obtained:*

$$\frac{dx_o}{dt} = \gamma(kx_i - x_o). \tag{3.2}$$

At steady state,

$$x_o = kx_i. \tag{3.3}$$

Thus, (3.2) shows how the gain operator can be implemented.

Lemma 2 [Summation/subtraction]

Consider the summation operation $x_o = x_i + x_d$, where x_i and x_d are the inputs

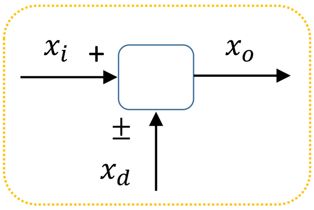


Figure 3.2: A block diagram representing summation or subtraction

and x_o is the output, as shown in Fig. 3.2. This operation is implemented using the following set of chemical reactions:

$$x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}; \qquad x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\pm}; \qquad x_o^{\pm} \xrightarrow{\gamma} \emptyset; \qquad x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset. \tag{3.4}$$

The subtraction operation $x_o = x_i - x_d$ is implemented using the following set of chemical reactions:

$$x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}; \qquad x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\mp}; \qquad x_o^{\pm} \xrightarrow{\gamma} \emptyset; \qquad x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset. \tag{3.5}$$

Note that, the only difference between sets (3.4) and (3.5) is reaction $x_d \xrightarrow{\gamma} x_d + x_o$. To perform addition, x_d^{+} produces x_o^{+} in (3.4) whereas, for subtraction operation x_d^{+} produces x_d^{-} in (3.5).

Proof: *Applying generalised mass-action kinetics to (3.4), the ODE obtained for summation operation is:*

$$\frac{dx_o}{dt} = \gamma(x_i + x_d - x_o). \tag{3.6}$$

At steady state,

$$x_o = x_i + x_d. \tag{3.7}$$

Similarly, applying mass-action kinetics to (3.5), ODE obtained for the subtraction operation is:

$$\frac{dx_o}{dt} = \gamma(x_i - x_d - x_o). \tag{3.8}$$

At steady state,

$$x_o = x_i - x_d. \tag{3.9}$$

Thus, (3.6) and (3.8) shows the implementation of summation and subtraction operators, respectively.

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

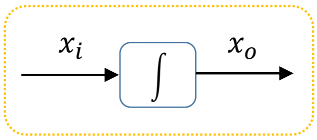


Figure 3.3: A block diagram representing integration

is the DC gain (see Fig. 3.3). This operation is implemented using the following set of chemical reactions:

$$x_i^{\pm} \xrightarrow{k} x_i^{\pm} + x_o^{\pm}; \qquad x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset. \tag{3.10}$$

Proof: *Using generalised mass-action kinetics, the following ODE is obtained:*

$$\frac{dx_o}{dt} = kx_i. \tag{3.11}$$

Thus, (3.11) shows implementation of the integration operator.

3.2 Nonlinear operators

This section shows how a number of nonlinear operators, namely multiplication, division and polynomials, may also be designed using chemical reactions, and subsequently implemented via DNA-based chemistry.

3.2.1 Multiplication, division and polynomial operators

Lemma 4 [Multiplication operator]

Consider the multiplication operation, $x_o = x_1 x_2$, where, x_1 and x_2 are the inputs

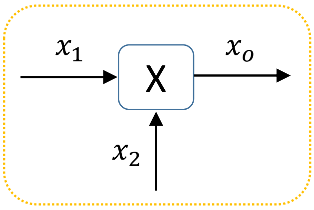


Figure 3.4: A block diagram representing multiplication

and x_o is the output product, as shown in Fig. 3.4. This operation can be implemented using the chemical reactions:

$$\begin{aligned} x_1^{\pm} + x_2^{\pm} \xrightarrow{\gamma_M} x_1^{\pm} + x_2^{\pm} + x_o^{+}; \qquad x_1^{\pm} + x_2^{\mp} \xrightarrow{\gamma_M} x_1^{\pm} + x_2^{\mp} + x_o^{-}; \\ x_o^{\pm} \xrightarrow{\gamma} \emptyset; \qquad x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset. \end{aligned} \tag{3.12}$$

Here, γ_M, γ and η are the binding, degradation and annihilation reaction rates, respectively.

Proof: *Using generalised mass-action kinetics, the following ODE is obtained:*

$$\frac{dx_o}{dt} = \gamma_M x_1 x_2 - \gamma x_o. \tag{3.13}$$

At steady state,

$$x_o = \frac{\gamma_M}{\gamma} x_1 x_2. \tag{3.14}$$

Hence, multiplication operation of inputs x_1 and x_2 can be implemented with (3.13).

Lemma 5 [Division operator]

Consider the system $\mathscr{S}_D$ shown in Fig. 3.5. Let the biomolecular signals u and z be its inputs. Then its output y computes the ratio u/z.

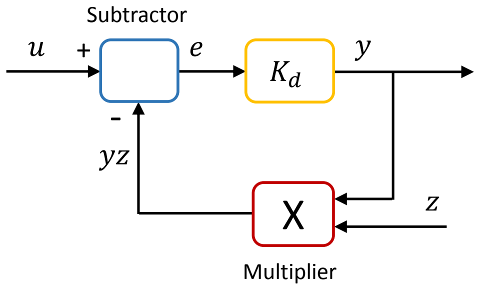


Figure 3.5: A block diagram representation of the feedback system $\mathscr{S}_D$ that computes the ratio $y = u/z$ where, u and z are biomolecular signals.

Proof: From Fig. 3.5, the error signal is $e = u - yz$ and $y = K_d e$. Substituting the former equation into the latter one and rearranging the variables, we get:

$$y = K_d(u - yz) = \frac{K_d u}{1 + K_d z} = \frac{u}{(1/K_d) + z}. \tag{3.15}$$

If K_d is chosen large enough, $y \approx u/z$.

Remark The configuration in Fig. 3.5 consist of a gain, a subtractor and a multiplication operator. The corresponding CRNs for these operators are given in (3.1), (3.5) and (3.12), respectively.

Lemma 6 [Polynomial x^n]

Let $x_{p,n}$ denote the polynomial of degree n defined as $x_{p,n} = x^n$ (see 'power component' block of Fig. 3.6). Then, output of power component $x_{p,n}$ is realised through the following set of chemical reactions:

$$x^{\pm} + x^{\pm} \xrightarrow{\gamma_p} x^{\pm} + x^{\pm} + (x_{p,2})^{+}, \quad (3.16a) \qquad (x_{p,2})^{\pm} \xrightarrow{\gamma_p} \emptyset, \quad (3.16c)$$

$$x^{\pm} + x^{\mp} \xrightarrow{\gamma_p} x^{\pm} + x^{\mp} + (x_{p,2})^{-}, \quad (3.16b) \qquad (x_{p,2})^{+} + (x_{p,2})^{-} \xrightarrow{\eta} \emptyset. \quad (3.16d)$$

$$\vdots$$

$$x^{\pm} + (x^{n-1})^{\pm} \xrightarrow{\gamma_p} x^{\pm} + (x^{n-1})^{\pm} + (x_{p,n})^{+}, \tag{3.17a}$$

$$x^{\pm} + (x^{n-1})^{\mp} \xrightarrow{\gamma_p} x^{\pm} + (x^{n-1})^{\mp} + (x_{p,n})^{-}, \tag{3.17b}$$

$$(x_{p,n})^{\pm} \xrightarrow{\gamma_p} \emptyset, \quad (3.17c) \qquad (x_{p,n})^{+} + (x_{p,n})^{-} \xrightarrow{\eta} \emptyset. \quad (3.17d)$$

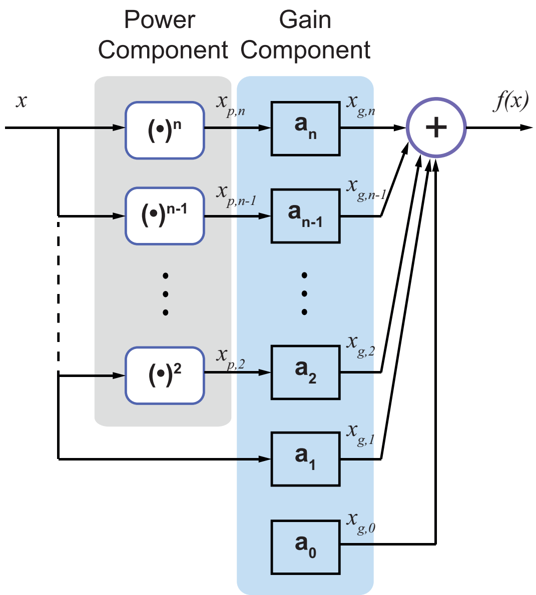


Figure 3.6: The input-output system derived in Lemma 7 to compute the univariate polynomial $f(x) = \sum_{i=0}^{n} a_i x^i$. The result uses intermediate variables $x_{p,i}$ which can be computed using the chemical reactions given by Lemma 6. The output of the power component is indicated as $x_{p,i}$ and the output of the gain component is indicated as $x_{g,i}$. This implementation requires $11n + 7$ chemical reactions, where n is the degree of the polynomial $f(x)$.

where, γ_p and η are the catalysis and degradation reaction rates, respectively. Here, the output species in (3.16) denotes $x_{p,2} = (x^2)$ whereas, output species in (3.17) denotes $x_{p,n} = (x^n)$. This way, n^{th} order component with $n-1$ CRNs can be computed as shown with (3.16), (3.17).

Proof: Using generalised mass-action kinetics, it can be verified that the CRN (3.17) is described using the following ODE:

$$\frac{dx_{p,n}}{dt} = \gamma_p (x^n - x_{p,n}). \tag{3.18}$$

Hence, using the final value theorem, it follows that the CRN (3.17) implements the desired function at steady-state with $1/\gamma_p$ as the time constant.

Lemma 7 [Univariate polynomial]

Consider the block diagram shown in Fig. 3.6. Let $f(x)$ be the univariate polyno-

mial of degree n defined as:

$$f(x) = \sum_{i=0}^{n} a_i x^i. \tag{3.19}$$

Then, $f(x)$ is realised through the feedforward system illustrated in Fig. 3.6, where the output of the power component is indicated as $x_{p,i}$ while the output of the gain component is indicated as $x_{g,i}$.

Proof: The proof follows trivially using the proofs of Lemmas 1-6.

Remark 1 It may be noted that the constant a_0 can be realised as, $\emptyset \xrightarrow{a_0} x^{\pm}_{g,0}$ so that, $x_{g,0} \longrightarrow a_0$ at steady-state with the time constant equal to $1/a_0$.

Remark 2 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 chemical reactions for both $\hat{u}$ and $\hat{z}$ can be realised using Lemma 7. 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 5.

Table 3.1 lists the DNA strand displacement reactions, CRNs and the corresponding ODEs for the implementation of each of the nonlinear system theoretic operators from Sections 3.1.1 and 3.2.1.

DNA Implementation	CRNs	ODEs
(a) Gain		
$x_i^{\pm} + G_1^{\pm} \xrightarrow{q_1} \emptyset + O_1^{\pm}$	$x_i^{\pm} \xrightarrow{\gamma k} x_i^{\pm} + x_o^{\pm}$	
$O_1^{\pm} + T_1^{\pm} \xrightarrow{q_{max}} x_i^{\pm} + x_o^{\pm}$		
$x_o^{\pm} + G_2^{\pm} \xrightarrow{q_2} \emptyset$	$x_o^{\pm} \xrightarrow{\gamma} \emptyset$	
$x_o^{+} + L_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_3 + B_3$		$\frac{dx_0}{dt} = \gamma(kx_i - x_o)$
$x_o^{-} + LS_3 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_3 + BS_3$	$x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset$	
$x_o^{-} + H_3 \xrightarrow{q_{max}} \emptyset$		
(b) Summation		
$x_i^{\pm} + G_4^{\pm} \xrightarrow{q_4} \emptyset + O_4^{\pm}$	$x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}$	
$O_4^{\pm} + T_4^{\pm} \xrightarrow{q_{max}} x_i^{\pm} + x_o^{\pm}$		
$x_d^{\pm} + G_5^{\pm} \xrightarrow{q_5} \emptyset + O_5^{\pm}$	$x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\pm}$	
$O_5^{\pm} + T_5^{\pm} \xrightarrow{q_{max}} x_d^{\pm} + x_o^{\pm}$		
$x_o^{\pm} + G_6^{\pm} \xrightarrow{q_6} \emptyset$	$x_o^{\pm} \xrightarrow{\gamma} \emptyset$	$\frac{dx_o}{dt} = \gamma(x_i + x_d - x_o)$
$x_o^{+} + L_7 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_7 + B_7$		
$x_o^{-} + LS_7 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_7 + BS_7$	$x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset$	
$x_o^{-} + H_7 \xrightarrow{q_{max}} \emptyset$		
(c) Subtraction		
$x_i^{\pm} + G_8^{\pm} \xrightarrow{q_8} \emptyset + O_8^{\pm}$	$x_i^{\pm} \xrightarrow{\gamma} x_i^{\pm} + x_o^{\pm}$	
$O_8^{\pm} + T_8^{\pm} \xrightarrow{q_{max}} x_i^{\pm} + x_o^{\pm}$		
$x_d^{\pm} + G_9^{\pm} \xrightarrow{q_9} \emptyset + O_9^{\pm}$	$x_d^{\pm} \xrightarrow{\gamma} x_d^{\pm} + x_o^{\mp}$	
$O_9^{\pm} + T_9^{\pm} \xrightarrow{q_{max}} x_d^{\pm} + x_o^{\mp}$		
$x_o^{\pm} + G_{10}^{\pm} \xrightarrow{q_{10}} \emptyset$	$x_o^{\pm} \xrightarrow{\gamma} \emptyset$	$\frac{dx_o}{dt} = \gamma(x_i - x_d - x_o)$
$x_o^{+} + L_{11} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{11} + B_{11}$		
$x_o^{-} + LS_{11} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{11} + BS_{11}$	$x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset$	
$x_o^{-} + H_{11} \xrightarrow{q_{max}} \emptyset$		
(d) Integration		
$x_i^{\pm} + G_{12}^{\pm} \xrightarrow{q_{12}} \emptyset + O_{12}^{\pm}$	$x_i^{\pm} \xrightarrow{k} x_i^{\pm} + x_o^{\pm}$	
$O_{12}^{\pm} + T_{12}^{\pm} \xrightarrow{q_{max}} x_i^{\pm} + x_o^{\pm}$		
$x_o^{+} + L_{13} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{13} + B_{13}$		$\frac{dx_o}{dt} = kx_i$
$x_o^{-} + LS_{13} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{13} + BS_{13}$	$x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset$	
$x_o^{-} + H_{13} \xrightarrow{q_{max}} \emptyset$		

Table 3.1: DNA Implementation reactions, CRNs and the corresponding ODEs for the implementation of components, where, x_i and x_o denote the input and output of each individual component, respectively: (b) Summation, modelled using 13 DNA reactions - approximated to 7 chemical reactions. (c) Subtraction, modelled using 13 DNA reactions - approximated to 7 chemical reactions. (d) Integration, modelled using 7 DNA reactions - approximated to 3 chemical reactions. The DNA implementation reaction rates are set to $q_i = 800$ /M/s ($i = 1, 2, ..21$), $q_{max} = 10^7$ /M/s and initial concentration of auxiliary species, $C_{max} = 1000$ nM. The reaction rate of annihilation, η, is set to $10 \cdot q_i C_{max}$. $\emptyset$ indicates inert or waste product [15; 31].

DNA Implementation	CRNs	ODEs
(e) Multiplication		
$x_1^{\pm} + L_{14}^{\pm} \xrightleftharpoons[q_{max}]{q_{14}} H_{14}^{\pm} + B_{14}^{\pm}$ $x_2^{\pm} + H_{14}^{\pm} \xrightarrow{q_{max}} O_{14}^{\pm}$ $O_{14}^{\pm} + T_{14}^{\pm} \xrightarrow{q_{max}} x_1^{\pm} + x_2^{\pm} + x_o^{+}$	$x_1^{\pm} + x_2^{\pm} \xrightarrow{\gamma_M} x_1^{\pm} + x_2^{\pm} + x_o^{+}$	$\frac{dx_o}{dt} = \gamma_M x_1 x_2 - \gamma x_o$
$x_1^{\pm} + L_{15}^{\pm} \xrightleftharpoons[q_{max}]{q_{15}} H_{15}^{\pm} + B_{15}^{\pm}$ $x_2^{\mp} + H_{15}^{\pm} \xrightarrow{q_{max}} O_{15}^{\pm}$ $O_{15}^{\pm} + T_{15}^{\pm} \xrightarrow{q_{max}} x_1^{\pm} + x_2^{\mp} + x_o^{-}$	$x_1^{\pm} + x_2^{\mp} \xrightarrow{\gamma_M} x_1^{\pm} + x_2^{\mp} + x_o^{-}$	
$x_o^{\pm} + G_{16}^{\pm} \xrightarrow{q_{16}} \emptyset$	$x_o^{\pm} \xrightarrow{\gamma} \emptyset$	
$x_o^{+} + L_{17} \xrightleftharpoons[q_{max}]{q_{max}} H_{17} + B_{17}$ $x_o^{-} + LS_{17} \xrightleftharpoons[q_{max}]{q_{max}} HS_{17} + BS_{17}$ $x_o^{-} + H_{17} \xrightarrow{q_{max}} \emptyset$	$x_o^{+} + x_o^{-} \xrightarrow{\eta} \emptyset$	
(f) Power Component		
$x^{\pm} + L_{18}^{\pm} \xrightleftharpoons[q_{max}]{q_{18}} H_{18}^{\pm} + B_{18}^{\pm}$ $x^{\pm} + H_{18}^{\pm} \xrightarrow{q_{max}} O_{18}^{\pm}$ $O_{18}^{\pm} + T_{18}^{\pm} \xrightarrow{q_{max}} x^{\pm} + x^{\pm} + (x_{p,2})^{+}$	$x^{\pm} + x^{\pm} \xrightarrow{\gamma_p} x^{\pm} + x^{\pm} + (x_{p,2})^{+}$	$\frac{dx_{p,n}}{dt} = \gamma_p(x^n - x_{p,n})$
$x^{\pm} + L_{19}^{\pm} \xrightleftharpoons[q_{max}]{q_{19}} H_{19}^{\pm} + B_{19}^{\pm}$ $x^{\mp} + H_{19}^{\pm} \xrightarrow{q_{max}} O_{19}^{\pm}$ $O_{19}^{\pm} + T_{19}^{\pm} \xrightarrow{q_{max}} x^{\pm} + x^{\mp} + (x_{p,2})^{-}$	$x^{\pm} + x^{\mp} \xrightarrow{\gamma_p} x^{\pm} + x^{\mp} + (x_{p,2})^{-}$	
$(x_{p,2})^{\pm} + G_{20}^{\pm} \xrightarrow{q_{20}} \emptyset$	$(x_{p,2})^{\pm} \xrightarrow{\gamma_p} \emptyset$	
$(x_{p,2})^{+} + L_{21} \xrightleftharpoons[q_{max}]{q_{max}} H_{21} + B_{21}$ $(x_{p,2})^{-} + LS_{21} \xrightleftharpoons[q_{max}]{q_{max}} HS_{21} + BS_{21}$ $(x_{p,2})^{-} + H_{21} \xrightarrow{q_{max}} \emptyset$	$(x_{p,2})^{+} + (x_{p,2})^{-} \xrightarrow{\eta} \emptyset$	
$\vdots$	$\vdots$	
$x^{\pm} + L_{22}^{\pm} \xrightleftharpoons[q_{max}]{q_{22}} H_{22}^{\pm} + B_{22}^{\pm}$ $x^{\pm} + H_{22}^{\pm} \xrightarrow{q_{max}} O_{22}^{\pm}$ $O_{22}^{\pm} + T_{22}^{\pm} \xrightarrow{q_{max}} x^{\pm} + x^{\pm} + (x_{p,n})^{+}$	$x^{\pm} + (x^{n-1})^{\pm} \xrightarrow{\gamma_p} x^{\pm} + (x^{n-1})^{\pm} + (x_{p,n})^{+}$	
$x^{\pm} + L_{23}^{\pm} \xrightleftharpoons[q_{max}]{q_{23}} H_{23}^{\pm} + B_{23}^{\pm}$ $x^{\mp} + H_{23}^{\pm} \xrightarrow{q_{max}} O_{23}^{\pm}$ $O_{23}^{\pm} + T_{23}^{\pm} \xrightarrow{q_{max}} x^{\pm} + x^{\mp} + (x_{p,n})^{-}$	$x^{\pm} + (x^{n-1})^{\mp} \xrightarrow{\gamma_p} x^{\pm} + (x^{n-1})^{\mp} + (x_{p,n})^{-}$	
$(x_{p,n})^{\pm} + G_{24}^{\pm} \xrightarrow{q_{24}} \emptyset$	$(x_{p,n})^{\pm} \xrightarrow{\gamma_p} \emptyset$	
$(x_{p,n})^{+} + L_{25} \xrightleftharpoons[q_{max}]{q_{max}} H_{25} + B_{25}$ $(x_{p,n})^{-} + LS_{25} \xrightleftharpoons[q_{max}]{q_{max}} HS_{25} + BS_{25}$ $(x_{p,n})^{-} + H_{25} \xrightarrow{q_{max}} \emptyset$	$(x_{p,n})^{+} + (x_{p,n})^{-} \xrightarrow{\eta} \emptyset$	

TABLE 3.1 (continued): DNA Implementation reactions, CRNs and the corresponding ODEs for the implementation of components, where, x_i and x_o denote the input and output of each individual component, respectively: (e) Multiplication, modelled using 17 DNA reactions - approximated to 7 chemical reactions. (f) Power component, modelled using 17(n-1) DNA reactions - approximated to 7(n-1) chemical reactions where, n is the power. The DNA implementation reaction rates are set to $q_i = 800$ /M/s ($i = 1, 2, ..21$), $q_{max} = 10^7$ /M/s and initial concentration of auxiliary species, $C_{max} = 1000$ nM. The reaction rate of annihilation, η, is set to $10 \cdot q_i C_{max}$. $\emptyset$ indicates inert or waste product [15; 31].

3.3 Forming functional circuits using linear/nonlinear operators

In this section, the way in which the individual linear and nonlinear operators described in the previous sections, can be combined to form a number of functional circuits is explained. Circuit designs to compute the fractional exponent of a signal, the absolute value of a signal, as well as the logarithm of arbitrary base, are presented. Designs based on exploiting the dynamics of covalent modification cycles are also presented and are shown to achieve significant reductions in circuit complexity.

Note that, the reaction rates and coefficients values listed for each complex circuit application are the manually tuned values that are well within their physical limits. The base-line or initial values used to simulate the results in this thesis are referred from the cited existing literature and references therein. An isolated component can be tested to perform its desired function using different sets of parameter values. However, when inserted in a multi-component system it might need to be tuned slightly again. For example, an isolated controller generates expected input-output characteristics using a set of parameter values but, when it is used in a closed loop feedback system to control a process, it needs to be tuned accordingly. It means, there can be multiple sets of parameter values for which the controller can generate a similar expected behaviour. However, we need to select the best suited parameter set that gives us better performance than others, for the system design purpose. Accordingly, the parameter values listed throughout the thesis have been collected.

3.3.1 Fractional exponent

Fractions are represented as the quotient a/b of two numbers, with numerator a and a non-zero denominator b. Here, it is illustrated how to compute the fractional exponent of a biomolecular signal $S^{\frac{m}{n}}$, where m and n are integers.

The Newton-Raphson method is a powerful technique in numerical analysis, based on the principle of linear approximation. It is used to find better approximations to the roots of a real-valued function. Due to the combination of simplicity

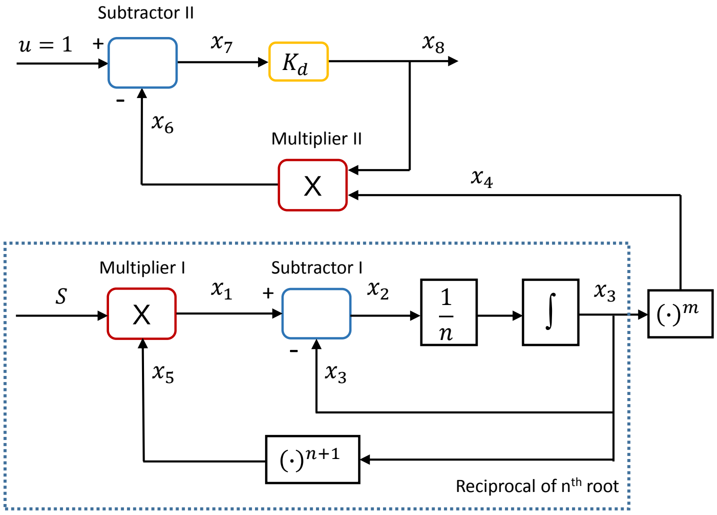


Figure 3.7: Computing fractional exponent: the system arrangement uses a Newton-Raphson block, a polynomial function and a divider where, the system output is x_8.

and power, it is one of the most widely used iterative methods. A reciprocal of n^{th} root function can be computed using Newton-Raphson in the following way:

$$x_{k+1} = x_k - \frac{S - \frac{1}{x_k^n}}{n x_k^{-n-1}} \tag{3.20}$$

where, S is the number for which its reciprocal of n^{th} root is to be computed. Rearranging (3.20), we get:

$$x_{k+1} - x_k = \frac{1}{n}\left(x_k - S x_k^{n+1}\right) \tag{3.21}$$

The left hand side of the (3.21) can be approximated by a derivative, which leads to the following:

$$\frac{dx}{dt} = \frac{1}{n}\left(x - S x^{n+1}\right) \tag{3.22}$$

The reciprocal of n^{th} root function can be obtained by taking integration on both sides of (3.22), which is essentially a scaled integration of the difference between the signal x and a product of Sx^{n+1}. The block diagram representation of this is shown by the dotted box in Fig. 3.7.

Now, rewriting $S^{\frac{m}{n}} = (S^{-\frac{1}{n}})^{-m} = \frac{1}{(S^{-\frac{1}{n}})^m}$, it can be seen that the fractional

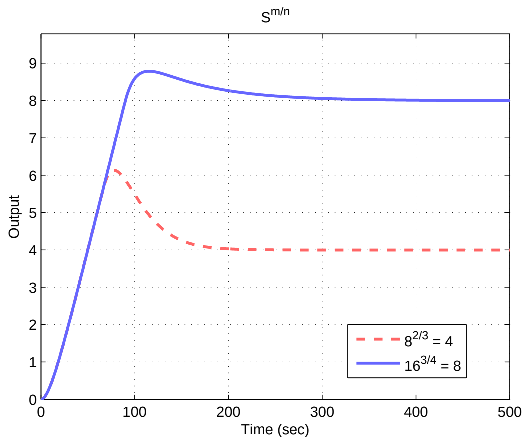


Figure 3.8: Computation of $S^{\frac{m}{n}}$. Two cases are considered: (a) Red (dashed), $S = 8$, $m = 2$, $n = 3$, i.e. $8^{\frac{2}{3}}$. (b) Blue (solid), $S = 16$, $m = 3$, $n = 4$, i.e. $16^{\frac{3}{4}}$.

exponent can be computed by first finding the reciprocal of n^{th} root and raised to the power of m before taking its inverse value. Thus, besides the computation of n^{th} root function, an additional polynomial function and a divider is required to compute the fractional exponent as depicted in Fig. 3.7. The simulation result is shown in Fig. 3.8, with reaction rates and coefficients (refer Table 3.1 for the CRNs of each individual component) set to be $\gamma_1 = 70$ /s, $\gamma_2 = 3$ /s, $\gamma_3 = 3$ /s, $\gamma_4 = 3$ /s, $\gamma_5 = 0.1$ /s, $k_{s1} = 100$ /s, $k_{s2} = 0.00001$ /s and $K_d = 10,000$. The signal, u is set to 1 since the reciprocal of $(S^{-\frac{1}{n}})^m$ is computed.

3.3.2 Absolute value

An absolute value (or modulus) can be defined as a non-negative value of a real number x, regardless of its sign. It can be understood as a *distance* of any number from zero on a number line and represented as $|x|$ or $abs(x)$. Namely, for $x = -5$ the absolute value is 5; and for $x = 5$ it is also 5.

Two approaches are proposed here to compute the absolute value of a given signal u. These approaches are compared to show that one of them can be implemented using significantly fewer chemical reactions than the other. The first approach uses a combination of operators whereby the input signal u is first squared before taking its square root, as shown in Fig. 3.9. The Newton-Raphson method is used for the computation of the square root y. A total of 7 (to compute the square)

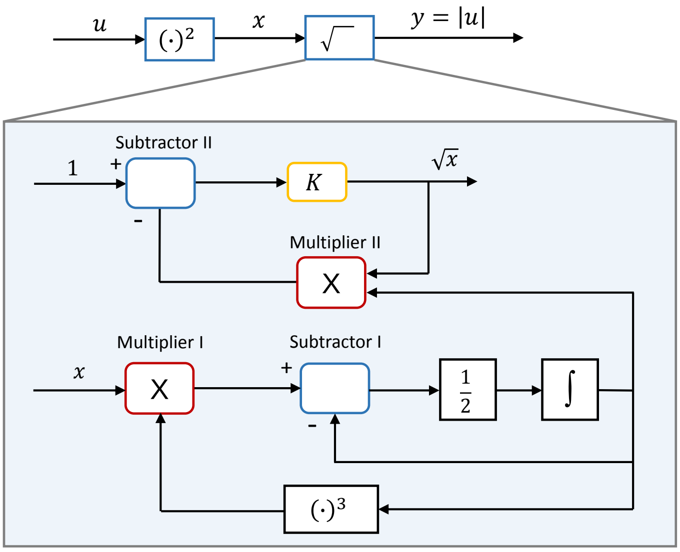


Figure 3.9: Computing absolute value using combination of operators. The input u is first squared before taking its square root using Newton-Raphson method.

+ 55 (to compute the square root) = 62 chemical reactions, are required to realise the computation of the absolute value (refer Table 3.1 for the CRNs of each individual component). The number of reactions required is quite high, mainly due to the presence of the fractional exponent.

The other approach uses the two regimes of the covalent modification cycle (see Chapter 2, Section 2.1.3), i.e. signal-transducing and threshold-hyperbolic operating regimes, for which the block diagram is shown in Fig. 3.10 (refer Table 3.1 for the CRNs of each individual component). The threshold-hyperbolic regime has a non-responsive region called a dead zone, followed by a hyperbolic response. To compute the absolute value, the dead zone range is required to be operated such that it does not respond to the negative valued input signal u and only responds when u is non-negative. In addition, u should respond in a linear manner following the dead zone. Note that, in the threshold-hyperbolic regime, any hyperbolic response contains an almost linear region when the input signal is small. Taking advantage of this property, one can ensure that the required threshold-hyperbolic regime has a linear instead of hyperbolic response, after the dead-zone region.

On the other hand, the signal-transducing regime has a linear region followed by a saturated response. This makes this regime suitable for responding

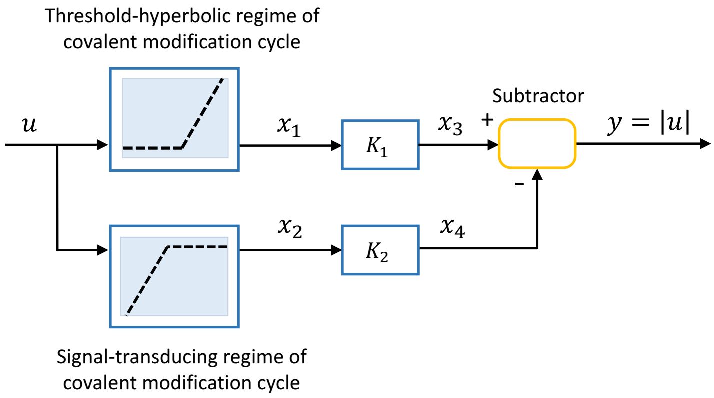


Figure 3.10: Block diagram of a circuit to compute the absolute value using the threshold-hyperbolic and signal-transducing regimes from the covalent modification cycle.

only to non-positive signals and not to strictly positive input signals. By combining these two regimes (signal-transducing and threshold-hyperbolic) with two gain components and one subtraction operator, 45 reactions are required to compute the absolute value, a reduction in circuit complexity of 17%, compared to the first approach.

The reaction rates to achieve this threshold-hyperbolic response in covalent modification cycle are set to $k_1 = 0.0027$ /M/s, $k_2 = 16,640$ /s, $k_3 = 0.043$ /M/s, $k_4 = 0.008$ /s and $X_e = 3.5$ M, for eq. (2.11) in Chapter 2, Section 2.1.3. The two gain components, K_1 and K_2, are intorduced for scaling purposes. For the signal-transducing response, suppose that due to the limitations imposed by the system, a unity gradient of the linear response cannot be achieved, resulting in the gradient of the linear response to be 20. In this case, the gain component is set to $K_2 = 1/20$. Likewise, the signal-transducing response is achieved with, $k_1 = 5$ /M/s, $k_2 = 100$ /s, $k_3 = 5$ /M/s, $k_4 = 630$ /s and $X_e = 1.8$ M, for eq. (2.11) in Chapter 2, Section 2.1.3.

Fig. 3.11 illustrates the simulation results for six different input signals, $u = 1, 2, ..., 6$. At time, $t = 10,000$ s, these input signals, u are switched to their negative counterpart ranging from $u = -1, -2, ..., -6$. The performance comparison of the circuit designed using a combination of operators and the covalent modification cycle is shown. Both the circuits perform remarkably well, although for the covalent modification cycle when $u = 1$ and 6 some deviations are observed. This is because

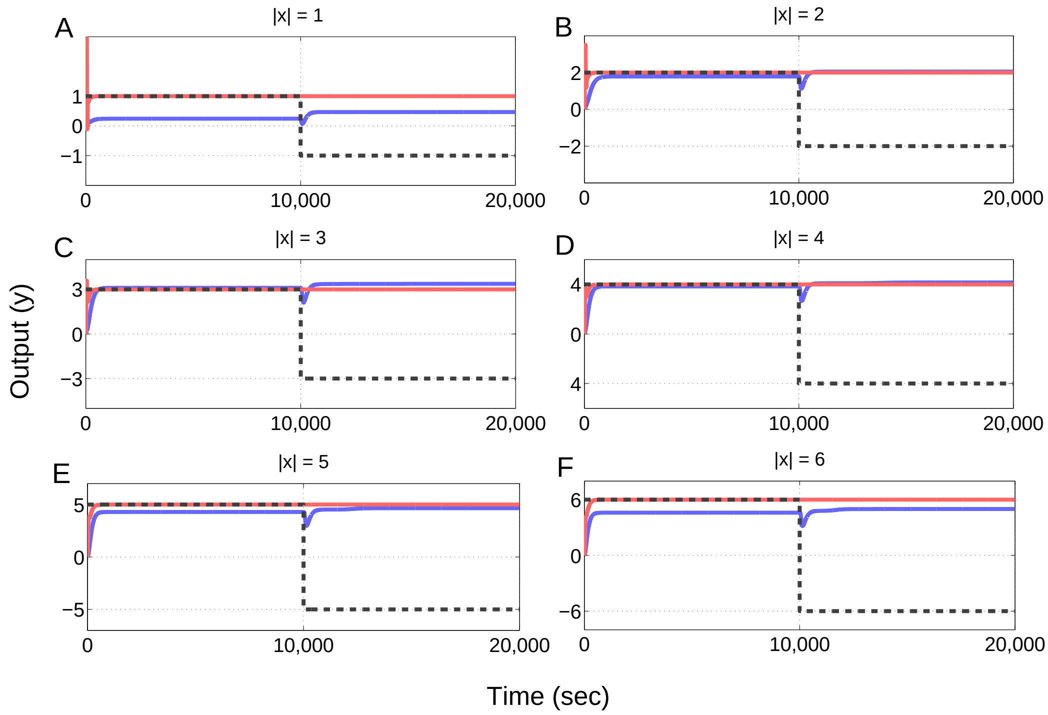


Figure 3.11: Computation of absolute value: In all the simulations- Black dashed line: the input value. Red solid line: using combination of operators. Blue solid line: using threshold-hyperbolic and signal-transducing regimes from the covalent modification cycle. (A) Input, $u = \pm 1$, (B) Input, $u = \pm 2$. (C) Input, $u = \pm 3$. (D) Input, $u = \pm 4$. (E) Input, $u = \pm 5$. (F) Input, $u = \pm 6$.

the threshold-hyperbolic and signal-transducing responses are not a perfect match to the ideal desired responses.

This is illustrated in Fig. 3.12, which shows a comparison of the ideal response with the simulated response for both the regimes. The ideal threshold-hyperbolic response is to have a dead-zone for a strictly negative input signal u and a linear response with unity gradient for the non-negative input signal as in Fig. 3.12 (A). The ideal signal-transducing response is to have a linear response with non-unity gradient for negative input signals and no response to strictly positive input signals, as shown in Fig. 3.12 (B). These response characteristics are intentionally chosen to illustrate the purpose of exploiting the gain operator that can be used for scaling. The gain operator could prove useful when there is difficulty in achieving the ideal unity gradient linear response in both the threshold-hyperbolic and signal-transducing regime. Thus, inclusion of this gain provides flexibility to

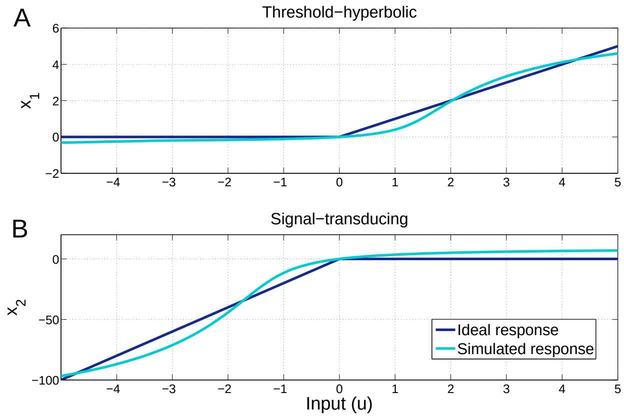


Figure 3.12: Comparison between ideal and simulated signal-transducing response. (A) Positive component: threshold-hyperbolic response. (B) Negative component: signal-transducing response.

achieve alternative similar responses when the initially intended responses (i.e. with unity gradient) cannot be attained. Since the threshold-hyperbolic response has a linear response with unity gradient, there is no requirement for the gain block, K_1 or equivalently, $K_1 = 1$. Though the ideal response for threshold-hyperbolic should have zero output value when the input value, u is strictly negative; and be strictly positive for the signal transducing regime. Our simulated response for threshold-hyperbolic on the other hand, show a small non-zero output for those ranges of input signal. Moreover, the linear responses for both threshold-hyperbolic and signal-transducing regimes are not exactly linear. These two factors contribute to the observed deviation of the simulation results in Fig. 3.11 for the case of input signal $u = 1$ and 6. Nevertheless, the simulation shows excellent results for the range of input signal $u = 2,3,4,5$.

3.3.3 Logarithm of arbitrary base

Consider the operation $c = \log_b a$, i.e. computing the logarithm of a to the base b. This logarithm can be computed through the change of logarithm base, i.e. $c = \frac{\ln a}{\ln b}$, where ln denotes the natural logarithm. In other words, c can be realised as a ratio of $\ln a$ and $\ln b$. Several numerical methods exist to compute the natural logarithm. The most commonly used method is to use Taylor series, but this method accurately

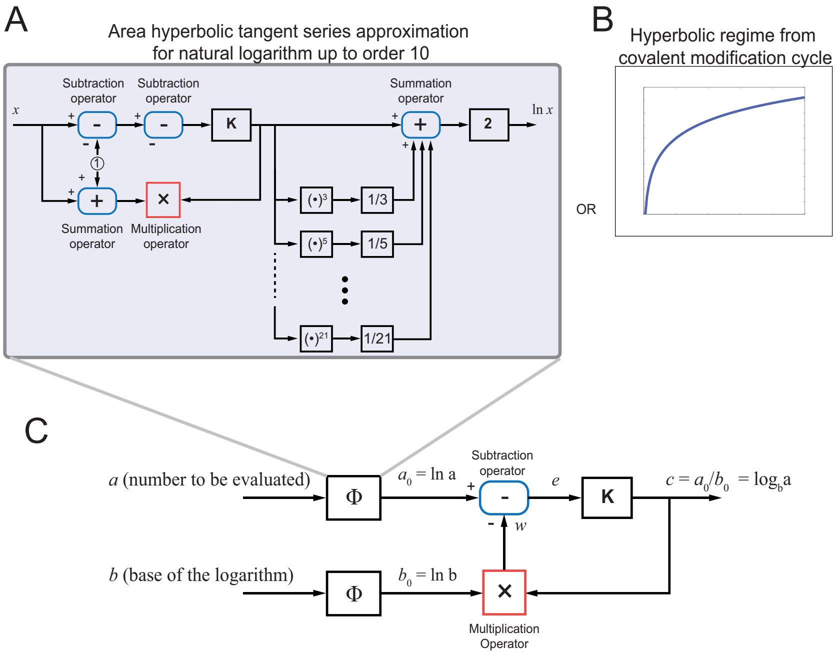


Figure 3.13: Computation of logarithm of arbitrary base. (A) Block diagram representation to compute natural logarithm based on area hyperbolic tangent series approximation of order $l = 10$. (B) The hyperbolic regime obtained from signalling cycle. (C) Block diagram representation to compute logarithm of arbitrary base by computing the ratio of two natural logarithms.

computes the logarithm of a number, denoted as x, only within the range $0 < x < 2$ [82]. A more efficient method to compute the natural logarithm for $x \geq 2$ is based on the area hyperbolic tangent series approximation [83]. Thus, using such approximation, the natural logarithm can be computed as follows:

$$\ln(x) = \ln\left(\frac{z-1}{z+1}\right) = 2\sum_{i=0}^{l}\frac{z^{2i+1}}{2i+1} \tag{3.23}$$

where l is the order of the series. The larger the order l is, the better the approximation, but the higher the complexity of the circuit. Here, $l = 10$ is chosen as this order allows us to compute the logarithm of numbers up to 10.

The block diagram of a circuit that can compute the natural logarithm using the area hyperbolic tangent series approximation of order $l = 10$ is shown in Fig. 3.13(A). This circuit uses a combination of several linear and nonlinear operators; summation, subtraction, gain, multiplication and power exponent, each

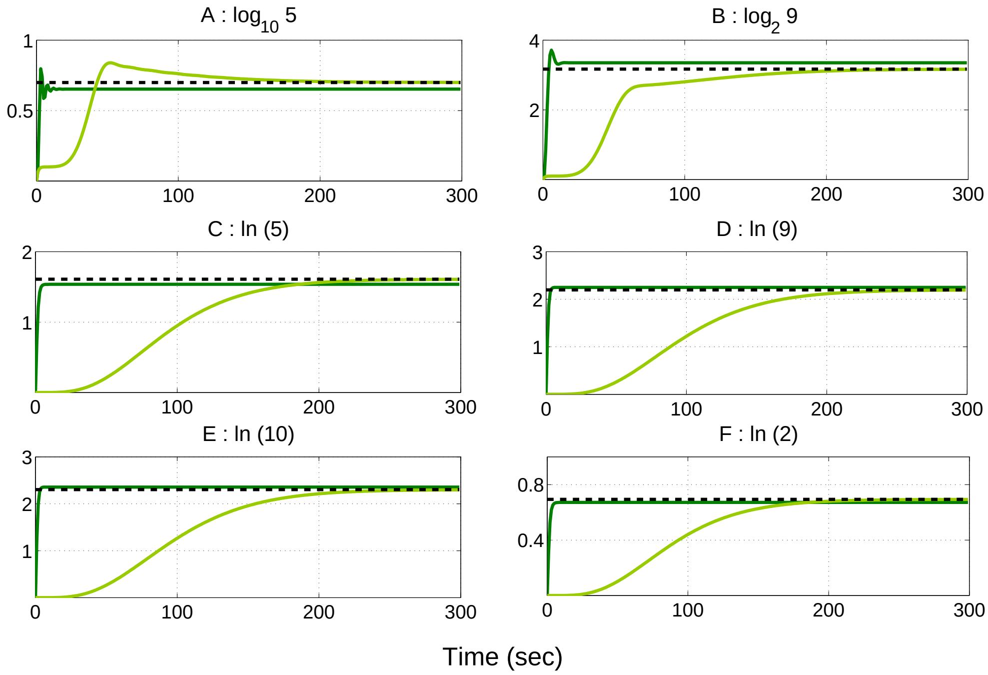


Figure 3.14: Simulation results of natural logarithm and logarithm of arbitrary base. In all simulations, Black dashed: Actual value, light Green solid: Using area hyperbolic tangent series approximation up to order 10. Dark Green solid: Using hyperbolic regime from signalling cycle. (A) $\log_{10} 5$. (B) $\log_2 9$. (C) $\ln 5$. (D) $\ln 9$. (E) $\ln 10$. (F) $\ln 2$.

of which may be implemented using a number of chemical reactions, details of which are given in Section 3.1.1 and 3.2.1. With the area hyperbolic tangent series approximation and $l = 10$, the circuit in Fig. 3.13(C) requires 13 summation and subtraction operators, 1 multiplication operator, 10 power exponent operators with exponents 3, 5,...,21 and 12 gain operators (refer Table 3.1 for the CRNs of each individual component). This results in a total of 928 chemical reactions. To compute the logarithm of arbitrary base, as shown in Figure 3.13(C), it requires one more Φ that computes the second natural logarithm and one each for the subtraction, multiplication and gain operator. Thus, this circuit requires a total of 1875 chemical reactions, which makes it completely intractable from an experimental point of view.

The huge number of chemical reactions required to implement the circuit described above means that alternative, more efficient, designs are required. It is noted that the response characteristics of a natural logarithm resemble the hyperbolic regime of the covalent modification cycle (see Chapter 2, Section 2.1.3, Fig. 2.3(b)) thus making this regime potentially useful for computing the natural logarithm. Interestingly, this response is not governed by the order of the series approximation. Thus, as long as one can obtain the appropriate reaction rates for k_1 to k_4 for CRN (2.10) in Chapter 2, Section 2.1.3, the natural logarithm can be computed using the hyperbolic regime. Moreover, this approach requires only 14 chemical reactions. To compute the logarithm of arbitrary base using this approach, the Φ block in Fig. 3.13(C) is replaced with the covalent modification cycle reactions that produce the hyperbolic regime, as shown in Fig. 3.13(B). This results in a total of 47 chemical reactions, more than 90% reduction in circuit complexity than with the area hyperbolic approach.

Simulation results for computing $\log_{10} 5$ and $\log_2 9$ using covalent modification cycle and area hyperbolic tangent series are shown in Fig. 3.14. To implement the hyperbolic response, reaction rates for (2.11) in Chapter 2, Section 2.1.3 are; $k_1 = 0.22$ /M/s, $k_2 = 0.43$ /s, $k_3 = 1.03$ /M/s, $k_4 = 35.10$ /s, and $X_e = 1$ M.

For both approaches, the computed logarithms are close to the actual value, however the circuit based on the covalent modification cycle is significantly faster in settling to the correct steady-state value, even though it uses far fewer chemical reactions. An alternative approach for the biological computation of logarithms has been designed and implemented in [84]. This approach utilises transcriptional regulation, which requires a host cell, while the approach presented here can be implemented in cell-free conditions (e.g. using DSD framework). Moreover, [84] considers only the computation of the natural logarithm, while our approach enables the computation of logarithms of arbitrary base.

Chapter 4

Design and Implementation of a Biomolecular Quasi Sliding Mode Controller

Exploiting *chemical reaction networks* (CRNs) as a programming language for the design of complex circuits and networks, this chapter shows how an important class of nonlinear feedback controllers can be designed to realize input-output dynamics that approximate an ideal *sliding mode controller* (SMC). The kinetics of the required chemical reactions can then be implemented as enzyme-free, enthalpy/entropy driven DNA reactions using a toehold mediated *DNA strand displacement* (DSD) mechanism. In this chapter, the approach of [31] and [17] is extended to allow the implementation of nonlinear feedback controllers. It is demonstrated with simulation results, that the closed loop response of the nonlinear *quasi sliding mode* (QSM) controller outperforms a traditional linear controller by facilitating much faster tracking response dynamics without introducing overshoots in the transient response. The resulting controller is highly modular and is less affected by retroactivity effects than standard linear designs.

Though some of the operators are mentioned already in Chapter 3, they are briefly described again here to maintain the nomenclature and continuity, in the context of the design of a closed-loop feedback system.

4.1 Saturation nonlinearity

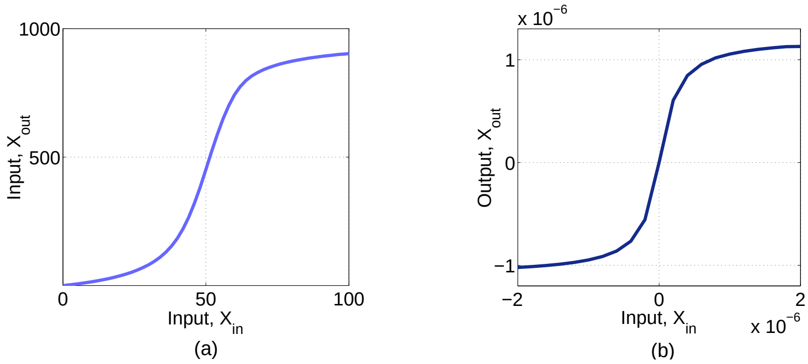


Figure 4.1: Input-output characteristic curve for: (a) covalent modification cycle where, positive output is produced for the positive values of the input signal; i.e. operates in the 1st quadrant. (b) Chemical reactions implementing saturation nonlinearity where, negative output is produced for the negative values of input. Similarly, positive output is produced for the positive values of input signal; i.e. operates in 4 quadrants.

Four operating regimes of a covalent modification cycle are discussed in Chapter 2, Section 2.1.3, where one of these is an ultrasensitive regime. In [59], the ultrasensitive regime operates for positive input values and produces a sigmoidal input-output relationship. In that case, for small values of the input, the output is nearly zero. Then, after a certain threshold value of the input signal, the output immediately rises to its highest value and remains saturated for the higher values of the input signal. In order to utilise this interesting behaviour to design a controller one should consider the fact that generally, in a closed-loop feedback system, the input of a controller is an error signal generated from the summation junction. Since the error is computed as the difference between a reference signal and the actual output signal, the resulting error value can be either positive or negative. In this section, it is thus shown how the input-output characteristic response of the ultrasensitive regime can be mapped from the 1st quadrant into 4 quadrants, as illustrated in Fig. 4.1, in order to allow this system to be used for controller design.

In practice, physical systems often exhibit some nonlinearity due to the presence of properties such as friction or hysteresis, actuator saturation, viscosity, chemical kinetics, geometric functions in robotics, and so on [37; 85]. The factors whose

Chemical reactions	
$X_{in}^{\pm} + X_{p}^{\pm} \xrightarrow{k_1} X_{C_1}^{+}$,	$X_{out}^{\pm} + X_{e} \xrightarrow{k_3} X_{C_2}^{\pm}$,
$X_{in}^{\mp} + X_{p}^{\pm} \xrightarrow{k_1} X_{C_1}^{-}$,	$X_{C_2}^{\pm} \xrightarrow{k_4} X_{p}^{\pm} + X_{e}$,
$X_{C_1}^{\pm} \xrightarrow{k_2} X_{out}^{\pm} + X_{in}^{\pm}$,	$X_{C_2}^{+} + X_{C_2}^{-} \xrightarrow{\eta} \phi$,
$X_{C_1}^{+} + X_{C_1}^{-} \xrightarrow{\eta} \phi$,	$X_{p}^{+} + X_{p}^{-} \xrightarrow{\eta} \phi$,
$X_{out}^{+} + X_{out}^{-} \xrightarrow{\eta} \phi$.	
Ordinary differential equations	
$\frac{dX_{out}}{dt} = k_2 X_{C_1} - k_3 X_{out} X_e$,	$\frac{dX_{C_1}}{dt} = k_1 X_{in} X_p - k_2 X_{C_1}$,
$\frac{dX_{C_2}}{dt} = k_3 X_{out} X_e - k_4 X_{C_2}$,	$\frac{dX_p}{dt} = -k_1 X_{in} X_p + k_4 X_{C_2}$.

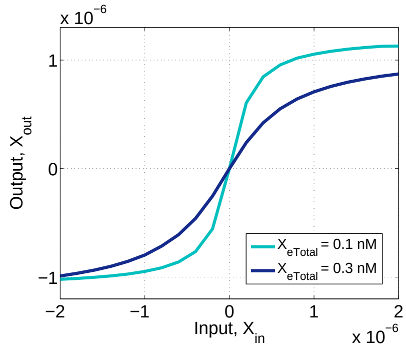


Figure 4.2: Saturation nonlinearity: The chemical reactions and ODEs used to generate the *saturation nonlinearity* (SN) behaviour for the operating range of input signal $X_{in} = [-2, 2]$. The slope of curve can be modified by tuning the key parameter, X_{eTotal}.

static characteristics between input and output does not satisfy linear relationship are defined as nonlinear factors. Some frequently appearing nonlinearities are called as *common nonlinearities*. They include saturation, hysteresis, backlash and deadzone among which the particular behaviour of interest here is a saturation nonlinearity. This is because the ultrasensitive operating regime of the covalent modification cycle, as discussed in Chapter 3, Section, 2.1.3, imitates the nature of a saturation nonlinearity, and can thus be exploited for the design of nonlinear biomolecular systems or controller.

For the chemical reactions listed in Fig. 4.2, k_1, k_3 are binding rates, k_2, k_4 are the catalytic reaction rates and η is the annihilation rate. As highlighted in Chapter 3, Section 2.1.3, these rates can be tuned to obtain one of the four possible operating regimes from the covalent modification cycle. Using the same CRN and through the appropriate tuning of the aforementioned rates, a *saturation nonlinearity* (SN) can be obtained as shown in Fig. 4.2. The ideal expected SN behaviour is when the input values are negative, the output should be negative. Similarly, when the input is positive, the output should be positive. The slope of the curve can be modified by setting different values of the parameter, $X_{e_{Total}}$ where, $X_{e_{Total}} = X_e + X_{C_2}$.

4.2 Implementing Nonlinear Feedback Controllers

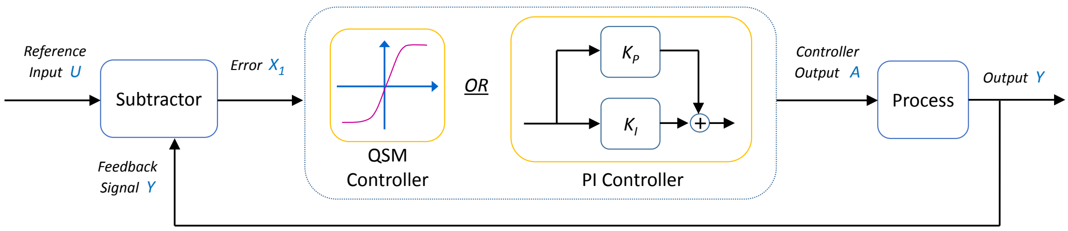


Figure 4.3: A prototype embedded biomolecular closed loop feedback control system.

This section shows how an important class of nonlinear feedback controllers can be designed using chemical reactions and implemented via DSD reactions. The nonlinear QSM controller is designed and its performance is compared with a classical linear controller. The simulated results demonstrate that the closed loop response of the nonlinear QSM controller outperforms a traditional linear controller by facilitating much faster tracking response dynamics without introducing overshoots in the transient response. The controller here is implemented on a prototype embedded closed loop feedback system that consists of three individual modules, a subtractor, a controller and a biomolecular process to be controlled, each realized by mass action kinetics at a molecular level and interconnected using a modular approach as shown in Fig. 4.3. In contrast to previous implementations of DNA-based feedback controllers, the biomolecular process to be controlled here is both dynamic and nonlinear. Note also that the subtractor module must be represented as a dynamical system, unlike in standard feedback control systems which assume the availability of an ideal subtractor. Analysis of the closed loop performance of the QSM controller reveals significant performance advantages compared to a linear *proportional+integrator* (PI) controller, particularly when retroactivity effects (see [40], [86], and [41]) are taken into account.

4.2.1 Biomolecular Quasi-sliding mode controller

Taking inspiration from the ultrasensitive input-output behaviour exhibited by covalent modification cycle signaling cascades, [87–89], a set of chemical reactions is presented that can be used to generate switch-like input-output responses. In Section 4.1, it is explained how the SN behaviour is generated using a set of chemical reactions. The main difference in the ultrasensitive responses exhibited by the system in [59] and the QSM controller described here is that the QSM controller has been designed to operate for positive as well as negative values of the input signal, which is the error signal in the feedback control loop.

Consider the following CRN, where a signal x is represented as $x = x^+ - x^-$ which is the difference between the concentrations of two DNA strands x^+ and x^-, having a free toehold each when implemented as elementary DNA reactions (see Chapter 2, Section 2.1.2). The chemical reactions that construct a nonlinear QSM feedback controller are:

$$X_1^{\pm} + B^{\pm} \xrightarrow{k_{b_1}} X_2^{+}, \tag{4.1a}$$
$$X_1^{\mp} + B^{\pm} \xrightarrow{k_{b_1}} X_2^{-}, \tag{4.1b}$$
$$X_2^{\pm} \xrightarrow{k_{c_1}} A^{\pm} + X_1^{\pm}, \tag{4.1c}$$
$$X_2^{+} + X_2^{-} \xrightarrow{\eta} \phi, \tag{4.1d}$$
$$A^{+} + A^{-} \xrightarrow{\eta} \phi, \tag{4.1e}$$
$$A^{\pm} + X_3 \xrightarrow{k_{b_2}} X_4^{\pm}, \tag{4.1f}$$
$$X_4^{\pm} \xrightarrow{k_{c_2}} B^{\pm} + X_3, \tag{4.1g}$$
$$X_4^{+} + X_4^{-} \xrightarrow{\eta} \phi, \tag{4.1h}$$
$$B^{+} + B^{-} \xrightarrow{\eta} \phi. \tag{4.1i}$$

Here, k_{b_1} and k_{b_2} denote the binding reaction rates whereas k_{c_1} and k_{c_2} denote the catalytic reaction rates and η is the degradation rate. The signal X_1 is the input and the signal A is the output of the controller. The CRN (4.1) realizes an ultrasensitive switch-like input-output response, as illustrated in Fig. 4.4 for the input range $u = [-4 \;\; 4]$. Interestingly, the input-output response of (4.1) can be made to closely approximate the ideal switch implemented using a SMC, by tuning the key parameter, X_{3Total}, where $X_{3Total} = X_3 + X_4$, defined as the total concentration of X_3 and X_4. It is assumed that X_{3Total} is conserved through the lifetime of the process and therefore is set to a constant value.

The CRN (4.1) is an approximation of elementary DNA reactions which can be realized using Visual DSD software, [79]. Using the software package Visual DSD [79], the strand displacement mechanism of the catalysis, bimolecular, degradation and annihilation reactions is illustrated in Figs. 2.5 to 2.9. Now, using mass action kinetics, (4.1) can be represented by the following set of ODEs:

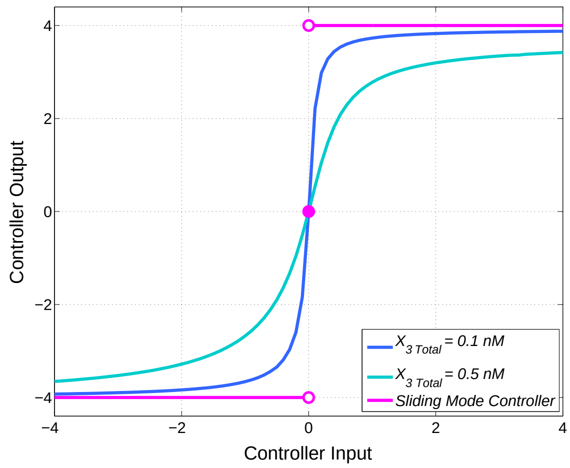


Figure 4.4: Input-output characteristics of an ideal sliding mode controller and quasi sliding mode controller for different values of the tuning parameter X_{3Total}.

$$\frac{dA}{dt} = k_{c_1}X_2 - k_{b_2}AX_3, \quad (4.2a) \qquad \frac{dB}{dt} = -k_{b_1}X_1B + k_{c_2}X_4, \quad (4.2c)$$

$$\frac{dX_2}{dt} = k_{b_1}X_1B - k_{c_1}X_2, \quad (4.2b) \qquad \frac{dX_4}{dt} = k_{b_2}AX_3 - k_{c_2}X_4. \quad (4.2d)$$

From (4.2) it can be seen that: $\frac{dA}{dt} + \frac{dX_2}{dt} + \frac{dB}{dt} + \frac{dX_4}{dt} = 0$. Hence, $A+B+X_2+X_4 = \text{constant} \doteq S_{qsm}$; where, S_{qsm} denotes the total concentration of four signal species. Accordingly, it can be said that signal B is variable and depends on the dynamic signals A, X_2, X_4. Thus, for simulations B is constructed as, $B = S_{qsm} - A - X_2 - X_4$. Since, X_1 also varies over time this means that the term $k_{b_1}X_1B$ in (4.2b) is nonlinear.

Now, from sliding mode control theory, a perfect SMC can be represented by a relay nonlinearity (see [34; 36; 37]). As shown in Fig. 4.4, this can be obtained as the limiting case of a controller implemented using (4.2). For example, as $X_{3Total} \to 0$, the output A of the controller can be described by the following relay-type saturation nonlinearity (see Fig. 4.4):

$$A(t) = k_{SMC} \cdot \text{sgn}(X_1(t)), \quad (4.3)$$

where $\text{sgn}(\cdot)$ denotes the signum function and $X_1(t)$ is the input to the controller (the error signal generated by the subtractor). Such a controller has a discontinuity on the straight line $X_1 = 0$ which is traditionally referred to as the *sliding manifold* $\sigma \stackrel{def}{=} X_1 = 0$, where σ is the sliding variable. The control signal A, defined by (4.3), is therefore designed to force the system to move towards the sliding manifold $\sigma = 0$ (the *reaching phase* of SMC) and then maintain this condition (i.e. $\sigma = 0$) for all future time (the *sliding phase* of SMC).

In practice, however, implementations of perfect sliding mode controllers cause the system's closed loop response to exhibit a zigzag motion of small amplitude and high frequency, due to imperfections in switching devices and delays [34; 36; 37]. This effect, known as *chattering*, is typically avoided by using continuous/smooth approximations of the discontinuous SMC, resulting in a so-called QSM controller.

The controller implemented using (4.2) is an example of such a function, since it approximates the nonlinearity $\text{sgn}(X_1)$. With a QSM controller, there is no ideal sliding mode in the closed loop system as the sliding variable (error) cannot be driven exactly to zero in a finite time, [34]. However, if the QSM controller is made more ultrasensitive (for example, by decreasing X_{3Total}), the input-output behaviour of our QSM controller approaches the limiting case of an ideal SMC, as illustrated in Fig. 4.4, and then the error signal can be made as small as desired.

4.2.2 Nonlinear process to be controlled

To act as a challenging benchmark control problem (reference tracking), a process to be controlled is selected that is composed of both unimolecular and bimolecular reactions, given as:

$$A^{\pm} + X_5^{\pm} \xrightarrow{k_{r1}} X_6^{+}, \tag{4.4a}$$

$$A^{\pm} + X_5^{\mp} \xrightarrow{k_{r1}} X_6^{-}, \tag{4.4b}$$

$$X_6^{\pm} \xrightarrow{k_{r2}} Y^{\pm} + X_5^{\pm}, \tag{4.4c}$$

$$Y^{+} + Y^{-} \xrightarrow{\eta} \phi. \tag{4.4d}$$

$$Y^{\pm} \xrightarrow{k_{r3}} \phi, \tag{4.4e}$$

Here, the process input signal is A and output signal is Y. k_{r_1} is a binding reaction rate, k_{r_2} is the catalytic reaction rate, k_{r_3} is the degradation rate, and η is the annihilation reaction rate. These reaction rates and their values are as listed further in Section 4.3.1 with simulation results.

This process was chosen because application of standard Michaelis-Menten kinetics to these reactions results in a set of ODEs with nonlinear response dynamics, given by:

$$\frac{dX_5}{dt} = -k_{r_1}AX_5 + k_{r_2}X_6, \quad (4.5a)$$

$$\frac{dX_6}{dt} = k_{r_1}AX_5 - k_{r_2}X_6, \quad (4.5b)$$

$$\frac{dY}{dt} = k_{r_2}X_6 - k_{r_3}Y. \quad (4.5c)$$

From (4.5a) and (4.5b): $\frac{dX_5}{dt} + \frac{dX_6}{dt} = 0$. Hence, it can be concluded that $X_{Total} \doteq X_5 + X_6$ is conserved through the lifetime of the process and therefore it is set to a constant value.

In the context of the feedback system shown in Fig. 4.3, the process input signal is the controller output A and the process output signal Y is fed back as an input signal to the subtractor. In the control literature it is well known that nonlinear systems are in general more difficult to control than linear systems. Also, previous work on the implementation of linear feedback controllers using nucleic acids considered only a static process to be controlled [17; 31]. The system described here represents the first attempt to design a DNA-based biomolecular feedback controller for a complex nonlinear biomolecular process.

4.2.3 Subtractor

Following [31] and [17], the subtraction $U - Y$ of two signals U and Y is implemented in this section. The subtraction operation can be achieved using the following set of reactions:

$$U^{\pm} \xrightarrow{k_s} U^{\pm} + X_1^{\pm}, \quad (4.6a)$$

$$Y^{\pm} \xrightarrow{k_s} Y^{\pm} + X_1^{\mp}, \quad (4.6b)$$

$$X_1^{\pm} \xrightarrow{k_s} \phi, \quad (4.6c)$$

$$X_1^{+} + X_1^{-} \xrightarrow{\eta} \phi. \quad (4.6d)$$

Here, signals U and Y are the inputs and X_1 is the output of the subtractor. In other words, the value of signal X_1 being produced is equivalent to that of the difference between the two input signals, U and Y. In addition, both the catalysis reaction rates in (4.6a) to (4.6b) are set to be equal to the degradation rate in (4.6c). Note that this subtractor module itself is a dynamical system and produces the desired result, i.e., subtraction of the two input signals, as its steady-steady output. Applying mass action kinetics to (4.6) gives:

$$\frac{dX_1}{dt} = k_s(U - Y - X_1). \tag{4.7}$$

By choosing a higher value of k_s, the response of the subtractor can be sped up so that the required steady-state value $U - Y$ can be computed more rapidly. In the context of the feedback system shown in Fig. 4.3, the inputs to the subtractor comprise the reference input signal U and the plant output Y while its output X_1 is fed as the input to the controller.

4.2.4 PI Controller

For the purposes of evaluating the performance of the designed nonlinear QSM controller, a linear PI controller [90; 91] has also been implemented. Following the approach of [31] and [17], the following representation for the PI controller is obtained — the proposed chemical reactions are slightly different from the ones given in [31] and [17] because they have been modified for the feedback system in Fig. 4.3. The PI controller is made up of an integrator implemented via the reactions:

$$X_1^{\pm} \xrightarrow{k_I} X_1^{\pm} + X_2^{\pm}, \tag{4.8a}$$

$$X_2^{+} + X_2^{-} \xrightarrow{\eta} \phi. \tag{4.8b}$$

and a proportional gain, implemented as:

$$X_1^{\pm} \xrightarrow{k_p} X_1^{\pm} + A^{\pm}, \tag{4.9a}$$

$$X_2^{\pm} \xrightarrow{k_c} X_2^{\pm} + A^{\pm}, \tag{4.9b}$$

$$A^{\pm} \xrightarrow{k_d} \phi, \tag{4.9c}$$

$$A^{+} + A^{-} \xrightarrow{\eta} \phi. \tag{4.9d}$$

Here, the signal X_1 is the input and A is the output. Furthermore, k_I, k_p and k_c denote the catalytic reaction rates while k_d denotes the degradation rate and η denotes annihilation rate. The parameter values are listed in Section 4.3.1 with simulation results. Using mass action kinetics, the following ODE representation is obtained for the PI controller:

$$\frac{dX_2}{dt} = k_I X_1, \qquad (4.10a) \qquad \frac{dA}{dt} = k_p X_1 + k_c X_2 - k_d A. \qquad (4.10b)$$

where, k_I denotes the integral gain and k_p denotes proportional gain.

The linear components of the feedback control system, i.e. the subtractor and PI controller, are built using a combination of catalysis–Fig. 2.6, degradation –Fig. 2.8 and annihilation –Fig. 2.9 reactions, given in Chapter 2. The non-linear components, i.e. the QSM controller and the process to be controlled use bimolecular–Fig. 2.7 reaction in addition to the above reactions, given in Chapter 2.

Here, the DNA implementation reactions, approximated CRNs and relevant ODEs for each module of the closed loop feedback control system shown in Fig. 4.3 are collected and presented in Table 4.1.

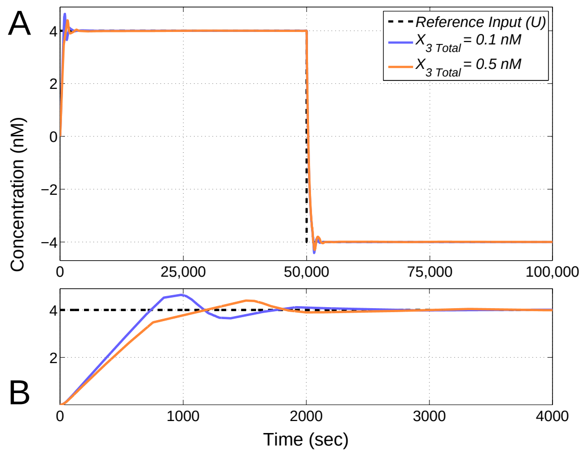
DNA Implementation	CRNs	ODEs
(a) QSM Controller		
$X_1^\pm + L_1^\pm \underset{q_{max}}{\overset{q_1}{\rightleftharpoons}} H_1^\pm + B_1^\pm$		
$B^\pm + H_1^\pm \xrightarrow{q_{max}} O_1^\pm + \phi$	$X_1^\pm + B^\pm \xrightarrow{k_{b_1}} X_2^+$	
$O_1^\pm + T_1^\pm \xrightarrow{q_{max}} X_2^+$		
$X_1^\mp + L_2^\pm \underset{q_{max}}{\overset{q_2}{\rightleftharpoons}} H_2^\pm + B_2^\pm$		
$B^\pm + H_2^\pm \xrightarrow{q_{max}} O_2^\pm + \phi$	$X_1^\mp + B^\pm \xrightarrow{k_{b_1}} X_2^-$	
$O_2^\pm + T_2^\pm \xrightarrow{q_{max}} X_2^-$		
$X_2^\pm + G_3^\pm \xrightarrow{q_3} \phi + O_3^\pm$	$X_2^\pm \xrightarrow{k_{c_1}} A^\pm + X_1^\pm$	
$O_3^\pm + T_3^\pm \xrightarrow{q_{max}} A^\pm + X_1^\pm$		
$X_2^+ + L_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_4 + B_4$		$\frac{dA}{dt} = k_{c_1}X_2 - k_{b_2}AX_3$
$X_2^- + LS_4 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_4 + BS_4$	$X_2^+ + X_2^- \xrightarrow{\eta} \phi$	
$X_2^- + H_4 \xrightarrow{q_{max}} \phi$		
$A^+ + L_5 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_5 + B_5$		$\frac{dX_2}{dt} = k_{b_1}X_1B - k_{c_1}X_2$
$A^- + LS_5 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_5 + BS_5$	$A^+ + A^- \xrightarrow{\eta} \phi$	
$A^- + H_5 \xrightarrow{q_{max}} \phi$		
$A^\pm + L_6^\pm \underset{q_{max}}{\overset{q_6}{\rightleftharpoons}} H_6^\pm + B_6^\pm$		$\frac{dX_4}{dt} = k_{b_2}AX_3 - k_{c_2}X_4$
$X_3 + H_6^\pm \xrightarrow{q_{max}} O_6^\pm + \phi$	$A^\pm + X_3 \xrightarrow{k_{b_2}} X_4^\pm$	
$O_6^\pm + T_6^\pm \xrightarrow{q_{max}} X_4^\pm$		
$X_4^\pm + G_7^\pm \xrightarrow{q_7} \phi + O_7^\pm$	$X_4^\pm \xrightarrow{k_{c_2}} B^\pm + X_3$	$\frac{dB}{dt} = -k_{b_1}X_1B + k_{c_2}X_4$
$O_7^\pm + T_7^\pm \xrightarrow{q_{max}} B^\pm + X_3$		
$X_4^+ + LS_8 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_8 + B_8$		
$X_4^- + LS_8 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_8 + BS_8$	$X_4^+ + X_4^- \xrightarrow{\eta} \phi$	
$X_4^- + H_8 \xrightarrow{q_{max}} \phi$		
$B^+ + L_9 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_9 + B_9$		
$B^- + LS_9 \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_9 + BS_9$	$B^+ + B^- \xrightarrow{\eta} \phi$	
$B^- + H_9 \xrightarrow{q_{max}} \phi$		

Table 4.1: DNA implementation reactions, CRNs and relevant ODEs : (a) The QSM controller is modelled using 38 DNA implementation reactions that are approximated to 14 chemical reactions.

DNA Implementation	CRNs	ODEs
(b) Process to be controlled		
$A^{\pm}+L^{\pm}_{10} \underset{q_{max}}{\overset{q_{10}}{\rightleftharpoons}} H^{\pm}_{10}+B^{\pm}_{10}$		
$X^{\pm}_{5}+H^{\pm}_{10} \xrightarrow{q_{max}} O^{\pm}_{10}+\phi$	$A^{\pm}+X^{\pm}_{5} \xrightarrow{k_{r_1}} X^{+}_{6}$	
$O^{\pm}_{10}+T^{\pm}_{10} \xrightarrow{q_{max}} X^{+}_{6}$		
$A^{\pm}+L^{\pm}_{11} \underset{q_{max}}{\overset{q_{11}}{\rightleftharpoons}} H^{\pm}_{11}+B^{\pm}_{11}$		
$X^{\mp}_{5}+H^{\pm}_{11} \xrightarrow{q_{max}} O^{\pm}_{11}+\phi$	$A^{\pm}+X^{\mp}_{5} \xrightarrow{k_{r_1}} X^{-}_{6}$	$\frac{dX_5}{dt}=-k_{r_1}AX_5+k_{r_2}X_6$
$O^{\pm}_{11}+T^{\pm}_{11} \xrightarrow{q_{max}} X^{-}_{6}$		
$X^{\pm}_{6}+G^{\pm}_{12} \xrightarrow{q_{12}} \phi+O^{\pm}_{12}$	$X^{\pm}_{6} \xrightarrow{k_{r_2}} Y^{\pm}+X^{\pm}_{5}$	$\frac{dX_6}{dt}=k_{r_1}AX_5-k_{r_2}X_6$
$O^{\pm}_{12}+T^{\pm}_{12} \xrightarrow{q_{max}} Y^{\pm}+X^{\pm}_{5}$		
$Y^{\pm}+G^{\pm}_{13} \xrightarrow{q_{13}} \phi$	$Y^{\pm} \xrightarrow{k_{r_3}} \phi$	$\frac{dY}{dt}=k_{r_2}X_6-k_{r_3}Y$
$Y^{+}+L_{14} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{14}+B_{14}$		
$Y^{-}+LS_{14} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{14}+BS_{14}$	$Y^{+}+Y^{-} \xrightarrow{\eta} \phi$	
$Y^{-}+H_{14} \xrightarrow{q_{max}} \phi$		
(c) Subtractor		
$U^{\pm}+G^{\pm}_{15} \xrightarrow{q_{15}} \phi+O^{\pm}_{15}$	$U^{\pm} \xrightarrow{k_s} U^{\pm}+X^{\pm}_{1}$	
$O^{\pm}_{15}+T^{\pm}_{15} \xrightarrow{q_{max}} U^{\pm}+X^{\pm}_{1}$		
$Y^{\pm}+G^{\pm}_{16} \xrightarrow{q_{16}} \phi+O^{\pm}_{16}$	$Y^{\pm} \xrightarrow{k_s} Y^{\pm}+X^{\mp}_{1}$	
$O^{\pm}_{16}+T^{\pm}_{16} \xrightarrow{q_{max}} Y^{\pm}+X^{\mp}_{1}$		
$X^{\pm}_{1}+G^{\pm}_{17} \xrightarrow{q_{17}} \phi$	$X^{\pm}_{1} \xrightarrow{k_s} \phi$	$\frac{dX_1}{dt}=k_s(U-Y-X_1)$
$X^{+}_{1}+L_{18} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{18}+B_{18}$		
$X^{-}_{1}+LS_{18} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{18}+BS_{18}$	$X^{+}_{1}+X^{-}_{1} \xrightarrow{\eta} \phi$	
$X^{-}_{1}+H_{18} \xrightarrow{q_{max}} \phi$		
$X^{+}_{1}(0)=4$ nM	$X^{+}_{1}(0)=8$ nM	$X_1(t)=\begin{cases}4\times10^{-9} & t\in[0,50000]\\ -4\times10^{-9} & t\in[50000,100000]\end{cases}$
$X^{-}_{1}(50000)=8$ nM	$X^{-}_{1}(50000)=16$ nM	
(d) PI Controller		
$X^{\pm}_{1}+G^{\pm}_{19} \xrightarrow{q_{19}} \phi+O^{\pm}_{19}$	$X^{\pm}_{1} \xrightarrow{k_I} X^{\pm}_{1}+X^{\pm}_{2}$	
$O^{\pm}_{19}+T^{\pm}_{19} \xrightarrow{q_{max}} X^{\pm}_{1}+X^{\pm}_{2}$		$\underbrace{\frac{dX_2}{dt}=k_I X_1}_{\text{Integration}}$
$X^{+}_{2}+L_{20} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{20}+B_{20}$		
$X^{-}_{2}+LS_{20} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{20}+BS_{20}$	$X^{+}_{2}+X^{-}_{2} \xrightarrow{\eta} \phi$	
$X^{-}_{2}+H_{20} \xrightarrow{q_{max}} \phi$		
$X^{\pm}_{1}+G^{\pm}_{21} \xrightarrow{q_{21}} \phi+O^{\pm}_{21}$		
$O^{\pm}_{21}+T^{\pm}_{21} \xrightarrow{q_{max}} X^{\pm}_{1}+A^{\pm}$	$X^{\pm}_{1} \xrightarrow{k_P} X^{\pm}_{1}+A^{\pm}$	
$X^{\pm}_{2}+G^{\pm}_{22} \xrightarrow{q_{22}} \phi+O^{\pm}_{22}$		
$O^{\pm}_{22}+T^{\pm}_{22} \xrightarrow{q_{max}} X^{\pm}_{2}+A^{\pm}$	$X^{\pm}_{2} \xrightarrow{k_c} X^{\pm}_{2}+A^{\pm}$	$\underbrace{\frac{dA}{dt}=k_pX_1+k_cX_2-k_dA}_{\text{Gain}}$
$A^{\pm}+G^{\pm}_{23} \xrightarrow{q_{23}} \phi$	$A^{\pm} \xrightarrow{k_d} \phi$	
$A^{+}+L_{24} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} H_{24}+B_{24}$		
$A^{-}+LS_{24} \underset{q_{max}}{\overset{q_{max}}{\rightleftharpoons}} HS_{24}+BS_{24}$	$A^{+}+A^{-} \xrightarrow{\eta} \phi$	
$A^{-}+H_{24} \xrightarrow{q_{max}} \phi$		

TABLE 4.1 (continued): DNA implementation reactions, CRNs and relevant ODEs : (b) The bimolecular process is modelled using 21 DNA implementation reactions that are approximated to 9 chemical reactions. (c) The subtractor module uses 13 DNA implementation reactions that are approximated to 7 chemical reactions. (d) The PI controller is modelled using 20 DNA implementation reactions that are approximated to 10 chemical reactions.

4.3 Performance comparison of the linear and non-linear controllers



(a)

	Parameters	Nominal Values
S_{qsm}	total substrate	4 nM
X_{3Total}	tuning parameter	0.1 nM
k_{b_1}	binding rate	10^7 /M/s
k_{b_2}	binding rate	10^7 /M/s
k_{c_1}	catalytic rate	100 $q_i \cdot C_{max}$ /s
k_{c_2}	catalytic rate	50 $q_i \cdot C_{max}$ /s

(b)

Figure 4.5: QSM controller: (a) Closed loop tracking response obtained using the QSM controller. Here, the reference input U is a square wave of magnitude 4 nM. The transient response can be made faster by reducing the controller tuning parameter X_{3Total}. The subfigure "B" is a zoomed-in version of the subfigure "A" to better illustrate the transient response in the region of interest. (b) Controller parameters and their nominal values.

In this section, the simulation results illustrate the performance comparison of linear PI controller with nonlinear QSM controller in a closed-loop feedback system. For the simulations, all reaction rates and total substrate values have been set to the nominal values given in Table 4.5(b) to 4.7(b). The second order reaction rates are tuned within the practical experimental limits (a maximum value of 10^7 /M/s) [15] and catalysis, degradation, annihilation rates have been chosen in terms of DNA implementation reaction rates, q_i and initial concentration of auxiliary species, C_{max}.

4.3.1 Simulation results with modularity

The system considered in Fig. 4.3 consists of individual modules that are designed to perform particular operations. Initially it is assumed that the closed-loop feed-

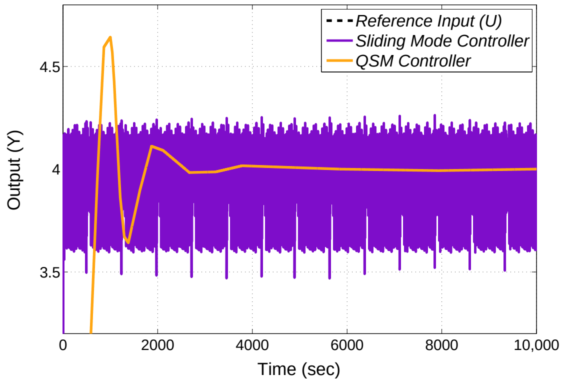


Figure 4.6: Closed loop responses with quasi and ideal sliding mode controllers: the undesirable phenomenon of *chattering*, i.e., high frequency oscillations, is observed in the closed loop response if the ideal SMC controller is used, but is avoided by the QSM controller.

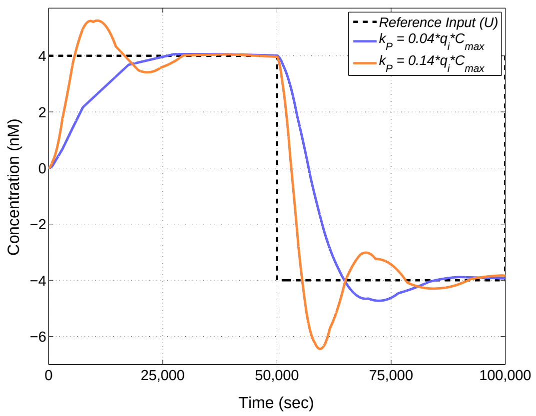
	Parameters	Nominal Values
k_{r_1}	forward binding rate	500×10^3 /M/s
k_{r_2}	catalytic reaction rate	$2 \times 10^3\ q_i\ C_{max}$ /s
k_{r_3}	degradation rate	$10 \times 10^{-3}\ q_i\ C_{max}$ /s
X_{Total}	total amount of $X_5 + X_6$	3 nM

Table 4.2: Process to be controlled — parameter values

back system is modular i.e. the performance of individual modules remain unchanged even after their interconnection to each other. However, in the next section such an effect of interconnection is quantified and simulated.

A square-wave input was chosen for the reference signal U to be tracked by the process output, in line with standard practice in control theory, since such signals generally result in the most challenging possible tracking problem for the control system (the output must track signals that are changing infinitely fast, in both directions). The magnitude of the square wave was chosen to be sufficiently large that it excites the nonlinear dynamics of the process to be controlled. Fig. 4.5(a) shows the closed loop tracking response for the system shown in Fig. 4.3 when the QSM controller is used. The output Y tracks the input U with a settling time of $2,500$ sec if X_{3Total} is set to 0.1 nM .

As shown in Fig. 4.6, the QSM controller also avoids the problem of chatter-



(a)

	Parameters	Nominal Values
k_I	catalytic reaction rate	0.002 q_i C_{max} /s
k_P	catalytic reaction rate	0.04 q_i C_{max} /s
k_c	catalytic reaction rate	0.2 q_i C_{max} /s
k_d	degradation rate	0.4 q_i C_{max} /s

(b)

Figure 4.7: PI controller: (a) Closed loop tracking response obtained using a PI controller. The transient response can be made faster by increasing the value of the controller tuning parameter k_P albeit at the cost of introducing progressively larger overshoots. (b) Controller parameters and their nominal values.

ing that is encountered when the ideal SMC, is used. Fig. 4.7(a) shows the closed loop tracking response for the system shown in Fig. 4.3 when the PI controller is used. The closed loop response dynamics that can be achieved with the PI controller are approximately an order of magnitude slower than those achieved using the QSM controller.

Initial values of the signals A, B, X_2, X_4 are set to zero, i.e. $A_0 = B_0 = X_{2_0} = X_{4_0} = 0$ nM. For the PI controller, the nominal values of the reaction rates and kinetic constants are shown in Table 4.7(b) and the initial concentrations of the non-auxiliary species in equations (4.8)-(4.9) are set to zero, i.e. $X_{2_0} = A_0 = 0$ nM. For the subtractor, k_s is set to its nominal value of $3000 \cdot q_i C_{max}$ /s where DNA implementation reaction rates are given by $q_i = 800$ /M/s ($i = 1, 2, \ldots, 21$), $q_{max} = 10^7$ /M/s [15; 31] and the initial concentration of auxiliary species, $C_{max} = 1000$ nM. The reaction rate of annihilation, η, is set to $10 \cdot q_i C_{max}$ /s.

4.4 Retroactivity

As described in the influential paper by [92], many biomolecular systems may be characterised as being composed of functional ‘modules’ whose interconnection

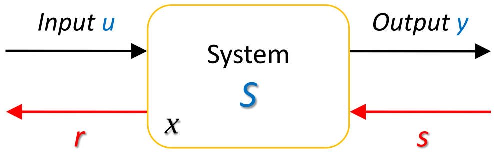


Figure 4.8: Retroactivity: System S having input u and output y. Signals originating from the connection of S to upstream and downstream components are denoted as red signals r and s, respectively.

allows the realisation of higher level functions. For example, the ability of cells to perform higher level operations such as information integration from multiple sources can be achieved by the sequence and pattern of the interconnected functional modules. In Synthetic Biology, individual modules are systematically designed to produce desirable behaviour and then incorporated together to form large, complex biological systems. However, the performance of modules in isolation can differ from its performance once it is connected to other modules. This effect is termed as *retroactivity* [39] which essentially is a biomolecular analogy of loading effects (or impedance effects) [93] in electronics. This effect is specifically studied for genetic networks in [40; 41; 94], and for covalent modification cycle cascades in [55; 95]. Fig. 4.8 illustrates the retroactivity effect for a system S with internal dynamics and input u and output y. Signal s denotes retroactivity from the downstream module on S whereas, signal r denotes retroactivity imposed by S on the upstream module.

As an example, consider a simple example of a water tank with a constant flow of water through an input pipe and an output pressure p_t that can be measured at the output pipe. When the output pipe is connected to another tank, the pressure p_t at the output pipe will be affected due to the pressure applied by the downstream tank. This phenomenon can be modelled as *retroactivity* from downstream to upstream modules connected to a load. The effect of retroactivity depends on the features of interconnection. In the case of the tank, the output of the upstream tank will not be affected by the downstream tank if the connecting pipe aperture is very small compared to the aperture of the output pipe of the upstream tank.

4.4.1 Simulation results with retroactivity

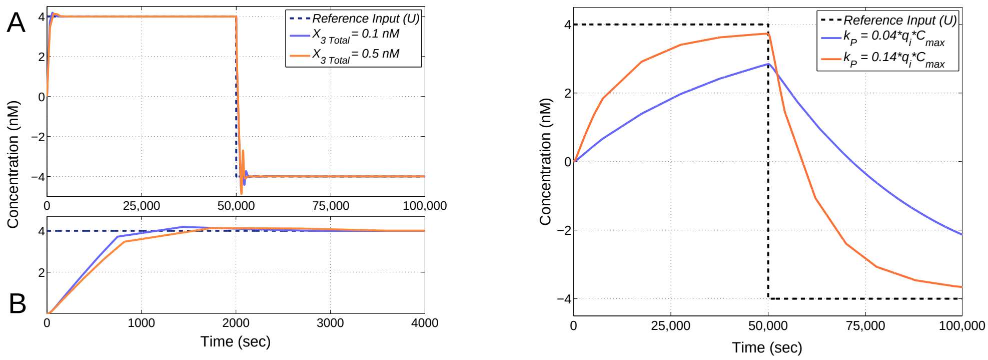
The closed loop responses shown in Figs. 4.5(a) and 4.7(a) assume perfect modularity of the different elements of the feedback system shown in Fig. 4.3, i.e. interconnection of elements does not change their dynamic response. Although this assumption is routinely made in the vast majority of systems traditionally encountered in engineering disciplines, it has recently been established that it does not hold for many biomolecular feedback systems, [40], since it often happens that different modules share the same molecular species. The concept of retroactivity has been introduced to quantify the manner in which the interconnection of two modules changes their dynamics with respect to their behaviour when isolated, [86]. For the system under consideration here, it should be noted that the interconnection of modules containing only unimolecular reactions produces no retroactivity effects. For example, in the context of Fig. 4.3, the interconnection of the subtractor and the PI controller will feature no retroactivity. However, if the system is an interconnection of two modules, one of which comprises unimolecular reactions while the other features bimolecular reactions (e.g. the subtractor and QSM controller) then it will feature a unidirectional retroactivity, since the ODE representation of the subtractor must consider the chemical reactions describing the downstream QSM controller. For the QSM, retroactivity affects the ODEs of two state variables as follows:

$$\frac{dX_1}{dt} = k_s(U - Y - X_1) \underbrace{-k_{b_1}X_1B + k_{c_1}X_2}_{retroactivity}, \tag{4.11}$$

$$\frac{dA}{dt} = k_{c_1}X_2 - k_{b_2}AX_3 \underbrace{-k_{r_1}AX_5}_{retroactivity}. \tag{4.12}$$

The additional term $(-k_{b_1}X_1B + k_{c_1}X_2)$ in equation (4.11) quantifies the retroactivity imposed by the downstream QSM controller on the upstream subtractor through the shared signal X_1, while the additional term $(k_{r_1}AX_5)$ in equation (4.12) quantifies the retroactivity effects between the QSM controller and the process to be controlled through the shared signal A.

As shown in Fig. 4.9(a), the nonlinear QSM controller is highly robust to retroactivity effects, with the major change to the closed loop response being a



(a) Tracking response – QSM controller (b) Tracking response – PI controller

Figure 4.9: (a) Closed loop tracking response obtained by using the QSM controller after accounting for retroactivity effects. (b) Closed loop tracking response obtained using the PI controller after accounting for retroactivity effects.

small reduction in overshoot. In the case of the PI controller, retroactivity affects the ODE of only one state variable, due to the interconnection of the controller and process to be controlled, as follows:

$$\frac{dA}{dt} = k_p X_1 + k_c X_2 - k_d A \underbrace{-k_{r_1} A X_5}_{retroactivity} . \tag{4.13}$$

As shown in Fig. 4.9(b), for the PI controller the presence of retroactivity results in significant changes in the closed loop response, which is now extremely sluggish - for a k_p value of 0.04 q_i C_{max} the controller is not able to track the reference signal even after 50,000 seconds.

4.5 Conclusions

This chapter presents new results on how chemical reactions can be used to design and implement an important class of nonlinear controllers using DSD reactions. These results exploit bimolecular as well as unimolecular reactions to significantly extend the design framework established for linear dynamical systems in [31], allowing the implementation of highly nonlinear synthetic control circuits based on sliding mode control theory. It is shown how a combination of four elementary chemical reactions, viz., bimolecular, catalysis, degradation, and annihilation can be used to realize all necessary functions and how these chemical reactions may be translated into enzyme-free, entropy/enthalpy driven DNA reactions. Simulation results indicate that, compared to a traditional PI controller, the implemented quasi sliding mode controller results are dramatically faster and more accurate in the tracking of reference signals, even in the presence of retroactivity. The proposed design approach is highly modular, fully exploits the inherently nonlinear nature of biomolecular reaction kinetics, and for the first time makes a direct link between the biological concept of ultrasensitivity and the engineering theory of sliding mode control.

Chapter 5

Robustness Analysis of Biomolecular Controllers to Parametric and Time Delay Uncertainties

5.1 Robustness analysis

Robustness is the ability of a system to function properly and be insensitive to the presence of both internal and external uncertainty and disturbances [85; 96–98]. Therefore, the aim is to achieve a robust control design despite such uncertainty, making the system more durable and resilient. The uncertainty may exist in the form of unknown parameters, parameter variations, unpredictable environment, and so on. In control design, feedback systems are widely used to compensate for the disturbance and maintain the desired closed loop operation [98]. Among the popular robust control methods, H_∞ and μ methods are effective robust control tools and extensively applied in the linear control system design. H_∞ loop-shaping is generally used for *multi input multi output* (MIMO) plants having nonlinear dynamics [99; 100] while, μ method is useful for the analysis of the parameter uncertainty and unmodeled dynamics effect on the performance and stability of multiloop feedback systems [101].

On the engineering-biology interface, the concept of robustness has gained significant focus due to the rigorous definition provided in the context of engineering control systems [102]. Biological systems exhibit certain robustness and exploring such mechanism helps understand evolvability in nature as discussed in [103]. [104] addresses the issue of sensitivity of signal transduction networks and proposes a mechanism for robust adaptation.

As shown in Chapter 4, a nonlinear *quasi sliding mode* (QSM) controller can be developed using a set of chemical reactions inspired by the ultrasensitive behaviour exhibited by a *covalent modification cycle* (CMC) cascade [59] and implemented using the *DNA strand displacement* (DSD) mechanism. An important requirement for any embedded bimolecular controller is that its design provides robustness to various forms of uncertainty and variability that could arise in its final implementation in DNA. In this chapter, the focus is on two important sources of such uncertainty - variability in the rate constants of the chemical reactions underlying the closed-loop control system, and uncertain time delays in the biomolecular process to be controlled. In practice, experimental biologists are rarely able to specify the reaction rates of chemical reactions exactly, and additionally, as highlighted in [31], unregulated chemical devices or leaky expressions can potentially affect production and degradation rates and subsequently alter the behaviour of the designed components. The other reasons why one might wish to include time delays in *chemical reaction network* (CRN) models of biomolecular processes, since this avoids cataloging potentially large numbers of intermediate species and their reactions, in favour of describing the dynamic relationships between the concentrations of key species. As a result, fewer concentration variables will generally be required, thus simplifying the overall circuit design problem. Also, in preliminary investigations of a new system, the level of description afforded by a low-order time delayed CRN model is often closer to our state of knowledge than is a detailed model, in which a certain amount of speculation about intermediate species is required, [105].

5.2 System description and methodology

The closed-loop feedback configuration considered to analyse the system robustness is shown in Fig. 5.1. The circuit consists of a number of dynamic components, namely, a subtractor, a controller and a second order nonlinear biomolecular pro-

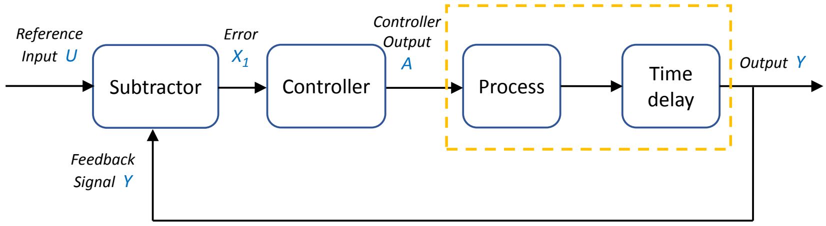


Figure 5.1: The biomolecular closed-loop feedback control system with the accumulative process time delay.

cess with an accumulative time delay. This system is based on DNA elementary reactions and these reactions might exhibit time delays in the form of nucleotide mismatch. DNA being more stable than RNA, it quickly resolves the mismatch and proceeds with the strand displacement. Although this delay is not very large it may be measured in seconds. For this analysis, it is assumed that each component contributes a certain amount of delay which is included in the accumulated form of a single *time delay*. Hence, the term '*accumulative*' time delay.

The controller analysed here is a nonlinear QSM controller and for the purposes of comparison, the level of performance is compared to that achieved using a classical linear *proportional+integrator* (PI) controller (see Chapter 4, Section 4.2.1). The methodology followed here is precisely explained in Chapter 2, Section 2.1.2 where, a signal x is represented as the difference in concentrations of two DNA strands, such that $x = x^+ - x^-$; where, species labelled as x^+ and x^- represent two individual DNA strands. Although some of the operators are mentioned already in the Chapters 3 and 4 they are written here again briefly to maintain the nomenclature and continuity, in the context of the closed-loop feedback system shown in Fig. 5.1.

5.2.1 QSM controller

The chemical reactions describing the QSM controller designed in Chapter 4, are given by:

$$X_1^{\pm} + B^{\pm} \xrightarrow{k_{b_1}} X_2^{+}, \tag{5.1a}$$
$$X_1^{\mp} + B^{\pm} \xrightarrow{k_{b_1}} X_2^{-}, \tag{5.1b}$$
$$X_2^{\pm} \xrightarrow{k_{c_1}} A^{\pm} + X_1^{\pm}, \tag{5.1c}$$
$$X_2^{+} + X_2^{-} \xrightarrow{\eta} \phi, \tag{5.1d}$$
$$A^{+} + A^{-} \xrightarrow{\eta} \phi, \tag{5.1e}$$
$$A^{\pm} + X_3 \xrightarrow{k_{b_2}} X_4^{\pm}, \tag{5.1f}$$
$$X_4^{\pm} \xrightarrow{k_{c_2}} B^{\pm} + X_3, \tag{5.1g}$$
$$X_4^{+} + X_4^{-} \xrightarrow{\eta} \phi, \tag{5.1h}$$
$$B^{+} + B^{-} \xrightarrow{\eta} \phi. \tag{5.1i}$$

where, X_1 is the input and A is the output of the QSM controller. k_{b_1} and k_{b_2} denote the binding reaction rates whereas k_{c_1} and k_{c_2} denote the catalytic reaction rates and η is the annihilation rate. The tuning of the QSM controller involves adjusting k_{b_1}, k_{b_2}, k_{c_1} and k_{c_2}. By tuning the total concentration X_{3Total} where, $X_{3Total} = X_3 + X_4$, the input-output response of the CRN can be made to closely approximate the ideal switch implemented by a *sliding mode controller* (SMC) [35]-[37], so that it implements a QSM controller.

By applying generalised mass action kinetics (see eg. [12]) to (5.1), we get the following set of ODEs:

$$\frac{dA}{dt} = k_{c_1}X_2 - k_{b_2}AX_3, \tag{5.2a}$$
$$\frac{dX_2}{dt} = k_{b_1}X_1B - k_{c_1}X_2, \tag{5.2b}$$
$$\frac{dB}{dt} = -k_{b_1}X_1B + k_{c_2}X_4, \tag{5.2c}$$
$$\frac{dX_4}{dt} = k_{b_2}AX_3 - k_{c_2}X_4. \tag{5.2d}$$

From (5.2), it can be seen that $S_{qsm} \doteq A + B + X_2 + X_4$ is constant. Thus, the signal B is variable and depends on the dynamic signals A, X_2 and X_4. Since, X_1 also varies over time; this means that the term $(k_{b_1}X_1B)$ in (5.2b) is nonlinear.

5.2.2 PI controller

The linear PI controller is constructed following the approach of [31] and [17] using three chemical reactions for the integration operation as:

$$X_1^{\pm} \xrightarrow{k_I} X_1^{\pm} + X_2^{\pm}, \tag{5.3a}$$
$$X_2^{+} + X_2^{-} \xrightarrow{\eta} \phi. \tag{5.3b}$$

and a proportional gain, implemented using seven chemical reactions as:

$$X_1^{\pm} \xrightarrow{k_p} X_1^{\pm} + A^{\pm}, \qquad (5.4a) \qquad\qquad A^{\pm} \xrightarrow{k_d} \phi, \qquad (5.4c)$$

$$X_2^{\pm} \xrightarrow{k_c} X_2^{\pm} + A^{\pm}, \qquad (5.4b) \qquad\qquad A^{+} + A^{-} \xrightarrow{\eta} \phi. \qquad (5.4d)$$

For (5.3) and (5.4), the signal X_1 is the input and A is the output. Furthermore, k_I, k_p and k_c denote the catalytic reaction rates while k_d denotes the degradation rate. Applying mass action kinetics to the above CRNs, the following system of differential equations is obtained for the PI controller:

$$\frac{dX_2}{dt} = k_I X_1, \qquad (5.5a) \qquad\qquad \frac{dA}{dt} = k_p X_1 + k_c X_2 - k_d A. \qquad (5.5b)$$

5.2.3 Process to be controlled

A second order nonlinear process that can be formed using a combination of unimolecular and bimolecular reactions, given as follows:

$$A^{\pm} + X_5^{\pm} \xrightarrow{k_{r1}} X_6^{+}, \qquad (5.6a) \qquad\qquad Y^{\pm}(t+\tau) \xrightarrow{k_{r3}} \phi, \qquad (5.6d)$$

$$A^{\pm} + X_5^{\mp} \xrightarrow{k_{r1}} X_6^{-}, \qquad (5.6b) \qquad\qquad Y^{+} + Y^{-} \xrightarrow{\eta} \phi. \qquad (5.6e)$$

$$X_6^{\pm} \xrightarrow{k_{r2}} Y^{\pm} + X_5^{\pm}, \qquad (5.6c)$$

where, k_{r_1}, k_{r_2}, k_{r_3} are the binding, catalytic and degradation reaction rates, respectively. The input signal to the process module is A and the output is Y. The term τ in (5.6d) indicates the accumulative time delay involved in the production of the output species Y. Applying mass action kinetics to (5.6), we get:

$$\frac{dX_6}{dt} = k_{r_1} A X_5 - k_{r_2} X_6, \quad (5.7a) \qquad\qquad \frac{dY}{dt} = k_{r_2} X_6 - k_{r_3} Y(t-\tau). \qquad (5.7b)$$

where, $X_{Total} \doteq X_5 + X_6$ is constant and conserved through the entire time of the process.

5.2.4 Subtractor

For the closed-loop feedback control, a module is required to compute the difference of the reference signal (U) and output signal (Y). Following [31; 32], the CRN that performs the subtraction operation is given by:

$$U^{\pm} \xrightarrow{k_s} U^{\pm} + X_1^{\pm}, \qquad (5.8a) \qquad\qquad X_1^{\pm} \xrightarrow{k_s} \phi, \qquad (5.8c)$$

$$Y^{\pm} \xrightarrow{k_s} Y^{\pm} + X_1^{\mp}, \qquad (5.8b) \qquad\qquad X_1^{+} + X_1^{-} \xrightarrow{\eta} \phi. \qquad (5.8d)$$

where k_s is the subtraction rate. Here, signals U and Y are the inputs and X_1 is the output of the subtractor. In other words, the value of signal X_1 being produced is equivalent to the difference between the two input signals, U and Y. In addition, both the catalysis reaction rates in (5.8a) and (5.8b) are set to be equal to the degradation rate in (5.8c). Applying mass action kinetics to (5.8) gives:

$$\frac{dX_1}{dt} = k_s(U - Y - X_1). \tag{5.9}$$

In the context of the feedback system shown in Fig. 5.1, the inputs to the subtractor comprise the reference input signal U and the process output Y while its output X_1 is used as the input to the controller.

5.3 Simulation results

The performance of the QSM controller with time delay, $\tau = 0$s and $\tau = 1000$s is shown in Fig. 5.2. In both the cases, the QSM controller is seen to accurately track the reference signal, with nearly the same settling time of approximately 12,000s. However, when the response of the PI controller is evaluated in the presence of $\tau = 1000$s, as shown in Fig. 5.2, large overshoots can be observed.

To analyse the robustness of closed-loop responses achieved with the QSM controller, a Monte Carlo simulation campaign was performed. All the parameters determining the rate constants of the chemical reactions underlying the closed-loop system are randomly drawn from a uniform distribution over repeated simulations. The number of Monte Carlo simulations required to achieve various levels of esti-

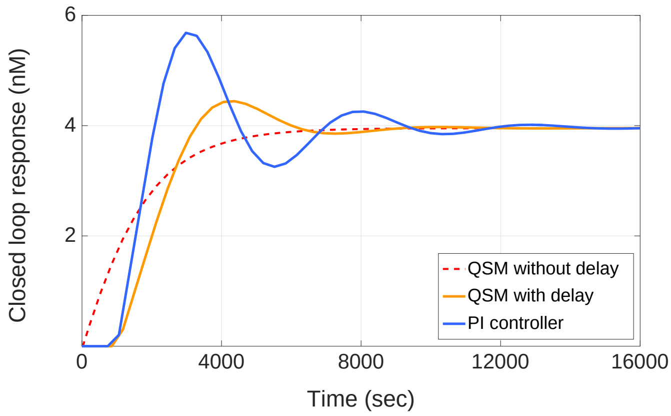


Figure 5.2: Comparing system performance of QSM controller with the PI controller for $\tau = 1000$s. The dashed line shows the response of the QSM controller for $\tau = 0$s.

mation uncertainty with known probability were calculated using the well-known Chernoff bound [106]. An accuracy of 0.05 and a confidence level of 90% were chosen for the Monte Carlo simulation analysis, which requires 1060 simulations, as discussed in [106]-[107]. To investigate the effect of different levels of uncertainty the parameters are varied within ranges of 20% and 50% around their nominal values. Mathematically, it is $p(1+\Delta P(x))$ where, $p \in \{k_s, k_{b_1}, k_{b_2}, k_{c_1}, k_{c_2}, k_I, k_p, k_c, k_d, k_{r_1}, k_{r_2}, k_{r_3}\}$, $P(x)$ is the probability distribution [108] and $\Delta \in \{0.2, 0.5\}$. The ranges 0.2 and 0.5 of Δ is chosen here as it help showing a significant and clear difference in the closed loop performance of both the controllers. This range is neither extreme nor insufficient. For example, if 0.1 and 0.3 is chosen then it could be difficult to show the difference in the response of two controllers. Certainly, one can simulate for any other range than the given one but, 0.2 and 0.5 appears to be adequate in this case.

In the simulations, the given step input U changes from 0 to 4 nM at time $t = 0$s and the role of the controller is to ensure that the process output Y tracks the reference input. As quantitative measures of control system performance, the step response characteristics are measured, which include settling time (t_s), rise time (t_r), percentage overshoot (M_{OS}) and steady state error (e_{ss}). It is desirable to achieve small values of t_s, t_r and M_{OS}, while $e_{ss} = 0$. First, the closed-loop response without parameter uncertainty is simulated, i.e. with nominal parameter values to use it as a

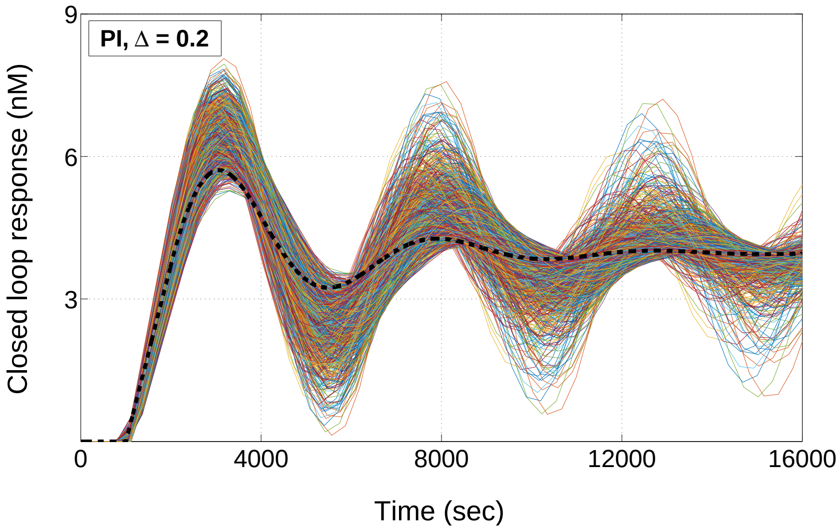


Figure 5.3: System performance with the PI controller for 20% uncertainty in parameters (i.e. $\Delta = 0.2$) and time delay; for the Monte Carlo simulation analysis (no. of simulations = 1060).

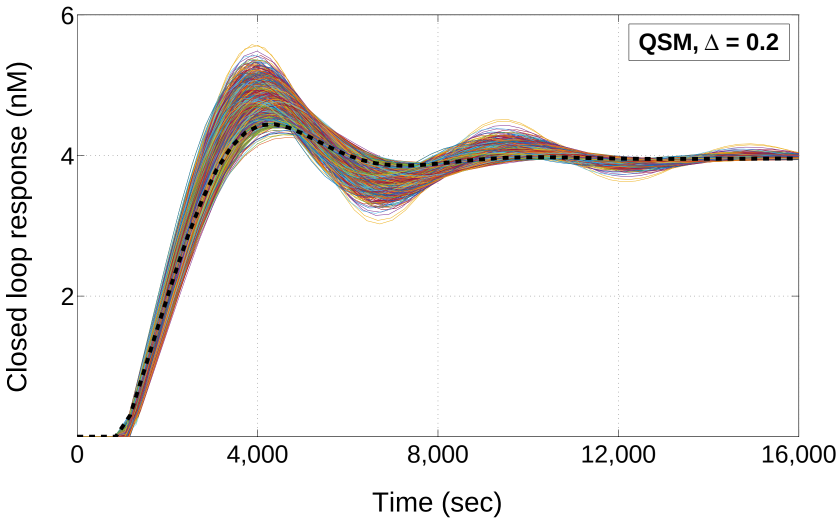


Figure 5.4: System performance with the QSM controller for 20% uncertainty in parameters (i.e. $\Delta = 0.2$) and time delay; for the Monte Carlo simulation analysis (no. of simulations = 1060).

benchmark for comparison. Tables 5.1 and 5.2 detail the results of the Monte Carlo simulation campaign for both the QSM and PI controllers. The PI controller was observed to lose closed-loop stability for $\Delta = 0.5$.

The worst case values of each of the step response characteristics and their

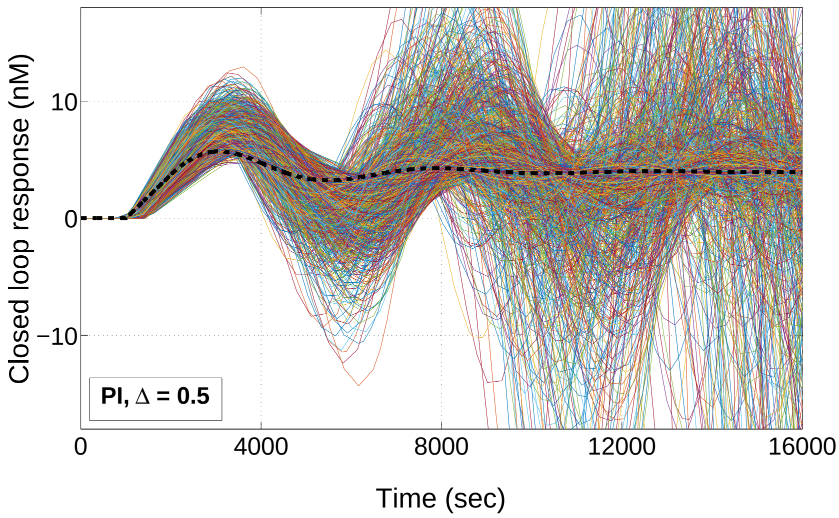


Figure 5.5: System performance with the PI controller for 50% uncertainty in parameters (i.e. $\Delta = 0.5$) and time delay; for the Monte Carlo simulation analysis (no. of simulations = 1060).

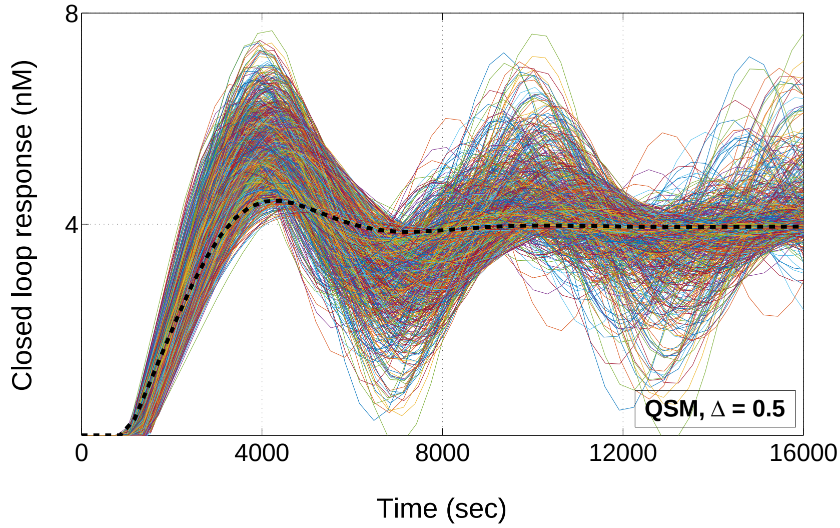


Figure 5.6: System performance with the QSM controller for 50% uncertainty in parameters (i.e. $\Delta = 0.5$) and time delay; for the Monte Carlo simulation analysis (no. of simulations = 1060).

associated parameter values are shown for each of the analysed uncertainty sets (i.e., $\Delta \in \{0.2, 0.5\}$) in Tables 5.1 and 5.2. Ranges are shown for the uncertain parameters since their worst-case values for each step response characteristic are different, e.g. the parameters yielding the worst t_s may not yield the worst t_r, M_{os} and e_{ss} and vice versa. Figs. 5.3 and 5.4 show the step responses produced by the Monte Carlo simulation campaign for each controller when $\Delta = 0.2$ and similarly, Figs. 5.5 and 5.6 show the step responses when $\Delta = 0.5$. As shown, the QSM controller displays significantly greater robustness to the applied levels of uncertainty, highlighting its potential for successful experimental implementation.

5.4 Conclusions

Within the framework of CRNs, this chapter presented an analysis of the performance and robustness properties of a nonlinear QSM controller and a linear PI controller, when subjected to potential accumulative process time delays in the production of the output species of interest. Different levels of variability are introduced in the parameters representing the reaction rates of the underlying chemical reactions, and a process time delay is introduced to investigate the robustness of both controllers to these uncertainties. The simulation results highlight the strong robustness properties of the QSM controller, indicating its suitability for implementation in wet-lab experiments.

Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$
t_s (s)	12,652	15,958	unstable
t_r (s)	718	11,259	unstable
M_{OS} (%)	43.75	283.17	unstable
e_{ss} (M)	0	0	unstable

Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$
Subtractor			
k_s (/s) $[10^{-3}]$	2.4	2.714-2.831	2.863-3.530
PI controller			
k_I (/M/s) $[10^{-6}]$	1.6	1.616-1.907	1.631-2.110
k_p (/M/s)	0.2	0.232-0.233	0.272-0.299
k_c (/s) $[10^{-4}]$	1.6	1.722-1.894	1.934-2.351
k_d (/s) $[10^{-1}]$	3.2	3.255-3.364	3.223-4.497
Nonlinear process			
k_{r_1} (/M/s) $[10^{2}]$	5	5.732-5.926	6.951-7.455
k_{r_2} (/s)	1.6	1.818-1.884	1.804-2.242
k_{r_3} (/s) $[10^{-6}]$	8	8.033-8.696	9.561-11.335
Time delay			
τ (s)	1000	700-1118	695-1436

Table 5.1: Step response characteristics and worst-case parameter ranges for the PI controller.

Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$
t_s (s)	9,654	15,562	15,954
t_r (s)	1,281	1,471	1,631
M_{OS} (%)	12.36	38.23	181.29
e_{ss} (M)	0	0	oscillatory

Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$
Subtractor			
k_s (/s) [10^3]	1	1.139-1.175	1.059-1.267
QSM controller			
k_{b_1} (/M/s) [10^{-3}]	40	41.060-47.587	40.642-59.807
k_{b_2} (/M/s) [10^{-3}]	40	43.094-47.103	42.489-54.347
k_{c_1} (/s) [10^3]	9	9.122-10.732	12.423-13.410
k_{c_2} (/s) [10^3]	10	10.201-11.946	10.269-14.694
Nonlinear process			
k_{r_1} (/M/s) [10^2]	5	5.077-5.900	5.038-7.185
k_{r_2} (/s)	1.6	1.769-1.884	1.651-2.368
k_{r_3} (/s) [10^{-6}]	8	9.040-9.310	9.079-11.390
Time delay			
τ (s)	1000	853-1169	853-1469

Table 5.2: Step response characteristics and worst-case parameter ranges for the QSM controller.

Chapter 6

Exploiting the Dynamic Properties of a Covalent Modification Cycle for Nonlinear Controller Design

Covalent modification cycles may exhibit different operating regimes, as previously discussed in Chapter 2, Section 2.1.3 namely, hyperbolic, signal-transducing, threshold-hyperbolic and ultrasensitive [59]. Interestingly, among all these regimes, the signal-transducing regime resembles the steady-state input-output mapping of the *proportional+integrator* (PI) controller as shown in Fig. 6.1. In industrial control systems, the most commonly used controller is the linear PI controller, and this type of controller has been successfully implemented for biomolecular systems using DNA based chemistry in previous studies [17; 31]. Here, the performance of a *covalent modification cycle* (CMC) controller, designed to operate in its signal-transducing regime, is compared with that of a classical PI controller. The chemical reactions used to respresent CMC and PI controller differ from those given in [31; 59] since, they are modified to incorporate the adapted methodology [31] (refer Chapter 2, Section 2.1.2).

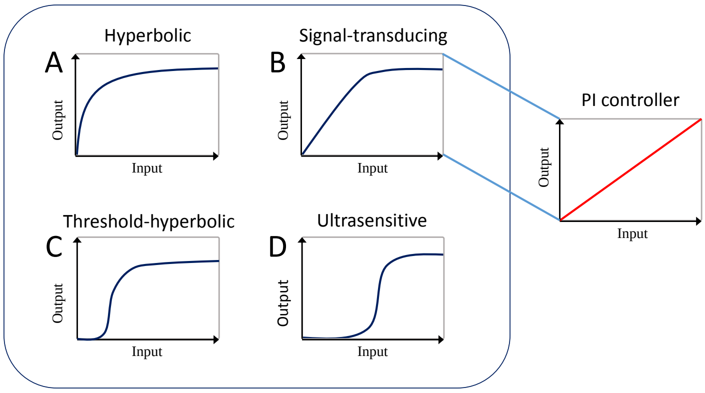


Figure 6.1: Four different mappings of input-output signals in a covalent modification cycle [59]. The signal-transducing mapping resembles the steady-state input-output mapping of a PI controller.

6.1 Designing a covalent modification cycle controller

The closed-loop feedback scheme, with controller (CMC or PI) and process to be controlled (linear or nonlinear), is shown in Fig. 6.2. The chemical reactions underlying the covalent modification cycle (see Chapter 2, Section 2.1.3) can be used to design a nonlinear biomolecular feedback controller - CMC controller.

6.1.1 Chemical reactions

For the closed-loop feedback system in Fig. 6.2, the CRN implementing the subtractor is given by:

$$R^{\pm} \xrightarrow{\gamma_{Sb}} R^{\pm} + E^{\pm}, \quad (6.1a) \qquad E^{\pm} \xrightarrow{\gamma_{Sb}} \emptyset, \quad (6.1c)$$

$$Y^{\pm} \xrightarrow{\gamma_{Sb}} Y^{\pm} + E^{\mp}, \quad (6.1b) \qquad E^{+} + E^{-} \xrightarrow{\eta} \emptyset. \quad (6.1d)$$

where, R and Y are the two inputs- reference signal and feedback signal, respectively and E is the output. The catalysis reaction rate γ_{Sb} is set equivalent to the degradation rate and η is the annihilation reaction rate.

The chemical reactions required to implement the CMC controller are given in (2.11), in Chapter 2, Section 2.1.3. To maintain the consistency of exposition and in the context of the given feedback system in Fig. 6.2, the chemical reactions are given here:

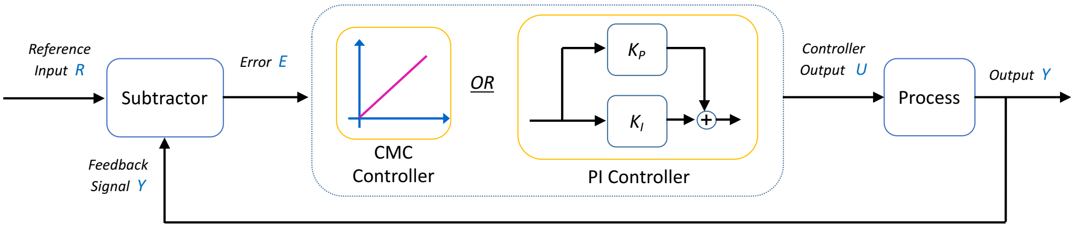


Figure 6.2: A prototype embedded biomolecular closed loop feedback control system with CMC or PI controller.

$$X_p^{\pm} + X_{in}^{\pm} \xrightarrow{k_1} X_{C_1}^{+}, \quad (6.2a)$$
$$X_p^{\pm} + X_{in}^{\mp} \xrightarrow{k_1} X_{C_1}^{-}, \quad (6.2b)$$
$$X_{C_1}^{\pm} \xrightarrow{k_2} X_{out}^{\pm} + X_{in}^{\pm}, \quad (6.2c)$$
$$X_{C_1}^{+} + X_{C_1}^{-} \xrightarrow{\eta} \phi, \quad (6.2d)$$
$$X_{out}^{+} + X_{out}^{-} \xrightarrow{\eta} \phi, \quad (6.2e)$$
$$X_{out}^{\pm} + X_e \xrightarrow{k_3} X_{C_2}^{\pm}, \quad (6.2f)$$
$$X_{C_2}^{\pm} \xrightarrow{k_4} X_p^{\pm} + X_e, \quad (6.2g)$$
$$X_{C_2}^{+} + X_{C_2}^{-} \xrightarrow{\eta} \phi, \quad (6.2h)$$
$$X_p^{+} + X_p^{-} \xrightarrow{\eta} \phi. \quad (6.2i)$$

The chemical reactions in (6.2) are required to implement the CMC controller with $E = X_{in}$ and $U = X_{out}$ (in the context of Fig. 6.2). The values chosen for the CMC controller's reaction rates place it in its signal-transducing input-output mapping regime, which closely resembles the steady-state input-output mapping of a PI Controller.

The classical PI controller considered here is designed according to the methodology of [31] that consists of one integrator, one proportional gain and one summation operator. A total of 15 chemical reactions are required to implement PI controller as follows:

[Integrator]:

$$E^{\pm} \xrightarrow{k_I} E^{\pm} + N^{\pm}, \quad (6.3a) \qquad N^{+} + N^{-} \xrightarrow{\eta} \emptyset. \quad (6.3b)$$

where, k_I is the integral gain of the PI controller and η is the annihilation rate.

[Proportional gain]:

$$E^{\pm} \xrightarrow{\gamma_K k_P} E^{\pm} + M^{\pm}, \quad (6.4a)$$
$$M^{\pm} \xrightarrow{\gamma_K} \emptyset, \quad (6.4b)$$
$$M^{+} + M^{-} \xrightarrow{\eta} \emptyset. \quad (6.4c)$$

where, k_P is the proportional gain of the PI controller, γ_K is the degradation reaction rate and η is the annihilation rate.
[Summation junction]:

$$M^{\pm} \xrightarrow{\gamma_{Sm}} M^{\pm} + U^{\pm}, \quad (6.5a) \qquad U^{\pm} \xrightarrow{\gamma_{Sm}} \emptyset, \quad (6.5c)$$

$$N^{\pm} \xrightarrow{\gamma_{Sm}} N^{\pm} + U^{\pm}, \quad (6.5b) \qquad U^{+} + U^{-} \xrightarrow{\eta} \emptyset. \quad (6.5d)$$

where, γ_{Sm} is the summation reaction rate.

The input to the PI controller is denoted as E, and the output as U. The tuning of this controller involves adjusting k_P, k_I and the reaction rates γ_K and γ_{Sm}. The CMC controller requires 14 reactions to implement, 1 fewer than the PI controller.

Comparative performance of the two controllers is evaluated for two biomolecular processes - a simple first order linear process and a more complex second order nonlinear process. The chemical reactions for both processes are given by:
[Linear process]:

$$U^{\pm} \xrightarrow{k_{p1}} U^{\pm} + Y^{\pm}, \quad (6.6a) \qquad Y^{+} + Y^{-} \xrightarrow{\eta} \emptyset. \quad (6.6c)$$

$$Y^{\pm} \xrightarrow{k_{p2}} \emptyset, \quad (6.6b)$$

where, k_{p1} and k_{p2} are the catalysis and degradation rates of the process.
[Nonlinear process]:

$$U^{\pm} + P^{\pm} \xrightarrow{k_{r1}} Q^{+}, \quad (6.7a) \qquad Y^{\pm} \xrightarrow{k_{r3}} \emptyset, \quad (6.7d)$$

$$U^{\pm} + P^{\mp} \xrightarrow{k_{r1}} Q^{-}, \quad (6.7b) \qquad Y^{+} + Y^{-} \xrightarrow{\eta} \emptyset. \quad (6.7e)$$

$$Q^{\pm} \xrightarrow{k_{r2}} Y^{\pm} + P^{\pm}, \quad (6.7c)$$

where, P and Q are intermediate species involved in the second order process reaction. k_{r1}, k_{r2} and k_{r3} are respectively the binding, catalytic and degradation rates of the process and η is the annihilation rate.

6.1.2 System of ordinary differential equations

Using generalised mass action kinetics, the ODEs corresponding to the CRNs for each component in Fig 6.2 from (6.1) to (6.7) are given as:

[Subtraction operator]:

$$\frac{dE}{dt} = \gamma_{Sb}(R - Y - E) \tag{6.8}$$

[CMC controller]:

$$\frac{dU}{dt} = k_2 X_{C1} - k_3 U X_e, \tag{6.9a}$$

$$\frac{dX_{C1}}{dt} = k_1 E - k_2 X_{C1}, \tag{6.9b}$$

$$\frac{dX_{C2}}{dt} = k_3 U X_e - k_4 X_{C2}. \tag{6.9c}$$

[PI controller]:

$$\frac{dN}{dt} = k_I E, \tag{6.10a}$$

$$\frac{dM}{dt} = \gamma_K (k_P E - M), \tag{6.10b}$$

$$\frac{dU}{dt} = \gamma_{Sm}(M + N - U). \tag{6.10c}$$

[Linear process]:

$$\frac{dY}{dt} = k_{p1} U - k_{p2} Y \tag{6.11}$$

[Nonlinear process]:

$$\frac{dQ}{dt} = k_{r1} U P - k_{r2} Q, \tag{6.12a}$$

$$\frac{dY}{dt} = k_{r2} Q - k_{r3} Y. \tag{6.12b}$$

In [31], the gain and summation operators used in the PI controller require identical reaction rates for multiple reactions (for eg. two catalysis reactions having same reaction rate). However, implementing this requirement in an experimental setting is unlikely to be feasible, as experimental biologists are rarely able to specify the exact reaction rates of chemical reactions. Additionally, in practice, as highlighted in [31], unregulated chemical devices or leaky expressions could potentially affect production and degradation rates and subsequently alter the behaviour of the designed component. To investigate these issues, a robustness analysis of both controllers is performed, focussing on the effect of uncertainties in the implemented reaction rates on the closed-loop stability and performance properties of the feedback system.

6.2 Simulation results

To analyse the performance and robustness of the closed-loop responses achieved by the feedback controllers with the linear process, step response tests and Monte Carlo simulations are performed, respectively. For the Monte Carlo simulations, all the parameters are randomly drawn from a uniform distribution. The number of Monte Carlo simulations required to achieve various levels of estimation uncertainty with known probability are calculated using the well-known Chernoff bound [106]. Following the guidelines provided in [109], an accuracy level of 0.05 and a confidence level of 99% are chosen for the Monte Carlo simulation analysis, which requires a total number of 1060 simulations [106; 107]. To investigate the effect of different levels of uncertainty the parameters are varied within ranges of 20%, 50%, 100% and 120% around their nominal values. Mathematically, it is $p(1+\Delta P(x))$, where $p \in \{\gamma_{Sb1}, \gamma_{Sb2}, \gamma_{Sb3}, \gamma_{K1}, \gamma_{K2}, \gamma_{Sm1}, \gamma_{Sm2}, \gamma_{Sm3}, k_I, k_P, k_1, k_2, k_3, k_4, k_{p1}, k_{p2}\}$, $P(x)$ is the probability distribution and $\Delta \in \{0.2, 0.5, 1.0, 1.2\}$.

In the simulations, a step change in the concentration of the reference species, R from 0 M to 1 M occurs at time, $t = 0$s and the purpose of the controller is to ensure that the process output reaches this new desired concentration. As quantitative measures of the control system performance, the step response characteristics are used, which comprise the rise time, t_r, settling time, t_s, percentage of overshoot, M_{OV} and steady-state error, e_{ss} [90].

6.2.1 Performance analysis of controllers with a linear process

For good closed-loop performance, it is desirable to achieve a small t_r, t_s and M_{OV} as well as having $e_{ss} = 0$. As a benchmark for comparison, first the step response characteristics without parameter uncertainty are calculated. Hereafter, those are referred as the set of results for the *nominal system*. The parameters for the nominal system in the required chemical reactions are:
[Linear process]: k_{p1} = 0.1 /s, k_{p2} = 0.1 /s.
[Subtractor dynamics]: γ_{Sb1}, γ_{Sb2}, γ_{Sb3} = 0.4 /s.
[CMC controller]: k_1, k_3 = 0.00185 /M/s, k_2, k_4 = 0.5 /s, $X_p + U + X_{C1} + X_{C2}$ = 27.5 M and $X_e + X_{C2}$ = 0.033 M.

[PI controller]: γ_{Sb1}, γ_{Sb2}, γ_{Sb3} = 0.4 /s, γ_{Sm1}, γ_{Sm2}, γ_{Sm3} = 0.8 /s, γ_{K1}, γ_{K2} = 0.0004 /s, k_P = 1 /s and k_I = 0.045 /s.

The step response characteristics for both of the nominal systems are tabulated in Table 6.1. For each of the analysed uncertainty sets, the worst-case values returned by Monte Carlo simulation for each of the step response characteristics and its associated parameter set are shown. Note that a range of parameters is given here as the parameter set associated with each worst-case characteristic is different. For example, the parameters yielding the worst t_r may not yield the worst t_s, M_{OV} and e_{ss} and vice versa. For illustration, the step responses depicting the nominal and worst-case responses for each step response characteristics for $\Delta \in \{0.2, 0.5, 1.0, 1.2\}$ are shown in Fig. 6.3 for both PI and CMC controllers.

The performance of the two nominal closed-loop systems is rather similar, which reflects the fact that the CMC controller is designed to reproduce the steady-state input-output mapping of the original PI controller. Interestingly, however, the robustness of the system can be clearly seen to be significantly improved when the CMC controller is used. With the PI controller, the closed-loop system become unstable when Δ = 1.2, while for the CMC controller, the closed-loop system becomes unstable only when Δ = 1.8 (not provided in the simulation results), showing that the CMC controller is able to tolerate more than a 50% larger variability in the values of the reaction rates in the underlying chemical reactions.

PI controller					
Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$	$\Delta = 1.2$
t_r (s)	29	44	75	157	173
t_s (s)	96	113	175	499	Unstable
M_{OV} (%)	9.14	22.83	52.25	114.17	Unstable
e_{ss} (M)	0.00	0.19	0.49	0.91	Unstable
Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$	$\Delta = 1.2$
γ_{Sb1} (/s)	0.400	0.431-0.476	0.532-0.595	0.475-0.791	0.584-0.599
γ_{Sb2} (/s)	0.400	0.401-0.471	0.400-0.584	0.413-0.569	0.428-0.875
γ_{Sb3} (/s)	0.400	0.401-0.473	0.414-0.593	0.466-0.721	0.412-0.853
k_I	0.045	0.048-0.054	0.048-0.061	0.052-0.086	0.058-0.085
k_P	1.000	1.016-1.165	1.130-1.359	1.137-1.549	1.159-1.367
γ_{K1} (/s) $[10^{-3}]$	0.400	0.436-0.477	0.424-0.515	0.434-0.674	0.505-0.795
γ_{K2} (/s) $[10^{-3}]$	0.400	0.403-0.466	0.401-0.454	0.473-0.666	0.570-0.683
γ_{Sm1} (/s)	0.800	0.809-0.948	0.863-1.099	0.827-1.544	0.825-1.410
γ_{Sm2} (/s)	0.800	0.835-0.943	0.904-1.012	0.849-1.205	1.152-1.548
γ_{Sm3} (/s)	0.800	0.832-0.958	0.823-1.140	0.841-1.536	0.853-1.279
k_1 (/s)	0.100	0.101-0.116	0.106-0.142	0.111-0.174	0.127-0.208
k_2 (/s)	0.100	0.101-0.114	0.102-0.139	0.103-0.199	0.123-0.211

CMC controller					
Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$	$\Delta = 1.2$
t_r (s)	29	37	65	89	117
t_s (s)	97	116	155	202	353
M_{OV} (%)	10.12	25.3	44.63	60.55	75.00
e_{ss} (M)	0.00	0.18	0.46	0.92	1.12
Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$	$\Delta = 1.2$
γ_{Sb1} (/s)	0.400	0.403-0.476	0.450-0.595	0.412-0.781	0.626-0.857
γ_{Sb2} (/s)	0.400	0.401-0.477	0.406-0.580	0.407-0.752	0.403-0.745
γ_{Sb3} (/s)	0.400	0.403-0.466	0.402-0.594	0.577-0.726	0.425-0.806
k_{b1} (/M/s) $[10^{-2}]$	0.185	0.186-0.221	0.202-0.274	0.199-0.342	0.224-0.365
k_{b2} (/s)	0.500	0.514-0.565	0.516-0.683	0.621-0.737	0.509-0.747
k_{b3} (/M/s) $[10^{-2}]$	0.185	0.187-0.213	0.187-0.245	0.242-0.354	0.230-0.318
k_{b4} (/s)	0.500	0.516-0.596	0.679-0.723	0.553-0.851	0.662-1.008
k_1 (/s)	0.100	0.100-0.119	0.104-0.148	0.130-0.199	0.101-0.174
k_2 (/s)	0.100	0.101-0.114	0.105-0.150	0.106-0.198	0.109-0.209

Table 6.1: Step response characteristics and worst-case parameter ranges for the PI and CMC controllers + the linear process.

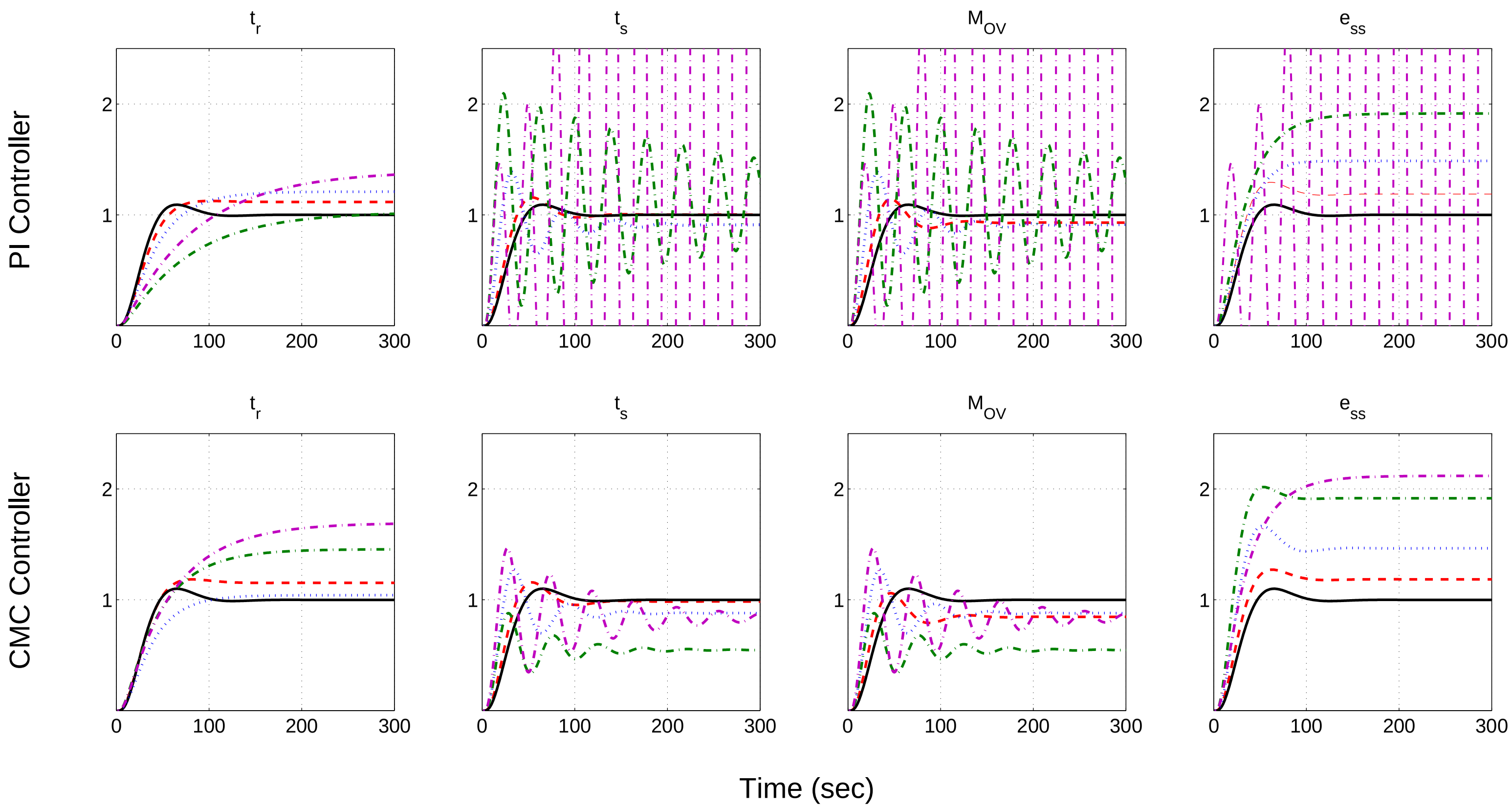


Figure 6.3: Top row: PI controller + linear process. Bottom row: CMC controller + linear process. Nominal and worst cases of t_r, t_s, M_{OV} and e_{ss} with $\Delta = 0.2, 0.5, 1.0, 1.2$. Black solid line: Nominal system. Red dashed line: worst-case response for $\Delta = 0.2$, Blue dotted line: worst-case response for $\Delta = 0.5$, green dash-dotted line: worst-case response for $\Delta = 1.0$, magenta dash-dotted line: worst-case response for $\Delta = 1.2$.

6.2.2 Performance analysis of the controllers with a nonlinear process

In this section, it shown how the two controllers perform when controlling a more complex second-order nonlinear process. The same step test and Monte Carlo simulations are carried out, with the parameters for the nominal system in the required chemical reactions given as:

[Nonlinear process]: k_{r1} = 0.00005 /M/s, k_{r2} = 1.6 /s, k_{r3} = 0.0008 /s, with the total concentration constrained so that, $P+Q = 5.5$ M.

[Subtractor dynamics]: γ_{Sb1}, γ_{Sb2}, γ_{Sb3} = 0.4 /s.

[CMC controller]: k_1 = = 0.0000055 /M/s, k_3 = 0.000018 /M/s, k_2 = 12.50 /s, k_4 = 140 /s, $X_p + U + X_{C1} + X_{C2} = 66$ M and $X_e + X_{C2} = 0.00012$ M.

[PI controller]: γ_{Sb1}, γ_{Sb2}, γ_{Sb3}, γ_{Sm1}, γ_{Sm2}, γ_{Sm3}, γ_{K1}, γ_{K2} = 0.0004 /s, k_P = 0.65 /s and k_I = 0.3 /s.

The parameter values given here for both the controllers with a nonlinear process are different than those for the linear process as the controllers are tuned to control the different processes. The step response characteristics for both of the nominal systems are tabulated in Table 6.2. As previously, the step responses depicting the nominal and worst-case responses for $\Delta \in \{0.2, 0.5, 1.0\}$ are shown in Fig. 6.4 for the PI and CMC controllers respectively. Note that the case for $\Delta = 1.2$ is not considered as the closed-loop system becomes unstable for $\Delta = 1.0$, when the PI controller + nonlinear process is used.

The performance of the two nominal closed-loop systems are rather similar, which again reflects the fact that the CMC controller is designed to reproduce the steady-state input-output mapping of the original PI controller. The closed-loop system with the CMC controller retains closed-loop stability up until $\Delta = 1.6$, again demonstrating a significantly higher level of robustness than exhibited by the linear PI controller.

PI controller				
Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$
t_r (s)	11,139	18,959	31,673	Unstable
t_s (s)	26,304	44,138	48,482	Unstable
M_{OV} (%)	2.42	24.88	44.44	Unstable
e_{ss} (M)	0.00	0.19	0.48	Unstable
Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$
γ_{Sb1} (/s) $[10^{-3}]$	0.400	0.445-0.480	0.492-0.596	0.472-0.730
γ_{Sb2} (/s) $[10^{-3}]$	0.400	0.402-0.475	0.400-0.453	0.533-0.735
γ_{Sb3} (/s) $[10^{-3}]$	0.400	0.427-0.467	0.433-0.591	0.402-0.473
k_I $[10^{-3}]$	0.300	0.301-0.353	0.310-0.420	0.486-0.504
k_P	0.650	0.673-0.771	0.790-0.908	0.806-1.282
γ_{K1} (/s) $[10^{-3}]$	0.400	0.432-0.461	0.446-0.573	0.498-0.747
γ_{K2} (/s) $[10^{-3}]$	0.400	0.401-0.422	0.446-0.586	0.404-0.740
γ_{Sm1} (/s) $[10^{-3}]$	0.400	0.427-0.479	0.424-0.587	0.641-0.791
γ_{Sm2} (/s) $[10^{-3}]$	0.400	0.423-0.462	0.469-0.553	0.615-0.730
γ_{Sm3} (/s) $[10^{-3}]$	0.400	0.409-0.478	0.418-0.539	0.401-0.431
k_{r1} (/M/s) $[10^{-4}]$	0.500	0.509-0.594	0.536-0.734	0.737-0.945
k_{r2} (/s)	1.600	1.633-1.865	1.749-2.232	1.860-2.843
k_{r2} (/s) $[10^{-3}]$	0.800	0.812-0.904	0.819-1.102	0.844-0.891

CMC controller				
Characteristics	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$
t_r (s)	11,147	15,501	25,753	30,838
t_s (s)	28,848	28,324	42,494	49,196
M_{OV} (%)	2.84	13.12	26.25	56.37
e_{ss} (M)	0.00	0.19	0.46	0.98
Parameters	Nominal	$\Delta = 0.2$	$\Delta = 0.5$	$\Delta = 1.0$
γ_{Sb1} (/s) $[10^{-3}]$	0.400	0.426-0.479	0.534-0.597	0.542-0.798
γ_{Sb2} (/s) $[10^{-3}]$	0.400	0.400-0.478	0.404-0.511	0.403-0.798
γ_{Sb3} (/s) $[10^{-3}]$	0.400	0.406-0.457	0.425-0.578	0.403-0.633
k_{b1} (/M/s) $[10^{-5}]$	0.550	0.567-0.644	0.577-0.808	0.619-1.075
k_{b2} (/s)	12.50	12.64-14.75	16.57-17.84	16.99-17.95
k_{b3} (/M/s) $[10^{-4}]$	0.180	0.182-0.207	0.203-0.238	0.196-0.274
k_{b4} (/s)	140.00	144.48-163.09	165.38-205.25	143.15-278.30
k_{r1} (/M/s) $[10^{-4}]$	0.500	0.503-0.593	0.523-0.712	0.538-0.807
k_{r2} (/s)	1.600	1.635-1.893	1.839-1.950	1.789-2.861
k_{r2} (/s) $[10^{-3}]$	0.800	0.803-0.943	0.804-1.162	0.808-1.525

Table 6.2: Step response characteristics and worst-case parameter ranges for PI and CMC controllers + nonlinear process.

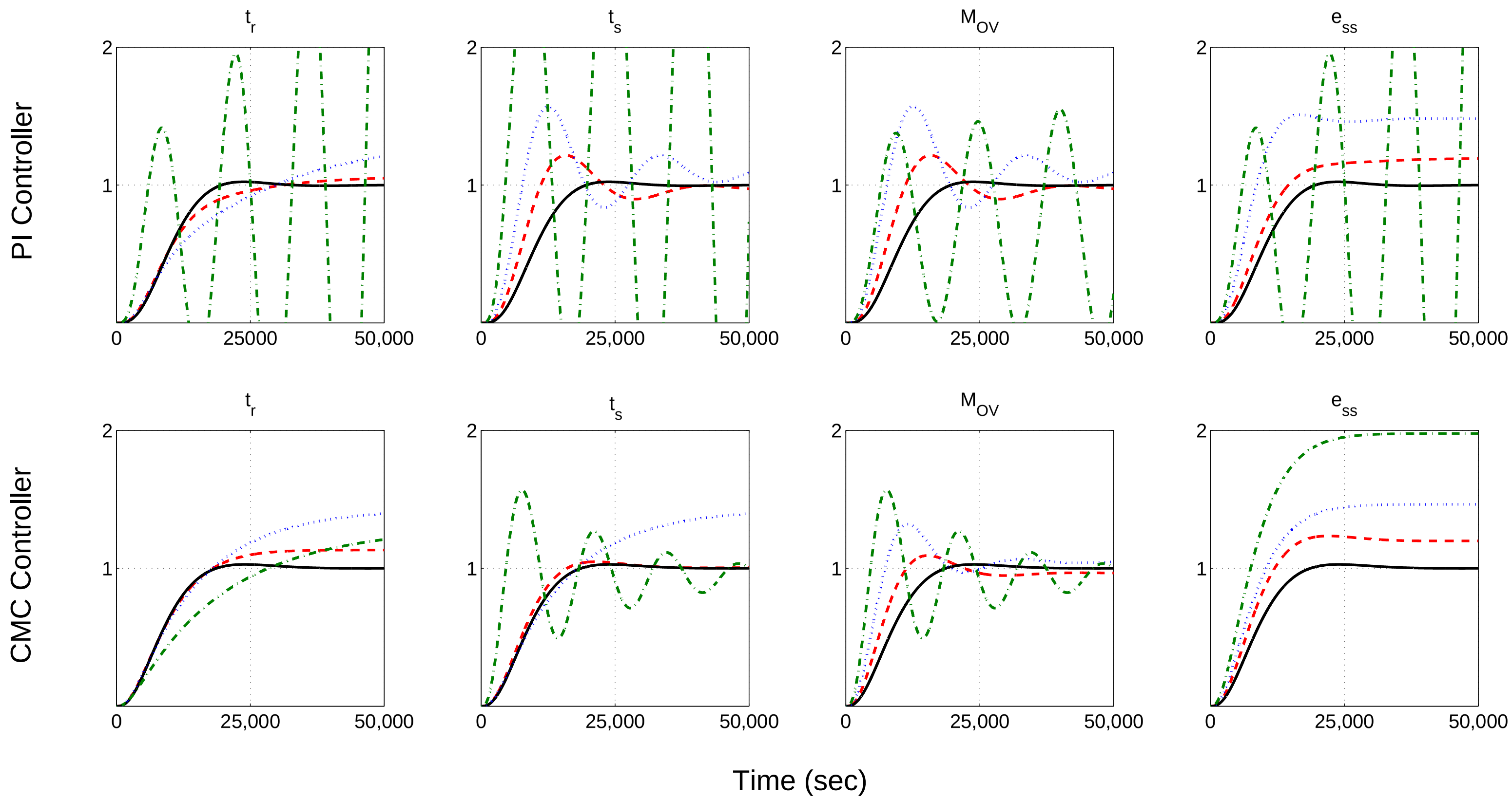


Figure 6.4: Top row: PI controller + nonlinear process. Bottom row: CMC controller + nonlinear process. Nominal and worst cases of t_r, t_s, M_{OV} and e_{ss} with $\Delta = 0.2$, 0.5 and 1.0. Black solid line: Nominal system. Red dashed line: worst-case response for $\Delta = 0.2$, Blue dotted line: worst-case response for $\Delta = 0.5$, green dash-dotted line: worst-case response for $\Delta = 1.0$.

6.3 Flexible input-output mapping improves robustness

The results thus far have shown consistently better robustness from the CMC controller compared to the PI controller. To explain this, the mapping of steady-state input-output signals of these two controllers is analysed. Fig. 6.5(A) shows the mapping of steady-state input-output signals of both controllers as they were implemented when controlling the linear process. The mapping of input-output signals for the nominal system and the maximum deviation from this response when $\Delta = 1.2$ are shown in black solid line and magenta dash-dotted line respectively. A significantly greater change to the gradient of the PI controller's input-output mappings can be observed compared to the CMC controller.

This intriguing observation leads to the question - how is the gradient of this mapping of steady-state input-output signals related to the robustness of the controller? Given that the process to be controlled is a linear process, its ODE representation (with $X := Y$) is given by:

$$\frac{dX}{dt} = -k_{p2}X + k_{p1}U \tag{6.13}$$

Here, (6.13) is in the standard state-space representation (i.e. $\frac{dx}{dt} = Ax + Bu,\ y = Cx + Du$) with $A = -k_{p2}$ and $B = k_{p1}$, $C = 1$ and $D = 0$. In linear control theory design using a state-space approach, [110], a standard control law can be written as $U = KX$ where, K is the controller gain. This linear control law can be viewed as a mapping of the input, X to the output, U with K being the gradient. Substituting $U = KX$ into (6.13), we have:

$$\frac{dX}{dt} = -k_{p2}X + k_{p1}KX = (k_{p1}K - k_{p2})X \tag{6.14}$$

As (6.14) is in scalar form, the overall process is stable if the real part of the eigenvalue of A (i.e. $k_{p1}K - k_{p2}$) is less than 0, Hence, the following condition, $K < \frac{k_{p2}}{k_{p1}}$ must hold. In other words, if the controller gain, K is less than the ratio of the process parameters k_{p2} to k_{p1}, one has a stable system. In the simulation, the process parameters of the nominal system are $k_{p1}, k_{p2} = 0.1$, thus, for the system

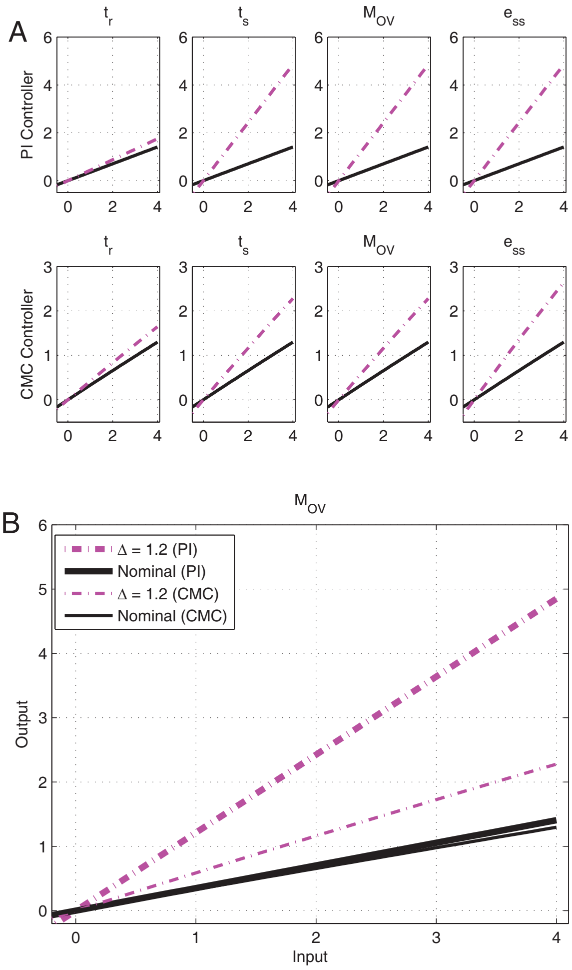


Figure 6.5: (A) The mapping of steady-state input-output signals of the PI controller (top row) and the CMC controller (bottom row) when controlling the linear process. Black solid line: Nominal system. Magenta dash-dotted line: worst-case response for $\Delta = 1.2$. (B) The zoomed-in version of M_{OV} from (A). Black solid line: nominal system. Magenta dash-dotted line: worst-case response for $\Delta = 1.2$.

to be stable, it is required that $K < \frac{k_{p2}}{k_{p1}} = 1$. Looking at Fig. 6.5(B), a zoomed-in version using M_{OV} as an illustration, the gradient of both the controllers' input-output mapping are less than 1 (i.e. ≈ 0.34); the closed-loop system is stable. Note that for the nominal system, both the controllers' input-output mappings are very similar, as expected, since the CMC controller was designed to reproduce the PI controller's steady-state input-output mapping.

Now, the effect of increasing the levels of variability in the values of the parameters in the chemical reactions implementing the feedback control system is considered. For the PI controller, at Δ = 1.2, the process parameters change from $k_{p1} = 0.100 \rightarrow 0.208$ and $k_{p2} = 0.100 \rightarrow 0.124$. Thus, the ratio $\frac{k_{p2}}{k_{p1}}$ changes from $1 \rightarrow 0.596$. Likewise, from Fig. 6.5(B), it can be observed that the gradient of the PI controller's steady-state input-output mapping changes to $1.213 > \frac{k_{p2}}{k_{p1}} = 0.596$, which accounts for the observed unstable behaviour.

On the other hand, the change of gradient for the CMC controller is smaller compared to the PI controller. At $\Delta = 1.2$, the process parameters change from $k_{p1} = 0.1 \rightarrow 0.174$ and $k_{p2} = 0.1 \rightarrow 0.109$, leading the ratio $\frac{k_{p2}}{k_{p1}}$ to change from 1 $\rightarrow 0.628$. However, the gradient of the CMC controller's steady-state input-output mapping changes to $0.588 < \frac{k_{p2}}{k_{p1}} = 0.628$, thus preserving the stability of the system.

What makes the CMC more robust (in terms of gradient change) to parameter uncertainty? The simulation results using the nonlinear process shed some light on this matter. The steady-state mapping of input-output signals simulated 1060 times at $\Delta = 1.0$ for both the controllers when controlling the nonlinear process are shown in Fig. 6.6(A). For the nominal system both the controllers' input-output mapping retains a linear behaviour. While the PI controller's steady-state input-output mapping stays linear for all 1060 uncertainty combinations, the CMC controller's input-output mapping displays a 'hyperbolic' behaviour for some parameter combinations. Recall that this 'hyperbolic' behaviour is one of the input-output signal mappings reported in [59] (see also Fig. 6.1). Thus, the simulation results here seem to indicate that parameter uncertainty has the capacity to change the operating regime of the CMC controller from signal-transducing to hyperbolic. Thus, the question of interest is whether this change in the mapping regime accounts for the better robustness of the CMC controller.

As the process is now nonlinear, the notion of eigenvalue no longer applies while the notion of stability for a nonlinear system is also more mathematically involved and beyond the scope of the analysis in this chapter. However, the difference in the robustness of both controllers can be informally explained by extending the arguments on the 'gradient' of the steady-state input-output mapping, as was done for the linear process. Fig. 6.6(B) shows the nominal and worst-case deviation in the input-output mapping for both controllers at $\Delta = 1.0$.

From Fig. 6.6(C), it can be seen that despite both controllers having very similar mapping of input-output signals for the nominal system, when subjected to parameter uncertainty, the gradient of the PI controller's steady-state input-output mapping becomes steeper and subsequently affects the stability of the system. On the other hand, not only does the CMC controller's input-output mapping show a smaller change in response to uncertainty, it becomes more hyperbolic. The CMC controller's innate ability to achieve hyperbolic behaviour seems to be able to prevent the adverse effect of parameter uncertainty, as it enables the gradient of its input-output mapping when subjected to parameter uncertainty to remain small.

6.4 Conclusions

In this chapter, it is shown how the set of chemical reactions underlying the covalent modification cycle motif may be used to design and implement a nonlinear feedback controller whose steady-state input-output behaviour mimics the well known PI controller. The resulting CMC controller is shown to be significantly more robust to variability in circuit parameters that will inevitably arise in experimental implementations of synthetic circuitry. Given the range of input-output mappings that can be produced by the set of chemical reactions underlying the covalent modification cycle, it is likely that they could be used to efficiently design many other types of operators and controllers. As the chemical reactions concerned are all represented either in unimolecular or bimolecular form, the resulting circuits can then be readily implemented using DNA-based chemistry either *in vitro* or *in vivo* for future Synthetic Biology applications.

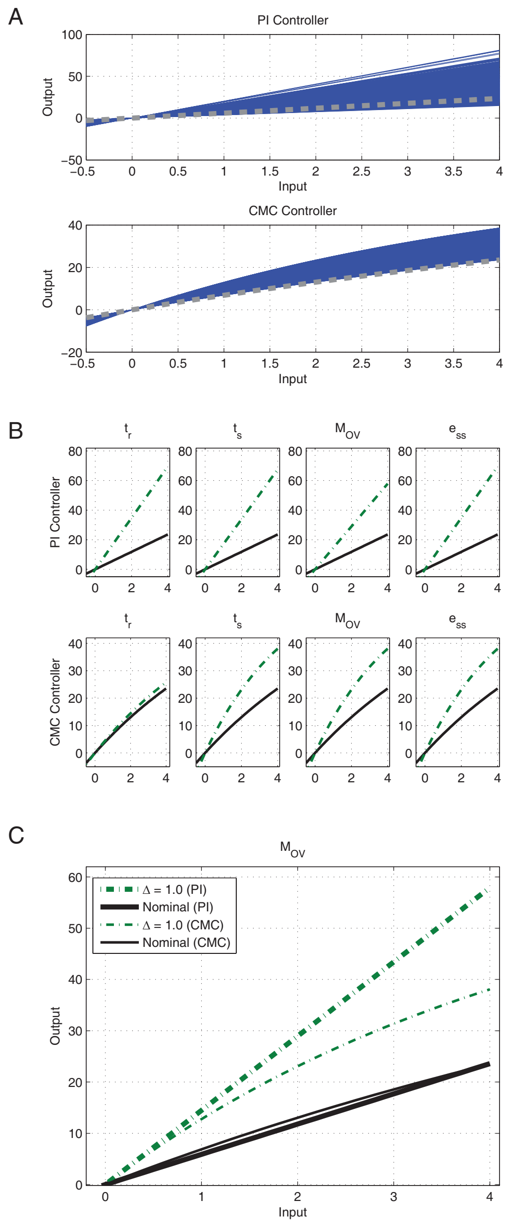


Figure 6.6: (A) The mappings of steady-state input-output signals of the PI controller and CMC controller simulated 1060 times for $\Delta = 1.0$. Nominal systems are shown in the thick dotted grey line. (B) The mapping of the steady-state input-output signals of the PI controller (top row) and the CMC controller (bottom row) when controlling the nonlinear process. Black solid line: Nominal system. Green dash-dotted line: worst-case response for $\Delta = 1.0$. (C) The zoomed-in version of the mapping of the steady-state input-output signals of the PI controller (bold line) and the CMC controller (thin line) for M_{OV} from (B). Black solid line: Nominal system. Green dash-dotted line: worst-case response for $\Delta = 1.0$.

Chapter 7

Conclusions and Future Work

A comprehensive approach for programming dynamic nonlinear devices within the framework of chemical reactions that are implementable via the *DNA strand displacement* (DSD) mechanism for enzyme-free, entropy/enthalpy driven DNA reactions, is presented in this thesis. An essential step in this approach is the mapping of *chemical reaction networks* (CRNs) into a system of *ordinary differential equations* (ODEs), by means of generalised mass action laws, allowing the formulation of mathematical models of the biological system under consideration. The dynamic behaviour of such mathematical models, i.e. evolution of chemical species over a finite time, can then be illustrated, observed and accordingly modified in software simulations (eg.: MATLAB, Simulink).

In order to design a biomolecular component or a system, there may exist more than one set of chemical reactions or more than one set of parameter values that can achieve the similar desired system response. For example, combinations of different types of chemical reactions can result in similar or maybe different ODEs that eventually perform identically. Also, the response generated by one set of parameter values can be equivalently produced by another set of parameter values. Considering these possibilities, one should select a suitable set of chemical reactions having a minimum number of reactions which is preferred or rather feasible for *in vivo/in vitro* implementations. Also, a suitable set of parameter values can be chosen for simulations, keeping in mind that these values are well within the practically achievable physical limit, for eg., DNA or mRNA concentration levels, reaction rates etc.

A precise overview on the background theory of CRNs, the computational methodology and DSD mechanism is given in Chapter 2. The underlying DSD mechanism of four chemical reactions, viz., catalysis, bimolecular, degradation, and annihilation, that are considered in this thesis, is illustrated using the software tool Visual DSD. Chapter 3 shows results on how chemical reactions can be used to design and implement a number of nonlinear system theoretic operators, thus significantly extending the results obtained in [31] that considered only linear systems. It is also shown how polynomial functions, rational functions and power components can be implemented by using a combination of the aforementioned four types of chemical reactions. Based on this, the new results are highlighted through three applications, namely, computation of (1) fractional exponent, (2) absolute value, and (3) logarithm of arbitrary base. In Chapter 4, an important class of nonlinear controllers is realised and implemented in a closed-loop feedback system as a reference tracking problem. The design exploits bimolecular as well as unimolecular chemical reactions, allowing the implementation of highly nonlinear synthetic control circuits based on sliding mode control theory. Simulation results for the closed-loop response indicate that, compared to a traditional *proportional+integrator* (PI) controller, the implemented *quasi sliding mode* (QSM) controller results in dramatically faster performance with more accurate tracking of reference signals, as well as providing a more modular approach that is less affected by the presence of *retroactivity*. The proposed design fully exploits the inherently nonlinear nature of biomolecular reaction kinetics, and makes for the first time a direct link between the biological concept of ultrasensitivity and the engineering theory of sliding mode control.

An important factor in the design of any closed-loop feedback system is to analyse the system robustness under parameter uncertainty or variance. The proposed feedback system is analysed when subjected to potential accumulative process time delays in the production of the output species of interest, as discussed in Chapter 5. Different levels of uncertainty are introduced in the parameters representing the reaction rates of the underlying chemical reactions, and a process time delay is also included to investigate the robustness of both controllers (PI and QSM) to these variabilities. Simulation results highlight the strong robustness properties of the QSM controller, indicating its suitability for implementation in *in vitro* experiments.

Further, while exploring the properties of the set of CRNs underlying the covalent modification cycle, it was noticed that one of the operating regimes - the so-called signal-transducing regime - can be observed to approximate the steady-state input-output behaviour of a PI controller. Accordingly, an investigation of the application of this proposed *covalent modification cycle* (CMC) controller in order to track reference signals with both a linear or a nonlinear process is described in Chapter 6. It is followed by a robustness analysis of the CMC controller as compared to a PI controller in the presence of parametric uncertainty.

Several avenues for further research are opened up by this study. For successful implementation of complex feedback control circuits it will be essential to understand the trade-offs between system performance and complexity (particularly in terms of the number of chemical reactions to be implemented experimentally), as well as the effect of experimental uncertainties on closed loop performance (e.g. robustness to variations in reaction rates for complex circuit designs, etc). It would thus be interesting to investigate whether there are alternative sets of CRNs that could implement a QSM controller using fewer chemical reactions. One way to achieve this is to seek to utilize CRNs without the '$+$' and '$-$' formalism, as this could significantly reduce the number of reactions required to implement the proposed circuits and controllers. Sliding mode controllers are only one of many potential nonlinear control schemes that could potentially be implemented using DNA-based chemistry, and much work remains to be done to forge closer links between nonlinear control theory, chemical reaction network theory, and the experimental realities of nucleic acid implementations of complex dynamical systems. The treatment in this thesis has focussed on deterministic CRNs, but there has been much recent work on CRNs within a stochastic systems framework that could also be applied in the context of the design of biomolecular controllers. Lastly, while the assumption of well-mixed conditions in *in vitro* systems seems valid, the implementation of DNA-based circuits *in vivo* will require careful consideration of spatial factors, motivating the extension of the underlying design framework to include partial differential equation-based models.

Bibliography

[1] William S Klug, Michael R Cummings, et al. *Concepts of genetics.* Number Ed. 7. Pearson Education, Inc, 2003.

[2] Anthony Bailey, An Le Couteur, I Gottesman, P Bolton, E Simonoff, E Yuzda, and M Rutter. Autism as a strongly genetic disorder: evidence from a british twin study. *Psychological medicine*, 25(01):63–77, 1995.

[3] Bruce Alberts, Dennis Bray, Julian Lewis, Martin Raff, Keith Roberts, James D Watson, and AV Grimstone. Molecular biology of the cell (3rd edn). *Trends in Biochemical Sciences*, 20(5):210–210, 1995.

[4] Donald Voet and Judith G Voet. Biochemistry 4th, 2011.

[5] Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *Science*, 337(6096):816–821, 2012.

[6] Prashant Mali, Luhan Yang, Kevin M Esvelt, John Aach, Marc Guell, James E DiCarlo, Julie E Norville, and George M Church. Rna-guided human genome engineering via cas9. *Science*, 339(6121):823–826, 2013.

[7] Le Cong, F Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D Hsu, Xuebing Wu, Wenyan Jiang, Luciano A Marraffini, et al. Multiplex genome engineering using crispr/cas systems. *Science*, 339 (6121):819–823, 2013.

[8] Patrick D Hsu, Eric S Lander, and Feng Zhang. Development and applications of crispr-cas9 for genome engineering. *Cell*, 157(6):1262–1278, 2014.

[9] James A Stapleton, Kei Endo, Yoshihiko Fujita, Karin Hayashi, Masahiro Takinoue, Hirohide Saito, and Tan Inoue. Feedback control of protein expression in mammalian cells by tunable synthetic translational inhibition. *ACS synthetic biology*, 1(3):83–88, 2011.

[10] Oliwia Andries, Tasuku Kitada, Katie Bodner, Niek N Sanders, and Ron Weiss. Synthetic biology devices and circuits for rna-based'smart vaccines': a propositional review. *Expert review of vaccines*, 14(2):313–331, 2014.

[11] Burton W Andrews and Pablo A Iglesias. Control engineering and systems biology. In *Mathematical Methods for Robust and Nonlinear Control*, pages 267–288. Springer, 2007.

[12] Martin Feinberg. Lectures on chemical reaction networks. *Notes of lectures given at the Mathematics Research Center, University of Wisconsin*, 1979.

[13] Mariaconcetta Bilotta, Carlo Cosentino, Declan G Bates, Luca Salerno, and Francesco Amato. Retroactivity analysis of a chemical reaction network module for the subtraction of molecular fluxes. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 941–944. IEEE, 2015.

[14] Corentin Briat, Ankit Gupta, and Mustafa Khammash. Antithetic integral feedback ensures robust perfect adaptation in noisy bimolecular networks. *Cell systems*, 2(1):15–26, 2016.

[15] David Soloveichik, Georg Seelig, and Erik Winfree. Dna as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

[16] Wlodzimierz Klonowski. Simplifying principles for chemical and enzyme reaction kinetics. *Biophysical chemistry*, 18(2):73–87, 1983.

[17] Boyan Yordanov, Jongmin Kim, Rasmus L Petersen, Angelina Shudy, Vishwesh V Kulkarni, and Andrew Phillips. Computational design of nucleic acid feedback control circuits. *ACS synthetic biology*, 3(8):600–616, 2014.

[18] Georg Seelig, David Soloveichik, David Yu Zhang, and Erik Winfree. Enzyme-free nucleic acid logic circuits. *science*, 314(5805):1585–1588, 2006.

[19] David Yu Zhang and Georg Seelig. Dynamic dna nanotechnology using strand-displacement reactions. *Nature chemistry*, 3(2):103–113, 2011.

[20] Kevin Montagne, Raphael Plasson, Yasuyuki Sakai, Teruo Fujii, and Yannick Rondelez. Programming an in vitro dna oscillator using a molecular networking strategy. *Molecular systems biology*, 7(1):466, 2011.

[21] Jongmin Kim and Erik Winfree. Synthetic in vitro transcriptional oscillators. *Molecular systems biology*, 7(1):465, 2011.

[22] Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from dna. *Nature nanotechnology*, 8(10):755–762, 2013.

[23] Teruo Fujii and Yannick Rondelez. Predator–prey molecular ecosystems. *ACS nano*, 7(1):27–34, 2012.

[24] Maximilian Weitz, Jongmin Kim, Korbinian Kapsner, Erik Winfree, Elisa Franco, and Friedrich C Simmel. Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator. *Nature chemistry*, 6(4):295–302, 2014.

[25] Alvin Tamsir, Jeffrey J Tabor, and Christopher A Voigt. Robust multicellular computing using genetically encoded nor gates and chemical/wires/'. *Nature*, 469(7329):212–215, 2011.

[26] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–342, 2000.

[27] Michael B Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[28] Subhayu Basu, Yoram Gerchman, Cynthia H Collins, Frances H Arnold, and Ron Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–1134, 2005.

[29] Takayuki Sohka, Richard A Heins, Ryan M Phelan, Jennifer M Greisler, Craig A Townsend, and Marc Ostermeier. An externally tunable bacterial band-pass filter. *Proceedings of the National Academy of Sciences*, 106(25): 10135–10140, 2009.

[30] Kate E Galloway, Elisa Franco, and Christina D Smolke. Dynamically reshaping signaling networks to program cell fate via genetic controllers. *Science*, 341(6152):1235005, 2013.

[31] Kazuaki Oishi and Eric Klavins. Biomolecular implementation of linear i/o systems. *Systems Biology, IET*, 5(4):252–260, 2011.

[32] Michael Pedersen and Boyan Yordanov. Programming languages for circuit design. In *Computational Methods in Synthetic Biology*, pages 81–104. Springer, 2015.

[33] Niranjan Srinivas, Thomas E Ouldridge, Petr Šulc, Joseph M Schaeffer, Bernard Yurke, Ard A Louis, Jonathan PK Doye, and Erik Winfree. On the biophysics and kinetics of toehold-mediated dna strand displacement. *Nucleic acids research*, 41(22):10641–10658, 2013.

[34] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. Introduction: Intuitive theory of sliding mode control. In *Sliding Mode Control and Observation*, pages 1–42. Springer, 2014.

[35] Christopher Edwards and Sarah Spurgeon. *Sliding mode control: theory and applications*. CRC Press, 1998.

[36] Vadim I Utkin. Scope of the theory of sliding modes. In *Sliding Modes in Control and Optimization*, pages 1–11. Springer, 1992.

[37] Hassan K Khalil and JW Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

[38] Brittany Rauzan, Elizabeth McMichael, Rachel Cave, Lesley R Sevcik, Kara Ostrosky, Elisabeth Whitman, Rachel Stegemann, Audra L Sinclair, Martin J Serra, and Alice A Deckert. Kinetics and thermodynamics of dna, rna, and hybrid duplex formation. *Biochemistry*, 52(5):765–772, 2013.

[39] Domitilla Del Vecchio. The impact of retroactivity on the behavior of biomolecular systems. In *Design and Analysis of Biomolecular Circuits*, pages 161–181. Springer, 2011.

[40] Domitilla Del Vecchio, Alexander J Ninfa, and Eduardo D Sontag. Modular cell biology: retroactivity and insulation. *Molecular systems biology*, 4(1): 161, 2008.

[41] Shridhar Jayanthi, Kayzad Soli Nilgiriwala, and Domitilla Del Vecchio. Retroactivity controls the temporal dynamics of gene transcription. *ACS synthetic biology*, 2(8):431–441, 2013.

[42] G Oster and A Perelson. Chemical reaction networks. *IEEE Transactions on Circuits and Systems*, 21(6):709–721, 1974.

[43] Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. Programmability of chemical reaction networks. In *Algorithmic Bioprocesses*, pages 543–584. Springer, 2009.

[44] Mark E Davis and Robert J Davis. *Fundamentals of chemical reaction engineering*. Courier Corporation, 2012.

[45] Péter Érdi and János Tóth. *Mathematical models of chemical reactions: theory and applications of deterministic and stochastic models*. Manchester University Press, 1989.

[46] Mathias Foo, Rucha Sawlekar, Jongmin Kim, Declan G. Bates, Guy-Bart Stan, and Vishwesh Kulkarni. Biomolecular implementation of nonlinear system theoretic operators. In *European Control Conference*. EUCA, 2016.

[47] Shalini Mathias, Richard N KOLESNICK, et al. Signal transduction of stress via ceramide. *Biochemical Journal*, 335(3):465–480, 1998.

[48] Chaohong Li and Qingbo Xu. Mechanical stress-initiated signal transductions in vascular smooth muscle cells. *Cellular signalling*, 12(7):435–445, 2000.

[49] Liming Xiong, Karen S Schumaker, and Jian-Kang Zhu. Cell signaling during cold, drought, and salt stress. *The plant cell*, 14(suppl 1):S165–S183, 2002.

[50] G Poli, G Leonarduzzi, F Biasi, and E Chiarpotto. Oxidative stress and cell signalling. *Current medicinal chemistry*, 11(9):1163–1182, 2004.

[51] László Homolya, Thomas H Steinberg, and Richard C Boucher. Cell to cell communication in response to mechanical stress via bilateral release of atp and utp in polarized epithelia. *The Journal of cell biology*, 150(6):1349–1360, 2000.

[52] RL Juliano. Signal transduction by cell adhesion receptors and the cytoskeleton: functions of integrins, cadherins, selectins, and immunoglobulin-superfamily members. *Annual review of pharmacology and toxicology*, 42 (1):283–323, 2002.

[53] BD Gomperts, IM Kramer, and PER Tatham. Receptors. *Signal Transduction*, pages 33–69, 2004.

[54] Albert Goldbeter and Daniel E Koshland. An amplified sensitivity arising from covalent modification in biological systems. *Proceedings of the National Academy of Sciences*, 78(11):6840–6844, 1981.

[55] Alejandra C Ventura, Peng Jiang, Lauren Van Wassenhove, Domitilla Del Vecchio, Sofia D Merajver, and Alexander J Ninfa. Signaling properties of a covalent modification cycle are altered by a downstream target. *Proceedings of the National Academy of Sciences*, 107(22):10032–10037, 2010.

[56] Edwin G Krebs and Joseph A Beavo. Phosphorylation-dephosphorylation of enzymes. *Annual review of biochemistry*, 48(1):923–959, 1979.

[57] Courtney A Miller and J David Sweatt. Covalent modification of dna regulates memory formation. *Neuron*, 53(6):857–869, 2007.

[58] John D Rodwell, Vernon L Alvarez, Chyi Lee, A Dwight Lopes, JW Goers, H Dalton King, Henry J Powsner, and Thomas J McKearn. Site-specific covalent modification of monoclonal antibodies: in vitro and in vivo evaluations. *Proceedings of the National Academy of Sciences*, 83(8):2632–2636, 1986.

[59] Carlos Gomez-Uribe, George C Verghese, and Leonid A Mirny. Operating regimes of signaling cycles: statics, dynamics, and noise filtering. *PLOS Computational Biology*, 3(12):e246, 2007.

[60] James D Watson, Francis HC Crick, et al. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[61] James D Watson et al. Molecular biology of the gene. *Molecular biology of the gene.*, (2nd edn), 1970.

[62] Harvey Lodish, David Baltimore, Arnold Berk, S Lawrence Zipursky, Paul Matsudaira, and James Darnell. *Molecular cell biology*, volume 3. Scientific American Books New York, 1995.

[63] SD Roseff. Sickle cell disease: a review. *Immunohematology/American Red Cross*, 25(2):67–74, 2008.

[64] Martin H Steinberg, Bernard G Forget, Douglas R Higgs, and David J Weatherall. *Disorders of hemoglobin: genetics, pathophysiology, and clinical management*. Cambridge University Press, 2009.

[65] Yamuna Krishnan and Friedrich C Simmel. Nucleic acid based molecular devices. *Angewandte Chemie International Edition*, 50(14):3124–3156, 2011.

[66] Wade W Grabow and Luc Jaeger. Rna self-assembly and rna nanotechnology. *Accounts of chemical research*, 47(6):1871–1880, 2014.

[67] John SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide dna nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95(4):1460–1465, 1998.

[68] John SantaLucia Jr and Donald Hicks. The thermodynamics of dna structural motifs. *Annu. Rev. Biophys. Biomol. Struct.*, 33:415–440, 2004.

[69] Robert M Dirks, Justin S Bois, Joseph M Schaeffer, Erik Winfree, and Niles A Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM review*, 49(1):65–88, 2007.

[70] Paul J Hagerman. Flexibility of dna. *Annual review of biophysics and biophysical chemistry*, 17(1):265–286, 1988.

[71] AA Travers and JMT Thompson. An introduction to the mechanics of dna. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 362(1820):1265–1279, 2004.

[72] Robert Carlson. The changing economics of dna synthesis. *Nature biotechnology*, 27(12):1091, 2009.

[73] Tom Ellis, Tom Adie, and Geoff S Baldwin. Dna assembly for synthetic biology: from parts to pathways and beyond. *Integrative Biology*, 3(2):109–118, 2011.

[74] David Yu Zhang. Towards domain-based sequence design for dna strand displacement reactions. In *International Workshop on DNA-Based Computers*, pages 162–175. Springer, 2010.

[75] Chris Thachuk. Logically and physically reversible natural computing: a tutorial. In *International Conference on Reversible Computation*, pages 247–262. Springer, 2013.

[76] David Yu Zhang and Erik Winfree. Control of dna strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314, 2009.

[77] Robert RF Machinek, Thomas E Ouldridge, Natalie EC Haley, Jonathan Bath, and Andrew J Turberfield. Programmable energy landscapes for kinetic control of dna strand displacement. *Nature communications*, 5, 2014.

[78] Andrew Phillips and Luca Cardelli. A programming language for composable dna circuits. *Journal of the Royal Society Interface*, 6(Suppl 4):S419–S436, 2009.

[79] Matthew R Lakin, Simon Youssef, Filippo Polo, Stephen Emmott, and Andrew Phillips. Visual dsd: a design and analysis tool for dna strand displacement systems. *Bioinformatics*, 27(22):3211–3213, 2011.

[80] *MATLAB version 8.3.0.532 (R2014a)*. The Mathworks, Inc., Natick, Massachusetts, 2014.

[81] HJ Buisman, Huub MM ten Eikelder, Peter AJ Hilbers, and Anthony ML Liekens. Computing algebraic functions with biochemical reaction networks. *Artificial life*, 15(1):5–19, 2009.

[82] Francis Begnaud Hildebrand. *Advanced calculus for applications*, volume 63. Prentice-Hall Englewood Cliffs, NJ, 1962.

[83] Milton Abramowitz, Irene A Stegun, et al. Handbook of mathematical functions. *Applied mathematics series*, 55:62, 1966.

[84] Ramiz Daniel, Jacob R Rubens, Rahul Sarpeshkar, and Timothy K Lu. Synthetic analog computation in living cells. *Nature*, 497(7451):619–623, 2013.

[85] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, 2013.

[86] Domitilla Del Vecchio and Shridhar Jayanthi. Retroactivity attenuation in transcriptional networks: Design and analysis of an insulation device. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 774–780. IEEE, 2008.

[87] Albeter Goldbeter and DE Koshland. Ultrasensitivity in biochemical systems controlled by covalent modification. interplay between zero-order and

multistep effects. *Journal of Biological Chemistry*, 259(23):14441–14447, 1984.

[88] Chi-Ying Huang and James E Ferrell. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proceedings of the National Academy of Sciences*, 93(19):10078–10083, 1996.

[89] Qiang Zhang, Sudin Bhattacharya, and Melvin E Andersen. Ultrasensitive response motifs: basic amplifiers in molecular signalling networks. *Open biology*, 3(4):130031, 2013.

[90] Gene F Franklin, J David Powell, and Abbas Emami-Naeini. Feedback control of dynamics systems. *Addison-Wesley, Reading, MA*, 1994.

[91] Karl Johan Åström and Tore Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.

[92] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, 1999.

[93] Tildon H Glisson. *Introduction to circuit analysis and design*. Springer Science & Business Media, 2011.

[94] Reza Ghaemi and Domitilla Del Vecchio. Stochastic analysis of retroactivity in transcriptional networks through singular perturbation. In *American Control Conference (ACC), 2012*, pages 2731–2736. IEEE, 2012.

[95] Hamid R Ossareh, Alejandra C Ventura, Sofia D Merajver, and Domitilla Del Vecchio. Long signaling cascades tend to attenuate retroactivity. *Biophysical journal*, 100(7):1617–1626, 2011.

[96] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

[97] SP Bhattacharyya, H Chapellat, and LH Keel. Robust control: the parametric approach. *Upper Saddle River*, 1995.

[98] Randy Freeman and Petar V Kokotovic. *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

[99] Richard C Dorf and Robert H Bishop. Modern control systems. 1998.

[100] Ian Petersen, Valery A Ugrinovskii, and Andrey V Savkin. *Robust Control Design Using H-infinity Methods*. Springer Science & Business Media, 2012.

[101] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.

[102] Carlo Cosentino and Declan Bates. *Feedback control in systems biology*. Crc Press, 2011.

[103] Hiroaki Kitano. Biological robustness. *Nature Reviews Genetics*, 5(11):826–837, 2004.

[104] N Barkal and Stan Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–917, 1997.

[105] Marc R Roussel. The use of delay differential equations in chemical kinetics. *The Journal of Physical Chemistry*, 100(20):8323–8330, 1996.

[106] Mathukumalli Vidyasagar. Statistical learning theory and randomized algorithms for control. *IEEE Control Systems*, 18(6):69–85, 1998.

[107] PP Menon, Ian Postlethwaite, Samir Bennani, Andrés Marcos, and DG Bates. Robustness analysis of a reusable launch vehicle flight control law. *Control Engineering Practice*, 17(7):751–765, 2009.

[108] Introducing Excel Tables. Introduction to probability and statistics. 1972.

[109] Peggy S Williams. A monte carlo dispersion analysis of the x-33 simulation software. In *AIAA atmospheric flight mechanics conference, August*, pages 6–9, 2001.

[110] Bernard Friedland. *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.