UCL Chemical Engineering
Doctoral Thesis

A Meshless Modelling Framework for Simulation and Control of Nonlinear Synthetic Biological Systems

Asif Hamid Bhatti

2016

A thesis submitted for the partial fulfilment of the requirements for a degree of Doctor of Philosophy at University College London

I, Asif Hamid Bhatti, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

_____

(Signature)

_____

(Date)

Acknowledgements

Firstly I would like to thank my supervisor, Dr Vivek Dua, for always being so helpful and patient with me as I learnt the skills necessary to do a PhD in chemical engineering. Thank you for teaching me new skills and for taking a chance on me for this PhD. I have truly appreciated the opportunities that you have given me and hope I haven't deterred you from taking on non-engineering students in the future.

I would like to thank my friends from the office, Shade, Alexandros, Matt, Elnaz and Vassilis for always providing me with funny stories, humour and support. The days went a lot better when we could laugh about our situations and PhD life. Shade, I can't thank you enough for being as wonderful as you are. We really grew together during this PhD and I will never forget the fun times that we shared. Thank you for including me in all your many ventures, I was as always more than happy to help, and that will never change. I know you're going to make a big name for yourself within the university and I am glad I was there to see it from the beginning. Alexandros, thank you for being so helpful with GAMS, and for always being there to relate with when things were going wrong with modelling. We both shared many gripes when it came to the programming language and solvers. Matt, thank you for being there to talk to when things got boring and for relating with all the tensions in the office. I will leave it to you to keep Vassilis in check, good luck! Elnaz, thank you for being on Team Vivek, and for understanding what that meant from the early days. I won't forget the tutorials we did together and how we managed to rein in all the students. Finally Vassilis, I can't thank you enough for all the help and support you have given me, from modelling to my personal life. Thank you for being there and understanding everything I was going through. I know you are going to achieve amazing things and I will be hearing about you in UCL news. Try to stay calm when I'm gone!

Throughout my PhD I was fortunate enough to be the seminar organiser and PhD representative. These roles gave me so many new experiences and I couldn't have done them without help. Thank you to Prof. Papageorgiou for helping me with the seminars and giving me the freedom I needed to implement the 1st year seminar day. I am glad I can leave this legacy for the department and hope it will continue in future. Thank you also to the administrative staff, Agata, Mae, Patty, Susie, Katy and Clare, for all their help over the years. I couldn't have achieved so much in my roles without it. Thank you also to the IT department, especially Martin for sorting out any issues I had with the laptop for the seminars. During my time as the PhD representative I formed a committee to help with the issues we had in the

department, so I would like to thank Shade, Vidal, Noor, Banjo, Rema, David and Alex for being part of this and helping to make the year better for everyone. Thank you also to Mike and Simon for helping to sort out any issues we were having and contacting the relevant departments and companies to fix these issues.

I was also fortunate enough to find help and friends for all sorts of problems that I was having. Thank you to Professor Sorensen for your helpful advice and for being available when I needed to talk through things. Thank you for also including me in all the outreach you were helping to perform for the department. Thank you to James Stone from IT for never failing to fix any computer issues I was having straight away and for taking time away from your busy schedule to help me first. Thank you also to Mithila for always proving a positive energy and for the help with MATLAB tutorials. A big thank you to Lee and Hassan for making sure I got my packages without having to trace them down throughout the university, and for the morning chats that always started my day off with a smile.

Finally I would like to thank my family for always giving me the support I needed to complete the PhD. I can't thank my parents enough for being patient while I went back to university to pursue a doctorate. I know it took longer than expected but I also know it will all be worth it.

Abstract

Synthetic biology is a relatively new discipline that incorporates biology and engineering principles. It builds upon the advances in molecular, cell and systems biology and aims to transform these principles to the same effect that synthesis transformed chemistry. What distinguishes synthetic biology from traditional molecular or cellular biology is the focus on design and construction of components (e.g. parts of a cell) that can be modelled, characterised and altered to meet specific performance criteria. Integration of these parts into larger systems is a core principle of synthetic biology. However, unlike some areas of engineering, biology is highly non-linear and less predictable. In this thesis the work that has been conducted to combat some of the complexities associated with dynamic modelling and control of biological systems will be presented.

Whilst traditional techniques, such as Orthogonal Collocation on Finite Elements (OCFE) are common place for dynamic modelling they have significant complexity when sampling points are increased and offer discrete solutions or solutions with limited differentiability. To circumvent these issues a meshless modelling framework that incorporates an Artificial Neural Network (ANN) to solve Ordinary Differential Equations (ODEs) and model dynamic processes is utilised. Neural networks can be considered as mesh-free numerical methods as they are likened to approximation schemes where the input data for a design of a network consists of a set of unstructured discrete data points. The use of the ANN provides a solution that is differentiable and is of a closed analytic form, which can be further utilised in subsequent calculations. Whilst there have been advances in modelling biological systems, there has been limited work in controlling their outputs. The benefits of control allow the biological system to alter its state and either upscale production of its primary output, or alter its behaviour within an integrated system. In this thesis a novel meshless Nonlinear Model Predictive Control (NLMPC) framework is presented to address issues related to nonlinearities and complexity. The presented framework is tested on a number of case studies.

A significant case study within this work concerns simulation and control of a gene metabolator. The metabolator is a synthetic gene circuit that consists of two metabolite pools which oscillate under the influence of glycolytic flux (a combination of sugars, fatty acids and glycerol). In this work it is demonstrated how glycolytic flux can be used as a control variable for the metabolator. The meshless NLMPC framework allows for both Single-Input Single-Output (SISO) and Multiple-Input Multiple-Output (MIMO) control. The dynamic behaviour of the metabolator allows

for both top-down control (using glycolytic flux) and bottom-up control (using acetate). The benefit of using MIMO (by using glycolytic flux and acetate as the control variables) for the metabolator is that it allows the system to reach steady state due to the interactions between the two metabolite pools.

Biological systems can also encounter various uncertainties, especially when performing experimental validation. These can have profound effect on the system and can alter the dynamics or overall behaviour. In this work the meshless NLMPC framework addresses uncertainty through the use of Zone Model Predictive Control (Zone MPC), where the control profile is set as a range, rather than a fixed set point. The performance of Zone MPC under the presence of various magnitudes of random disturbances is analysed.

The framework is also applied to biological systems architecture, for instance the development of biological circuits from well-characterised and known parts. The framework has shown promise in determining feasible circuits and can be extended in future to incorporate a full list of biological parts. This can give rise to new circuits that could potentially be used in various applications.

The meshless NLMPC framework proposed in this work can be extended and applied to other biological systems and heralds a novel method for simulation and control.

Contents

List of Figures

2. Review of Synthetic Biology

List of Tables

Abbreviations

AcCoA	Acetyl Co-Enzyme A
Ack	Acetyl Kinase
AcP	Acetyl Phosphate
Acs	Acetyl Synthase
Acs_{LAA}	Low Stability Acetyl Synthase
ANN	Artificial Neural Network
ANN-MIMO	Artificial Neural Network – Multiple-Input Multiple-Output framework / model
ANN-RK4	Artificial Neural Network – Runge-Kutta Fourth Order Framework / Model
AP/OA/ER	Augmented Penalty on Outer Approximation with Equality Relaxation
aTC	Anhydrotetracycline
ATTλ	Lambda attachment site
BBD	Branch Reduce and Bound with Disaggregation
bla	Gene resistant to Ampicillin
BRB	Branch Reduce and Bound
CAD	Computer Aided Design
cat	Gene resistant to Chloramphenicol
CONOPT3	Solver in GAMS
CPLEX	IBM ILOG CPLEX Optimization Studio software
CPU	Central Processing Unit
CRIM	Cross-Reactive Immunological Material
CSTR	Continuous Stirred Tank Reactor
DDE(s)	Delay Differential Equation(s)
DICOPT++	Solver in GAMS
DMC	Dynamic Matrix Control
DNA	Deoxyribonucleic Acid
DOPF	Distribution Optimal Power Flow
Dvgly	Velocity constraint placed on Glycolytic Flux
E. Coli	*Escherichia Coli* (Bacteria)
ELM	Energy Loss Minimisation
FACTS	Flexible AC Transmission Systems
FD	Fluorescence Detector
GAMS	Generic Algebraic Modelling System (Software)
GBD	Generalised Benders Decompositions

GFP	Green Fluorescence Protein
gfpmut3.$_{\text{IAAV}}$	GFP Variant with intermediate stability
*gln*A	Gene
glnApL	Gene
*gln*G	Gene
HIV	Human Immunodeficiency Virus
HOAcE	Acetate (Protonated form)
HPLC	High Performance Liquid Chromatography
IPTG	Isopropyl β-D-1-thiogalactopyranoside
KNITRO	Solver in GAMS
LacI	*lac* repressor
LDC	Load Duration Curve
LP	Linear Programming
M1	Metabolite Pool containing Acetyl Co-Enzyme A
M2	Metabolite Pool containing Acetyl Phosphate
MIDO	Mixed Integer Dynamic Optimisation
MILP	Mixed Integer Linear Programming
MIMO	Multiple-Input Multiple-Output
MINLP	Mixed Integer Nonlinear Programming
MIPANN	Mixed Integer Programming with Artificial Neural Network
MPC	Model Predictive Control
mRNA	Messenger Ribonucleic Acid (RNA)
NADH	Nicotinamide Adenine Dinucleotide oxidised form
NLMPC	Nonlinear Model Predictive Control
NLP	Nonlinear Programming
NRI/NRIp	Repressor / Phosphorylated form
OA	Outer Approximation
OA/ER	Outer Approximation with Equality Relaxation
OAc	Acetate (Un-protonated form)
OCFE	Orthogonal Collocation on Finite Elements
OCFE-MIMO	Orthogonal Collocation on Finite Elements – Multiple-Input Multiple-Output Framework/Model
OCFE-NLMPC	Orthogonal Collocation on Finite Elements – Nonlinear Model Predictive Control Framework/Model
ODE(s)	Ordinary Differential Equation(s)
OPF	Optimal Power Flow
PDE(s)	Partial Differential Equation(s)

PEF3	Plasmid
PEMFC	Polymer Electrolyte Membrane Fuel Cell
PLM	Power Loss Minimisation
Pta	Phosphate Acetyltransferase
Pta_{LAA}	Low Stability Phosphate Acetyltransferase
$Ptet_2$	Promotor for the *tetR* protein
RK	Runge-Kutta
RK4	Fourth Order Runge-Kutta
RKF7	Runge-Kutta-Fehlberg
RNA	Ribonucleic Acid
ROPF	Reconfigure Integrated with Optimal Power Flow
RSM	Response Surface Method
S. cerevisiae	*Saccharomyces Cerevisiae* (Bacteria)
SAR	Structure Activity Relationship
SBB	Solver in GAMS
siRNA	Small Interfering Ribonucleic Acid
SISO	Single-Input Single-Output
SNIC	Saddle Node Bifurcation
SNOPT	Sparse Nonlinear Optimizer (Solver in GAMS)
SQP	Sequential Quadratic Programming
tac	Promotor
tan	Gene resistant to Kanamycin
TCA	Tricarboxylic Acid Cycle
tetR	Protein
TLC	Thin Layer Chromatography
VAck	Acetate Kinase flux
Vgly	Glycolytic Flux (Sugars, Fatty Acids and Glucose)
Vout	Acetate intercellular transport flux
Vpta	Phosphate Acetyltransferase flux
V_{TCA}	Tricarboxylic Acid Cycle flux

1. Introduction

Synthetic biology has acquired much interest in the last few years as a means of understanding how biological systems can be adapted or exploited to provide novel outcomes. Synthetic biology stems from the understanding of the biological system, coupled with engineering principles to make these systems more predictable. The term 'synthetic biology' was first introduced by Szybalski and Skalka (1978) and described how a new era in genetic engineering was emerging where genes could not only be described, but new genetic arrangements could be discovered and analysed. Modelling for design of engineered biological systems to predict system performance before fabrication is an important component of synthetic biology. In this sense it is similar to systems biology as they both rely heavily on computational modelling. The principles of synthetic biology are utilised in order to produce new biological entities such as gene circuits, enzymes, cells and new systems through redesign of existing well-characterised parts. This aspect distinguishes it from traditional molecular and cellular biology as these new parts can be tuned to meet specific performance criteria. Due to the nonlinearities present in biological systems there is limited knowledge on the parts within a biological system and how they interact. Whilst advances within biology have let to databases with well-characterised parts, such as the Biomodels database, there are still a lot of parts that have yet to be understood.

One of the agendas of synthetic biology is the creation of new systems by building models and measuring differences between expected and observed data. Attempts made at manipulating living systems at the molecular level will lead to better understanding and therefore new types of biological components and systems. Biological systems have evolved and adapted to continue to exist, rather than being optimised for human understanding. Thoughtful redesign of these systems allows for simultaneous understanding and implementation of engineered systems that can be utilised for the purposes of producing energy, manufacturing chemicals, fabricating materials and processing information. Progress in the field of synthetic biology has been made practical by the advancement in two of its fundamental technologies, DNA sequencing and synthesis. Sequencing has increased the understanding of components and the natural organisation of biological structures, and synthesis has provided the ability to test the designs of new synthetic parts and systems.

Synthetic biology has the potential to produce new and exciting products and as such has found application within a variety of industries such as healthcare, energy systems, nanotechnology, electrical engineering and genetic engineering.

Systems engineering in synthetic biology looks at how new products or new circuits in host cells can yield products beneficial to the pharmaceutical industry. One such example is the work by Ro *et al.* (2006) who found that a precursor to an anti-malarial drug could be synthesised in engineered yeast cells (*S. cerevisiae)*. Advances like this showcase how synthetic biology can aid in product manufacture, whereby the host cells act as reactors and produce high yields of the desired product. Not only would this decrease production costs, but if these systems are accurately modelled the waste normally seen in traditional manufacture processes can be decreased. Much work in the field has looked at genetic circuits, which have been characterised experimentally and redesigned using model based design principles. Systems such as the genetic toggle switch (Gardner *et al.*, 2000), the repressilator (Elowitz and Leibler, 2000) and the gene metabolator (Fung *et al.*, 2005) showcase how biological circuits can be modelled accurately based upon experimental data and analysis. Many of these circuits utilise *E. Coli* as a natural chassis for testing due to its ability to replicate easily and simple genetic system that can be manipulated easily. Each of the circuits mentioned have possible advantageous characteristics, such as the toggle switch being used as a logic gate in circuits, the repressilator exhibiting sustained limit cycles and highlighting the potential of constructing circuits from parts that are not found together naturally and the metabolator showing that *in silico* models can accurately correlate with experimental data.

Whilst these systems have paved the way forward for advancement in the field of synthetic circuit engineering, there is a distinct lack of control of such systems. Much like in the case of the repressilator where the transcription rate of mRNA played an important factor in the outcome of the system, whereby it could either exhibit steady state or reach a sustained limit cycle, a control system would enable either of the two outcomes to be favoured and hence alter the system dynamics. Typically biological systems have not been controlled due to the fact that they are difficult to characterise, however for a well-characterised system it is possible to control its outcome using modelling. This thesis will demonstrate a novel framework for meshless control of biological systems. Specific aims of this work are listed as follows:

1. Design of a new modelling methodology utilising a meshless framework that can control biological system outputs
2. Comparison of the new framework to traditional methods for optimising dynamic systems
3. Design of a model that can adapt to uncertainty within biological systems

1.1 Aims & Objectives

The aim of this work is to design a new framework for solving dynamic nonlinear systems which utilises a meshless artificial neural network (ANN). Once developed, the framework will be applied to biological systems firstly to simulate, and then to control their outcomes through the use of nonlinear model predictive control (NLMPC). It is envisaged that the development of a new framework will herald a novel way of designing a controlling biological systems, and will pose as an alternative to traditional techniques. The objectives for this research are detailed in the remainder of this section, which explores the implications of developing the new framework.

1.1.1 Design of a Meshless Framework for Control of Biological Systems

As mentioned previously there have been many methods that have successfully modelled biological systems and their dynamic nature. However there has not been any system that has been able to control the outputs from these systems. Whilst the systems have correlated well with experimental data, there is a need for control of these systems as this can improve yield, alter dynamics and change the overall outcome. It is envisaged that the control model will improve the overall system output and can lead on to profound findings, especially through experimental validation.

1.1.2 Comparison of the New Framework to Traditional Methods

It is important that the new framework developed in this work can give similar results to traditional methods that have been utilised in previous research. One such method discussed briefly before is orthogonal collocation on finite elements (OCFE). The new meshless framework developed in this work works in a similar principle to OCFE, however it does not discretise within finite elements. Instead it has a time step that uses a rolling time horizon to solve the model. This work will showcase the differences between each approach and evaluate which is best suited for dynamic systems.

1.1.3 Adapting to Uncertainties in Biological Systems

There are many uncertainties that can affect a biological system, ranging from changes in temperature, pH, inhibition of proteins, lack of feed source etc. This work will showcase how Zone MPC can be utilised to formulate a model that can adapt to external disturbances, which is important, as mathematical models of biological systems need to be feasible. Therefore any disturbance that the biological system can undergo *in vivo* needs to be replicated and taken account for *in silico*. This work takes a step forward in feasible models of biological systems through the use of random disturbance, with aims to create a model that is feasible both mathematically and experimentally.

1.2 Thesis Outline

The rest of the thesis is outlined as follows:

Chapter 2: A review of synthetic biology
This chapter will introduce the field of synthetic biology and discuss current trends in research. It will also discuss recent advances as well as potential scope for this work to fit in with the challenges faced in synthetic biology.

Chapter 3: Development of the meshless framework
This chapter will detail the new framework and show the mathematical concept of the meshless NLMPC. This chapter will also showcase how the framework can be verified using fourth order Runge-Kutta and show how it differs to the traditional method of OCFE.

Chapter 4: A synthetic gene metabolator case study
This chapter will introduce the case study used for this work and show how it integrates with the new framework. Results presented in this chapter will highlight key differences between using the meshless framework and OCFE to model the system. The concept of control will also be shown as well as optimising under disturbance.

Chapter 5: Comparison of the new meshless framework with OCFE
This chapter highlights the differences between the new meshless framework developed in this work and OCFE. The metabolator case study from the previous chapter is re-visited and a detailed comparison between results from the two

techniques is discussed. Smaller studies are also present within this chapter to showcase the benefits of each technique further.

Chapter 6: Utilising the meshless framework for systems architecture

This chapter will introduce how the meshless framework can be used to solve MINLP problems based on biological systems architecture problems. Essentially the framework is applied to find the optimal system from given biological parts, which act like building blocks for a biological circuit. The MINLP is converted to an NLP and solved using the developed ANN-RK4 framework.

Chapter 7: Conclusions and future work

This chapter will summarise the key contributions of this work and also detail how the framework can be used in future studies.

2. Review of Synthetic Biology

In this chapter the concept of synthetic biology will be introduced and discussed. The core principles as well as the general scope for the field will be reviewed and key achievements in literature will be cited. The overall aims of synthetic biology will also be outlined as well as how this project can help fill gaps within the literature.

Synthetic biology is a branch of systems biology, which is an interdisciplinary field that looks at the complex interactions within biological systems. The main aim of systems biology research is to model and discover the various properties of cells, tissues and organisms by looking at metabolic networks or cell signalling pathways. A cellular network can be modelled mathematically using methods from chemical kinetics and control theory. Whereas systems biology looks at modelling a complete system through simulation, synthetic biology aims to forge new systems from existing pathways through synthesis or design of systems. A possible definition of the term 'synthetic biology' is the engineering of biological components and systems that do not exist in nature and the re-engineering of existing biological elements. It is determined on the intentional design of artificial biological systems, rather than on the understanding of natural biology. Much of synthetic biology is configuring new biological systems based on different parts of a cell. Each system is brought together and altered to work with one another to give new outcomes and products. These systems work well individually *in vivo*, but are not necessarily found in configuration with one another naturally. In many respects one of the greatest challenges in process design is to design a configuration that can produce chemicals in a reliable, safe and economical manner, whilst also producing a high yield and little waste (Seider *et al.*, 2010). Synthetic biology has similar aims in the sense that a system is designed to have a more desired outcome and is tested against the existing model or design.

2.1 The Aims of Synthetic Biology

Synthetic biology is essentially the engineering of biology. It involves the synthesis of complex systems, which are either biologically based or inspired, to perform functions that do not appear naturally. It is believed that this approach can manifest into a rational and systematic design of systems that can help to address major challenges faced currently and in the future. Possible applications of synthetic biology could include creation of systems to generate power (such as biological fuel

cells), nano-scale biological computers (like lab on a chip), new medical applications, biosensors for healthcare, new approaches for cleaning waste and even security applications. Previously biologists have sought to understand existing biological systems, and in doing so have acquired profound knowledge on the construction and functions of these systems. The aim of synthetic biology then is to design components in a standardised manner and combine these to construct novel genetic devices, metabolic pathways and optical or electronic devices. It is now becoming easier to synthesise gene and large DNA fragments due to recent advances in biology, so the standardisation of biological parts gives greater understanding on how these components can be put together in a bacterial chassis. Due to the complexity of synthetic biology it is necessary to utilise skills from a variety of experts including engineers, biologist, chemists and physical scientists. Together they are able to design cells, create enzymes and introduce new biological modules that can be used in a wide variety of industries including pharmaceuticals (Pieru *et al.*, 2005; Ro *et al.*, 2006), biological fuel cells (Yong *et al.*, 2011), polymers (Verdezyne, 2011) and even tissue engineering (Hu *et al.*, 2008). The key features of synthetic biology are on different levels of living system and are summarised below:

- Engineering complex living systems containing components of artificial biologically compatible and functional structures which can be maintained through the natural cell life cycle.
- Deployment of artificial regulators in circuits with designed functions rather than simple single regulator modifications.
- Using artificially regulated genes used for structural changes or catalysts within both existing and artificial pathways.
- Production of artificial molecules as primary or secondary gene products.

The artificial molecules produced through synthetic biology can be separated into macromolecules and small molecules as follows:

- Macromolecules: For intracellular use as catalysts or extracellular as molecular sensors, building blocks for complex subcellular structures or for other functions e.g. RNAs or enzymes.
- Small molecules: For use as biological drugs, building blocks for chemical synthesis, or for the purpose of engineering of more complex nanotechnology for different applications.

Synthetic biology can integrate with many related and interdisciplinary fields of science and technology, a summary of which is presented in figure 2.1. The field of synthetic biology covers many areas from other fields and utilises the knowledge gained from these fields to enhance system understanding and design. Biology, in particular bio-informatics, is used to gain understanding of the system through the use of novel software. This software can also aid in systems biology as it allows simulation of the biological entities and can aid in deeper understanding of the dynamics of the system. Engineering technologies are important and are utilised by synthetic biologists in order to design new systems, both *in silico* and *in vivo*. The knowledge gained from molecular biologists on the structure and functions of macromolecules (e.g. proteins, enzymes and nucleic acids) is vital in order to ensure that the systems designed through synthetic techniques are biologically feasible. Lastly nanotechnology allows for characterisations of the small molecules present within biological systems and yields greater understanding of their functions. From an engineering viewpoint synthetic biology can describe both top down approaches, such as creation of *de novo* artificial life, and bottom up approaches such as genetic engineering. The differences of both of these approaches will be discussed in section 2.1.1 and 2.1.2.

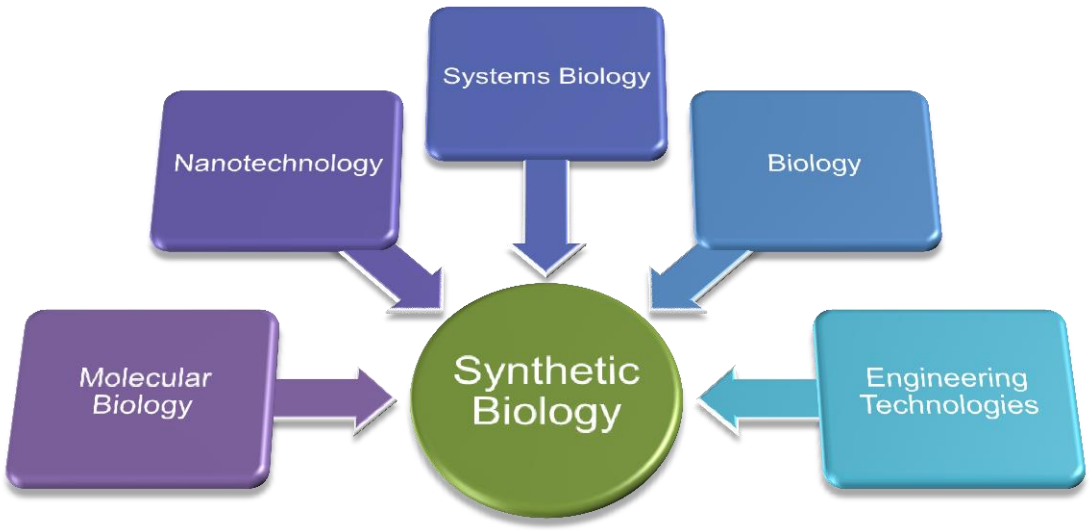

Figure 2.1: Synthetic Biology and its integration in related fields of science and technology

2.1.1 Top-Down Synthetic Biology

The theory of artificial carbon based life has been a subject of interest in recent years. The observations made in macro and microorganisms in nature show that it is possible to emulate life processes and could potentially create artificial life in the future (Grant, 1991). The theory looks at how artificial components of biological functionality can be integrated into synthetic systems which have the necessary qualities to sustain life. In this fashion, by looking at the system in greater detail by delving deeper into the system mechanics and processes, a top-down approach is generated. Each system is explored, and sub-systems are characterised until a greater understanding is achieved. This knowledge is then applied to other similar systems and processes are evaluated before they are altered. Although this is still in an early developmental phase, there have been certain areas where it has shown promise, such as the artificial leaf by Nocera (2011). This also has benefit for the advancement of nanotechnology to emulate living systems by applying natural nanostructures into systems that could process features of living systems. This is a goal of many nanotechnologists as they move towards the minimal cell (Luisi, 2002; Rasmussen *et al.*, 2004). A prerequisite for such systems would be that at least one component of the synthesis precursors for the minimal cells should be able to self-maintain a life cycle, and the fusion product should behave in the same way. It is foreseen that the eventual goal of this research would be to combine the artificial minimal cells with a natural biological system, such as human beings, and has the potential to eradicate disease, correct genes and even cure chronic illness.

2.1.2 Bottom-Up Synthetic Biology

The main area that utilises the bottom-up approach to design in synthetic biology is genetic engineering. This approach differs from top-down as it pieces together a system from known parts, thereby giving rise to more complex systems. An artificial DNA sequence is one that has been modified by human influence so that it is no longer naturally occurring, and the main challenge is to achieve this without losing functionality. It is regarded that having a variety of sequences is crucial to have different functionalities, or even create new functionality. With regards to this there are three main areas where functionality plays a role; regulation of function, regulated function for specific control and sensor function. Whilst there are many different applications for the generation of recombinant cells through genetic engineering, the location of the point at which simple recombinant DNA experiments end and synthetic biology begins is up for discussion. Firstly, most recombinant

DNA experiments in the past were performed for the sole purpose of understanding biological functions such as:

- The regulation of cell function, pathways and structure.
- Structure-function relationships e.g. affinity, binding ability etc.
- Catalytic properties such as kinetic behaviour, sensitivity and selectivity as well as qualitative features such as stability, temperature and pH.

Although the DNA constructs that were engineered for this purpose were stable, increased knowledge of how the wild type function works has made these constructs obsolete. There are however many successful experiments of genes for bio production in organisms such as fungi, cell cultures, microorganisms, plants and some transgenic animals (Sorensen and Mortensen, 2005). These experiments typically produce small proteins, and there is debate on whether or not this is considered synthetic biology due to the simplicity of the resulting proteins. In order to define if the small proteins developed in this way are considered synthetic biology one has to look at the purpose for which the protein is intended as some are used for a systems biology application. Generally systems biology is the analysis of how regulatory and sensor functions control natural pathways and how they interact at the systemic level. With the increasing knowledge about systemic behaviour however, synthetic biology has the opposing intention, whereby it aims to design artificial pathways to optimise regulation or extend the catalyst capabilities of systems to gain a technical solution. These technical solutions include the use of mathematical models as simulation algorithms to create information for optimisation. Synthetic biology therefore applies solutions to biological systems in order to alter or influence their function towards a desired optimum, by generating artificial designs that are compatible with biological functions at a molecular level.

2.2 Current Trends in Synthetic Biology

Synthetic biology aims to utilise existing biological systems and alter their function in order to forge novel systems that can help to combat issues across various sectors in science and engineering. As such there are a number of advances in recent years that showcase the breadth and scope for synthetic biology, as well as the benefits of the emerging field to tackle grand challenges of current interest. This section will detail advances in the field of synthetic biology and detail key findings from

breakthrough research that can impact on healthcare, energy systems, material engineering and genetic engineering.

2.2.1 Synthetic Biology Applications in Healthcare

Healthcare and pharmaceuticals is one of the most applicable fields that will utilise technologies created by synthetic biology. Many biological systems can be altered to give new functions or products and these have significant uses within healthcare. Prominent cases for the production of pharmaceuticals can be seen in research from Pieru *et al.* (2005), who showed how *E. coli* could be used to produce an antibiotic, Erythromycin C, and an anti-malarial drug Artemisinin from Ro *et al.* (2006) who highlighted the benefits of using synthetic biology for pharmaceutical manufacture. The success of the latter has sparked interest in exploiting bacteria for large scale production of pharmaceuticals and has the potential to replace traditional methods. Further research into application for healthcare in synthetic biology has looked at the production of biosensors. Instances include recent developments of biological markers for detection of diseases in the gut (Kotula *et al.*, 2014) or probiotics for urinalysis and cancer detection (Danino *et al.*, 2015). Kotula *et al.* (2014) engineered a bacterial strain of *E. coli* to secrete biomarkers in response to flora in the gut, which would in turn diagnose any intestinal issues or diseases within the subject. Artificial biosensors are important in healthcare as they can be exploited to interact with both organic and inorganic molecules, and can respond to external influences such as temperature and light to generate signals (Hellinga and Marvin, 1998; Nivens *et al.*, 2004). Similarly, work by Danino *et al.* (2015) showed how using *E. coli* Nissle 1917, a probiotic, could be formulated as an oral dose for metastatic detection in the liver of rats. This profound research can be extended in future to allow for personalised cancer testing in patients and can revolutionise the treatment of such cancers. The concept of attacking cancer cells using engineered bacteria has also been researched for colorectal cancers by Anderson *et al.* (2006). The concepts developed within this research were coupled with research from Xiang *et al.* (2006) to produce a gene promotor that could re-programme the human commensal gut bacterium *Bacterioides thetaiotaomicron* in response to stimuli in mice (Mimee *et al.*, 2015), which has benefits for cellular sensing as well as localised treatment at the primary target site. This is largely due to the fact that the engineered construct does not trigger an immune response and so can integrate with the host cell without causing long term damage. However, scale-up of such systems is required for more complex gut flora.

Much research focusses on the use of a bacterial chassis for delivering a novel construct or system into a host system. However immune-centric cells like T cells can also be exploited as vectors for synthetic biology. Due to their nature as innate immune cells they have the ability to be used in treatment of cancer and immunological diseases. Two well known cases of T cells as vectors for synthetic biology are presented in research by Kalos and June (2013), who utilised T cells for cancer immunotherapy, and June and Levine (2015), who showed how T cells can also be used to treat HIV. Both of these cases demonstrate the usefulness of synthetic biology constructs as powerful tools for treatment and eradication of both acute and chronic diseases. Furthermore, antibiotic therapies can also be produced using engineered bacteria that can attack pathogens within the body to help eradicate infections. Microbes can be used to treat pathogens such as *Pseudomonas aeruginosa* (Saeidi *et al.*, 2011) or used as adjuvants for antibiotic therapies (Lu and Collins, 2009; Krom *et al.*, 2015). The future perspective of these technologies affect the medical and pharmaceutical industries, with enabling technologies such as *de novo* synthesis of genes and stem cells, manufacturing of drugs outside of cells and traditional chemical synthesis becoming obsolete to an extent.

2.2.2 Synthetic Biology Applications in Genetic Engineering

Much like the constructs made for healthcare applications, whereby bacteria or cells were engineered to attack diseases *in vivo*; genetic engineers are using similar principles to help re-programme cells to correct gene disorders, as well as exploiting both RNA and DNA as sources for this manipulation. As synthetic biology involves a paradigm shift from molecular biology to 'modular biology' (Hartwell *et al.*, 1999) there is much effort in designing artificial regulatory components that exhibit technical features. The intention is to generate sets of biologically compatible switches that are composed of only a few parts but can create varieties of different circuits (Gardner *et al.*, 2000; Endy and Yaffe, 2003; Wall *et al.*, 2005). Their functional characteristics are intended to be optimised and the behaviour of these switches is intended to be predictable (Atkinson *et al.*, 2003). In order to achieve predictability of single components in gene assemblies different regulators are needed to those in nature. This can be either regulating complex schemes of networks integrating many signals to form different circuits, or providing improved productivity for the purpose of high throughput product yield. In order to understand biological functions previous genetic engineering experiments looked at changing regulatory parameters that depend on or affect the DNA itself, or on the

protein whose function is to bind to DNA (Buchler *et al.*, 2003; Alper *et al.*, 2005). Regulation of the function of RNA (Isaacs and Collins, 2005) is effective mostly on the translation level of cellular function and the stability of molecules like RNA and proteins. Degradation rate of these molecules are important and have also been studied (Buchler *et al.*, 2005). Approaches for synthetic biology rely on more than just the modification of a single regulatory parameter, such as complex networks that can control more than one genetic function for regulated synthesis of products (Elowitz and Liebler, 2000; Butler *et al.*, 2004; Campbell, 2005).

Recent activity in using synthetic biology for genetic engineering has looked at the use of synthetic mRNA for re-programming of cells (Warren *et al.*, 2010) and genome activation and repression though CRISPR/Cas methods (Kiani *et al.*, 2015). Both of these have the ability to alter the genetic makeup of the cells they are treated against and has given geneticists a promising avenue to further research for future applications. Another highly regarded field is regenerative medicine, whereby cells are modified to forego the usual cell cycle and correct any disorders that the cell may induce. Although this is in its infancy for synthetic biology (Davies, 2016), there has been promising research to use such technology for therapeutic application (Lienert *et al.*, 2014). A global project looked at the creation of a fully functioning synthetic chromosome (Annaluru *et al.*, 2014). This achievement utilised the knowledge of the *S. cerevisiae* chromosome III and synthesised a smaller designer eukaryotic chromosome called synIII. As the first of its kind the research provides ground-breaking insight into how bacteria can be utilised for synthetic constructs for eukaryotic genomes. The potential applications for engineered products include the optimised productions of xeno-biomolecules such as therapeutic proteins, biologics, enzymes, antibodies, fine chemicals or biobased commodities (Straathof *et al.*, 2002; Panke *et al.*, 2002; Panke and Wubbolts, 2002, Panke *et al.*, 2004; Panke and Wubbolts, 2005). The enzymatic biotransformation of chemicals from artificial educts to generate product is under tremendous study and is expected to impact greatly on the future of 'green' chemistry as it intends to replace traditional chemical production (Muller, 2004; Panke and Wubbolts, 2005).

Genes are also not the only target to create a shift in biological function, and research is looking into designing whole microorganisms that can be used in both industrial and clinical applications (Yokobayashi *et al.*, 2002; Ferber, 2004). Engineered circuits however can have difficulty in deployment beyond the lab as they require controlled environments in order to function. Research by Pardee *et al.* (2014) showcased how a paper-based cell free system can be used as an abiotic distribution method for synthetic biology technologies. They show how this novel system can be used as a colorimetric band detector, much like litmus paper, for *in*

vitro diagnostics, including glucose detection and Ebola virus sensors. This technology shows how synthetic biology is able to produce an efficient and relatively inexpensive technology that can adapt to have many uses. This is the key advantage of using synthetic biology within the field of genetic engineering and the promise of new and exciting products that utilise the advances from previous research can potentially solve the myriad of problems faced by geneticists today. One key ethical advantage is that many of the constructs are formulated in bacteria, and so can circumvent the use of animal models.

2.2.3 Synthetic Biology Applications in Material Engineering

The fast growing field of material engineering is a prime candidate for synthetic biology, as the biological constructs can provide a fast, efficient and low-cost alternative to traditional manufacturing techniques. Different applications can be designed for complex regulatory schemes and these in turn can be used to produce enzymes, small molecules in a cell reactor and products that are produced *in vitro* or *in vivo*. Furthermore, whilst these systems have to have some level of basic functionality, they must also have an effect on more complex biological functions, such as programmed pattern formation of multicellular systems, organs and organisms (Basu *et al.*, 2005). This approach can be applied in the future for tissue engineering, stem cell therapy, drug delivery and regenerative medicine. Recent work in tissue engineering ranges from synthetic bone (Hu *et al.*, 2008) to artificial blood vessels (Ma *et al.*, 2010) and bladder smooth muscle (Tian *et al.*, 2010). Advances in this field have also produced various materials like biopolymers from *E, coli* (Kelwick *et al.*, 2015) and biomaterials from bacterial cellulose (Florea *et al.*, 2016). These materials can have a myriad of uses and cement synthetic biology as a plausible option for large scale manufacture. More research into using this technique for production of organic tissues and polymers is highly sought after and in years to come the commercialisation of these materials will be more viable, therefore providing materials at a much faster rate than current methods.

2.3 *In Silico* Synthetic Biology

Design of biological systems is useful in accelerating the evolution of certain systems and can be used to substitute for natural genetic rules. The advancement of this research relies heavily on computers and bioinformatics and there are now a large

number of computer algorithms for the analysis of biomolecule sequence and structure related issues. Initially, computing was used to find sequence motives for nucleic acids and protein structures, and algorithms enabled the design of single biomolecular components, such as RNA and glycoproteins. Structural and sequence data, as well as functional data, were collected from experimental results and stored in databases (Palsson, 2002; Mester *et al.*, 2004; Adai *et al.*, 2004). This created data and information for the rational design of molecules that can be manipulated by changing structure, specificity and selectivity (Bathelt *et al.*, 2002). Other algorithms have been developed for the simulation of designed evolution based processes (Fox, 2005). In order to design gene regulatory circuits the basic principles of systems and networks must be understood and the main task of systems biology is to gain this understanding by modelling the natural systems using data from the 'omic' projects (Venter *et al.*, 2003). Systems biology software, in contrast to bioinformatic software, is not focussed on the analysis and design of single components of a regulatory system but the analysis of all of the components. This is possible *in silico* through computational studies of gene networks creating *in numero* molecular biology (Arkin *et al.*, 1998; Hooshangi *et al.*, 2005). The simulation of cellular behaviour is composed in virtual environments (Slepchenko *et al.*, 2003), however most algorithms in biocomputing are secondary to experiments and research is being undertaken to separate *in silico* synthetic biology from databases of experimental algorithms (Ruben and Landweber, 2000; Schmidt *et al.*, 2004).

In Silico synthetic biology focusses on the design of artificial regulatory systems and circuits. It moves towards the simulation of predictive artificial molecular processes on the molecular and genomic level, and the level of regulatory circuits of total pathways (Hasty *et al.*, 2002; Isaacs *et al.*, 2003; Francois and Hakim, 2004). Software packages should provide tools for the design of regulatory circuits and individual components. Research aiming at providing the data for this has become the subject of intense development, with the most advanced being the registry of standard biological parts (MIT, 2016). The registry offers a standardisation of biological parts, which have been characterised through research from geneticists, biologist and engineers. This was the subject of interest in research by Dasika and Maranas (2008) who utilised the registry to find parts that could potentially form circuits. They developed software, called OptCircuit, which takes data from parts and by using algorithms set by user requirements, to form circuits that have the potential to be biologically feasible. The work carried out by the team behind OptCircuit showcased how the software can be used to form constructs similar to the gene oscillator from Gardner *et al.* (2000). Furthermore the software has shown it is able to form a genetic decoder through the use of logic gates, as well

as a concentration band detector for green fluorescent protein. Much of the research into *in silico* synthetic biology has looked at creating models based on experimental data and analysis. Whilst these models are accurate in depicting the biological system, there is a lack of design based on modelling techniques. Systems like the Goodwin oscillator (Lutz and Bujard, 1997), the Repressilator (Elowitz and Leibler, 2000), the metabolator (Fung *et al.*, 2005) and the mammalian oscillator (Tigges *et al.*, 2009; Tigges *et al.*, 2010) all were firstly constructed in the laboratory before they were modelled. This is largely due to the fact that many of the processes within these constructs had not been quantified and the dynamics of the systems were not yet fully realised. However the work from these researchers gave accurate models of these systems, which can now be used as a basis for design. This thesis will showcase the use of these systems for design and control of biological systems, which has not been achieved before.

2.4 Summary of 'Synthetic Biology'

Synthetic biology can be defined as the boundary between biological and engineering sciences in which technical approaches are employed to provide novel applications. All current, or intended applications, focus on the design of artificial living systems, such as specialised cells for bioproduction of molecules for *in vivo* or *in vitro* use. The key features of synthetic biology are on different levels of living systems:

1. Deployment of living systems to engineer complex patterns containing components of biologically compatible and functional assemblies, which are managed through continued life cycles.
2. Artificial assemblies of regulators with designed functions rather than simple modifications.
3. Functional assemblies of artificial pathways by modified genes with either structural or catalytic function.
4. Production of artificial molecules as primary or secondary products:
    a. Macromolecules: Like RNA or enzymes, for intracellular use as catalysts, or extracellular use as molecular sensors.
    b. Small molecules for use as biological drugs, chemical synthesis or for nanotechnology engineering or more complex units for different applications.

Therefore the first word, 'synthetic', describes the synthesis of artificial and natural components forming a new artificial living system. As technology for systems design, synthesis and optimisation mature, there will be rapid growth in the capabilities of synthetic systems which can have a wide range of applications. The secondary intention of synthetic biology is the deployment of highly functional artificial assemblies of designed regulatory circuits for effective highly controlled production of natural products, biochemicals and xenobiotics (Herrera, 2005). The intended application of *in vitro* synthetic biology products is highly regarded as truly novel science. These applications need to use parts, compounds or building blocks which are provided by *in vivo* synthetic biology, which cements this as an indispensable prerequisite for providing materials for synthetic biology applications. The field is objective driven and is not primarily a discovery science as it builds on current understanding, whilst also simplifying some of the complex interaction characteristics of natural biology. It also looks at design based engineering of systems based on biological functions and laws, and aims to provide new functions that are not present in nature.

The next section (Section 2.5) will detail these systems and discuss their strengths and weaknesses, as well as showcasing the development of oscillators within the field of synthetic biology.

2.5 Synthetic Gene Oscillators

Gene oscillators are primary candidates for synthetic biology approaches as they benefit greatly from technology discovered within the field. Research focus is on model based design and validation, with some performing experiments to test the theory. Trends in the data show that there are unique ways in which an oscillator can be constrained, characterised and dynamically implemented (Purcell *et al.*, 2013). There are two paradigmatic types of networks that are of interest in the scientific community; switches and oscillators (Tyson *et al.*, 2008), and over the years there have been numerous designs that have been proposed for both in prokaryotic and eukaryotic cells. This section will review the characteristics of the various different oscillators that have been published and highlight the advantages and disadvantages. The complexities of the oscillators range from the very early Goodwin oscillator (Goodwin, 1965) to more recent oscillators that have been constructed within mammalian cells (Tigges *et al.*, 2010).

2.5.1 The Goodwin Oscillator

The Goodwin oscillator, Goodwin (1965), was the first synthetic genetic oscillator to be studied and is also the simplest. It comprises of a single gene that represses (inhibits) itself (Figure 2.2a). Whilst early theoretical work was promising, it is only recently that models have been used to characterise and study the oscillator. These models use ODEs (Mueller *et al.*, 2006), delay differential equations, DDEs, (Smith, 1987) and stochastic simulations using the Gillespie algorithm (Bratsun *et al.*, 2005; Gillespie, 2007; Stricker *et al.*, 2008). The presence of oscillations within the Goodwin network were confirmed using *in silico* experiments using various techniques. An example of which (Stricker *et al.*, 2008) showed a detailed simulation that correlated well with *in vivo* work. It was shown that oscillation decay over time when predicted with a deterministic model, but persist when the Gillespie algorithm is applied. This does however only occur under certain conditions and suggests that noise within the system is parameter dependent. A robust period does however arise and was later confirmed by Lewis (2003). Given the nature of the oscillator it was deemed important to construct it *in vivo* for further study. This used a $P_{LlacO-1}$ promoter (Figure 2.2b), which is repressed by LacI (Lutz and Bujard, 1997) and is capable of giving transcription, where DNA is converted into RNA, at high levels when unrepressed and produces a negative feedback loop. The results were consistent with simulations, however, oscillations for both simulations and experimental results were highly irregular, thereby making this oscillator difficult to utilise. Figure 2.2b shows a typical schematic of a gene system, and is common in many biological papers; it essentially shows the promotor and corresponding gene, but these schematics can also be used to show gene-inducer, promotor-inducer, gene-gene and gene-transcript pairs.

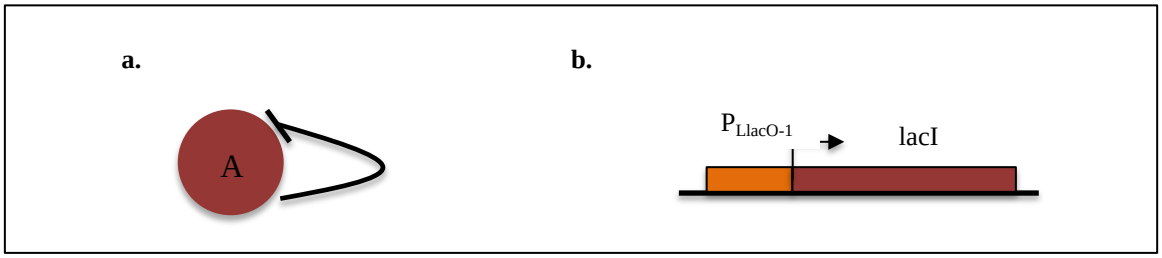

Figure 2.2: a. Topology of the Goodwin Oscillator showing gene A, **b.** *in vivo* implementation of the oscillator (Lutz and Bujard, 1997).

2.5.2 Repressilators

Repressilators (Elowitz and Liebler, 2000) can be thought of as extensions to the Goodwin oscillator as they are defined as a regulatory network of one or more genes with each gene repressing its successor in the cycle (Mueller *et al.*, 2006). Repressilators are capable of producing periodic oscillation and have been modelled extensively using ODEs (Elowitz and Liebler, 2000; Mueller *et al.*, 2006), DDEs (Smith, 1987; Wang *et al.*, 2005) and discrete stochastic simulations using the Gillespie algorithm (Yoda *et al.*, 2007). Repressilators can also be seen in other fields of study such as in neuroscience where cyclic networks of neurons, referred to as neural ring networks, are present and in electronics where a cycle of an odd number of NOT gates are referred to as a ring oscillator (Pasemann, 1995).

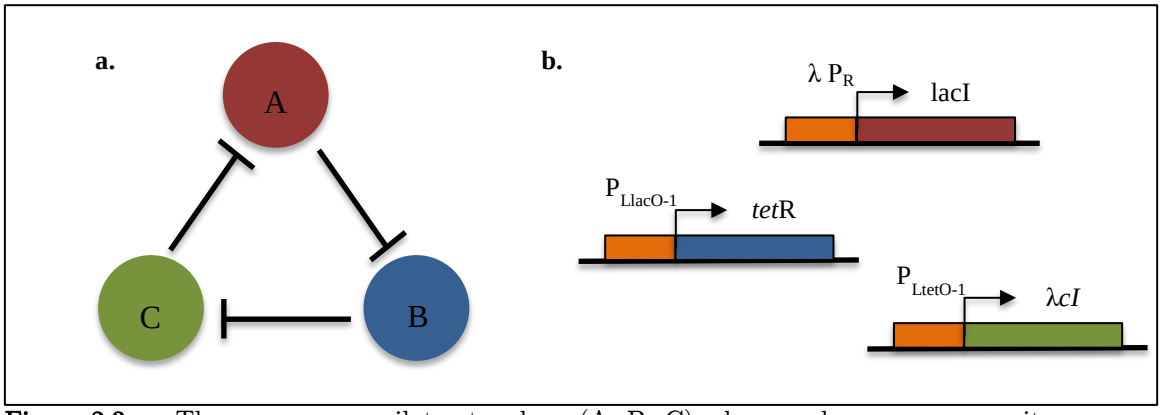

Figure 2.3: a. Three-gene repressilator topology (A, B, C) where each gene represses its successor in the cycle, b. *in vivo* implementation of the three-gene repressilator where LacI represses *tetR* through $P_{LlacO-1}$, TetR represses λ *cI* through $P_{LtetO-1}$ and cI represses *LacI* through λ P_R completing the cycle. All genes contain an ssrA sequence tag to promote rapid degradation.

Network topologies can be symmetrical as shown in research by Elowitz and Liebler (2000) and Mueller *et al.* (2006); however, these networks are difficult to replicate *in vivo* due to the fact that each gene has identical parameter sets. However, a three gene repressilator (Figure 2.3) was constructed *in vivo* using LacI from *E. Coli*, TetR from the Tn10 transposon and cI from the λ phage (Elowitz and Liebler, 2000). Within this network LacI represses transcription of *tetR*, TetR represses transcription of *cI* and cI closes of the cycle by repressing *lacI*. The repressilator was copied into a *lac* operon deficient strain of *E. Coli* and it was shown that 40% of the cell exhibited oscillations. Oscillations were periodic and had unique characteristic whereby they were not governed by cell division, indicating that the network is decoupled from the cell division cycle, but oscillations stopped when *E. Coli* was in stationary phase, thereby indicating that the dynamics are coupled with global

regulation and effects of cell growth and division (Elowitz and Liebler, 2000). The repressilator became the first synthetic genetic oscillator to be successfully implemented *in vivo*, however, due to the fact that less than half of the cells experienced oscillation, it lacked robustness. Improvements to the design have been studied by Tsai *et al.* (2008), where adding a positive feedback loop expands the region in parameter space over which the core repressilator oscillates thereby enhancing robustness. This addition also allows the repressilator to exhibit greater range of frequencies for given amplitudes and increases its potential in more complex synthetic networks.

2.5.3 Amplified Negative Feedback Oscillators

The Goodwin oscillator and the repressilator are both formed using repressive links, however network topologies can be formed where genes can activate one another, where one gene promotes (amplifies) its own transcription through positive self-feedback loop as well as activating another gene. The second gene also represses the first gene forming a negative feedback loop (Figure 2.4). There are various different topologies that can utilise this amplified negative feedback loop including repression by transcriptional control (Guantes and Poyatos, 2006), repression through dimerization (Hilborn and Erwin, 2008) and repression by proteolysis (protein breakdown) (Guantes and Poyatos, 2006; Conrad *et al.*, 2008).

2.5.3.1 Repression by Transcriptional Control

These networks have been simulated using both ODEs and the Gillespie algorithm by Atkinson *et al.* (2003), Guantes and Poyatos (2006) and Conrad *et al.* (2008). Simulations from Guantes and Poyatos (2006) showed that oscillations occurring from a saddle-node bifurcation (SNIC) have extended regions with high activator and repressor concentrations, which therefore provide longer periods. Immediately after the bifurcation it was also shown that an increase in the ratio of activator and repressor degradation rates causes the period to decrease and plateau. The work by Atkinson *et al.* (2003) showed the only type of oscillator with amplified negative feedback which can be implemented *in vivo* (Figure 2.4b). The system shows damped oscillations in a system after Hopf bifurcation with a cell doubling time of 2 hours, which is considerably longer than the Repressilator described previously. The

network was implemented *in vivo* and consisted of the activator, NRI, and the repressor, LacI, and was achieved by fusing *glnG*, which encodes NRI, to a control region based on a *glnA* promotor. The design intended to take advantage of DNA looping; during activation phosphorylated NRI (NRIp) interacts with RNA polymerase via a DNA loop, while during repression, LacI bound to the operators also forms a loop, thereby ensuring stable repression. The two loops are antagonistic and formation is mutually exclusive. Results from single cell observations showed initially that three damped oscillations were present, which also exhibited similar amplitude and frequency as simulated results. This is promising for the case of having systems with repression via transcriptional control, however further analysis into the applicability of the model in single-cells is required as cells have weak stochastic coherence with the model.

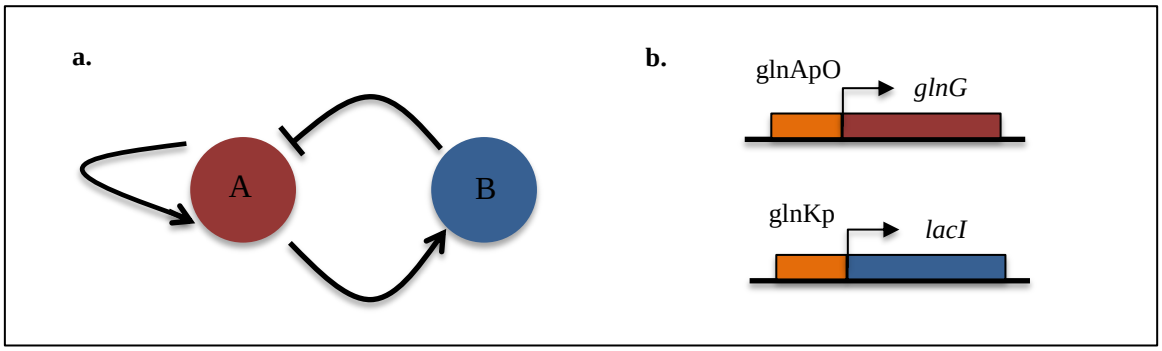

Figure 2.4: a. Amplified negative feedback topology, with repression by transcriptional control. Gene A activates its own transcription as well as gene B, whilst B represses transcription from A. **b.** *in vivo* implementation of repression by transcriptional control (Atkinson *et al.*, 2003).

2.5.3.2 Repression by Dimerization

Proteins can undergo dimerization and this process can repress genetic oscillators through sequestration as proteins from the first gene can dimerize with proteins from the second (Barkai and Leibler, 2000). This network topology (Figure 2.5a) has been modelled using ODEs (Vilar *et al.*, 2002; Hilborn and Erwin, 2008), SDEs (Hilborn and Erwin, 2008) and Gillespie simulations (Barkai and Leibler, 2000; Steuer *et al.*, 2003; Hilborn and Erwin, 2008), but has not been implemented *in vivo*. The extent of the oscillations has been studied and describes the evolution of the repressor and activator-repressor complex. These *in silico* models have been shown to give qualitatively the main features of complete models where oscillations have been shown to exist over broad ranges, which suggest robustness, however only in the presence of intermediate repressor degradation (Vilar *et al.*, 2002). It has been

observed that the repressor degradation plays an important factor into the period of oscillations and has been determined that lower degradation rates increase the period, which is to be expected. Amplitude of the oscillations were sensitive to the transcription and translation rates and the periods were predicted to around 20 hours, which is comparable to the model from Atkinson *et al.* (2003). In this system noise plays a more consequential role and expands the oscillatory parameter region to generate stochastic coherence. The approximate models mentioned are lower order and determine if a system can oscillate, however they can fail to show more subtle features such as stochastic coherence.

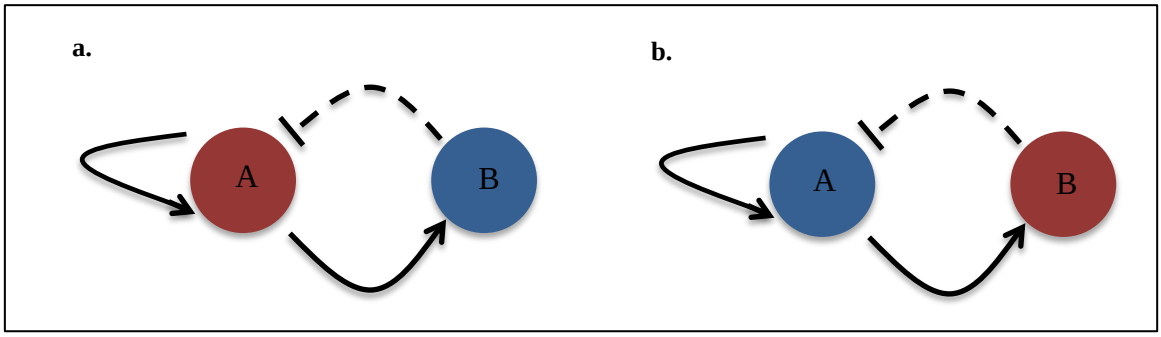

Figure 2.5: a. Amplified negative feedback topology, with repression by dimerization, where red genes show the first gene and blue genes show the second gene in the repression process. Solid lines represent direct transcriptional control and dashed lines represent repression by dimerization. **b.** Amplified negative feedback topology, with repression by proteolysis. Here dashed lines represent repression by proteolysis.

2.5.3.3 Repression by Proteolysis

Another implementation of amplified negative feedback where repression is obtained through degradation of the protein from the first gene by proteases encoded by the second (Figure 2.5b) was researched by Guantes and Poyatos (2006) and Conrad *et al.* (2008), and was simulated using ODEs as well as the Gillespie algorithm, but has again not been implemented *in vivo*. Oscillations in networks reported by Guantes and Poyatos (2006) showed that the activator required significantly faster dynamics than the repressor, and this was achieved through faster degradation and translation rates for the activator. The oscillations exhibited a finite frequency, which increases as the activator dynamics become faster than that of the repressor, and can reach plateaus of greater magnitude than networks with repression by dimerization. Changing the network from nonlinear transcriptional repression via dimerization to linear repression via proteases yields an oscillator with high frequency, low amplitude oscillations and stronger stochastic coherence. This highlights how a small

change in the repression mechanism can have a large effect on the dynamics and overall function.

2.5.4 Synthetic Mammalian Oscillators

A requirement for amplified negative feedback oscillators is that the activator presents faster dynamics than the repressor. This is the same effect as adding a delay in the negative feedback loop as both allow the activator concentration to reach a significant level before repression overtakes, thereby allowing sustained oscillations. Research from Tigges *et al.* (2009, 2010) looked into the effect of having a delay in the negative feedback loop and how it will affect oscillations *in vivo*. These synthetic oscillators are the only ones to be implemented in eukaryotic cells and herald a new form of topology that can host more complex networks (Figure 2.6). The oscillator comprises of two genes with transcription (formation of proteins) occurring from both. The transcription is translated and the resulting protein is fed back into itself and promotes both transcription and the activation of the second gene. The novelty comes from the second gene which activates transcription from the first gene, however it is not translated into a protein, but rather hybridizes with the transcript and represses protein production at translation completing the negative feedback loop. In this sense the protein from the second gene is inhibiting the first gene and stops protein formation as it combines with any product from the first gene. The important step of note is the addition of a delay in the negative feedback loop which causes the repression. Simulations using ODEs showed that the system is sensitive to gene dosage, but remains robust to changes in mRNA and protein degradation. Control is achievable by inhibiting activation by the first protein and allowed for a switch mechanism on the oscillations. Undamped oscillations were present and periods altered depending on the delay in the negative feedback, while cell-cell variability was reported as confirmed by stochastic simulations. Oscillations matched predictions and showed that this network can be suitably tuned to user preference. Lower frequency variants of the synthetic oscillator network (Tigges *et al.*, 2010) were also constructed, which use direct interference of smaller proteins instead of full protein chains. Here the oscillations presented a period of 26 hours, and are considerably longer than any other synthetic network, however this network is not robust as only 18 percent of cells exhibited oscillations. These oscillators do however provide valuable insight into the relationship between gene dosage and dynamics and could potentially allow users to fine tune network outputs.

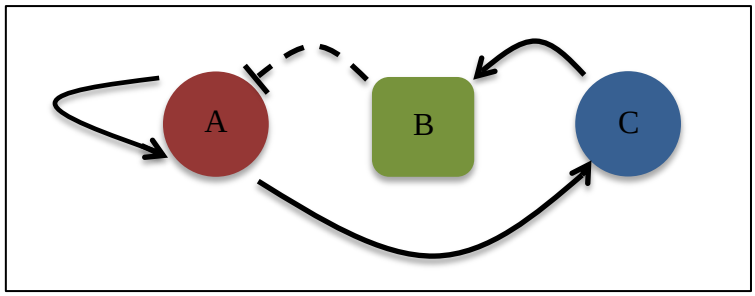

Figure 2.6: Topology of the synthetic mammalian oscillator. Gene A promotes is own transcription, and also activates gene C, which promotes transcription of RNA B (antisense A). RNA B represses A by hybridization and the translational level. Solid lines represent direct transcriptional control, while dashed lines show the repression by sense-antisense hybridization.

2.5.5 Robust Oscillators

The oscillators described so far have not been able to demonstrate robustness, which in synthetic biology is having all cells display the same responses and behaviour, with the percentage of oscillating cells being low, or not reported. However, robustness is vital in synthetic networks especially if they are to become components in larger synthetic systems, or interface with natural systems (Lu *et al.*, 2009). A robust oscillator (Smolen *et al.*, 1998) comprises of two genes. The first, gene A, promotes its own transcription and that of the other gene (gene B). Gene B inhibits itself and the transcription from gene A. It is this self-inhibiting loop on the second gene that differentiates this topology (Figure 2.7) from the amplified negative feedback oscillators. ODE models of this can be found in Smolen *et al.* (1998), Hasty *et al.* (2002) and Stricker *et al.* (2008). Oscillations arise in this network due to the activator degradation rate being two to three times faster than the repressor. The Smolen oscillator (Smolen *et al.*, 1998) is robust and highly tunable, with oscillations ranging over various time periods and as quickly as 13 minutes. The characteristic robustness and tenability provide both reliability and utility, which can be exploited in the construction for future synthetic networks. The rapid oscillatory behaviour appears to occur because of the negative feedback loop added to what is simply an amplified negative feedback topology, and if realised *in* vivo could expand the applications of this oscillator. Work from Stricker *et al.* (2008) showed a more complex network utilising a higher dimensional model that expressed two limit cycles simultaneously, and both the model and *in vivo* implementation showed complimentary results. Results showed that the oscillations were fine-tuned over a

wide range of conditions, with possibility to extend the region by tuning parameters. Over 99 percent of cells displayed oscillations in accordance with simulations, and these findings are a clear demonstration of robustness.

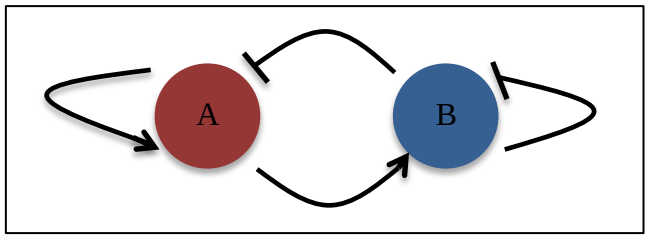

Figure 2.7: Robust oscillator topology (Smolen *et al.*, 1998). Gene A activates its own transcription and that of gene B also, while gene B represses its own transcription and that of gene A.

2.5.6 The Metabolator

The metabolator (Fung *et al.*, 2005) was the first oscillator that incorporates metabolic oscillations within the network topology. It comprises of two genes where one gene produces an enzyme that converts one metabolic pool (M2) to another (M1), and its production is activated by M2. The second gene produces an enzyme that converts M1 into M2 with its production being repressed by M2. As a comparison to the other topologies seen (Figure 2.8a) two genes, A and B, activate one another as well as self-repress by increasing or decreasing the metabolite pool, M2. Within Figure 2.8b, the negative feedback link from gene A to itself via gene C is equivalent to gene A repressing itself in the topology shown in Figure 2.8a. Solid lines represent direct transcriptional control whilst dashed lines ending with a bar show indirect repression. Dashed lines ending with an arrow represent both indirect and direct repression depending on the circuit dynamics. The implementation of this topology in *E. Coli* (Figure 2.8b) was modelled using ODEs, a detailed explanation of which will be presented in Chapter 4. Oscillations were observed in 60 percent of cells with a period of around 45 minutes and lasts approximately 4 hours. It is important to note that this network showed cell division to be un-correlated with oscillatory behaviour, which indicates that the dynamics are de-coupled from the cell cycle. The *in vivo* observations show that the dynamics of this network can be predictably controlled using external metabolite sources. For an in depth analysis of the metabolator the reader is referred to Chapter 4 of this thesis.

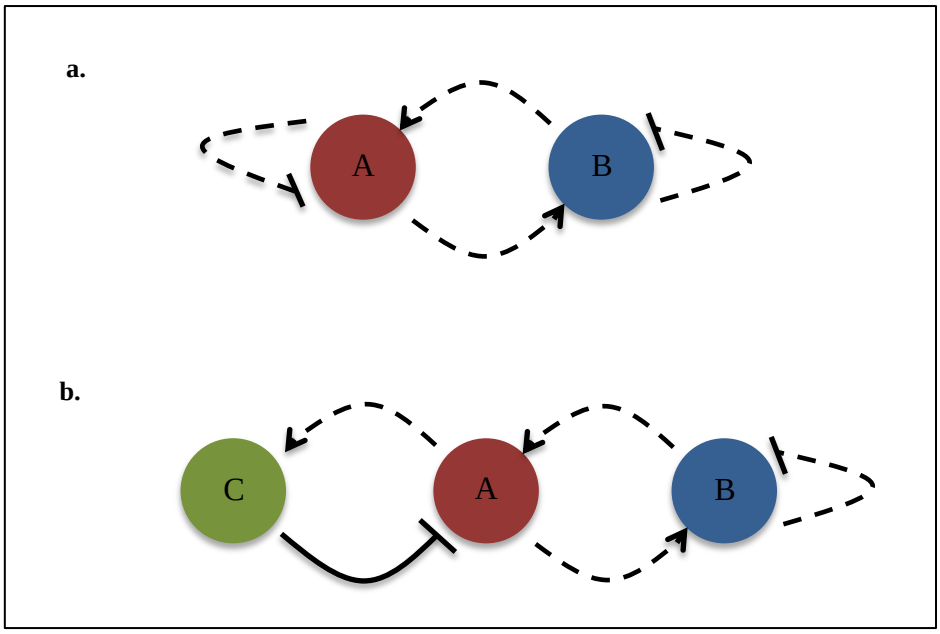

Figure 2.8: **a.** Conceptual schematic topology of the metabolator network. Gene A represses itself whilst also promoting gene B, which in turn can repress itself as well as activate gene A. **b.** Implementation topology of the metabolator.

2.5.7 Summary of Synthetic Oscillators

Section 2.5 has dealt with the plethora of synthetic genetic networks that have been reported in literature. The evolution of these networks from the Goodwin oscillator to more recent Smolen oscillators has been discussed and a summary of all oscillators mentioned is shown in Table 2.1. Whilst it would be expected that one could compare the oscillators that have been published, there currently does not exist a characterisation standard that can be utilised to do so. This is because of the different networks and how they function. Some networks self-repress, whilst others have genes in sequence that supress one another. It is difficult therefore to characterise these networks using conventional techniques, and so comparison is difficult. Furthermore, reported characteristics can vary significantly between oscillators. For instance the number of oscillating cells, which is a good indicator of robustness, is only reported in half of the *in vivo* implementations of the different topologies discussed in section 2.5. With the exception of mammalian oscillator (Tigges *et al.*, 2009) which was implemented in eukaryotic cells, the remainder are constructed within prokaryotes. Of those reported, the prokaryotic systems are easy to manipulate and the mechanisms of gene regulation are simpler than that of the mammalian oscillator. This simplicity is key to allowing the system to be predictable as the essential features of the system to be easily captured. Oscillatory periods vary

significantly from 13 minutes (Smolen *et al.*, 1998) to 26 hours (Tigges *et al.*, 2010), and the reasons for this is unclear, however, the systems are able to be tuned with different factors than can affect system output e.g. temperature, small molecule concentration and plasmid dosage.

Modelling of these networks has been predominantly predictive and often in agreement with *in vivo* results, with model parameters determined through heuristics. Simpler models are used to determine the potential for oscillations, whereas more complex models are developed to obtain more comprehensive qualitative and quantitative predictions of the network behaviour. Detailed models have shown advantages in determining complex behaviour, as seen in the case of the Smolen oscillator (Smolen *et al.*, 1998) where the co-existence of limit cycles was observed. However, a key point is to consider that complex models should not detract from the fact that these behaviours have not been realised *in vivo*, and it is important to test if they are an artefact of increased model detail rather than due to the dynamics of the network.

Robustness is still a significant issue for synthetic oscillators, even in light of the robustness demonstrated by the Smolen oscillator (Smolen *et al.*, 1998). Although many of the networks are de-coupled from the cell cycle, there are networks that exist that have not yet been fully characterised. There is potential for interference from the host cell, which could explain the relatively low numbers of oscillating cells observed. The case of the Repressilator (Elowitz and Leibler, 2000) demonstrates this theory as the oscillations within the three gene network were halted by the system going into the stationary phase (where the growth rate and death rate of the cells is equal). Most of the networks also utilise mutated genes, or gene topologies where one has been removed, to avoid interference, however as the topologies become larger the number of interfering components also increases. Predicting how a network will function if subjected to high level interference from host cells is paramount to the success of synthetic oscillators. For the Repressilator, work by Goh *et al.* (2008) looked at addressing this issue. They found that non-specific interactions forming coherent couplings are more likely to maintain oscillations and non-specific interactions that have a regulatory affect are also likely to maintain oscillations. The effect of cell cycle such as changes in volume and levels of host cells for network function are also being considered (Tuttle *et al.*, 2005; Yoda *et al.*, 2007).

There is also increasing amount of research into coupled genetic oscillators, focussing mainly on repressilators (Ullner *et al.*, 2007; Zhou *et al.*, 2008). These networks have predominantly been studied *in silico*, however recently coupling has been used *in vivo* to give cells that can display synchronised oscillations (Danino *et al.*, 2010). It is envisaged in future work that combinations of these technologies will

aid in the design and implementation of networks that are reliable, tunable and robust, which can then be incorporated into new synthetic genetic regulatory networks.

The networks and oscillators detailed in this section have all shown how synthetic biology aids in designing systems. However, although all of the oscillators mentioned are well characterised, have been studied in the laboratory and have been simulated using mathematical models, there has been little research in terms of controlling the system outputs. The gap in literature regarding control of oscillators and subsequent design is something that will be addressed in this work. It is envisaged that by controlling these systems one can increase production of a useful product, control system oscillation and re-configure the network to provide new products. In terms of the work that will be presented in this thesis, control is used to optimise the system outcomes, and in the case of the Metabolator (Chapter 4) it is used to alter system dynamics to a pseudo-steady-state.

2.6 Summary

This chapter has detailed the various aspects of synthetic biology and how it can be applied for a variety of different problems ranging from grand challenges to simple alternatives to commonly used methods. The next chapter will look at how synthetic biology has been adapted for modelling various biological systems and the methods used to achieve this.

Table 2.1: A summary of the oscillators discussed in section 2.5

Name	Feedback Mechanism	Noise Effect	*in vivo* Implementation	Oscillation Characteristic	Robust Aspects	Tunable Aspects	References
Repressilator	N negative	Amplitude variation, enlarged oscillatory parameter range, stochastic coherence	Prokaryotic	Period: 16%-40 min	Dynamics are decoupled from cell cycle	Dynamics are coupled to cell growth	Elowitz and Leibler (2000)
Goodwin	1 negative	Enlarged oscillatory parameter region, irregular oscillations	Prokaryotic	Period: ˜ 30 min	Period is resistant to IPTG	N/A	Stricker *et al.* (2008), Muller *et al.* (2006)
Amplified negative feedback transcription	2 positive, 1 negative	Weak stochastic coherence, damped oscillations	Prokaryotic	Period: 10/20 hours	N/A	Period and amplitude via cell doubling	Atkinson *et al.* (2003), Guantes and Poyatos (2006)
Robust	2 positive, 2 negative	Bi-modal oscillations, stochastic coherence	Prokaryotic	Period: 13-58 min	Decoupled from doubling time and cell cycle	Period by controlling IPTG, arabinose and temperature	Stricker *et al.* (2008)
Metabolator	3 positive, 2 negative	Amplitude variation	Prokaryotic	Period: 45%-10 min. Lasted ˜ 4 hours	Decoupled from cell cycle	Oscillations can be switched on/off by influx or efflux rates	Fung *et al.* (2005)
Mammalian	3 positive, 1 negative	Cell-cell variability at low gene dosage, additional oscillations at high dosage	Eukaryotic	Period: 17%-71 min	N/A	Period and amplitude through changes in gene dose	Tigges *et al.* (2009)
Low-frequency mammalian	2 positive, 1 negative	N/A	Eukaryotic	Period: 26%-8.5 hours	Period is insensitive to relative plasmid dosage	N/A	Tigges *et al.* (2010)

3. Solution Strategies for Dynamic Models

This chapter will look at the various simulation strategies that can be employed in optimising dynamic models, in particular models that utilise ordinary differential equations (ODEs). Due to the dynamic nature of biological systems it can be increasingly difficult to accurately simulate processes within a mathematical model. Typically systems that are well characterised and have been researched extensively will have their dynamics realised in ordinary differential equations. A brief discussion of how ODEs are formed are shown in section 3.1, leading on to the various NLP and MINLP formulations seen in literature (section 3.2). The remaining sections deal with the formulation of the various models used in this study and are summarised as follows:

1. The ANN approach for solving ODEs is presented in section 3.3. This is used to simulate the processes and provides the basis for nonlinear model predictive control (NLMPC).
2. Using fourth order Runge-Kutta (RK4) to verify the results of the neural network approximator for differential equations (ANN-ODE) formulation is shown in section 3.4, with an example of how the performance of coupling the two methods provides optimal results shown in the example problem (section 3.4.1).
3. OCFE is formulated and presented in section 3.5 and is used to compare results obtained from the ANN-RK4 model to highlight the advantages of using both verification methods.
4. The NLMPC formulation is shown in section 3.6 and how it couples with the ANN-ODE model.
5. Zone NLMPC is formulated and presented in section 3.6.2.

3.1 Ordinary Differential Equations (ODEs)

Ordinary differential equations have been used extensively to model dynamic processes. Within the realm of synthetic biology some choice examples of the use of ODEs have modelled problems regarding drug resistance (Xu *et al.*, 2007) and virus dynamics (Komarova and Wodarz, 2010). The general form of ODE's are presented in equations 3.1-3.3:

$$\frac{dZ_j(t)}{dt} = f_j(Z(t), \theta, t) \qquad j \in J \tag{3.1}$$

$$Z_j(t=0) = Z_j^0 \qquad j \in J \tag{3.2}$$

$$t \in \lfloor 0, t_f \rfloor \tag{3.3}$$

where Z is the J dimensional vector of the state variables in the given ODE system, time points are represented by t and θ is the vector of parameters.

3.2 NLP and MINLP Models

This section will discuss nonlinearities in models that often arise due to the dynamic nature of systems and their solution strategies using nonlinear and mixed integer programming. Whilst they can be modelled using several techniques and can often rely on systems that utilise several techniques in various stages, they are increasingly being modelled using nonlinear techniques. These techniques are of particular importance for biological systems and play an important role in the case studies throughout this thesis. The structure of nonlinear and mixed integer optimisation models are described by Floudas (1995) and take the following form (Equation 3.4):

$$\begin{aligned} & \min_{x,y} f(x,y) \\ \text{s.t.} \quad & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X \subseteq \Re^n \\ & y \in Y \qquad \text{Integer} \end{aligned} \tag{3.4}$$

Where x is a vector of n continuous variables, y is a vector of integer variables, $h(x,y) = 0$ are m equality constraints, $g(x,y) \leq 0$ are p inequality constraints, and $f(x,y)$ is the objective function. This type of formulation can consider a number of optimisation problems through elimination of its elements. If the set of integer variables are empty and the objective function constraints are linear, then this becomes a linear programming problem. If however the set of integer variables are not empty and the objective function contains nonlinear terms and constraints then an MINLP problem is formed. If the objective function contains nonlinear terms as

well as or in lieu of nonlinear terms in the constraints, and the integer variables are empty, then an NLP problem is formulated. Nonlinear programming (NLP) is used often with ODEs to construct models, as presented by Tamimi and Li (2010). This research looked at combining a collocation method with a multiple shooting method to create a nonlinear model predictor control (NLMPC) of fast systems. The multiple shooting method is used to discretely analyse the dynamic model, and the optimal control problem is transformed to an NLP problem.

3.2.1 Review of MINLP Applications and Methods

Mixed integer nonlinear programming (MINLP) has provided solution techniques for many different types of models. It provides a way of modelling dynamic processes and can been used to model proteins and enzymes. Protein modelling has gained interest in recent years and research from Lienqueo *et al.* (2009) looked at how computer-aided design (CAD) could be used to select polypeptide tags that can be used for recombinant protein purification processes. The aim was to select the best polypeptide tag from a comprehensive list of commonly used tags, and in doing so also to maximise the purification process profit. The model used was a simplification of the model proposed by Simeonidis *et al.* (2005). The model and its solution method were implemented in GAMS and the SBB solver was used. It was seen that as the purity level of the protein increased the number of infeasibilities in the GAMS solution also increased. This is expected as the optimisation problem becomes harder to implement and therefore cannot find feasible solutions for every tag.

Solutions for MINLP problems have also been researched in the energy and refinery industries. A genetic algorithm solution method in MATLAB can be useful in optimising hydrogen refinery systems, as demonstrated by Khajehpour *et al.* (2009). The research looked at minimising hydrogen waste into fuel gas within the hydrogen network systems of refineries. The research was based on a superstructure method described by Hallale and Liu (2001). It was seen that this sort of methodology became complex when dealing with large networks and could cause the problem to be unsolved. It was deemed vital to remove some of the improper complexities and simplify the assumptions without losing the accuracy. Reducing the superstructure required the examination of the variables and eliminating those that were unrealistic. This step was crucial in reducing the computational time and ultimately led to faster results. The model was then applied to a case study refinery, and the optimisation of this proved that production of hydrogen reduced by 22.6% thus leading to a saving of approximately $1.19 million annually.

3.3 Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are systems made up of nodes and weighted interconnections, which possess the ability to process information based on the dynamic effects that inputs have on the system (Baughman and Liu, 1995). They can be labelled as surrogates of a process system that can create mapping sequences between input and output datasets obtained from the process. The mapping between the input and outputs is achieved using historical data to train the network, which is utilised to create the architecture required to predict process outputs for any data that is not used as the training set. Due to their nature, ANNs are highly desired as tools for efficiently mapping and capturing nonlinear properties of process systems (Hornik *et al.*, 1989; Pourboghrat *et al.*, 2003). The general format for an ANN structure is presented in Figure 3.1, which shows a typical feedforward ANN. The relationships between the input layer, the hidden layer and the output layer are connected with weighted links and the nodes have biases, both of these aspects represent the system parameters (Prasad and Bequette, 2003). The ANN receives information through the input layer and processes the information to send to the hidden layer. The hidden layer computationally processes the information further and sends this data to the output layer which provides the information of the process system. Generally the nodes in the input layer receive information from the training/validation datasets, whilst the nodes in the hidden and output layers receive information directly from the nodes in the preceding layer of the network.

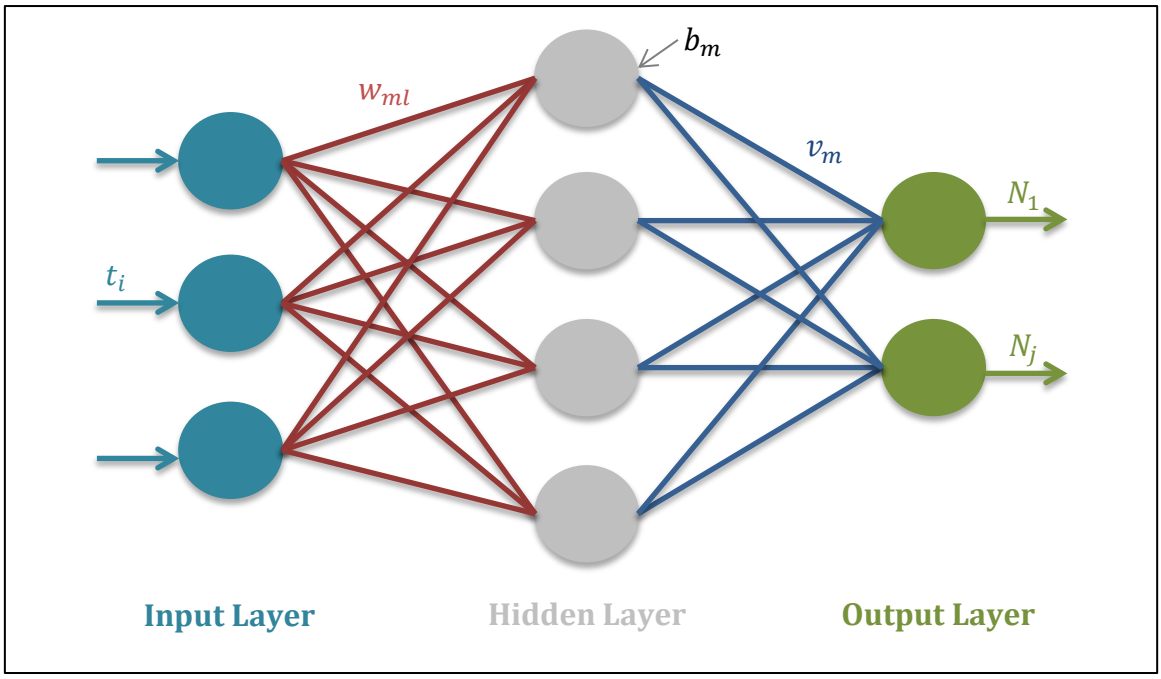

Figure 3.1: Typical structure of a feedforward artificial neural network (ANN)

A typical ANN relies on solving NLP formulations and their form has been described by Dua (2010), given by equations 3.5-3.9:

$$\min_{a,b,w,h,W,B,u} E_1 = \sum_{k=1}^{N_0} (\hat{u}_k - u_k)^2 \tag{3.5}$$

Subject to:

$$a_j^1 = \sum_{i=1}^{N_x} w_{ji}^1 x_i + b_j^1 \quad j = 1, \dots, N_h \tag{3.6}$$

$$h_j^l = \tanh(a_j^l) \quad l = 1, \dots, N_h \tag{3.7}$$

$$a_j^l = \sum_{i=1}^{N_n} w_{ji}^l \, h_j^{l-1} + b_j^l \quad j = 1, \dots, N_n \quad l = 2, \dots, N_h \tag{3.8}$$

$$u_k = \sum_{i=1}^{N_n} W_{ki} n_j^{Nh} + B_k \quad k = 1, \dots, N_0 \tag{3.9}$$

Here (3.5) represents the ANN prediction error objective function, (3.6) represents the activation variables of the first hidden layer, (3.7) indicates the nonlinear

transformation of the activation variables, (3.8) represents the activation variables of the remaining hidden layers, and finally (3.9) represents the ANN output. Where x_i denotes the input values to the network, $i = 1, \dots, N_x$ shows the number of inputs, N_n indicates the linear combinations of these inputs and gives the activation variables, a_j^1, where N_h is the number of hidden layers. The superscript 1 denotes the index of the first hidden layer. Weights are given by w_{ij} and the biases by b_j. These activation variables are then transformed non-linearly to give h_j^l, which is the output of the hidden layer. This then becomes the input of the next hidden layer. The outputs of the last hidden layer, h_j^{Nh}, are combined to provide the outputs, u_k. The number of nodes in the output layer is given by N_0 and w_{ki}, and B_k are the weights and biases respectively. The desired output for the system is indicated by $\hat{u}_k$ and training of the network can be employed as minimisation of the error function, E_1.

Applications of ANNs have looked at synthesis problems (Basri *et al.*, 2007) which utilise ANN frameworks for estimation capabilities of response surface methods (RSM), in the synthesis of palm-based wax ester. Wax esters are used commonly in the pharmaceutical, cosmetic and lubricant industries due to their excellent wetting agent properties. They can be produced artificially from enzymatic hydrolysis of palm oil. In order to optimise this process RSM was used, which is an effective statistical technique for developing, improving and optimising a complex process (Bas and Boyaci, 2007). RSM is a collection of statistical and mathematical techniques that is used to define relationships between the responses and independent variables. In doing so it can generate a mathematical model that can view the independent variables either alone or in a combination with the process. This method utilises a known optimal structure of the ANN. However ANNs are capable of having various structures as they can contain multiple hidden layers. This can make finding the optimal configuration difficult, however, in work by Dua (2010) a mixed-integer approach (MIPANN) was utilised to find the optimal configuration of a neural network. It was noted by Hornik *et al.* (1989) that a successful network is one that can approximate highly-linear processes through postulating an interconnected network structure. Dua (2010) looked at introducing 0-1 binary variables to indicate the presence or absence of nodes and interconnections. The objective was to minimise the overall structure and in doing so minimise the error between the ANN prediction and desired output. This MIPANN approach is advantageous as it can produce a simplified structure and can eliminate nodes and interconnections that are not required for an optimal solution. This in

turn also reduces the computational effort on the system to find the solution. The simplified model was tested using three examples, all of which provided excellent correlation in data and optimal solutions seen. It was seen that although the process of training using an MIPANN model is more complex than a traditional ANN, the model could be used for on-line processes, where a simpler structure is easier to maintain.

The next section will look at the verification methods that can be employed to verify solutions from ANNs.

3.4 ANN Based Solution of ODEs and Runge-Kutta Verification

Artificial neural networks have also been used to solve ODEs and other types of dynamic problems as seen in research from Fogel *et* al (1995), Ge and Zhang (1999), Prasad and Bequette (2003) and Goh *et al.* (2008). Significant research into utilising ANNs to solve ODEs and partial differential equations (PDEs) for initial and boundary value problems is presented in Lagaris *et al.* (1998). The use of an ANN as a solution technique for ODEs has a number of advantageous features, such as:

- The ANN solution provides a continuous differentiable form that can be utilised in subsequent calculations. The solution therefore can be accessed at any point within the domain (Lagaris *et al.*, 1998; Wojciechowski, 2012).
- The ANN provides a solution that requires little memory space and is compact (Parisi *et al.*, 2003; Caetano *et al.*, 2011).
- The dimensionality of the problem can easily be increased through the addition of new input nodes in the ANN (Wojciechowski, 2012).
- Mesh points or spaced collocation points are not required to find the solution of the ODEs (Wojciechwoski, 2012).

In a paper by Filici (2008) a single-layer ANN is utilised to solve ODEs. This neural approximator, in the form of a feed-forward perceptron (Lagaris *et al.*, 2000), aimed to solve systems of ODEs for the whole interval. The perceptron is a linear combination of differentiable functions and their derivatives with respect to the inputs will therefore be well defined. In the model the initial value problems are solved by means of an ANN. A cost function is extended in the model to obtain an approximate solution to these problems. The model was assessed against other techniques such as the Runge-Kutta-Fehlberg (RKF7) integration method (Fehlberg,

1969). It was seen that in cases where the criterion of validity held up, the model was capable of reducing the approximation errors in both of the sets used to train and test the samples. The iterations showed small error and the trained network successfully approximates the solution of the initial value problem.

Further work by Dua (2011) looked at ANN modelling could be used to approximate parameter estimations of ODEs. Typically parameter estimation of ODEs can take on one of two methods. Either the optimisation problem can be decoupled from the integration of the differential equations, or these equations can be converted into algebraic equations and integrated into the overall problem. However these approaches have the main limitation that one can get trapped into a local optimum and may fail to find the global optimum. Also with integration of ODEs often the whole data set is taken into account. If a data set is quite large then this can put extra strain on computing the solution. As mentioned earlier the use of an ANN will reduce the data set and computational effort. An ANN model was created and tested around a series of sub-problems. The method proposed for decomposition based approach in order to calculate the optimum solution was divided into three stages. An ANN was obtained for the data set, then it was used to form a simplified optimisation problem to obtain the parameter estimates, and finally these estimates were used as initial starting points to calculate the solution.

Within this work the ANN methodology for solving ODEs was utilised. In order for the solution for the ODEs to be verified, a fourth order Runge-Kutta transformation (RK4) was used. Here, this method is used to give approximations to the ODEs, and is used in combination with the ANN framework (Equations 3.5-3.9). A brief description of the RK4 based approach (Lagaris *et* al, 1998) is presented next which is coupled with the work by Dua and Dua (2012) who profound the following trial solution for ODE problems:

$$\varepsilon_1 = \min_{\theta, x(t)} \sum_{i \in I} \sum_{j \in J} \{ \hat{z}_j (t_i) - z_j(t_i) \}^2 \tag{3.10}$$

subject to:

$$\frac{dz_j(t)}{dt} = f_i(z(t), \theta, t) \qquad j \in J \tag{3.11}$$

$$z_j(t = 0) = u_j^0 \qquad j \in J \tag{3.12}$$

$$t \in [0, t_f] \tag{3.13}$$

where z is the J-dimensional vector of the state variables from the ODE system, $\hat{z}_j(t_i)$ is the experimentally observed values of these state variables at time points t_i and θ is the vector of the parameters that are estimated such that the error, ε_1, between the observed and predicted values from the model are minimised. The solution for the ODE model from equations 3.11-3.13 can be given by an ANN (Lagaris *et al.*, 1998). If an ANN is considered with l inputs, a single hidden layer, m nodes within the hidden layer and a linear output, the output from the ANN is given by:

$$N = \sum_m v_m \sigma_m \tag{3.14}$$

$$\sigma_m = \frac{1}{1 + e^{-a_m}} \tag{3.15}$$

$$a_m = \sum_l w_{ml} x_l + b_m \tag{3.16}$$

where w_{ml} is the weight from the input l to the hidden node m, v_m is the weight from the hidden node m to the output of the network, b_m represents the bias and σ_m is the sigmoid transformation. The kth derivative of the output with respect to the lth input is given by:

$$\frac{\delta^k N}{\delta x_l^k} = \sum_m v_m w_{ml}^k \sigma_m^{(k)} \tag{3.17}$$

where $\sigma_m^{(k)}$ represents the kth derivative of the sigmoid. For the ODE model given in equations 3.11-3.13, $k = 1$ and the inputs x_l are given at time points t. A trial solution for the ODE model is given by:

$$z_j^{ANN} = z_j^0 + tN_j \tag{3.18}$$

where the ANN model (N_j) is considered for each trial solution z_j^{ANN}. In order to satisfy the initial conditions (3.11), the trial solution (3.18), is constructed as follows:

$$\frac{dz_j^{ANN}}{dt} = N_j + t\frac{dN_j}{dt} \tag{3.19}$$

where $(dN_j)/dt$ is given by equation 3.17 where $x = t$ and $k = 1$. This therefore means that the solution for the ODE model (3.11-3.13) for given values of θ can be formulated as the following nonlinear programming (NLP) problem (Lagaris *et al.*, 1998):

$$\varepsilon_2 = \min_{z^{ANN},N,\sigma,w,v,a,b} \sum_i \sum_j \left\{ \frac{dz_j^{ANN}(t_i)}{dt} - f_i(z(t_i), \theta, t_i) \right\}^2 \tag{3.20}$$

subject to equations 3.14-3.19, where v is the weight from $m = 1, \ldots, M$ hidden nodes to the j-th output node, a is the activation level computed from Equation 3.16 and b is the nodal bias.

To ensure a robust numerical convergence the fourth order Runge-Kutta (RK4) method verifies the solution of the dynamic equations obtained by using the ANN method. In general an initial value problem is set and then a step-size is chosen. The method utilises four transformations, and in averaging the four increments, greater weight is applied to the midpoint. The weight is chosen such that f is independent of z and in doing so the differential equation is transformed to a simple integral. The generic equations for this method are presented below:

$$z_{k+1} = z_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 \tag{3.21}$$

where k_1, k_2, k_3 and k_4 have the form:

$$k_1 = h \cdot f(t_k, z_k) \tag{3.22}$$

$$k_2 = h \cdot f(t_k + a_1 h, z_k + b_1 k_1)$$

$$k_3 = h \cdot f(t_k + a_2 h, z_k + b_2 k_1 + b_3 k_2)$$

$$k_4 = h \cdot f(t_k + a_3 h, z_k + b_4 k_1 + b_5 k_2 + b_6 k_3)$$

The RK4 method involves using an appropriate step size to converge to the same optimal solution as the ANN framework. This was researched using trial and error and an investigation to find the best step size to use in the model was performed and it was found that smaller step sizes, e.g. 0.001, had greater chance to find the correct solution. The same objectives and simulation conditions are applied to the RK4 and the ANN, therefore allowing the RK4 implementation to verify the results from the ANN. The ANN framework contained 7 nodes in the hidden layer, which was found to be the optimal configuration through previous investigations performed when testing the model on case examples, which are illustrated throughout this thesis. The following section will detail an example of the technique and showcase how the ANN-RK4 framework can be applied to a set of ODEs.

3.4.1 Example ANN Problem

In order to test the ANN framework in its ability to solve ODE's a simple example was chosen from the Mathworks website. A simple nonstiff system describing the motion of a rigid body without external forces was considered. The system is set out with the following ODE's:

$$\frac{dy_1}{dt} = y_2 y_3 \qquad y_1(0) = 0 \tag{3.23}$$

$$\frac{dy_2}{dt} = -y_1 y_3 \qquad y_2(0) = 1 \tag{3.24}$$

$$\frac{dy_3}{dt} = -0.51 y_1 y_2 \qquad y_3(0) = 1 \tag{3.25}$$

This was solved in MATLAB using the 'odeset' command and the results are presented in Figure 3.2:

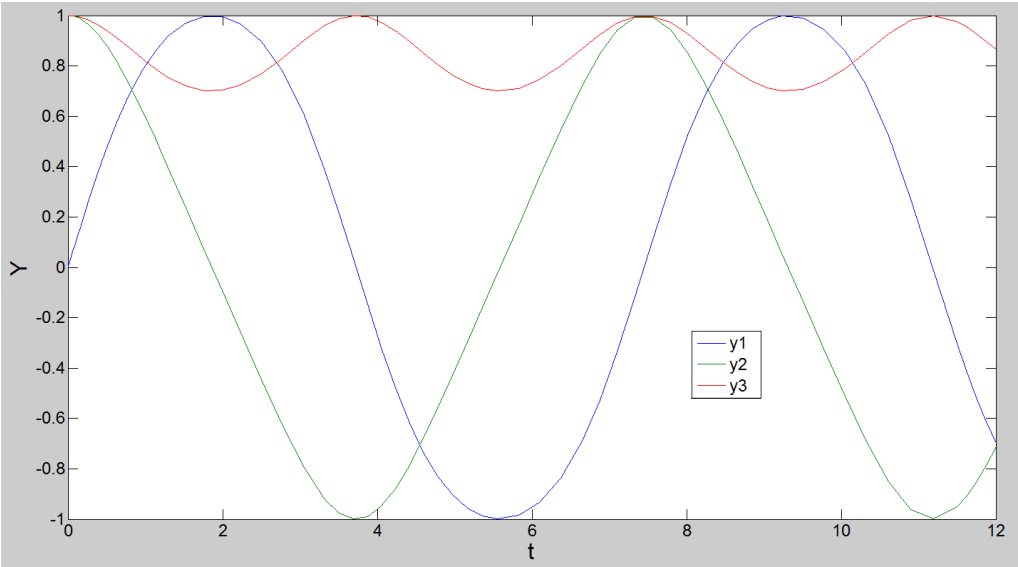

Figure 3.2: The motion of a rigid body without external forces

As can be seen in the graph the system displays a periodic wave motion. This system was then modelled and solved in GAMS using the ANN-RK4 framework and results are seen in Figure 3.3. The results from the GAMS implementation show the same periodic wave motion as the MATLAB results. There is however a slight disagreement, especially when looking at the results of y_3, which starts to decrease in magnitude. Furthermore the periodic wave motion of all the rigid bodies start off being similar, up to t=5, but after this the results differ. This could be attributed to the ANN trying to converge to an optimal solution which involves converging he oscillations of each body. The results do show that the ANN-RK4 framework is capable of modelling ODE systems, and therefore can be used in further studies of dynamic systems.

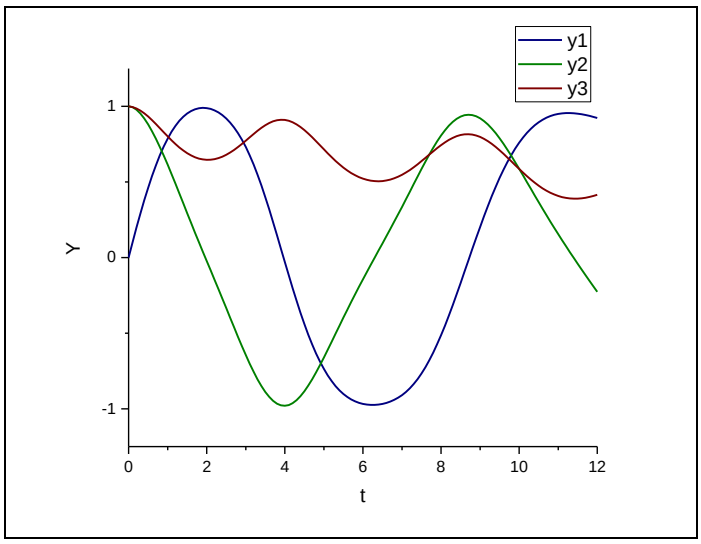

Figure 3.3: The motion of a rigid body without external forces using the ANN framework in GAMS

The following section will detail how ANNs can be embedded into control problems through the use of nonlinear model predictive control (NLMPC), which is used as a framework for several case studies throughout this thesis as a means of controlling dynamic systems. The following section will now discuss another verification method for the ANN framework known as orthogonal collocation on finite elements (OCFE) and discuss how it is integrated with the ANN.

3.5 Orthogonal Collocation on Finite Elements (OCFE)

Orthogonal collation on finite elements (OCFE) is a solution method for problems whose solution has steep gradients and is applicable to time dependent problems. It uses a combination of ODEs and algebraic equations, and therefore can be seen as a more complex solution methodology when compared to others, such as RK4. The method divides the domain into finite elements, and sets the residual to zero at collocation points interior to the elements. Figure 3.4 shows the general outline of OCFE.

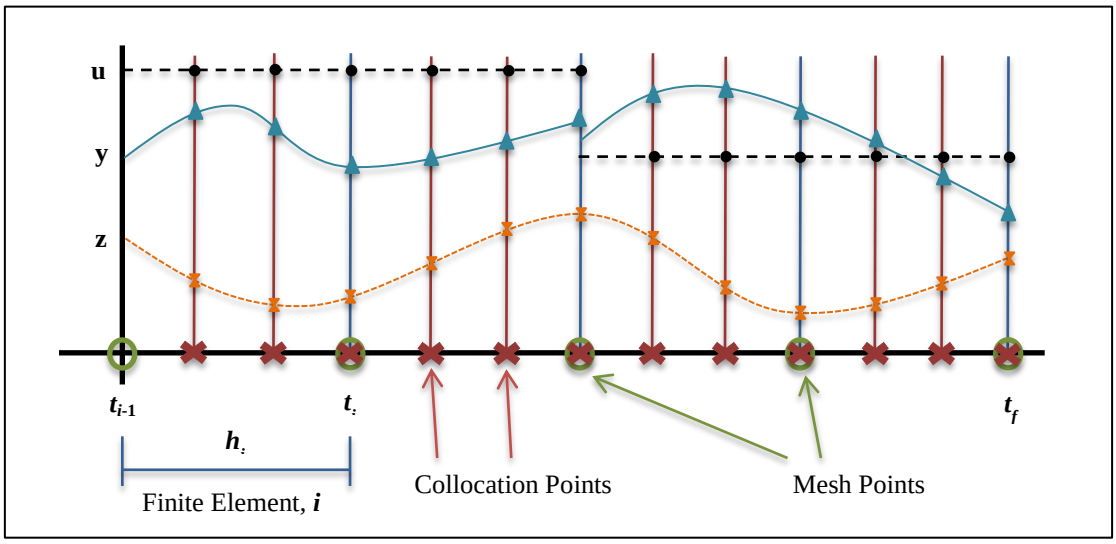

Figure 3.4: Example of OCFE transformation. Here $\boldsymbol{u}$ is the control variable, $\boldsymbol{h_i}$ is the length of the element $\boldsymbol{i}$ and the length of the horizon is defined by the period $\boldsymbol{t_{i-1}}$ to the final point $\boldsymbol{t_f}$. Both $\boldsymbol{y}$ and $\boldsymbol{z}$ are state variables.

The following monomial basis representation of OCFE describes how the optimisation problem can approximate the state and control variables through the finite elements method (Biegler, 2007):

$$Z_i = z_{i-1} + h_i \sum_{q=1}^{K} \Omega_q \left(\frac{t - t_{i-1}}{h_i}\right) \frac{dz}{dt_{i,q}} \tag{3.26}$$

where: z_{i-1} is the value of the differential variable at the beginning of the element i, h_i is the length of the element i, $dz/dt_{i,q}$ is the value of the first derivative in element i and collocation point q and Ω_q is a polynomial of order K which satisfies:

$$\begin{aligned} \Omega_q(0) &= 0 \qquad \text{for } q = 1, \dots, K \\ \Omega_q'(\rho_r) &= \delta_{q,r} \qquad \text{for } q, r = 1, \dots, K \end{aligned} \tag{3.27}$$

where: ρ_r is the location of the rth collocation point within the element. Continuity of the differential profiles is achieved by enforcing:

$$z_i = z_{i-1} + h_i \sum_{q=1}^{K} \Omega_q(1) \frac{dz}{dt_{i,q}} \tag{3.28}$$

In addition to this the algebraic and control variable profiles are approximated using Lagrange representation and take the form of:

$$y(t) = \sum_{q=1}^{K} \psi_q \left(\frac{t - t_{i-1}}{h_i}\right) y_{i,q} \tag{3.29}$$

$$u(t) = \sum_{q=1}^{K} \psi_q \left(\frac{t - t_{i-1}}{h_i}\right) u_{i,q} \tag{3.30}$$

where: $y_{i,q}$ and $u_{i,q}$ indicate the values of the algebraic and control variables respectively in element i at collocation point q, and ψ_q is a Lagrange polynomial of degree K that satisfies:

$$\psi_q(\rho_r) = \delta_{q,r} \qquad \text{for } q, r = 1, \dots, K \tag{3.31}$$

As seen from equation 3.26, the differential variables must be continuous throughout the time horizon, while the algebraic and control variables can be discontinuous at

the boundaries of the elements. It also allows bounds on the differential variables that are enforced at the element boundaries through z_i and can be further enforced at collocation points by including point constraints. This method is best utilised with the Radau collocation points method (Hairer and Wanner, 1999), which is preferred as it allows constraints to be set at the end of each element and allows the system to stabilise efficiently if high level ODEs are present in the model. Investigations in this work that utilise OCFE as a means of comparing against the ANN-RK4 framework (Chapter 5) utilise Radau collocation as the dynamic synthetic biology systems studied are highly nonlinear in nature. Essentially the concentration at the points interior to elements are taken and the tridiagonal system is solved to give the points at all the collocation points. The derivatives at any collocation point can then be evaluated by knowing what element we are in using the orthogonal collocation matrices. In general the solutions become better when more finite elements are used, and this is typical of all numerical methods that use piecewise approximations. The advantages of the finite elements method over tradition orthogonal collocation are that it can be used for smaller time scales as the solution is steep and a global polynomial (in orthogonal collocation) requires many terms to approximate the solution.

Expansion on the collocation method was researched by Chang *et al.* (1979) where the unknown solution is expanded with a series of functions, which are often polynomial, and can include unknown parameters. The expansion is substituted into the differential equations to form the residual, which is set at zero at the collocation points. These equations then provide the parameters for the expansion. The method is used as a comparison to the Galerkin criterion, where the residuals are usually piecewise polynomials defined over small regions (elements) and are zero elsewhere. It is a precursor to the finite elements method, which was later, applied by Carey and Finlayson (1975). The advantage of the collocation method is that the equations are easier to set up and solve when compared with the Galerkin method. The following section will detail an example problem to showcase how the OCFE framework functions with the ANN.

3.5.1 Example OCFE Model

A simple isothermal CSTR (Sistu and Bequette, 1995) was modelled using the ANN-RK4 modelling framework (Dua, 2006) and OCFE with a view to compare the results and analyse which verification method can achieve good results without compromising computational effort. The system is described in equations 3.32-3.34.

$$A \xrightarrow{k_1} B \quad B \xrightarrow{k_2} C \quad 2A \xrightarrow{k_3} D \tag{3.32}$$

$$\frac{dx_A}{dt} = -k_1 x_A - k_3 x_B^2 + (x_{AF} - x_A)u \tag{3.33}$$

$$\frac{dx_B}{dt} = k_1 x_A - k_2 x_B - x_B u \tag{3.34}$$

where: x_A and x_B are the state variables representing the concentrations of A and B resprectively, $x_{AF} = 10$ mol litre^{-1} is the concentration of A in the feed, u is the dilution or feed rate, $k_1 = 50$ hr^{-1}, $k_2 = 100$ hr^{-1} and $k_3 = 10$ litre (mol hr)$^{-1}$. The objective is to maintain $x_B = 1.0$ and this can be achieved for two steady state solutions of the model equations: $\{x_A, x_B, u\} = \{2.5, 1.0, 25\}$ and $\{6.67, 1.0, 233.33\}$, which are the optimum solutions to the problem; the first solution is preferred as it requires a lower dilution rate. The following objective function (Equation 3.35) was analysed by Meadows and Rawlings (1997) for various values of γ_1, γ_2 and γ_3, which are weights in the system, as well as different control and prediction horizons.

$$J = \int_t^{t+T} \gamma_1[x_1(t) - 2.5]^2 + \gamma_2[x_2(t) - 1.0]^2 + \gamma_3[u(t) - 25]^2 dt \tag{3.35}$$

For the example the model equations were discretised using Euler's method and the controller was given a sampling time of 0.002 hrs, a horizon length of 10, γ_1= γ_2=1000 and $\gamma_3 = 1$. The feed rate, u, is given the bounds $0 \leq u \leq 500$. The results for the ANN-RK4 method are presented in Figure 3.5 and the results from the OCFE method are shown in Figure 3.6.

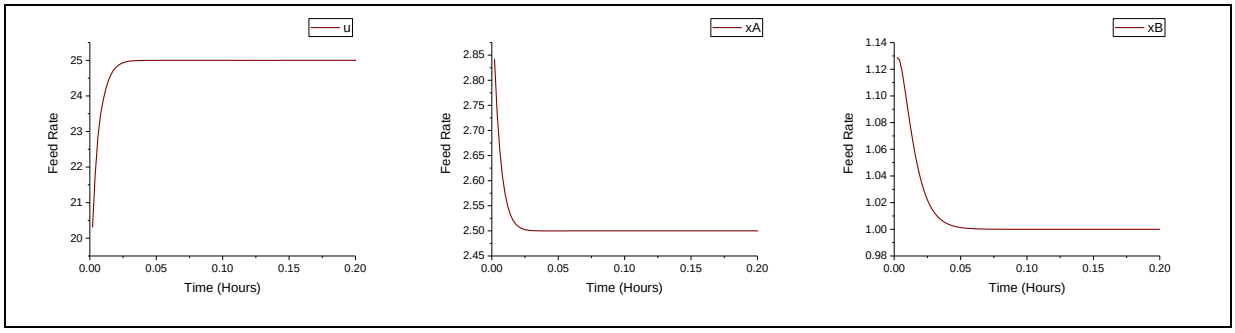

Figure 3.5: Controller performance for the ANN-RK4 model, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$ and $0 \leq u \leq 500$. Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz with a total solution time of 00:33:45 hours for 100 iterations.

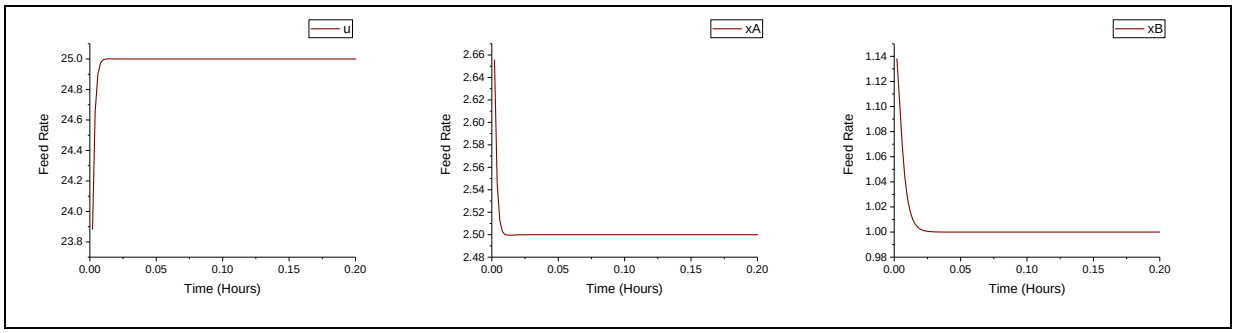

Figure 3.6: Controller performance for the OCFE model, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$ and $0 \leq u \leq 500$. Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz with a total solution time of 00:00:27 hours for 100 iterations.

As shown in Figures 3.5 and 3.6 the OCFE model is able to solve the problem at a faster CPU time, as well as providing results that exactly match the set points. The objective function is minimised to the order of $1x10^{-16}$ which indicates a good convergence for the model. Whilst the ANN-RK4 model is also able to solve the problem, the use of collocation points over the traditional RK4 modelling system here gives a better solution. It is known that OCFE is superior to RK4 as it discretizes at multiple points within the given horizon, therefore the resulting solution is more precise. The trade-off between using either of the verification methods depends mainly on two factors, the CPU time (or computational effort) and the final solution. At times it can be said that a less precise solution can be preferred if the computational effort is significantly lower than the more precise method. This ultimately will save both time and money and hence can be considered to be the optimal method for problem verification. Further analysis of using the ANN/RK4 and OCFE methods will be looked at in chapters 4, 5 and 6 and the advantages of both will be discussed. Further investigation into the dynamic systems studied in

this thesis involved control of the system outputs, and hence a control strategy was employed to achieve this and will be discussed in the following section.

3.6 Model Predictive Control (MPC)

Model predictive control is also referred to as moving horizon control. The optimisation algorithm tries to determine the process dynamics at each control interval by calculating input values that meet the control requirements. The control system feedback is handled by model updates at each time interval, and the iterative process is looped until the defined control objectives are met.

In general the root of many of the algorithms used for control is the receding horizon approach described in the following steps (Garcia *et al.*, 1989):

1. The open loop optimal control problem is solved at the present time step k when $z_{c0} = z_c(k)$. Using past control inputs, the plant outputs are predicted over the discrete time intervals from the present system state to the prediction horizon, H_p, i.e. $z_c(k+1)$ to $z_c(k+H_p)$.
2. The predicted value at any point, $z_c(k+m)$, where $m = 1, \ldots, H_p$, depends on the previous control moves and the planned control moves.
3. The planned control moves, $\Delta u(k+n)$, where $n = 1, \ldots, H_{c-1}$, implemented over the control horizon, H_c, are computed by minimisation of the MPC objective function, which is usually quadratic. This function accounts for the deviation of the output variables from the set points and the control steps. Although MPC calculates the vector of planned control moves, only the first control step, $u(k)$, is implemented.
4. The MPC applies a feedback strategy by reducing the error between the measured process variables and the predicted values. These steps are continued throughout the various sampling times and are known as the receding horizon strategy (Figure 3.7).

Linear MPC problems have been the major usage of MPC techniques (Lee and Cooley, 1997; Mayne *et al.*, 2000; Qin and Badgwell, 2003), however, the inherent nonlinearities in chemical processes together with tightly imposed process constraints and economic considerations, mean that linear models are inadequate at defining the

dynamics of such systems. This has given rise to nonlinear model predictive control (NLMPC) algorithms.

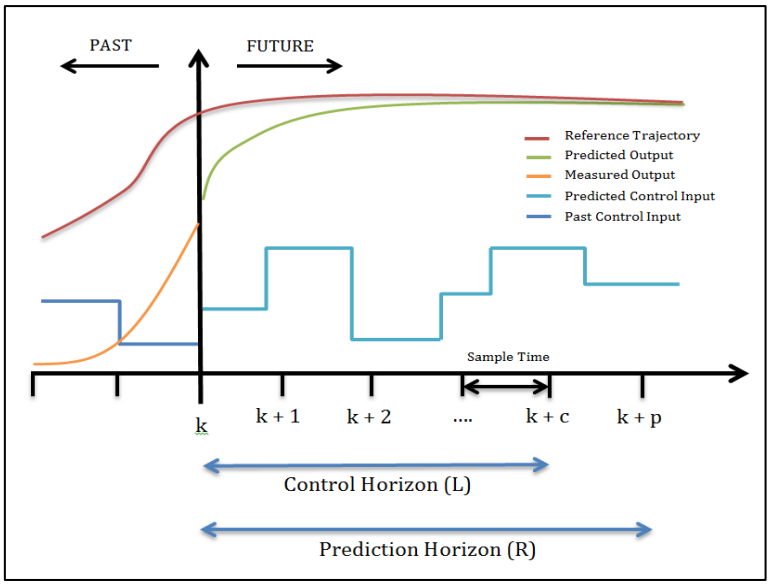

Figure 3.7: Conceptual representation of the NLMPC modelling principal (Bequette, 2009).

Model predictive control (MPC) solution approaches are based on the idea of solving on-line a finite horizon open-loop control problem subject to usual constraints and system dynamics. It operates as an open-loop optimal control strategy (Qin and Badgwell, 1997; Allgower *et* al, 2004) with feedback delivered by a disturbance estimate, which accounts for model uncertainty (Bequette, 1991). There have been various different forms of MPC developed ranging from dynamic matrix control (DMC) (Cutler and Ramaker, 1980) to quadratic DMC (Garcia and Morshedi, 1986), which incorporates all process constraints. Clarke *et al.* (1987) even proposed a generalised predictive control method. A more generic form of control can also be found in Lee and Sullivan (1988), which highlights the benefit of having a generic structure that can infer many different types of control processes.

Work from Mujtaba *et* al (2006) showed how three types of nonlinear control strategies, generic model predictive control (GMC) (Lee and Sullivan, 1988), direct inverse model predictive control (DIC) and internal model predictive control (IMC) could be implemented with neural networks. The work looked at obtaining the optimal reactor temperatures for operability, whilst also tracking these temperatures on-line. The neural networks act as the dynamic estimator for heat release and as the controller for the batch reactor through the implementation of the control techniques. The GMC strategy uses nonlinear models of the process in order to determine the control action. The advantage for the batch reactor is that this

strategy provides feedback control of the rate of change of the control variable, here being the temperature change, and therefore can be used directly as the control variable for the on-line operation. Furthermore the relationship between both the feedforward and feedback control is stated within the GMC algorithm. For the DIC strategy, the inverse model of the process acts as the controller with the process without any feedback. Therefore in this case the neural network has to output the control parameters for the target set point. Finally the IMC strategy has both the forward and inverse models utilised directly as elements in the feedback loop. It is similar to the DIC approach, however the forward model is placed in parallel to the plant and caters for model mismatch, and the error between the plant output and neural network is taken from the set point before being fed back into the inverse model. Robust analysis of these models, where the process or operating parameters were changed indicated that the GMC controller was able to accommodate the changes better than the DIC and IMC controllers. It was deemed the better choice, but it was also mentioned that given *a priori* training for the neural networks that they could also function as robust controllers to cover all possible conditions that the process can undergo.

Further work on MPC strategies for simultaneous control and optimisation of batch processes was researched from Rossi *et al.* (2014). The real-time methodology they developed, the simultaneous model-based dynamic optimisation and control methodology (SMBO&C), solves the dynamic real time optimisation coupled with nonlinear model predictive control (NLMPC), whilst simultaneously re-optimising the same batch time of operations to maximise yield. Firstly the initial number of control variables and the length of each interval are assigned as input data. Starting from a general time point where the process is known to be working the optimal variable profiles and optimal residual operational times are estimated through the optimisation framework. The optimal values of the manipulated variables are implemented and the initial time for the next control move is calculated. The control system stops if the control interval is zero, as this indicates that the optimal operation time has been reached. The overall algorithm is based on the differential and differential-algebraic solvers and optimisers in BzzMath (Buzzi-Ferraris and Manenti, 2012). The SMBO&C algorithm is capable as an optimisation tool, and can drive a discontinuous profile towards profitability, as well as simultaneously controlling the random perturbations entering the controlled system.

3.6.1 Nonlinear Model Predictive Control (NLMPC)

Nonlinear model predictive control (NLMPC) works based on the MPC strategy outlined in the steps shown equations 3.36-3.44. It embeds a nonlinear model for prediction and optimisation, which helps deliver better control performance (Roman *et al.*, 2009). The usage of nonlinear models is beneficial if comprehensive first principle models are obtained as performance can be enhanced. If a comprehensive model is not available then it is advisable that other control strategies should be utilised, as it may be difficult to obtain good nonlinear models using system identification techniques (Allgower *et al.*, 2004). The stability of the closed loop system is a core prerequisite for optimal performance. A system is deemed unstable if the output response does not remain bounded towards the end of the prediction horizon. The NLMPC method for control is given by the general form as follows (Bequette, 1991):

$$\min_{u(k),\ldots,u(k+L-1)} \Phi(u) = \int_{t_k}^{t_k+T_p} e^2 dt = \sum_{i=k+1}^{k+R} (x_i(t_k) - \hat{x}_i(t_k))^2 \tag{3.36}$$

Subject to:

$$\frac{dz}{dt} = f(z,u,p) \tag{3.37}$$

$$y_m = g(x) \tag{3.38}$$

$$u_{min} \leq u(i) \leq u_{max} \tag{3.39}$$

$$u(i-1) - \Delta u_{max} \leq u(i) \leq u(i-1) + \Delta u_{max} \tag{3.40}$$

$$u(i) = u(k+L-1) \qquad \text{for all} \quad i > k+L-1 \tag{3.41}$$

$$x_{min} \leq x(t) \leq x_{max} \tag{3.42}$$

$$y_{min} \leq y(t) \leq y_{max} \tag{3.43}$$

$$x(k) = x_k \tag{3.44}$$

The objective function (3.36) is the sum of the squares of the residuals between the model predicted outputs ($x(t)$) and the set point values ($\hat{x}(t)$) over the prediction horizon of R time steps. The optimisation decision variables are the control actions L

time steps into the future, and it is assumed that after the Lth time step that the control action is constant (3.41). Equations (3.39) and (3.40) relate to the absolute and velocity constraints on the manipulated variables. State and output variables are included in equations (3.42) and (3.43). The dynamic model constraints in equation (3.37) are difficult to determine, and so proper initial conditions for the state variables at the beginning of the prediction horizon must be chosen (3.44). The model outputs (y_m) are a function of the state variables in equation (3.38); however a correction must be applied to y_m to obtain a better prediction of the outputs, $\hat{x}_i$ (Bequette, 1991). In this work, the ANN framework (Section 3.3, Equations 3.10-3.20) is then embedded into the NLMPC formulation and the resulting equations are presented in Equations 3.45-3.54.

$$\min E_{MPC} = \sum_{k=0}^{H_p} Q_z(z_c^{pred}(k) - z_c^{sp}(k))^2 + \sum_{k=0}^{H_c} Q_u(u(k) - u^{sp}(k))^2 \tag{3.45}$$

$$\left\{ E_{ODE} = \sum_i \sum_j \{ \frac{dz_j^{ANN}(t_i)}{dt} - f_j(z(t_i), u, \theta, t\}^2 \right\} \leq \xi_{ANN} \tag{3.46}$$

$$N_j = \sum_m v_m \sigma_m \qquad \forall_j = 1, \dots, J \tag{3.47}$$

$$\sigma_m = \frac{1}{1 + e^{-a_m}} \tag{3.48}$$

$$a_m = \sum_l w_{ml} t_l + b_m \tag{3.49}$$

$$z_j^{ANN} = z_j^0 + tN_j \tag{3.50}$$

$$\frac{dz_j^{ANN}}{dt} = N_j + t\frac{dN_j}{dt} \tag{3.51}$$

$$\frac{dN_j}{dt} = \sum_m v_m w_m \sigma_m \tag{3.52}$$

$$z^{Lo} \leq z(k) \leq z^{Up} \tag{3.53}$$

$$u^{Lo} \leq u(k) \leq u^{Up} \tag{3.54}$$

Where N_j = ANN model and z_j^{ANN} = trial solution, H_p = prediction horizon, z_c = controlled variables (in the metabolator examples shown, Figures 4.7-4.10, this is AcCoA), H_c = control horizon (the same as L in equation 3.41), u = manipulated variables (in the examples for the metabolator, Figures 4.7-4.10, this is Vgly) and s_p = set points (the values that the system will track). Equation 3.45 relates to the tolerance (ξ) of the ANN and aims to reduce the error in ODE calculation. From the ANN, z = the J dimensional vector of the state variables in the ODE system, θ = vector of the given parameters, w_{ml} = weight from input l to hidden node m, v_m = weight from hidden node to output, b_m = bias and σ = sigmoid transformation. The concept of NLMPC is graphically shown in Figure 3.7. Equation 3.36 explains how control is imposed on the system, and this is extended to both the manipulated and control variables in equation 3.45, where each entity is given a set point to track, and the solution is the least squared error between the observed values and the set points. Having set points imposed on the control variable (as shown in equation 3.45, and illustrated in chapter 4, figures 4.10 and 4.16) can help to restrict variation in the control values, however, it is not required and the control variable can act freely within the system.

There are several methods that can be used in order to handle ODE equality constraints with a constrained nonlinear optimisation program (Bequette, 2009):

I. *Sequential solution* – Iteratively solving the ODE's as an 'inner loop' to evaluate the objective function.

II. *Simultaneous solution* – Transforming the ODE's into algebraic equations which are then solved as nonlinear equality constraints in the optimisation.

III. *Intermediate solution* – Transforming the ODE's to algebraic equations which are solved as an 'inner loop' in order to evaluate the objective function.

IV. *Linear approximation* – Approximating either by a single linearisation over the prediction horizon, or by linearisation at a number of times steps within the prediction horizon.

The method in this study employs the use of finding the simultaneous solution. This solution technique results in more decision variables since the value of the state variables at each collocation point are included in the decision. This method also does not require the constraints to be satisfied at each iteration; therefore a faster

convergence is achieved to find the optimum. Another form of this method has been used by Cuthrell and Biegler (1987) and is known as sequential quadratic programming (SQP). The advantage of this system is that the state-variable constraints are easily handled; however a disadvantage to this could potentially mean that the convergence can occur at an infeasible point. Patwardhan *et al.* (1990) used NLMPC to evaluate the effect of parameter uncertainty and manipulated-variable of an exothermic CSTR (continuous stirred-tank reactor) and compared their results with state linearisation and linear control. The conclusion of this work was that the NLMPC demonstrated superior performance in the presence of error in the model parameters and the process was brought to the set point without offsetting the steady state. They found that the NLMPC dealt with the constraints of the system in an explicit manner without compromising the quality of the control objective. The concept of NLMPC is presented in Figure 3.8, where the set points for the system are shown in Figure 3.8a and the resulting response of the control variable to achieve the set points is shown in Figure 3.8b. This is representative of the type of control strategy used within this thesis for the dynamic systems studied.

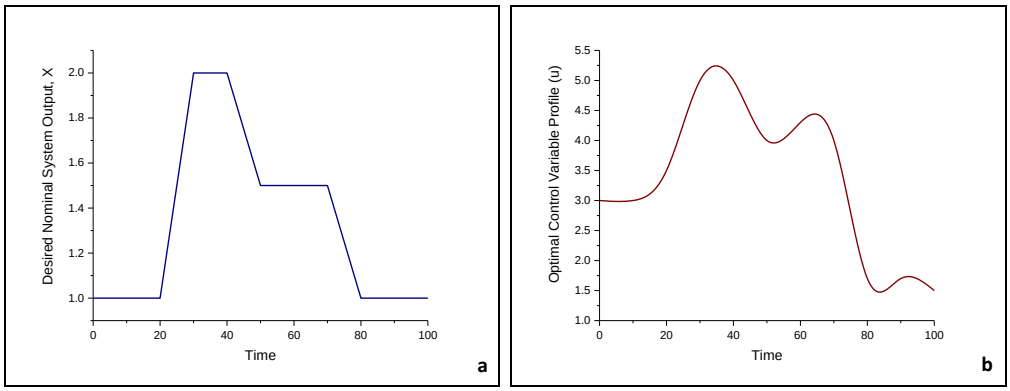

Figure 3.8: Conceptual representation showing the outcome of a deterministic design performance, **a.** desired nominal state (or output) variable profile X (where X = $\hat{\boldsymbol{x}}^{\boldsymbol{N}}$), **b.** computed profile of the optimal control or input variable, u.

The CSTR example presented in section 3.5.1 is re-visited with the aim to add control to the system and fix the outputs. The fixed values for each of the entities are as follows; $x_A = 2.5$, $x_B = 1.0$ and $u = 25.0$. The results for the ANN-RK4 formulation and the OCFE formulation are presented in figures 3.9 and 3.10 respectively. It is noted here that the results for the OCFE formulation (Figure 3.10) show little difference to the set points, and this is because the set point is reached within a single iteration of the model. In order to see the effectiveness of the OCFE when compared to the ANN-RK4 formulation, disturbance was added to the system

in the form of a uniform disturbance over a given range (0.01-1), and added to the RHS of equations 3.33 and 3.34. The results for this are presented in figures 3.11, for the ANN-RK4 formulation, and 3.12 for the OCFE formulation. As shown in figure 3.12 the OCFE formulation is able to track well even under considerable disturbance in the system, however when looking at the control performance when tracking the control variable u and the feed rate of x_B, there is discrepancy between the observed and set data. The formulation is unable to track the set point for x_B and remains close to the desired set, but does not reach it throughout any of the iterations. This is due to the disturbance itself and causes the OCFE formulation to lose the ability to track well. The ANN-RK4 formulation results (Figure 3.11) do not show the same discrepancies, and is still able to track the feed rate of x_A well. The results show that the set points for both x_A and x_B are reached, with discrepancy from the set points only noticed for the control variable u. When comparing both formulations on their ability to track the control variable (Figure 3.13) it is noted that for the OCFE formulation the system does not actually achieve the set point for the control variable, unlike the ANN-RK4 formulation. This is due to the disturbance added to the system being large and the OCFE formulation being unable to reject the disturbance at each finite element. Furthermore there are more fluctuations for the OCFE formulation than the ANN-RK4 formulation. This provides a theory that the ANN-RK4 formulation may provide a more optimal solution method for ODE systems under disturbance. Although the total time taken for each model shows that the ANN-RK4 formulation takes more time, for cases with and without disturbance, it also provides accurate results, especially when disturbance is present. Therefore it is a feasible solution technique and further solidifies this work from a theoretical standpoint.

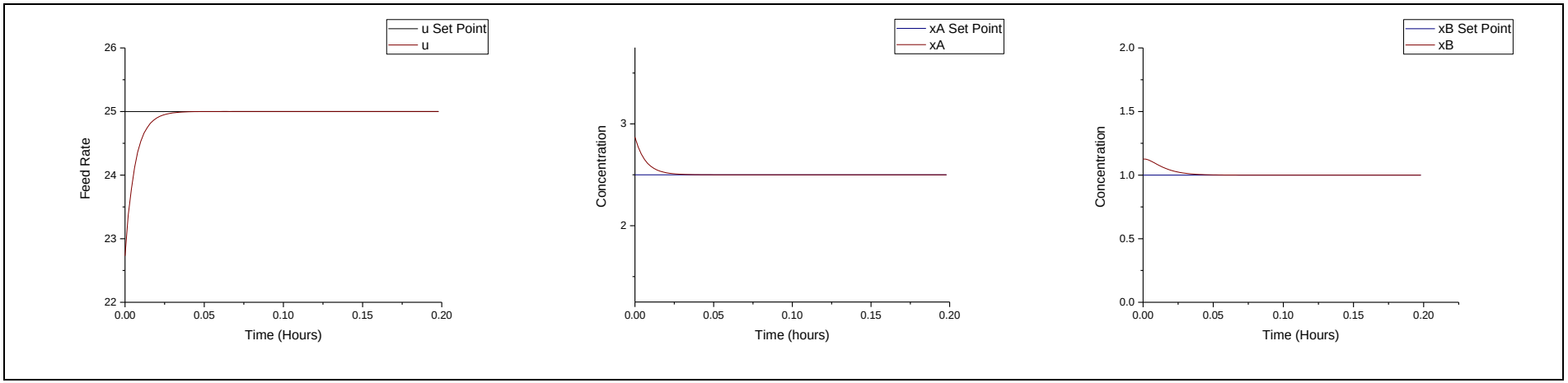

Figure 3.9: Controller performance for the ANN-RK4 model, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$ and $0 \leq u \leq 500$. Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz with a total solution time of 00:05:22 hours for 100 iterations.

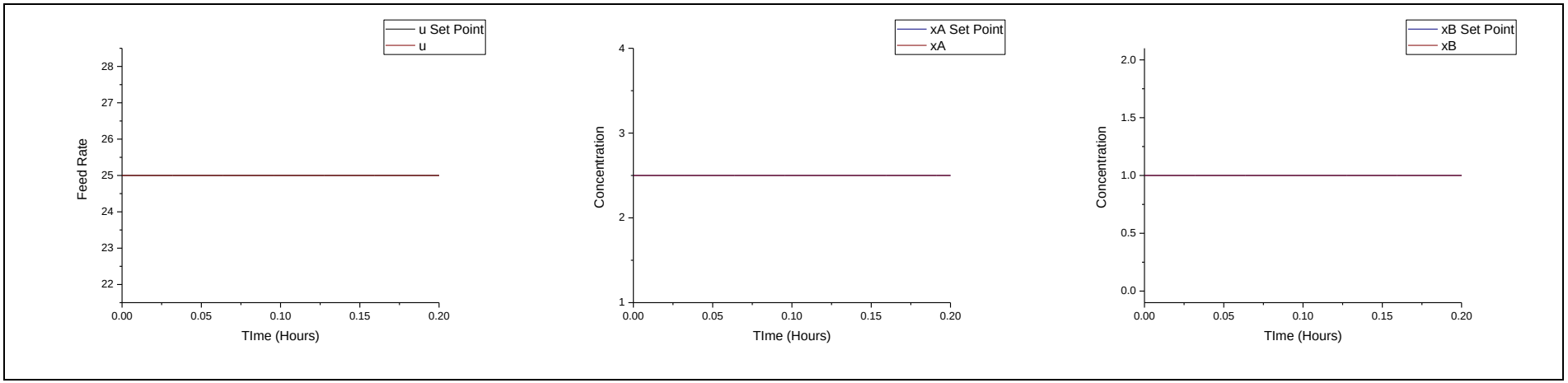

Figure 3.10: Controller performance for the OCFE model, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$ and $0 \leq u \leq 500$. Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz with a total solution time of 00:00:38 hours for 100 iterations.

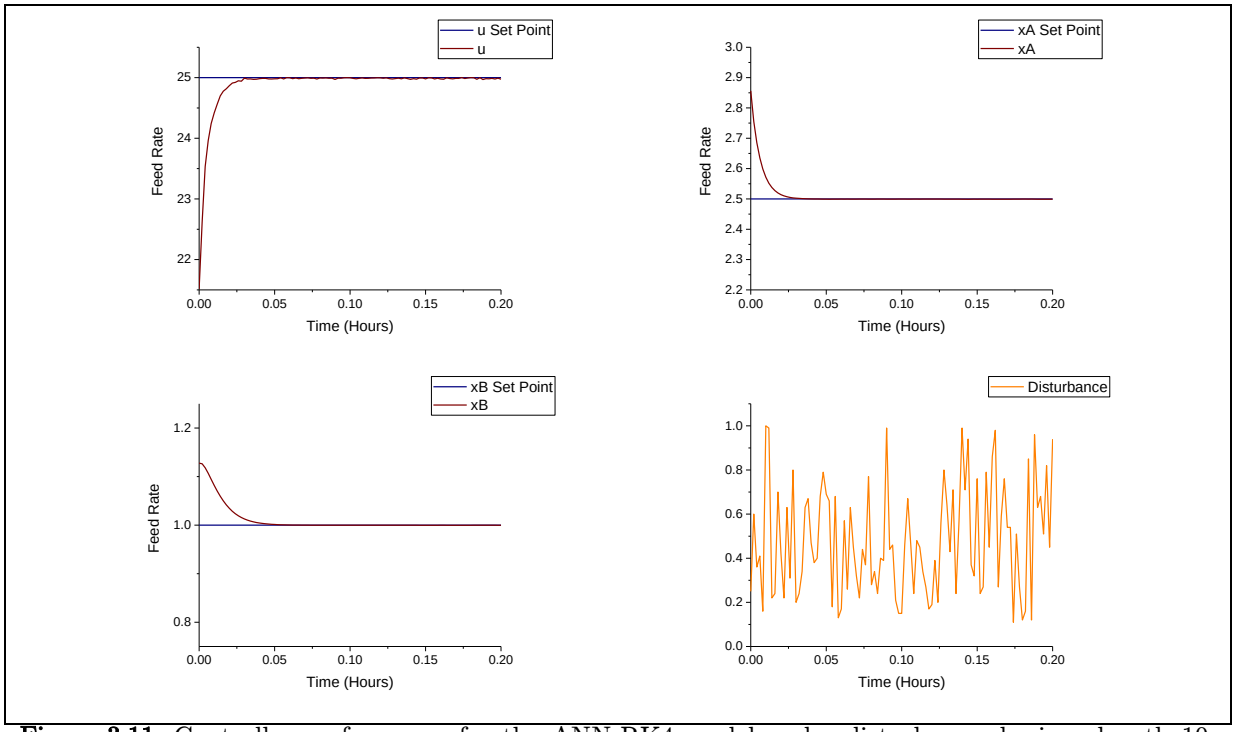

Figure 3.11: Controller performance for the ANN-RK4 model under disturbance, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$, $0 \leq u \leq 500$ and $0.1 \leq d \leq 1.0$ Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz. Total solution time = 00:04:17 hours for 100 iterations.

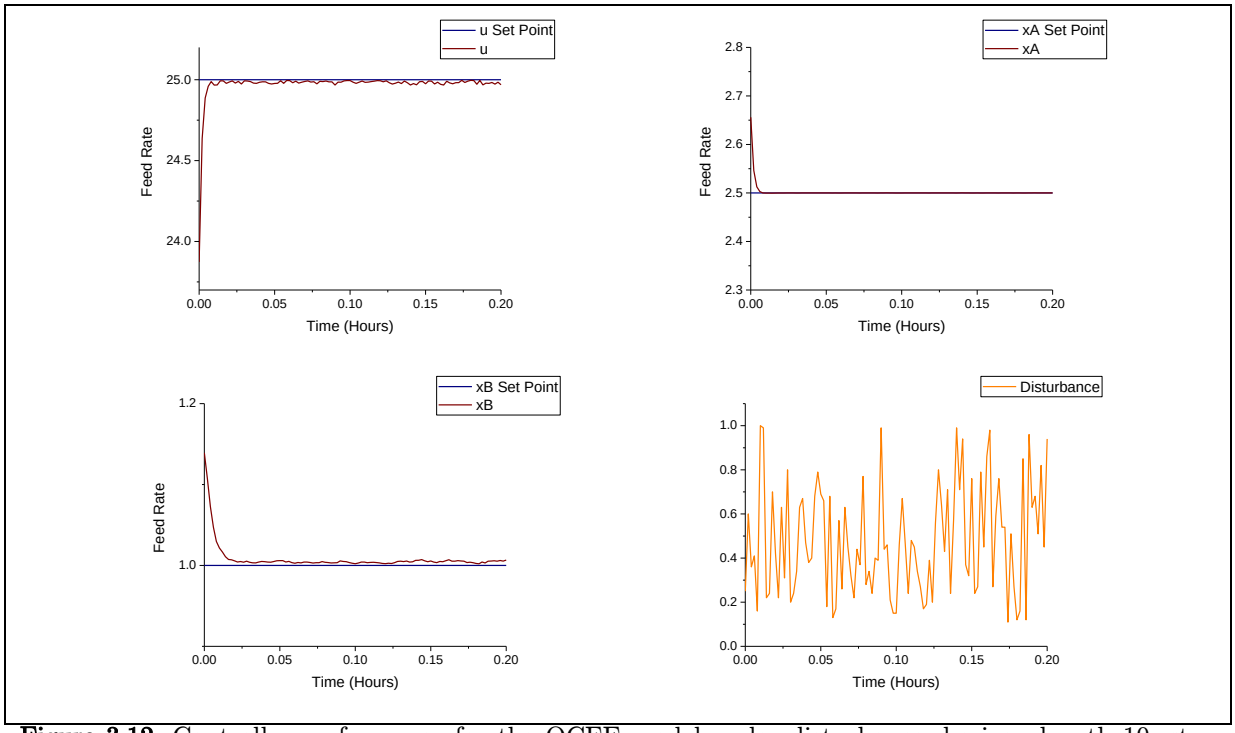

Figure 3.12: Controller performance for the OCFE model under disturbance, horizon length 10, step size 0.002, $\gamma_1 = \gamma_2 = 1000$, $\gamma_3 = 1$, $0 \leq u \leq 500$ and $0.1 \leq d \leq 1.0$. Solved in GAMS using SNOPT solver, Intel Core™2 Duo CPU 2.8 GHz. Total solution time = 00:00:19 hours for 100 iterations.

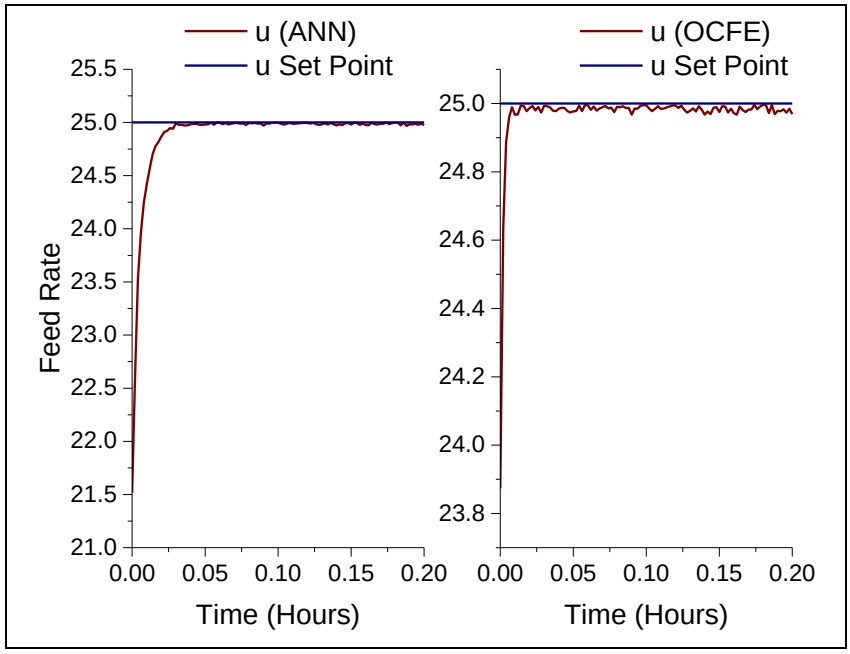

Figure 3.13: Comparison of the ANN-RK4 formulation with the OCFE formulation for the CSTR example under disturbance.

3.6.2 Zone NLMPC

Further strategies linked to NLMPC involve extending the framework to include Zone NLMPC (Grosman *et al.*, 2010; Ferramosca *et al.*, 2010) where the objective function relates to reducing the error between two set points, an upper and lower limit, and the optimal solution residing within this range. This is advantageous as the control is implemented within a range rather than a fixed set point, therefore allowing more flexibility within the system (Ferramosca *et al.*, 2010). The Zone NLMPC strategy is highlighted in equation 3.55, where the NLMPC formulation (equation 3.36) is altered to an NLP optimisation problem for zone control.

$$\min_{u(k),\ldots,u(k+l-1)} \Phi(u) = \sum_{i=k+1}^{k+R} ((y_{sp.H}(i) - y_{pred}(i))^2 + (y_{sp.L}(i) - y_{pred}(i))^2) \tag{3.55}$$

Where $y_{sp.H}$ = set points upper limit and $y_{sp.L}$ = set points lower limit. For the case when uncertain parameters, θ, are present in the system the derived performance is

given by a 'tube' represented by $\pm\Delta n$ deviation from the nominal value x^N, as shown in Figure 3.14, which also shows the concept of Zone NLMPC.

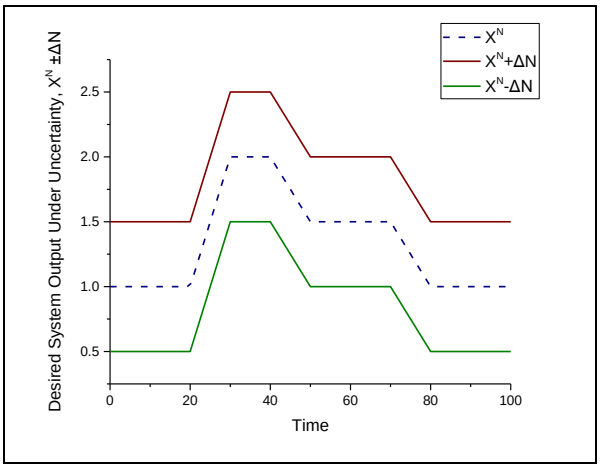

Figure 3.14: Conceptual outcome of a robust design performance under uncertainty

3.6.3 Restricting Velocity of the Control Variable

It is possible to restrict the velocity of the control variable, which allows the output to be more refined and less noisy. This is advantageous if the control variable is used in practice to modify the system outputs, and in terms of a biological system, can greatly affect experimental application of a system. Therefore simulations in this study were trialled with a velocity constraint, (Du), where u can represent the control variable, and is used to constrict the velocity as well as minimise the error between the given velocity and the constrained velocity as seen in equation 3.56.

$$Du = Du(nd) - Du(nd+1) \tag{3.56}$$

3.6.4 Solvers

All NLMPC and Zone NLMPC problems in this study are solved using the Sparse Nonlinear Optimiser (SNOPT) (Gill *et al.*, 2005), CONOPT3 (Drud, 1985) and KNITRO (Byrd *et al.*, 2006) and were solved in GAMS. The solver provides rigorous control of the optimisation sequence to ensure convergence and makes

provision for infeasible sub-problems. They all solve nonlinear problems but have some distinct differences, which will be discussed in this section.

3.6.4.1 SNOPT

SNOPT utilises quadratic approximations at every interval until the solution is found. It does not require specified initial points as the solver will optimise and find the most appropriate starting point through a feasible path method. As such it may require more computational effort and time as it has to search for an appropriate starting point. However, once this is achieved it is capable of solving complex nonlinear models and as will be shown in Chapters 4 and 5, gives accurate results for both the developed ANN framework and OCFE.

3.6.4.2 CONOPT3

CONOPT3 utilises a generalised reduce gradient method, which linearises the nonlinear model and eliminates variables until it finds a solution. It can be deemed faster than SNOPT as it converts the NLP into an LP with each iteration, and generally LP problems are simpler to solve. However, CONOPT3 is highly dependent on initial values, and if these values are not optimal, CONOPT3 can get stuck in local optimal solutions and may not provide the global optimal value. As shown in Chapters 4 and 5, CONOPT3 is capable of solving many of the models in the case studies, however is unable to reject disturbance in some cases.

3.6.4.3 KNITRO

KNITRO utilises yet another method for solving nonlinear models, the interior point method. The NLP is converted into an augmented problem through transformation of the constraints, and it is this augmented function that is solved. KNITRO defines the feasible space and stays within this to find the solution to the augmented problem through minimisation of the augmented objective function. It is capable of solving less complex nonlinear models, however, as shown in Chapter 5 it is unable to deal with complex problems, like the Metabolator, especially in cases where disturbance is added.

3.7 Summary

This chapter has detailed the various methods that can be used to simulate and optimise dynamic models. The artificial neural network framework that has been utilised for case studies within this thesis has been presented and outlined. Furthermore the methods of fourth order Runge-Kutta and orthogonal collocation on finite elements, which were been used to verify the solution of the neural network, have been outlined. Lastly control of dynamic systems has been discussed and the frameworks for nonlinear model predictive control and zone nonlinear model predictive control have been introduced and outlined.

The next chapter in this thesis will look at a prominent case study of the metabolator (Fung *et al.*, 2005) and how the frameworks presented in this chapter have been applied.

4. A Gene Metabolator Case Study

4.1 The Metabolator

In this chapter the first case study that utilises the ANN-NLMPC approach to solve for a dynamic biological system will be introduced. Work by Fung *et al.* (2005) detailed a metabolator, which is a biological process that uses Acetyl Co-Enzyme A (AcCoA) and produces acetate as a product. The metabolator integrates transcriptional regulation into metabolism of *Escherichia Coli* (*E. Coli*). The conceptual design of the metabolator consists of a flux-carrying network with two inter-converting metabolite pools (M_1 and M_2) catalysed by two enzymes (E_1 and E_2), whose expressions are negatively and positively regulated by M_2, (Figure 4.1). A high input metabolic influx into M_1 drives conversion to M_2, and as a result E_1 is expressed over E_2. However, accumulation of M_2 causes a repression of E_1 and upregulates E_2. When the backward reaction rate exceeds the sum of the forward reaction rate and the output rate, M_2 level decreases and M_1 level increases. In this case E_1 is expressed yet again and E_2 is degraded, returning to its first stage. Conversely, if the input of flux is low, M_2 does not accumulate sufficiently fast to cause a large change in gene expression, and a stable steady state can be reached. The design of the metabolator allows metabolic physiology to influence gene expression cycles, which is a characteristic that is typically seen in circadian regulation.

The conceptual design is realised using the acetate pathway in *E. Coli* (Figure 4.2). The M_1 pool is AcCoA and the M_2 pool consists of acetyl phosphate (AcP), acetate (OAc^-) and its protonated form (HOAcE). AcCoA is a metabolic product of fatty acids, sugar and some amino acids, and is a primary component of the tri-carobxylic acid (TCA) cycle. AcCoA is converted to AcP in *E. Coli* by phosphate acetyltransferase (Pta), which corresponds to the enzyme E_1 in Figure 4.1, and then to acetate through the action of acetate kinase (Ack). The protonated form of acetate is able to permeate the cell membrane and can be excreted into the cell environment. The remaining flux goes to produce either acetate or ethanol. In wild type *E. Coli*, the enzyme Acetyl co-enzyme A synthetase (Acs) is induced in the presence of acetate, however, this induction is under catabolic repression of glucose in wild-type strains so as to avoid futile cycling. In their design, Fung *et al.* (2005) utilise Acs as their secondary enzyme, E_2, and re-structure the system to respond to the action of the M_2 pool.

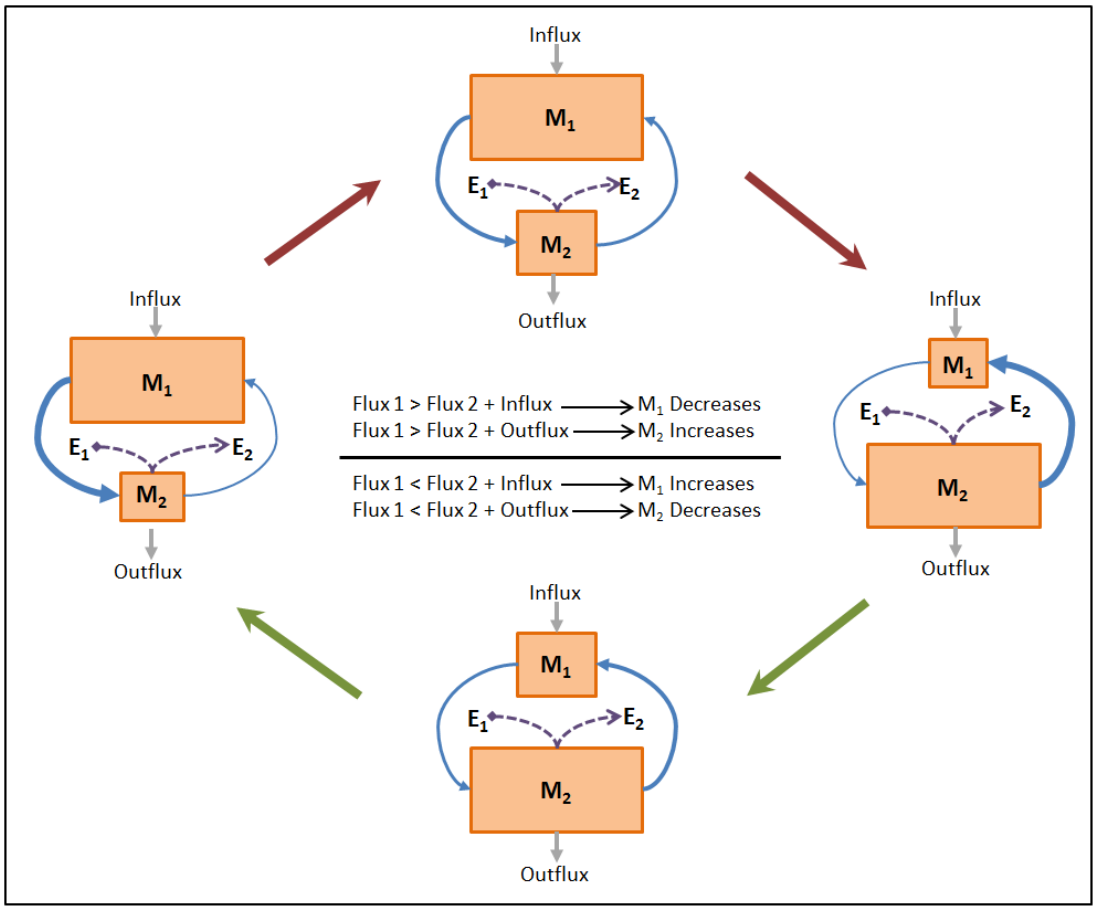

Figure 4.1: Conceptual design of the oscillatory dynamics. Solid lines indicate metabolic flux; dashed lines indicate positive (arrow) and negative (diamond arrow) transcriptional or translational regulation.

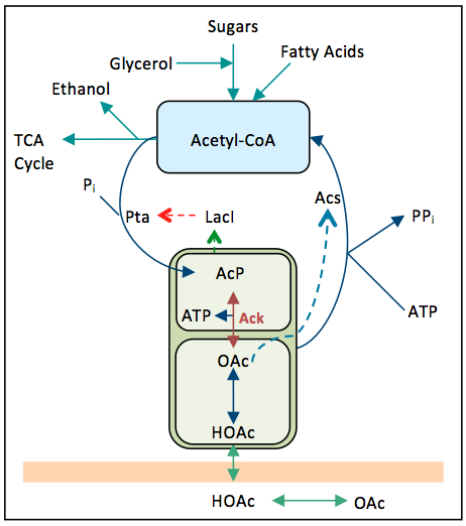

Figure 4.2: *In vivo* realisation of the dynamic system (Figure 4.1). The blue and green boxes indicate the metabolic pools (M1 and M2), and the enzymes, E1 and E2, are represented by Pta and Acs respectively. LacI is a transcriptional factor that is designed to repress the activity of Pta.

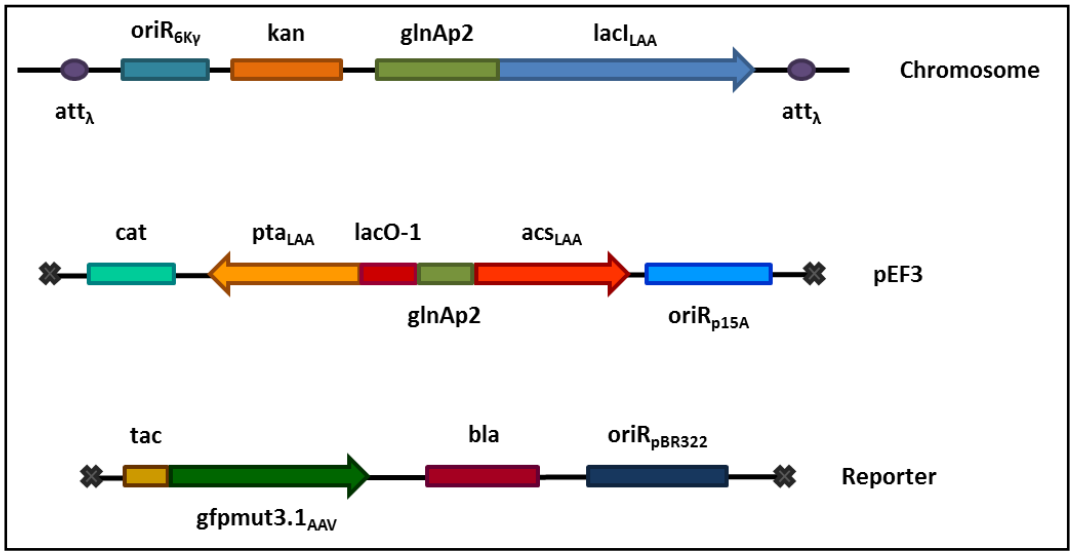

Figure 4.3: *E.Coli* and plasmid constructs used by Fung *et al.* (2005). The lacI gene ($LacI_{LAA}$) under the control of the glnAp2 promoter was inserted into the *E.Coli* chromosome at the lambda attachment site (att_{λ}) using the CRIM method (Haldimann and Wanner, 2001). The pEF3 plasmid expresses low stability pta (pta_{LAA}) and acs (acs_{LAA}), which are controlled by the lac0-1 and glnAp2 promoters respectively. The reporter plasmid expresses an intermediate-stability GFP variant ($gfpmut3.1_{AAV}$) under control of the tac promoter. Crosses at the end of the pEF3 and Reporter indicate that the circular plasmids have been linearised for illustration. The genes for resistance to Kanamycin, Chloramphenicol and Ampicillin are shown by kan, cat and bla respectively.

For the host strain Fung *et al.* (2005) constructed plasmids, which were placed in the *E. Coli*, cultured and eventually experimented with. The two plasmids, pEF3, which expressed Pta and Acs, and the Reporter, which expressed green fluorescence protein (GFP), were both under control of a *tac* promoter. A detailed schematic of the promoters and the chromosome is presented in Figure 4.3.

As part of their research, Fung *et al.* (2005) solved differential equations that represented the dynamics of the system in MATLAB using RK4 (chapter 3, section 3.4). Exact parameter values were determined using linear stability analysis, with specific interest in Hopf bifurcation, which was used to characterise the transition from a steady state solution to a critical periodic state (Guckenheimer and Holmes, 1983; Looss and Joseph, 1989). In the work presented here, the system was modelled in GAMS (General Algebraic Modelling System) with a view to carry out a model-based optimal design. The gene oscillator system model (Fung *et al.*, 2005) is given by the following system of simultaneous ODEs:

$$\frac{dAcCoA}{dt} = \frac{k_2 \cdot Acs \cdot OAc}{Km_2 + OAc} - \frac{k_1 \cdot Pta \cdot AcCoA}{Km_1 + AcCoA} + V_{gly} - (K_{TCA} \cdot AcCoA) \quad (4.1)$$

$$\frac{dAcP}{dt} = \frac{k_1 \cdot Pta \cdot AcCoA}{Km_1 + AcCoA} - (k_{Ack.f} AcP - k_{Ack.r} OAc) \quad (4.2)$$

$$\frac{dOAc}{dt} = \left(k_{Ack.f} AcP - k_{Ack.r} OAc\right) - \left(C\left(OAc \cdot H - k_{eq} HOAc\right)\right) - \frac{k_2 \cdot Acs \cdot OAc}{Km_2 + OAc} \quad (4.3)$$

$$\frac{dHOAc}{dt} = \left(C\left(OAc \cdot H - k_{eq} HOAc\right)\right) - k_3(HOAc - HOAc_E) \quad (4.4)$$

$$\frac{dLacI}{dt} = R_{Lacl} - Rd_{Lacl} \quad (4.5)$$

$$\frac{dPta}{dt} = R_{Pta} - Rd_{Pta} \quad (4.6)$$

$$\frac{dAcs}{dt} = R_{Acs} - Rd_{Acs} \quad (4.7)$$

The differential equations take into account the various factors that govern the rate and flux of the resulting compounds from the system. Equation (4.1) represents the dynamic balance for AcCoA expressed by the sum of Vacs (Acs flux), Vpta (Pta flux), Vgly (Glycolytic flux) and Vtca (TCA flux). Equation (4.2) relates to the dynamic balance equation for AcP, which is calculated by the sum of Vpta and Vack (Ack Flux). Equation (4.3) relates to the dynamic balance for OAc^-, calculated from the result of Vack, Vace (Acid-base equilibrium for acetic acid) and Vacs. Equation (4.4) relates to HOAc, which is calculated using Vace and Vout (HOAcE intercellular transport rate). Equations 4.5-4.7 describe the degradation rates of the proteins within the system where R represents the degradation rates of the proteins and these are defined in table 4.1. The synthesis rates for these entities are governed by the following equations:

$$R_{LacI} = \frac{\alpha_1 (AcP/K_{g1})^n}{1 + (AcP/K_{g1})^n} + \alpha_0 \quad (4.8)$$

$$R_{Acs} = \frac{\alpha_2 (AcP/K_{g2})^n}{1 + (AcP/K_{g2})^n} + \alpha_0 \tag{4.9}$$

$$R_{Pta} = \frac{\alpha_3}{1 + (LacI/K_{g3})^n} + \alpha_0 \tag{4.10}$$

where K_{gx} , n and α_x are constants describing the synthesis and degradation of the moieties. The model parameters for these equations are given in table 4.1. One of the aims of this work is to apply NLMPC to dynamic systems and control their output, and for the metabolator this is possible by using the state variables of the system, which represent the main entities in the two metabolite pools. Both AcCoA and AcP are potential state variables in the system and can be tracked using a fixed path. The control variables for the system are the main entities that affect the state variables and in the metabolator these are glycolytic flux (Vgly) and acetate concentration. The work by Fung *et al.* (2005) simulated the system model and this is used as a starting point for the control. The work presented in this chapter will showcase how design can be achieved for the metabolator by controlling the system outputs. This therefore moves towards utilising synthetic biology techniques to engineer a new system and alter an existing system outcome.

Table 4.1: Model parameters for equations 4.1-4.10 (Fung *et al.*, 2005).

Parameter	Description	Value
Km_2	Michaelis-Menten rate coefficient for Acs flux	0.1
K_2	Rate coefficient for Acs flux	0.8
Km_1	Michaelis-Menten rate coefficient for Pta flux	0.06
K_1	Rate coefficient for Pta flux	80
Ktca	Rate coefficient for AcCoA	10
Kackf	Rate forward reaction converting AcP to OAc^-	1
Kackr	Rate backward reaction converting OAc^- to AcP	1
C	Concentration of the Acid-Base equilibrium	100
H	Concentration of Hydrogen ions (H^+)	10^{-7}
Keq	Molar equivalence of the Acid-Base equilibrium	$10^{-4.5}$
K_3	Rate coefficient for HOAc flux	0.01
HOAcE	Amount of HOAc remaining after flux	0
Vgly	Glycolytic flux	$10^{-4} \leq Vgly \leq 10^4$
α_0	Constant	0
α_1	Constant	0.1
α_2	Constant	2
α_3	Constant	2
K_{g1}	Constant	10
K_{g2}	Constant	10
K_{g3}	Constant	0.001
n	Constant	2
R_{dX} (X = Lacl, Acs, Pta)	Degradation rate of proteins	$R_{dX} = K_dX$ $K_d = 0.06$

4.2 ANN-RK4 Simulation

The system described above in equations 4.1-4.10 was simulated using the ANN-RK4 framework described in Chapter 3, Sections 3.3 and 3.4. Results (Figure 4.4) match well with those achieved from the case paper (Fung *et al.*, 2005), and it is noted that an increase in the glycolytic flux (Vgly, which is a combination of glycerol, fatty acids and sugars as presented in Figure 4.2) results in increased activity and oscillation of the metabolites AcCoA and AcP.

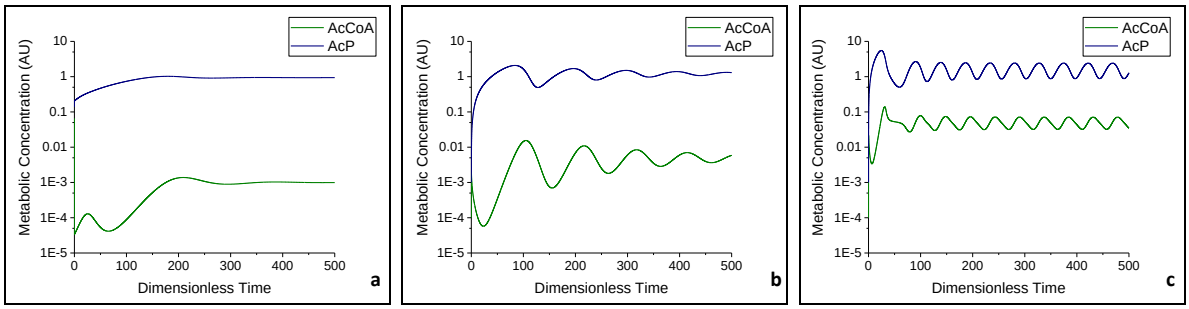

Figure 4.4: Oscillation of the metabolite AcCoA and its co-factor, AcP, against time using the ANN-RK4 method, step size = 0.001, a. Vgly = 0.01, b. Vgly = 0.05, c. Vgly = 0.5.

In Figure 4.4 a loss of activity is seen in the metabolite AcCoA when the glycolytic rate is reduced. The case for when Vgly = 0.001 was also simulated; however the model was unable to converge to find an optimal solution without resorting to negative values for activity. Therefore the step size had to be decreased from 0.01 to 0.001 to allow the model to process lower Vgly values, which also gave results that were analogous to those published. Furthermore, when the acetate concentration was increased there was a distinct breakdown of AcCoA activity, Figure 4.5, and oscillations reach a steady state. This is expected as if acetate is increased there is an increase in the acidity of the environment, which therefore causes breakdown of the metabolator. This matches with the results obtained from Fung *et al.* (2005) as they showed that increasing levels of acetate will cause a loss of activity within the system. As the results from the ANN-RK4 model also agree with this, the model can effectively simulate the system dynamics.

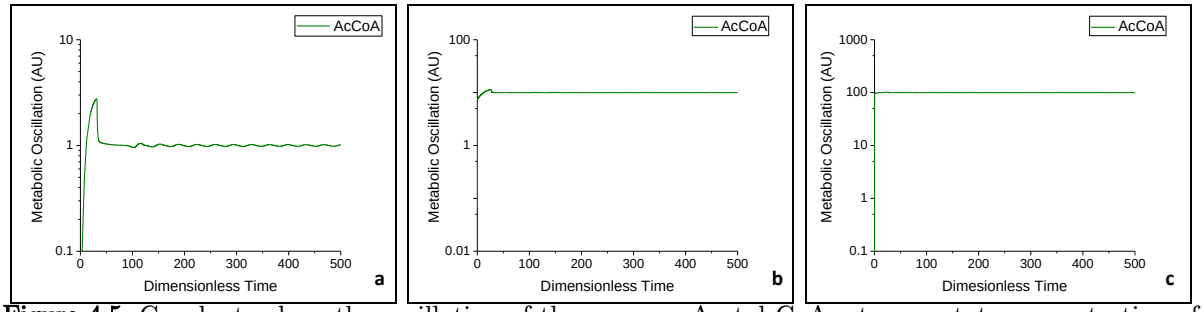

Figure 4.5: Graphs to show the oscillation of the enzyme, Acetyl-CoA, at an acetate concentration of 100mM using the ANN-RK4 model, step size = 0.01, a. Vgly = 10, b. Vgly = 100, c. Vgly = 1000.

The following results detail those found when the NLMPC formulation (see equation 3.47) was utilised. Within each change in the model a number of parameters are fixed and others that vary, these include:

- The dimensionless time step size for each model is 0.01
- When shown, Q_z and Q_u are the penalties on deviation of the state variables (AcCoA, or AcP where appropriate) and control variables (Vgly, or HOAc where appropriate)
- The model does not take into account external factors that can affect the system, such as pH and temperature

4.3 NLMPC Results

As stated earlier one of the aims of this work is to implement control for nonlinear dynamic systems and for the metabolator NLMPC this can be achieved by controlling the output of the two main state variables, AcCoA and AcP. Controlling AcCoA within the system is important as it can ultimately change the level of acetate produced, which can affect the amount of acetate leaving the cell. If the system is controlled to continuously produce acetate this can then be used to produce plastics, nylon and used in other systems that utilise acetate such as the citric acid cycle and biofuels (Menezes *et al.*, 2015). The model was then developed with aim to control the system using the glycolytic flux (Vgly) as presented in Figure 4.6. This control variable was able to successfully track the set activity for AcCoA that was selected for the model and results are presented in Figure 4.7. As shown in Figure 4.7a the system is able to track variations in the set points, and the resulting concentrations of Vgly are expected based on the increase or decrease in the set AcCoA concentrations. Further work carried out using this model involved

adding a velocity constraint on Vgly. This bounded constraint, Dvgly (see Section 3.6.3, Equation 3.56), was envisaged to allow the system to track the set points more closely, as well as give an outcome for Vgly that would have fewer or less fluctuation as the problem is constrained within a defined range. This would be advantageous *in vivo* as controlling glucose concentration (Vgly) in the media would be easier to achieve with fewer fluctuations. Further tests were carried out using a positive range for DVgly (as Vgly feeds into the system and cannot be taken out) and results are presented in Figure 4.8.

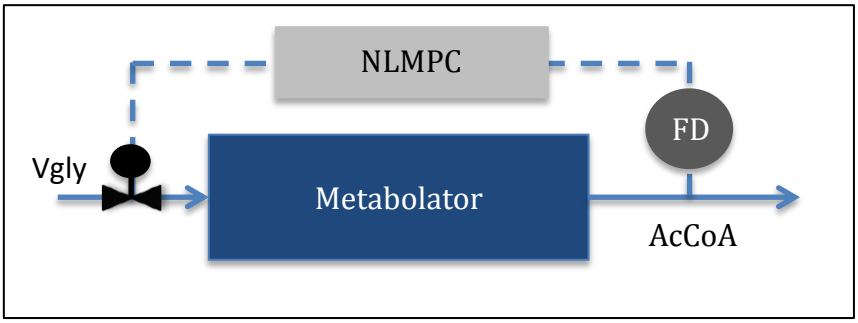

Figure 4.6: Control block showing the metabolator optimised with Vgly as the control variable for the state variable AcCoA. The online concentration of AcCoA can be determined using fluorescence imaging, here represented as a fluorescence detector (FD) by measuring green fluorescent protein (GFP) from the host strain of *E. Coli*.

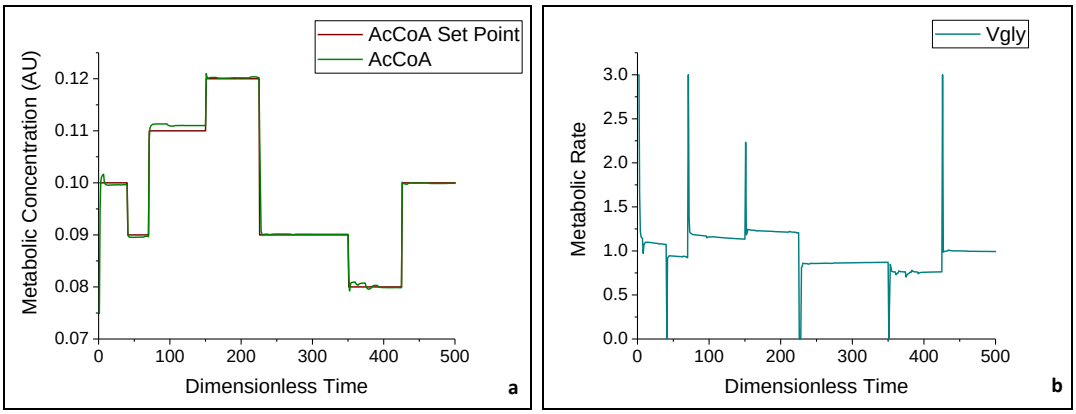

Figure 4.7: a. NLMPC results for AcCoA at small variations in the set points (red) and the tracking of the set points (green), **b.** Concentration of Vgly in the system showing the fluctuations required to track the set points. Qz = 1, Qu = 0.

Figure 4.8 shows the various velocity constraints placed on Vgly and their resulting tracking of the set points. It is noted that the tighter the bounds for the constraints, the fewer sharp changes are present in the rate of Vgly. The tracking of the set

points remains fairly consistent, however it is also noted that the results for AcCoA 1 (where the bounds on the velocity constraint on Vgly are large) do not track the set points as closely as the other conditions. The resulting Vgly have also been displayed in Figure 4.8, which shows the distinct change in the fluctuations of the rate of Vgly, and clearly shows that the tightest range on velocity, Figure 4.9d ($1x10^{-6} \leq Dvgly \leq 0.01$), results in fewer sharp changes in the rate, and therefore indicates a profile that can be replicated experimentally. This is further illustrated in Figures 4.9a-c which show sharp peaks in flux. If in an experimental setup, this would mean sharp changes in the concentration of glucose to be added to the system, and for biological systems this can be too fast for the system to react to, especially in sharp decreases in concentration. Tight ranges were therefore employed to keep the velocity, and overall rate of Vgly, low.

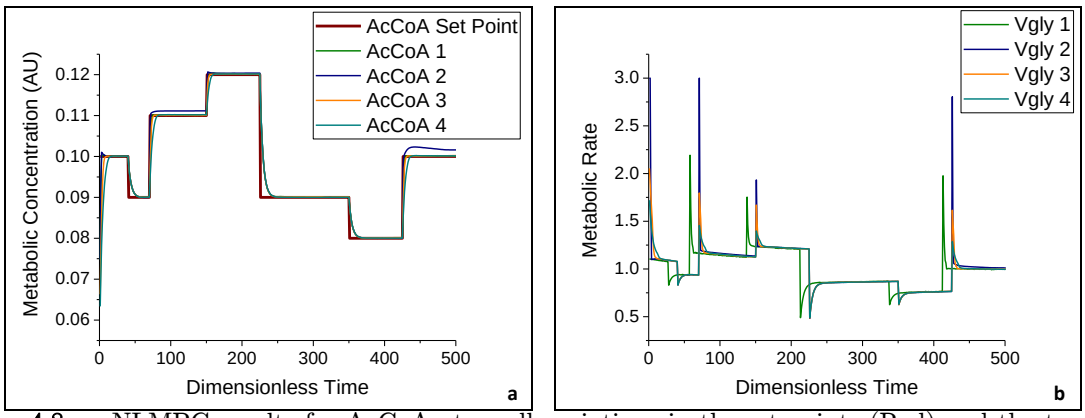

Figure 4.8: **a.** NLMPC results for AcCoA at small variations in the set points (Red) and the tracking of the set points, **b.** The rate of Vgly in the system showing the fluctuations required to track the set points. Here AcCoA 1 is optimised with $1x10^{-9} \leq Dvgly \leq 10$ (Vgly 1), AcCoA 2 is optimised with $1x10^{-6} \leq Dvgly \leq 10$ (Vgly 2), AcCoA 3 is optimised with $1x10^{-6} \leq Dvgly \leq 0.1$ (Vgly 3) and AcCoA 4 is simulated with $1x10^{-6} \leq Dvgly \leq 0.01$ (Vgly 4). The step size is 0.01 for all simulations.

Another aspect that had been investigated was the inclusion of tracking the control profile for Vgly as well as AcCoA. This involved including set points for the control variable, whereby the system would track both the control variable set point and the manipulated variable set point. These results are presented in Figure 4.10 where the decrease in the bounds of Dvgly gives tighter control and has greatly reduced the sharp changes in the rate. However, although Vgly is able to track the set point, there are still violations present, as the system cannot strictly adhere to the set points for both the control and state variable. The system does prefer to adhere to the state variable set points over the control variable set points as there are fewer violations. Figure 4.9c shows the tightest bounds placed on Vgly, and gives the best

results in terms of tracking the manipulated variable as well as the Vgly set points. However, when compared to results from Figure 4.8 there is little difference, and therefore the case for when Qu = 0 (Figures 4.7 and 4.8) are deemed to be better suited for the metabolator when using Vgly as a single control variable.

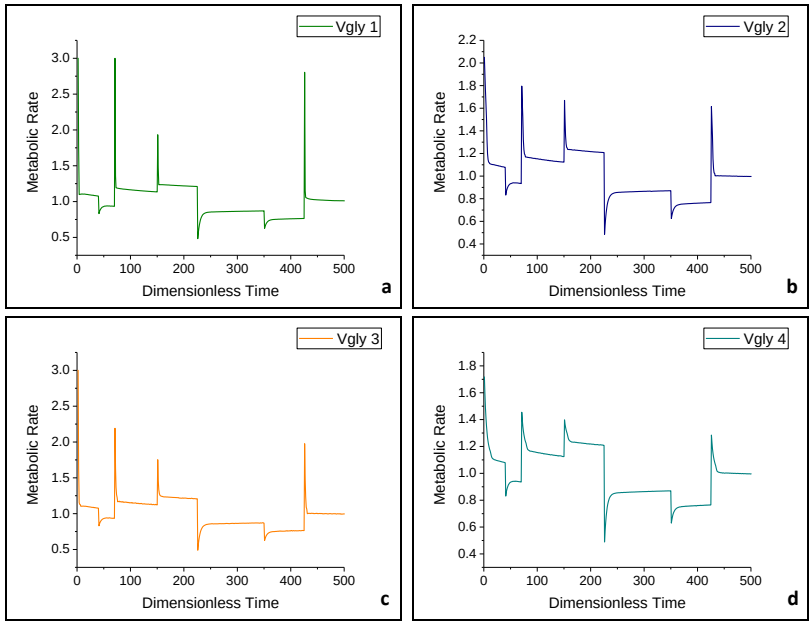

Figure 4.9: The rates of Vgly required for each simulation in Figure 4.7, where **a.** $1x10^{-9} \leq$ Dvgly ≤ 10 (AcCoA 1), **b.** $1x10^{-6} \leq$ Dvgly ≤ 10 (AcCoA 2), **c.** $1x10^{-6} \leq$ Dvgly ≤ 0.1 (AcCoA 3) and **d.** $1x10^{-6} \leq$ Dvgly ≤ 0.01 (AcCoA 4). The step size is 0.01 for all simulations.

Glycolytic flux (Vgly) is also used as the control variable for set point tracking of AcP within the metabolator due to the synergistic relationship of the process. The control block for this investigation of the model is shown in Figure 4.11. The results for this investigation are presented in Figure 4.12, where it is apparent that the rate of Vgly increases when there is a decrease in the set point of AcP. This relationship is expected as an increase in Vgly means more production of AcCoA, which in turn leads to less production of AcP as the relationship between the two metabolite pools favours this dynamic. The results for Vgly in Figure 4.12 show that it remains at a baseline constant level until there is a decrease in AcP, where the rate of flux also decreases. This simulates the stability of the system when AcCoA is produced and as a result there is little production of AcP. There is a threshold to this process however and this is achieved when there is no Vgly present, whereby the production of AcCoA will decrease thereby leading to an increase in AcP. This will then propagate the negative feedback loop in the system and cause AcP production to feed Acs back into the first metabolite pool containing AcCoA. This is the cyclical

process of the Metabolator, and the results in Figure 4.12 show the case where Vgly is present, therefore leading to a decrease in production of AcP. In terms of tracking the set points, the system is able to track well, as shown in Figure 4.12c.

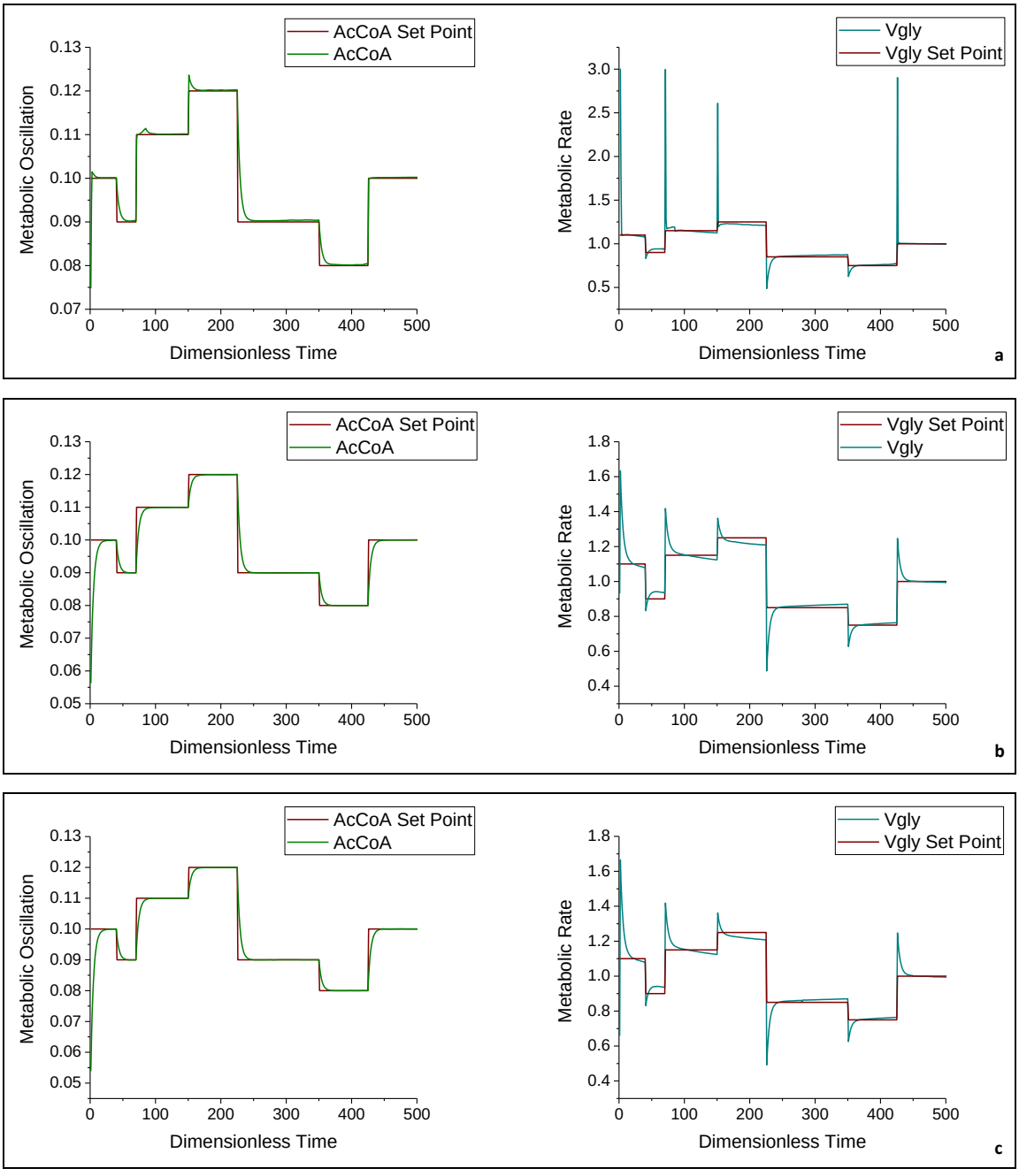

Figure 4.10: NLMPC results for AcCoA at small variations in the set points (red) and the tracking of the set points (green), and the rate of Vgly against the Vgly set point in response to tracking AcCoA, where **a.** $1x10^{-3} \leq Dvgly \leq 1$, **b.** $1x10^{-6} \leq Dvgly \leq 1x10^{-4}$ and **c.** $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$. Qz = 1, Qu = 1.

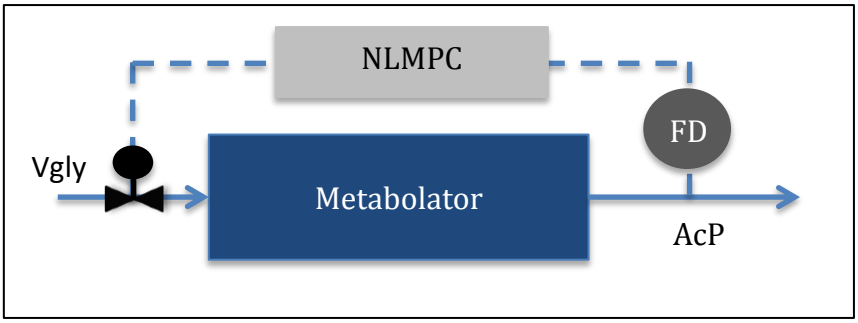

Figure 4.11: Control block showing the metabolator optimised with Vgly as the control variable for the state variable AcP. The online concentration of AcP can be determined using fluorescence imaging, here represented as a fluorescence detector (FD) by measuring green fluorescent protein (GFP) from the host strain of *E. Coli*.

From the results seen in Figures 4.7-4.10 it can be noted that the system is able to track a set points for AcCoA, by manipulating Vgly. The results in Figure 4.12 show that the system can respond well when Vgly is used as the control variable for controlling AcP. Further investigation into adding disturbance into the system is presented in the following section (Section 4.4).

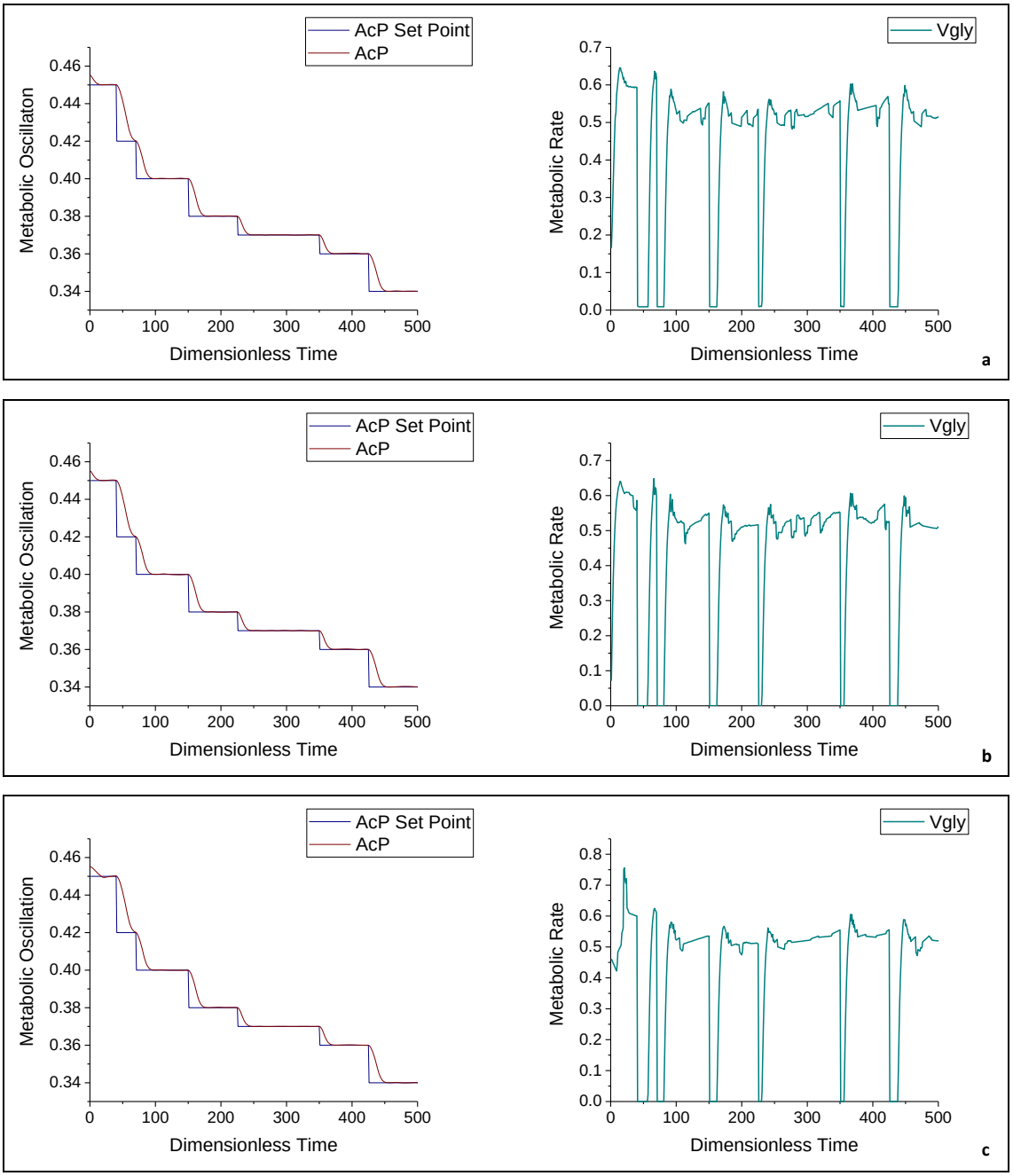

Figure 4.12: NLMPC results for AcP at small variations in the set points (blue) and the tracking of the set points, and the rate of Vgly in the system required to track AcP, where **a.** $1x10^{-3} \leq$ Dvgly ≤ 1, **b.** $1x10^{-6} \leq$ Dvgly $\leq 1x10^{-4}$ and **c.** $1x10^{-9} \leq$ Dvgly $\leq 1x10^{-6}$. Qz = 1.

4.4 NLMPC with Disturbance

As it is noted in Fung *et al.* (2005) the system has many types of disturbances that can affect performance. Stochastic disturbance in the model was thought to be the reason for the discrepancy they noticed between experimental and simulated results.

They utilised a variation of the Langevin technique involving a Gaussian process and a Euler's scheme to quantify this disturbance. It was noted that this stochastic noise can have great effect on the dynamics of the proteins in the system, LacI, Pta and AcP. Therefore, in this work, disturbance, d, was added to the system in the form of a uniform disturbance given over a confined range and added to the RHS of equation 4.1. The tracking of the set points would invariably be disrupted due to the disturbance as presented in Figure 4.13; however what is noted is that the system tries to adhere to the set points as closely as possible. Although there are violations from the set point the tracking tries to remain as close to the set path as possible despite the disturbance that is added to the system. It is important to note here that disturbances of up to 1000% of the nominal value of the state variables AcCoA and AcP has been added to the model.

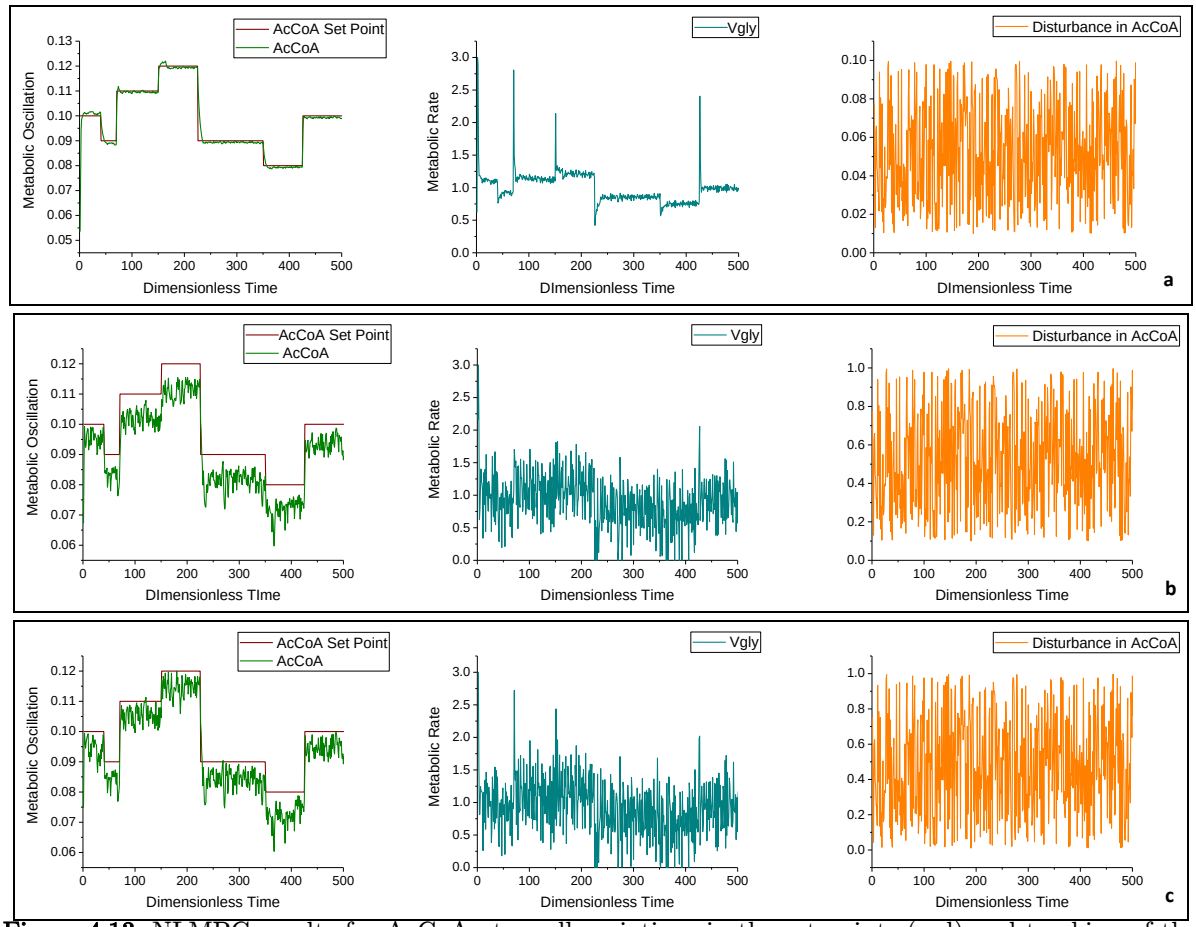

Figure 4.13: NLMPC results for AcCoA at small variations in the set points (red) and tracking of the set points (green), the rate of Vgly required to track the set points and the disturbance, d, added to AcCoA, where d is a random disturbance with uniform distribution. **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. $Qz = 1$, $Qu = 0$ and $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$ for all cases.

Further investigation into disturbance is also carried out for the case when Vgly is the control variable for AcP, by adding d to the RHS of equation 4.2. Results in Figure 4.14 show this case and as shown at large disturbances the tracking is lost. From the results presented in Figure 4.13 it can be noted that the system is still able to track the AcCoA set point in the presence of disturbance. Although the system tries to track the case for when AcP is chosen as the manipulated variable (Figure 4.14), there is a loss in tracking at larger disturbance (Figures 4.14b ad 4.14c). The possibility of having this disturbance (uncertainty) in a model with two set points (Zone NLMPC) for combating the issues with tracking will be investigated in the following section as a means of preventing violation of the set points.

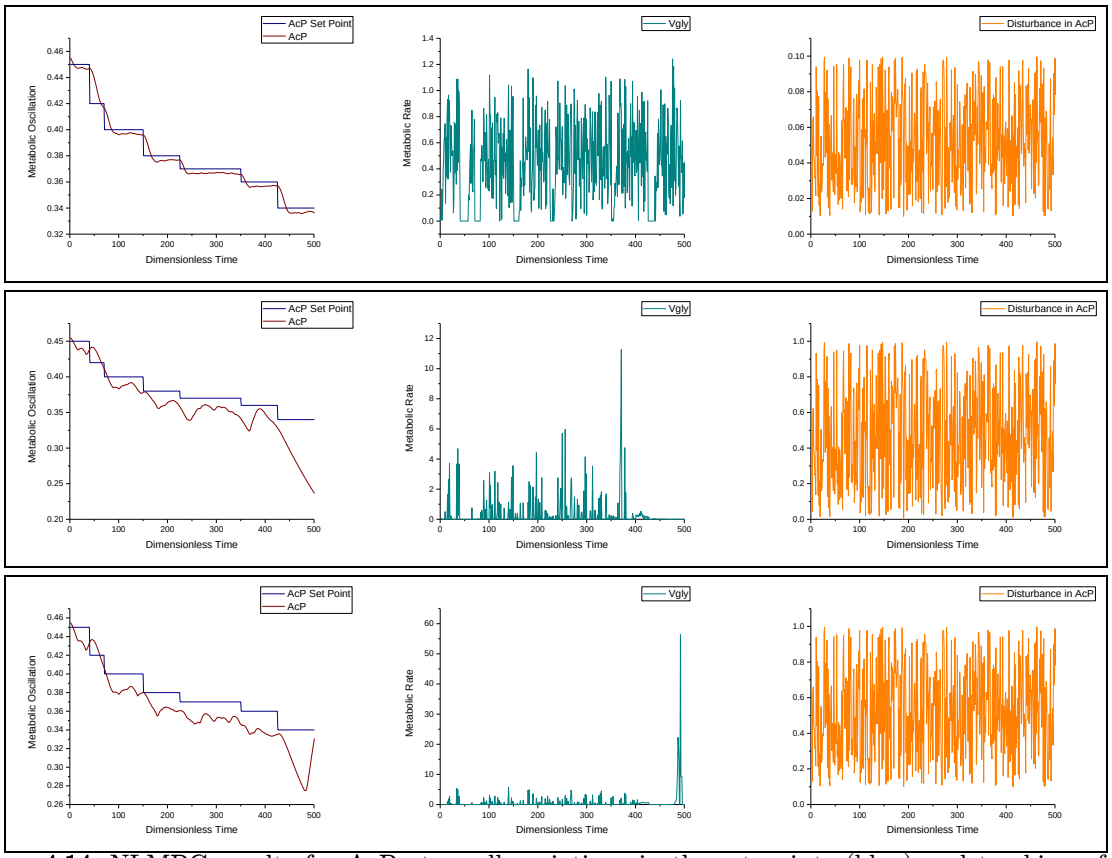

Figure 4.14: NLMPC results for AcP at small variations in the set points (blue) and tracking of the set points (red), the rate of Vgly required to track the set points and the disturbance, d, added to AcP, where d is a random disturbance with uniform distribution. **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. $Qz = 1$, $Qu = 0$ and $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$ for all cases.

4.5 Zone Model Predictive Control Results

Once the NLMPC was successfully implemented, work progressed to control using Zone Control, where two set points were used, and the metabolator was confined within this range. This was trialled with and without the velocity constraint, Dvgly, to see if this had any effect on the performance. Different step sizes for the RK4 convergence were also investigated to see the effect on the metabolator performance. The results from this investigation showed that smaller step sizes gave results analogous to the ANN, and in fact the same step size was used (0.01). Results for when Zone Control was implemented (Figure 4.15) show that the state variable, AcCoA, remains within the range without violation. In terms of the rate of Vgly, having a tight restriction on the rate, Figure 4.15c, shows less fluctuation, which is advantageous in trying to replicate results experimentally. Again, further investigation into this system utilised a set point for Vgly to decrease the fluctuations in the rate (Figure 4.16). As with previous results (Figure 4.10) the addition of set points for Vgly does not affect the tracking of the system greatly. However, it is positive that the system is able to adhere to the tracking of Vgly, as well as responding to the tracking of AcCoA (the manipulated variable). The system is able to track within the given range, which is promising as it allows for stricter control of AcCoA concentration, and as a result will control the metabolator performance.

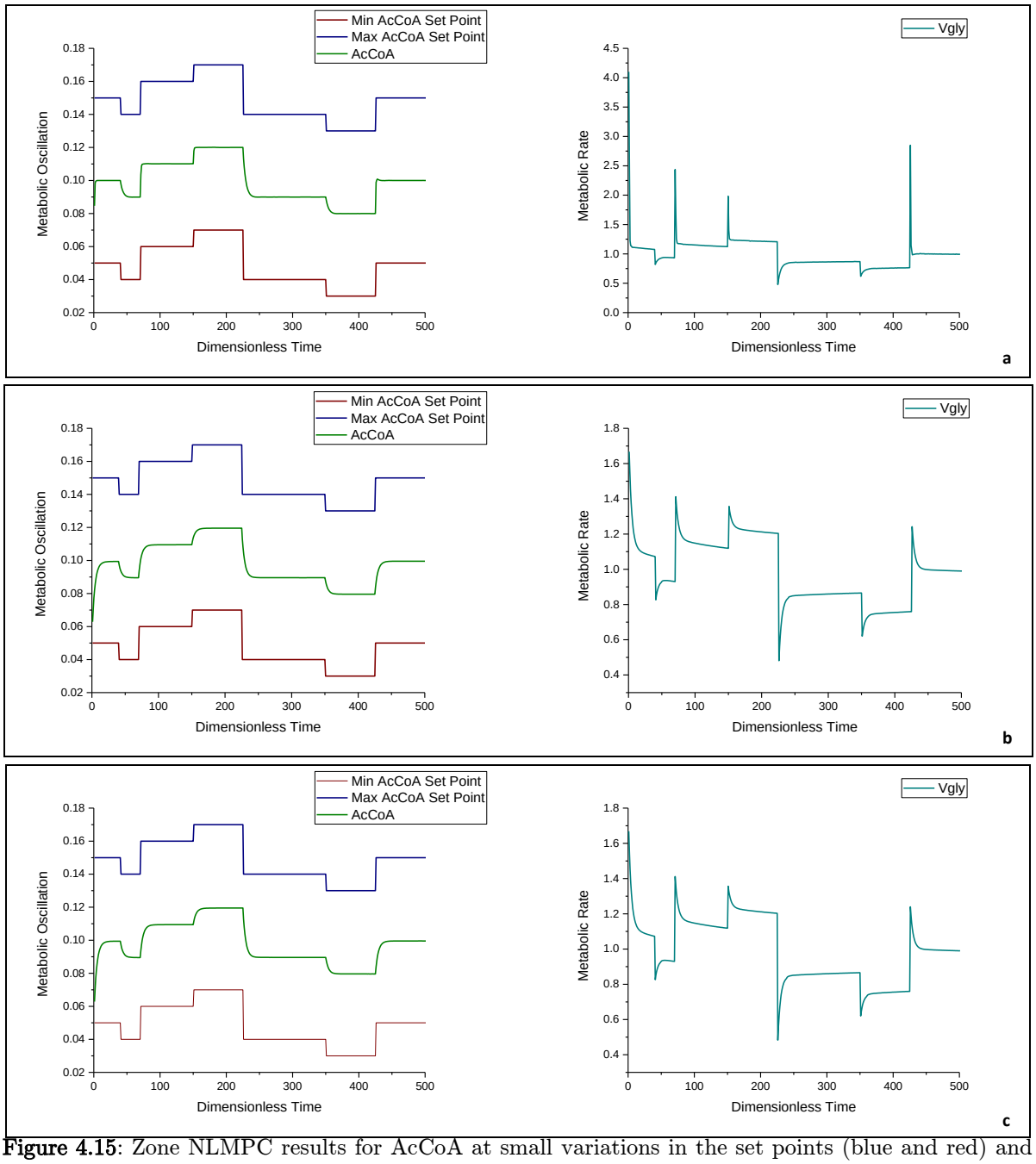

Figure 4.15: Zone NLMPC results for AcCoA at small variations in the set points (blue and red) and tracking of the set points (green), and graphs to show the concentration of Vgly in the system showing the fluctuations required to track the set points, where **a.** $1x10^{-3} \leq Dvgly \leq 1$, **b.** $1x10^{-6} \leq Dvgly \leq 1x10^{-4}$ and **c.** $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$. $Qz = 1$, $Qu = 0$.

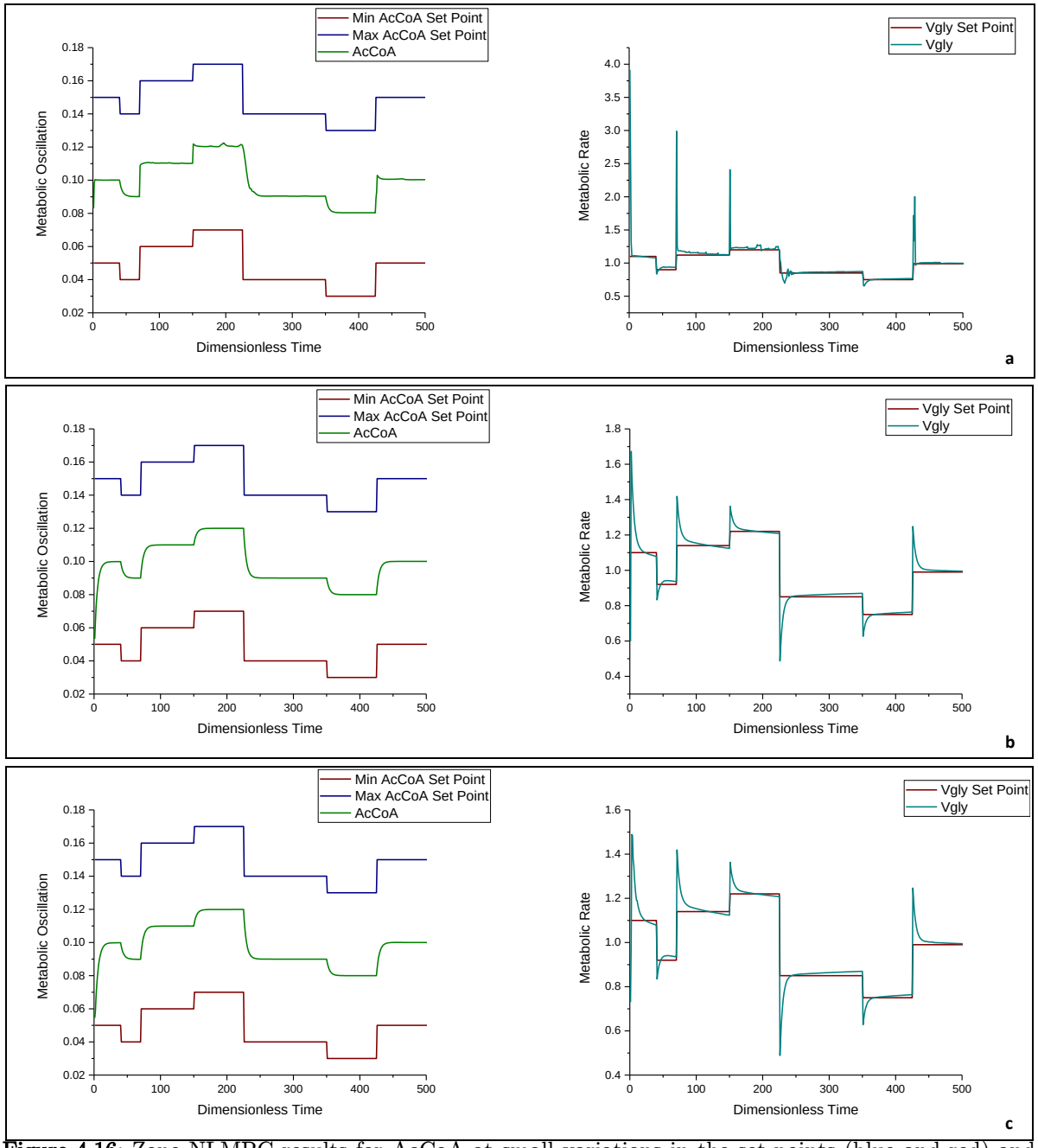

Figure 4.16: Zone NLMPC results for AcCoA at small variations in the set points (blue and red) and the tracking of the set points (green), and graphs to show the rate of Vgly against the Vgly set point in response to tracking AcCoA, where **a.** $1x10^{-3} \leq$ Dvgly ≤ 1, **b.** $1x10^{-6} \leq$ Dvgly $\leq 1x10^{-4}$ and **c.** $1x10^{-9} \leq$ Dvgly $\leq 1x10^{-6}$. Qz = 1, Qu = 0.1.

Zone Control was also implemented for the case where Vgly controls AcP in the metabolator, which is presented in Figure 4.17. Again the system is able to track within the given range, and the response rate of Vgly shows little change from Figure 4.17a-4.17c. This can also be attributed to the system dynamics, whereby the control of AcP using Vgly will not show the same rapid response as AcCoA due to

the many stages in the reaction as shown in Figure 4.2, where Vgly enters the system and causes the first metabolite, AcCoA, to oscillate. This reaction causes Pta to be produced and this then enters the second metabolite pool, containing AcP, to cause this to oscillate. This three step reaction is not instantaneous and therefore delays the response of AcP from Vgly.

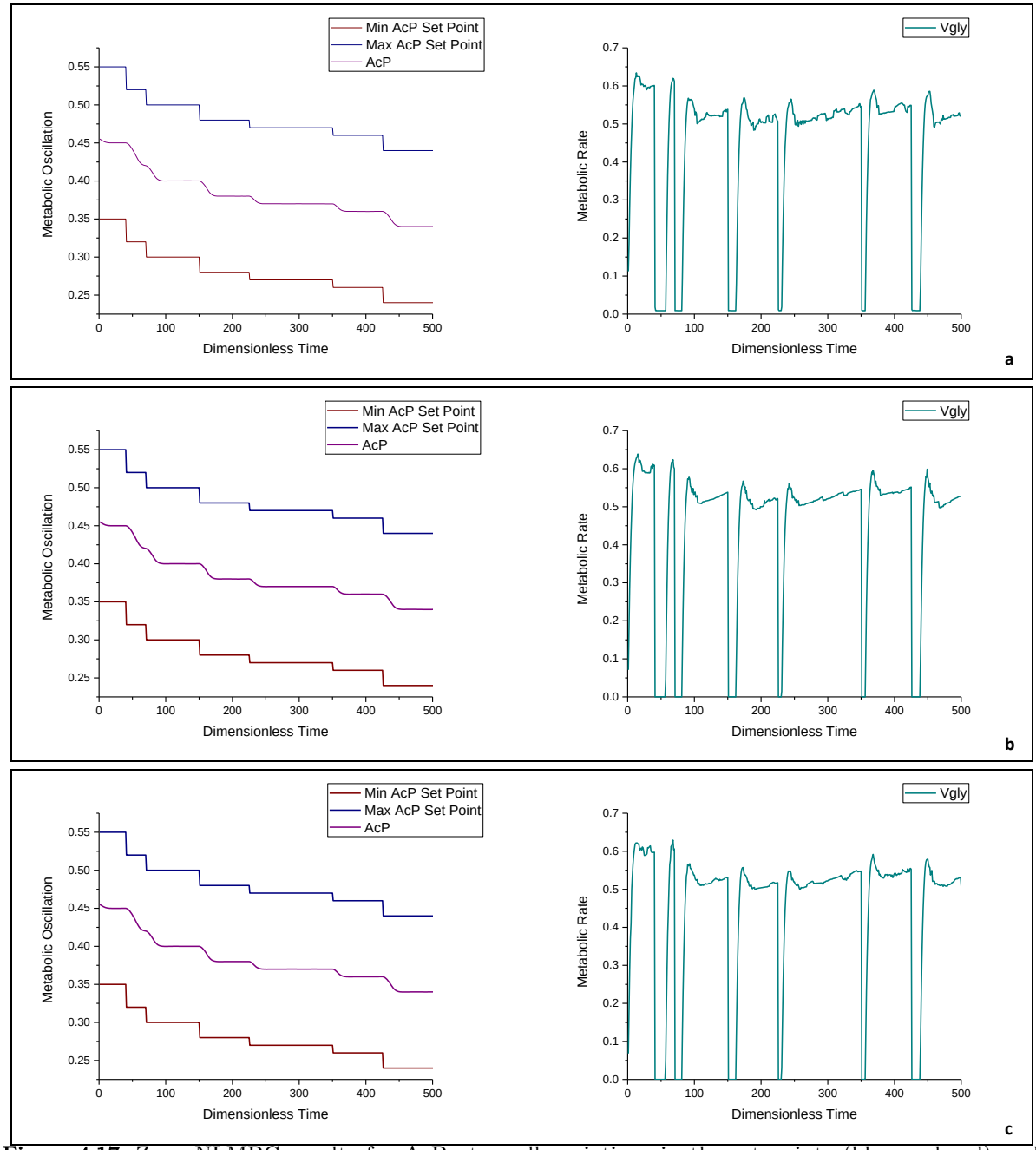

Figure 4.17: Zone NLMPC results for AcP at small variations in the set points (blue and red) and tracking of the set points (purple), and graphs to show the concentration of Vgly in the system showing the fluctuations required to track the set points, where **a.** $1x10^{-3} \leq$ Dvgly ≤ 1, **b.** $1x10^{-6} \leq$ Dvgly $\leq 1x10^{-4}$ and **c.** $1x10^{-9} \leq$ Dvgly $\leq 1x10^{-6}$. Qz = 1, Qu = 0.

It is noted that Zone NLMPC yields excellent tracking of the system as there is a wider range for the system to track, and generally the tracking remains within the centre of this zone. Furthermore, decreasing the bounds of the velocity of Vgly results in a smoother tracking profile, as seen when comparing Figure 4.15a to Figure 4.15c, where not only does the rate of Vgly undergo less sharp change, but the tracking of the system shows greater transition from increasing and decreasing set points of AcCoA. The advantage of using Zone NLMPC for the metabolator is that it allows experimental validation of the system to be performed easily. Rather than with a single set point NLMPC where the resulting rate of Vgly shows sharp change in rate (Figure 4.10a), Zone NLMPC allows relaxation in the tracking problem, thereby resulting in a Vgly rate which does not change as sharply (Figure 4.16a). Experimental validation of this system can then be performed easily as glucose (which can take the place of Vgly) can be added to the system to manipulate AcCoA to follow the set zone rather than the fixed path, which can be difficult to achieve *in vivo*. Furthermore as there is a zone for tracking, the concentration of AcCoA can freely vary within this zone and still perform with the desired user specified range and be deemed optimal. If there was a single set point for tracking, this may not be achieved as the biological system may deviate from the set path and therefore tracking may be lost, resulting in a system that is not meeting the user specified requirement. Instances where Zone NLMPC is favoured over NLMPC is shown in research from Grosman *et al.* (2010) where Zone NLMPC was used to create a region for an automated insulin monitor in diabetic patients. This region defined the optimal levels of insulin needed in the patient following typical day activities and is successful in maintaining patient's glucose levels to the same levels as a normal healthy person.

4.6 Zone Model Predictive Control with Disturbance

Similarly as with the NLMPC model, disturbance was considered for the case of Zone NLMPC in the form of a randomised disturbance, d, added to the RHS of equation 4.1, in order to see what effect it has on the system dynamics. A range of disturbance profiles (0.01-1) were modelled and the results are presented in Figure 4.18, where 4.18a shows that the disturbance profile has not affected tracking of the set points; therefore a larger disturbance was added, as shown in Figure 4.18b. Here it is apparent that there are some rapid fluctuation in both tracking and the response rate of Vgly. Increasing the range for the disturbance, Figure 4.18c, shows similar rapid fluctuation in the tracking and rate of Vgly. We do however see that

the system is still able to track within the given bounds even in a disturbance of up to 1000% the nominal amount of AcCoA. Figure 4.19 shows the results where Vgly is used as the control variable for AcP. As shown in Figure 4.19b, where disturbance is at its highest, there is a distinct loss of tracking at around 450 time steps, however this can be attributed to the scale of the disturbance. The response rate of Vgly is in line with previous results for this type of model (Figure 4.17) where the sharp peaks are also noticed. It is interesting to note that the tracking of the system, for Vgly and AcCoA, under disturbance seems to adhere more towards the upper bound of the zone. The main possibility for this is the presence of the disturbance itself, which causes a positive shift in the oscillatory habit of the metabolator, and therefore increases the amount of AcCoA.

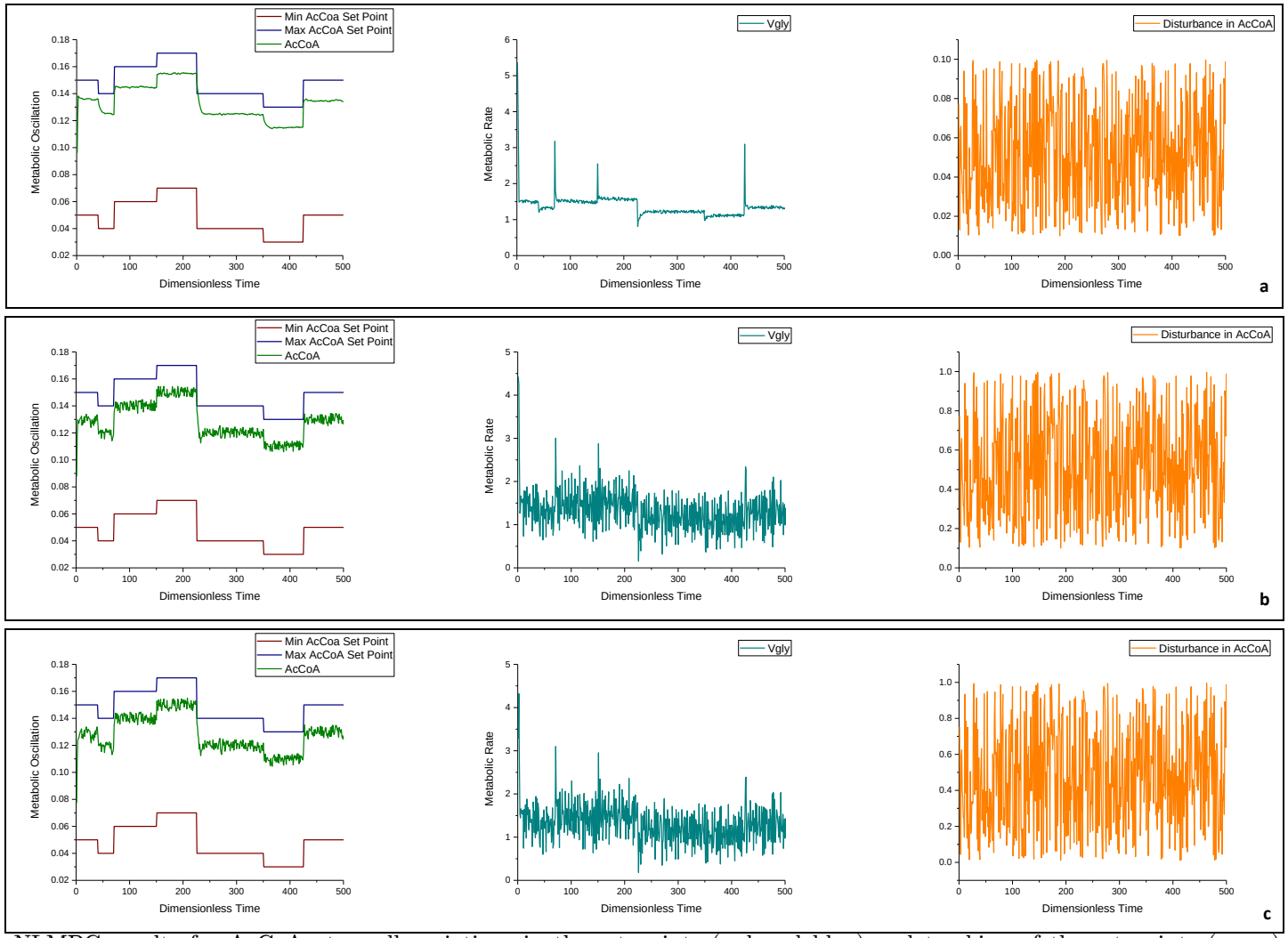

Figure 4.18: Zone NLMPC results for AcCoA at small variations in the set points (red and blue) and tracking of the set points (green), the rate of Vgly required to track the set points and the disturbance, d, added to AcCoA, where **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. Qz = 1, Qu = 0 and $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$ for all cases.

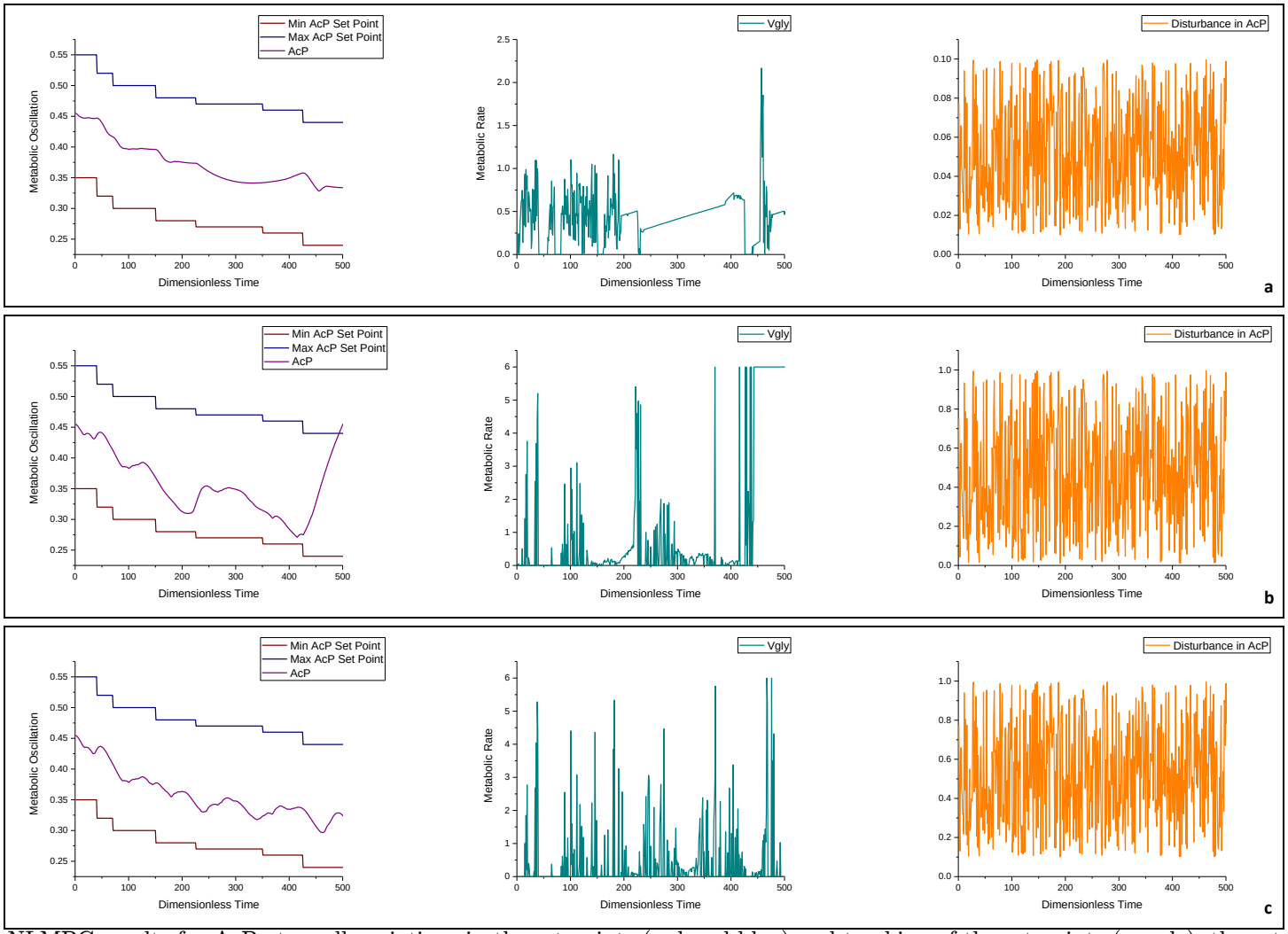

Figure 4.19: Zone NLMPC results for AcP at small variations in the set points (red and blue) and tracking of the set points (purple), the rate of Vgly required to track the set points and the disturbance, d, added to AcP, where **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. Qz = 1, Qu = 0 and $1x10^{-9} \leq$ Dvgly $\leq 1x10^{-6}$ for all cases.

4.7 Acetate as a Control Variable

Due to the synergistic relationship of the metabolite pools in the metabolator acetate (HOAcE) was used as a control variable to propogate the negative feedback reaction via AcP (Figure 4.20). Therefore a new controller was implemented utilising acetate as the control variable. Similar to Vgly this model was then altered from NLMPC to Zone NLMPC. The results for the NLMPC and Zone NLMPC utilising acetate as a control variable are presented in this section.

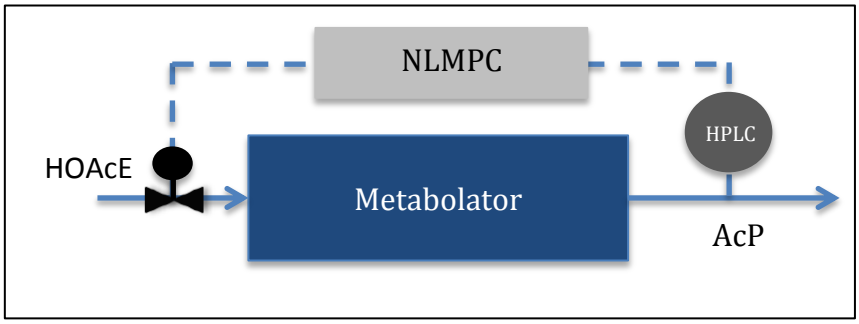

Figure 4.20: Control block showing the metabolator optimised with acetate (HOAcE), which can be measured using HPLC analysis as shown, as the control variable for the state variable AcP.

As shown in Figure 4.21 the system does try to track the set point for AcP, however there is a delay. This could be due to the fact that the concentration of acetate is external from the cell, and the relationship between AcP and acetate does not occur in the same dynamic equation of the ODEs. However, the system is able to track the set points and decreasing the bounds on the velocity of HOAcE gives fewer sharp changes in its concentration and can be seen comparing Figure 4.21a and Figure 4.21c, where the graph depicts the standard and zoomed in results for the same profile.

Acetate can also be used as a control variable for AcCoA (Figure 4.22). As previously stated, the metabolator is cyclical and interconnected, therefore a bottom up reaction from acetate to AcCoA can also control the outcome of the system. Results from this are presented in Figure 4.23, where it is noted that the tracking for AcCoA does have some violation from the set points. Also the concentration of acetate required for the tracking changes rapidly, especially for results seen in Figures 4.23a and 4.23b. A reason for this can be due to the fact that the reaction for acetate to AcCoA has to go through multiple steps before it can have an effect, whereas the reaction from Vgly to AcCoA is a direct reaction. This can alter the tracking of the system and can explain the differences (delay) seen. Further work

into using acetate as a control variable progressed into using Zone NLMPC, presented in Figure 4.24. As shown in Figure 4.24 the system is able to track within the set points.

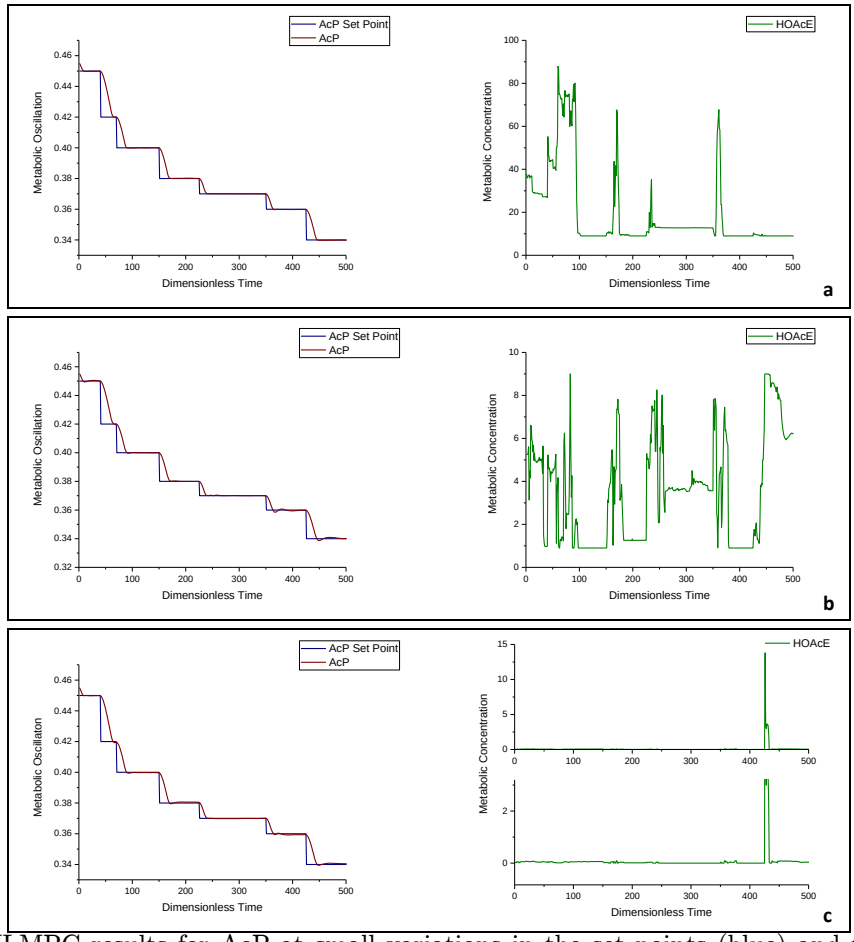

Figure 4.21: NLMPC results for AcP at small variations in the set points (blue) and tracking of the set points (red), and graphs to show the concentration of acetate (HOAcE) in the system showing the fluctuations required to track the set points, where **a.** $1 \leq DHOAcE \leq 10$, **b.** $0.1 \leq DHOAcE \leq 1$ and **c.** $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$. $Qz = 1$, $Qu = 0$.

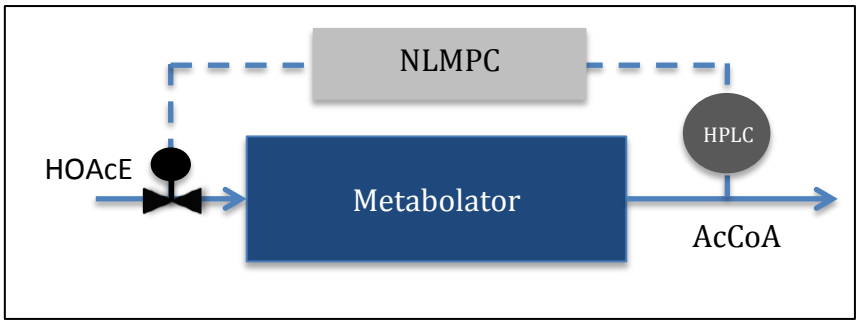

Figure 4.22: Control block showing the metabolator optimised with acetate (HOAcE), which can be measured using HPLC analysis as shown, as the control variable for the state variable AcP.

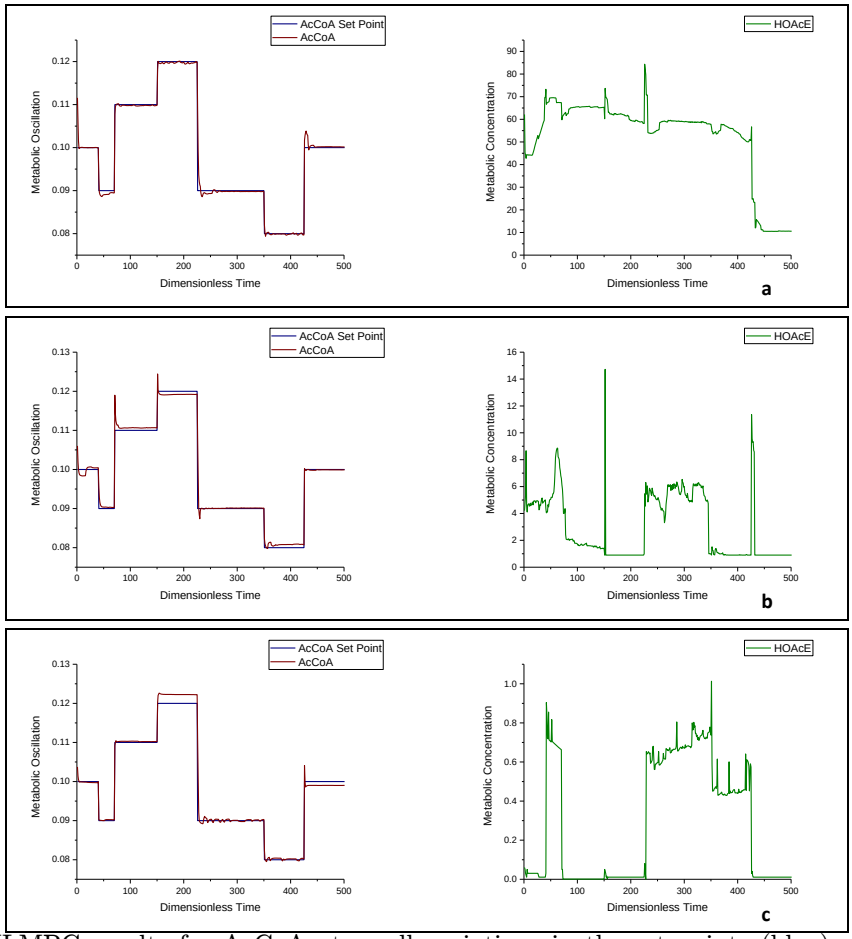

Figure 4.23: NLMPC results for AcCoA at small variations in the set points (blue) and tracking of the set points (red), and graphs to show the concentration of acetate (HOAcE) in the system showing the fluctuations required to track the set points, where **a.** 1 ≤ DHOAcE ≤ 10, **b.** 0.1 ≤ DHOAcE ≤ 1 and **c.** $1x10^{-4}$ ≤ DHOAcE ≤ $1x10^{-2}$. Qz = 1, Qu = 0.

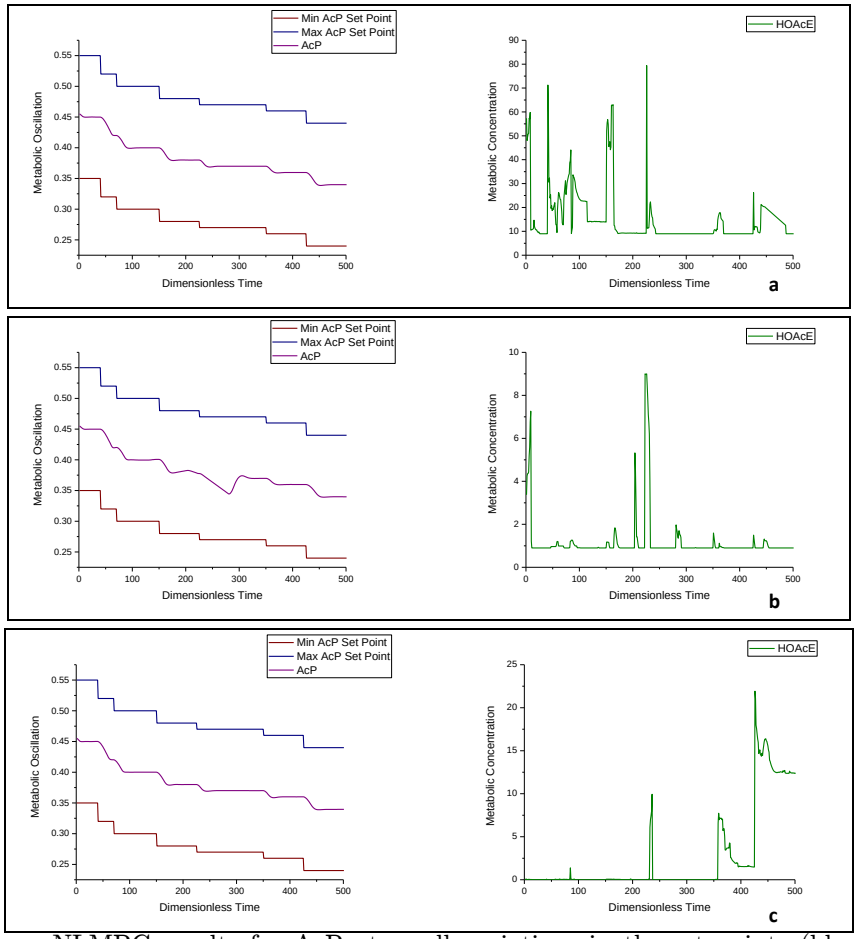

Figure 4.24: Zone NLMPC results for AcP at small variations in the set points (blue and red) and tracking of the set points (purple), and graphs to show the concentration of acetate (HOAcE) in the system showing the fluctuations required to track the set points, where **a.** $1 \leq DHOAcE \leq 10$, **b.** $0.1 \leq DHOAcE \leq 1$ and **c.** $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$. $Qz = 1$, $Qu = 0$.

Zone NLMPC was also applied for the case when AcCoA was tracked using acetate as the control variable (Figure 4.22). The results for this investigation are presented in Figure 4.25, where the tracking of the set points remains within the zone, and the concentration of acetate does now fluctuate as highly as seen in Figure 4.24.

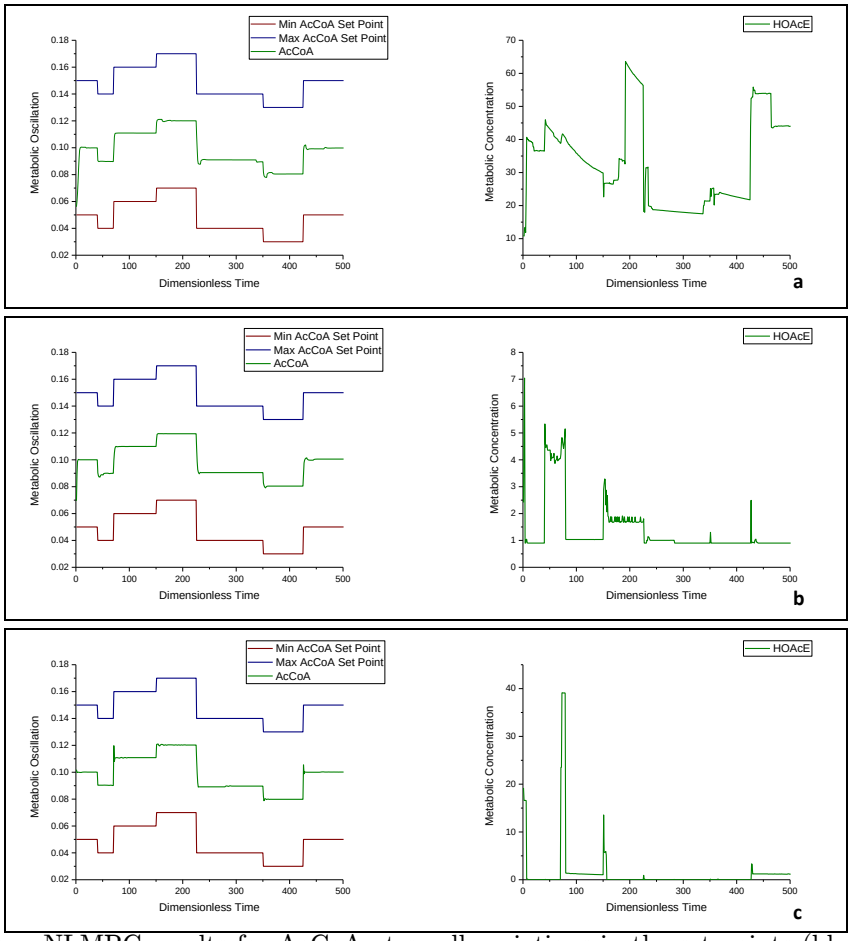

Figure 4.25: Zone NLMPC results for AcCoA at small variations in the set points (blue and red) and tracking of the set points (green), and graphs to show the concentration of acetate (HOAcE) in the system showing the fluctuations required to track the set points, where **a.** 1 ≤ DHOAcE ≤ 10, **b.** 0.1 ≤ DHOAcE ≤ 1 and **c.** $1x10^{-4}$ ≤ DHOAcE ≤ $1x10^{-2}$. Qz = 1, Qu = 0.

As with the profiles using Vgly as a control variable, disturbance was considered for both the NLMPC and Zone NLMPC model for acetate. Again the disturbance, d, was uniformally distributed in a given range. The results for the NLMPC models are shown in Figures 4.26 and 4.27, and the results for Zone NLMPC are presented in Figures 4.28 and 4.29.

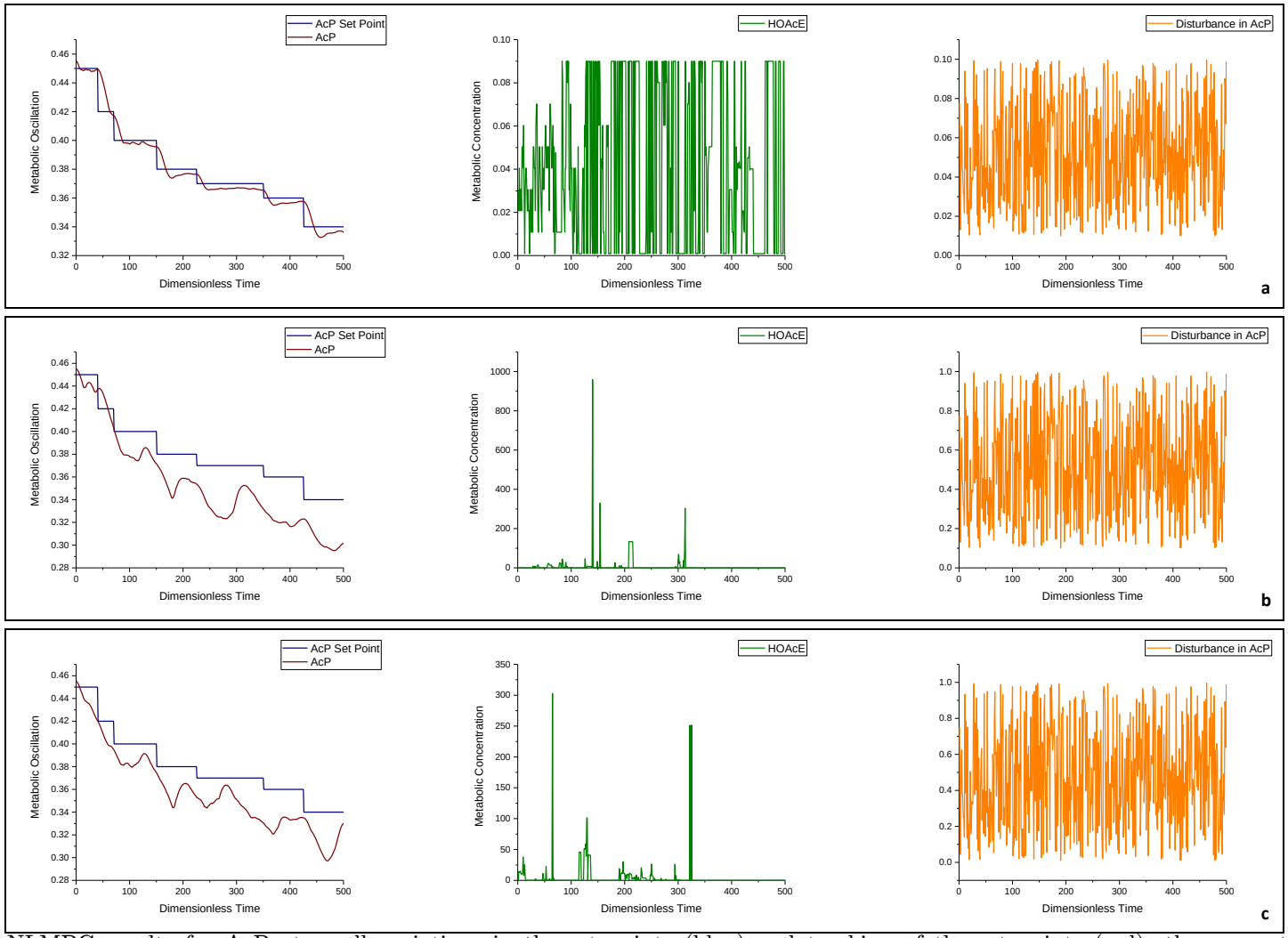

Figure 4.26: The NLMPC results for AcP at small variations in the set points (blue) and tracking of the set points (red), the concentration of HOAcE required to track the set points and the disturbance, d, added to AcP, where **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. Qz = 1, Qu = 0 and $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$ for all cases.

As presented in Figure 4.26 an increased disturbance in the system causes the tracking of AcP to become imbalanced with regards to adhering to the set points. Unlike the case where Vgly was tracking AcP without disturbance (Figure 4.14), where the tracking remained positive with few violations, here we see a distinct loss of tracking (Figure 4.26b and Figure 4.26c). Figure 4.27 shows the case where acetate is used to control AcCoA under disturbance also. Generally, even with large disturbances (Figure 4.27b) the tracking of the system is good and there is little violation. Where there is a reduction in the metabolic oscillation of AcCoA, there is an increase in the concentration of acetate, and this coincides with the behaviour of the metabolator (Figure 4.1).

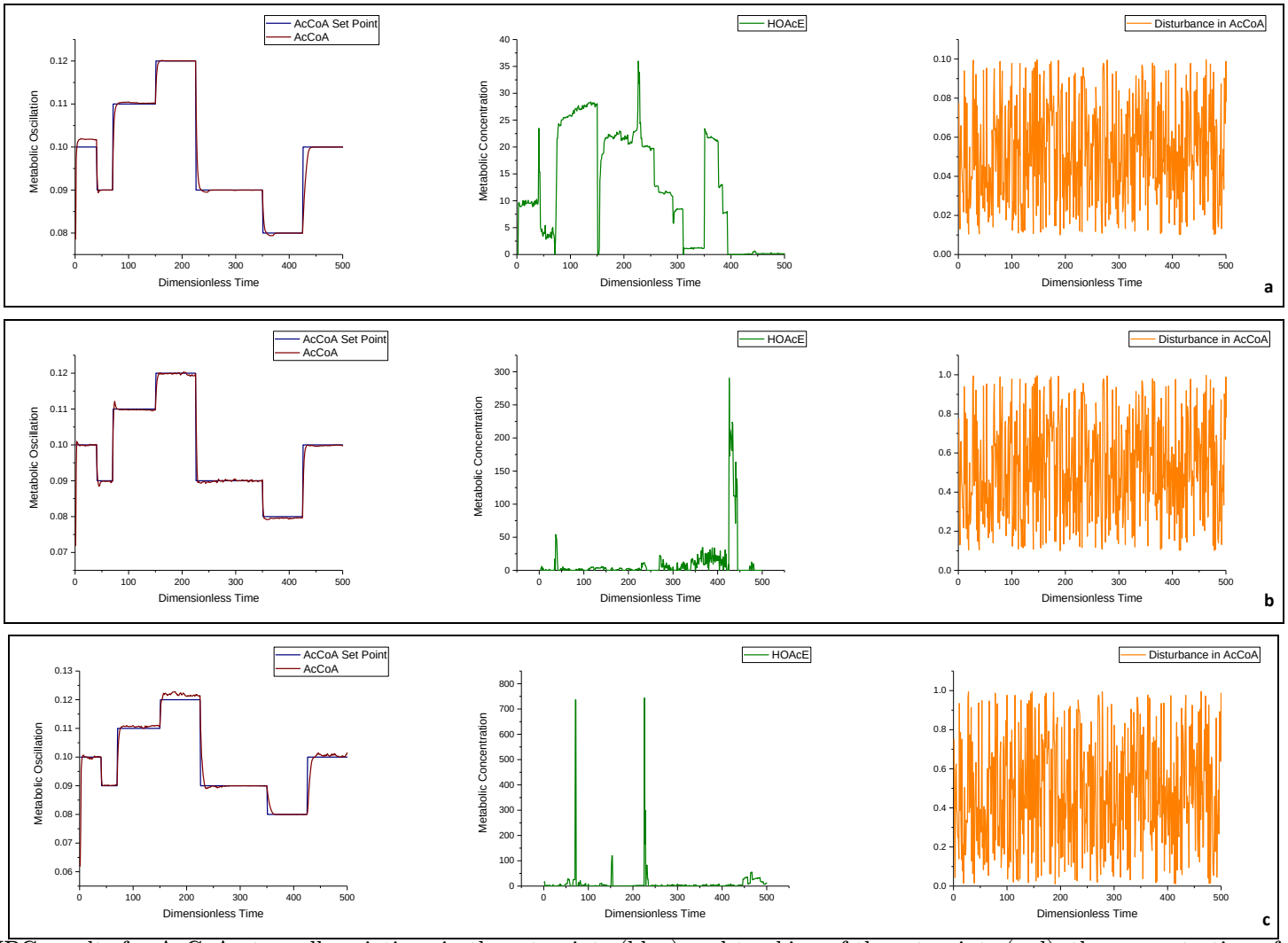

Figure 4.27: NLMPC results for AcCoA at small variations in the set points (blue) and tracking of the set points (red), the concentration of acetate (HOAcE) required to track the set points and the disturbance, d, added to AcCoA, where **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. Qz = 1, Qu = 0 and $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$ for all cases.

Zone NLMPC with disturbance was then implemented, and the results for this are presented in Figures 4.28 and 4.29. In Figure 4.28 the tracking within the zone has not been greatly affected in the presence of a small disturbance (Figure 4.28a), however in the presence of a larger disturbance (Figure 4.28b) there is a change in the tracking behaviour of the system. The system does still track within the given bounds, much like when disturbance was added to the simulations where Vgly was the control variable. Figure 4.29 shows good results for the tracking of AcCoA, and the result remains consistantly in the middle of the range.

It can also be noted that the investigations whereby a set point was used for acetate as well as AcP (similar to the case presented in Figure 4.16 for AcCoA and Vgly) showed that the system did not adhere to the set point. It was noted that although the system was able to track the set points for AcP, the resulting concentrations of acetate did not adhere to their set points. The conclusion from this was that the system is dominated by AcCoA, and therefore was unable to track set points for acetate. There is also possibility that any effect on acetate concentration will not directly affect AcP as within the system acetate needs to undergo re-uptake and inter-conversion to free acetate ions before being able to affect AcP, as shown in Figure 4.2. This indirect reaction can also explain the behaviour of acetate in the system following disturbance.

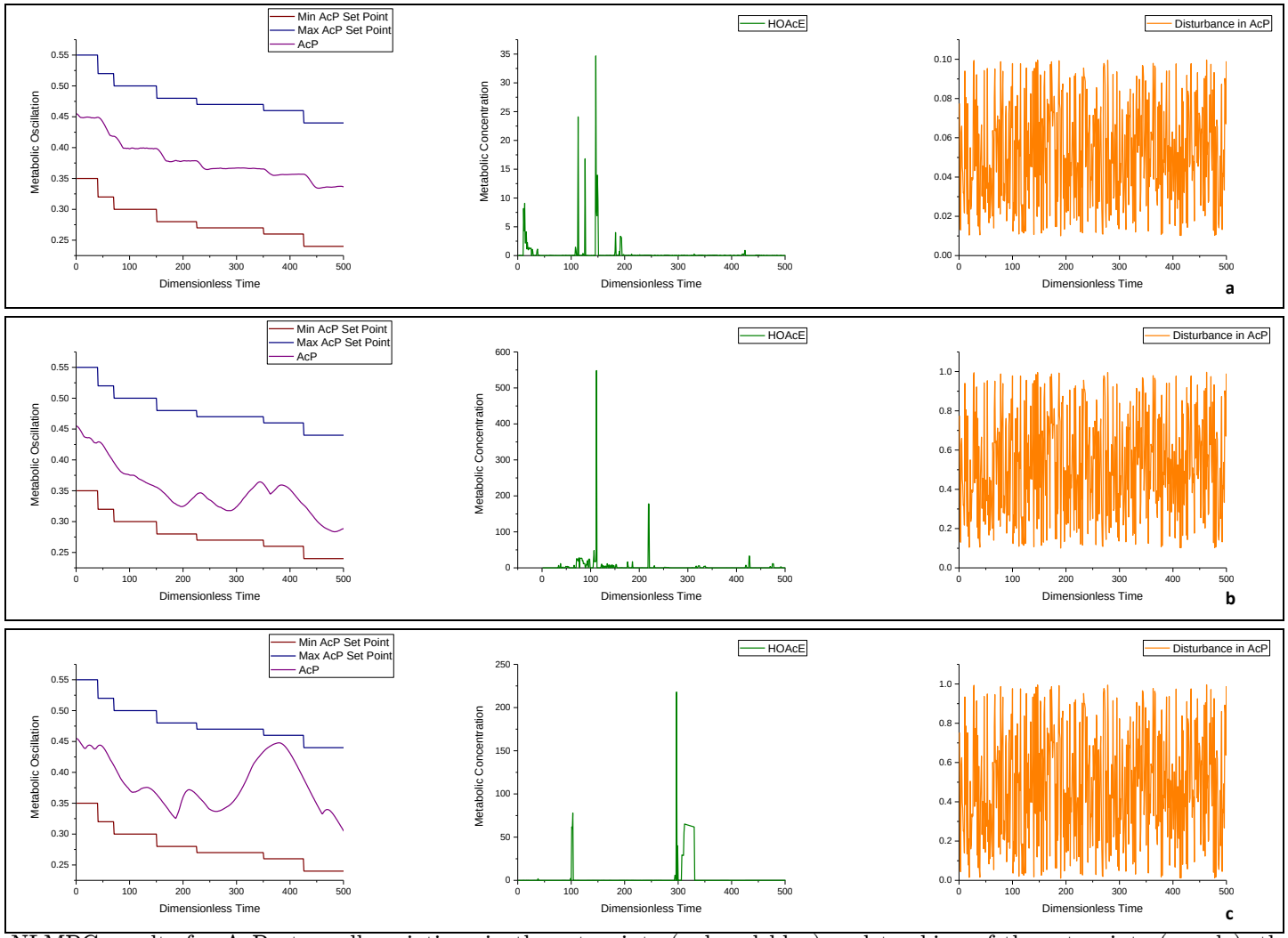

Figure 4.28: Zone NLMPC results for AcP at small variations in the set points (red and blue) and tracking of the set points (purple), the concentration of acetate (HOAcE) required to track the set points and the disturbance (d) added to AcP, where, **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. $Qz = 1$, $Qu = 0$ and $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$ for all cases.

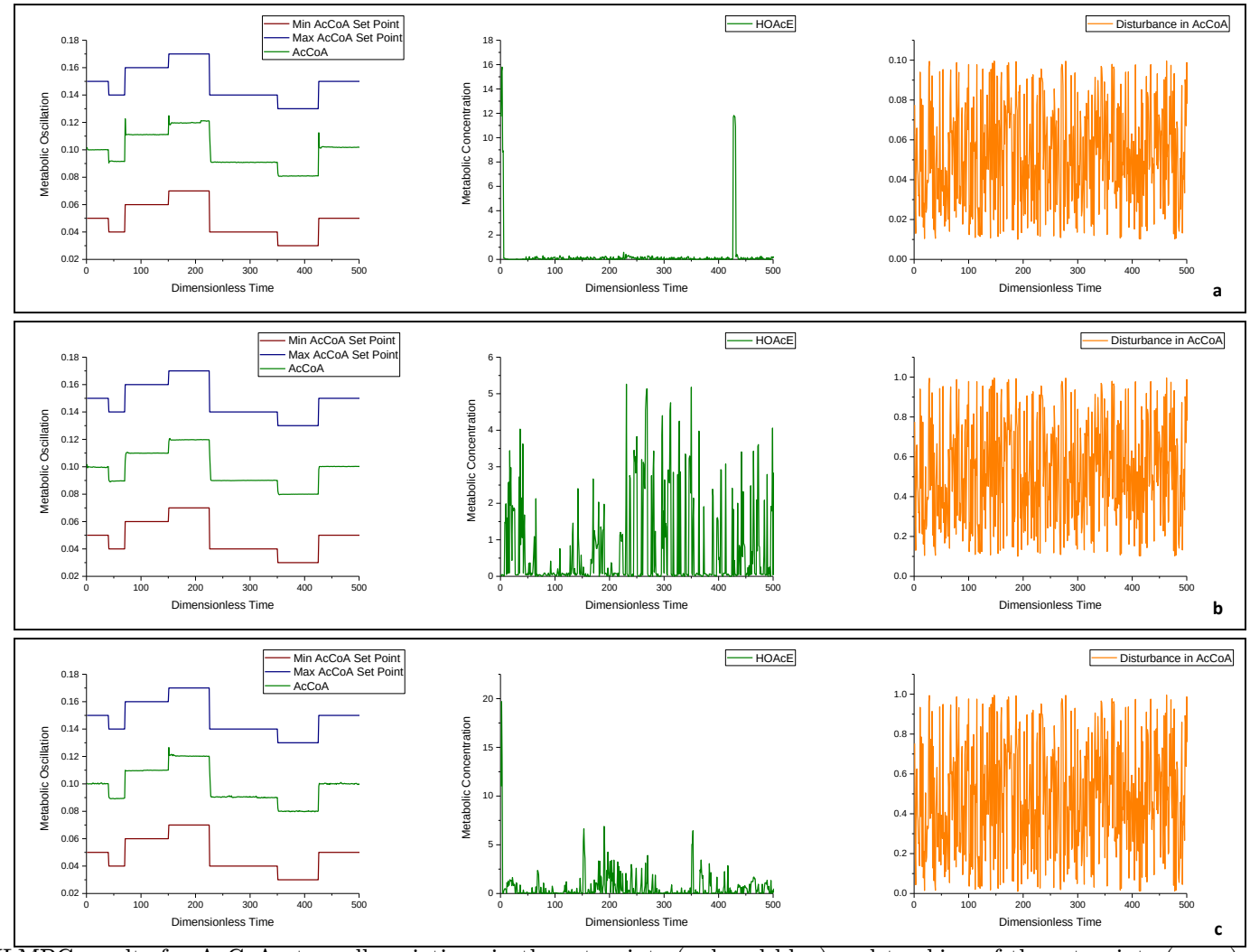

Figure 4.29: Zone NLMPC results for AcCoA at small variations in the set points (red and blue) and tracking of the set points (green), the concentration of acetate (HOAcE) required to track the set points and the disturbance (d) added to AcCoA, where, **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. Qz = 1, Qu = 0 and $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$ for all cases.

4.8 Dual Control (Multiple-Input Multiple-Output) Results

The addition of another input, or control variable, to the model gave rise to a new type of approach for the metabolator control, multiple-input multiple-output model based control (MIMO). Two control variables, Vgly rate and acetate concentration (HOAcE), were utilised to control the system dynamics (Figure 4.30). As shown previously, Figure 4.5, increasing acetate concentration results in a loss of activity for the principal metabolite, AcCoA, which is to be expected as in a highly acidic environment the metabolator would denature. Therefore an investigation into controlling the synergistic relationship of the two metabolite pools is integral to modelling the complete dynamics of the set points.

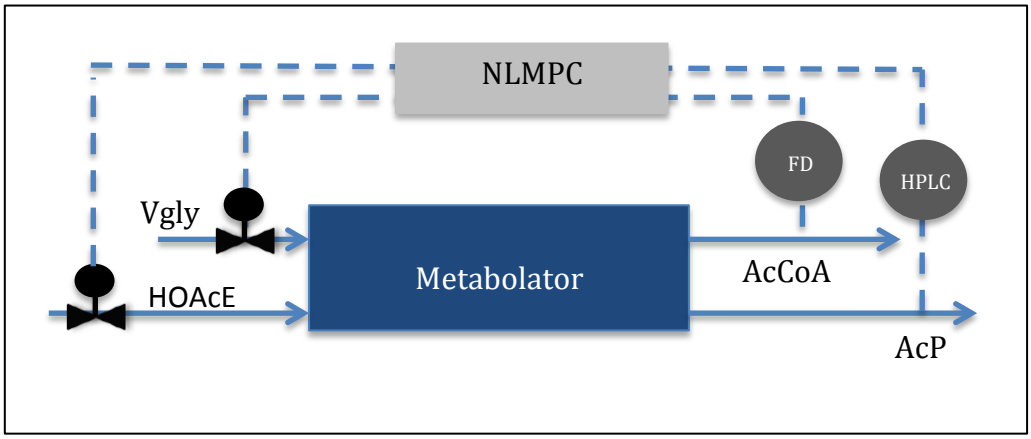

Figure 4.30: Control block showing the metabolator optimised with Vgly and acetate (HOAcE) as the control variables for the state variables AcCoA and AcP. This is an example of Multiple-Input Multiple-Output (MIMO) model based control. Here Vgly can be measured online using fluorescence detection (FD) and HOAcE can be measured using HPLC analysis.

As shown in Figure 4.31 the system is able to track both AcCoA and AcP. There is little violation of the set points and the predicted concentrations of Vgly and acetate compare well with previous results. The set points for AcCoA and AcP reflect the synergy between the two metabolite pools, and a decrease in the concentration of AcCoA results in an increased concentration of AcP and vice versa. This is advantageous as the model now reflects the dynamic biological processes that are occurring within the metabolator with respect to the flux of Vgly and acetate production. There are however factors that are not considered by the model. In order for the model to take into account the true dynamics of the set points it must also include cell reproduction and cell death. These are important factors as they

govern the concentration of each moiety in the system, but are also notoriously difficult to model. Again this is a key investigation that should be performed with the experimentation and these can be measured using either green fluorescent protein embedded in the *E. Coli* genome, or by using a carbon-13 glucose source.

As with previous models, disturbance was also considered for the dual control (MIMO) model (Figure 4.32). In this instance disturbance was considered for both equations 4.1 and 4.2 in order to present a case where disturbance was affecting both metabolite pools. This can be the case biologically whereby the disturbance in AcCoA and AcP can arise from stochasticity in the system.

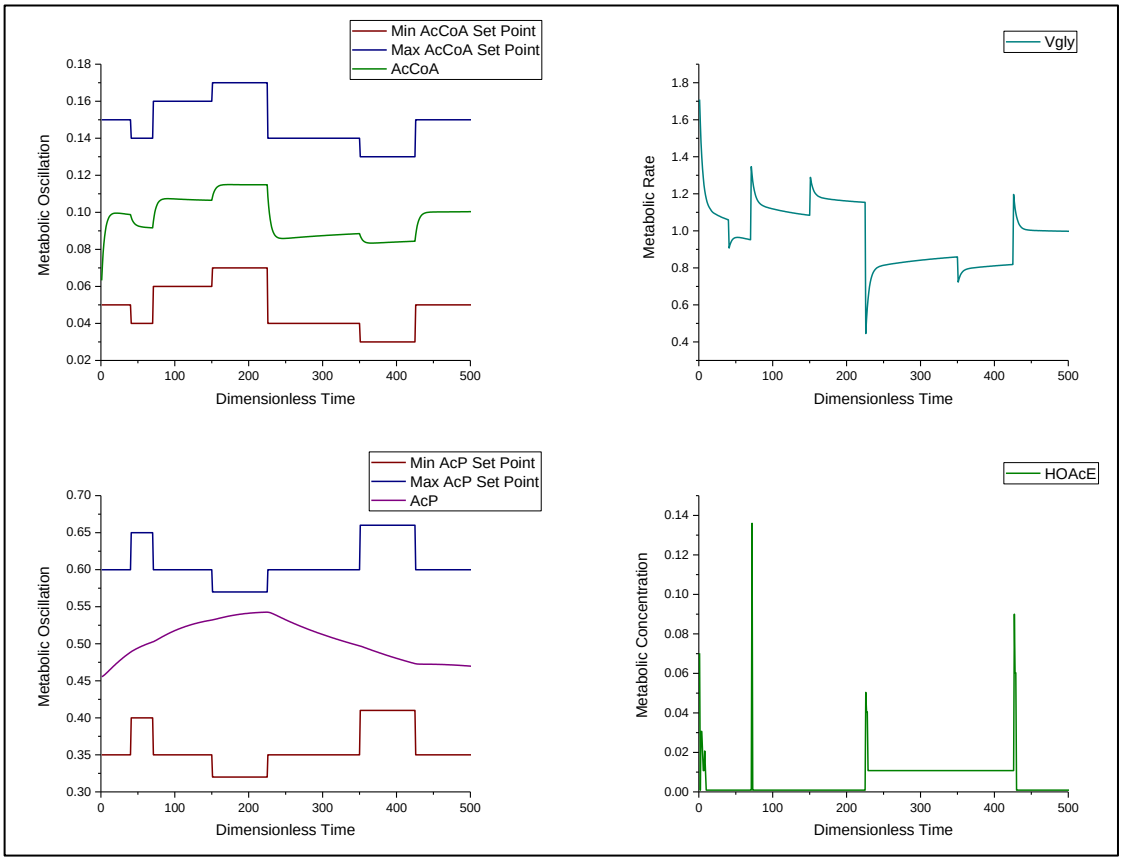

Figure 4.31: Zone NLMPC results for AcCoA at small variations in the set points (blue and red) and tracking of the set points (green), the concentration of Vgly in the system showing the fluctuations required to track the set points of AcCoA, the Zone NLMPC results for AcP at a decreasing concentration, with set points (blue and red) and tracking of the set points (purple) and the concentration of acetate in the system showing fluctuations required to track the set points of AcP. Qz = 1, $1x10^{-9} \leq$ Dvgly $\leq 1x10^{-6}$, $1x10^{-4} \leq$ DHOAcE $\leq 1x10^{-2}$.

When comparing results from Figures 4.31, 4.15 and 4.23 it can be noted that MIMO control of the metabolator yields better results than Single-Input Single-Output (SISO) implementation (as shown in Figures 4.15 and 4.23). In terms of AcP tracking it is noted that MIMO control results in a much better profile of acetate where the concentration levels show much less sharp changes when comparing

results from Figures 4.31 and 4.23, however, the tracking of AcP is lost. The resulting concentration of AcP does not follow the trend of the set points, but does remain within the set zone. This is advantageous as this can be replicated easily in experimentation by altering the concentration of acetate in the cell media. The results in Figure 4.31 show an optimum performance of the metabolator and highlight its synergistic relationship between the two metabolite pools. An increase in the activity of AcCoA results in a decrease in activity for AcP and vice versa (Figure 4.1).

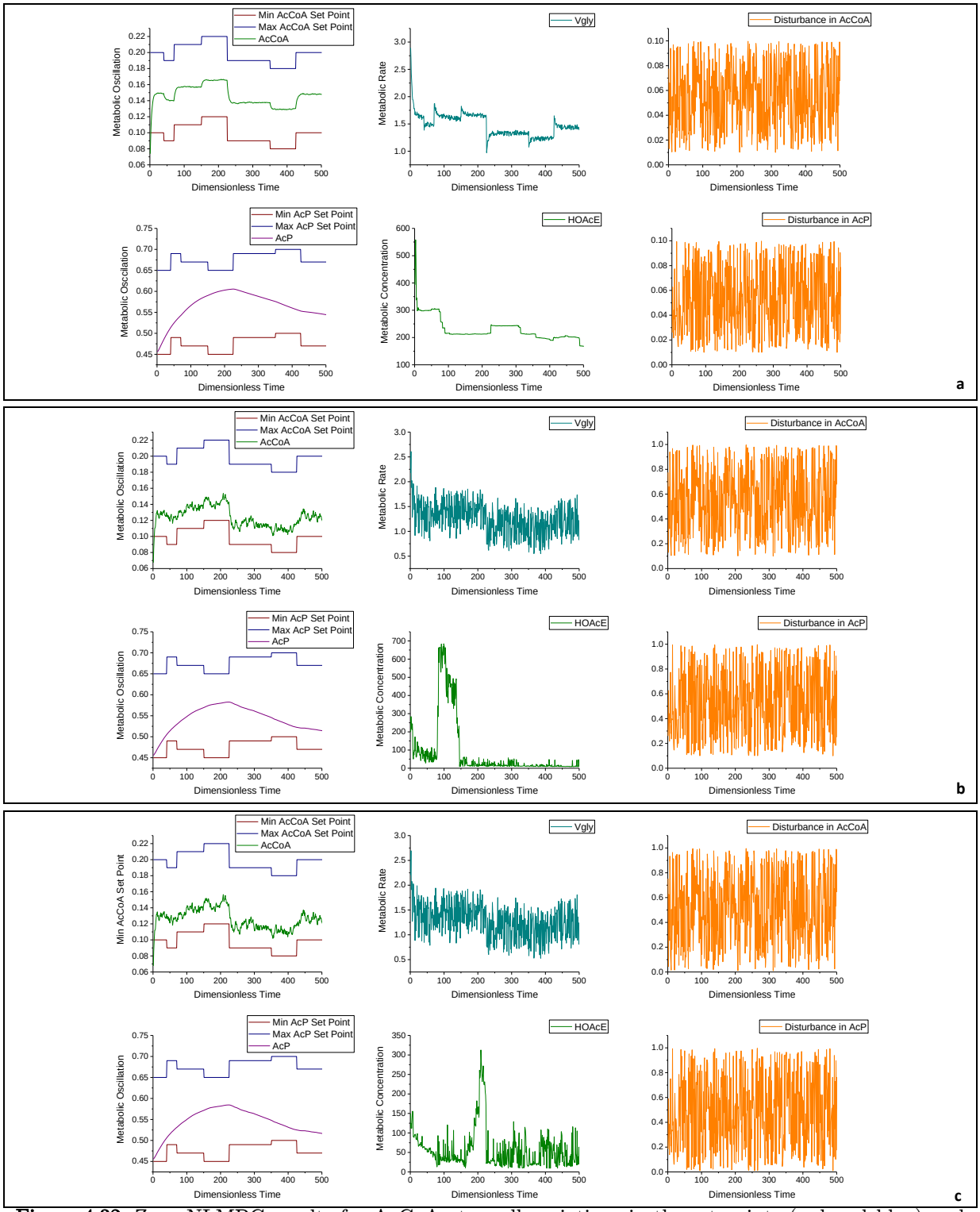

Figure 4.32: Zone NLMPC results for AcCoA at small variations in the set points (red and blue) and tracking of the set points (green), the rate of Vgly required to track the set points and the disturbance (d) added to AcCoA, where d is a random disturbance with uniform distribution, the Zone NLMPC results for AcP at small variations in the set points (red and blue) and tracking of the set points (purple), the concentration of HOAcE required to track the set points and the disturbance (d) added to AcP, where d is a random disturbance with uniform distribution. **a.** $0.01 \leq d \leq 0.1$, **b.** $0.1 \leq d \leq 1$ and **c.** $0.01 \leq d \leq 1$. $Qz = 1$, $Qu = 0$, $1x10^{-9} \leq Dvgly \leq 1x10^{-6}$ and $1x10^{-4} \leq DHOAcE \leq 1x10^{-2}$ for all cases.

The results in Figure 4.32 show that the increase in disturbance affects mainly AcCoA within the system, where frequent sharp changes are present in its optimal activity. However, it is noted that the system does not violate the given bounds of the Zone NLMPC in both AcCoA and AcP, which shows that the system can effectively track even in the presence of disturbance. This is profound as the disturbance added to the system is up to 1000% of the nominal concentrations of AcCoA and AcP, and this is typically higher than what would normally be considered as disturbance for the metabolator (Fung *et al.*, 2005).

4.9 Concluding Remarks

The metabolator has been successfully modelled using various approaches, ranging from ANN-RK4 validation to Zone Control with multiple inputs (MIMO). The system has been shown to react to disturbance, and still adhere to the control systems applied to it. This level of control is profound as it allows the biological system to alter its response depending on the user specified constraints placed on it. Results from all investigations showcase how adaptable the metabolator is and how responsive it can be towards subtle changes in design, such as altering the control set point or having large disturbance present in the system. Although some results do violate the control system, such as results in Figures 4.13c and 4.26c when using a single control set point under disturbance, this can be mitigated using Zone NLMPC as shown in Figures 4.18c and 4.28c for the same conditions. Furthermore control using just one entity, Vgly, is novel for this type of system as it typically will have other feed sources. Control using multiple inputs, Vgly and acetate (Figures 4.31 and 4.32), shows that the system is able to be controlled using both the positive feed-forward and negative feedback nature of the metabolator.

One key element of NLMPC is that one can measure or estimate the state variables at each time interval and then use the data obtained to compute the control variable at each time interval. In an experimental setup this would involve online detection of the concentrations of AcCoA and AcP to then estimate the levels of Vgly and acetate within the system. This can be achieved in many different ways depending on the experimental setup. If one takes Vgly to be glucose, which is a correct assumption as glucose is part of the flux that goes into the first metabolite pool (Figure 4.2), then a simple glucose detector, such as the abcam glucose assay kit (abcam n.d.), can be used in the media around the cells. This will indicate the level of glucose and from there it can be calculated how much glucose has been taken into the cells, therefore increasing AcCoA concentration. Detection of AcCoA

can also be achieved using reverse phase ion-exchange chromatography as shown in research from Yamato *et al.* (1992). The method described in this research utilises an immobilised enzyme reactor in the post column which specifically detects AcCoA. This can then be used to calculate the concentration of AcCoA in the solution. In terms of an experimental setup with the metabolator this will involve taking a sample and centrifuging the cells down to lyse their internal contents, then testing this solution for AcCoA. Unfortunately there is not a way, to tis researches knowledge, of methods that can detect the levels of AcCoA without causing harm to the cell, and therefore whilst detection of AcCoA can be performed it will not be completely on-line, due to the nature of taking a sample to destroy the cells. Research from Klein *et al.* (2007) showcased a thin layer chromatography (TLC) approach for the detection of AcP through quantifying the amount of phosphorous taken up by the cell. This again involved centrifuging down the sample and snap freezing to stop any further reaction. This detection method is useful, but again causes damage to the cell. These methods are as close to on-line detection methods that one can get without altering the metabolator or changing reactions by adding new solutions. The detection of AcCoA and AcP will need to be carefully monitored when performing experiments on the metabolator in future work.

5. A Comparative Analysis of Nonlinear Solution Techniques for Dynamic Problems

As mentioned previously (Chapter 3) dynamic models can be converted to algebraic equations and solved using GAMS. However this is one of many techniques that can be applied to simulate and optimise dynamic process models. A method that can be applied is Orthogonal Collocation on Finite Elements (OCFE) (Carey and Finlayson, 1975), which employs a simultaneous solution strategy for dynamic optimisation. Unlike neural networks which transform the differential equations through their hidden layer and solve for the domain, OCFE divides the domain into smaller subdomains, termed finite elements. The residuals are set to zero at the collocation points interior to the elements. This ensures continuation of the trial function between each node and the boundary points. The process can be considered as an extension to the double collocation method proposed by Villadsen and Sorensen (1969) where again the domain is divided into elements, but in the case of nonlinear equations the algebraic equations involve terms in one of the finite elements and not for the whole domain. The general outline of OCFE is presented in Chapter 3, Figure 3.4. It is best used for problems that have steep gradients and shows best applicability on time dependent problems. It can be seen as a more complex solution methodology when compared to other techniques using a combination of ODEs and algebraic equations such as RK4. This Chapter will look into investigating which method, OCFE or the ANN-RK4 framework, gives the most optimal results for synthetic biology systems and control problems. The main benefits of each technique will be discussed and conclusions will be drawn based on performance of the models and results. This chapter will show the strengths and weaknesses of the developed meshless ANN framework and will ultimately discuss its validity as a viable solution method for nonlinear problems. It will also solidify the use of the framework for the prominent case study, the Metabolator (Fung *et al.*, 2005).

5.1 Uses of OCFE

Orthogonal collocation on finite elements (OCFE) provides a computationally efficient and accurate approach to solve both ODEs and partial differential equations

(PDEs) as presented in research by Babuska *et al.* (2007). Therefore it has applicability in many different dynamic processes ranging from biological systems (Rogers and McCulloch, 1994) to fuel cells (Ziogou *et al.*, 2013), where NLMPC was combined with OCFE to provide an on-line user interface to control power generation of a polymer electrolyte membrane fuel cell (PEMFC). Collocation methods can also be combined with other solution techniques that aid in optimising highly nonlinear systems with irregular domains, such as biological systems and processes as presented in work by Saucerman and McCulloch (2004) and can also be used to approximate ODEs in control problems (Kawathekar *et al.*, 2007). Furthermore collocation methods can be imposed in systems that have previously shown great computational expense, such as the pseudo two-dimensional (P2D) model for lithium ion cells (Doyle *et al.*, 1993; Fuller *et al.*, 1994) as shown by Cai and White (2012).

The following section (Section 5.2) will present a number of ways in which OCFE can aid in modelling dynamic systems, with most representation showing that it is a fast and efficient solution technique that gives highly accurate results. The methodology of OCFE will be applied to a number of case studies and the merits of this technique will be discussed and compared to the developed ANN-RK4 method discussed in Chapters 3 and 4.

5.2 Comparative case studies for OCFE and the ANN-RK4 methods

5.2.1 An Isothermal CSTR Example

An isothermal CSTR model (Sistu and Bequette, 1995) was considered for simulation using the ANN framework and OCFE (see Chapter 3, Section 3.5.1). For this example with the ANN formulation the prediction horizon is set at 0.002 hours, the horizon length is 5, $\gamma_1 = \gamma_2 = 1000$ and $\gamma_3 = 1$. The results for the ANN and OCFE implementations for this model are presented in Figure 5.1. From the results in Figure 5.1 it is shown that both ANN and OCFE based implementations of NLMPC are able to track the set points easily and maintain a steady state. The total run times for each show that the OCFE model takes less than half the time to solve. The average number of CPU seconds for each time interval for the ANN model is 0.424, whereas with the OCFE model it is 0.049.

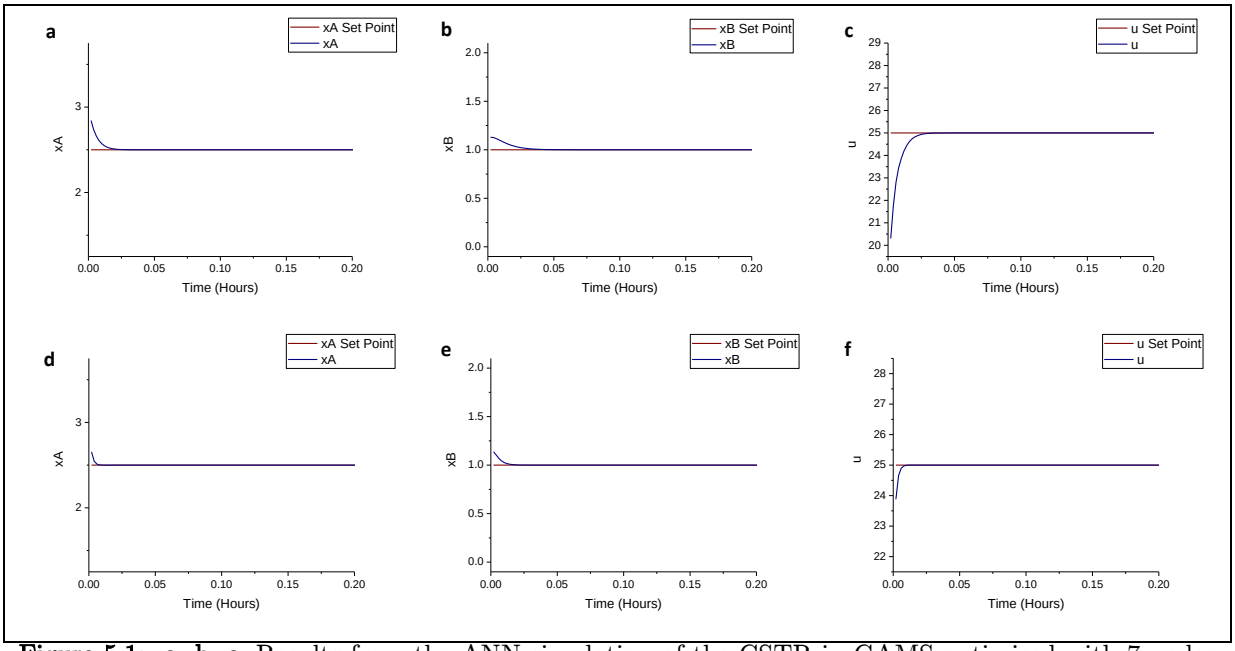

Figure 5.1: a. b. c. Results from the ANN simulation of the CSTR in GAMS optimised with 7 nodes in the hidden layer and a total run time of 49 seconds for 100 iterations, **d. e. f.** results for the OCFE simulation of the CSTR in GAMS optimised with 3 collocation points and a total run time of 15 seconds for 100 iterations. CPU: Intel Core™2 Duo E7400 2.8GHz.

5.2.2 Lower order distillation column

A distillation column described by Rasmussen and Jorgensen (1999) and Prasad and Bequette (2003) was chosen as the next example problem. The distillation tower consists of five trays which includes the total condenser and reboiler. The system of ODEs governing this system is given by equations 5.1-5.5.

$$\frac{dx_1}{dt} = \frac{1}{H_r}\left(\left(L + L_f\right)(x_2 - x_1) + V\left(x_1 - \left(\frac{\alpha x_1}{1 + (\alpha - 1)x_1}\right)\right)\right) \tag{5.1}$$

$$\frac{dx_2}{dt} = \frac{1}{H_t}\left(\left(L + L_f\right)(x_3 - x_2) + V\left(\frac{\alpha x_1}{1 + (\alpha - 1)x_1} - \left(\frac{\alpha x_2}{1 + (\alpha - 1)x_2}\right)\right)\right) \tag{5.2}$$

$$\frac{dx_3}{dt} = \frac{1}{H_t}\left(\left(L_f x_f + L x_4\right) - \left(L + L_f\right)(x_3) + V\left(\frac{\alpha x_2}{1 + (\alpha - 1)x_2} - \left(\frac{\alpha x_3}{1 + (\alpha - 1)x_3}\right)\right)\right) \tag{5.3}$$

$$\frac{dx_4}{dt} = \frac{1}{H_t}\left(L(x_5 - x_4) + V\left(\frac{\alpha x_3}{1+(\alpha-1)x_3} - \left(\frac{\alpha x_4}{1+(\alpha-1)x_4}\right)\right)\right) \tag{5.4}$$

$$\frac{dx_5}{dt} = \frac{1}{H_t}\left(V\left(\left(\frac{\alpha x_3}{1+(\alpha-1)x_3}\right) - x_5\right)\right) \tag{5.5}$$

where x_1, x_2, x_3, x_4 and x_5 are the state variables of the column and represent the liquid mole fraction of the light component at the reboiler, second tray, third tray (feed), fourth tray and the condenser respectively. The molar holdup on the reboiler, trays and condenser are represented by H_r, H_t and H_c. Feed flow rate into the column is shown by L_f and the light component feed composition is given by x_f. Finally L and V represent the liquid reflux flow and vapour flow rate back into the column respectively. Parameters used for the model are presented in table 5.1.

Table 5.1: Model parameters for the distillation column (Prasad and Bequette, 2003).

Parameter	Value
H_c	30 mol
H_r	30 mol
H_t	20 mol
α	5
L_f	10.0 mol min^{-1}
x_f	0.5
L	27.3755 mol min^{-1}
V	32.3755 mol min^{-1}

The system was then modelled using the ANN and OCFE frameworks. The control problem was then implemented whereby the value for x_5, the distillate, was set and controlled using the vapour flow rate (Figure 5.2). A schematic of the column is presented in Figure 5.3. The results for the ANN and OCFE implementation are presented in Figures 5.4 and 5.5, and the implementation of both formulations using NLMPC are presented in Figure 5.6.

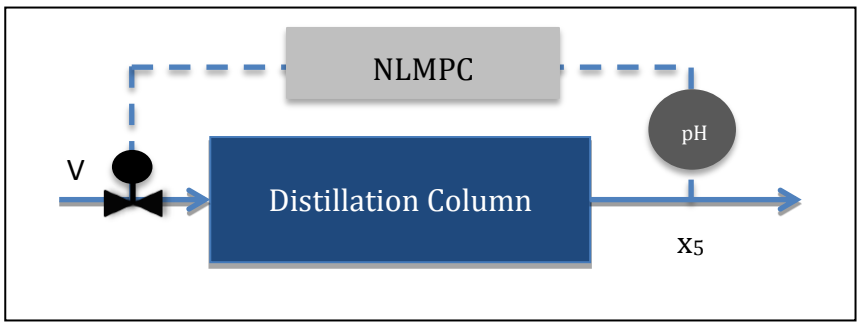

Figure 5.2: Control block showing the lower order distillation column optimised with V (vapour flow rate) as the control variable for the state variable x_5, which is the mole fraction in the condenser and its concentration can be measured using a pH meter.

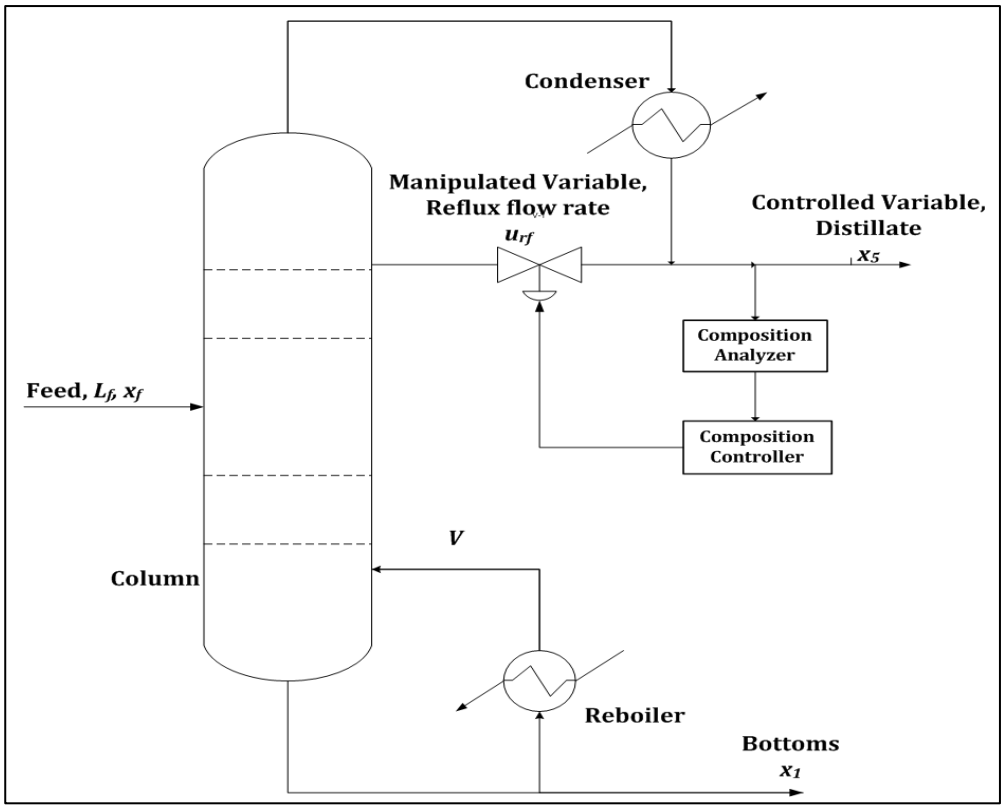

Figure 5.3: Schematic of the distillation column

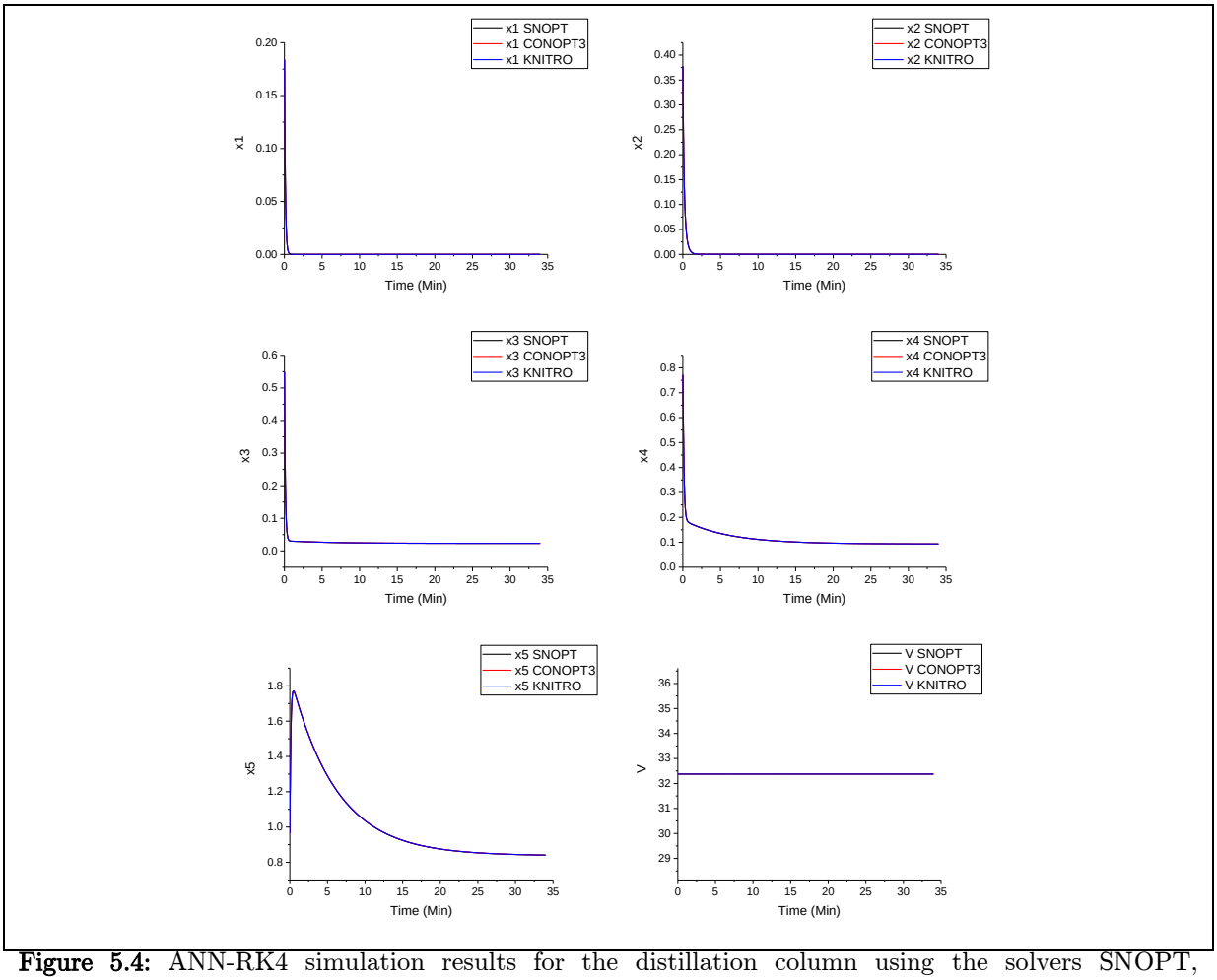

Figure 5.4: ANN-RK4 simulation results for the distillation column using the solvers SNOPT, CONOPT3 and KNITRO in GAMS. Total CPU time for SNOPT = 10 minutes 59 seconds, CONOPT3 = 19 minutes 25 seconds and KNITRO = 26 minutes 8 seconds for 3400 iterations, step size = 0.01. CPU: Intel Core™2 Duo E7400 2.8GHz.

As presented in Figures 5.4 and 5.5 the results for each solver provide the exact same solution. For the ANN formulation (Figure 5.4) SNOPT outperforms the other solvers in terms of CPU time, and in the OCFE formulation (Figure 5.5) SNOPT also outperforms the others. It is apparent that generally the OCFE formulation is much faster than the ANN. Further analysis of the system looks at incorporating control, where the level of x_5 is set at 0.98. The results for the ANN-NLMPC and OCFE-NLMPC control simulations are presented in Figure 5.6.

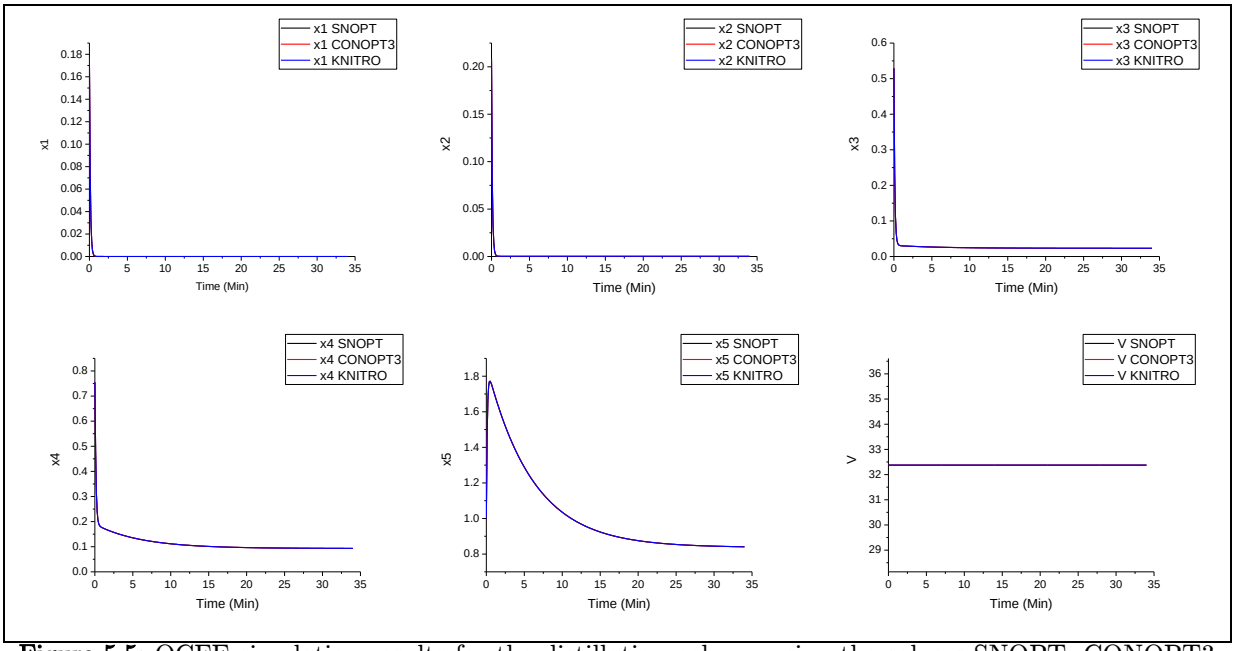

Figure 5.5: OCFE simulation results for the distillation column using the solvers SNOPT, CONOPT3 and KNITRO in GAMS. Total CPU time for SNOPT = 6 minutes 37 seconds, CONOPT3 = 7 minutes 58 seconds and KNITRO = 10 minutes 22 seconds for 3400 iterations, step size = 0.01. CPU: Intel Core™2 Duo E7400 2.8GHz.

From the results in Figure 5.6 it is noted that both formulations (ANN and OCFE) are able to track the set point for x_5, with the OCFE-NLMPC formulation performing slightly faster for SNOPT and KNITRO and the ANN-NLMPC formulation performing better for CONOPT3. The OCFE-NLMPC results also show that there is a deviation from the set point in the order of 5 x 10^{-4} and as such the system does not actually achieve the set point. Taking this into consideration both formulations are comparable for model based control, but as there is relatively little difference between the CPU times it can be argued that the ANN-NLMPC formulation is the better choice.

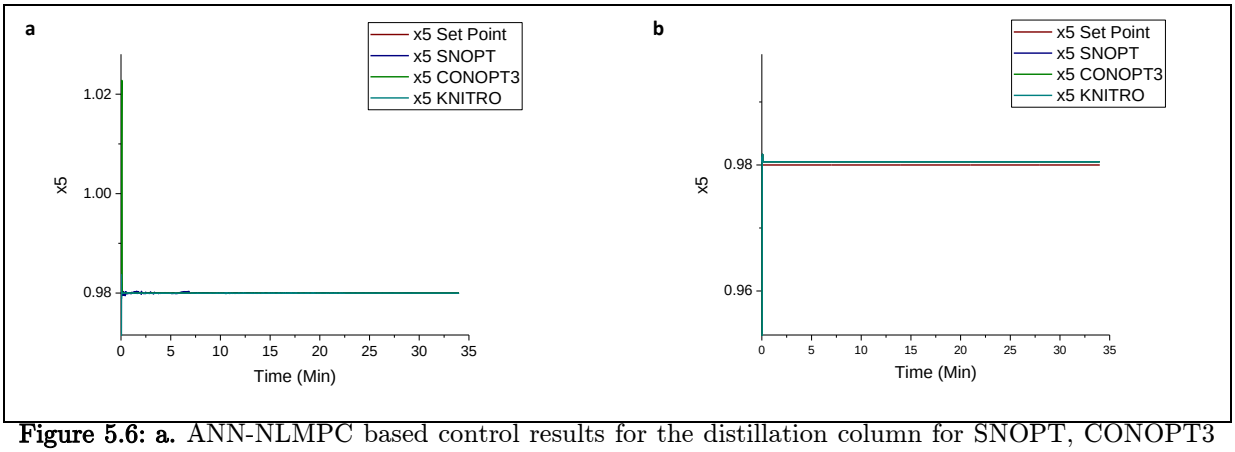

Figure 5.6: a. ANN-NLMPC based control results for the distillation column for SNOPT, CONOPT3 and KNITRO solvers in GAMS. Total CPU time for SNOPT = 10 minutes 20 seconds, CONOPT3 = 9 minutes 32 seconds and KNITRO = 18 minutes 42 seconds for 3400 iterations, step size = 0.01, **b.** OCFE-NLMPC based control results for the distillation column for SNOPT, CONOPT3 and KNITRO in GAMS. Total CPU time for SNOPT = 8 minutes 13 seconds, CONOPT3 = 10 minutes 36 seconds and KNITRO = 18 minutes 20 seconds for 3400 iterations, step size = 0.01. CPU: Intel Core™2 Duo E7400 2.8GHz.

The next problem deals with a more complex system of ODEs that govern a biological system termed the metabolator (Fung *et al.*, 2005), previously shown in Chapter 4.

5.2.3 The Metabolator

The metabolator (Fung *et al.*, 2005) which was simulated and controlled using ANN in Chapter 4 is now revisited and modelled with OCFE based techniques. The results for both frameworks, ANN and OCFE, are compared using three solvers in GAMS: SNOPT, CONOPT3 and KNITRO (for more information about how these solvers function the reader is referred to Chapter 3, Section 3.6.4). The control problem remains the same as previous work in this thesis and results for the NLMPC approach controlling AcCoA are presented in Figure 5.7. It can be seen that both the ANN and OCFE formulations are able to adhere to the set points, however the computational effort for the OCFE model is greater than the ANN model. This difference in time shows that whilst the OCFE formulation is capable of modelling problems with less nonlinearity faster than the ANN formulation, it lacks the ability to model highly nonlinear systems as easily. However, the adherence to the set points is closer than the ANN formulation. Whilst the results for the ANN from CONOPT3 and KNITRO solvers deviate from the set points, the results from SNOPT adhere much more closely. This shows that whilst both generally show some

adherence to the set points, the ANN formulation can be considered better for SNOPT solver. The increased computational effort can mean that using the OCFE-NLMPC formulation, especially when using the KNITRO solver would mean that although the control results are more favourable, the ANN-NLMPC formulation can also achieve adequate results and therefore could be used as an alternative. Overall given the information gained from these results it can be said that the ANN-NLMPC formulation solved using SNOPT is the best choice in terms of speed and accuracy, especially if one is looking at gaining an optimal solution quickly.

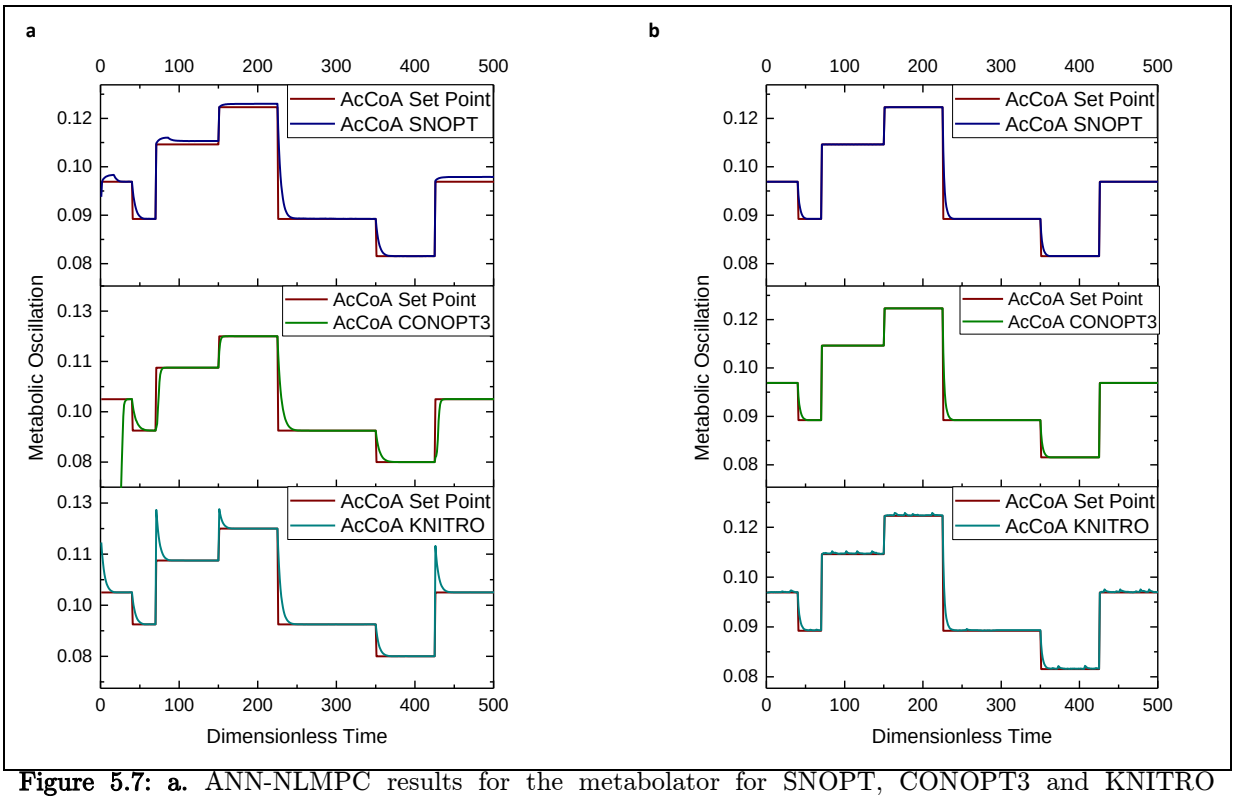

Figure 5.7: a. ANN-NLMPC results for the metabolator for SNOPT, CONOPT3 and KNITRO solvers in GAMS. Total CPU time for SNOPT = 1 minute 41 seconds, CONOPT3 = 1 minute 58 seconds and KNITRO = 4 minutes 34 seconds for 500 iterations, step size = 0.01, **b.** OCFE-NLMPC results for the metabolator for SNOPT, CONOPT3 and KNITRO in GAMS. Total CPU time for SNOPT = 4 minutes 46 seconds, CONOPT3 = 1 minutes 49 seconds and KNITRO = 6 minutes 3 seconds for 500 iterations, step size = 0.01. CPU: Intel Core™2 Duo E7400 2.8GHz.

5.2.3.1 Zone control

With NLMPC the objective function has one set of values to track, however one can set a 'zone' between two sets of values for the system performance, and this is termed as zone model predictive control (Zone MPC). The set zone or band allows the system to perform freely within the controlled region, which can have beneficial

applications for systems like the automated diabetes control monitor (Grosman *et al.*, 2010) as there is less restriction on the system performance which the same for single set control. With the metabolator a zone can be set so that the system can oscillate within the region and therefore alter the production of acetate by manipulating glycolytic flux (sugars, fatty acids and glycerol – Vgly). Figure 5.8 presents the results for the Zone MPC formulation, where both the ANN and OCFE formulations are able to adhere to the set zone. There is little variation in the results for different solvers, except when looking at the response rate of Vgly. For KNITRO in the OCFE-Zone MPC formulation the response rate fluctuates rapidly in order to produce the desired response within the set zone. Also for both the ANN and OCFE formulations it is the solver that takes the longest to solve. Again it can be noted that the ANN formulation is faster overall, with the best results achieved using the SNOPT solver. This indicates that the OCFE takes longer for both single set point control (NLMPC) (Chapter 3, Section 3.6), and Zone NLMPC (Chapter 3, Section 3.6.2) for this example. The following section will consider disturbance in the system analyse its effect for both single set control and dual set (zone) control profiles.

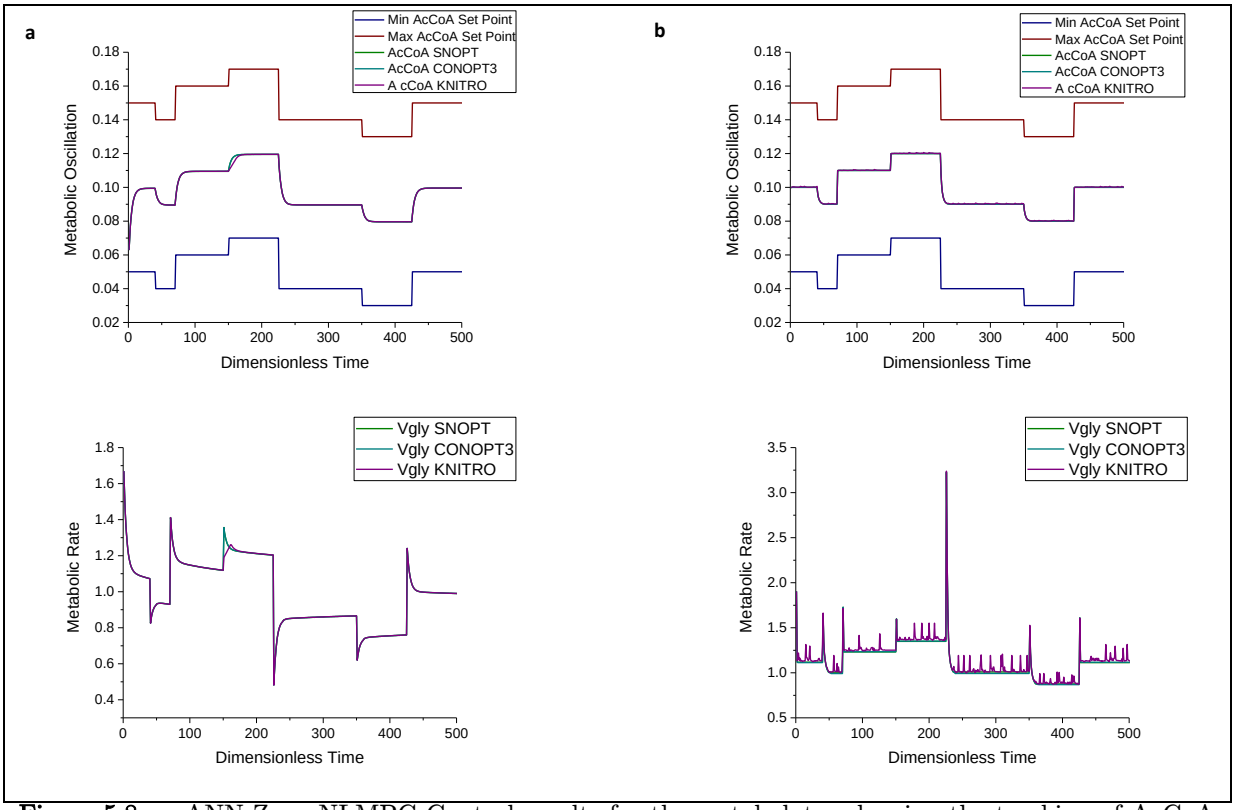

Figure 5.8: a. ANN-Zone NLMPC Control results for the metabolator showing the tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. Total CPU time for SNOPT = 1 minute 43 seconds, CONOPT3 = 1 minute 23 seconds and KNITRO = 5 minutes 29 seconds for 500 iterations, step size = 0.01, **b.** OCFE-Zone NLMPC control results for the metabolator showing tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. Total CPU time for SNOPT = 1 minute 49 seconds, CONOPT3 = 2 minutes 50 seconds and KNITRO = 5 minutes 37 seconds for 500 iterations, step size = 0.01.

5.2.3.2 Control with disturbance

In the previous Section the metabolator was modelled as a deterministic system with no disturbances. However, its action *in vivo* can involve disturbances from factors like cell death, temperature and pH. These factors can be determined experimentally, quantified and incorporated in the model in order to achieve a more biologically feasible outcome. Within their work, Fung *et* al (2005) explained how stochastic disturbance is present in the system and explained the discrepancies they noticed between the simulated and experimental results. It was noted that this had great effect on the three proteins LacI, Acs and Pta. As such a uniform disturbance (d) over a range was added to the RHS of equation governing AcCoA concentration (Chapter 4, Equation 4.1) and the results were analysed for zone control. In this Section disturbance for the case of single set-point tracking is first considered and

the results are presented in Figures 5.9 and 5.10. Disturbance is then considered for the case where Zone NLMPC strategies are employed and the results for this investigation are presented in Figures 5.11 and 5.12. For all simulations (Figures 5.9-5.12) a restriction on the velocity of Vgly was added, where $1 \times 10^{-6} \leq DVgly \leq 10$.

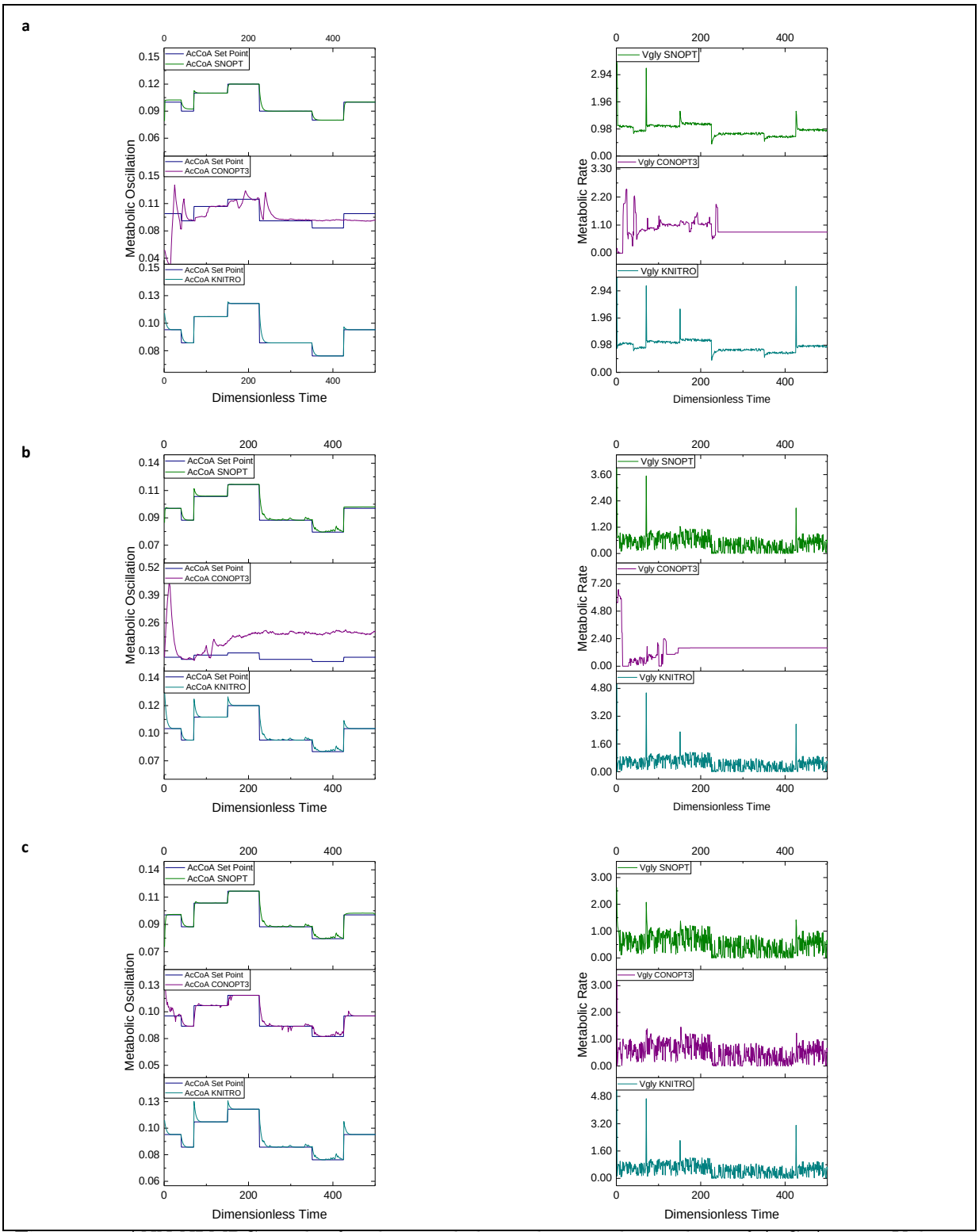

Figure 5.9: ANN-NLMPC results for the metabolator showing the tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. **a.** total CPU time (hours) for SNOPT = 00:02:18, CONOPT3 = 00:03:20 and KNITRO = 00:05:40, **b.** total CPU time (hours) for SNOPT = 00:18:56, CONOPT3 = 00:02:56 and KNITRO = 00:04:21, **c.** total CPU time (hours) for SNOPT = 00:03:29, CONOPT3 = 00:05:28 and KNITRO = 00:07:58. All simulations run for 500 iterations, step size = 0.01, $0.01 \leq d \leq 1.0$ for all simulations.

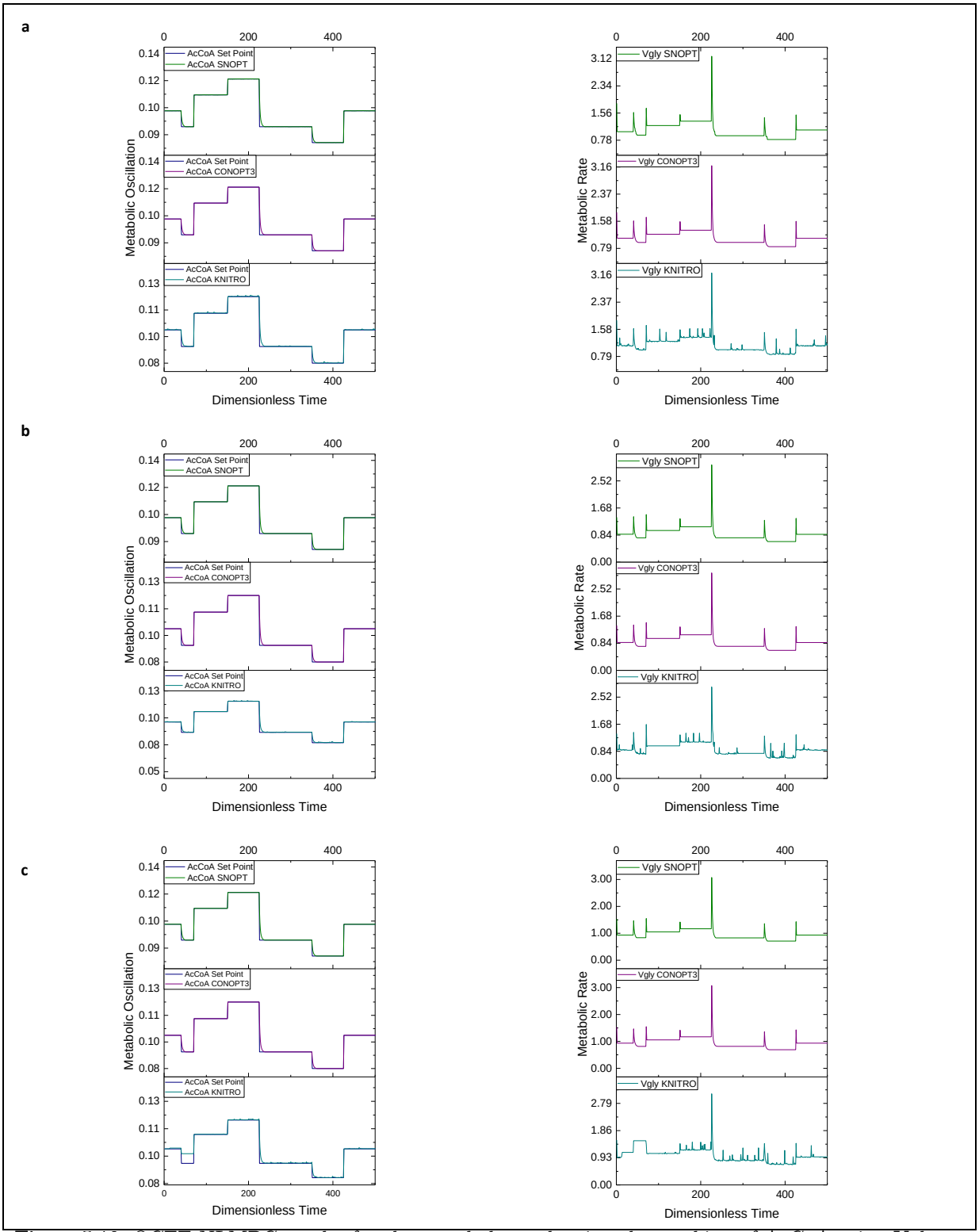

Figure 5.10: OCFE-NLMPC results for the metabolator showing the tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. **a.** total CPU time (hours) for SNOPT = 00:07:34, CONOPT3 = 00:01:41 and KNITRO = 00:05:06, **b.** total CPU time (hours) for SNOPT = 00:04:05, CONOPT3 = 00:01:49 and KNITRO = 00:06:21, **c.** total CPU time (hours) for SNOPT = 00:04:25, CONOPT3 = 00:02:53 and KNITRO = 00:06:14. All simulations run for 500 iterations, step size = 0.01, $0.01 \leq d \leq 1.0$ for all simulations.

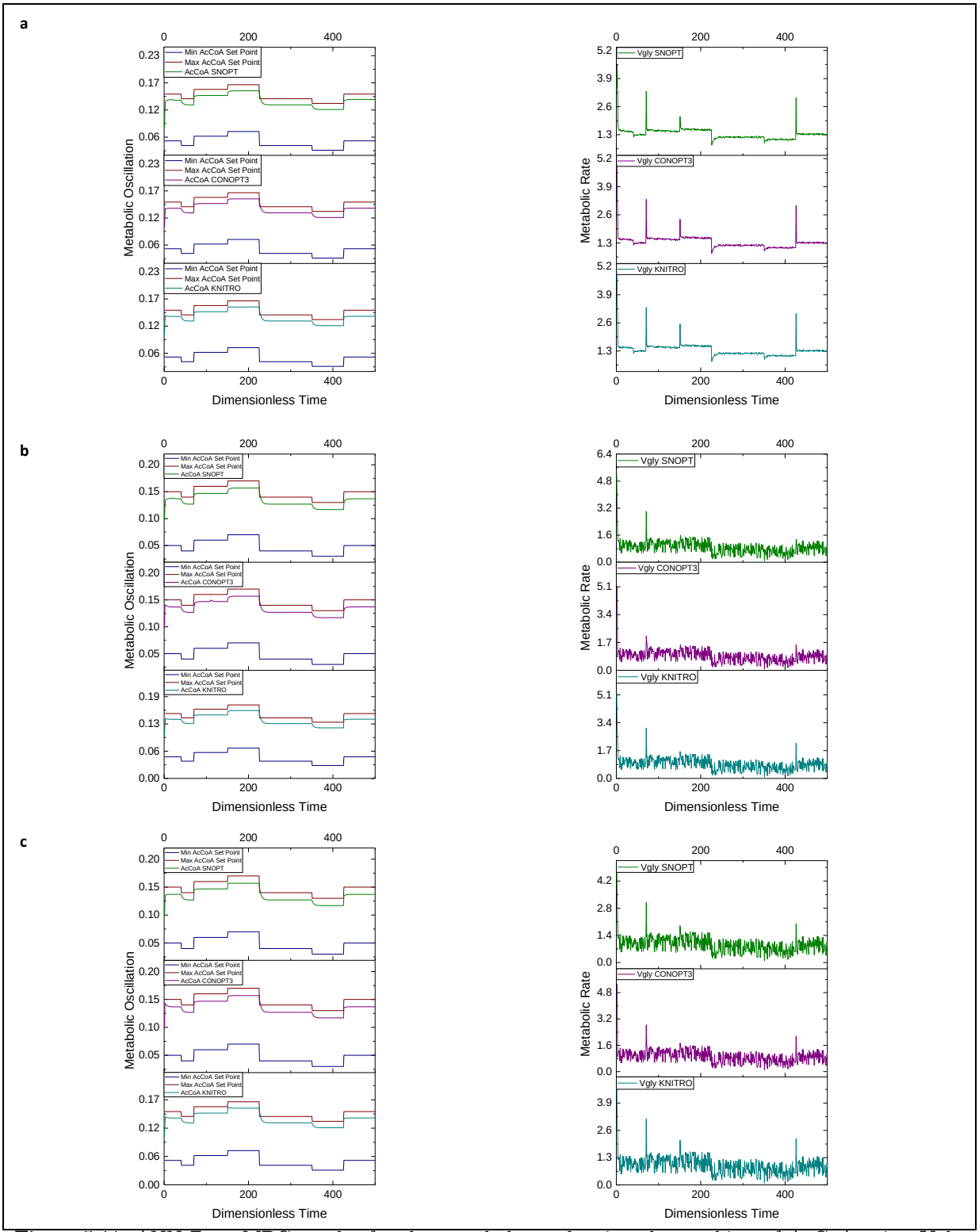

Figure 5.11: ANN-Zone MPC results for the metabolator showing the tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. **a.** total CPU time (hours) for SNOPT = 00:01:59, CONOPT3 = 00:05:20 and KNITRO = 00:08:12, **b.** total CPU time (hours) for SNOPT = 00:02:44, CONOPT3 = 00:03:21 and KNITRO = 00:08:36, **c.** total CPU time (hours) for SNOPT = 00:02:25, CONOPT3 = 00:04:42 and KNITRO = 00:08:47. All simulations run for 500 iterations, step size = 0.01, $0.01 \leq d \leq 1.0$ for all simulations.

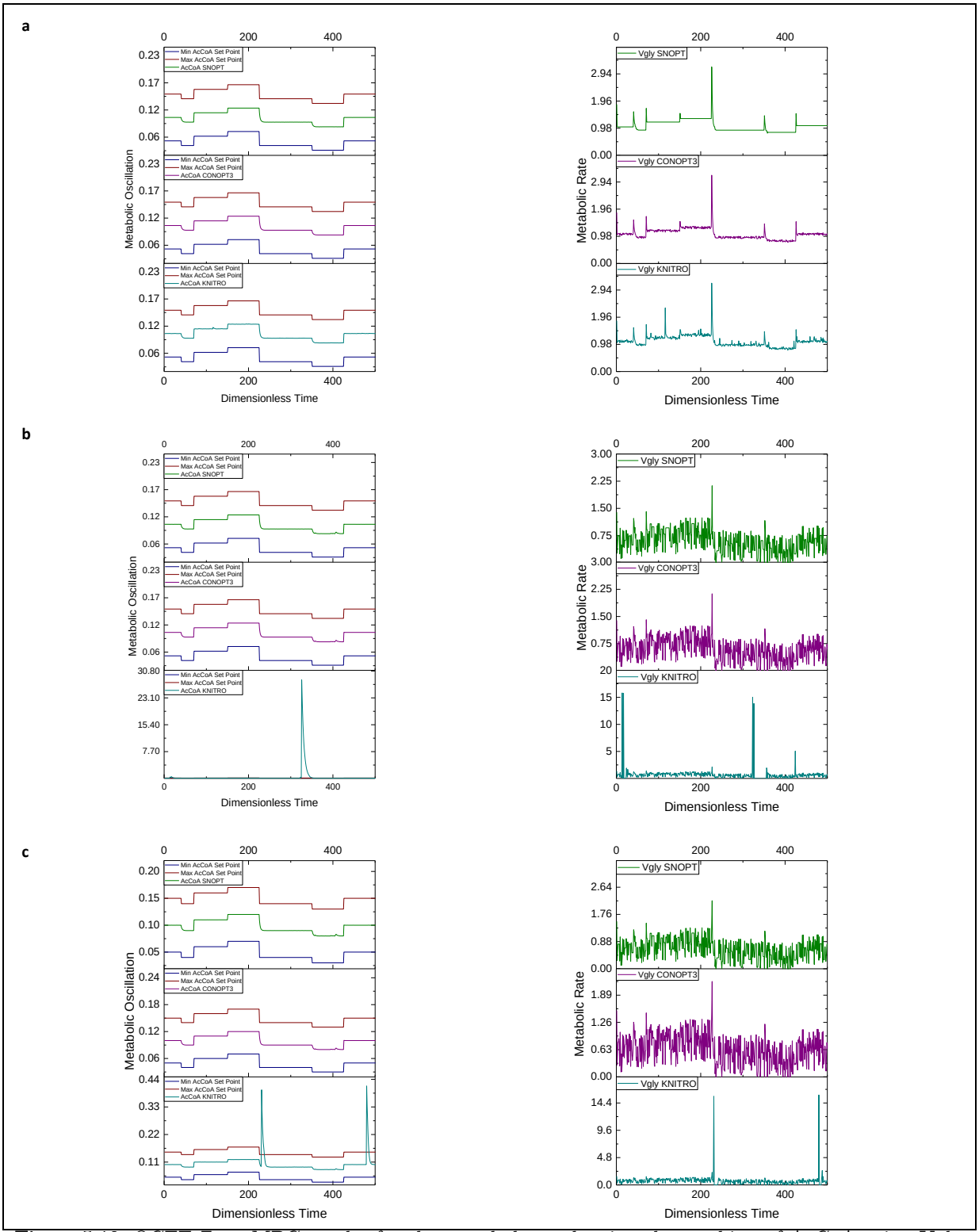

Figure 5.12: OCFE-Zone MPC results for the metabolator showing the tracking of AcCoA using Vgly as the control variable for SNOPT, CONOPT3 and KNITRO solvers in GAMS. **a.** total CPU time (hours) for SNOPT = 00:03:56, CONOPT3 = 00:02:26 and KNITRO = 00:06:29, **b.** total CPU time (hours) for SNOPT = 00:02:39, CONOPT3 = 00:02:59 and KNITRO = 00:16:53, **c.** total CPU time (hours) for SNOPT = 00:02:22, CONOPT3 = 00:02:53 and KNITRO = 00:10:02. All simulations run for 500 iterations, step size = 0.01, $0.01 \leq d \leq 1.0$ for all simulations.

The results presented in Figure 5.9 show the case when the ANN framework has been used for single-set tracking (NLMPC) of the state variable AcCoA using Vgly as the control. Results show that for SNOPT and KNITRO the control under disturbance does not deviate from the set points, however this is not the case when using CONOPT3 where a distinct loss of tracking is noticed for both small disturbance (i.e. $d = 0.01 - 0.1$) (Figure 5.9a) and larger disturbance (i.e. $d = 0.1 - 1$) (Figure 5.9b). The SNOPT solver performs fastest for cases when disturbance is low (i.e. $d = 0.01 - 0.1$) (Figure 5.9a) and when the full range of disturbance (i.e. $d = 0.01 - 1$) is implemented (Figure 5.9c). In general the SNOPT solver performs the best as it shows the greatest accuracy in adhering to the set points, and bar the effect of large disturbance (i.e. $d = 0.1 - 1$) (Figure 5.9b) it performs the fastest. This further reinforces this technique, ANN-NLMPC, as an effective framework when used with the SNOPT solver. However, when comparing with the results for this solver using the OCFE-NLMPC technique (Figure 5.10), it is shown to have better tracking when the ANN is not utilised. Results for this investigation show that for the OCFE-NLMPC formulation CONOPT3 gives better results. All three solvers show great accuracy when adhering to the set points, except for KNITRO when the full range of disturbance (i.e. $d = 0.01 - 1$) is implemented (Figure 5.10c) where there is slight deviation from the set points at around 50-80 time steps. It can be concluded from these results (Figure 5.9 and Figure 5.10) that for the ANN-NLMPC formulation the SNOPT solver is best and for the OCFE-NLMPC formulation the CONOPT3 solver is best.

When analysing results for the Zone NLMPC formulations (Figures 5.11 and 5.12) it is apparent that KNITRO is unable to handle larger disturbances in the system and tracking within the zone is lost for the OCFE-Zone MPC strategy when disturbance is high (i.e. $d = 0.1 - 1$) (Figure 5.12b) and when the full range of disturbance (i.e. $d = 0.01 - 1$) (Figure 5.12c) is implemented. The ANN-Zone NLMPC formulation (Figure 5.11) however is able to adhere to the set points even when the disturbance is up to 1000% the nominal amount of AcCoA. Tracking within the zone however does fair better for the OCFE-Zone NLMPC formulation (Figure 5.12). With the exception of KNITRO for larger disturbances (i.e. $d = 0.1 - 1$ and $d = 0.01 - 1$), the oscillation of AcCoA remains steadily within the middle of the zone, whereas for the ANN-Zone NLMPC formulation tracking tends to favour the upper limit of the zone. The OCFE-Zone NLMPC strategy may remain in the middle of the zone as it is in the family of implicit RK methods and is inherently a-stable (Hairer and Wanner, 1999) and therefore will optimise to the most stable point within the zone. The ANN however may overshoot this stable point, and ways

to combat this include training the ANN to respond to large disturbances. It is unknown however why KNITRO suddenly loses the ability to track within the set points. Analysis of the disturbance values for the case where $d = 0.1 - 1$ shows that the values where the oscillations deviate from the zone are less than the maximum disturbance throughout the entire 500 time steps, however the computational resources used by the solver increases dramatically. Figure 5.13 shows the computational resources used for KNITRO for the cases where $d = 0.1 - 1$ and $d = 0.01 - 1$ and compares them with the computational resources for SNOPT for the same conditions. As can be seen there are clear spikes in the number of seconds for the iterations when the system fails to adhere to the set points.

The OCFE-Zone NLMPC formulation performs much better for SNOPT and CONOPT3, and the total CPU times are less than for the ANN-Zone NLMPC formulation. As the resulting profile for AcCoA remains within the middle of the zone it performs better than using the ANN formulation. The OCFE-Zone NLMPC formulation shows little variation from the set point for AcCoA and can be deemed the better solution technique. The results show that whilst the ANN-Zone NLMPC strategy is better when disturbance is not present, the OCFE-Zone NLMPC is favoured when disturbance is present. This is due to the nature of OCFE, which discretises over the finite elements, and therefore integrates the solution to the set point more frequently that the ANN-Zone NLMPC strategy. This therefore means that there will be less deviation from the set points as the system is inherently trying to force the system to stay within the bounds. Whilst this can also be achieved with the ANN, by using a smaller step size, it will increase computational effort. This means that if one solely wants to control the system then using the ANN is better as the results are adequate and can quickly determine the performance of the system using zone control, but if one wants to consider disturbance too then the OCFE formulation is favoured as the system generally can handle disturbances better.

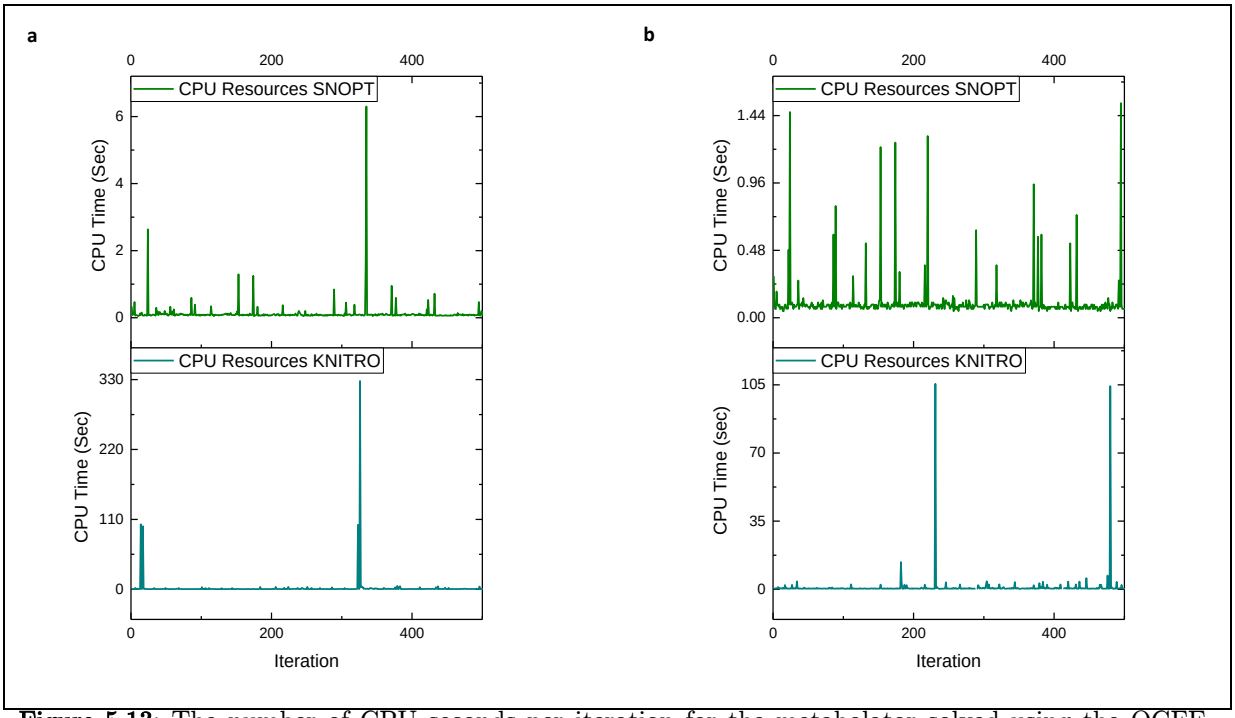

Figure 5.13: The number of CPU seconds per iteration for the metabolator solved using the OCFE-Zone MPC with disturbance formulation with SNOPT and KNITRO solvers in GAMS, **a.** d = 0.1-1.0, **b.** d = 0.01-1.0 for 500 iterations, step size = 0.01. CPU: Intel Core™2 Duo E7400 2.8GHz.

5.2.3.4 Multiple-Input Multiple-Output (MIMO) Control

Due to the nature of the metabolator it is possible to control using both Vgly and acetate concentration (HOAcE). It was noted by Fung *et al.* (2005) that high levels of acetate caused the system to denature and oscillations were lost. This is expected as *E. Coli* does not thrive in acidic environments. However the reverse reaction from the secondary metabolite pool containing AcP to the first pool containing AcCoA can be controlled by setting a profile for the external concentration of acetate. The concept of MIMO control therefore aims to control the system using both of these inputs, with the system trying to attain a steady state. The set profiles for both metabolites ideally would be in synergy with one other, as when AcCoA is oscillating AcP is not, and vice versa. The results for this are presented in Figures 5.14-5.16.

As shown in Figure 5.14 the results for MIMO control show that both the ANN and OCFE formulations are capable of tracking within the set points. There are however some deviations from the set point for the OCFE formulation at around 200 time steps. This is not the case for the ANN formulation. Furthermore, the optimal solution for the OCFE-MIMO Control formulation (Figure 5.13a) shows that HOAcE is not present within the system, which is usually found in the external

media of the cell once it has permeated the cell wall. The absence of HOAcE within the system shows that there is no flux of HOAcE to the external media of the cell, and therefore means that HOAcE is not presented in the system. The cell may be going through internal feedback, which can occure given the dynamics of the Metabolator, and could be the case found by OCFE. The ANN-MIMO Control shows that Vgly is present at specific intervals within the time frame and HOAcE fluctuates in level throughout. Both were optimised using KNITRO however the results were extremely poor, with the average CPU time to solve in the order of 5 hours. Disturbance was added to the system and results for this presented in Figures 5.15 and 5.16.

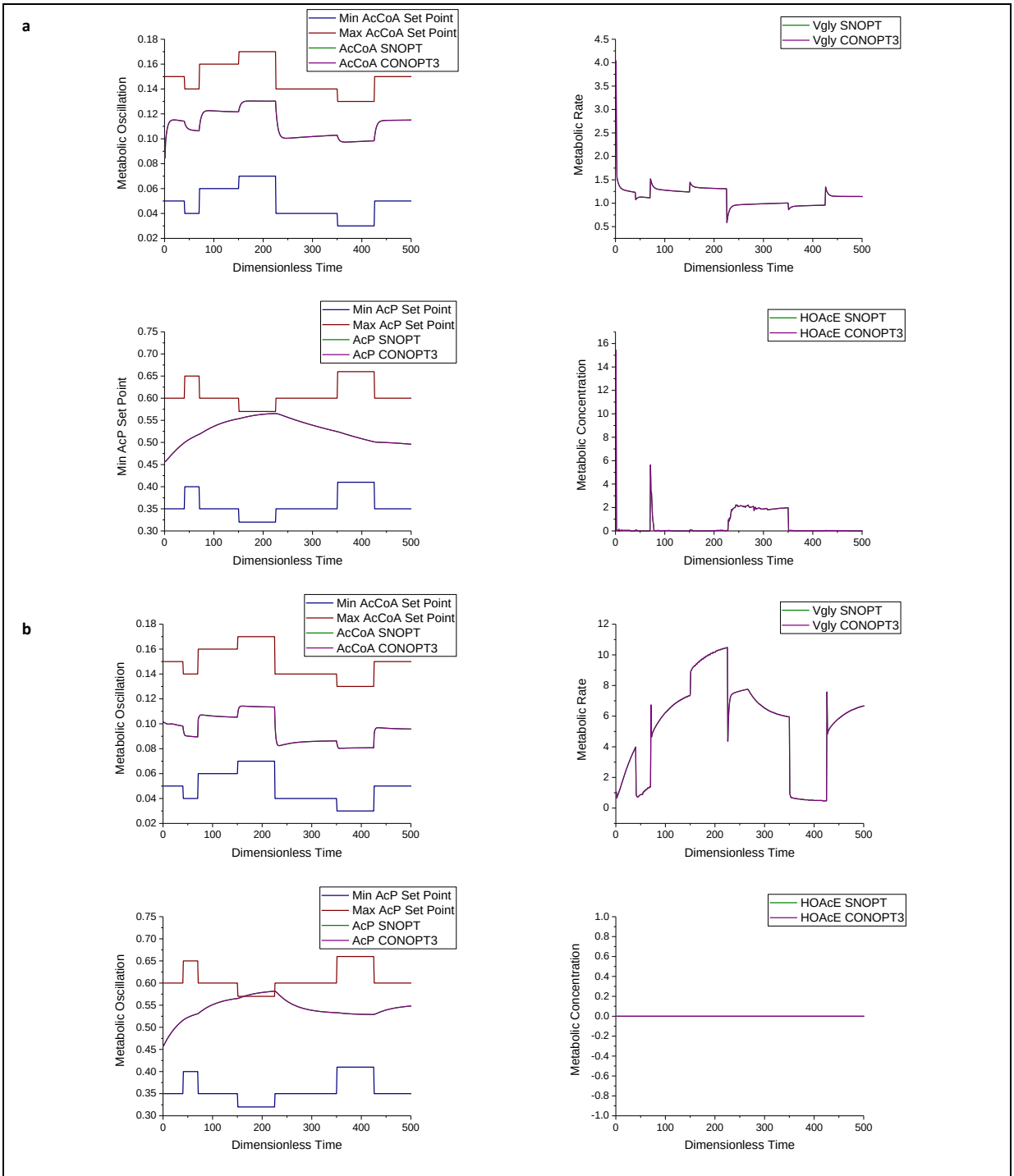

Figure 5.14: a. ANN-MIMO Control results for the metabolator using Vgly and HOAcE as the control variables for SNOPT and CONOPT3 solvers in GAMS, total CPU time (hours) for SNOPT = 00:03:22 and CONOPT3 = 00:02:30, **b.** OCFE-MIMO Control results for the metabolator using Vgly and HOAcE as the control variables for SNOPT and CONOPT3 solvers in GAMS, total CPU time (hours) for SNOPT = 00:02:43 and CONOPT3 = 00:02:10. All simulations run for 500 iterations, step size = 0.01.

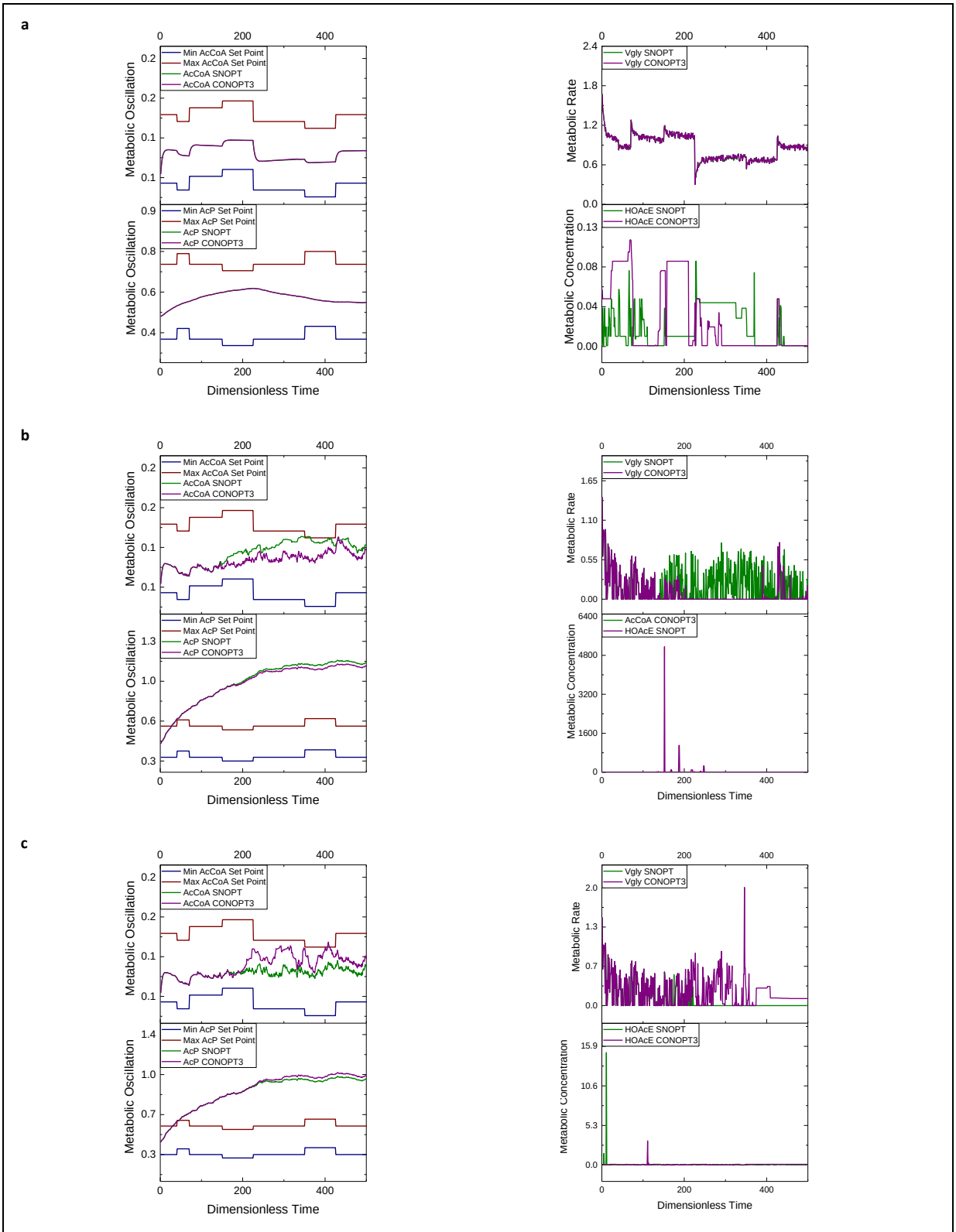

Figure 5.15: ANN-MIMO Control results for the metabolator using Vgly and HOAcE as the control variables for SNOPT and CONOPT3 solvers in GAMS, **a.** d = 0.01-0.1, total CPU time (hours) for SNOPT = 00:03:16 and CONOPT3 = 00:06:17, **b.** d = 0.1-1, total CPU time (hours) for SNOPT = 04:16:34 and CONOPT3 = 00:06:50, **c.** d = 0.01-1, total CPU time (hours) for SNOPT = 00:04:58 and CONOPT3 = 00:06:41. All simulations run for 500 iterations, step size = 0.01.

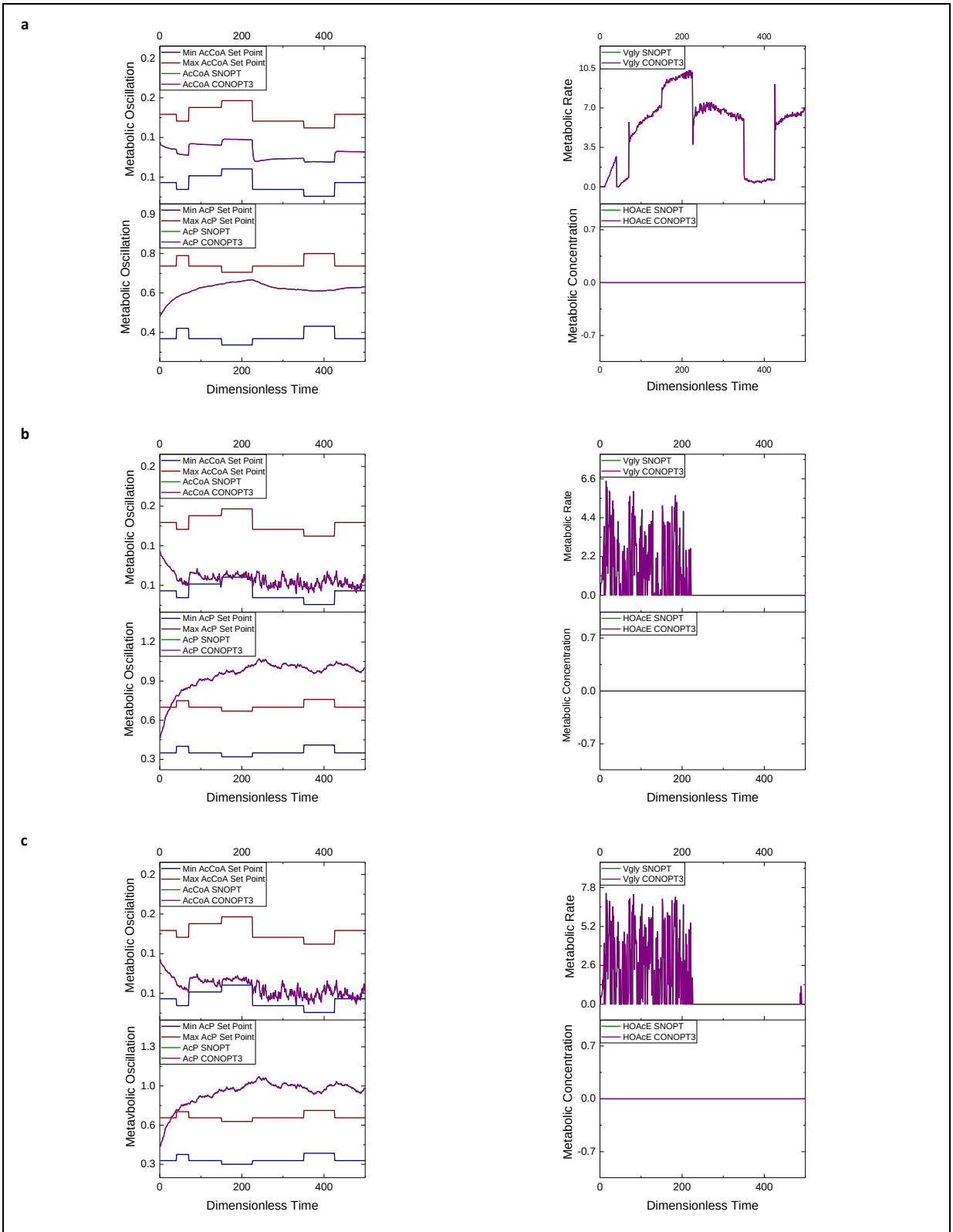

Figure 5.16: OCFE-MIMO Control results for the metabolator using Vgly and HOAcE as the control variables for SNOPT and CONOPT3 solvers in GAMS, **a.** d = 0.01-0.1, total CPU time (hours) for SNOPT = 00:03:19 and CONOPT3 = 00:04:15, **b.** d = 0.1-1, total CPU time (hours) for SNOPT = 00:01:56 and CONOPT3 = 00:02:17, **c.** d = 0.01-1, total CPU time (hours) for SNOPT = 00:02:54 and CONOPT3 = 00:02:29. All simulations run for 500 iterations, step size = 0.01.

Both Figures 5.15 and 5.16 show that at higher levels of disturbance tracking is lost for AcP in both formulations. The system dynamics are compromised and violation of the set zone is seen throughout. This could be due to the fact that the disturbance added to these profiles (Figure 5.15b and 5.15c and Figure 5.16b and 5.16c) can reach up to 1000% of the nominal values of AcP and AcCoA. However both formulations are capable of optimising with low levels of disturbance. In terms of which formulation performs better, the ANN does as the results for AcCoA are more favourable than the OCFE results. Both SNOPT and CONOPT3 perform similarly, with SNOPT, bar a few exceptions, performing faster for both formulations.

5.3 Summary of Results

Results in this chapter show that the OCFE formulation outperforms the ANN when the problems are of a small scale, however the ANN formulation is better suited towards highly nonlinear problems such as the metabolator. Problems that are less complex indicate that OCFE can solve the models quickly and provide highly accurate results, but can deviate from set point tracking during control. Traditionally when using neural networks, fine tuning of the weight and matrices of the network must be performed offline through parameter estimation, however in the formulation used here these are computed as part of the NLMPC formulation. OCFE can also eliminate the need for offline fine tuning as the collocation points are well defined. There is possibility and scope to use the 5 collocation points method, but this will increase computational effort, and is unlikely to give better results for the two small case studies in Sections 5.2.1 and 5.2.2 than the 3 collocation points method used. For the metabolator case study (Section 5.2.3) the ANN-NLMPC formulation performs faster in terms of set point tracking, and therefore is better suited to more complex dynamic systems. However in the case when disturbance is added (Section 5.2.3.2) the OCFE formulation is favoured as it handles disturbances better, with exception of the KNITRO solver, and can track safely within the set zone (Figure 5.11). When the system is modelled using MIMO Control the ANN formulation performs better in general. A summary of the computational effort and times for the isothermal CSTR and the distillation column examples are presented in table 5.2, and a summary of the computational times for the metabolator example are shown in table 5.3.

As mentioned previously the results indicate that there is a difference in using ANNs and OCFE as solution methods for solving ODEs. It has been shown that OCFE outperforms the ANN when the problem contains fewer ODEs and are simpler in nature, however fails to solve highly nonlinear models in short time

frames. The ANN on the other hand provides faster solutions for highly nonlinear problems. When replicating results of the metabolator (Fung *et al.*, 2005) the ANN-RK4 model was capable of reproducing the results found in their MATLAB model. The OCFE formulation also replicated the results but the difference in the solution times outweighs the benefits of using OCFE for this type of problem. The more complex the model is the more computational effort is required for the OCFE, which is typical as it needs to discretise through more points in the domain space. The trade-off in this case is the level of accuracy from the OCFE compared to the ANN. The ANN formulation provides accurate results for more nonlinear problems, especially when using MIMO control, but the OCFE formulation performs better when Zone NLMPC is utilised.

Table 5.2: Summary of the total CPU times (minutes) for the isothermal CSTR and distillation column case studies and their solution methods. CPU: Intel Core™2 Duo E7400 2.8GHz.

Case Study	Solver	Solution Method (CPU Time – Minutes)			
		ANN/Simulation	OCFE/Simulation	ANN-NLMPC	OCFE-NLMPC
Isothermal CSTR	SNOPT			00:49	00:20
Low order distillation column	SNOPT	10:59	06:37	10:20	08:13
	CONOPT3	19:25	07:58	09:32	10:36
	KNITRO	26:08	10:22	18:42	18:20

Table 5.3: Summary of the total CPU times (minutes) for the metabolator case study and its solution methods. CPU: Intel Core™2 Duo E7400 2.8GHz. d = disturbance added to the system.

Case Study	Solver	Solution Method (CPU Time – Minutes)			
		ANN-NLMPC	OCFE-NLMPC	ANN-Zone NLMPC	OCFE-Zone NLMPC
Metabolator	SNOPT	01:41	04:46	01:43	01:49
	CONOPT3	01:58	01:49	01:23	02:50
	KNITRO	04:34	06:03	03:29	05:37
Metabolator/Disturbance					
d = 0.01 – 0.1	SNOPT	02:18	07:34	01:59	03:56
	CONOPT3	03:20	01:41	05:20	02:26
	KNITRO	05:40	05:06	08:12	06:29
d = 0.1 – 1.0	SNOPT	18:56	04:05	02:44	02:39
	CONOPT3	02:56	01:49	03:21	02:59
	KNITRO	04:21	06:21	08:30	16:53
d = 0.01 – 1.0	SNOPT	03:29	04:25	02:25	02:22
	CONOPT3	05:28	02:53	04:42	02:53
	KNITRO	07:58	06:14	08:47	10:02
Metabolator/MIMO	SNOPT			03:22	02:43
	CONOPT3			02:30	02:10
Metabolator/MIMO/Disturbance					
d = 0.01 – 0.1	SNOPT			03:16	03:19
	CONOPT3			06:17	04:15
d = 0.1 – 1	SNOPT			256:34	01:56
	CONOPT3			06:50	02:17
d = 0.01 – 1	SNOPT			04:58	02:54
	CONOPT3			06:41	02:29

5.4 Conclusion of Techniques

A comparison between solving ordinary differential equations using artificial neural networks (ANNs) and orthogonal collocation on finite elements (OCFE) has been presented in this Chapter. Result findings suggest that whilst both are able to deal with systems of ordinary differential equations, orthogonal collocation on finite elements provides better results for problems where disturbances are present in the system, whereas artificial neural networks are better suited towards highly dynamic problems without disturbance. In the case of multiple-input multiple-output control however, the neural network framework outperforms the collocation method. User preference will dictate which of the two solution techniques is best used. The trade-off is the level of accuracy required, where the neural network framework does provide accurate results for less complex problems, but the collocation method formulation has a greater level of accuracy for highly nonlinear problems. In terms of computational effort both methods are comparable to one another, with the neural network formulations in general performing faster than their collocation method counterparts. From results found when using each of the solvers it can be said that the results obtained from SNOPT can be regarded as superior, and therefore justifies the use of this solver in previous work (Chapter 4). Due to the fact that the neural network outperforms the collocation method when utilising multiple-input multiple-output control, it can be deemed the better choice for modelling the nonlinear systems. This further reinforces the development and use of this framework throughout this thesis.

6. Systems Architecture

As mentioned previously there is scope to use the developed framework for systems architecture problems, which are problems where a standard list of well characterised biological parts are presented and the framework selects parts to form biological circuits. The developed framework has already shown to be effective in simulating and controlling nonlinear systems, and so work progressed onto one of the many avenues that this could be applied to. This chapter showcases how the developed framework can be applied to these problems by highlighting how the framework can optimally find suitable biological circuits. There has been much advancement in synthetic biology and its use to help find optimal system architectures that can provide useful products. With the standardisation of biological parts (MIT, 2005) this is now becoming a reality as well characterised parts can effectively be put in sequence to either enhance or create new products. This type of research looks to provide a pseudo plug-and-play approach to genetic systems, as modelling is required to show the optimal configurations and experiments are crucial to test the feasibility of these architectures. However, automated biological circuit design tools have not yet been realised despite the advances in part availability. Recent advances include adaptations of electrical engineering concepts such as Boolean optimisation and Carnaugh maps to provide biological circuits with digital functions (Marchisio and Stelling, 2011). There have also been approaches that translate user-defined specifications into genetic circuits, which adhere to digital logic (Pedersen and Phillips, 2009; Densmore *et al.*, 2010; Beal *et al.*, 2011) as software for future consideration. The more traditional or 'analogue' method of synthetic circuit design typically employs heuristic methods such as evolutionary algorithms (Francois and Hakim, 2004; Wu *et al.*, 2011) and simulated annealing (Rodrigo *et al.*, 2007).

More relevant approaches looked at exploring functional space from a given library (Rodrigo *et al.*, 2011) and global sensitivity analysis to determine which mutation sites are required to achieve functionality (Feng *et al.*, 2004). Mixed integer non-linear programming (MINLP) is widely used for the application of selecting appropriate circuits for genes and biological systems. Binary variables can be used to model sequences, candidates, existence and non-existence of units, whereas variables can be used for example in biological systems to indicate different cells. Finally continuous variables are used to model the input-output and interaction relationships among the system units as well as any interconnected

system to these. This chapter will detail two methods, one deterministic and one automated, that showcase how MINLP can be used to obtain biological circuits with desired function, as well as how the developed meshless ANN framework detailed in Chapter 3, Section 3.3, can be applied to such problems.

6.1 Formulations of MINLP Problems

There are many forms of MINLP problems ranging from generalised benders decomposition models (Sahinidis and Grossman, 1991) to outer approximation models (Duran and Grossman, 1986). Each type of formulation has its own advantages in modelling MINLP problems and will be discussed in this section.

6.1.1 Generalised Benders Decompositions

Generalised benders decompositions (GBD) are a class of MINLP solution algorithms that impose the upper and lower bound of the MINLP problem at each iteration, which are updated to allow the problem to converge within a finite number of iterations. Generally, the upper bound is found from the primal problem, with fixed y variables and the lower bound is found from the master problem using duality theory. As the problem iterates the upper bounds no longer increase and the lower bounds no longer decrease therefore allowing the problem to converge within a finite number of iterations. Use of GBD has been seen in research by Sharif *et al.* (2001). Here an efficient method for minimising energy loss over time was investigated. The proposed method had four main advantages, the total energy loss within the system is minimised, the number of physical plant changes is kept low, the number of controls variables are reduced and the likelihood of an infeasible solution is minimised when compared to older models using power loss minimisation (Elkady *et al.*, 1986). The problem is transformed in two parts, the power loss minimisation (PLM) model and the energy loss minimisation (ELM) model. The PLM model is used at the beginning of the VAr (volt-ampere reactive) dispatches from interval two onwards. The ELM model is executed at the beginning of each interval, and is solved using GBD. It was found that the ELM gives a smoother voltage profile than the PLM method. It was also noted that the energy loss was reduced over time whilst ensuring that the important control variables are not altered during the process.

The concept of optimal power flow (OPF) has also been studied by Khodr and Martínez-Crespo (2009). Here a new and efficient method is investigated using optimal power flow and GBD. OPF has been used extensively in power systems in the generation or transmission level to denote the difficulty in finding an optimal value for the control variable when minimising the total operation cost, whilst still adhering to the technical constraints of the network and equipment. Recent developments of these types of systems have led to the introduction of uncertainty within the models, which may contribute to voltage control and optimisation. The concept of distribution optimal power flow (DOPF) is introduced as a means to integrate all of the decision problems in a unified model, which can reconfigure the model to find the optimal operating point for the distribution network. The decision problem (master problem) is formulated as an MINLP model that uses a slave problem in the form of an NLP model. The NLP model is used to define the feasibility of the master problem through use of OPF, and gives information to formulate the linear Benders cuts that transfer information from the slave to the master problem. This feasibility seeking aspect of the Benders algorithm is shown to be efficient at reconfiguring the OPF of large scale distribution models with little computation time. This is known as the reconfiguration integrated with optimal power flow (ROPF) model, and has proven to give better results than previous literature (Khodr and Martínez-Crespo, 2009).

Power optimisation is one of the main areas where GBD can be utilised to solve complex problems. Issues such as congestion in network systems have been looked at by Shrestha and Fonseka (2006). Here a framework was created to look at alleviation of congestion in networks using network expansion and flexible AC transmission systems (FACTS). It was noted that simple addition of new line capacities was not necessarily the best way to deal with time-dependant congestion in various lines of the network as there is a lack of control over grid flows. Therefore the idea of using FACTS posed as a solution to this issue as this is able to give adjustable line capacities. The work proposed here was deemed a continuation of previous work (Shrestha and Fonseka, 2004), which looked at long-term network planning solutions. The long-term problem can be coupled into the master problem (investment problem) and the operational sub-problem of power dispatch. This partitioning process allows relaxation of the nonlinearities within the flow equations of the model. This may not always guarantee a precise optimal solution, but a near optimal solution may be obtained within the user specified bounds. The investment problems ran on a yearly time scale and investment decisions are made based on the operational sub-problem, which is estimated using an annual load duration curve (LDC). The operational sub-problem provided the expected values of annual

congestion costs and shadow prices corresponding to the line capacity, the phase shifter capacity and the series compensation limits to be used in the Bender's cuts of the master problem. This allowed the inclusion of time-varying bidding details of supply and demand into the expansion planning model. Practice models showed robust results and the network improvement costs were justified against the long-term congestion cost savings. The method can be developed further to provide network reinforcement schemes, which are time-ordered sets of investment decision that can be used for the network planner. It was also seen that although the model was capable to solving the practice model (IEEE bus 24 model), the implementation of the model to a practical system would require more realistic data.

6.1.2 Outer Approximation

Outer approximations (OA) methods are used in modelling to help solve mixed integer problems. This approach involves solving approximations to a mathematical problem, where the approximation contains the original feasible region. This approach is capable of solving complex problems with relatively small amounts of sub-problems. Research by Viswanathan and Grossmann (1990) looked into a combined penalty function and OA algorithm capable of optimising an MINLP problem. The model contained an equality relaxation (OA/ER) (Duran and Grossmann, 1986), which featured an exact penalty function that allowed violations of the linearisation of non-convex constraints. The OA/ER algorithm relies on the convexity of the functions within the model, as well as the quasi-convexity of nonlinear equality constraints (Kocis and Grossmann, 1987). When these conditions are met the algorithm is able to find the optimal solution. However, if these are not met then the NLP sub-problems may only solve for the local optimum, and the linearisation of the master problem can cut into the feasible region, leading to a sub-optimal solution (Kocis and Grossmann, 1988). In order to overcome this issue, Kocis and Grossmann (1988), developed a two-phase strategy. The first phase involved the OA/ER algorithm, and the second phase identified the linearisations of non-convex functions by using local and global testing, thereby relaxing the master problem. This scheme allowed for obtaining the global optimum in 80% of a set of test problems.

Viswanathan and Grossmann (1990) expanded this algorithm to incorporate an augmented penalty function, which detects violations of the linearisation of the nonlinear functions. This AP/OA/ER function does not require an initial set of 0-1 variables as the algorithm begins with a solution to a relaxed NLP problem.

Whereas the OA/ER function has been proven to be successful (Kocis and Grossmann, 1980), it relies on the assumptions made of the convexities within the algorithm. Whereas the AP/OA/ER function does not rely on such assumptions, as the new master MILP problem uses linear approximations instead. The MILP problem however does not produce valid lower bounds for the objective function, and therefore a termination criterion was added that allowed the function to be stopped based on the progress of the NLP sub-problems. Although the algorithm has provisions in place to optimise the function, it cannot guarantee a global solution. This is due to the fact that if the NLP sub-problems have multiple local solutions, the function will converge to a sub-optimal solution. Also if the NLP sub-problems for fixed binary values have different local optima, the algorithm may be trapped into a local solution. Despite this, computational efforts with the algorithm are promising. The proposed AP/OA/ER function was implemented using the DICOPT++ program in GAMS. It was tested on a set of 20 MINLP problems, which involved up to 60 0-1 variables, 709 continuous variables and 719 constraints. It was noted that 11 of these 20 MINLP problems were non-convex, and the results obtained for these showed great robustness as they correlated with the global optimum.

The OA algorithm is capable of finding the global optimum of a problem, provided there are convexities within the function. However, when a problem contains non-convex functions it needs to be transformed to allow the OA/ER solver to find an optimal solution. Constraint violations must be allowed within the MILP master problem through the introduction of slack variables to the linearised constraints. The termination criterion must also be modified, as through the introduction of the slack variables we have lost the properties of the outer approximate. The DICOPT++ program allows the function to be solved using OA/ER if the problem is transformed from a non-convex to a convex function accordingly. This can be highlighted with research from Bergamini *et al.* (2008). Here they looked at how OA could be used to find global optimum solutions to MINLP problems involving bilinear and concave terms. The model, extended from previous research (Bergamini *et al.*, 2005), can solve for both MINLP and generalised disjunctive programming (GDP) problems. The model follows the same general principle in that the first stage involves initialisation of the upper bound, followed by optimisation of the outer values. The bounds of the variables involved in the non-convex terms are then contracted. If at least one if these is infeasible, the algorithm must find a new outer. If however there are feasible the global optimal solution can be calculated. This was tested using GAMS and the solvers CONOPT3, for the NLP and LP problems, and CPLEX for the MINLP problems. The algorithm was successful in finding solutions to complex bounded MINLP problems. However it

has been noted that the MILP sub-problems have increased in size due to the nature of the algorithm. This increases computing effort, even though each successive MILP problem only differs by addition of some grid points. It has been proposed that a possible solution to this could be to simultaneously solve the MILP problems and update the grids within the same iteration.

Other methods of optimising processes can include the use of R-graphs (Farkas *et al.*, 2008). Here research was conducted using R-graphs in constructing a new distillation column, which would later be optimised using OA and MINLP. R-graphs are graphical simulations of models that utilise R-programming languages. They are particularly useful for linear and nonlinear modelling, time series analysis and statistical testing. The objective of the research was to design the optimal configuration for a distillation column, which required the least amount of annualised cost. It was noted that the computational difficulty in modelling the MILP/MINLP problem is greatest with increasing numbers of binary variables. Therefore it was noted that in order to reduce these variables the model could either be modified or the appropriate superstructure could be calculated. Difficulties arose when the researchers tried to apply the model to known examples. It was noticed that with the most complex of distillation systems the algorithm could not use traditional programs to solve i.e. GAMS, SBB or DICOPT MINLP solvers. This is due to the system containing large amounts of nonlinear equations and this led to the systems being deemed infeasible. There is however a feasible solution, so the models were altered so that the initial values for the NLP sub-problems were substituted in an effort to reduce the number of equations. A new step was inserted between the MILP master problem and the NLP sub-problems. Each appropriate value for the variables is calculated between each stage of the model using the binary vector obtained from the master problem. This reduces solution time greatly as the NLP problems have a good initial value. When comparing the model to previous research by Yeomans and Grossmann (2000), it is seen that the computation time has decreased greatly and the local optima is more indicative of the true optimal solution. Once the difficulties with regards to computation were addressed the final model was able to provide good solutions for processes containing as many as 160 stages in the distillation process.

6.1.3 Branch Reduce and Bound

Initial problems containing non-convexities arising from discreteness used the branch-reduce and bound (BRB) method, which has since been extended to allow for

global optimisation (Cheon *et al.*, 2006). The scheme acts by recursively branching (partitioning) the feasible region in search for a global optimal solution. For any given partition element, bounds on the optimal solution are used to determine whether to examine the partition or to discard it from further consideration (bounding). This method is particularly useful in modelling discrete and combinatorial optimisation problems. The use of BRB is heavily focussed in work by Balasubramanian and Grossmann (2002), which looked at the problem of scheduling a flowshop plant with uncertain processing times. Flowshop plants involve plants where the jobs associated with the manufacturing of products use the same units and machines in the same order. A possible solution to this can involve specifying the order of the process, and essentially there are many different possible solutions. The uncertainty of the processing times was modelled using discrete probability distributions. However this can lead to combinatorial explosion of the state space, and when this is coupled with the sequencing can generate a difficult model to optimise. The BRB method poses as a solution to this issue as it can possibly select the sequence with the minimum makespan. As the models can be difficult to optimise the proposed solution was to couple the BRB method with a disaggregation step, henceforth known as BBD. This potentially gave the ability to reduce the uncertainties within the model and therefore allow for a more robust solution. However the BBD model required refining in order to be applicable to practical problems. A proposed MILP model was therefore created and tested on a UIS flowshop plant. This model derived from the analytical expression for the expected makespan of the stochastic zero-wait flowshop plant. It gave near optimal solutions and in a reasonable computation time.

The BRB algorithm has also been used to optimise the pharmaceutical process and this can be seen in work carried out by Siddhaye *et al.* (2004). Here they researched the use of BRB coupled with MILP to create a two-step model for designing new molecules. The first step uses topological indices to develop structure activity relationships (SAR) for the properties of interest. The second step uses MILP programming to solve the model and optimise the molecule design. In order to develop SAR structures a comprehensive knowledge is required for the molecules topological properties. Research by Kier and Hall (1976) described these indices and how they can be applied to the field of medicinal chemistry. In order to obtain an accurate SAR model the zeroth and first order connectivity indices are used to develop linear correlations for three physical properties of interest, the octanol-water partition coefficient, the melting point and the water solubility. The MILP included a linear objective function, the property correlations and linear structural constraints, which ensure that the optimum solution is a stable and connected molecule. The result found an optimum solution containing an aromatic ring and a

carboxyl group. The correlation is acceptable and the molecule had estimated properties closest to the desired physical properties of interest.

The concept of BRB and MILP modelling has also been of particular use in economics as highlighted by research from Amaro and Barbosa-Póvoa (2009). This work focussed on the supply chain issues that arise from decisions made in response to demand in the pharmaceutical industry. The main aims of the research were to reduce the global operating costs, whilst maximising the profit realised from the supply. An MILP model was then created, which focussed on the planning constraints of the system with a means to optimise the whole supply chain process. The variables were organised into three groups, the variable bounds, the modelling events and the material balances. In order to visualise the effect of uncertainty three cases were considered, a single product family and market position (soft-tablet medicines), the effect of combined market positions for the same uncertain final product demand and the effect of different final product uncertainties at the same market level. The planning was performed under market demand or price uncertainties, whilst accounting for different partnership structures. The approach considered the simultaneous integration of operational, economical and market aspects. Although the results of the model are promising, further work is needed in extending the demand to price variable to larger planning horizons with different periods of occurrence. This will give a more comprehensive overview of the cumulative effect of demand uncertainties induced by price change.

6.2 Deterministic Optimisation

It has been widely regarded that building circuits to meet multiple inducer requirements is challenging (Hasty *et* al, 2002; Sprinzak and Elowitz, 2005; Endy, 2005). The fact that many of the interactions between parts have not been fully described, as well as the number of ways one can choose to interconnect the components, leads to increased complexity. There have been several small-scale circuits constructed to meet specific functionality such as the genetic toggle switch (Gardener *et* al, 2000), the repressilator (Elowitz and Liebler, 2000), the metabolator (Fung *et* al, 2005) and programmable cells (Kobayashi *et* al, 2004). The potential for using modelling and computational tools to better understand the function of biological circuits has been recognised and several models have been proposed to describe the interactions between genetic elements (Hasty *et* al, 2001; Gilman and Arkin, 2002; Glass *et* al, 2005). There have however been several studies that describe how a circuit can fail to exhibit desired functionality due to improper

assembly of its basic elements. Work conducted by Tuttle *et* al (2005) confirmed that repressilator circuits constructed with wild-type promoters do not result in oscillations. Furthermore, studies conducted by Hooshangi *et* al (2005) revealed that the behaviour of transcriptional cascade depends on the promoter leakiness and expression levels at the previous stage. To alleviate the difficulties found in building circuits that can meet inducer requirements, a deterministic optimisation framework by Dasika and Maranas (2008) was proposed. This framework, called OptCircuit, utilises ordinary differential equations, but is general enough to also accommodate stochastic simulations, and has two major advantages:

I. It can automatically identify the circuit components from a list and the connections required to bring forth the desired functionality
II. It can rectify or redesign an existing (non-functional) biological circuit and restore its functionality by modifying an existing component (e.g. through modification of the kinetic parameters) and/or identify additional components to append the circuit

Literature sources were used to create a comprehensive list of promoter-protein, protein-protein and protein-inducer pairs and the desired circuit response is given as a maximisation/minimisation of an objective function. The process is also iterative, and thus can yield an ensemble of circuits that all display the same functionality, thereby allowing the user to choose whichever they see fit. A comparison can be made from electrical circuits to biological circuits, and synthetic biology utilises both of these fields to generate new synthetic circuits. Two choice examples of this integration are presented in Figure 6.1, which details the natural (biological) and electrical circuits whose principals were combined to form the genetic toggle switch (Gardner *et* al., 2000) and the Repressilator (Elowitz and Leibler, 2000). The toggle switch takes inspiration from set-reset circuits (SR circuits) (Shirakawa *et al.*, 1995) and the Repressilator takes inspiration from LC (tuned) circuits (Shieh *et al.*, 1997). OptCircuit utilises a different framework as it selects parts from a standardised list of compatible biological parts and forms new circuitry from this. The overall concept of OptCircuit can be seen in Figure 6.2, where R and S represent reset and set respectively, cro and cI are genes that code for the proteins P_I and P_{II}, cI-ts is a bi-stable repressor, P_{tet2} is a gene coding for the *tet* protein, *lacI* is a protein that is induced by IPTG within the circuit, *GFP* is a gene coding for green fluorescent protein. For the oscillator, C and L represent the capacitor and inductor respectively, *kai(x)* represent the genes that are found in cyanobacteria that are needed for circadian rhythm, Kai(x) are the proteins form from the genes, $P_t$tetOI

and $P_t$lacOI are promotors within the circuit coding for *tet* and *lacI* proteins and finally cI is a gene coding for the inducer of the circuit *cI*.

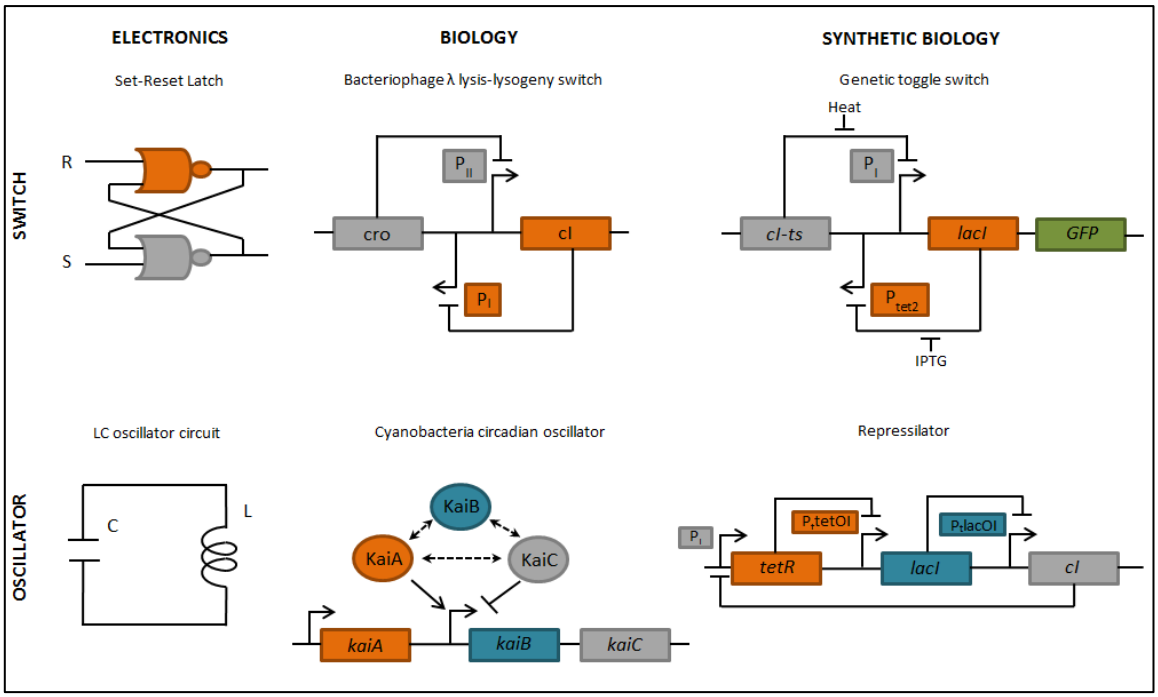

Figure 6.1: An illustration of how synthetic biology combines electrical circuits with natural circuits, showing the genetic toggle switch (Gardner *et al.*, 2000) and the Repressilator (Elowitz and Leibler, 2000). Each gene, inducer, protein and promotor is described in the main text of Section 6.2. Lines ending with a bar represent induction of the circuit, arrows represent protein formation and dashed arrows represent protein-protein interaction.

Three examples were used to show the various architectures that can be obtained using the OptCircuit framework, the first of which is an investigation into designing a circuit that can discriminate between inducer molecules, which is described in Section 6.1.1.

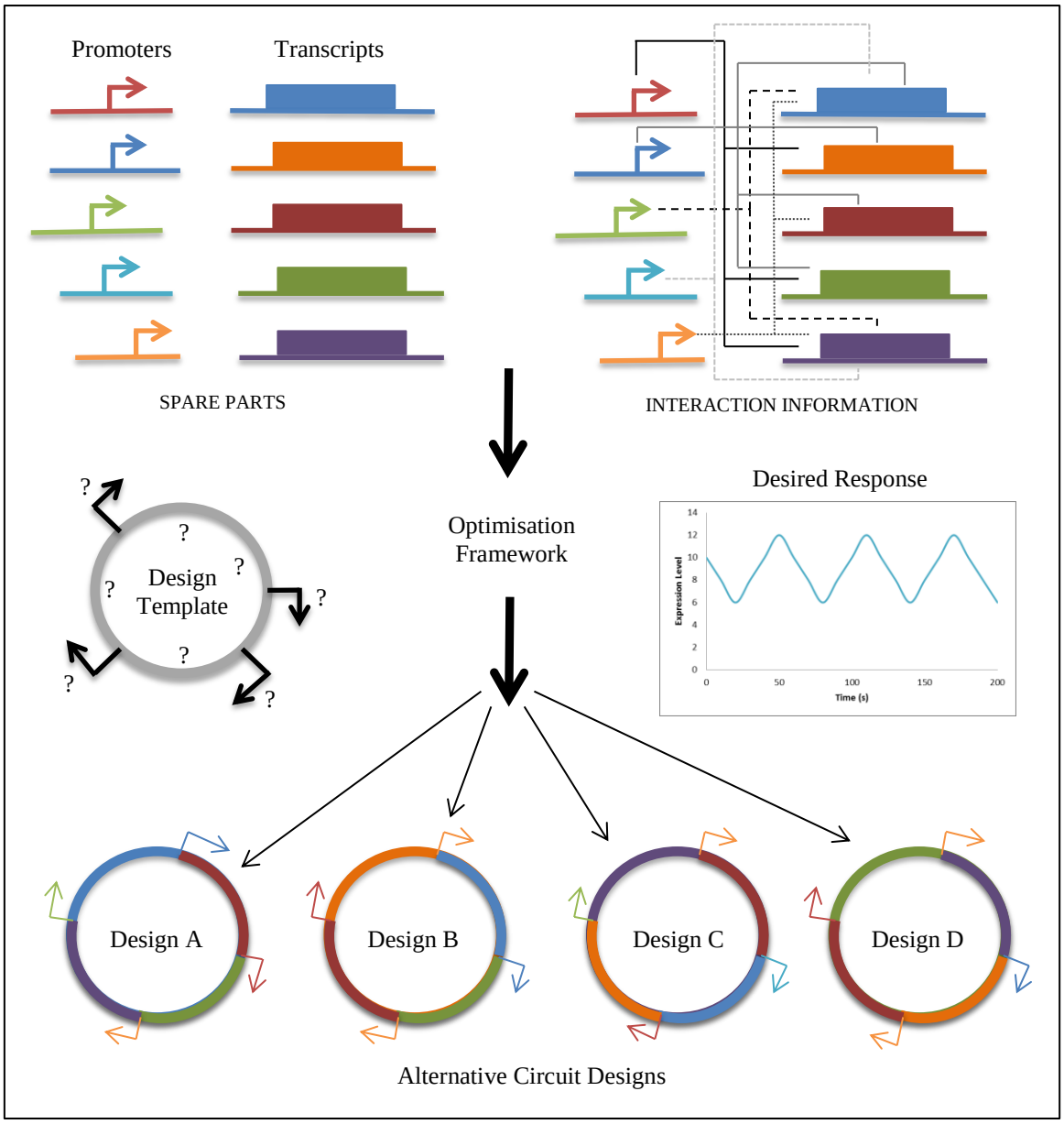

Figure 6.2: An illustration of the OptCircuit framework. Three main components of the framework are the basic genetic elements (i.e. the promoters, transcripts and inducers), the underlying mechanisms that drive the circuit and finally the desired behaviour of the circuit. Integration of these components is achieved using an optimisation based framework embedded into OptCircuit.

6.2.1 Design of Circuits with Inducer-Specific Responses

Within this example generating circuit designs whose responses are contingent on the presence/absence of different inducer molecules tested the OptCircuit framework. In order to determine if the architectures were feasible, the results were compared to well-known designs (Gardner *et al.*, 2000). The pre-requisites for the circuit were that in the presence of anhydrotetracycline (aTC) only *lacI* should be

expressed, and in the presence of IPTG the circuit must only express *tetR*. This desired circuit response is imposed through Equation 6.1, by maximising the scaled difference between the expression of the desired and undesired fluorescent protein in response to the two different inducers.

$$Maximise\ z = \left(\left(\frac{p_{lacI}^{aTC} - p_{lacI}^{IPTG}}{p_{lacI}^{aTC}}\right) + \left(\frac{p_{tetR}^{IPTG} - p_{tetR}^{aTC}}{p_{tetR}^{IPTG}}\right)\right)/2 \qquad (6.1)$$

Where: p_{lacI}^{aTC} and p_{tetR}^{aTC} represent the levels of the transcripts *lacI* and *tetR* in the presence of inducer *aTC*. Similarly, p_{lacI}^{IPTG} and p_{tetR}^{IPTG} represent the levels of *lacI* and *tetR* in the presence of inducer *IPTG* respectively. Different architectures were identified using OptCircuit (Figure 6.3) with up to two promoter-transcript pairs, with the best circuit shown in Figure 6.3a. The configuration is reminiscent of the genetic toggle switch (Gardener *et al.*, 2000) and is in line with its dynamics, as in the presence of aTC, the activity of the protein tetR is supressed (Figure 6.3b), which leads to the expression of lacI from P_{tet2} promoter (as tetR supresses expression from P_{tet2}). In the presence of IPTG (Figure 6.3c) the activity of protein lacI is supressed, which enables the expression of tetR from the P_{lac1} promoter. Once the search was performed for circuits with only two promoter-transcript pairs was successful, the framework was utilised for more complex architectures containing three or four promoter-transcript pairs. The resulting circuits (Figure 6.4) show that OptCircuit is capable of giving both simple and intuitive designs as well as non-intuitive designs with added complexity, thereby increasing the opportunity for kinetic parameter tuning.

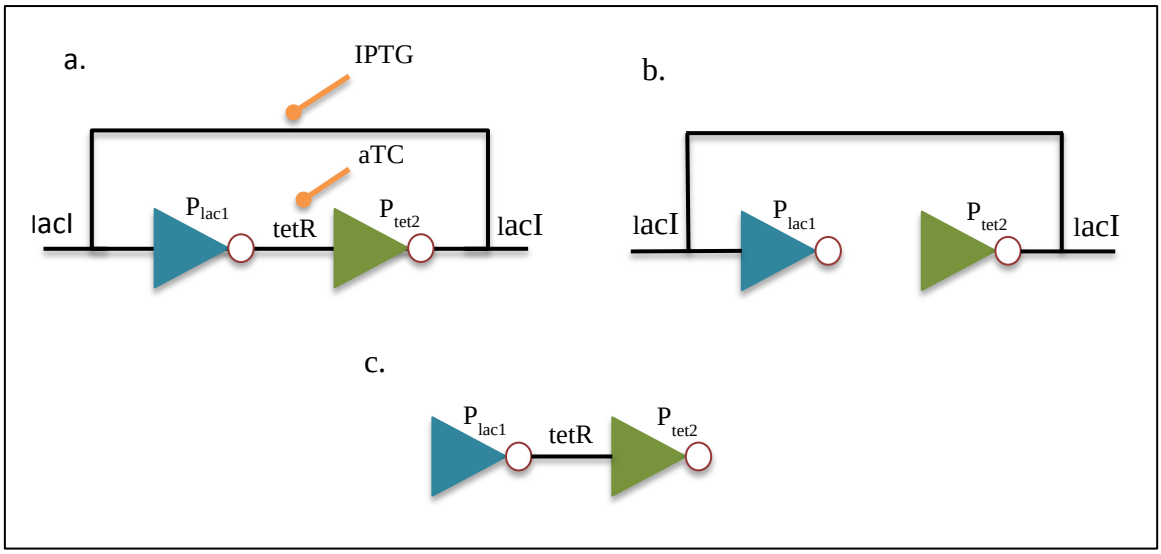

Figure 6.3: a. The simple circuit identified by OptCircuit which is reminiscent of the genetic toggle switch, **b.** activity of the circuit in the presence of aTC, which supresses the activity of tetR so lacI is expressed, **c.** activity of the circuit in the presence of IPTG, which supresses lacI so tetR is expressed. The triangles with open circles at the vertices represent the promoter elements. The triangles with open circles at the vertices represent the promoter elements and arrows with a circular end represent inducers.

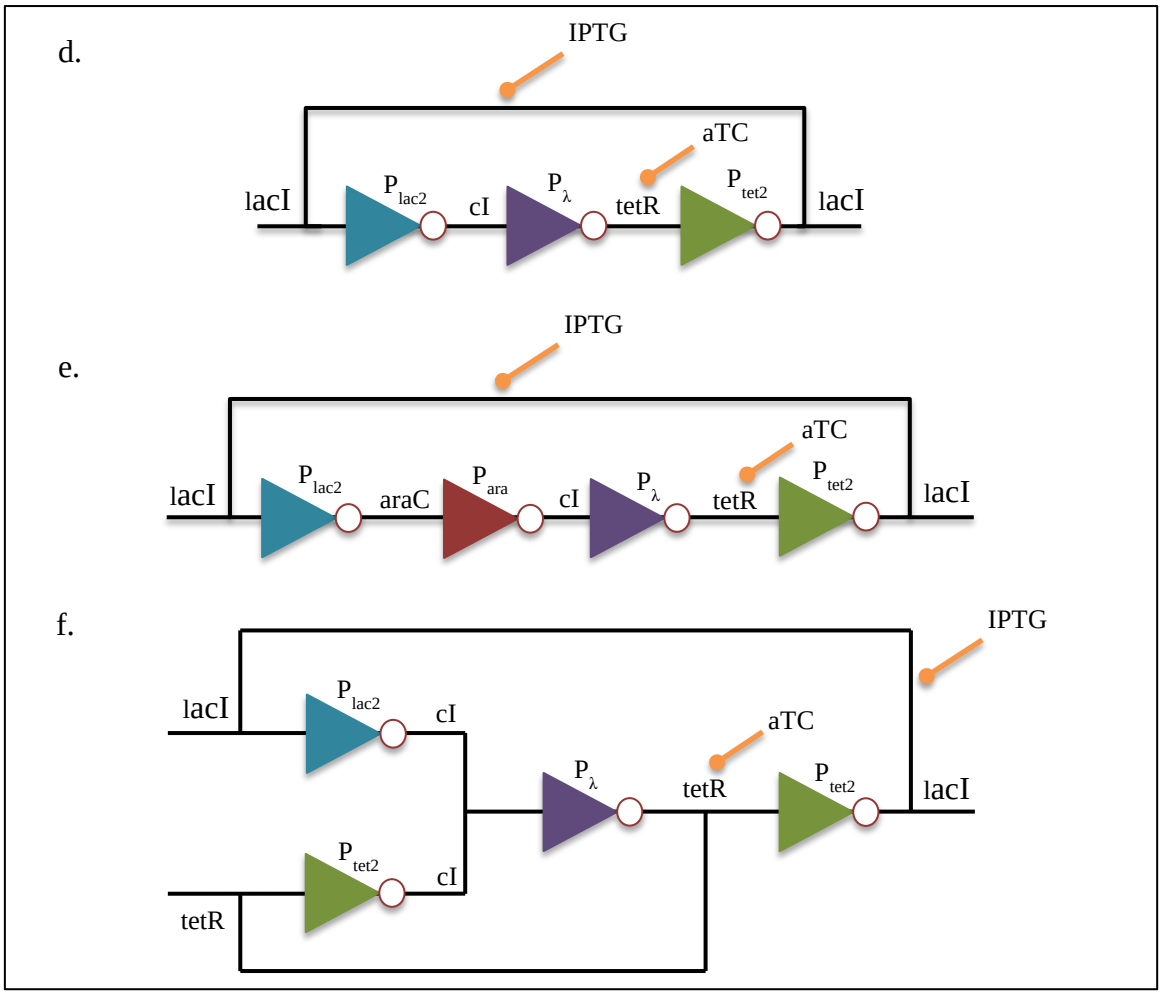

Figure 6.4: Alternative circuits proposed by OptCircuit indicating that OptCircuit is able to identify more complex architecture to realise a specified outcome. The triangles with open circles at the vertices represent the promoter elements.

The ODEs used to model this system can be seen in Equations 6.2-6.7. The equations provide a mechanistic description that governs the time evolution of the protein levels in the system. The binary variables, Y_{ij}, determine if a transcript (protein) is expressed from a promoter.

$$\frac{d[lacI]}{dt} = \sum_{i=1,\dots,4} Y_{P_{lac(i)},lacI} \frac{a_{lac}}{1 + K^{lacr(i)}.[lacI]^4} + Y_{P_\lambda lacI} \frac{a_\lambda}{1 + K^\lambda.[cI]^2} + \sum_{i=1,2} Y_{P_{tet(i)},tet} \frac{a_{tet}}{1 + K^{tet(i)}.[tetR]^2} + Y_{P_{ara}lacI} \frac{a_{ara}}{1 + K^{araC}.[arac]^2} - K^f.[lacI].[IPTG] + K^b[lacI - IPTG] - K^{decay}.[lacI] \tag{6.2}$$

$$\frac{d[lacI - IPTG]}{dt} = K^f.[lacI].[IPTG] - K^b[lacI - IPTG] - K_{cpx}^{decay}.[lacI - IPTG] \tag{6.3}$$

$$\frac{d[tetR]}{dt} = \sum_{i=1,\dots,4} Y_{P_{lac(i)},tetR} \frac{a_{lac}}{1 + K^{lacr(i)}.[lacI]^4} + Y_{P_\lambda tetR} \frac{a_\lambda}{1 + K^\lambda.[cI]^2} + \sum_{i=1,2} Y_{P_{tet(i)},tetR} \frac{a_{tet}}{1 + K^{tet(i)}.[tetR]^2} + Y_{P_{ara}tetR} \frac{a_{ara}}{1 + K^{araC}.[arac]^2} - K^f.[tetR].[aTC] + K^b[tetR - aTC] - K^{decay}.[tetR] \tag{6.4}$$

$$\frac{d[tetR - aTC]}{dt} = K^f.[tetR].[aTC] - K^b[tetR - aTC] - K_{cpx}^{decay}.[tetR - aTC] \tag{6.5}$$

$$\frac{d[cI]}{dt} = \sum_{i=1,\dots,4} Y_{P_{lac(i)},cI} \frac{a_{lac}}{1 + K^{lacr(i)}.[lacI]^4} + Y_{P_\lambda cI} \frac{a_\lambda}{1 + K^\lambda.[cI]^2} + \sum_{i=1,2} Y_{P_{tet(i)},cI} \frac{a_{tet}}{1 + K^{tet(i)}.[tetR]^2} + Y_{P_{ara}cI} \frac{a_{ara}}{1 + K^{araC}.[arac]^2} - K^{decay}.[cI] \tag{6.6}$$

$$\frac{d[araC]}{dt} = \sum_{i=1,\dots,4} Y_{P_{lac(i)},araC} \frac{a_{lac}}{1 + K^{lacr(i)}.[lacI]^4} + Y_{P_\lambda araC} \frac{a_\lambda}{1 + K^{\lambda}.[cI]^2} + \sum_{i=1,2} Y_{P_{tet(i)},araC} \frac{a_{tet}}{1 + K^{tet(i)}.[tetR]^2} + Y_{P_{ara}araC} \frac{a_{ara}}{1 + K^{araC}.[arac]^2} - K^{decay}.[araC] \quad (6.7)$$

where: a_x is the transcriptional efficiency of the promoters (*lac*, *tet*, λ and *ara*), K^x is the cumulative constants representing protein dimerization and binding to the promoters, K^{decay} is the decay rates of the proteins, K_{cpx}^{decay} is the decay rate of the protein-inducer complex, K^f is the association constant for *lacI*-IPTG/*tetR*-aTC binding and K^b is the dissociation constant for *lacI*-IPTG/*tetR*-aTC binding. The values for these parameters can be found in Table 6.1.

Table 6.1: Model parameters for the genetic toggle switch example, Equations 6.2-6.7 (Dasika and Maranas, 2008).

Parameter	Description	Value
a_{lac}	Transcriptional efficiency of P_{lac} promoter	1.215
a_{tet}	Transcriptional efficiency of P_{tet} promoter	1.215
a_{λ}	Transcriptional efficiency of P_{λ} promoter	2.92
a_{ara}	Transcriptional efficiency of P_{ara} promoter	1.215
K^{λ}	Cumulative constant representing *cI* dimerization and binding to P_{λ} promoter	0.33 nm^{-2}
K^{tet1}	Cumulative constant representing *tetR* dimerization and binding to P_{tet} promoter	0.014 nm^{-2}
K^{tet2}	Cumulative constant representing *tetR* dimerization and binding to P_{tet} promoter	1.4 nm^{-2}
K^{lac1}	Cumulative constant representing *lacI* tetramerization and binding to P_{lac1} promoter	10 nm^{-3}
K^{lac2}	Cumulative constant representing *lacI* tetramerization and binding to P_{lac2} promoter	0.01 nm^{-3}
K^{lac3}	Cumulative constant representing *lacI* tetramerization and binding to P_{lac3} promoter	0.001 nm^{-3}
K^{lac4}	Cumulative constant representing *lacI* tetramerization and binding to P_{lac4} promoter	0.00001 nm^{-3}
K^{ara}	Cumulative constant representing *araC* dimerization and binding to P_{ara} promoter	2.5 nm^{-2}
K^{decay}	Decay rates of proteins	*cI* – 0.0693 s^{-1} *lacI* – 0.0346 s^{-1} *tetR* – 0.0346 s^{-1} *araC* – 0.0115 s^{-1}
K_{cpx}^{decay}	Decay rate of protein-inducer complex	0.0693 s^{-1}
K^{f}	Association constant for *lacI*-IPTG/*tetR*-aTC binding	0.05 nm^{-1} s^{-1}
K^{b}	Dissociation constant for *lacI*-IPTG/*tetR*-aTC binding	0.1

6.2.2 OptCircuit Modelling Framework

Using the basic elements that constitute a genetic circuit (promoter elements, inducers and protein/transcript molecules) OptCircuit is defined using the following sets and variables:

Sets:

$I = \{i\}$ = Set of promoters

$J = \{j\}$ = Set of transcripts

$K = \{k\}$ = Set of inducers

$T = \{t\}$ = Time

Variables:

$P_j(t)$ = Protein level of transcript j at time t

$Y_{ij} \begin{cases} 1 \\ 0 \end{cases}$

Where: $Y_{ij} = 1$ if transcript j is expressed from promoter i and $Y_{ij} = 0$ if otherwise. In genetic circuits, as opposed to digital or binary circuits, the presence/absence of a particular set of interactions alone is insufficient to accurately predict all possible responses. To this effect, kinetic description of each element is embedded into the OptCircuit framework. More specifically for each transcript, j, the ODE that governs the time evolution of the protein is given by equation 6.8. The first term accounts for the cumulative rate of production of the protein j from the promoter elements and the second term represents the first order decay of the protein. A note to remember is that the production of protein j from promoter i is turned ON if the binary variable Y_{ij} is equal to one.

$$\frac{dP_j}{dt} = \sum_i Y_{ij}[Rate\ of\ production\ of\ j\ from\ i] - K^j_{decay} P_j(t) \quad \forall j \tag{6.8}$$

For the optimisation, the desired responses are partitioned into inducer-dependent and inducer-free and are translated into the circuit design. For all cases the objective function is the minimisation of the sum of the squared departures from the target responses at all time points, as highlighted by equation 6.9, Where: $P^{exp}_{j^*}$ denotes the experimentally observed profile.

$$minimise\ Z = \sum_t (P_{j^*}(t) - P^{exp}_{j^*}(t))^2 \tag{6.9}$$

6.2.3 OptCircuit Optimisation Model

The problem of designing a circuit that exhibits a desired response was formulated using mixed integer dynamic optimisation (MIDO) (Bansal *et al.*, 2003; Flores-

Tlacuahuac and Biegler, 2004). Here the objective function (Equation 6.10) models the circuit response imposed by the designer as seen below.

$$min/\max Z = f\left(P_j(t)\right) \tag{6.10}$$

Subject to:

$$\frac{dP_j}{dt} = \sum_i Y_{ij}[Rate\ of\ production\ of\ j\ from\ i] - K^j_{decay}P_j(t) \quad \forall j \tag{6.11}$$

$$\sum_j Y_{ij} \leq P^{max} \quad \forall i \tag{6.12}$$

$$\sum_i Y_{ij} \leq T^{max} \quad \forall j \tag{6.13}$$

$$\sum_i \sum_j Y_{ij} \leq M_{max} \tag{6.14}$$

Where: P^{max} denotes an upper limit on the number of transcripts a particular promoter i can express, T^{max} denotes an upper limit on the number of times a particular transcript j can be expressed from different promoters and M_{max} imposes a limit on the total number of promoter-transcript pairs within the circuit. The Boolean constraints highlighted in Equations 6.12-6.14 offer flexibility in incorporating the design of an existing biological circuit to determine its behaviour.

The OptCircuit framework was then applied with the developed meshless ANN-RK4 framework. In order to determine if the ANN-RK4 framework could be applied the OptCircuit model was analysed as an NLP formulation. The result of this investigation is presented in the next section.

6.3 Application of the ANN-RK4 Framework to OptCircuit

The OptCircuit framework detailed a system of ODEs that can be exploited in order to determine the most optimum structure from a list of parts. In order to determine how effective the developed meshless ANN-RK4 model could simulate the OptCircuit framework, the system of ODEs were modelled as an NLP problem. The binary inducer-promotor relationships were fixed for each given structure (Figure 6.3), with each structure modelled separately. Each structure was analysed and the

binary-relationships for it to form were inferred then simulated as separate NLP problems. This resulted in three models where the inferred binary relationships were fixed and optimised. The ANN-RK4 framework can be utilised for MINLP problems, however the study of systems architecture problems is considered to be preparation for future work of this thesis, and so the MINLP was converted to an NLP for ease of optimisation. The measures of success for the results were the levels of IPTG and aTC within the systems. Structure A details a system that produces LacI from the p_{tet_2} promotor. The inducers, IPTG and aTC, are not present within this system, therefore the results from the optimisation of this structure (Figure 6.5) should reflect this. This means that for structure A to form both the promotor and proteinmust be present, but the inducers should be absent. Each structure that was found from OptCircuit in its initial search was tested with the ANN-RK4 framework. The results for structure A are presented in Figure 6.4, structure B in Figure 6.6 and structure C in Figure 6.7. Each corresponding structure from Figure 6.3 is presented with the results for direct comparison.

From the results presented in Figure 6.5 it is noted that the ANN-RK4 framework can accurately identify the necessary proteins needed to form structure A, which are tetR and LacI. The levels of the two promotors, LacI and tetR, increase with every iteration, which is to be expected as the optimum structure (Figure 6.5e) has both of these proteins expressed. The levels of the inducers IPTG and aTC remain at zero with every iteration, which is also expected as they cannot be present in order for structure A to exist. If aTC was present it would supress tetR from forming, and therefore would alter the structure of the circuit. Similarly if the inducer IPTG is present it would supress LacI from forming, and again would alter the structure. The promotor-inducer relationships that are expected in order for structures B and C to assemble are shown in Figures 6.6 and 6.7 respectively.

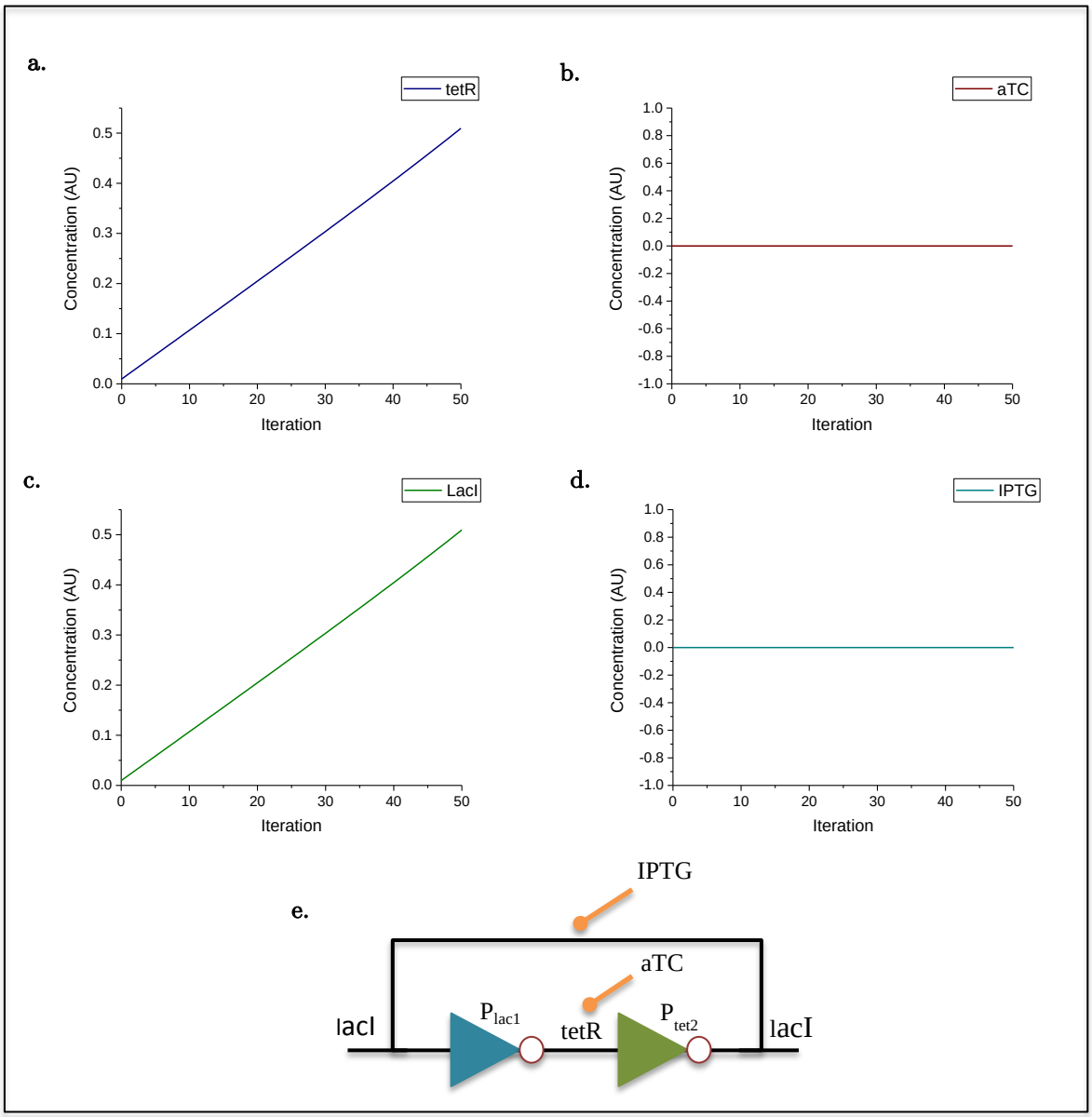

Figure 6.5: The ANN-RK4 simulation results for structure A found by OptCircuit. The results show the iterative evolution of search for the optimum concentration levels of each protein, **a.** The concentration levels of tetR, **b.** the concentration levels of aTC, **c.** The concentration levels of LacI, **d.** The concentration levels of IPTG and **e.** Structure A found from OptCircuit. The model was run for 50 iterations and solved using CONOPT3, total CPU time = 9 seconds, CPU: Intel Core™2 Duo E7400 2.8GHz.

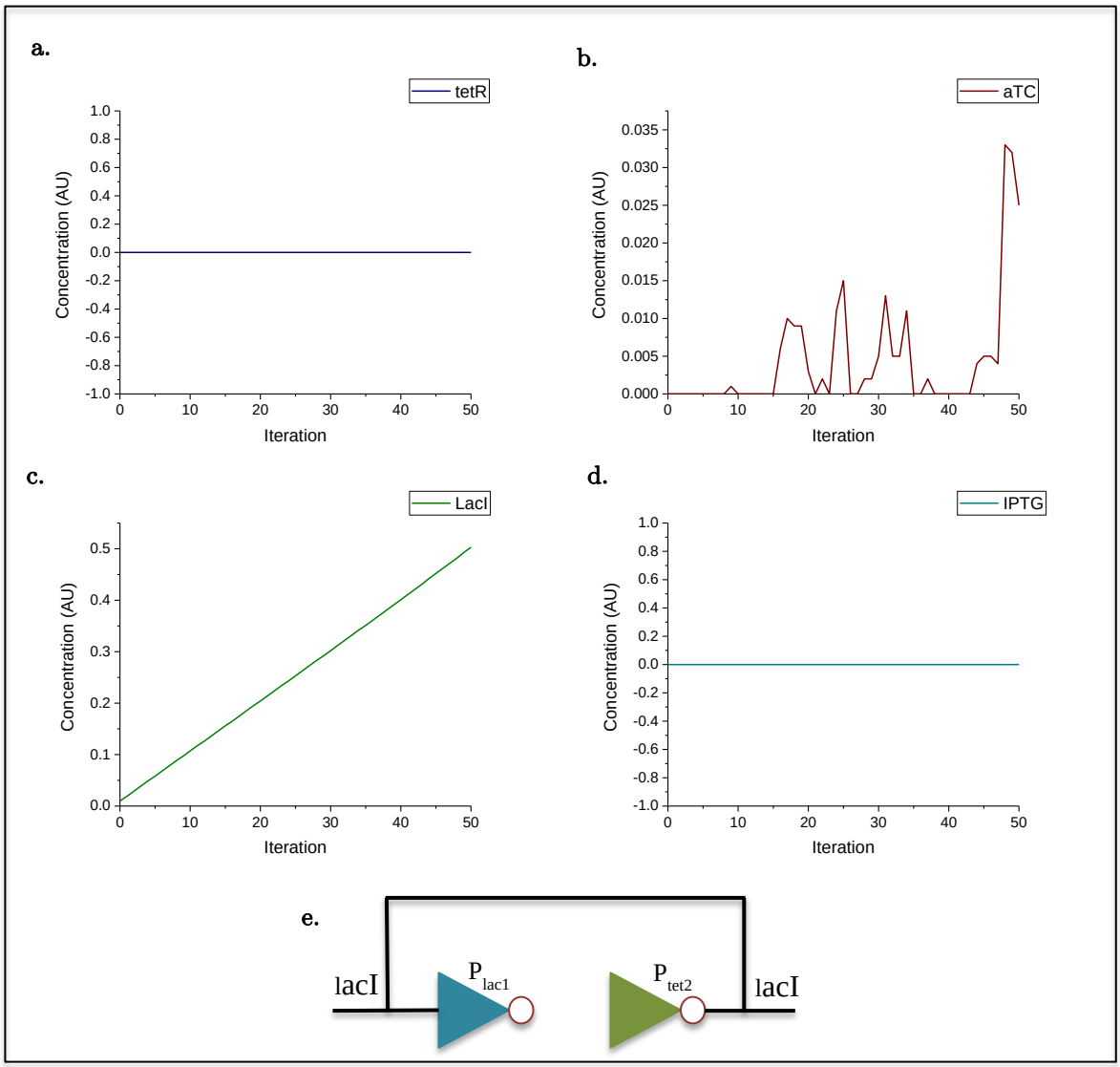

Figure 6.6: The ANN-RK4 simulation results for structure B found by OptCircuit. The results show the iterative evolution of search for the optimum concentration levels of each protein, **a.** The concentration levels of tetR, **b.** the concentration levels of aTC, **c.** The concentration levels of LacI, **d.** The concentration levels of IPTG and **e.** Structure B found from OptCircuit. The model was run for 50 iterations and solved using CONOPT3, total CPU time = 7 seconds, CPU: Intel Core™2 Duo E7400 2.8GHz.

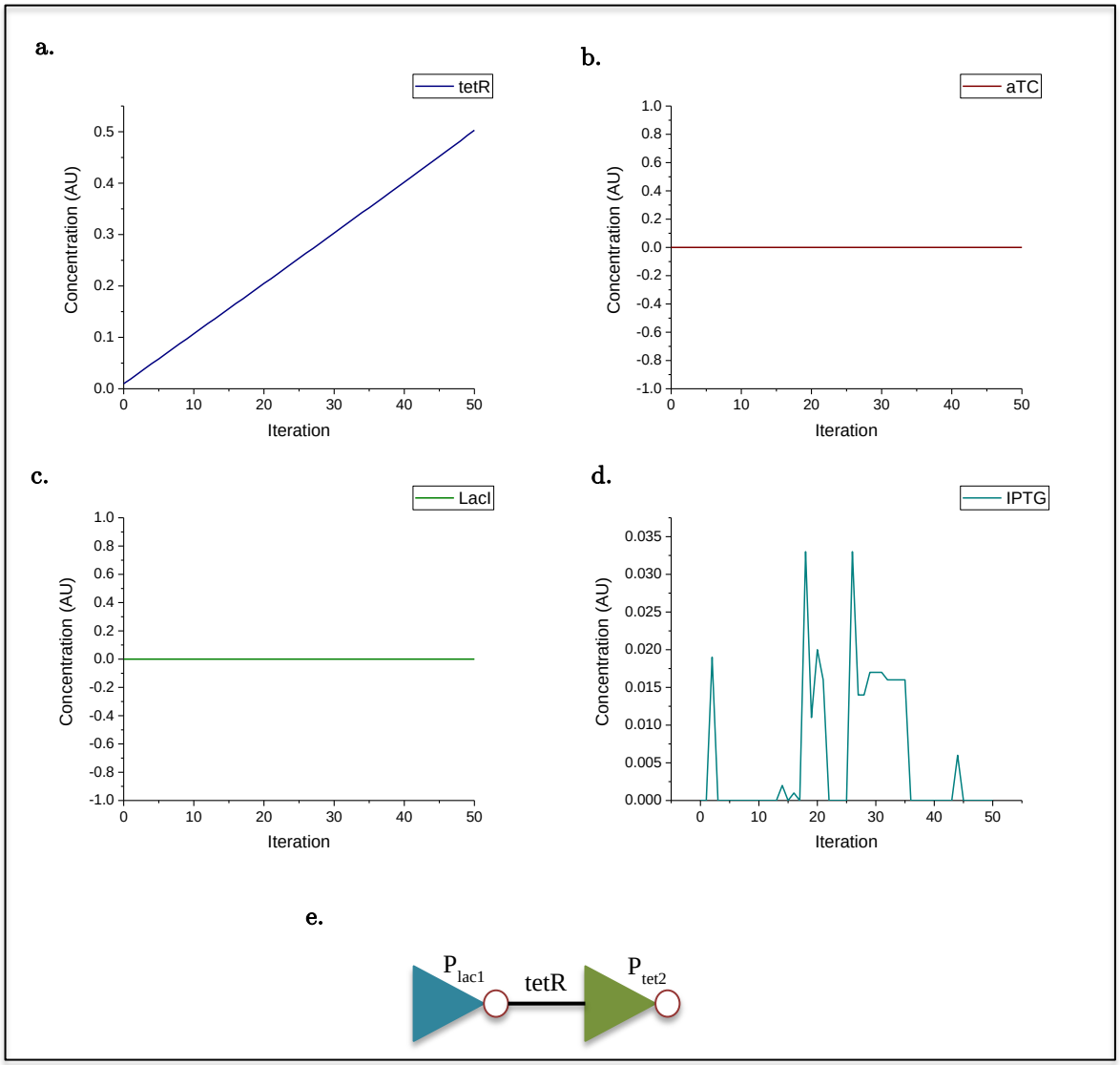

Figure 6.7: The ANN-RK4 simulation results for structure C found by OptCircuit. The results show the iterative evolution of search for the optimum concentration levels of each protein, **a.** The concentration levels of tetR, **b.** the concentration levels of aTC, **c.** The concentration levels of LacI, **d.** The concentration levels of IPTG and **e.** Structure C found from OptCircuit. The model was run for 50 iterations and solved using CONOPT3, total CPU time = 7 seconds, CPU: Intel Core™2 Duo E7400 2.8GHz.

The results presented in Figures 6.6 and 6.7 conform to the schematic of the structures found from OptCircuit. Both show the respective promotors absent within the structures due to their corresponding inducer acting upon them. For example, the concentration level of tetR in Figure 6.5a is zero and this is due to the action of aTC (Figure 6.6b). When aTC is present the tetR protein is supressed and this then

leads to the expression of LacI from the p_{tet_2} promotor. The inducer-promotor relationship is also seen in results presented in Figure 6.7, where LacI is not present due to the action of IPTG.

6.4 Conclusion

From the results presented in Figures 6.5-6.7 it is apparent that the ANN-RK4 framework is capable of simulating systems architecture problems for fixed structures. Each of the results conforms to those found in the initial study of OptCircuit (Dasika and Maranas, 2008). This further enhances the validity of the findings from utilising the developed meshless framework in Chapters 4 and 5, and also shows that it is a viable option for various types of systems modelling. Further work into utilising this framework can be to optimise the structures from a list of well-characterised, biologically compatible parts, as well as to form new structures. This would invariably return this model to a mixed integer nonlinear programming model and would require optimisation of the ANN-RK4 framework to deal with binary problems. The work presented in this chapter shows the possible capabilities of the developed ANN-RK4 framework and how it can be utilised for synthetic system identification.

7. Concluding Remarks and Future Work

7.1 General Perspective

Over recent years the field of synthetic biology has seen a striking increase in the complexity of the models under investigation. However, close collaborations between modellers and experimentalists are still rare, leading to a decreased availability of data for dynamic modelling. Whilst it can be difficult to obtain full models of biological systems, the examples given in this thesis were found to be the most comprehensive. However, as shown in Chapter 4, error in relaying dynamic equations can render data ill-suited for inference. The situation is further exacerbated by the fact that the biological questions we wish to answer necessitate models of a certain complexity. Interestingly this gives rise to various approaches for mathematical modelling. Some modellers believe that all model structures and associated parameters can be defined *a priori* and that additional parameter estimation or uncertainty analysis is not necessary, as the predictions are based on a physically correct model of reality. Here model parameters are often taken from literature or estimated using various experiments. In this paradigm, any shift from the model predictions constitutes a failure of the model and results in extending or changing the model. Moreover, parameter values in literature are rarely reported along with an assessment of their identification. An additional complication with this approach is that enzymes and proteins tend to behave differently *in vivo* than *in vitro* (Teusink *et al.*, 2000). Whether this is attributable to missing interaction mechanisms (Teusink *et al.*, 2000), the composition of the experimental medium (Vanlier *et al.*, 2009) or variations between cells (Kalita *et al.*, 2011) is unclear.

Another popular approach is to make subjective decisions on which parameters are considered adjustable and perform parameter estimation and model sensitivity analysis over pre-defined physiological ranges (Schmitz *et al.*, 2010; van Eunen *et al.*, 2010). This form of uncertainty analysis is highly pragmatic and can work well if one ensures that all uncertainties relevant to the problem are sufficiently probed, and the assumptions regarding the physiological ranges are justified. However, the effects of such assumptions are rarely reported in literature and more often than not the information reported is insufficient in reproducing analyses. Despite the difficulties it proposes, the future of synthetic biology depends on accurately exploring and reporting the uncertainties present in our mathematical models, inferences and predictions. It has been shown that the uncertainties in the developed mathematical models can be accounted for by using disturbances, and

experimentally these disturbances can be quantified. This thesis provides some practical approaches particularly suitable for modelling dynamic biological systems.

7.2 Summary

The applicability of using a meshless framework through the use of artificial neural networks (ANNs) for numerical integrations of dynamic process models has been demonstrated in this thesis. This integration scheme allows for nonlinear model predictive control (NLMPC), which can then be further evolved to zone control. This can not only control the system output, but can also react to randomised disturbances, which is key for future experimental validation of some of the examples shown. The novelty of controlling a biological system using a single control variable, as well as MIMO control of a system from both feedback loops has also been shown. The main contributions of this work are as follows:

1. **Dynamic Process Simulation:** In Chapter 3 (Section 3.3) the ANN integration method for process system models was presented. Solving an NLP problem in GAMS tested the use of ANNs as a solution approach. This approach showcases the best aspects of the ANNs by providing succinct solutions as well as capturing highly nonlinear characteristics of such processes. The accuracy of the ANN is determined through the hidden layer, and optimisation of this provides the least prediction error and reduced computational time. Results from the ANN solution were validated using MATLAB and comparison with dynamics seen in Chapter 3 (Section 3.4.1) show similar results over the entire simulation. This validation of the ANN implementation highlights its usefulness for solving process system ODEs. The optimal ANN topology was then used for control of the dynamic systems.

2. **Nonlinear Model Predictive Control (NLMPC):** In Chapter 3 (Section 3.5.1) the mathematical formulation for NLMPC is presented. The ANN formulation was embedded within the NLPMC optimisation framework and is solved as an NLP problem. Setting a pre-defined tolerance for the ANN prediction error ensures the accuracy of the model solution. Validation for the use of this approach within controllers is achieved by implementing RK4 to verify the results provided by the ANN. The controller was proficient at set point tracking, zone control and

disturbance rejection as shown in Chapter 4. The overall NLMPC scheme based on ANN implementation has shown good performance in meeting control objectives.

3. **Comparison with OCFE**: Chapter 5 has detailed how the developed meshless framework compares with traditional methods of solving ODE systems, in particular OCFE. It was shown that the new framework gives positive results and in many ways the models developed were better, in terms of accuracy and computational effort, than their OCFE counterparts. These findings solidified the use of the developed framework throughout this thesis.

4. **Systems Architecture:** Chapter 6 detailed how the ANN-RK4 framework can be applied to systems architecture problems where biological structures can be simulated to show how compatible parts can form useful structures. It further enhances the validity of the developed meshless model. The chapter also states how future work on the framework can develop it into an MINLP optimiser for biological structures.

To conclude, this work shows how ANNs can be used to simulate dynamic biological systems through integration. The examples shown in Chapters 4 and 5 show how ANNs can be embedded within an NLMPC controller, and in Chapter 6 can be utilised to simulate NLP forms of MINLP models for systems architecture problems.

7.3 Future Work

There are various avenues that can be explored when furthering this work. Whilst the work conducted so far has shown success in terms of modelling the systems in the examples, further investigation is required to showcase the validity of these findings. This section will focus on future steps to help validate this work.

7.3.1 Experimental Validation

Previously in Chapter 4 the first case study from Fung *et al.* (2005) was introduced, discussed and simulated in GAMS using the ANN-RK4 and NLMPC models. Limitations in experimentally validating this case study were due to the fact that

the original authors (Fung *et al.*, 2005) had misplaced the biological samples containing the bacteria when they moved offices. This loss of the metabolator meant that re-creating it would be difficult, as the promotors used were hard to source. However, there is possibility of re-creating the metabolator and performing experiments with a view to validate the results from the control models. Using the NLMPC results to dictate the levels of glucose (glycolytic flux) to be added to the system it is possible to have a relatively simple experimental setup (Figure 7.1). Both glucose entering the system and samples leaving the system will need to be measured in order to see the effect of having a controlled glucose profile. From researching the possible ways the experiment can occur it became apparent that sampling of the reaction mixture would occur quite heavily around the intervals where glucose feed is entering, in order to see how quickly it is being taken in by the *E. Coli* cells, and samples taken after this interval will show the depletion of glucose back to baseline. The NLMPC control models can be altered to have any profile for AcCoA activity that the user requires. Initially it can be planned that a simple step-up and step-down set point for AcCoA can be modelled. This will allow for a simpler profile of glucose to be added to the experimental system, and this will gauge how easily samples can be taken, as well as the rate of glucose uptake. Once these factors are determined a more complex set point (such as those seen in Figures 4.7 and 4.31) can be implemented. The challenge here is to determine how the efficiency of the *in silico* models can be translated *in vivo*.

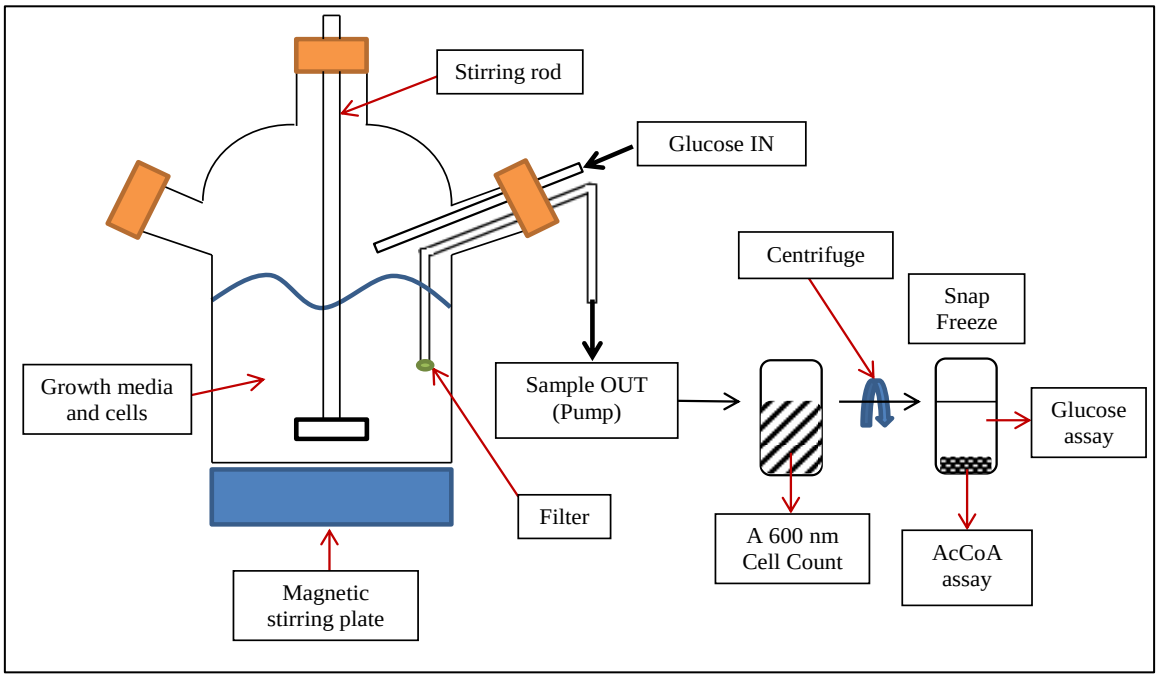

Figure 7.1: Proposed schematic visualisation of the experimental setup

Due to the nature of the experiment, whereby glucose is fed into the system to try and control the activity of AcCoA and ultimately the level of acetate produced, it is important to have a glucose limited medium for the cells to react within. This is important as any residual glucose in the media will greatly alter the results as the cells will have a feed source already present in the reactor. Whilst there are a few types of glucose limited media available, the media selected for this experiment is a mineral media (Hornsten, 1995; Schneebeli and Egli, 2013). Whilst a mineral medium does contain some glucose, the idea is to have it present in limited quantity, and to know the concentration so it can be used to calculate bacterial growth. It is essential to have the *E. Coli* grow in a glucose rich media, then to suspend them into the mineral media after. This ensures that the bacteria will initially propagate and grow in normal conditions, but then starve in the new media due to the low levels of glucose (carbon feed) present. Therefore, when glucose is added to the media as part of the experiment, the cells are likely to uptake the feed source quickly, ensuring that acetate is also produced quickly as a result. It is envisaged that once these experiments are achieved a more comprehensive analysis of the uncertainties in the system can be put back into the model to ensure biological feasibility.

7.3.2 Mixed Integer Modelling

Whilst the framework for a mixed integer model has been presented in Chapter 6, actual simulation of the model needs to be carried out. Although it is envisioned that this will not be a difficult task, there are challenges that arise with using the ANN approach with mixed integer linear models. Although the metabolator is well defined, there is possibility that another metabolite pool when combined with AcCoA could lead to other useful products. Using the MINLP formulation one could select from a range of pathways and choose the best yielding product to engineer. There is also possibility to have multiple pathways present, and to switch between them to give a cell that can produce multiple products at various schedules depending on the system input. This can combine the principles of the multiple-input multiple-output model and evolve it to having separate inputs at different time points.

7.3.3 Uncertainty

Uncertainty will be added to both the metabolator and experimental models from data obtained from experiments. Uncertainties that can be present in the

experimental model could arise from the flux of glucose into the cell from the surrounding media. This process is not instantaneous, and this delay with need to be accounted for when trying to model the system. Assumptions will be made on the cell samples that are analysed, for instance the amount of acetate produced is directly correlated across the whole batch, which may not be the case as cell to cell variability can occur. Furthermore, cells can expire due to the amount of acetate present in the reactor, which can lower the pH past the optimum for *E. Coli* growth and maturation. Intrinsic noise within the metabolator can result in the amplitude of the oscillations to be less than those determined *in silico*.

7.4 Future Scope

There is much scope for this work to help aid in many different issues faced currently in many industries. This section will detail exciting avenues that can use this work at their foundation and help to tackle key issues and challenges that are being researched presently, as well as potentiating future projects.

7.4.1 Total Cell Model

Chapter 4 showcased the metabolator and presented the control of a system using glycolytic flux. The main metabolite in this system, AcCoA, is present in many biochemical reactions within a cell. It is therefore of interest to explore these reactions and try to forge a total cell model. This has applications within the pharmaceutical industry as it can potentially give way to new drug targets and explain physiological changes of a drug *in vivo*. This type of research can shift the boundaries of synthetic and systems biology, as it will require knowledge of both fields to be successful. As a grand challenge of modern times, the development of a predictive model of a living cell can also help in the understanding of how cancers develop. It has already been shown that AcCoA is part of the tricarboxylic acid (TCA) cycle in cells, and this can be simulated and controlled using NADH, calcium and citrate. Research from Wu *et al.* (2007) showed a comprehensive computer model of the TCA cycle and it is possible to utilise the ANN-NLMPC scheme to simulate and control this model. This can then be coupled to the metabolator that has already been simulated in this work, and will build upon the idea of having a total cell model. It is then envisioned that other processes in the cell can be simulated in the same way.

7.4.2 Materials

The main product from the metabolator (Chapter 4) is acetate. Acetate can be utilised in many industries to make useful products such as fabrics, plastics, filters and even surgical products. With many uses it is a viable option to have a fully biologically synthesised acetate fibre, as it will circumvent the need to treat cellulose with chemicals in order to produce a fibre. Using *E. Coli* as a chassis it is possible to also model other production pathways and produce other materials, such as ethanol (Boumba *et al.*, 2013). Research from Ro *et al.* (2006) have already shown how synthetic biology can be used to create an antimalarial precursor, Arteminisin, and create new products from the cells themselves, which can be exploited to create drugs *in situ*. This can also be explored with the framework developed here to help control the amount of drug produced from cells, and could potentially lead to higher yields than chemical based methods of production. Other organisms can also be exploited using synthetic biology such as yeast for the production of alcohols, which are useful in many industries and even algae and fungi for biofuel production. Essentially the kinetics and dynamics of the system need to be known and the framework built here can model the system to control production and can shift dynamics to a continuous phase of product formation.

7.4.3 Experimental Regulation

Both the metabolator and the experimental model propose systems that are able to be controlled using just glucose. If results from the experimental validation of the metabolator are positive, as it is envisioned, this will herald a remarkable discovery in terms of biological regulation. To regulate a dynamic process using a simple controller (glucose) will give great insight into how a biological system can be trained. The theory can then be applied to other biological constructs within a cell chassis, which can then lead to continuous production of a useful products, or give rise to new products of yet unknown use.

7.4.4 Biological Transistors

The scope for Zone NLMPC (Chapter 4) with the metabolator can lead to the idea of having a tuneable biological cell that acts like a transistor in a circuit. Transistors are used to amplify or switch electronic signals and electrical power. As an amplifier the usefulness of a transistor comes from its ability to use a small signal applied between a pair of its terminals to control a much larger signal at another pair of

terminals. As a switch a transistor can be used to turn current on or off in a circuit where the amount of current is determined by other circuit elements. Whilst traditional transistors have many advantages some key disadvantages include that they can age and fail, they are not applicable (or preferred) in systems with high power, high frequency operation (such as televisions) and they are susceptible to damage from very brief electrical and thermal events. The metabolator can be key to creating a biological transistor due to the fundamentals involved in its design. Essentially parts can be created that can fit together in a circuit and placed within a biological chassis. The resulting construct could utilise the cells innate physiology and metabolic processes to create energy, which can be fine-tuned using an auto regulatory profile governed by a Zone NLMPC problem. This however will need to be heavily researched as it is in a very early theoretical stage.

7.5 Concluding Remarks

Control of biological systems is a challenging task as trying to control a dynamic process *in silico* may not necessarily translate to the same level of control *in vivo*. Biological variability ensures that uncertainties need to be accounted for and this data utilised within models generate more feasible results. However, the work presented here shows that the theory of control can be applied to biological systems with relative success. This work presents the first step in allowing for feasible biological control of systems, and marks a new avenue to try and generate computer models that behave sympathetically with cellular experiments.

8. Bibliography

ADAI, A. T., DATE, S. V., WIELAND, S. & MARCOTTE, E. M. 2004. LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks. *Journal of Molecular Biology,* 340, 179-190.

ALLGOWER, F., FINDEISEN, R. & NAGY, Z. K. 2004. Nonlinear model predictive control: From theory to application. *Journal of the Chinese Institute of Chemical Engineers,* 35, 299-315.

ALPER, H., FISCHER, C., NEVOIGT, E. & STEPHANOPOULOS, G. 2005. Tuning genetic control through promoter engineering. *Proceedings of the National Academy of Sciences of the United States of America,* 102, 12678-12683.

AMARO, A. C. S. & BARBOSA-POVOA, A. 2009. The effect of uncertainty on the optimal closed-loop supply chain planning under different partnerships structure. *Computers & Chemical Engineering,* 33, 2144-2158.

ANDERSON, J. & PAPACHRISTODOULOU, A. 2009. On validation and invalidation of biological models. *Bmc Bioinformatics,* 10.

ANNALURU, N., MULLER, H., MITCHELL, L. A., RAMALINGAM, S., STRACQUADANIO, G., RICHARDSON, S. M., DYMOND, J. S., KUANG, Z., SCHEIFELE, L. Z., COOPER, E. M., CAI, Y., ZELLER, K., AGMON, N., HAN, J. S., HADJITHOMAS, M., TULLMAN, J., CARAVELLI, K., CIRELLI, K., GUO, Z., LONDON, V., YELURU, A., MURUGAN, S., KANDAVELOU, K., AGIER, N., FISCHER, G., YANG, K., MARTIN, J. A., BILGEL, M., BOHUTSKI, P., BOULIER, K. M., CAPALDO, B. J., CHANG, J., CHAROEN, K., CHOI, W. J., DENG, P., DICARLO, J. E., DOONG, J., DUNN, J., FEINBERG, J. I., FERNANDEZ, C., FLORIA, C. E., GLADOWSKI, D., HADIDI, P., ISHIZUKA, I., JABBARI, J., LAU, C. Y. L., LEE, P. A., LI, S., LIN, D., LINDER, M. E., LING, J., LIU, J., LIU, J., LONDON, M., MA, H., MAO, J., MCDADE, J. E., MCMILLAN, A., MOORE, A. M., OH, W. C., OUYANG, Y., PATEL, R., PAUL, M., PAULSEN, L. C., QIU, J., RHEE, A., RUBASHKIN, M. G., SOH, I. Y., SOTUYO, N. E., SRINIVAS, V., SUAREZ, A., WONG, A., WONG, R., XIE, W. R., XU, Y., YU, A. T., KOSZUL, R., BADER, J. S., BOEKE, J. D. & CHANDRASEGARAN, S. 2014. Total Synthesis of a Functional Designer Eukaryotic Chromosome. *Science,* 344, 55-58.

ARKIN, A., ROSS, J. & MCADAMS, H. H. 1998. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics,* 149, 1633-1648.

ATKINSON, M. R., SAVAGEAU, M. A., MYERS, J. T. & NINFA, A. J. 2003. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell,* 113, 597-607.

BABUSKA, I., NOBILE, F. & TEMPONE, R. 2007. A stochastic collocation method for elliptic partial differential equations with random input data. *Siam Journal on Numerical Analysis,* 45, 1005-1034.

BALASUBRAMANIAN, J. & GROSSMANN, I. E. 2002. A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Computers & Chemical Engineering,* 26, 41-57.

BANSAL, V., SAKIZLIS, V., ROSS, R., PERKINS, J. D. & PISTIKOPOULOS, E. N. 2003. New algorithms for mixed-integer dynamic optimization. *Computers & Chemical Engineering,* 27, 647-668.

BARKAI, N. & LEIBLER, S. 2000. Biological rhythms - Circadian clocks limited by noise. *Nature,* 403, 267-268.

BAS, D. & BOYACI, I. H. 2007. Modeling and optimization II: Comparison of estimation capabilities of response surface methodology with artificial neural networks in a biochemical reaction. *Journal of Food Engineering,* 78, 846-854.

BASRI, M., RAHMAN, R., EBRAHIMPOUR, A., SALLEH, A. B., GUNAWAN, E. R. & RAHMAN, M. B. A. 2007. Comparison of estimation capabilities of response surface methodology (RSM) with artificial neural network (ANN) in lipase-catalyzed synthesis of palm-based wax ester. *Bmc Biotechnology,* 7.

BASU, S., GERCHMAN, Y., COLLINS, C. H., ARNOLD, F. H. & WEISS, R. 2005. A synthetic multicellular system for programmed pattern formation. *Nature,* 434, 1130-1134.

BATHELT, C., SCHMID, R. D. & PLEISS, J. 2002. Regioselectivity of CYP2B6: homology modeling, molecular dynamics simulation, docking. *Journal of Molecular Modeling,* 8, 327-335.

BAUGHMAN, D. R. & LIU, Y. A. 1995. *Neural networks in bioprocessing and chemical engineering*. Academic Press, p.488.

BEAL, J., LU, T. & WEISS, R. 2011. Automatic Compilation from High-Level Biologically-Oriented Programming Language to Genetic Regulatory Networks. *Plos One,* 6.

BEQUETTE, B. W. 1991. NONLINEAR CONTROL OF CHEMICAL PROCESSES - A REVIEW. *Industrial & Engineering Chemistry Research,* 30, 1391-1413.

BEQUETTE, B. W. 2009. *Model predictive control: an overview and selected applications (webinar)*. Available at: http://www.aiche.org/academy/webinars/model-predictive-control-overview-and-selected-applications (Accessed: 12 February 2016).

BERGAMINI, M. L., AGUIRRE, P. & GROSSMANN, I. 2005. Logic-based outer approximation for globally optimal synthesis of process networks. *Computers & Chemical Engineering,* 29, 1914-1933.

BERGAMINI, M. L., GROSSMANN, I., SCENNA, N. & AGUIRRE, P. 2008. An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Computers & Chemical Engineering,* 32, 477-493.

BIEGLER, L. T. 2007. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing,* 46, 1043-1053.

BONAMI, P., BIEGLER, L. T., CONNA, A. R., CORNUEJOLS, G., GROSSMANN, I. E., LAIRD, C. D., LEE, J., LODI, A., MARGOT, F., SAWAYA, N. & WACHTER, A. 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization,* 5, 186-204.

BOUMBA, V. A., KOURKOUMELIS, N., GOUSIA, P., ECONOMOU, V., PAPADOPOULOU, C. & VOUGIOUKLAKIS, T. 2013. Modeling microbial ethanol production by E. coli under aerobic/anaerobic conditions: Applicability to real postmortem cases and to postmortem blood derived microbial cultures. *Forensic Science International,* 232, 191-198.

BRATSUN, D., VOLFSON, D., TSIMRING, L. S. & HASTY, J. 2005. Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences of the United States of America,* 102, 14593-14598.

BUCHLER, N. E., GERLAND, U. & HWA, T. 2003. On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences of the United States of America,* 100, 5136-5141.

BUCHLER, N. E., GERLAND, U. & HWA, T. 2005. Nonlinear protein degradation and the function of genetic circuits. *Proceedings of the National Academy of Sciences of the United States of America,* 102, 9559-9564.

BURKE, J. V. & HAN, S. P. 1989. A ROBUST SEQUENTIAL QUADRATIC-PROGRAMMING METHOD. *Mathematical Programming,* 43, 277-303.

BUZZI-FERRARIS, G. & MANENTI, F. 2012. BzzMath: Library Overview and Recent Advances in Numerical Methods. *22 European Symposium on Computer Aided Process Engineering,* 30, 1312-1316.

BYRD, R. H., NOCEDAL, J. & WALTZ, R. A. 2006. KNITRO: An integrated package for nonlinear optimization. *Large-Scale Nonlinear Optimization,* 83, 35-59.

CAETANO, C., REIS, J. L., AMORIM, J., LEMES, M. R. & DAL PINO, A. 2011. Using Neural Networks to Solve Nonlinear Differential Equations in Atomic and Molecular Physics. *International Journal of Quantum Chemistry,* 111, 2732-2740.

CAI, L. & WHITE, R. E. 2012. Lithium ion cell modeling using orthogonal collocation on finite elements. *Journal of Power Sources,* 217, 248-255.

CAMPBELL, A. M. 2005. Meeting report: synthetic biology Jamboree for undergraduates. *Cell biology education,* 4, 19-23.

CAREY, G. F. & FINLAYSON, B. A. 1975. ORTHOGONAL COLLOCATION ON FINITE-ELEMENTS. *Chemical Engineering Science,* 30, 587-596.

CHANG, P. W., PATTEN, T. W. & FINLAYSON, B. A. 1979. Collocation and Galerkin finite-element methods for viscoelastic fluid-flow.1. Description of method and problems with fixed geometry. *Computers & Fluids,* 7, 267-283.

CHEON, M. S., AHMED, S. & AL-KHAYYAL, F. 2006. A branch-reduce-cut algorithm for the global optimization of probabilistically constrained linear programs. *Mathematical Programming,* 108, 617-634.

CLARKE, D. W., MOHTADI, C. & TUFFS, P. S. 1987. Generalized predictive control. 1. The basic algorithm. *Automatica,* 23, 137-148.

CONRAD, E., MAYO, A. E., NINFA, A. J. & FORGER, D. B. 2008. Rate constants rather than biochemical mechanism determine behaviour of genetic clocks. *Journal of the Royal Society Interface,* 5, S9-S15.

CUTHRELL, J. E. & BIEGLER, L. T. 1987. On the optimization of differential-algebraic process systems. *Aiche Journal,* 33, 1257-1270.

CUTLER, C. R. & RAMAKER, B. L. 1980. Dynamic matric control – a computer control algorithm. *Joint Automatic Control Conference*. San Francisco, California.

DANINO, T., PRINDLE, A., KWONG, G. A., SKALAK, M., LI, H., ALLEN, K., HASTY, J. & BHATIA, S. N. 2015. Programmable probiotics for detection of cancer in urine. *Science Translational Medicine,* 7.

DASIKA, M. S. & MARANAS, C. D. 2008. OptCircuit: An optimization based method for computational design of genetic circuits. *Bmc Systems Biology,* 2.

DE JONG, M. D., THANH, T. T., KHANH, T. H., HIEN, V. M., SMITH, G. J. D., CHAU, N. V., CAM, B. V., QUI, P. T., HA, D. Q., GUAN, Y., PEIRIS, J. S. M., HIEN, T. T. & FARRAR, J. 2005. Brief report - Oseltamivir resistance during treatment of influenza A (H5N1) infection. *New England Journal of Medicine,* 353, 2667-2672.

DENSMORE, D., HSIAU, T. H. C., KITTLESON, J. T., DELOACHE, W., BATTEN, C. & ANDERSON, J. C. 2010. Algorithms for automated DNA assembly. *Nucleic Acids Research,* 38, 2607-2616.

DIMITRIADIS, V. D. & PISTIKOPOULOS, E. N. 1995. Flexibility analysis of dynamic-systems. *Industrial & Engineering Chemistry Research,* 34, 4451-4462.

DOMINGUEZ, L. F. & PISTIKOPOULOS, E. N. 2010. Multiparametric programming based algorithms for pure integer and mixed-integer bilevel programming problems. *Computers & Chemical Engineering,* 34, 2097-2106.

DOMINGUEZ, L. F. & PISTIKOPOULOS, E. N. 2011. Recent Advances in Explicit Multiparametric Nonlinear Model Predictive Control. *Industrial & Engineering Chemistry Research,* 50, 609-619.

DOYLE, M., FULLER, T. F. & NEWMAN, J. 1993. Modeling of galvanostatic charge and discharge of the lithium polymer insertion cell. *Journal of the Electrochemical Society,* 140, 1526-1533.

DRUD, A. 1985. CONOPT- A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming,* 31, 153-191.

DUA, V. 2006. Stability Analysis of Nonlinear Model Predictive Control: An Optimization Based Approach. *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering,* 21, 1287-1292.

DUA, V. 2010. A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Engineering Research & Design,* 88, 55-60.

DUA, V. 2011. An Artificial Neural Network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Computers & Chemical Engineering,* 35, 545-553.

DUA, V. & DUA, P. 2012. A Simultaneous Approach for Parameter Estimation of a System of Ordinary Differential Equations, Using Artificial Neural Network Approximation. *Industrial & Engineering Chemistry Research,* 51, 1809-1814.

DURAN, M. A. & GROSSMANN, I. E. 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming,* 36, 307-339.

ELKADY, M. A., BELL, B. D., CARVALHO, V. F., BURCHETT, R. C., HAPP, H. H. & VIERATH, D. R. 1986. Assessment of real-time optimal voltage control. *Ieee Transactions on Power Systems,* 1, 98-107.

ELOWITZ, M. B. & LEIBLER, S. 2000. A synthetic oscillatory network of transcriptional regulators. *Nature,* 403, 335-338.

ENDY, D. 2005. Foundations for engineering biology. *Nature,* 438, 449-453.

ENDY, D. & YAFFE, M. B. 2003. Signal transduction - Molecular monogamy. *Nature,* 426, 614-615.

FARKAS, T., REV, E., CZUCZAI, B., FONYO, Z. & LELKES, Z. 2005. R-graph-based distillation column superstructure and MINLP model. *European Symposium on Computer-Aided Process Engineering-15, 20A and 20B,* 20a-20b, 889-894.

FEHLBERG, E. 1969. Classical fifth-order and seveth-order Runge-Kutta formulas with stepsize control. *Computing,* 4, 93-&.

FENG, X. J., HOOSHANGI, S., CHEN, D., LI, G. Y., WEISS, R. & RABITZ, H. 2004. Optimizing genetic circuits by global sensitivity analysis. *Biophysical Journal,* 87, 2195-2202.

FERBER, D. 2004. Synthetic biology: Microbes made to order. *Science,* 303, 158-161.

FERRAMOSCA, A., LIMON, D., GONZALEZ, A. H., ODLOAK, D. & CAMACHO, E. F. 2010. MPC for tracking zone regions. *Journal of Process Control,* 20, 506-516.

FILICI, C. 2008. On a neural approximator to ODEs. *Ieee Transactions on Neural Networks,* 19, 539-543.

FLOREA, M., HAGEMANN, H., SANTOSA, G., ABBOTT, J., MICKLEM, C. N., SPENCER-MILNES, X., GARCIA, L. D. A., PASCHOU, D., LAZENBATT, C., KONG, D., CHUGHTAI, H., JENSEN, K., FREEMONT, P. S., KITNEY, R., REEVE, B. & ELLIS, T. 2016. Engineering control of bacterial cellulose production using a genetic toolkit and a new cellulose-producing strain. *Proceedings of the National Academy of Sciences of the United States of America,* 113, E3431-E3440.

FLORES-TLACUAHUAC, A. & BIEGLER, L. T. 2005. A robust and efficient Mixed-Integer Non-linear Dynamic Optimization approach for simultaneous design and control. *European Symposium on Computer-Aided Process Engineering-15, 20A and 20B,* 20a-20b, 67-72.

FLOUDAS, C. A. 1995. *Nonlinear and mixed-integer optimization fundamentals and applications*. New York: Oxford University Press, p.4.

FOGEL, D. B., WASSON, E. C. & BOUGHTON, E. M. 1995. Evolving neural networks for detecting breast-cancer. *Cancer Letters,* 96, 49-53.

FOX, R. 2005. Directed molecular evolution by machine learning and the influence of nonlinear interactions. *Journal of Theoretical Biology,* 234, 187-199.

FRANCOIS, P. & HAKIM, V. 2004. Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences of the United States of America,* 101, 580-585.

FULLER, T. F., DOYLE, M. & NEWMAN, J. 1994. Simulation and optimization of the dual lithium insertion cell. *Journal of the Electrochemical Society,* 141, 1-10.

FUNG, E., WONG, W. W., SUEN, J. K., BULTER, T., LEE, S. G. & LIAO, J. C. 2005. A synthetic gene-metabolic oscillator. *Nature,* 435, 118-122.

GARCIA, C. E. & MORSHEDI, A. M. 1986. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications,* 46, 73-87.

GARCIA, C. E., PRETT, D. M. & MORARI, M. 1989. Model predictive control – theory and practice – a survey. *Automatica,* 25, 335-348.

GARDNER, T. S., CANTOR, C. R. & COLLINS, J. J. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature,* 403, 339-342.

GE, S. S., HANG, C. C. & ZHANG, T. 1999. Nonlinear adaptive control using neural networks and its application to CSTR systems. *Journal of Process Control,* 9, 313-323.

GILL, P. E., MURRAY, W. & SAUNDERS, M. A. 2005. SNOPT: and SQP algorithm for large-scale constrained optimization. *SIAM J Optim. 47(4): p.99-131.*

GILLESPIE, D. T. 2007. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry,* 58, 35-55.

GILMAN, A. & ARKIN, A. P. 2002. Genetic "Code": Representations and dynamical models of genetic components and networks. *Annual Review of Genomics and Human Genetics,* 3, 341-369.

GLASS, L., PERKINS, T. J., MASON, J., SIEGELMANN, H. T. & EDWARDS, R. 2005. Chaotic dynamics in an electronic model of a genetic network. *Journal of Statistical Physics,* 121, 969-994.

GOH, C. K., TEOH, E. J. & TAN, K. C. 2008a. Hybrid multiobjective evolutionary design for artificial neural networks. *Ieee Transactions on Neural Networks,* 19, 1531-1548.

GOH, K. I., KAHNG, B. & CHO, K. H. 2008b. Sustained oscillations in extended genetic oscillatory systems. *Biophysical Journal,* 94, 4270-4276.

GOODWIN, B. C. 1965. Oscillatory behavior in enzymatic control pro-cesses. *Advance Enzyme Regulat,* 3, 425-438.

GROSMAN, B., DASSAU, E., ZISSER, H., JOVANOVIC, L. & DOYLE, F. J. 2010. Automatic Zone Model Predictive Control for the Artificial Pancreatic beta-Cell. *Diabetes,* 59, A144-A144.

GROSSMANN, I. E. & FLOUDAS, C. A. 1987. Active constraint strategy for flexibility analysis in chemical processes. *Computers & Chemical Engineering,* 11, 675-693.

GUANTES, R. & POYATOS, J. F. 2006. Dynamical principles of two-component genetic oscillators. *Plos Computational Biology,* 2, 188-197.

GUCKENHEIMER, J., & HOLMES, P. J. 1983. *Nonlinear oscillations, dynamical systems, and bifurcation of vector fields*. Springer-Verlag, Berlin.

HAIRER, E. & WANNER, G. 1999. Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics,* 111, 93-111.

HALDIMANN, A. & WANNER, B. L. 2001. Conditional-replication, integration, escision, and retrieval plasmid-host systems for gene structure-function studies of bacteria. *Journal of Bacteriology*. 183, 6384-6393

HALLALE, N. & LIU, F. 2001. Refinery hydrogen management for clean fuels production. *Advances in Environmental Research,* 6, 81-98.

HARTWELL, L. H., HOPFIELD, J. J., LEIBLER, S. & MURRAY, A. W. 1999. From molecular to modular cell biology. *Nature,* 402, C47-C52.

HASTY, J., MCMILLEN, D. & COLLINS, J. J. 2002. Engineered gene circuits. *Nature,* 420, 224-230.

HASTY, J., MCMILLEN, D., ISAACS, F. & COLLINS, J. J. 2001. Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews Genetics,* 2, 268-279.

HELLINGA, H. W. & MARVIN, J. S. 1998. Protein engineering and the development of generic biosensors. *Trends in Biotechnology,* 16, 183-189.

HERRERA, S. 2005. Synthetic biology offers alternative pathways to natural products. *Nature Biotechnology,* 23, 270-271.

HILBORN, R. C. & ERWIN, J. D. 2008. Stochastic coherence in an oscillatory gene circuit model. *Journal of Theoretical Biology,* 253, 349-354.

HOOSHANGI, S., THIBERGE, S. & WEISS, R. 2005. Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proceedings of the National Academy of Sciences of the United States of America,* 102, 3581-3586.

HORNIK, K., STINCHCOMBE, M. & WHITE, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks,* 2, 359-366.

HORNSTEN, E. G. 1995. On culturing Escherichia-Coli on a mineral salts medium during anaerobic conditions. *Bioprocess Engineering,* 12, 157-162.

HU, J., LIU, X. & MA, P. X. 2008. Induction of osteoblast differentiation phenotype on poly(L-lactic acid) nanofibrous matrix. *Biomaterials,* 29, 3815-3821.

ISAACS, F. J. & COLLINS, J. J. 2005. Plug-and-play with RNA. *Nature Biotechnology,* 23, 306-307.

ISAACS, F. J., HASTY, J., CANTOR, C. R. & COLLINS, J. J. 2003. Prediction and measurement of an autoregulatory genetic module. *Proceedings of the National Academy of Sciences of the United States of America,* 100, 7714-7719.

JOHANSEN, T. A. 2004. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica,* 40, 293-300.

JOHANSEN, T. A. & IEEE. 2002. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. 41st IEEE Conference on Decision and Control, Dec 10-13 2002 Las Vegas, Nv. 2768-2773.

JUNE, C. H. & LEVINE, B. L. 2015. T cell engineering as therapy for cancer and HIV: our synthetic future. *Philosophical Transactions of the Royal Society B-Biological Sciences,* 370.

KALITA, M. K., SARGSYAN, K., TIAN, B., PAULUCCI-HOLTHAUZEN, A., NAJM, H. N., DEBUSSCHERE, B. J. & BRASIER, A. R. 2011. Sources of Cell-to-cell Variability in Canonical Nuclear Factor-kappa B (NF-kappa B) Signaling Pathway Inferred from Single Cell Dynamic Images. *Journal of Biological Chemistry,* 286, 37741-37757.

KALOS, M. & JUNE, C. H. 2013. Adoptive T Cell Transfer for Cancer Immunotherapy in the Era of Synthetic Biology. *Immunity,* 39, 49-60.

KAWATHEKAR, R. & RIGGS, J. B. 2007. Nonlinear model predictive control of a reactive distillation column. *Control Engineering Practice,* 15, 231-239.

KELWICK, R., KOPNICZKY, M., BOWER, I., CHI, W., CHIN, M. H. W., FAN, S., PILCHER, J., STRUTT, J., WEBB, A. J., JENSEN, K., STAN, G.-B., KITNEY, R. & FREEMONT, P. 2015. A Forward-Design Approach to Increase the Production of Poly-3-Hydroxybutyrate in Genetically Engineered Escherichia coli. *Plos One,* 10.

KHAJEHPOUR, M., FARHADI, F. & PISHVAIE, M. R. 2009. Reduced superstructure solution of MINLP problem in refinery hydrogen management. *International Journal of Hydrogen Energy,* 34, 9233-9238.

KHODR, H. M. & MARTINEZ-CRESPO, J. 2009. Integral methodology for distribution systems reconfiguration based on optimal power flow using Benders decomposition technique. *Iet Generation Transmission & Distribution,* 3, 521-534.

KIANI, S., CHAVEZ, A., TUTTLE, M., HAL, R. N., CHARI, R., TER-OVANESYAN, D., QIAN, J., PRUITT, B. W., BEAL, J., VORA, S., BUCHTHAL, J., KOWAL, E. J. K., EBRAHIMKHANI, M. R., COLLINS, J. J., WEISS, R. & CHURCH, G. 2015. Cas9 gRNA engineering for genome editing, activation and repression. *Nature Methods,* 12, 1051-1054.

KIER, L. B. & HALL, L. H. 1976. Molecular connectivity. 7. Specific treatment of heteroatoms. *Journal of Pharmaceutical Sciences,* 65, 1806-1809.

KLEIN, A. H., SHULLA, A., REIMANN, S. A., KEATING, D. H. & WOLFE, A. J. 2007. The intracellular concentration of acetyl phosphate in Escherichia coli is sufficient for direct phosphorylation of two-component response regulators. *Journal of Bacteriology,* 189, 5574-5581.

KOCIS, G. R. & GROSSMANN, I. E. 1987. Relaxation strategy for the structural optimization of process flow sheets. *Industrial & Engineering Chemistry Research,* 26, 1869-1880.

KOCIS, G. R. & GROSSMANN, I. E. 1988. Global optimization of nonconvex mixed-integer nonlinear-programming (MINLP) problems in process synthesis. *Industrial & Engineering Chemistry Research,* 27, 1407-1421.

KOMAROVA, N. L. & WODARZ, D. 2010. ODE models for oncolytic virus dynamics. *Journal of Theoretical Biology,* 263, 530-543.

KOTULA, J. W., KERNS, S. J., SHAKET, L. A., SIRAJ, L., COLLINS, J. J., WAY, J. C. & SILVER, P. A. 2014. Programmable bacteria detect and record an environmental signal in the mammalian gut. *Proceedings of the National Academy of Sciences of the United States of America,* 111, 4838-4843.

KROM, R. J., BHARGAVA, P., LOBRITZ, M. A. & COLLINS, J. J. 2015. Engineered Phagemids for Nonlytic, Targeted Antibacterial Therapies. *Nano Letters,* 15, 4808-4813.

LAGARIS, I. E., LIKAS, A. & FOTIADIS, D. I. 1998. Artificial neural networks for solving ordinary and partial differential equations. *Ieee Transactions on Neural Networks,* 9, 987-1000.

LAGARIS, I. E., LIKAS, A. C. & PAPAGEORGIOU, D. G. 2000. Neural-network methods for boundary value problems with irregular boundaries. *Ieee Transactions on Neural Networks,* 11, 1041-1049.

LASDON, L. S., FOX, R. L. & RATNER, M. W. 1974. Nonlinear optimization using generalised reduced gradient method. *Revue Francaise D Automatique Informatique Recherche Operationnelle,* 8, 73-103.

LEE, J. H., COOLEY, B. L. & AMER AUTOMAT CONTROL, C. 1997. Stable min-max control for state-space systems with bounded input matrix. 1997 American Control Conference, Jun 04-06 1997 Albuquerque, Nm. 2945-2949.

LEE, P. L. & SULLIVAN, G. R. 1988. Generic model control (GMC). *Computers & Chemical Engineering,* 12, 573-580.

LEWIS, J. 2003. Autoinhibition with transcriptional delay: A simple mechanism for the zebrafish somitogenesis oscillator. *Current Biology,* 13, 1398-1408.

LIENERT, F., LOHMUELLER, J. J., GARG, A. & SILVER, P. A. 2014. Synthetic biology in mammalian cells: next generation research tools and therapeutics. *Nature Reviews Molecular Cell Biology,* 15, 95-107.

LIENQUEO, M. E., SALGADO, J. C., GIAVERINI, O. & ASENJO, J. A. 2009. Computer-aided design to select optimal polypeptide tags to assist the purification of recombinant proteins. *Separation and Purification Technology,* 65, 86-94.

LOOSS, G., & JOSEPH, D. D. 1980. *Elementary stability and bifurcation theory.* Springer-Verlag, New York.

LU, T. K. & COLLINS, J. J. 2009. Engineered bacteriophage targeting gene networks as adjuvants for antibiotic therapy. *Proceedings of the National Academy of Sciences of the United States of America,* 106, 4629-4634.

LUISI, P. L. 2002. Toward the engineering of minimal living cells. *Anatomical Record,* 268, 208-214.

MA, H., HU, J. & MA, P. X. 2010. Polymer Scaffolds for Small-Diameter Vascular Tissue Engineering. *Advanced Functional Materials,* 20, 2833-2841.
MARCHISIO, M. A. & STELLING, J. 2008. Computational design of synthetic gene circuits with composable parts. *Bioinformatics,* 24, 1903-1910.
MARCHISIO, M. A. & STELLING, J. 2011. Automatic Design of Digital Synthetic Gene Circuits. *Plos Computational Biology,* 7.
MAYNE, D. Q., RAWLINGS, J. B., RAO, C. V. & SCOKAERT, P. O. M. 2000. Constrained model predictive control: Stability and optimality. *Automatica,* 36, 789-814.
MEADOWS, E. S., & RAWLINGS, J. B. 1997. *Model predictive control.* In: Henson, M. A., & Seborg, D. E., Editors, *Nonlinear process control*, chapter 5. Prentice Hall.
MENEZES, A. A., CUMBERS, J., HOGAN, J. A. & ARKIN, A. P. 2015. Towards synthetic biological approaches to resource utilization on space missions. *Journal of the Royal Society Interface,* 12.
MESTER, D. I., RONIN, Y. I., NEVO, E. & KOROL, A. O. 2004. Fast and high precision algorithms for optimization in large-scale genomic problems. *Computational Biology and Chemistry,* 28, 281-290.
MIT. 2016. (online) http://parts.igem.org/Main˙Page (Accessed: 10 March 2016).
MUJTABA, I. M., AZIZ, N. & HUSSAIN, M. A. 2006. Neural network based modelling and control in batch reactor. *Chemical Engineering Research & Design,* 84, 635-644.
MULLER, S., HOFBAUER, J., ENDLER, L., FLAMM, C., WIDDER, S. & SCHUSTER, P. 2006. A generalized model of the repressilator. *Journal of Mathematical Biology,* 53, 905-937.
NAGY, Z. K., MAHN, B., FRANKE, R. & ALLGOWER, F. 2007. Evaluation study of an efficient output feedback nonlinear model predictive control for temperature tracking in an industrial batch reactor. *Control Engineering Practice,* 15, 839-850.
NIVENS, D. E., MCKNIGHT, T. E., MOSER, S. A., OSBOURN, S. J., SIMPSON, M. L. & SAYLER, G. S. 2004. Bioluminescent bioreporter integrated circuits: potentially small, rugged and inexpensive whole-cell biosensors for remote environmental monitoring. *Journal of Applied Microbiology,* 96, 33-46.
PALSSON, B. 2002. In silico biology through "omics". *Nature Biotechnology,* 20, 649-650.
PANKE, S., HELD, M. & WUBBOLTS, M. 2004. Trends and innovations in industrial biocatalysis for the production of fine chemicals. *Current Opinion in Biotechnology,* 15, 272-279.
PANKE, S., HELD, M., WUBBOLTS, M. G., WITHOLT, B. & SCHMID, A. 2002. Pilot-scale production of (S)-styrene oxide from styrene by recombinant Escherichia coli synthesizing styrene monooxygenase. *Biotechnology and Bioengineering,* 80, 33-41.
PANKE, S. & WUBBOLTS, M. 2005. Advances in biocatalytic synthesis of pharmaceutical intermediates. *Current Opinion in Chemical Biology,* 9, 188-194.
PANKE, S. & WUBBOLTS, M. G. 2002. Enzyme technology and bioprocess engineering. *Current Opinion in Biotechnology,* 13, 111-116.
PARDEE, K., GREEN, A. A., FERRANTE, T., CAMERON, D. E., DALEYKEYSER, A., YIN, P. & COLLINS, J. J. 2014. Paper-Based Synthetic Gene Networks. *Cell,* 159, 940-954.
PARISI, D. R., MARIANI, M. C. & LABORDE, M. A. 2003. Solving differential equations with unsupervised neural networks. *Chemical Engineering and Processing,* 42, 715-721.
PASEMANN, F. 1995. Characterization of periodic attractors in neural ring networks. *Neural Networks,* 8, 421-429.
PATWARDHAN, A. A., RAWLINGS, J. B. & EDGAR, T. F. 1990. Nonlinear model predictive control. *Chemical Engineering Communications,* 87, 123-141.

PEDERSEN, M. & PHILLIPS, A. 2009. Towards programming languages for genetic engineering of living cells. *Journal of the Royal Society Interface,* 6.

PEIRU, S., MENZELLA, H. G., RODRIGUEZ, E., CARNEY, J. & GRAMAJO, H. 2005. Production of the potent antibacterial polyketide erythromycin C in Escherichia coli. *Applied and Environmental Microbiology,* 71, 2539-2547.

POURBOGHRAT, F., PONGPAIROJ, H., LIU, Z. Q., FARID, F. & AAZHANG, B. 2003. Dynamic neural networks for modeling and control of nonlinear systems. *Intelligent Automation and Soft Computing,* 9, 61-70.

PRASAD, V. & BEQUETTE, B. W. 2003. Nonlinear system identification and model reduction using artificial neural networks. *Computers & Chemical Engineering,* 27, 1741-1754.

PURCELL, O., JAIN, B., KARR, J. R., COVERT, M. W. & LU, T. K. 2013. Towards a whole-cell modeling approach for synthetic biology. *Chaos,* 23.

QIN, S. J. & BADGWELL, T. A. 1997. *An overview of industrial predictive control technology*. In: Kantor, J. C., Garcia, C. E. and Carnahan, B. *Proceedings of 5th International Conference on chemical process control*. AICHE and CACHE. 155-171.

QIN, S. J. & BADGWELL, T. A. 2003. A survey of industrial model predictive control technology. *Control Engineering Practice,* 11, 733-764.

RASMUSSEN, K. H. & JORGENSEN, S. B. 1999. Parametric uncertainty modeling for robust control: a link to identification. *Computers & Chemical Engineering,* 23, 987-1003.

RASMUSSEN, S., CHEN, L. H., DEAMER, D., KRAKAUER, D. C., PACKARD, N. H., STADLER, P. F. & BEDAU, M. A. 2004. Transitions from nonliving to living matter. *Science,* 303, 963-965.

RO, D. K., PARADISE, E. M., OUELLET, M., FISHER, K. J., NEWMAN, K. L., NDUNGU, J. M., HO, K. A., EACHUS, R. A., HAM, T. S., KIRBY, J., CHANG, M. C. Y., WITHERS, S. T., SHIBA, Y., SARPONG, R. & KEASLING, J. D. 2006. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature,* 440, 940-943.

RODRIGO, G., CARRERA, J. & JARAMILLO, A. 2007. Genetdes: automatic design of transcriptional networks. *Bioinformatics,* 23, 1857-1858.

RODRIGO, G., CARRERA, J. & JARAMILLO, A. 2011. Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors. *Nucleic Acids Research,* 39.

ROGERS, J. M. & MCCULLOCH, A. D. 1994. A collocation-Galerkin finite-element model of cardiac action-potential propagation. *Ieee Transactions on Biomedical Engineering,* 41, 743-757.

ROMAN, R., NAGY, Z. K., CRISTEA, M. V. & AGACHI, S. P. 2009. Dynamic modelling and nonlinear model predictive control of a Fluid Catalytic Cracking Unit. *Computers & Chemical Engineering,* 33, 605-617.

ROSSI, F., MANENTI, F., MUJTABA, I. M. & BOZZANO, G. 2014. A Novel Real-Time Methodology for the Simultaneous Dynamic Optimization and Optimal Control of Batch Processes. *24th European Symposium on Computer Aided Process Engineering, Part A and B,* 33, 745-750.

RUBEN, A. J. & LANDWEBER, L. F. 2000. The past, present and future of molecular computing. *Nature Reviews Molecular Cell Biology,* 1, 69-72.

SAEIDI, N., WONG, C. K., LO, T.-M., HUNG XUAN, N., LING, H., LEONG, S. S. J., POH, C. L. & CHANG, M. W. 2011. Engineering microbes to sense and eradicate Pseudomonas aeruginosa, a human pathogen. *Molecular Systems Biology,* 7.

SAHINIDIS, N. V. & GROSSMANN, I. E. 1991. Convergence properties of generalized benders decomposition. *Computers & Chemical Engineering,* 15, 481-491.

SAUCERMAN, J. J. & MCCULLOCH, A. D. 2004. Mechanistic systems models of cell signaling networks: a case study of myocyte adrenergic regulation. *Progress in Biophysics & Molecular Biology,* 85, 261-278.

SCHMIDT, K. A., HENKEL, C. V., ROZENBERG, G. & SPAINK, H. P. 2004. DNA computing using single-molecule hybridization detection. *Nucleic Acids Research,* 32, 4962-4968.

SCHMITZ, M. H. A., HELD, M., JANSSENS, V., HUTCHINS, J. R. A., HUDECZ, O., IVANOVA, E., GORIS, J., TRINKLE-MULCAHY, L., LAMOND, A. I., POSER, I., HYMAN, A. A., MECHTLER, K., PETERS, J. M. & GERLICH, D. W. 2010. Live-cell imaging RNAi screen identifies PP2A-B55 alpha and importin-beta 1 as key mitotic exit regulators in human cells. *Nature Cell Biology,* 12, 886-893.

SCHNEEBELI, R. & EGLI, T. 2013. A Defined, Glucose-Limited Mineral Medium for the Cultivation of Listeria spp. (vol 79, pg 2503, 2013). *Applied and Environmental Microbiology,* 79, 6520-6520.

SEIDER, K., HEYKEN, A., LUTTICH, A., MIRAMON, P. & HUBE, B. 2010. Interaction of pathogenic yeasts with phagocytes: survival, persistence and escape. *Current Opinion in Microbiology,* 13, 392-400.

SHARIF, S. S., TAYLOR, J. H. & HILL, E. F. 2001. Dynamic online energy loss minimisation. *Iee Proceedings-Generation Transmission and Distribution,* 148, 172-176.

SHIEH, J. J., PAN, C. T. & CUEY, Z. J. 1997. Modelling and design of a reversible three-phase switching mode rectifier. *Iee Proceedings-Electric Power Applications,* 144, 389-396.

SHIRAKAWA, M., TAKEMORI, T. & OHTSUBO, J. 1995. OPTICAL LATCHES BASED ON A SELECTOR LOGIC. *Optics Communications,* 119, 505-512.

SHRESTHA, G. B. & FONSEKA, P. A. J. 2004. Congestion-driven transmission expansion in competitive power markets. *Ieee Transactions on Power Systems,* 19, 1658-1665.

SHRESTHA, G. B. & FONSEKA, P. A. J. 2006. Flexible transmission and network reinforcements planning considering congestion alleviation. *Iee Proceedings-Generation Transmission and Distribution,* 153, 591-598.

SIDDHAYE, S., CAMARDA, K., SOUTHARD, M. & TOPP, E. 2004. Pharmaceutical product design using combinatorial optimization. *Computers & Chemical Engineering,* 28, 425-434.

SIMEONIDIS, E., PINTO, J. M., LIENQUEO, M. E., TSOKA, S. & PAPAGEORGIOU, L. G. 2005. MINLP models for the synthesis of optimal peptide tags and downstream protein processing. *Biotechnology Progress,* 21, 875-884.

SIMON, L. L., NAGY, Z. K. & HUNGERBUHLER, K. 2009. Model based control of a liquid swelling constrained batch reactor subject to recipe uncertainties. *Chemical Engineering Journal,* 153, 151-158.

SISTU, P. B. & BEQUETTE, B. W. 1995. MODEL-PREDICTIVE CONTROL OF PROCESSES WITH INPUT MULTIPLICITIES. *Chemical Engineering Science,* 50, 921-936.

SLEPCHENKO, B. M., SCHAFF, J. C., MACARA, I. & LOEW, L. M. 2003. Quantitative cell biology with the virtual cell. *Trends in Cell Biology,* 13, 570-576.

SMITH, H. 1987. Monotone semiflows generated by functional differential equations. *Journal of Differential Equations,* 66, 420-442.

SMOLEN, P., BAXTER, D. A. & BYRNE, J. H. 1998. Frequency selectivity, multistability, and oscillations emerge from models of genetic regulatory systems. *American Journal of Physiology-Cell Physiology,* 274, C531-C542.

SORENSEN, H. P. & MORTENSEN, K. K. 2005. Advanced genetic strategies for recombinant protein expression in Escherichia coli. *Journal of Biotechnology,* 115, 113-128.

SPRINZAK, D. & ELOWITZ, M. B. 2005. Reconstruction of genetic circuits. *Nature,* 438, 443-448.

STEUER, R., KURTHS, J., FIEHN, O. & WECKWERTH, W. 2003. Observing and interpreting correlations in metabolomic networks. *Bioinformatics,* 19, 1019-1026.

STILIANAKIS, N. I., PERELSON, A. S. & HAYDEN, F. G. 1998. Emergence of drug resistance during an influenza epidemic: Insights from a mathematical model. *Journal of Infectious Diseases,* 177, 863-873.

STRAATHOF, A. J. J., PANKE, S. & SCHMID, A. 2002. The production of fine chemicals by biotransformations. *Current Opinion in Biotechnology,* 13, 548-556.

STRICKER, J., COOKSON, S., BENNETT, M. R., MATHER, W. H., TSIMRING, L. S. & HASTY, J. 2008. A fast, robust and tunable synthetic gene oscillator. *Nature,* 456, 516-U39.

SZYBALSKI, W. & SKALKA, A. 1978. Nobel-prizes and restriction enzymes. *Gene,* 4, 181-182.

TAMIMI, J. & LI, P. 2010. A combined approach to nonlinear model predictive control of fast systems. *Journal of Process Control,* 20, 1092-1102.

TEUSINK, B., PASSARGE, J., REIJENGA, C. A., ESGALHADO, E., VAN DER WEIJDEN, C. C., SCHEPPER, M., WALSH, M. C., BAKKER, B. M., VAN DAM, K., WESTERHOFF, H. V. & SNOEP, J. L. 2000. Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *European Journal of Biochemistry,* 267, 5313-5329.

TIAN, H., BHARADWAJ, S., LIU, Y., MA, H., MA, P. X., ATALA, A. & ZHANG, Y. 2010. Myogenic differentiation of human bone marrow mesenchymal stem cells on a 3D nano fibrous scaffold for bladder tissue engineering. *Biomaterials,* 31, 870-877.

TIGGES, M., DENERVAUD, N., GREBER, D., STELLING, J. & FUSSENEGGER, M. 2010. A synthetic low-frequency mammalian oscillator. *Nucleic Acids Research,* 38, 2702-2711.

TIGGES, M., MARQUEZ-LAGO, T. T., STELLING, J. & FUSSENEGGER, M. 2009. A tunable synthetic mammalian oscillator. *Nature,* 457, 309-312.

TSAI, T. Y. C., CHOI, Y. S., MA, W. Z., POMERENING, J. R., TANG, C. & FERRELL, J. E. 2008. Robust, tunable biological oscillations from interlinked positive and negative feedback loops. *Science,* 321, 126-129.

TUTTLE, L. M., SALIS, H., TOMSHINE, J. & KAZNESSIS, Y. N. 2005. Model-driven designs of an oscillating gene network. *Biophysical Journal,* 89, 3873-3883.

TYSON, J. J., ALBERT, R., GOLDBETER, A., RUOFF, P. & SIBLE, J. 2008. Biological switches and clocks. *Journal of the Royal Society Interface,* 5, S1-S8.

ULLNER, E., ZAIKIN, A., VOLKOV, E. I. & GARCIA-OJALVO, J. 2007. Multistability and clustering in a population of synthetic genetic oscillators via phase-repulsive cell-to-cell communication. *Physical Review Letters,* 99.

VAN EUNEN, K., DOOL, P., CANELAS, A. B., KIEWIET, J., BOUWMAN, J., VAN GULIK, W. M., WESTERHOFF, H. V. & BAKKER, B. M. 2010. Time-dependent regulation of yeast glycolysis upon nitrogen starvation depends on cell history. *Iet Systems Biology,* 4, 157-168.

VANLIER, J., WU, F., QI, F., VINNAKOTA, K. C., HAN, Y., DASH, R. K., YANG, F. & BEARD, D. A. 2009. BISEN: Biochemical Simulation Environment. *Bioinformatics,* 25, 836-837.

VENTER, J. C., LEVY, S., STOCKWELL, T., REMINGTON, K. & HALPERN, A. 2003. Massive parallelism, randomness and genomic advances. *Nature Genetics,* 33, 219-227.

VILAR, J. M. G., KUEH, H. Y., BARKAI, N. & LEIBLER, S. 2002. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Sciences of the United States of America,* 99, 5988-5992.

VILLADSEN, J. & SORENSEN, J. P. 1969. SOLUTION OF PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS BY A DOUBLE COLLOCATION METHOD. *Chemical Engineering Science,* 24, 1337-+.

VISWANATHAN, J. & GROSSMANN, I. E. 1990. A combined penalty-function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering,* 14, 769-782.

WALL, M. E., DUNLOP, M. J. & HLAVACEK, W. S. 2005. Multiple functions of a feed-forward-loop gene circuit. *Journal of Molecular Biology,* 349, 501-514.

WANG, R. Q., JING, Z. J. & CHEN, L. N. 2005. Modelling periodic oscillation in gene regulatory networks by cyclic feedback systems. *Bulletin of Mathematical Biology,* 67, 339-367.

WARREN, L., MANOS, P. D., AHFELDT, T., LOH, Y.-H., LI, H., LAU, F., EBINA, W., MANDAL, P. K., SMITH, Z. D., MEISSNER, A., DALEY, G. Q., BRACK, A. S., COLLINS, J. J., COWAN, C., SCHLAEGER, T. M. & ROSSI, D. J. 2010. Highly Efficient Reprogramming to Pluripotency and Directed Differentiation of Human Cells with Synthetic Modified mRNA. *Cell Stem Cell,* 7, 618-630.

WESTERLUND, T. & PORN, R. 2002. Solving Pseudo-Convex Mixed Integer Optimization Problems by Cutting Plane Techniques. *Optimization and Engineering,* 3, 253-280.

WOJCIECHOWSKI, M. 2012. Solving Differential Equations by Means of Feed-Forward Artificial Neural Networks. *Artificial Intelligence and Soft Computing, Pt I,* 7267, 187-195.

WU, C. H., LEE, H. C. & CHEN, B. S. 2011. Robust synthetic gene network design via library-based search method. *Bioinformatics,* 27, 2700-2706.

WU, F., YANG, F., VINNAKOTA, K. C. & BEARD, D. A. 2007. Computer modeling of mitochondrial tricarboxylic acid cycle, oxidative phosphorylation, metabolite transport, and electrophysiology. *Journal of Biological Chemistry,* 282, 24525-24537.

XIANG, S. L., FRUEHAUF, J. & LI, C. J. 2006. Short hairpin RNA-expressing bacteria elicit RNA interference in mammals. *Nature Biotechnology,* 24, 697-702.

XU, Y., ALLEN, L. J. S. & PERELSON, A. S. 2007. Stochastic model of an influenza epidemic with drug resistance. *Journal of Theoretical Biology,* 248, 179-193.

YAMATO, S., NAKAJIMA, M., WAKABAYASHI, H. & SHIMADA, K. 1992. Specific detection of Acetyl-Coenzyme-A by reversed-phase ion-pair high-performance liquid-chromatography with an immobilized enzyme reactor. *Journal of Chromatography,* 590, 241-245.

YEOMANS, H. & GROSSMANN, I. E. 2000. Optimal design of complex distillation columns using rigorous tray-by-tray disjunctive programming models. *Industrial & Engineering Chemistry Research,* 39, 4326-4335.

YODA, M., USHIKUBO, T., INOUE, W. & SASAI, M. 2007. Roles of noise in single and coupled multiple genetic oscillators. *Journal of Chemical Physics,* 126.

YOKOBAYASHI, Y., WEISS, R. & ARNOLD, F. H. 2002. Directed evolution of a genetic circuit. *Proceedings of the National Academy of Sciences of the United States of America,* 99, 16587-16591.

YONG, Y.-C., YU, Y.-Y., LI, C.-M., ZHONG, J.-J. & SONG, H. 2011. Bioelectricity enhancement via overexpression of quorum sensing system in Pseudomonas aeruginosa-inoculated microbial fuel cells. *Biosensors & Bioelectronics,* 30, 87-92.

ZHOU, T. S., ZHANG, J. J., YUAN, Z. J. & CHEN, L. N. 2008. Synchronization of genetic oscillators. *Chaos,* 18.

ZIOGOU, C., PAPADOPOULOU, S., GEORGIADIS, M. C. & VOUTETAKIS, S. 2013. On-line nonlinear model predictive control of a PEM fuel cell system. *Journal of Process Control,* 23, 483-492.