Qualitative and semi quantitative systems approaches to complex systems modelling – Intuitive and flexible models of biological systems

---

A thesis

submitted in partial fulfilment

of the requirements for the Degree of

Doctor of Philosophy

at

Lincoln University

by

Mohammad Abdallah Alsharaiah

Lincoln University
2018

Abstract of a thesis submitted in partial fulfilment of the

requirements for the Degree of Doctor of Philosophy

Qualitative and semi quantitative systems approaches to complex systems modelling – Intuitive and flexible models of biological systems

by

Mohammad Abdallah Alsharaiah

Abstract

Biological systems such as cell cycle are complex systems consisting of an enormous number of elements. These elements interact in ways that produce nonlinear and complex systems behaviour such as oscillations. A number of modelling approaches have been used to explain these kinds of systems; they are classified into four classes (continuous, discrete, stochastic and hybrid). Ordinary Differential Equations (ODEs) are used to mimic the continuous dynamic behaviour of system components, while discrete models can simulate the biological elements as straightforward binary variables providing a qualitative view of system behaviour. Stochastic models are used to model the effect of noise in biological systems. Combined methods together introduce hybrid models to cover the limitations of individual models and take advantage of their strengths.

This study introduces a series of advanced models with increasing resolution from discrete to continuous in a systematic way to model the mammalian cell cycle system. Each model includes the essential controllers of mammalian cell cycle. Specifically, they reveal cell regulators that control cell cycle transitions from one phase to another in cell division. In the first model, this work introduces a new biological network based on a qualitative approach-Boolean network. A new Boolean model with 13 proteins can capture the essential aspects of cell cycle phases and it can simplify cell cycle control system. The developed new, simple and intuitive Boolean model can mimic the fluctuation of Cyclins during cell cycle. Furthermore, it can show cell cycle phases based on the changes in Cyclins since each Cyclin leads the transition of cell cycle phases. Also, some important proteins that are missing in most

of the current models, such as SCF ubiquitin ligase and c-Myc, have been added to our model. This study provides a deep understanding of the mechanism of mammalian cell cycle regulation, especially when growth factors are present, and it can reveal the cell cycle process in terms of flow cytometry of Cyclin proteins. In addition to the aforementioned advancements related to the developed Boolean model, the model has captured the periodic sequence of activity of cell cycle regulatory proteins over repeated cell cycles. The existing discreet models failed to represent the periodic behaviour of cell cycle phase transitions and the correct activities for system species over repeated cell cycles.

In the second model, another computing model based on degree of truth rather than the usual Boolean model with true or false (1 or 0) values is implemented. This second developed model is a fuzzy logic model. It involves the use of artificial intelligence to model a fuzzy cell cycle control system. Fuzzy logic model is a rule-based model that depends on the practical knowledge and heuristic design of complex systems. Specifically, this work utilizes fuzzy inference system features in the development process. Indeed, it is expected to provide approximate continuous dynamics for the discrete events using limited available data. When applied to the cell cycle system, the model reveals the intermediate states of concentrations of proteins during each cell cycle event. In addition, applying fuzzy logic provides more accurate representation of the cell cycle system than previous Boolean approaches. It can follow/explain the flow cytometry measurement levels of protein concentrations such as Cyclins. Also, the fuzzy model is expected to simplify and realistically represent the system and address the existing shortcomings of both discrete and continuous representations, specifically, discrete models' restriction to 0 and 1 values for the states of proteins and the scarcity of available kinetic parameters in continuous models such as ODEs.

Furthermore, for more accurate results and to automate the development of the fuzzy controller, the core of the fuzzy inference system has been successfully optimized using artificial intelligent approach Particle Swarm Optimization (PSO).

Key words: Mammalian cell, Cell signalling networks, Synchronous and Asynchronous Boolean Networks, Attractors, Fuzzy Systems, Particle swarm optimization (PSO).

Manuscripts under Preparation

Alsharaiah, M., Samarasinghe, S. and Kulasiri, Don (2018). Qualitative modelling Approach to Simplify mammalian cell cycle regulatory system. PNAS.

Alsharaiah, M., Samarasinghe, S. and Kulasiri, Don (2018). Semi Quantitative Cell Cycle Controller Model – An Intuitive Fuzzy Logic Model. Journal of Theoretical Biology

Alsharaiah, M., Samarasinghe, S. and Kulasiri, Don (2018). Tuning of a Fuzzy Logic Controller for mammalian cell cycle using Particle Swarm Optimization. Journal of Theoretical Biology.

Acknowledgements

This research was carried out during March 2015-June 2018 period at the Complex Systems, Big Data and Informatics Initiative (CSBII) and the Centre for Advanced Computational Solutions (C-fACS), Lincoln University, New Zealand. After an intensive study period, today is the day: writing this annotation of thanks is the final touch on my thesis. It has been a period of intense learning for me, not only in the scientific field, but also on a personal level. Writing this thesis has had a big influence on me. The biggest pleasure of this achievement is to recall the past and remember who are behind the achievement of this difficult but fulfilling journey.

First, I would like to express my deep and sincere appreciation to my supervisor, Prof. Sandhya Samarasinghe, Professor of Complex Systems Modelling, at the Complex Systems, Big Data and Informatics Initiative (CSBII), Lincoln University, New Zealand. Her enormous knowledge and logical thinking have been of great value to me. Her sympathetic encouragement, inspiration, supervision, detailed review, deep questioning, positive critique, nonstop positive feedback and brilliant advice during the research have provided me with a solid and creative basis and motivation for the present thesis. Moreover, her friendship, kindness and care taken to keep us in a stress free environment is highly appreciated.

In addition, I am sincerely thankful to my co-supervisor, for his comprehensive and productive comments, and for his enormous support throughout this work - Prof. Don Kulasiri, Professor of Computational Modelling and Systems Biology, Department of Wine Food and Molecular Bioscience, Faculty of Agriculture and Life Sciences, Lincoln University for sharing his vast knowledge, awareness and proficiency through regular weekly face to face lab meetings. These labs are aimed to refine our profession with more practical and academic understandings that are beyond the research objectives.

Special thanks are due to Lincoln University, Faculty of Environment, Society and Design and all campus members. If not them and their support in many ways to keep me satisfied, it would have been hard to complete this thesis. Furthermore, my special gratefulness is due to my friends. They gave me moral support to tolerate all difficult moments I faced.

Nobody has been more important to me in the pursuit of this work than the members of my family. I owe my deep gratitude to my family for their enormous support and patience. I would like to thank my parents, my brothers and my sisters, whose love and guidance are with me in whatever I pursue.

Last but not least, to my caring, loving, and supportive wife, Fatima: I offer my deepest gratefulness. Your nonstop encouragement when the times got rough are much appreciated and admirably noted. It was an inordinate comfort and assistance to know that you were willing to take care of and manage our kids and household activities while I was away from you during my PhD study. My wife and my kids are the ones who sacrificed the most while I was away from them for the entirety of my PhD study; especially, when my mother in law passed away. They were so patient, bearing an endless love in their heart, regardless of the ups and downs in our lives during my PhD period. Honestly, I am grateful to my wife for making my studies one of the top priorities in our lives with all the support and care provided. Special thanks go to her for making me so resilient to tolerate any challenging situation in my life. My heartfelt thanks to all.

Thank you very much, everyone!

Table of Contents

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Overview

Biological systems such as mammalian systems are characterized as complex systems consisting of a large number of cells. In this system, a cell is considered as the smallest basic unit of all living organisms. Also, this unit can control its structure with independent behaviors and it can reproduce itself. Simply, cells together create the composite of life's building blocks Hay (1991). Each cell can reproduce itself by an ordered sequence of events called cell cycle. Cell cycle is considered a complex system because it contains a large number of simple and different elements that interact with each other to produce complex behavior in the system (Kitano, 2002). Further, elements have persistent interactions during cell cycle to produce two cells from a mother cell; this process is called cell proliferation.

Biological systems are organized as complex networks that are different in type depending on the component elements, for example, Gene Regulatory Networks (GRN), protein to protein interactions networks and metabolic networks. Similarly, cell cycle has a large number of proteins and genes organized in Gene Regulatory Networks and protein – protein interaction networks. GRN networks consist of genes (DNA (Deoxyribonucleic acid) segments) inside the cell nucleus that produce copy of DNA - Ribonucleic Acid (RNA) that gets converted into mRNA and subsequently to protein products that interact with other substances in the cell. This process is controlled by rates that determine how fast a selected set of genes in the network is transcribed into mRNA and how various proteins rapidly interact activating or inhibiting each other (Vijesh, Chakrabarti, & Sreekumar, 2013).

The underlying biological process of molecular mechanisms can be systematically understood by using GRN and associated protein networks (De Jong, 2002; Swain, Mandel, & Dubitzky, 2010). Genes and protein interactions are organized in the cell cycle system in

a particular way to perform the related functions. When these elements work together they provide a variety of complex dynamic behaviors called emergent properties, such as stable functioning to achieve cell division, but when they work in fragmentary ways, it can produce undesirable or chaotic behavior that could lead to some malfunction or diseases such as cancer. The large number of interacting entities makes it difficult to understand the interactions between them, which makes the system quite complex to understand and model. In addition, these interactions differ in speed. However, modelling approaches are useful for understanding complex systems including their element and system dynamic behavior and specific system events such as stages of cell cycle. Many modelling approaches have been used to address complex system problems. However, many issues need more investigation.

1.2 Definition of the Problem

Most studies on biological systems have been made based on various assumptions in their theoretical aspects and have focused on gene regulatory networks, metabolic networks and protein-protein signaling networks (Alm & Arkin, 2003). Further, criteria for selecting the best model for a case study are based on the availability of known information related to the biological system, including reaction rates and the type of the data collected to describe the system activities. However, most modelling approaches face problems with the availability of data, especially in cell cycle signaling network.

In modelling signaling networks, various mathematical, statistical and computational approaches can be used to simulate the biological system processes, events, pathways and networks. These approaches can be used to provide a useful framework that provides researchers with more information to understand the gaps, issues, and insights into the biological system. Such frameworks can explain the system component interactions and generate biological insights into the system by testing hypotheses related to system properties, functions and structures (David Gilbert et al., 2006).

Understanding and examining the system elements with their internal relations alongside system behavior and reactions raises the need for the development of new models and approaches that explain the individual and collective behavior of genes, proteins, enzymes,

and metabolites in biological systems (Kitano, 2002). Researchers and modelers can use mathematical and computational approaches to advance research in biology. They depend on available information to develop models to address one or more of these issues. For example, they use gene regulatory networks and protein – protein networks to analyze cell cycle events and activities in relation to gene expression (Kohn, 1998).

Modelling approaches can be classified into several classes (Csikász-Nagy, 2009), Firstly, Continuous Models (ODE) are used to express continuous changes in protein concentrations in biological systems such as Bela Novak and John J Tyson (2004). This kind of models provide realistic results but have several limitations in handling complexity associated with large networks due to the requirement of a large number of kinetic parameters (rate constants) which are limited in biology. These rates are not yet completely available. Simpler modelling approach may sufficiently represent some aspects of the system. For this reasons, modelers have started looking for and developing alternative solutions. These solutions should be simpler and easier. Hence, since the available biological knowledge is either incomplete or imprecise, this gives the modeler flexibility to introduce new models that can still adequately describe system activities. Therefore, they have introduced the second class of models - logical or discrete models such as Boolean networks - that describe a system behaviour in realistic qualitative form (Glass & Kauffman, 1973; Thomas, 1973).

Boolean networks are very useful and effective in comparison with other methods such as differential equations that need kinetic data that is not available or limited. The Boolean network is used to model complex biological systems such as the cell cycle system based on available biological knowledge such as Fauré, Naldi, Chaouiya, and Thieffry (2006)model. Boolean models are more useful to study the qualitative aspects of the biological system. For instance, when detailed information is not available on the system, Boolean networks can explain in a simpler way how protein interactions happen and which protein is active or inactive (Garg, 2009; Tyson, Novak, Chen, & Val, 1995). They are useful for understanding a system when full details are not available. Further, some aspects or parts are best be modeled qualitatively or simply due to their nature, relevance or importance to the whole system. However, Boolean models lack the ability to quantify system behavior. Additionally, Boolean model outcome depends on the model update

scheme and producing realistic temporal system dynamics is a current challenge in Boolean models. Further, representing slow and fast reactions together in a Boolean framework is also a challenge.

The third class of models represents biological systems along with noise; these are called stochastic models, an example of which is Kapuy, He, Uhlmann, and Novák (2009) model; they simulated mitotic exit of mammalian cells. The fourth class is hybrid models. In this model, a combination of different techniques is used. Recently, Singhania, Sramkoski, Jacobberger, Tyson, and Beard (2011) have developed a new cell cycle model. This model relies on a mix of continuous ODE and Boolean networks in one model to dynamically study the system by relying on Cyclin behaviour during cell cycle. Although this model is a recent and a benchmark model, it lacks some important elements in cell cycle system such as Cyclin D, MYC, SCF and TFB,.., etc. Also, the Boolean part does not contain Boolean functions; they assumed a predetermined set of states to mimic the actual states.

Most current mathematical models of cell cycle have been established for simple organisms, such as yeast. However, mammalian cell cycle system has been less studied and modeled due to its higher complexity. Furthermore, no single modelling technique can introduce all aspects of biological events. Each method has advantages and disadvantages that create real limitations and gaps in cell cycle modelling. Hence, efforts continue to enhance models to investigate the complex cell cycle system (Vijesh et al., 2013).

Currently, most of the computational modelling approaches face problems with the availability of data. For instance, modelling gene and protein expressions need knowledge about the concentrations and rates of production which at the moment is not completely available (Davidich & Bornholdt, 2008).For this reason, the exact values for quantities of species are not always the main concern in a biological setting (De Jong, 2002). However, the general trend of behavior of molecular species is emphasized. Furthermore, current benchmark mammalian cell cycle models such as Fauré et al. (2006) Boolean model and Singhania et al. (2011) hybrid model have several limitations. For instance, these models were built depending on the old available biological knowledge and they may not correctly represent recently discovered knowledge. For example, these models miss newly discovered important proteins in cell cycle system such as SCF ubiquitin ligase, c-Myc, TFB,

etc.. These biological elements can provide a more complete view of the underlying mechanism of the cell cycle control system. As a consequence, this kind of models needs further development since the biological knowledge is continiously updated.

Moreover, there is a need for modelling approaches that can provide a greater understanding of complex systems and are useful to study and analize the dynamic behavior of biological systems such as cell cycle. Modelers have started looking for and developing alternative solutions and these solutions should be simpler, easier and more effective than existing models.

1.3 Research Aim

Currently, modelling methods and techniques have become popular for modelling biological systems because they can provide a deep understanding and insight into them. For instance, they can help understand the biological processes and provide novel or additional information and knowledge related to them. Also, they can be used for prediction and diagnosis of diseases and treatment of diseases such as cancer. Models attempt to provide an accurate dynamic description of phenomena; they attempt to either explain, expose or predict biological phenomena. There are many challenges to this process at the moment in systems biology. This study addresses some of these challenges in modelling complex systems of mammalian cell cycle.

This study provides a new method to model cell cycle controller systems through advancement in modelling approaches. The new model can enhance the knowledge of cell cycle signaling network and generate more insights into **how network components are connected to each other and why, and for what purpose, they are placed in this form**.

The first specific objective of the thesis is to study the structural and dynamic analysis of mammalian cell cycle controller system. As demonstrated before, most current models simulate and study simple organism such as *yeast* model, but few models that simulate and study mammalian cell cycle control network are available in the literature; accordingly, there is a need to develop a mammalian cell cycle control model**.** Since biological

knowledge is incomplete and not updated fast enough, there are limited computational approaches in the literature for modelling this protein network. However, there is a necessity for a comprehensive analysis to better understand this mechanism. In addition, research needs to generate more insights into how, where and what is the essential information in the cell cycle signaling network. Therefore, the primary aim of this thesis is to find proper answers to the question of **how we can design a single system with the scarcely available information to represent convincing dynamic behaviour of cell cycle control system?**

In this research, the main interest is to develop specific protein - protein interaction networks that are useful for modelling biological systems such as cell cycle system. Specifically, the target is to reveal cell regulators that control cell transitions from one phase to the other during cell division process. Specifying an appropriate mathematical model demands a solid comprehension of the behaviour of signaling system component interactions. Therefore, the **second specific objective of the thesis is to propose a new biological network (protein - protein interaction network) based on a qualitative approach-Boolean network to model cell cycle controller.** This Boolean network can mimic the controller species activity over cell cycle phases of mammalian cells. The previous attempt at elucidating this network was described in 2006 by (Fauré et al., 2006)using a synchronous and asynchronous Boolean model; this model was developed based on ODE model (B. Novak & J. J. Tyson, 2004) with old biological knowledge. A few years later, another mammalian cell cycle model was developed by Singhania et al. (2011). These two models are referred to as bench marks in biological modelling studies.

Therefore, in this thesis, the process to improve the mammalian cell cycle models is based on these two models. The new model appends the recently discovered biological knowledge including new important proteins which are missing in most of the current models. For instance, this work studied, investigated and enhanced the Fauré et al. (2006) Boolean model – specifically, it added a set of new biological elements to the system such as SCF and MYC. Further, this study introduced updating the Boolean rules based on the available and recent biological knowledge.

Additionally, the developed Boolean model rebuilds and extends the recent hybrid model in Singhania et al. (2011). Although this is a recent and recommended model for biology researchers, it faces some of limitations. Therefore, our work overcomes this limitation by adding some important biological elements like CycD, P27, and RB. Also, this research has rebuilt the model in proper Boolean form to explain abundance of system elements within cell cycle phases. Singhania model did not provide any Boolean rules. This enhancement increased the strength of the proposed new model by overcoming the existing limitations in the previous models.

The third objective of this study is to understand the dynamics of the network and compare the results with other results from the existing networks such as Faure 2006 model. The enhanced model should recapture the generic behaviour of the system with additional advancements. The new enhancement which is accomplished by adding novel information would not contradict the available results or biological knowledge, but it shall fill the missing part of the system and provide a fuller view of system interactions within cell cycle phases. For instance, as mentioned before, Sanghania model missed some important species such as CYCD, P27, and RB, while Faure 2006 model missed MYC and SCF species in their model. The aforementioned models have been built on some of assumptions since in that time some biological facts not discovered yet which recently should be modified in a proper form with up-to-date knowledge. While fulfilling this objective, current Boolean models were systematically enhanced. This enhancement can be used to answer the following questions:

I. **Can the Boolean model explain the transition of cell cycle phases and system dynamics?**
II. **Can the Boolean model reproduce the temporal behaviour of the observed species such as Cyclins during cell cycle?**
III. **Can the model provide a visualization of the activity of system components within cell cycle phases (system attractors), including the long term behaviour in steady state dynamics?**
IV. **What are the changes made by the new enhancement?**
V. **What are the essential components of the network needed to accomplish cell cycle?**
VI. **How robust is the cell cycle signalling network?**

Providing satisfactory answers to the above questions with novel information makes a significant contribution to understanding cell cycle.

The fourth objective of the study is to introduce better logic based models such as fuzzy logic to model cell cycle controller system; herein, this objective involves fuzzy representation of cell cycle and implementation of a fuzzy inference system to mimic cell cycle controller behavior. Logic models mainly rely on knowledge based theory. This simplifies the representation and exploration of system components, interactions, behavior and events. Also, logical models have the ability to generate accurate qualitative results for system dynamics in discrete form (Thomas, 1973). They can provide a simplified view of systems activities that is useful for gaining insight. As a consequent, another novel model is proposed based on knowledge based theory. This model can intuitively represent the detailed dynamic behavior of cell regulators that control cell transitions between phases during cell division process. Specifically, biological knowledge presented in Boolean form is converted to a form of human reasoning in this model. This conversion comprises natural intelligence and artificial intelligence features of fuzzy logic. Fuzzy logic has the ability to provide methods of computing with words in a coherent manner. Moreover, Fuzzy logic is a suitable method to represent biological systems because of the nature of these systems – for example, gene regulation is inherently fuzzy (not exact); however, most cell cycle studies depend on crisp models. Another prompting factor to use fuzzy methods is that biological data are also mostly imprecise. The imprecision of data is due to the nature of biological experiments themselves. It makes fuzzy logic very suitable to mimic many biological problems such as cell cycle controller system.

Additionally, fuzzy models provide approximate continuous dynamics for discrete as well as continuous events using limited available data and incomplete information expressed in linguistic terms. Fuzzy logic allows variables to be continuous and more finely valued than other discrete models such as Boolean networks. It can enable a model to consider all the intermediate levels of species concentration levels without leaving the qualitative modelling framework. Furthermore, it can state which element is more active than others. Also, fuzzy logic considers element states in a range of linguistic terms rather than a set of two numbers. In a Boolean model, a Boolean element is either a member of a limited set or not, such as active or inactive.

Therefore, this objective employs computational intelligence methods to enhance the bio-realism of the cell cycle controller models. This work utilizes fuzzy inference systems to mimic species activity based on protein concentration measurements. Essentially, Fuzzy inference system has been used to represent and process vague information in many real-life problems. This influential reason was behind the current work's proposal to use fuzzy inference system to mimic cell cycle controller model.

The fifth objective of the study is develop an approach to automate the development of the fuzzy inference system developed in Objective 4. Therefore, this work optimized the fuzzy inference system by **employing an intelligent systems approach - Particle Swarm Optimization (PSO)**. PSO is a computational optimization method inspired by social behavior of bird flocking or fish schooling (Akkar, A-Amir, & Saleh, 2015; Clerc & Kennedy, 2002; Kenndy & Eberhart, 1995; Omizegba & Adebayo, 2009). This thesis used PSO to optimize and automate the development of the fuzzy inference system. It provides robust improvements of the inference system leading to optimum performance of the system.

These developed models such as Boolean model and fuzzy models can give a deeper understanding of this complex system and its dynamic behavior. They are more suitable than previously mentioned individual methods for modelling the dynamic behavior of the processes in cell cycle. They help overcome the shortcoming and limitations of these other models such as discrete model in Fauré et al. (2006), ODE model in Iwamoto, Hamada, Eguchi, and Okamoto (2011) and hybrid model in Singhania et al. (2011).This work provides a fuller and more realistic explanation of cell cycle functions by providing qualitative and semi quantitative descriptions.

1.4 Research Contributions

This thesis presents qualitative and semi quantitative models such as Boolean network and fuzzy logic system, respectively, based on knowledge based theory. It provides a significant methodological enhancement to the current modelling approaches of cell cycle control system. The developed models help explore the complex dynamic behaviour with scarce

biological data and parameters. In addition, these models make the following contributions:

1. Construction of novel protein - protein interaction networks that are valuable for modelling biological systems such as cell cycle system. Precisely, they better and more completely model cell regulators that control cell transitions between phases during cell division process.

2. Development of a new Boolean network based on recent knowledge to model cell cycle control system. This includes enhancing the current bench mark models such as Fauré et al. (2006) model and Singhania et al. (2011) model. Further, this involves a comparative, robustness and resilience study of these models. This study is based on biological evidence and assumptions.

3. Utilisation of imprecise, vague and incomplete knowledge and data to create seamless and realistic knowledge and data by using fuzzy logic theory to mimic the biological interactions in cell cycle control system. The result was the development of a fuzzy cell cycle controller model.

4. Employment of Artificial Intelligence approach - Particle swarm optimization (PSO)- to optimise the fuzzy logic based inference system to improve the performance of the mammalian cell cycle complex system.

All aforementioned contributions have improved the representation, analysis and understanding of mammalian cell cycle control system as well as its species since cell cycle consists of several phases, and each phase contains several species that interact with each other in complex biological processes. We assess and analyse how the model helps explain cell cycle phenomena such as Cyclins activity to control cell cycle phase transitions and other cell cycle phenomena including system dynamics.

In the next section, we briefly demonstrate how the thesis is organised to link and present the steps taken to achieve the research objectives and the acquired results. It provides a summary of the models, analyses and outcomes.

1.5 Overview of Chapters

This thesis consists of several chapters that give the reader a full description of our research on modelling complex biological systems. It covers the cell cycle modelling system from both biological and computational aspects and explains all information related to it.

- Chapter 1 provides a brief introduction to the research problem such as issues and gaps in current cell cycle models, followed by the objectives of the study, research contributions and an overview of thesis chapters.
- Chapter 2 provides a background and available information on the biological system (cell cycle) and components, including biological cell types, most common cell cycle engine components and checkpoints. Furthermore, it provides a classification of the past models of cell cycle. It also shows the advantages and disadvantages of existing models.
- Chapter 3 explains the detailed method used in Boolean modelling approaches. Also, this chapter investigates the most popular models to be enhanced. Further, it develops an advanced Boolean model of cell cycle control system and explores the results of this modelling attempt and compares with the existing models.
- Chapter 4 shows and describes the proposed methods for developing a fuzzy logic model and how fuzzy logic can convert discrete models to semi quantitative models mathematically. In addition, the fuzzy model components and their results are discussed and compared with other models.
- Chapter 5 explains PSO concepts, method and clarifies how the optimised fuzzy inference system provides robustness improvements and more accurate results from the developed models.
- Finally, Chapter 6 provides a summary of the results obtained for each objective of the study. Moreover, conclusions of the research are represented in this chapter. Also, it presents the future research directions necessary to improve the models to even better understand the system dynamics of the cell cycle signaling network.

Chapter 2: Literature Review

This study aims at understanding the behaviour of complex biological systems, such as cell cycle signalling network regulators. To have more accurate and deep understating of these systems, and how their components interact with each other, we need to know the relevant information about the most important regulatory elements in the cell cycle system. This information can enhance our knowledge to identify and construct our model and introduce advancements or improvements to it.

This chapter, provides background information that gives a solid understanding of the mammalian cell cycle system, particularly in two aspects; the biological and the computational. Regarding the biological aspect, this chapter presents a biological review of the mammalian cell cycle system; including, cell cycle events or phases, components of cell cycle regulators, cell proliferation, checkpoints and, finally, DNA damage control during the cell cycle. Concerning the computational aspect, this chapter discusses modelling approaches that have previously been used for modelling the cell cycle process, highlighting their advantages and disadvantages in order to pave the way for the advanced model proposed in this research.

Basically, a good understanding of the mentioned topics can help the modeller to integrate or synthesise a comprehensive computational model that is useful for modelling biological systems. This will be undertaken in this study and to aid the process some of the outstanding issues related to the cell cycle are presented in Sections 2.1 and 2.2.

2.1 Biological Background

2.1.1 Mammalian Cell Classifications: General Review

The cell is the smallest unit of all living organisms. It can control its structure independently and can reproduce itself. Simply, cells, together constitute biological life, and are known as the “building blocks of life” (Hay, 1991). Cells are organised and classified into two types

depending on the existence of a nucleus. Cells with a nucleus are called eukaryotic cells and those without a nucleus are called prokaryotic cells.

2.1.1.1 The Prokaryotic Cell

This kind of cell is represented in the literature as the first form of life on our planet. As mentioned above, prokaryotic cells do not contain a nucleus and this makes them simpler and smaller than eukaryotic cells (Stein & Pardee, 2004). Prokaryotic cells are organised into two domains: bacteria and archaea (Koonin & Wolf, 2008). Prokaryotic cells can also be characterised by three architectural regions, as shown in Figure 2.1: First, the cell flagella and pili project from the cell surface. Second, a cell envelope is formed and this is used to enclose the cell - it is like a wall that covers the cell membrane and separates the cell components from the outside environment. Third, the DNA regions (cytoplasm), the ribosomes and various inclusions are formed. These regions are represented in Figure 2.1 (OpenStax-CNX, 2013).

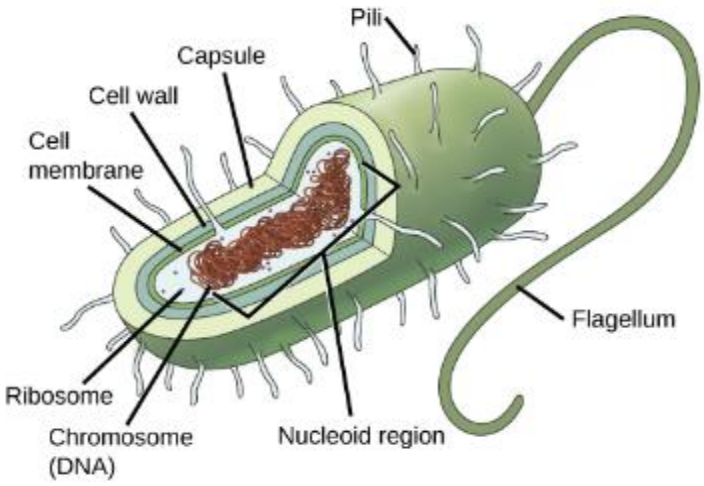


Figure 2.1: Diagram of a typical prokaryotic cell (OpenStax-CNX, 2013)

2.1.1.2 The Eukaryotic Cell

The second type of cell is called a eukaryotic cell (Bell & Dutta, 2002). The volume of this cell is a round thousand times larger than a prokaryote cell; there is actually a wide range of sizes for both prokaryotic and eukaryotic cells. It is also more complex in its structure and contents. Some of the differences are summarised in Table 2.1. The structure of this kind of cell is displayed in Figure 2.2 (OpenStax-CNX, 2013).

Table 2.1: Differences between prokaryotic and eukaryotic cells

Prokaryotic cells	Eukaryotic cells
Always unicellular organisms	**Often multicellular organisms**
Small cells – **0.1–5.0 µm diameter**	**Larger cells -** **According to one site: 10–100 µm diameter**
No nucleus or any membrane-bound organelles, such as mitochondria	**Always have nucleus and other membrane-bound organelles**
DNA is circular, without proteins	**DNA is linear and associated with proteins to form chromatin**
Ribosomes are small (70S)	**Ribosome is large (80S)**
No cytoskeleton	**Always has a cytoskeleton**
Cell division is by binary fission	**Cell division is by mitosis or meiosis**
Reproduction is always asexual	**Reproduction is asexual or sexual**

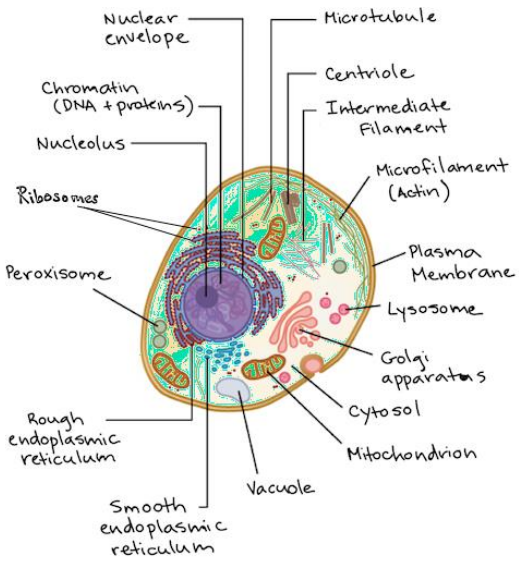


Figure 2.2: Structure of eukaryotic cells: a typical animal cell (OpenStax-CNX, 2013)

2.1.2 Cell Cycle Events, Controllers and Pathways

This research aims to study and model the interactions and events between the cell cycle system components. Cell components, such as proteins and genes, have complex interactions that take place at different speeds. Understanding the complex behaviour of these components is a very challenging task. Therefore, to model this issue, we first have to understand the major components of the system. The cell is considered to be a complex system with a very large and varied number of components that have a high level of complexity.

Cell proliferation process depends on both the correct, and fixed, order of chromosome replication and division during cell cycle. Cell cycle has a set of concentrates such as checkpoints to ensure the integrity of the genome and to maintain the nucleo-cytoplasmic ratio of the cells within viable bounds (Tyson et al., 1995). All events in cell cycles are found in a dependent form; this means that if any event is blocked during cell cycle, the next event will not appear or take place as a result. For example, if DNA replication is blocked by external factors, such as drugs, the next event, such as a cell division event, will be blocked. Another example is when a growth-inducing signal is blocked by factors, such as anti-growth signals or nutrient deprivation, individually or together, it will arrest cell division (Novák, Sible, & Tyson, 2003).

2.1.2.1 Cell Cycle Events

The cell cycle is the way to produce two daughter cells from a single cell being led through the duplication (replication) and division processes (Behl & Ziegler, 2014a). Eukaryotic cells need around 24 hours to divide (Knoblauch, Hibberd, Gray, & van Bel, 1999; Wille, Pittelkow, Shipley, & Scott, 1984) through an ordered and well-organised sequence of events. At the end of the cell cycle, cells produce a copy of themselves; however, abnormal events can force the cell to die (Florens et al., 2002). Specifically, a mother cell produces two daughter cells; this process depends on a specific set of operations that take place during cell cycle regulation, such as cell growth, DNA replication, chromosome segregation (separation of the replicated DNA from the original DNA) and cell division (separation into

two cells) (Braastad et al., 2004; Csikász-Nagy, 2009). In order to accomplish these events cells move through several events called phases, as shown in Figure 2.3 (OpenStax-CNX, 2013).

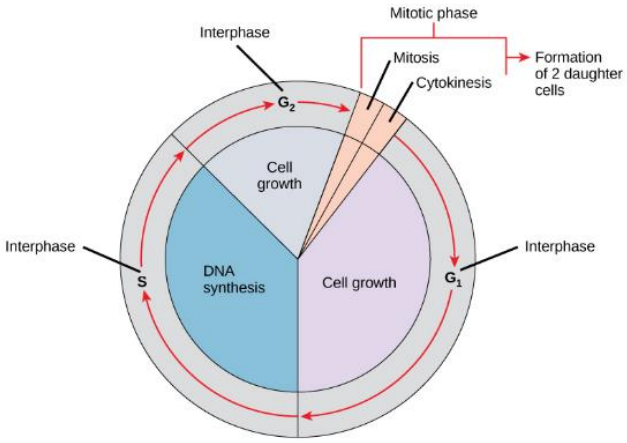


Figure 2.3: Eukaryotic cell cycle phases (OpenStax-CNX, 2013)

In this research, the eukaryotic cell is taken as a case study. The proliferation of eukaryotic cells consists of several events as the cell grows and divides into two daughter cells.

Biologically, each new born daughter cell stays in G0 waiting for growth factors signals to start the cell cycle; this phase is called quiescent. Then, when a cell receives growth factor signals, it starts the cell cycle process. This cycle has a set of phases (Figure 2.3) with the two main phases being the S phase and M phase, which are separated by two gap phases, G1 and G2. So the sequence of the cell cycle is G0, G1, S, G2 and M. These phases are ordered in sequence in a dependent manner.

These phases comprise:

- When growth factors are present the cell cycle moves from G0 to the G1 phase; this phase starts after the previous cell division (pre-synthesis phase); sometimes the G0 phase can be quite long. However, during this phase, the cell starts to grow and the contents of the cell (cytoplasm), along with its functional machinery, is developed. The time for this phase is approximately 1/3 of the 24 hours that Eukaryotic cells need to divide 24 hours.

- The S phase – the synthesis phase: This phase is responsible for DNA replication and the creation of the histone proteins needed for packaging genomic DNA. The average time for this phase is seven hours.
- The G2 phase or post-synthesis phase: In this phase a cell prepares itself to be split into two daughter cells. It grows and increases in size. The time for this phase is approximately four hours.
- The mitosis or division phase (M-phase): In this phase, the DNA is segregated and the cell divided. The time needed for this phase is between 30 to 60 minutes, and it can be divided into five smaller phases (prophase, prometaphase, metaphase, anaphase and telophase); for more details refer to (B Alberts et al., 2002; Behl & Ziegler, 2014b). As shown in Figure 2.4 the five stages of mitosis can be summarised as follows (OpenStax-CNX, 2013):
    1. Prophase: the chromatin condenses into chromosomes (see Figure 2.4) by dehydrating and coiling. The nucleolus and the nuclear envelope disappear. The centriole (animal cells only) divides into two centrosomes, which move apart, creating the spindle.
    2. Prometaphase: this phase is following the prophase and preceding the metaphase, the spindle assembly is carried out in prometaphase right before the chromosomes alignment on the spindle plate that happens during metaphase.
    3. Metaphase: the cell poles are defined when the chromosome is connected to spindle fibers form. In addition, the cell equator, can be identified as the equatorial plate which are attached to the mitotic spindle apparatus via microtubule. Basically, in this part the chromosomes move to the equator of the cell and the centromeres (in the chromosomes) attach to the spindle fibres so that the sister chromatids line up in the centre of the cell.
    4. Anaphase: the centromeres of each chromosome divide and are pulled apart by the contraction of the spindle fibres, thus moving the chromosomes to opposite poles of the cell.

5. Telophase: the chromosomes reach the poles, the spindle disappears and the chromosomes then return to their functional chromatin state by rehydrating and uncoiling. A new nuclear envelope begins to form around the chromosomes at each end of the cell, each with its own nucleolus.

The mitosis phase is followed by cytokinesis in which the cytoplasm is divided into two daughter cells; these are roughly equal in size.

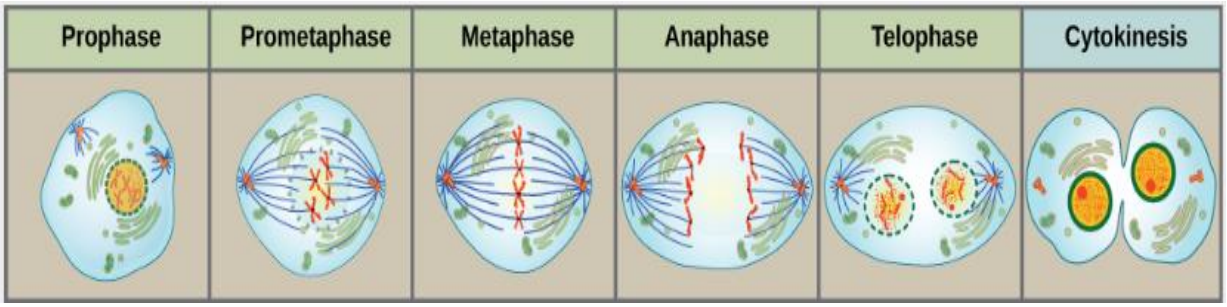


Figure 2.4: Cell cycle phases (OpenStax-CNX, 2013)

As demonstrated previously, one of the study interests is to study the elements that control the phases in the complex cell cycle system; this includes the major components that control and manage each phase. Each phase is considered to be a sub-system with its own dynamic behavior. Thus, the next sub-section explains the major species such as cyclins protiens and there related activators and diactivators that contol the transition process inside the abovementioned cell cycle phases.

2.1.2.2 Cell Cycle Controller Components

This research has several objectives, one of which is to study and model the interactions of the cell cycle controller elements, their relations and effect on the system, such as changes in the status of the entities (*i.e.,* concentration). Obtaining a profound understanding of these aspects requires a sound knowledge of the key controllers of the cell cycle, their roles and interactions.

Cell cycle controller regulation is a motivating topic for bioinformatics researchers as it is one of the best studied systems in biology. The cell cycle control system was still a black box for researchers until the 1980s when the key proteins in the control system were identified. At this time a realisation came that they are distinct from the proteins that perform the processes of DNA replication, chromosome segregation, and so on (Bruce et al., 2010). In fact, cell cycle control proteins are a specialised group of proteins that control these processes in the cell cycle.

A cell cycle controller has a positive and negative feedback mechanisms that help it move from phase to phase. It contains many products and items that organise the cell cycle controller activities. These are organised into two categories – positive and negative. Positive mechanism start and accelerate the processes, such as CycD, CycE, CycA, and CycB. Further, the negative mechanism stop or delay the processes at various points and these include cdk inhibitors and cyclin degradation entities. Under the control of these categories of entities, cell division follows a highly ordered sequence of events (Novák et al., 2003). The next sub-sections discuss the details of the most common cell cycle controller species.

2.1.2.3 Cell Signalling Pathways

Signalling networks are divided into three types of pathway; the first is the gene regulatory network (GRN). This network contains DNA segments (genes) that communicate with each other indirectly using protein products. This kind of network governs the expression levels of protein and mRNA (Düvel et al., 2010). The second network is the signal transduction pathway. This network is predominantly a protein – protein interaction network and is considered to be a very complex communication system. It is responsible for controlling most cellular activities and cell actions (Vander Heiden, Cantley, & Thompson, 2009). The third network is a metabolic pathway that transforms the nutrients into energy (Vander Heiden et al., 2009).

In real biological system, gene regulatory and protein-protein interaction pathways may well have an actual physical boundary, like many processes take place in individual organelles. In cell cycle control system, the different pathways are unified and work as an

integrated whole in achieving the aim of cell division (Calder, Gilmore, Hillston, & Vyshemirsky, 2010; Novák et al., 2003).

2.1.2.4 Cyclins

The first discovery of cyclins was in marine invertebrates (Bela Novak & John J Tyson, 2004). Cyclins are important elements in cell cycle system because cyclin proteins have a significant influence in regulating all major events in the eukaryotic cell cycle. For instance, cyclins are the regulatory components of the cyclin-dependent kinases (CDKs) that control all the major events in eukaryotic cell cycle. Mainly, cyclin-dependent kinases (CDKs) binds with a regulatory protein such as cyclin. Subsequently, without cyclin, CDKs has slight kinase activity; and only work as cyclin-CDK complex is an active kinase (Evans, Rosenthal, Youngblom, Distel, & Hunt, 1983).

There are four important classes of cyclins in mammalian cells (Pinheiro & Sunkel, 2012), and these classes control the cell cycle phases. The first is G1 cyclin (e.g., CycD). These cyclins control the progression through the G1 phase and makes a commitment to the S-phase. The second is S cyclin (e.g., CycE). The main job of CycE is to initiate and complete the replication of DNA. The third class is a G2 cyclin (e.g., CycA), which helps to increase the size of the cell. Lastly, a M cyclin (e.g., CycB) guides the cell cycle in eukaryotes into mitosis (cell division stage) and then re-enters the cell cycle to start again with the G0 then the G1 phases, as shown in Figure 2.5. This figure shows the activity of cyclin products during the cell cycle. For instance, in the cell cycle, CycD and CycE control the G1 phase and CycA and CycE control the S phase, while CycA and CycB, respectively, regulate the G2 and M phases.

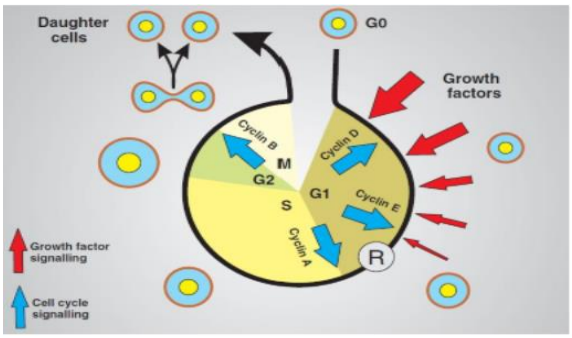


Figure 2.5: Cyclins drive the cell cycle phases **(Pinheiro & Sunkel, 2012)**

Cyclins were identified as proteins in the cell cycle, which can be accumulated, degraded and oscillated during cell cycle phases, as shown in Figure 2.6. This figure shows the changes in cyclin concentrations during cell cycle; these changes in concentration levels reflect the activity of each cyclin species (Pinheiro & Sunkel, 2012). As shown, cyclin concentrations vary during the cell cycle, with their level indicating the phase where their activity is the strongest. For instance, at the beginning of the G1 phase the Cyclin D concentration level increases and then becomes stable in the S phase; it starts lowering protein production during the G2 phase and, in the last stage of mitosis, the concentration level of Cyclin D becomes very weak. Cyclin E protein production starts building up in the middle of the G1 phase and reaches its peak at the beginning of the S phase. It briefly retains peak level and reaches its lowest limit during the last quarter of the S phase.

Cyclin A production starts gradually in the last quarter of the G1 phase and continues building up until it reaches its peak in the first quarter of the G2 phase. After reaching its peak, the protein production starts decreasing and becomes very low in the first quarter of the mitosis phase. Unlike the other cyclins (i.e., Cyclin D, E, and A), which increase in the G1 phase, Cyclin B production increases from the start of the S phase. The production of this protein gradually reaches its peak in the first quarter of the mitosis phase and then it decreases sharply and reaches its lowest limit in the middle of the same phase.

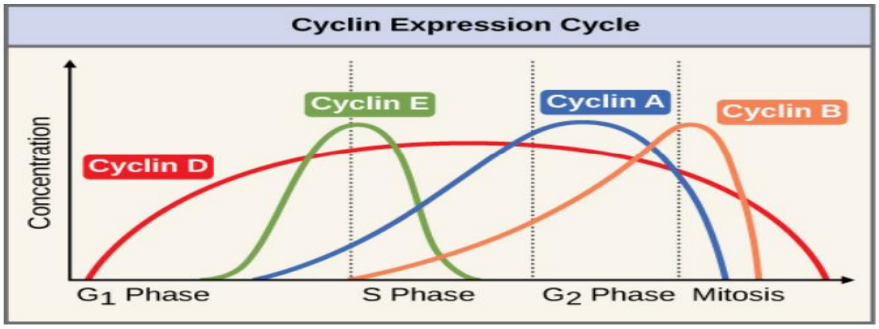


Figure 2.6: Cyclin expression cycle **(Pinheiro & Sunkel, 2012)**

The switching activity of cyclins during cell cycle events is considered a good reason for researchers to study and monitor the changes in cyclin levels during cell cycle (Novák et al., 2003). Accordingly, cyclin activity will be included in our study.

2.1.2.5 Protein Level Regulators (Transcription Factors and Ubiquitination Machinery) During Cell Cycle

The most important transcription factors for cyclins are Myc, TFE and TFB. For example, Myc activates CycD synthesis early in the G1 phase; TFE works with CycE and CycA to lead cyclin synthesis in the S and G2 phases, while TFB works with CycB to activate CycB synthesis in the late G2 and early M phase (Singhania et al., 2011).

Auxiliary proteins combine with in order to direct its ubiquitin ligase activity towards specific substrates. For instance, CDC20A with APC (APC/CDC20A) can process Cyclin A degradation (Hilioti, Chung, Mochizuki, Hardy, & Cohen-Fix, 2001) and CDC20B can combine with APC to form APC/CDC20B that is active during the anaphase and causes the degradation of securin and Cyclin B (Singhania et al., 2011). The Cdh1-form of APC (APC-Cd1) is active during G1 phase of cell cycle and keeps Cyclin B levels very low. In our model, we make assumptions that agree with Fauré et al. (2006) model for these proteins. Our model mainly includes the abovementioned protein regulators and also recent discovered elments such as SCF; this element has significant roles in the ubiquitination of proteins inside the cell cycle such as degrading of CycE. Further, SCF marks several other cellular proteins for destruction (Morgan, 2007).

2.1.2.6 Cyclin-Dependent Kinase Inhibitor (Ckis) And Retinoblastoma Protein (RB)

Cyclin activity can be inhibited by cyclin-dependent kinase inhibitors (CKIs). In the cell cycle, there is a set of inhibitors; for example, P15, P16, P18, P19, P21, P27, P57, and these inhibitors dynamically interact with cyclins during the cell cycle phases (Besson, Dowdy, & Roberts, 2008). For instance, P27 inhibits CycE interactions in the cell cycle (Russo, Jeffrey, Patten, Massagué, & Pavletich, 1996) and for this reason we included P27 in our model.

Moreover, retinoblastoma protein (RB) can inhibit excessive growth in the cell as it prevents cell cycle progression until the cell becomes ready to divide; for this reason RB is called a tumour suppressor. RB also has an impact on cyclin activity inside cell cycle phases. Specifically, RB inhibits transcription factors, such as TFE, which regulates CycE and CycA

activity. In addition, the inhibition of TFE will not push the cell into the S phase. This restricts the replication of DNA inside mammalian cells since it prevents the progression from the first gap phase (G1) to the synthesis phase (S) of cell division cycle. The aforementioned regulations by RB make it an important player in cell cycle and therefore is employed in our model.

2.1.2.7 Checkpoints

The concept of checkpoints was introduced by Hartwell and Weinert (1989). They assumed that an uncompleted cell cycle event must send an inhibitory signal to the next event. These signals are called checkpoint pathways (Novák et al., 2003). As shown in Figure 2.7, the cell cycle in eukaryotic cells is guarded by three checkpoints. These are at the boundaries between the cell cycle phases. The first is at the G1/S boundary, the second at the G2/M boundary, and the third is at the metaphase/anaphase boundary.

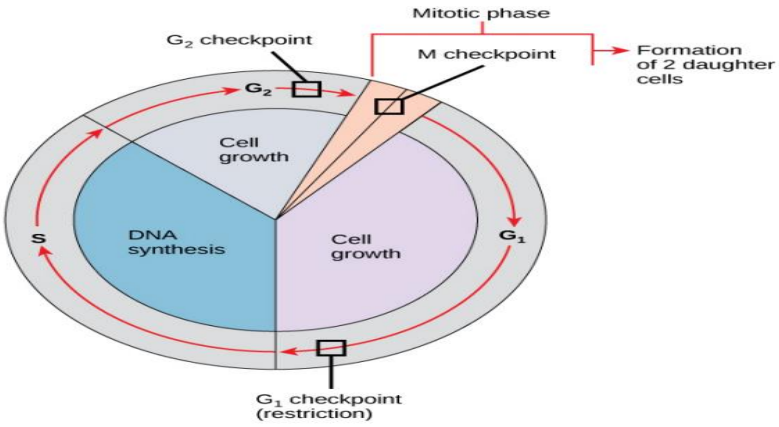


Figure 2.7: Checkpoints in cell cycle **(OpenStax-CNX, 2013)**

Both yeast cells and mammalian cells have checkpoints. Therefore, the G1 checkpoint, is the first control point in yeast is called the START point and in the mammalian cell cycle system it is called the Restriction (R) Point.

The "*START checkpoint*" in *budding yeast* was defined as the earliest genetically controlled step in the cell cycle (Hartwell, Culotti, Pringle, & Reid, 1974). In yeast cells, any cell that

faces a deprivation of nutrients before this point is arrested at the *START point*. But cells that face the same problem after the *START point* will continue the cycle and be arrested when they are in the next cycle. Hartwell et al. (1974) suggest that the most important component of yeast cells is the CDC28 gene as this helps yeast cells pass the *START point.* Later research has shown that it encodes for an important cyclin dependent kinase (Cdk). Generally, in budding yeast, the activation of G1 cyclin-Cdk complexes are usually blocked by checkpoint signals at the *START point* (Novák et al., 2003).

Similarly, the Restriction Point in mammals and growth factors show related activity. The efforts to understand the impacts of growth factors on cell growth process started about 60 years ago. It has been studied using mathematical models (Zetterberg & Larsson, 1985). Subsequently, many mathematical models have been developed for this purpose. These models provide knowledge about the interactions that control cell cycle processes (Csikász-Nagy, 2009). Cells in multicellular organisms need special growth factors (GFs) to continue and complete their proliferation. If a cell gets the required amount of GFs, it continues the proliferation process by moving from phase to phase in cell cycle; otherwise, it enters a resting state (G0).

In the G1 phase, the first point that cells enter is at the Restriction Point (RP) (Bartek, Bartkova, & Lukas, 1996; A. B. Pardee, 1989; Planas-Silva & Weinberg, 1997). RP is a point of no-return in the mammalian cell cycle. A cell can be halted at the RP if the GFs are prevented for some reason. But, if the GFs are prevented after this restricted point, the cell continues the cycle to the next proliferation phases. DNA damage is the major threat during cell cycle as it affects the genome's integrity. DNA damage triggers a signalling network to address the situation. When a cell is in the cell cycle one action is to arrest the cell cycle, which stops and keeps the cell cycle in specific places by inactivating Cdks. This allows time for the DNA repair pathways to repair the damage (Walworth, 2000).

In mammalian cells, DNA damage can be due to internal or external factors. Internal damage is typically due to replication errors or oxidation and these are typically repaired. External factors, such as ionising radiation, can also cause DNA damage. This can produce breaks in DNA strands, either single stranded or double stranded breaks. Several elements work together in response to DNA damage, such as, ATM, P53 and P21 (CKI inhibitor). They act on Cyc/Cdk to inhibit cell cycle progression and initiate the repair or apoptosis (cell

death) pathways, depending on the damage; for more details see (Bartek & Lukas, 2001) and (Novák et al., 2003).

The transition from G2 to M needs a fully replicated DNA. The G2-M checkpoints control this process by using CycB-Cdk1 to block the transition from the G2 phase to the M phase in mammals. In addition, the metaphase (spindle) checkpoint, must ensure that the separation of sister chromatids in the anaphase happens only when all chromosomes are connected to the bipolar mitotic spindle through their kinetochores. In the case of a cell that has free kinetochores it does not exit mitosis or undergo anaphase. The spindle checkpoint blocks both the metaphase-anaphase and anaphase-telophase interactions.

Checkpoints are considered to be surveillance and quality control mechanisms in the cell cycle regulation process (Braastad et al., 2004; Stein & Pardee, 2004). Checkpoints have three important components: (i) sensor, the main job of this component is to detect the errors; (ii) signal generator from the sensor via the transduction pathway; (iii) response element that takes action, such as blocking cell cycle progression (Stein & Pardee, 2004). In a cell cycle system if any condition for cell division has not been met the cycle is halted at any of these checkpoints.

However, in this study, we are not investigating DNA damage or the elements of cell cycle checkpoint activity. Checkpoints will be part of our future work as an extension of this research. The models we rely on, such as (Fauré et al., 2006) and Singhania et al. (2011), did not involve checkpoints in their implementation. We are focusing in this study on the major controllers of cell cycle system activity, like cyclins, that flow over the cell cycle. Therefore, all the aforementioned biological elements that control cyclin activities will be investigated in Chapter 3 where we explore the relations and biochemical reactions between these elements to incorporate them in the models we develop.

The next section explains computational modelling for cell cycle systems. It discusses several computational models and approaches that have been developed to model cell cycle. Models can help researchers study, predict, understand and provide explanations without the need to use laboratory experiments, as these are costly challenging and time-consuming.

2.2 Computational Approaches to Modelling Cell Cycle

Study of biological systems can provide understanding of a system's structure and its dynamic behaviour (Bruggeman & Westerhoff, 2007; Ilsley, Luscombe, & Apweiler, 2009; Kremling & Saez-Rodriguez, 2007; Wolkenhauer, Kolch, & Cho, 2004). Systems biology research depends on, and provides, very large amounts of data which can be collected by several means, such as high throughput technology, including mass spectrometry (Reinders, Lewandrowski, Moebius, Wagner, & Sickmann, 2004; Xie, Liu, Qian, Petyuk, & Smith, 2011) and microarrays (Barrett & Kawasaki, 2003; Lamartine, 2006). Current studies use these data sets to elucidate molecular networks and their properties, including their dynamic behaviour, but these data sets should also have the ability to provide phenotypic properties for specific biological systems (De Backer, De Waele, & Van Speybroeck, 2010).

Biological systems are elusive mechanisms and consists of many interactions that need to be clarified. For instance, the dynamic and complex behaviour of any system can emerge from individual interactions that happen between molecules. Moreover, this emergent behaviour can be studied using computational-based models.

In this section, we explore modelling of biological systems and provide meaning for system components as well as the computational models and approaches that have previously been developed for cell cycle.

2.2.1 Modularized Biological System Components

The main objective of modelling biological systems is to explain and investigate natural phenomena in a simplified way (Büchsenschuß, 2014). Biological systems, such as cellular compartments and biological cells, consist of entities and their interactions. Therefore, any model that represents a biological system is abstract and uses simple representations for its entities and interactions.

2.2.1.1 System Entities

Essentially, a system includes sets of components that represent a system's species, subjects or a specific part of biological system, or all of them together. For instance, entities can be formed from biological molecules (specie/s) or sets of similar biological molecules (subject/s) like whole set of transcription factors of the same kind. Further, these entities can also be parts of molecules when a gene becomes part of the chromosome molecule. Any environmental condition can also be considered as an entity in the system.

Over time, system entities can change their states from one state to another. The current state of a system is defined depending on the entirety of all the current states of the elements that comprise the system. Moreover, the current state of each entity depends on their properties as well as their previous state. Each entity has a set of properties that describes it. For example, each protein in the cell has a shape, a mass and a location that describe them. In addition, a set of proteins of a specific type also has the concentration of its molecules as a property. Each current state can be specified using numerical or linguistic values.

2.2.1.2 Interactions

Entities in the system communicate with each other through interactions, as the proteins themselves can interact together. Furthermore, cellular processes can be classified as a set of interactions between pairs of entities or sets of entities. These kinds of interactions formulate the dynamic behaviour of biological systems; otherwise the system will display static represntation. More details on interaction dynamics can be found in the next sub-section.

System interactions are the main effector on the states of the system entities. Entities affect each other via interactions to change their states. Each entity can be an effector that influences another entity, or it can be a target for another entity to change its state. Sometimes the entity can be either an effector or a target depending on the task to be undertaken. This task is managed by system interactions that control its behaviour.

2.2.1.3 System Dynamics

Basically, system components, such as entities and interactions, provide a static representation of the model. A static model can be used to represent the topology of a biological system (Xu, Wang, & Ding, 2004). This behaviour can be transformed into dynamic behaviour if these components are provided with temporal and spatial dimensions. For example, the interactions between entities can emerge as dynamic behaviour. As mentioned previously, the current state of the system is specified by the current states of all system elements, and the interactions, e.g., an enzymatic conversion or transport process, can change the states.

The dynamic behaviour of the system can be graphed as trajectories of these states where they change in space over time, while a static description cannot provide the dynamic behaviour due to the complexity of these dynamics. Studying dynamic behaviour is very important for explaining and scrutinising biological phenomena, such as the localisation of system components and oscillations in protein concentrations. This is one of the main objectives of this study.

2.2.1.4 Description Of Models

In models, each entity is mostly represented by one state, such as the level of protein concentration. Accordingly, in computational-based models, states are represented by numerical values and mathematical functions represent the reactions. Modellers and researchers always try to simplify complex systems, either to understand them, or to explain them better.

Therefore, the most popular technique to simplify a complex system is to use a graphical representation. Most models are represented by a graphical network that consists of nodes and edges; this is an attempt to simplify the nature of the system's activity. Biological systems can be represented as a network where nodes represent biological elements/species and the edges in the graph represent the biological interactions. If the

edges in the graph have no sign, this is called an unsigned graph. In this kind of graph, each edge represents the relations between elements; but the sign of the edge is useful to represent the type of relations (Activation or inhibition). Any network that has signed edges can be called a sign-directed graph.

A sign-directed graph is part of graph theory where the graph consists of nodes/vertices and edges/arcs. These components represent the reactionary molecular elements in biological systems. For instance, the nodes represent biological elements and the edges represent their biological interactions. All edges are signed with the direction from the source node to the target(s) node(s) (Adhikary & Eilers, 2005).

Furthermore, in a sign-directed graph, all the edges are directed from the head node, the source/regulator of the tail node, to the node being regulated (target or tail). In addition, all components in a sign-directed graph are assembled from an extensive survey of biological information in the literature.

This kind of graph accurately representing a model can be very helpful for several reasons:

- It enables structural analysis of the network, such as network connectivity and process cycles.
- It helps to visualise our knowledge relating to biological systems by using graphical representations of biological entities and their reactions.
- These graphical representations reflect the static structure of the system. They also reflect the component connectivity of a system and their relations, such as the effector and target entities. However, they cannot explain the dynamic changes in entities' states over time.

Moreover, dynamic models are more complex than static models because of the need to provide extra information for temporal processing. Furthermore, the interactions are represented by functions that calculate the current and next states for both the activator and target entities.

2.2.1.5 Knowledge-Based Models

A knowledge-based model is a model that is built depending on the knowledge available within a current hypothesis and the information available about system components and interactions. Existing knowledge can be classified into two classes: first, experimental data, such as results observed during experiments; secondly, previous knowledge, such as the knowledge acquired from the scientific literature. For instance, the reaction rates and concentration levels for proteins can be obtained from experimental data. Therefore, these models are built on the current knowledge together with hypotheses.

These kinds of models can provide new knowledge and new hypotheses. This allows the researcher to compare the new knowledge with old knowledge, and then modify the old knowledge or create new knowledge with new hypotheses.

2.2.2 Current Computational Models and Approaches for Cell Cycle

Complex biological systems, such as cell cycle, have several issues that need addressing; these include the system structure, interactions, function and dynamic behaviour. For instance, cell cycle system consists of a large number of elements interacting dynamically. Most of these aspects are unclear. Medical and biological researchers are currently trying to develop approaches and methods to investigate and understand the cell cycle functionality and its mechanisms. These investigations can help improve treatment and diagnoses of diseases (Hanahan & Weinberg, 2000).

Modelling approaches greatly help in understanding and addressing issues in the cell cycle system. Researchers depend on the available information to develop models to address one or more of these issues. For example, they use gene regulatory networks and protein-protein networks to analyse cell cycle events and activities in relation to gene expression (Kohn, 1998). Researchers have come to understand the structure and behaviour of these networks depending on the limited data available (Csikász-Nagy, 2009).

Modelling approaches can provide specific information. One of the main outcomes of modelling is to predict temporal changes in the states or levels of the interacting elements. Another outcome is addressing system events or emerging properties arising from the collective behaviour of the elements. These may include a cell passing various cell cycle phases with the support of participatory elements. Yet another outcome is an even higher level system; where a system malfunction gives rise to diseases, such as cancer. A deep understanding with rich knowledge can be attained by studying important aspects of the cell cycle, such as cellular signalling and other processes involved in cell cycle regulation (Hanahan & Weinberg, 2000). Understanding biochemical reactions between multiple proteins, genes and their mRNAs (Kohn, 1998) aids our understanding at a fundamental level.

Modelling approaches can be classified into several classes (Csikász-Nagy, 2009). The first class is logical or discrete models that describe a system in a qualitative form. The second class is continuous models, which are used to express continuous changes in protein concentrations in biological systems. The third class represents biological systems with noise; these are called stochastic models. The fourth class is hybrid models; in this type of models a combination of different methods are used. Two or more methods work together to complement each other in representing a biological system. In some cases, more than three methods can be used.

Several computational models and approaches have recently been developed for the cell cycle. They have provided useful information that helps researchers (biologists) study and predict outcomes. This section provides a review of systems biology and mathematical modelling approaches used to describe properties and behaviour of complex systems, such as the cell cycle system.

2.2.2.1 Logical or Discrete Models

This modelling method, which depends on discrete or logic-based concepts, is considered a simple and basic deterministic model; an example of which is the Boolean Network. The first Boolean model was developed by (Glass & Kauffman, 1973; Thomas, 1973) for gene

regulatory networks. Boolean models consist of a set of entities that represent a biological system's components, such as genes, proteins and other small molecules. The edges represent the relations between the entities (nodes) and the activity state of each node is represented as inactive or active in the form of [0, 1].

Boolean models of the cell cycle system have been proposed previously, most of these were developed to represent and mimic the yeast cell cycle. However, very few models have been developed to model the mammalian cell cycle controller system.

The concentrations of the system components can be discretised by their threshold concentrations. For example, all concentrations that have values lower than a threshold value are "OFF"/0, while all other concentrations that are greater than the threshold value are "ON"/1. The edges represent the relations between entities (nodes), and the activity state of each node is represented as inactive or active in the form of [0, 1].

In a biological system, nodes represent proteins and genes while edges represent interactions. Figure 2.8a represents the structure of a Boolean network where A, B and C represent network nodes and the directed arrows denote activation or inhibition. Figure 2.8b shows the rules that manage the activities in a Boolean network system. In the cell cycle system the components can be represented by Boolean entities with binary values referred to as "OFF" and "ON."

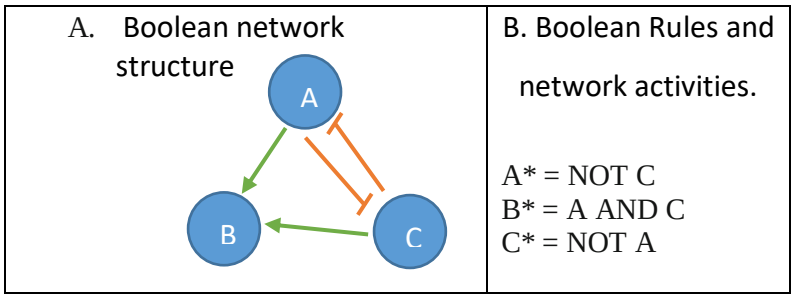


Figure 2.8: A simple regulatory Boolean network model. (A) Network representation: A, B, and C are the system components or network nodes. The directed edge ---> or —| denotes activation or inhibition, respectively. (B) Boolean rules controlling the dynamics of node states of the Boolean network in (A). Each state is represented by the node name and the * in the label denotes the next state of each node over time. So A* means state of A is 'on' next time if the conditions on the right hand side are satisfied.

A well-known Boolean model that represents the yeast cell model was developed by Li, Long, Lu, Ouyang, and Tang (2004). They examined the temporal robustness of the cell cycle concerning checkpoint conditions in budding yeast for the construction of a Boolean network. However, their approach is only suited to small and well-characterised systems. Braunewell and Bornholdt (2007) also proposed a Boolean network to study dynamical attractors in the yeast cell cycle. Their proposed Boolean model consists of 11 genes that coordinate the dynamics of the yeast cell cycle. Braunewell and Bornholdt (2007) developed the framework for their proposed model using the concept of discrete threshold dynamics and this allowed them to introduce fluctuations in the processing time. They introduced noise into the model to account for the effects of biochemical stochasticity. However, Braunewell and Bornholdt (2007) used a very simple model and neglected to provide a detailed description of the system with different time scales involved in different processes. Their results cannot be translated directly into the cell cycle system.

Another model was introduced to study fission yeast, *Schizosaccharomyces pombe.* Davidich and Bornholdt (2008) introduced a Boolean network model for the cell cycle of the fission yeast, *Schizosaccharomyces pombe*, which was based on known biochemical interaction topologies. Using this model, they dropped the 47 essential kinetic constants and parameters of the ODE approach and still obtained the same results for the biological sequence of activation. However, their work was limited to a single cell size checkpoint. Furthermore, they also neglected DNA damage checkpoints.

Studying the wild type and mutant phenotypes has been undertaken in Boolean networks by researchers such as Irons (2009), who developed a new Boolean model for the budding yeast cell cycle that described a wide range of wild type and mutant phenotypes. Irons (2009) model was robust against perturbations in reaction times and network component states. It overcame the FEAR pathway limitations of the Boolean model developed by Li et al. (2004). They included the MEN pathway components in their proposed model. Furthermore, their model was suitable for sub-network analyses that can be used to identify key sub-dynamics for viable cell cycle control. The Irons (2009) model produced different times it took to pass through each phase compared to the actual biological times.

The yeast model was limited in providing a better understanding of complex biological systems due to its simplicity (C Gérard & A Goldbeter, 2012). To understand the behaviour of complex biological systems, the integration of molecular data into a formal dynamical model is required (Fauré et al., 2006). This is the situation with the mammalian cell cycle. Fauré et al. (2006) proposed a Boolean model for the mammalian cell cycle based on Bela Novak and John J Tyson (2004) ODE system to describe cell cycle behaviour. Bela Novak and John J Tyson (2004) model was based on 27 equations and other experimentally-derived parameters. The Fauré et al. (2006) model is still considered a benchmark for all the Boolean models developed later. The major contribution of Fauré et al. (2006) model was its simplicity due to the reduction in the number of equations and the focus on important elements, including cyclins D, A, E and B, Rb and P27. However, the limitation of their model is that it did not incorporate cell cycle checkpoints. Moreover, this model does not account for the effect of some key cell cycle regulatory proteins, such as SCF, c-MYC and TFB; more details are investigated in the next chapter – the Boolean model.

Boolean networks are easy to design and program and they provide information on the interactions and system properties simply and qualitatively. Furthermore, this kind of network does not depend on the model parameters. However, these models also have limitations. For instance, the discrete model presents its results in qualitative form, and this is not enough for researchers to gain a deep understanding of system behaviour. Furthermore, Boolean models cannot describe the intermediate activity levels of elements, or which element is more active than others, because they are not part of Boolean model functionality; this is one of the major drawbacks of discrete models.

This kind of model can also only represent the local state changes of proteins (nodes) in discrete time steps. In the temporal development of the system, time is assumed to be in a synchronous or asynchronous form. In synchronous form, all nodes update their states at each time step; whereas, in asynchronous form, randomly selected nodes update their states at each time step. More details relating to Boolean update schemes can be found in the next chapter within section 3.1.4.

2.2.2.2 Continuous Models

This model class mainly represents the mathematical approaches of ordinary differential equations (ODEs). It is a deterministic model like Boolean models (Glass & Kauffman, 1973; Thomas, 1973). ODE models can successfully explain the continuous behaviour of biological systems.

For the cell cycle, several ODE models have been developed to provide detail information; for example, the concentrations of proteins and other cell cycle components. Over the last century, researchers have discovered that molecular interactions controlled cell cycles. Over that era, many groups have been working on mathematical models to analyse these interactions, as mentioned in Table 3.1. The group of Novak and Tyson (1993) as one of the significant leaders in the field; has published more than 40 papers on cell-cycle regulation. Currently, their models are considered benchmarks for computational systems biology.

In 1993, Novak and Tyson (1993) studied the mitosis phase in eggs of the frog, *Xenopus laevis,* and discovered that their model could provide a reliable switch for entry into mitosis. Novak and Tyson (1993) revealed the "hysteresis" that exists in the MPF-cyclin relationship. Later, they provided a detailed model for cell-cycle regulation by describing the Cdk control network in the budding yeast, *Saccharomyces cerevisiae*. Afterwards, Bela Novak and John J Tyson (2004) developed a new mathematical model with the inclusion of restriction points that controlled the mammalian and yeast cell cycles. Their model was mainly based on the effects of growth factors on the cell cycle. Moreover, they examined the changes and mutations in some key elements of the mammalian cell cycle, including Cyclin E and Retinoblastoma protein (RB). Their model had many shortcomings such as in regard to cell growth and division.

More recent models, such as that of K. C. Chen et al. (2004), researchers proposed another continuous model for the yeast cell cycle based on biochemical rate equations. However, their model documented several inconsistencies between the model and their experiments. Their model also requires estimating the effective values of the biochemical rate constants, which are difficult to measure. (Ferrell, Tsai, & Yang, 2011) also proposed a mathematical model to explain the oscillatory biochemical circuits in the context of the

Xenopus embryonic cell cycle and the stability of the system. The major contribution of their model was the specification of kinetic parameter sets that produced oscillations. Furthermore, they documented the reasons for the oscillations in their model.

The above models and other models in Table 2.2 can properly represent continuous biochemical reactions as shown in Figure 2.9. ODE models can provide realistic results for continuous changes in protein concentrations of biological systems. Furthermore, ODE models can mimic the dynamic behaviour over time in a better way than other models, such as Boolean models. Figure 2.9 explains the major differences between ODE models and discrete models since almost all models can be classified into one of them.

Table 2.2. Some ODE mathematical models of mammalian cell cycle

Year	Organism/Cell Type	Type of Model	Reference
1969	No specific organism	ODE	(Sel'kov, 1969)
1974	No specific organism	ODE	(DA Gilbert, 1974)
1999	Mammalian somatic cells	ODE	(Aguda & Tang, 1999)
2006	Mammalian somatic cells	Delay differential equations	(Srividhya & Gopinathan, 2006)
2008	Mammalian somatic cells	ODE	(Yao, Lee, Mori, Nevins, & You, 2008)
2009	Mammalian somatic cells	ODE	(Alfieri et al., 2009)
2010	Mammalian somatic cells	ODE	(Ling, Kulasiri, & Samarasinghe, 2010)
2010	Mammalian somatic cells	ODE	(Iwamoto et al., 2011)
2011	Mammalian somatic cells	ODE	(L. Zhang, Cheng, & Liew, 2013)
2014	Mammalian somatic cells	ODE	(Weis, Avva, Jacobberger, & Sreenath, 2014)

a.

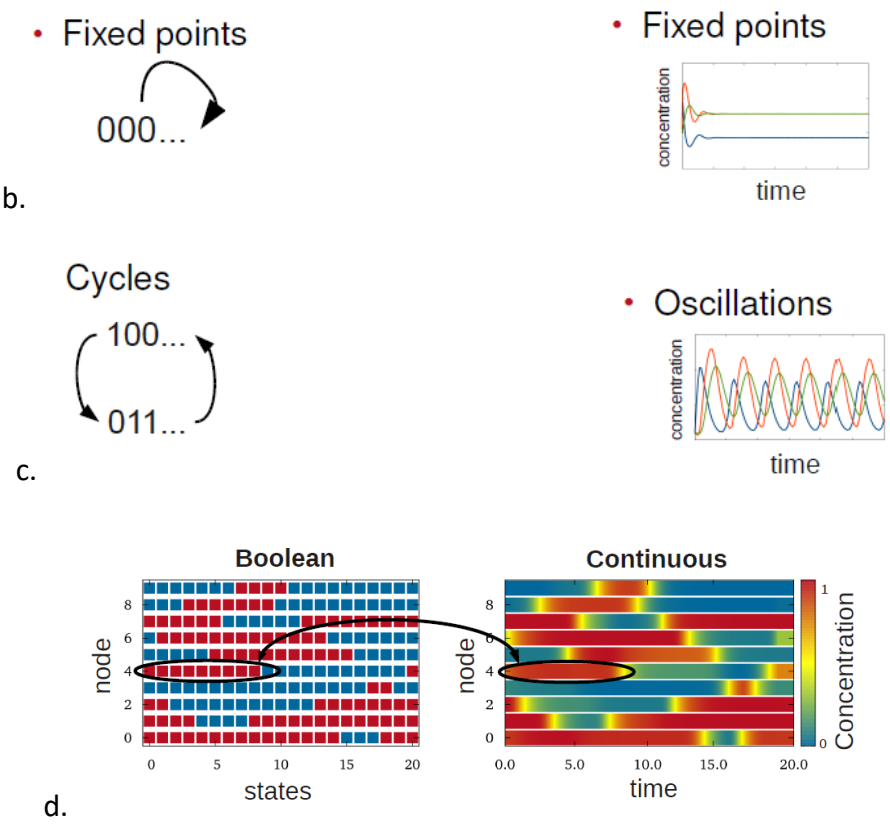


Figure 2.9: The functionality in ODE and Boolean models and how they explain the dynamic behaviour of system components (node/s) in a biological system. a) Information representation; for example, on the left side, the Boolean model represents node/s in a binary digit form, 0 OR 1, while ODE provides exact number values. b) Boolean and ODE representation of the fixed point states when the biological system becomes silent over time. c) Boolean network representation of the entire system activity and the global behaviour as repeated cycles of time steps, while ODE representation shows repeated cycles in continuous form like oscillations. d) Boolean model explanation of the changes in system components from state to state, 0 to 1, and vice versa, over time, while the ODE model explanation shows the changes in concentrations for system components over time.

However, the problem with ODE models is that when the model has a large number of components, such as genes and proteins, it becomes very difficult to analyse them (Irons, 2009). Moreover, the source of the parameters used in continuous models is generally experimental estimations or inferences. Either way, researchers or modellers always need to face the problem of parameter availability and reliability. As a consequence, if ODE is used for large complex systems, such as the cell cycle system, a large number of unknown

parameters need to be estimated. These models can also not provide a clear view of the order and sequence of biological events.

2.2.2.3 Stochastic Models

Stochastic models represent a complex system, such as a cell cycle system, along with its inherent noise. It randomly chooses items or nodes. In this kind of model, stochastic behaviour is offered using mathematical methods. The most popular mathematical methods used are the Gillespie Stochastic Simulation Algorithm (GSSA) (Kar, Baumann, Paul, & Tyson, 2009) and the Stochastic Langevin Equations (SLA) (Steuer, 2004).

Some stochastic models of cell cycles have already been produced, such as that by Chiorino and Lupi (2002), who proposed a stochastic ordinary differential equations approach to investigate G1 phase variability such as CycD, CycE, RB ,P27 and E2F from a molecular point of view. In addition, they modelled the transition from the G1 phase into the DNA synthesis (S) phase. Their model can be used successively to predict protein behaviour. However, their model cannot describe cell cycle progression so easily and linearly as the other models.

Y. Zhang et al. (2006) also proposed a stochastic model for the network regulating the cell cycle of budding yeast. They advanced a probabilistic Boolean network model consisting of 11 nodes. The stochastic effect was controlled by a Markov chain that depended on a temperature factor, such as parameter β. They concluded that both the stationary biological states and the biological pathway had a stable state for a wide range of 'temperatures'. However, their model showed a sharp transition in behaviour at a critical temperature, βc, which made the dynamics dominated by noise. Another model was built by Bean, Siggia, and Cross (2006) to investigate the coherence of the *START* phase in the cell division cycle; Bean et al. (2006) developed a stochastic model based on quantitative time-lapse fluorescence microscopy on a multi-cell cycle timescale. Multiple fluorescent markers were used to examine the coherence of the *START* phase in the cell division cycle. The major contribution of their model involved its applicability to yeast grown under time-lapse conditions and its applicability to semi-automated assignment of micro-colony

pedigrees. Furthermore, their model could be used for stochastic effects in prokaryotic transcriptional regulation.

Later, Braunewell and Bornholdt (2007) examined the cell cycle of budding yeast based on the concept of discrete threshold dynamics to study the stability and dynamical attractors of the cell cycle system. However, the major limitation of their model was that it did not represent checkpoint mechanisms. Simplified stochastic models have now been developed, such as that of Zámborszky, Hong, and Nagy (2007). In their model, they developed a simplified version of a four-variable mammalian circadian clock model. They adapted Novak and Tyson's mammalian model (2004) to build their model. The model was built and a set of assumptions were made to simplify the system; for example, they assumed that some elements in the system were the same and some system components were more stable than others, which introduced an autocatalytic positive feedback into the system. In the stochastic simulations, Zámborszky et al. (2007) they introduced noise into the cell cycle regulatory equations. They rewrote the cell cycle model equations as Langevin type equations with multiplicative noise (Steuer, 2004). As a result of their simulation they stated that the clock would play an important role in cell size control via a Wee1 element in their model. However, Zámborszky et al. (2007) did not address a comprehensive mammalian circadian rhythm model.

More recent models such as that of Kapuy et al. (2009) simulate the mitotic exit of mammalian cells. Their model predicted that in a time interval of 30 to 70 min the percentage of cells that completed their irreversible mitotic exit would gradually increase. They observed that mammalian and yeast cells shared an important point, the activation of Cdh1/APC in the model and the inactivation of CycB transcription at low levels of Cdk1 activity. This happens due to the additional feedback loops that help in the irreversibility of the mitotic exit. However, their model has a challenge, particularly in mammalian cells, as the phosphatase that counteracts Cdk1–CycB is yet to be identified; therefore, illustrating the implied system level mechanisms will be a challenge for future research.

Furthermore, Claude Gérard and Albert Goldbeter (2012) developed a model to investigate the conditions that make the mammalian cell cycle entrained by a circadian clock. They observed that entrainment to a circadian period could happen when the period of the cell cycle before coupling was either smaller or larger than 24 h. This stochastic model reported a potent prediction model. Also, the transition from an entrained period of 24 hours to the doubled time of 48 hours might be the outcome from the level of growth factor or by a decrease in coupling strength. However, coupling to the circadian clock can lead to complex periodic oscillations or chaotic oscillation dynamics of the cell cycle in the form of an end replication.

Generally, the advantages of the above stochastic models are that they can represent a system more realistically than deterministic models. In addition, this kind of model can be helpful when the protein concentration is too low. Further, they can be used when it becomes hard for the ODE models to handle the system (Wolkenhauer, Ullah, Kolch, & Cho, 2004). Since they are not deterministic models, the ODE models come up with different results each time they are run. They can also produce unreliable or unpredictable results in each interaction.

2.2.2.4 Hybrid Models

In hybrid models, modellers combine several methods to benefit from their advantages. The expected outcome from a hybrid model is to cover shortcomings and issues in one method by another and, thus, provide a better understanding of complex systems. For instance, hybrid models can use both discrete and continuous aspects of the same model to realistically represent corresponding aspects in the system.

Singhania et al. (2011) proposed a hybrid model to represent the mammalian cell cycle. Their model was based on the best features of continuous differential equations, discrete Boolean networks and stochastic approaches. The model represented the continuous changes in cyclin (CycE, CycA, CycB) synthesis and degradation. They were tracked by

piecewise linear differential equations, while the regulators of cyclin synthesis (transcription factors (TFE and TFB) and the ubiquitination machinery (SCF, Cdc20, and Cdh1) were represented by discrete variables (0 or 1), as shown in Figure 2.10. The variation in the state of discrete variables was based on a predetermined sequence without the need for Boolean rules. The timing between transitions was also determined, in part, by cyclin accumulation and degradation.

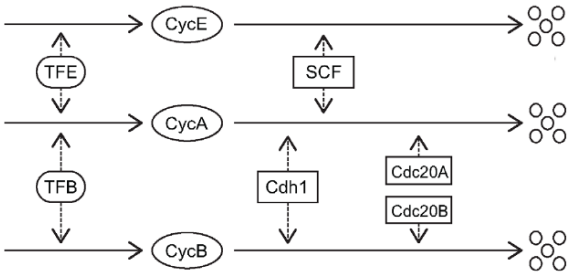


Figure 2.10: Hybrid model (Boolean- ODE): used in modelling mammalian cell cycle regulation. Singhania et al. (2011)

This is a simple model that represents the most important regulatory elements in the cell cycle system. However, this model misses some important elements, such as CycD. More can be done by incorporating additional elements, such as CycD, and their interactions to ascertain new knowledge. In addition, this is challenged by the available experimental data because this model needs to estimate realistic values for many of the kinetic constants that determine the reaction rates. This model has been chosen for our study and further investigation and analysis will be presented in the next chapter.

In addition, Noël, Vakulenko, and Radulescu (2013) proposed another hybrid model for the mammalian cell cycle. Their model was based on a combination of continuous and discrete methods to model cell cycle dynamics. The model depended on hybridisation from a smooth biochemical model. However, the appropriate hybridisation scheme used depend on learning from training a set of smooth model trajectories.

Another hybrid model, developed by Noel, Grigoriev, Vakulenko, and Radulescu (2012), was based on a tropical geometry heuristics that unravel the commonalities of cell cycle

models. The model combined quasi-equilibrium states that are represented by slow invariant manifolds and excitability. Their system had a simple structure of monomial differential or differential-algebraic equations. However, their model was limited to ODE for the availability of experimental data.

Recently, Behaegel, Comet, Bernot, Cornillon, and Delaunay (2015) documented that time was pivotal in many biological systems, especially the cell cycle. Their model was based on a five-variable model of the mammalian cycle. They determined the parameters by applying formal methods to an underlying discrete model using timing observations of the cell cycle.

Based on this review, most popular mammalian cell cycle models were selected for our study. They have been chosen to enhance and improve both their biological and computational aspects. Further, they have a significant influence on the underlying mechanisms and principles of the mammalian cell cycle system. The next chapter provides more details about this study, including the enhancement and development processes for the models mentioned above.

Moreover, an alternative method is introduced to describe and cover the shortcomings of both discrete and continuous models. This method depends on fuzzy logic theory (fuzzy logic model); the next section presents an overview of fuzzy logic.

2.2.2.5 Fuzzy Logic

Most of the models mentioned in the literature can be categorised based on the functionality of their models into two categories - discrete and continuous approaches; this is a general form. Continuous models are based on mathematical models, such as Ordinary Differential Equations (ODE), that provide accurate continuous solutions. However, the lack of real data and parameters create limitations to the use of continuous models. Nevertheless, a dynamical system, such as the Ordinary Differential Equations, provides a natural and detailed description of molecular events. For this reason, this technique is

considered the most successful (Shin & Bleris, 2010). However, these models are not suitable for representing large networks, as discussed previously. The availability of information on proteins and pathways is also limited (Morris, Melas, & Saez-Rodriguez, 2013). On the other hand, small-scale models are not comprehensive so they cannot be considered as full representations of the whole system.

ODE models provide realistic values but they cannot provide a direct interpretation of entity states; this is a drawback of these models. For instance, ODE models can provide a value, but cannot specify whether it is low, medium, or high, and at which level it affects other targeted elements. In addition, most ODE models in literature use fuzzy (vague) expressions to explain and investigate their results. For example, to explain the real number values for protein concentrations seen in the results, they use fuzzy expressions to explain these values and they attach additional illustrations to the values to determine whether the real number value is low or high. Indeed; this knowledge is hidden from ODE models while fuzzy logic models provide an interpretation of the entity states. Furthermore, fuzzy logic allows the modelling of imprecise, uncertain and limited knowledge and data domains to capture essential trends and behaviour.

In contrast, discrete models represent a system of logical models with graphical methods to explain the relationships between network elements, such as protein interactions that can be formed into a logical sequence of events. Discrete models use limited data and can still be helpful in modelling and understating complex systems. As discussed earlier, one of the promising discrete models is the Boolean model that represents interactions among network elements using simple logic-based rules to represent system dynamics.

A Boolean network entity is called a species (Kauffman, 1969). The state of these biological entities can be in one of two forms, “ON”/present or “OFF”/absent. These make a solid border between entity states. This format is acceptable to represent system activity when there is no need for extra details. However, this format is unnatural and cannot be held as an intuitive biological representation. On the other hand, fuzzy sets can be more flexible and unconstrained to expose the transitions from state to state to explain the intermediate state of the biological entity. The intermediate state in fuzzy logic eliminate the drawbacks in the Boolean network, such as unnatural strict discretisation of borders in the Boolean states (Windhager, 2013). More details will be presented in Chapter 4. Furthermore,

further refinements of Fuzzy logic models that enable automated model generation through Particle Swarm Optimisation are presented in Chapter 5.

This chapter reviewed literature on cell cycle and modelling and highlighted the pros and cons of the current and existing computational modelling approaches. Based on this review, few Boolean models have been selected for further expansion within the Boolean framework and further model improvement using Fuzzy logic in Chapters 3, 4 and 5.

Chapter 3: A novel Boolean Model – Qualitative Mammalian Cell Cycle Controller System

The main aim of current systems biology is to clarify the complex behaviours of biological systems. An emphasis is on understanding how these behaviours emerge from the interaction of system components. Predominantly, a combination of experimental knowledge and computational approaches are used to obtain global insights into complex biological systems.

Mathematical modelling, such as Boolean Networks and Ordinary Differential Equations (ODE), can provide insights into complex biological systems. Boolean models have become popular because they can be easily used for large-scale and complex networks. A biological system can be considered as a set of biological networks represented by Boolean Networks. They are easily designed and programmed as they use information on the interactions and system elements in a simple and qualitative way. Yet, they can provide valuable insights into the overall system behaviour and properties. In addition, this kind of network does not depend on model parameters, as for example ODE models do (Wawra, Kühl, & Kestler, 2007).

The ability of Boolean networks to mimic biological networks can confer insights into the dynamics of biological systems, such as cell cycle system, especially when describing stable states and system cycles (attractors) in networks. This simplicity makes them a popular class of modelling methods for biological networks. Therefore, our interest is to develop a

qualitative system that is useful for modelling biological systems, such as cell cycle control system. This can be undertaken after understanding the components of the mammalian cell cycle system and their interactions and any outstanding issues relating to cell cycle, as presented in the previous chapter.

Hence, this chapter provides background information to understanding the fundamentals of mathematical Boolean networks and how Boolean networks can mimic biological systems, especially cell cycle control system. In addition, Boolean network features are also explained in this chapter. For example, we explain Boolean attractors, analysis of a Boolean model and also demonstrate how a Boolean model can be applied to perform structural analysis of signalling networks in the context of cell cycle control system. Moreover, this chapter discusses Boolean models that have been used for modelling the cell cycle process, and highlights their pros and cons in order to pave the way for the advanced model proposed in this research.

This chapter presents a novel Boolean network which mimics the cell cycle controller system. In addition, it presents a number of approaches used to identify attractors in synchronous and asynchronous Boolean networks and their use in the BoolNet *R* package inside *R* language environment and Matlab software.

3.1 Boolean Networks

3.1.1 Fundamentals of Boolean network functionality

Boolean Networks have become a useful and effective alternative to quantitative methods such as differential equations that require kinetic parameters when in reality these data are either not available or are limited. Boolean networks are used to model complex biological systems, such as cell cycle system.

The structure of a Boolean network consists of a set of connected nodes. Each node has a binary state: [0, 1], [false, true], or [active, inactive]. Boolean networks are built based on a set of rules, hypotheses and assumptions to represent the interactions of their elements. Boolean rules are organised and managed by logical operators. They are responsible for setting up the logical expressions of the Boolean model by combining each of the regulators within each expression. Modellers use three common Boolean operators when building Boolean Networks, which are “AND,” “OR,” and “NOT.” The operator “AND” represents the association of two or more regulators in a dependent manner where all the regulators are needed for the activation of a node. The operator “OR” is used to combine more than one of the independent regulators together, especially when the node can be stimulated by any of its regulators. The negative regulation (inhibition) process in a biological system can be represented by the operator “NOT.” The symbols representing each operator are summarised in Table 3.1.

Table 3.1 Boolean operator symbols

Rule	Symbol
OR	|
AND	&
NOT	!
No regulator	<>

Boolean operators will be expounded further in the next section of the chapter. Modellers can harness Boolean operators such as “AND,” “OR,” and “NOT” in Boolean functions. In other words, the relations and interactions between nodes of the network can be determined by Boolean functions. Consequently, Boolean functions can explain the interactions between biological elements.

3.1.2 Determining Transfer Functions

Each node in the Boolean model can change its state by using a Boolean transfer function. The Boolean transfer function can be specified, depending on the available knowledge, to organise the interactions between the system's components (nodes). The transfer function can use the operator "AND" to indicate conditional regulations where two (or more) regulators are needed to activate the target node. The operator "OR" can be used in the case where any component is regulated by more than one regulator, but where one of them is enough to activate it. This kind of activity is considered as independent activation. In the case where the node is being regulated by more than two nodes, the transfer function will have a complicated form with a combination of operations that use "AND," "OR," and "NOT." The implementation of the Boolean network allows the representation of the transfer function in a logical truth table. This truth table shows the states for all network nodes and the results of the action of their regulators.

Finally, we mention that in some biological cases, it is hard to figure out the structure of the Boolean network since biological knowledge is not complete. In addition, the rules for transfer functions may not be fully known. In this case, the solution to the problem can be realised by creating several variants of the Boolean network and then compare the output responses with available observations from the real system. This can help with finding the missing part or aspect of a Boolean model (Davidich & Bornholdt, 2008). However, some biological systems or biological cases are poorly investigated. This kind of systems are hard to be represented by Boolean networks or translated into Boolean functions. Therefore, machine learning is considered as a solution to learn the topology or transfer functions of Boolean networks from available high throughput temporal data. On the other hand, this solution is also considered a key machine-learning problem(Lähdesmäki, Shmulevich, & Yli-Harja, 2003).

Moreover, in the cell cycle system, components can be represented by Boolean entities with binary values referred to as “OFF” and “ON.” A Boolean model can easily express the activities of a biological system, such as proteins, genes, and their Interactions. In a biological system, nodes represent proteins and genes, while interactions are represented by edges. The system activity is represented by Boolean functions, as shown in Figure 3.1. Figure 3.1.a. represents the structure of a simple regulatory Boolean network where “A,” “B” and “C” represent network nodes and the arrows denote activation and inhibition. Figure 3.1.b. shows the rules that manage the activities in this Boolean network system.

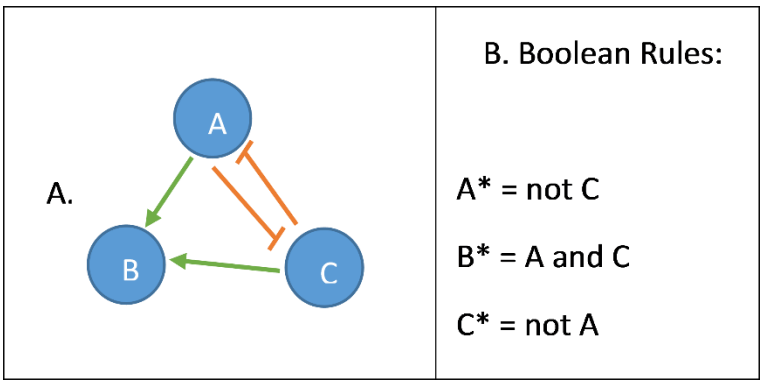


Figure 3.1: An example Boolean network: A. Boolean network structure, B. Boolean rules of the network. In the Network representation in (A): A, B and C represent the system’s components (network nodes). The directed edges —> and —| denote activation and inhibition, respectively. In (B), Boolean rules control the dynamic states of the nodes of the Boolean network in (A). Each state represents the node name, the * in the label denotes the next state of each node over time.

The reliability of Boolean modelling basically depends on the definition of the transition rules in Boolean functions, as this is a parameter-free modelling technique. For example, if the regulation of node B is considered, the corresponding Boolean function determines the future state of B from the present state of nodes A and C as shown in Figure 3.1 in a way that the presence of nodes A and C activates node B at the same time, as shown in Figure 3.1.

The transition from one state to another can be made by relying on an updating scheme, which can be either a synchronous or an asynchronous updating scheme. A model that uses a synchronous update is called a synchronous model. In this model, all nodes update

their states simultaneously in a single step depending on the last state of the system. In addition, this model treats the time for biological interactions in the same manner - it assumes that the time scales of all biological processes are similar. The synchronous model has an advantage that if we utilize the same initial conditions, it leads to the same attractor in replicate simulations; the reason behind this kind of behaviour is that the intermediate dynamic state sequence of the system is considered deterministic. In reality, biological processes differ in their time scales from fractions of seconds to hours: for example, protein phosphorylation, which is faster than protein synthesis or transcriptional regulation. However, in synchronous models, they assume that all system elements are temporally the same. Further details are discussed in the next sections.

Asynchronous models update their nodes in a nonsynchronous order with different time scales (Chaves, Albert, & Sontag, 2005). In this updating scheme every unit of time for each node is updated randomly once. In addition, due to the stochasticity of this kind of updating, the same initial condition can lead to different states with different attractors. However, this update technique can provide more intuitive and reasonable results if the time periods for the biological processes are known. Further details shall be discussed in the next sections.

3.1.3 Two Ways to Construct Boolean Networks

A Boolean network can be built in two common ways. The first is the knowledge based method, which depends on qualitative literature statements and current biological facts (Hopfensitz, Müssel, Maucher, & Kestler, 2013). This type of knowledge can be transferred into Boolean rules by experts (R. Albert & Othmer, 2003). For instance, the statement

"Gene 1 inhibits Gene 2"

can be stated as a Boolean rule with a time factor.

$$Gene2\ (t+1) = \neg Gene1\ (t).$$

The second method constructs a Boolean network by inferring the Boolean network from time series measurements such as gene expression data. This method uses algorithms that infer dependencies between biological elements, such as genes, by analysing changes in the expression of genes over time; for instance, using popular algorithms such as the Best-Fit Extension (Lähdesmäki et al., 2003) and REVEAL (Liang, Fuhrman, & Somogyi, 1998).

3.1.4 Types Of Boolean Networks

Essentially, a Boolean model can only represent changes in the local state of the proteins (nodes) in discrete time steps. Consequently, Boolean networks are classified into two main types mentioned previously - *Synchronous Boolean Network* (Kauffman, 1969) and *Asynchronous Boolean Network* (Harvey & Bossomaier, 1997; Thomas, 1991), and they are discussed in more detail here.

3.1.4.1 Synchronous Boolean Networks:

In synchronous Boolean networks, the assumption is that all nodes update their states simultaneously at each time step. Synchronous Boolean networks (Alberch, 1994; Kauffman, 1969) consist of a set of Boolean variables as shown in Eq.3.1:

$$X = (X1, \dots, Xn) \tag{3.1}$$

In addition, a set of transition functions, where one can be specified for each variable is defined as in Eq.3.2:

$$F = (f1, \dots, fn) \tag{3.2}$$

The nature of these functions is such that they can map input Boolean variables in X to a Boolean value.

In the implementation of a Boolean network, values of the variables are presented in qualitative form (0 or 1). The state of the network at time ***t*** is illustrated by a Boolean vector

(X(t) = (X1(t), ... , Xn(t)). The successor state X(t+1) is then calculated by stratifying the transition of the ***n***-dimensional synchronous state function F to x(t), as shown in Eqs.3.3 and 3.4:

$$\mathbf{X}(t+1) = F(\mathbf{X}(t)) \quad (3.3)$$

Where, $F = (f1, \dots, fn)$ and $Fi(\mathbf{x}) = fi(\mathbf{x})$ (3.4)

In this case, all network functions are exercised simultaneously. In biological situations, Boolean variable ***X*** identifies the state of genes or proteins in the network, and the transition functions explain and manage the corresponding relations between them.

3.1.4.2 Asynchronous Boolean networks:

Asynchronous Boolean networks (Harvey & Bossomaier, 1997; Thomas, 1973) are composed of a set of variables (***X***) and another set of transition functions (***F***) that are the same as the ones in the synchronous Boolean network. In asynchronous Boolean networks, the assumption is that at each point of time ***t***, only one of the transition functions $fi* \in F$ fires at random. The result is that one out of the number of potential Boolean variables updates its status at that time step giving the network output evolution a pseudo temporal quality.

An asynchronous update state can be explained by the transition function as shown in Eq. 3.5 (Harvey & Bossomaier, 1997; Thomas, 1973).

$$T_{async}^{(i*)} = \left(T_1^{(i*)}, \dots, T_n^{(i*)}\right) \quad (3.5)$$

Which is linked by the transition of gene *i**. This can be defined in the form of Eq.3.6 (Harvey & Bossomaier, 1997; Thomas, 1973):

$$T_i^{(i*)}(X) = \begin{cases} f_i(X) & i = i^* \\ x_i & i \neq i^* \end{cases} \quad (3.6)$$

Each gene ***i**** can have ***n*** different state transition functions. The Boolean network can calculate the successor state by choosing the function and applying it to the current state, i.e., Eq.3.7 (Harvey & Bossomaier, 1997; Thomas, 1973):

$$X(t+1) = T_{async}^{(i*)}(X(t)) \qquad (3.7)$$

3.1.5 Boolean attractors

This section explains Boolean attractors. Each Boolean network has a finite number of states, and the transition through all states would lead to either to a fixed point, limit cycle or chaotic attractor without a predictable trajectory. The fixed point defines a static steady state attractor comprising only one state. All transitions from this state result in the state itself. These kinds of attractors can be the same for both synchronous and asynchronous updates, as they are considered a special case of simple and complex cycle attractors (Hopfensitz et al., 2013), as shown in Figure 3.2C where the numbers indicate the state vector. A cycle attractor is a self-loop of states traversed over the time evolution of a network. This kind of attractor can explain the long-term oscillatory behaviour of system dynamics (Figure 3.2 A &B). Chaotic limit cycles traverse a set of states without an identifiable pattern of movements.

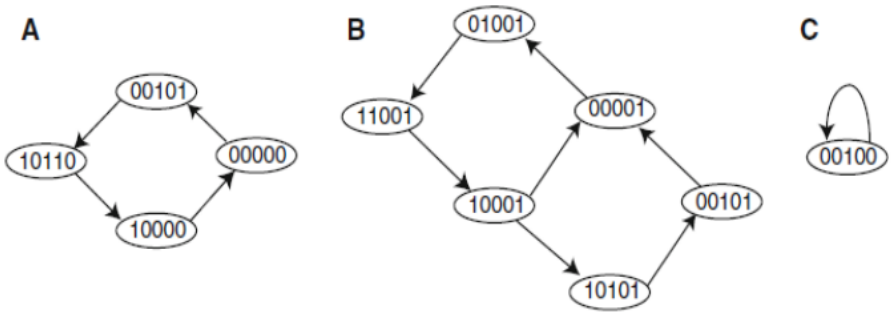


Figure 3.2: Boolean network attractors. A: A simple limit cycle in a synchronous Boolean network. B: A complex attractor in an asynchronous Boolean network. C. A steady-state attractor in a synchronous or asynchronous Boolean network.

Iterating the state transitions can easily lead to attractors in synchronous and asynchronous networks. The types of attractors differ depending on the network type. Self-loop attractors can be classified into several types depending on the behaviour of the attractor. The first type is the simple loop attractor (synchronous Boolean network), which comprises a set of states with exactly one successor, as shown in Figure 3.2A. The second type is the complex loop attractor (asynchronous Boolean network), where one state of a network can have more than one successor state in time progression as shown in Figure 3.2.B. The last of which is the steady-state attractor (in both synchronous and asynchronous Boolean Networks), as shown in Figure 3.2C.

On the whole, attractors can be a set of fixed points or closed trajectories (cycles) in the state space that is visited repeatedly during the time evolution of a network. These states may serve as an indication of achieving the goals of the system. A fixed-point attractor in a Boolean network is defined by a single state vector (Figure 3.2C) that is reached by many initial conditions of the system. In this case, the system can remain in this state until perturbation from external or other signals. In the case where the system has more than one fixed attractor, there exists a catchment of initial conditions which leads the system to a specific attractor.

Simple limit cycles or loops of states occur in Boolean networks when any state in the cycle is reached again after a fixed number of successive state transitions (the cycle length) (Figure 3.2 A). The synchronous and asynchronous Boolean network updates however can lead to differences in their attractors as shown in Figure 3.2 A &B. As stated earlier, in synchronous Boolean networks, such attractors are due to each state having one possible successor state (Figure 3.2 A). However, in asynchronous updates, there may be more than one successor state for a particular state leading to complex (loose) attractors (Figure 3.2 B).

Complex attractors occur in asynchronous update due to overlapping loops when there is the possibility of more than one successor state as a result of using this kind of update.

Complex attractors have a set of states, and the transitions of these states lead to other states in the same set and each set can be reached by all the other states in the set. Synchronous networks produce simple loops which may also appear in asynchronous network but this is not necessarily the case (Garg, Di Cara, Xenarios, Mendoza, & De Micheli, 2008).

Fundamentally, each Boolean network with ***n*** nodes should provide 2^n possible initial conditions. A Boolean system should eventually converge to a limited set of attractors. Both synchronous and asynchronous systems can have a steady state (fixed point attractors) or k-cycles (***k*** represents states that are repeated regularly.

Thus depending on the initial state of the network, a system model could develop a number of attractors. In regard to updating schemes in Boolean network models, synchronous updating scheme can provide simple loops for the long-term behaviour of the system. The attractors of asynchronous updating scheme thus can have one of these forms such as fixed point, simple loops, complex loops or chaotic limit cycles. Chaotic limit cycles traverse a set of states without an identifiable pattern of movements.

In addition, attractors have a basin of attraction. The basin of attraction represents all states that lead to a specific attractor. Usually, biological network attractors have a large basin of attraction (Huang, 1999; Li et al., 2004).

3.1.6 Using Boolean Attractors to Explain Biological Phenomena

Recently, Boolean models have become quite popular as they can be easily used for modelling large scale systems and can explain the dynamic behaviour of complex networks by using, for instance, attractors. Boolean network attractors are used to explain various biological phenomena. For instance, attractors are used to explain cell cycle events, such

as cell cycle phases; several studies have depended on attractors to elucidate cell cycle phases (Fauré et al., 2006) and Li et al. (2004).

In other studies, attractors are used to study wild-type gene-expression patterns; R. Albert and Othmer (2003) used companion attractors in a Boolean model of *Drosophila melanogaster* with wild-type gene-expression patterns of the segment polarity genes. Adult tissue cell states, such as growth, differentiation and apoptosis (programmed cell death), can be explained by attractors. Huang (1999) considered three states and showed that cycles of multiple states represented the growth of the cell, while single-state attractors represented differentiation or apoptosis. On the other hand, disruptions in cell state balances (attractors) are a characterisation of neoplasia (i.e. tumours).

Boolean models have also been used in investigations of diseases such as cancer. This kind of modelling can provide new insights into the origins of cancer. For instance, Sahin et al. (2009) used a Boolean network model to show single and multiple losses of protein function. In their models, they used single and double knockdowns of network elements before measuring their effect on the G1/S transition within the cell cycle process. Moreover, they studied the sensitivity and resistance of cells and how these cells responded to some of the perturbations. Sahin et al. (2009) discovered that MYC is a potential novel target protein in breast cancer cells (Orlando et al., 2008; Sahin et al., 2009).

In this proposed Boolean model, we studied the dynamic behaviour of the system using Boolean network attractors. Boolean network attractors usually represent the steady activation state of the biological components. In addition, it can represent cell phenotypes (Fauré et al., 2006; Li et al., 2004). Once we characterise the system attractors, we can study system dynamics and system robustness. Furthermore, we can examine the steady states of the elements by comparing them with the existing literature and to answer the questions below:

- **Can the model reproduce and reveal the observed importance of cyclins during cell cycle, with a visualisation of the activity of system components during cell cycle phases (system attractors)?**
- **Can the model reveal system dynamics of the whole cell cycle process?**

3.1.7 Search Algorithms for Boolean Attractors

There are different search algorithms for finding attractors in a Boolean network with synchronous and asynchronous update scheme. These algorithms depend on a number of factors including the size of the network. Boolean attractor search can be performed in an exhaustive manner or by using a heuristic that allows to starts from predefined states. These search algorithms are categorised into three modes in attractor search:

- **Exhaustive Synchronous State Space Search**

In this search, attractors can be recognised by calculating the successors of all 2^n states (where **n** is the number of nodes and the maximum number of nodes allowed for an exhaustive search is 29). In this method, synchronous state transitions are carried out from each of the possible states until an attractor is reached. This identifies all synchronous attractors. Furthermore, this algorithm can generate a full state transition graph.

- **Heuristic Synchronous State Space Search**

In this search, attractors can be recognised only by using a subset of possible states; depending on these states, synchronous transitions are carried out until an attractor is reached. This method is in contrast to the exhaustive synchronous search as it depends only on a subset of the possible states being used. This subset is defined using the *start states* method.

- **Exhaustive Synchronous SAT-Based Search**

In this search, the attractor search problem is solved using *Armin Biere's PicoSAT solver*. This method depends on searching for a satisfying assignment of a chain constructed by unfolding the transition relationships. This method is suitable for large-scale networks, where the size of network can be up to several hundreds of genes. For more details, see (Dubrova & Teslenko, 2011).

- **Heuristic Asynchronous Search**

This algorithm depends on the asynchronous state transitions, and it can also recognise steady-states and complex attractors, where these attractors represent a set of states, and these states from all probable asynchronous transitions lead to a state that is a member of the set as well.

We closely tested our model's functionality depending on the values of the initial conditions of the simulation. Two types of searches were used - exhaustive and heuristic space searches for synchronous and asynchronous updates. Further details can be found in the simulation section.

3.2 Procedure for Constructing Boolean Networks

Construction of any Boolean model for any biological network must follow six steps as shown in Figure 3.3. This represents the formal procedure for a good model structure. Our study will use this procedure because it can help us construct our new model effectively with updated elements.

The first step is to synthesise the network structure by collecting information and data from experiments and literature.

The second step is to determine the Boolean functions depending on the information obtained from literature and experimental observations. Each function defines the state of an item in the network.

The third step is to simulate the Boolean network scheme (*i.e.,* the transition from one system state to another over time). We have two updating schemes for Boolean networks: The first is synchronous updating scheme, where all the nodes in the model are updated simultaneously at each time step. Thus, this assumes that the time scales for all reactions and events in the biological system are similar. But, in real biological systems, the time scale for events are different and can range from fractions of a second to hours (Papin, Hunter, Palsson, & Subramaniam, 2005). Therefore, the synchronous scheme cannot describe the changes taking place in a real situation. The second updating scheme is the asynchronous updating scheme, where the system is updated in an asynchronous manner depending either on real time scales for an individual biological event or by allowing randomly-selected nodes to update themselves at each time step. The asynchronous update provides more realistic results as the update time is different for each step (Wang, Saadatpour, & Albert, 2012).

The fourth step is analysing the system output. In this step, the long-term behaviour of the system is studied and this can reveal the steady state (attractors) and the activity of the components.

In the fifth step, the correctness of the model is tested. A good model is expected to reproduce the previous experimental observations, such as cellular responses. With this step, we validate the model.

The final step is about biological implications and predictions. In this step, we investigate the network to study its long-term behaviours, loops, the rules of the network and various model perturbations.

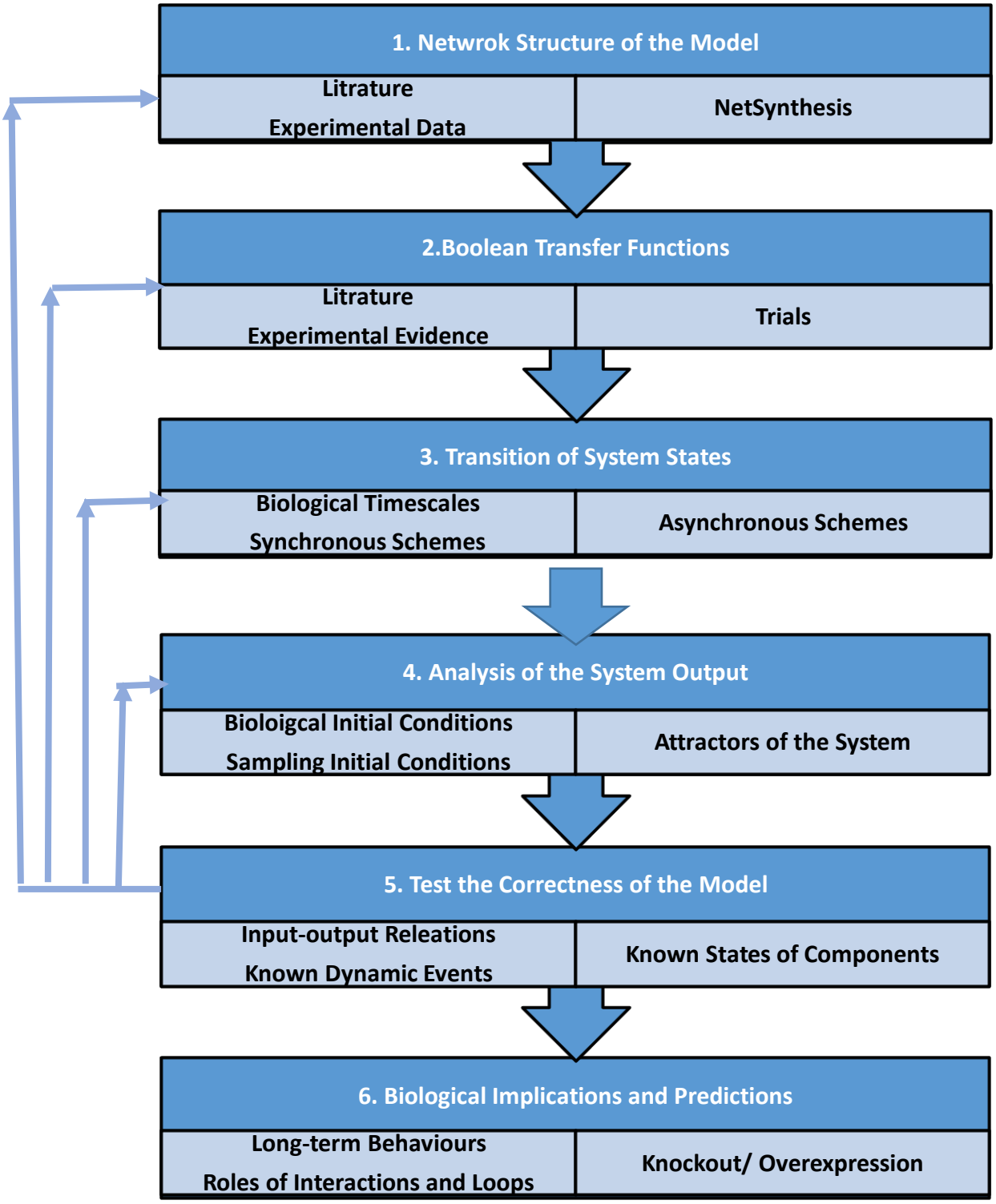


Figure 3.3: The main steps in the Boolean modelling of biological systems (Wang et al., 2012)

We can use specialised software to construct and simulate the model (Wang et al., 2012). A number of software programs are available to develop Boolean networks for biological systems; these include the BooleanNet (I. Albert, Thakar, Li, Zhang, & Albert, 2008), the

BoolNet (Müssel, Hopfensitz, & Kestler, 2010), the SimBoolNet (Zheng et al., 2010) and the ChemChains (Helikar & Rogers, 2009). We selected the BoolNet package to construct, study, analyse and simulate the Boolean network for the cell cycle controller we constructed in this study. Furthermore, testing and validating a Boolean model is essential for testing the hypotheses and assumptions used in the model and to assess its usefulness.

However, as discussed before, with this proposed Boolean model we are primarily interested in studying the system attractors as the attractors usually represent the biological phenomena arisen from the activity of the components.

3.3 The Proposed Boolean Model

As demonstrated above, our Boolean model can be built depending on the available knowledge. This kind of knowledge can help the modellers construct Boolean networks, as it supports the processes through the assembly of network components (nodes) and formulation of the logic of biological interactions found in the literature.

The model in this study is the cell cycle controller signalling network. In chapter 2, we described the cell cycle controller signalling network, highlighting the role of each member as has been experimentally discovered to date. In our study, we introduced several vital new components of the cell cycle controller-signalling network, primarily cyclins activators and degraders. Furthermore, this study covers the most important biological elements in cell cycle controller system, as it is based on a thorough investigation of the most popular existing models for mammalian cell cycles, such as the models made by Singhania et al. (2011) and (Fauré et al., 2006) that represent the most important regulatory elements in cell cycle systems and we expand/complement them with the newly added important elements that were missing in the previous models . Furthermore, this section provides abbreviated descriptions of the most important biological elements in our model.

As explained in literature review (Chapter 2), almost all current and recent models have investigated the simple yeast cell cycle, while the mammalian cell cycle has been less studied or modelled. In this section of the study, we relied on the most popular mammalian cell cycle models. These models had been created by (Fauré et al., 2006) and Singhania et al. (2011). Before we discuss the model we developed, we will investigate these prior models. However, in this thesis, we investigate the mammalian cell cycle system from both its biological and computational aspects using our developed model based on extensions to Faure 2006 model and Singhania 2011 model.

3.3.1 Faure's 2006 Model

To understand the behaviour of complex biological systems, such as the cell cycle system, we have to integrate the molecular data into a formal dynamical model (Fauré et al., 2006); this is the situation for the mammalian cell cycle. One of the most popular logical models is the one made by Fauré et al. (2006), who proposed a Boolean model for the mammalian cell cycle based on Bela Novak and John J Tyson (2004) ODE system to describe cell cycle behaviour. Bela Novak and John J Tyson (2004) model was based on 27 equations and experimentally-derived parameters. Fauré et al. (2006) model is still considered as a benchmark for all later Boolean models. The major contribution of the Fauré et al. (2006) model was the reduction in the number of equations and the focus on some important elements, including cyclins (D, E, A and B), Rb and E2F, as shown in Figure 3.4. However, in this model Cyclin D is considered as a cell growth signal receiver and a partial element without any Boolean rules. In addition, they, for convenience, assumed that Cyclin D is the starter for the cell cycle system and it should be active for the all cell cycle phases. Biologists have recently reported that MYC is the receiver of the cell growth signals in the mammalian cell cycle system. Therefore, in the model we developed, we covered these shortcomings by involving MYC. There are other additional elements, such as SCF and TFB, which control cyclin activities in our model. In addition, we systematically updated system networks, such as rules and interactions. Faure's 2006 model is based on biological

knowledge that was available at the time as found in Bela Novak and John J Tyson (2004) model, while we used more recent biological knowledge from various studies such as (Iwamoto et al., 2011). Enhancements in our model cover the abovementioned limitations by reconstructing a new Boolean network for the mammalian cell cycle controller system.

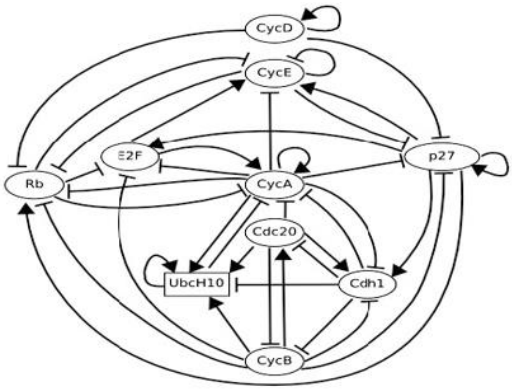


Figure 3.4: Faure et al. (2006) Boolean model for the mammalian cell cycle system (Fauré et al., 2006)

3.3.2 Singhania Et Al. (2011) Model

In hybrid models, modellers combine several methods to benefit from their advantages. The expected outcome from a hybrid model is to cover the shortcomings and issues in one method by another and, thus, provide a better understanding of complex systems. For instance, hybrid models can use both the discrete and continuous aspects of the same model to realistically represent the corresponding aspects in the system. Singhania et al. (2011) proposed a hybrid model to represent mammalian cell cycle signalling network regulators; specifically, their model was evaluated in terms of flow cytometry measurements of the cyclin proteins in mammalian – human cell lines. They relied on studying cyclin proteins to investigate cell cycle phases. They used their model to simulate the accumulation and degradation of cyclin proteins during cell proliferation. Their model was based on the best features of continuous differential equations, discrete Boolean networks and stochastic approaches as shown in Figure 3.5.

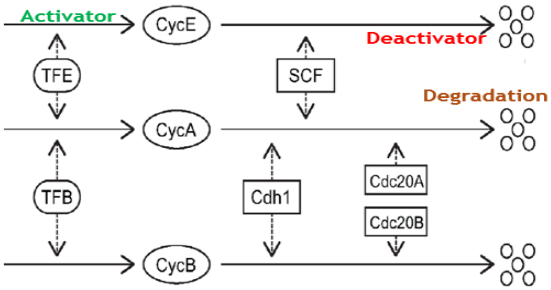


Figure 3.5: Singhania et al. (2011) hybrid model containing the most important regulatory elements in the cell cycle system.

The model represented the continuous change in cyclin synthesis and degradation (CycE, CycA, CycB), which were tracked by piecewise linear differential equations, while the regulators of cyclin synthesis (transcription factors (TFE and TFB) and ubiquitination machinery (SCF, Cdc20 and Cdh1) were represented by discrete variables (0 or 1). The variation in the state of discrete variables was based on a predetermined sequence. Therefore, they did not use any Boolean rules to control system activity. In addition, the transition between phases were determined, in part, by cyclin accumulation and degradation. This is a simple model which represents the most important regulatory elements in the cell cycle system. However, this model missed some other important regulatory elements such as CycD, Myc, Rb and P27. For instance, Myc is considered as growth factor receiver to initiate cell cycle processes. Further, CycD is an important regulator for G1 phase and can move the cell for G0 phase to G1 phase, and both RB and P27 have impacts on cell growth and cyclin activities. Rb controls transcription factor of CycE and p27 supresses CycE/Cdk2 and CycA/Cdk2 complexes.

The model we developed substantially enhanced Singhania et al. 2011 model by incorporating the aforementioned elements; especially by adding these elements to complement their interactions to ascertain new knowledge. This provides a full image about the system activity during cell cycle phases and more realistically simulates the accumulation and degradation of cyclin proteins during cell proliferation. The next section describes the specific enhancements made to the two abovementioned models.

3.3 The Proposed Model and Definition of Boolean Functions in the Cell Cycle Control Network

The model we obtained represents the cell cycle signalling network regulators. This study covers the most important biological elements in the cell cycle system. It includes items available from existing models of mammalian cell cycle, such as (Fauré et al., 2006; Singhania et al., 2011) models. These models represent the most important regulatory elements in the cell cycle system and describes cell cycle dynamics based on cyclin activities in cell cycle phases. However, in the previous model such as Singhania et al. (2011) they represented a small subset of the elements in discrete form; but without depending on or providing any Boolean rules or transfer functions that represent the interactions of system elements. This section provides abbreviated descriptions of the most important biological elements in our model. The following paragraph provides a description of each function in the cell cycle control model. Furthermore, we justify how we established specific logic gates to represent each of the regulatory mechanisms.

Our Boolean model studies the behaviour of cyclin production and degradation in cell cycle and control of transition between cell cycle phases, so we built our system rules relying on cyclins and their relationships with synthesis regulators (transcription factors and ubiquitination machinery) during cell cycle. Cyclin proteins have a substantial effect on the control of all the major events of the eukaryotic cell cycle. Cyclins are the regulatory components of cyclin-dependent kinases (CDKs) that control all the major events of the eukaryotic cell cycle. There are four important classes of cyclins in mammalian cells: D, E, A and B (Evans et al., 1983).

Cyclin concentrations change in a cyclical fashion during cell cycle. The fluctuations or oscillations of cyclins are made by variations in cyclin gene expression and degradation. Cyclin destruction happens through ubiquitin-mediated proteasome pathway; this promotes oscillations in CDK activity that leads the cell cycle process. Cyclins complex with CDK when binding with it; this event needs phosphorylation (Morgan, 2007). Cyclins are

classified into four classes named according to their role in various cell cycle stages: G1/S cyclins, S cyclins, G2 cyclins and M cyclins. These cyclin products were identified in the last chapter, but here we highlight the biological effect of each cyclin in cell cycle.

Basically, we assembled our Boolean rules systematically; each rule relied on the literature for the biological knowledge that was converted into a Boolean rule. We based it on recently available biological knowledge from sources such as (Iwamoto et al., 2011; Tashima, Hamada, Okamoto, & Hanai, 2008) systems to describe cell cycle behaviour, while Faure's 2006 model was based on prior reference knowledge from sources such as Bela Novak and John J Tyson (2004). Furthermore, we depended on the antagonism between elements such as (CycB and CDH1/APC), retinoblastoma protein (Rb), and the antagonism between P27 and Cyclins A and E to build the system rules.

In our Boolean model, each cyclin (CycD, CycE, CycA and CycB) and the other system components are represented by Boolean nodes. The relations and interactions between these nodes and other network elements are organised by Boolean rules. We updated the system's component rules one by one and, and each time, we did several experiments to validate the element's activity individually along with their impact on the system's global dynamic behaviour. Each Boolean rule consists of one or more Boolean functions, to represent the relations and interactions between system species/elements. Furthermore, in our model, we depended on the logical operators: "OR" gates to represent independent regulation (any input can regulate the target) and "AND" gates to represent the cumulative effect of more than one regulator. In addition, the operator "NOT" is used to perform the inhibitory regulations. A Boolean representation of the system elements can be formed with these operators and we discuss how Faure et al. (2006) model specifies these rules.

The previous sections explained how we reconstructed the cell cycle control signalling network. In this signalling network, we depended on Boolean functions to represent the biological reactions. For this reason, most realistic Boolean functions are required. These Boolean functions decide the "ON"/"OFF" states of nodes. In addition, setting up the initial conditions for the model with a correctly defined updating scheme is crucial.

The following sections provide the details of each key step we used to construct the Boolean model of the cell cycle controller signalling network. The following paragraphs provide a description of each function in the control model. Furthermore, they justify how we established specific logic gates to represent each of the regulatory mechanisms.

Biologically, each new born daughter cell starts its own cell-cycle when it receives growth factor signals. Upon receiving growth signals, synthesis of Cyclin D (or G1 Cyclin) is launched at the G0-G1 transition; this is called early G1 phase (A. Pardee, 1989). These signals make the cell size increase and unless it reaches a certain size threshold, the cell cycle is not initiated. It has been proven that a growth factor activates a transcription factor called c-Myc (receiver) that eventually promotes the synthesis of Cyclin D as shown in Figure 3.6. (Adhikary & Eilers, 2005; B. Alberts, 2008; Morgan, 2007). In our model, we assumed that a growth factor (GF) is available from early G1 and c-Myc is the most important link between GF and cell cycle machinery to trigger its initiation (by inducing the synthesis of Cyclin D). This means that following the presence of GF, c-Myc becomes activated and, subsequently, stimulates synthesis of Cyclin D as shown in Figure 3.6.

Figure 3.6: Growth factor signal activation of c-MYC leading to Cyclin D synthesis

Regarding Cyclin D (CycD) protein, in (Fauré et al., 2006) model, they assumed that CycD is an input for their model and considered it as a constant, either 0 or 1 denoting absence or presence.They used this node to mimic the presence of growths factors, so CycD keeps to state 0 when there are no growth factors and it becomes 1 when it senses growth factors; however, in our model, we expanded this aspect. We assumed that CycD involves other biological elements and interactions that can help demonstrate realistic biological behaviour of CycD. We know that CycD leads the G1 phase at the beginning of cell cycle.

So, we assumed that CycD becomes active when Myc becomes active. This means that CycD production process starts after the growth factors come and activate c-Myc, and then c-Myc promotes CycD synthesis (Massagué, 2004). However, there are a set of elements that have been discovered that inhibit CycD as shown in Figures 3.7a and 3.7b; for example, when SCF, CDC20A and CDC20B become active, they deactivate CycD and make it inactive (Alao, 2007). The main reason for this is that those elements (SCF, CDC20A and CDC20B) are E3 ubiquitin ligases that play crucial roles in degrading cell cycle proteins. In brief, CycD can be active only when the activator (c-Myc) is present and the degradation factor (!SCF & !CDC20B &! CDC20A) is absent as shown in Boolean functions (BF) 3.8.

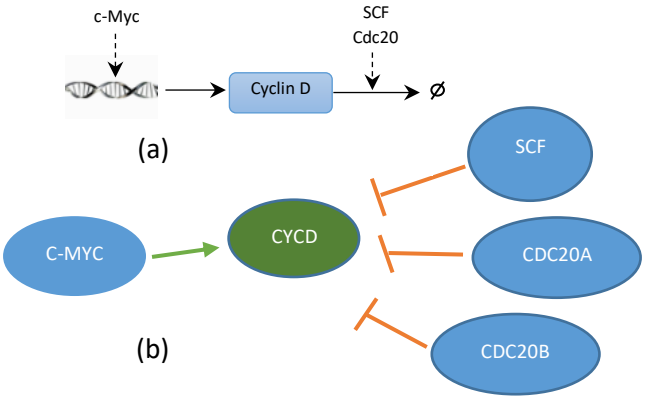


Figure 3.7: CycD activity controller. a) CycD production and degradation and b) CycD regulation in the Boolean Network model. The green node represents CycD while the other nodes represent CycD activator and deactivators. The directed edges —> or —| denote activation or inhibition, respectively.

CYCD, MYC & (!SCF & (!CDC20A & !CDC20B)) (3.8)

Cyclin E (CycE) is the second cyclin in the process. In our model, we assumed that CycE needs transcription factors such as TFE to start its synthesis early in the cell cycle (comparable to the E2F family of transcription factors) (Trimarchi & Lees, 2002). However, Cyclin E will stop accumulation if a degradation factor becomes present, such as ubiquitin-ligase complexes (APC-C and SCF). Here, we introduced SCF's impact on CycE in our model

(Ang & Harper, 2004). Moreover, P27 has a negative impact on Cyclin E so its presence lowers the levels of CycE as shown in Figures 3.8a and 3.8b (Chu, Hengst, & Slingerland, 2008). While in Faure et al. (2006) model, they assumed that CycE is controlled by the presence of E2F, with the assumption that E2F is already released from Rb/E2F complex which keeps E2F in check (Helin, 1998) they did not involve SCF in their model, which was discovered later. However, in this case, our assumptions for the Boolean functions that can organise Cyclin E activity can be used in the form shown in BF 3.9.

CYCE, TFE | (!SCF & !P27) (3.9)

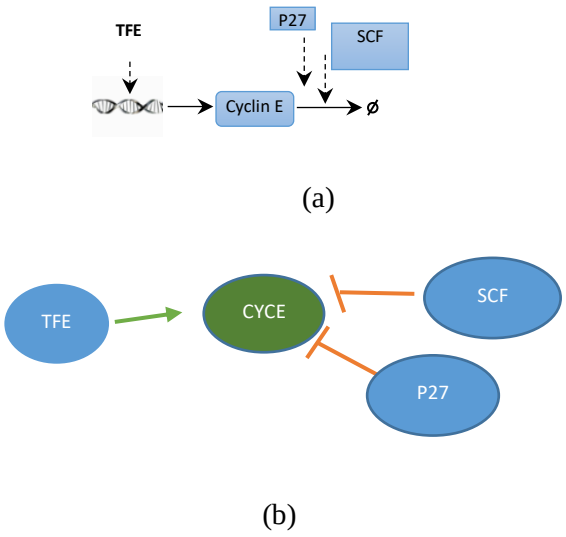
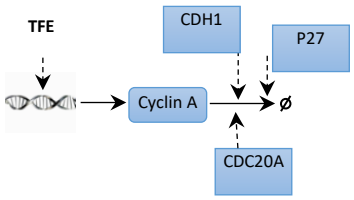


Figure 3.8: CycE activity controller. a) CycE production and degradation and b) CycE regulation in the Boolean network model. The green node represents CycE while the other nodes represent CycE activator and inhibitors. The directed edges —> or —| denote activation or inhibition, respectively.

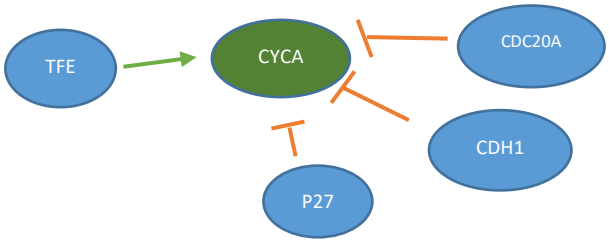
With regard to Cyclin A (CycA) protein in our model, we assumed that CycA needs transcription factors, such as TFE (comparable to the E2F family of transcription factors), as well starting its synthesis relatively early in the cell cycle. As with Cyclin E (Singhania et al., 2011), P27 can inhibit CycA activity (Chu et al., 2008). However, Cyclin A will accumulate until degradation factors or conditions become present. In order for Cyclin A to be degraded, two conditions must be present at the same time to stop CycA production and

start the degradation process. CDH1 and CDC20A can work simultaneously to inhibit Cyclin A and make it inactive ("OFF"), otherwise, CycA will still be active until the two degradation conditions are available (Singhania et al., 2011). While this protein's activity was managed in Faure et al (2006) model by assuming that CycA can be produced by E2F (Helin, 1998) and degraded by CDH1, the degradation of CycA can use CDC20 as well as a pair formed with CDH1 as shown in Figure 3.9. However, in this case, the Boolean functions that can organise CycA activity in our model have the form of BF 3.10.

CYCA, (TFE & !P27) | (!CDC20A & !CDH1) (3.10)



a.



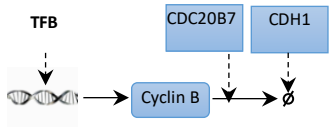
(b)

Figure 3.9: CycA activity controller. a) CycA production and degradation and b) CycA regulation in the Boolean network model. The green node represents CycA while the other nodes represent CycA producer and deactivators. The directed edges —> or —| denote activation or inhibition, respectively.

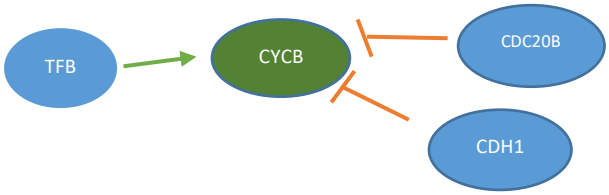
Concerning Cyclin B (CycB) protein in our model, basically, Fauré et al. (2006) model presupposes that the CycB protein can be active only in the absence of both CDC20 and CDH1, as they destroy CycB (Harper, Burton, & Solomon, 2002). In our model, we systematically built these rules following the processes that make CycB become active.

However, we also assumed that CycB required an additional element for its accumulation – another transcription factor such as TFB to initiate synthesis late in the cell cycle (comparable to FoxM1 and Myc) (Trimarchi & Lees, 2002). However, CycB will remain active until degradation factors or conditions become present. For Cyclin B level to decrease, two conditions must be present at the same time to halt Cyclin B production and start the degradation process. CDH1 and CDC20B can work together to inhibit Cyclin B and make it inactive ("OFF") (Harper et al., 2002; Singhania et al., 2011) as shown in Figure 3.10. In this case, the Boolean functions that organise Cyclin B activity can take the form of BF 3.11.

CYCB, TFB | (!CDH1 & !CDC20B) (3.11)



(a)



(b)

Figure 3.10: CycB activity controller. a) CycB production and degradation and b) CycB regulation in the Boolean network model. The green node represents CycB while the other nodes represent CycB producer and deactivator. The directed edges —> or —| denote activation or inhibition, respectively.

Moreover, in our proposed model, we have to represent transcription factors. A transcription factor is sometimes called a "sequence-specific DNA-binding factor." It is a protein that controls the rate of genetic information transcription from DNA to messenger RNA (Martirosyan, Figliuzzi, Marinari, & De Martino, 2016). Controlling the rate can be by

binding with other DNA sequences; this protein operates alone, or with other proteins in complex to activate or repress the recruitment of RNA polymerase (an enzyme that is responsible for executing the transcription process related to the movement of genetic information from DNA to RNA) (Lee & Young, 2000; Nikolov & Burley, 1997; Roeder, 1996).

In the cell cycle system, transcription factors participate in several tasks to control various processes, such as cell cycle regulation process. In addition, it helps in specifying what the cell size is and when the mitosis or division process happens in the mother cell (Meyyappan, Atadja, & Riabowol, 1996; Wheaton, Atadja, & Riabowol, 1996). In our proposed model, we have transcription factors, such as TFE, TFB and Myc.

TFE protein is a generic name for transcription factors for E-type cyclins in mammalian cells; namely, transcription factors of the E2F family. In our Boolean model, we relied on biological knowledge from the literature review to build TFE rules and conditions. TFE is active in the absence of CycA and CycB, which inhibits TFEs, such as E2F (B. Novak & J. J. Tyson, 2004). We share the same rules and assumptions with Faure et al. (2006) model for TFE activity shown in Figure 3.11. The Boolean functions which organise TFE activity can be represented in the form of BF 3.12.

TFE, (!RB & !CYCA & !CYCB) | (P27 & !RB & !CYCB) (3.12)

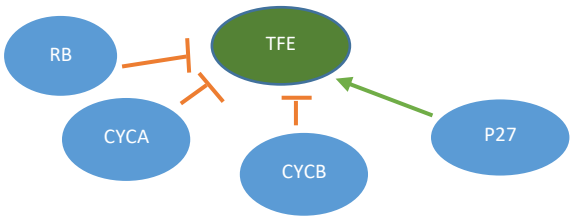


Figure 3.11: TFE Activity regulation in the Boolean network model, the green node represents TFE while the other nodes represent the TFE activator and deactivators. The directed edges —> or —| denote activation or inhibition respectively.

In our model, we have another transcription factor called TFB protein. TFB is a generic name for transcription factors for B-type cyclins in mammalian cells; namely, transcription factors, such as FoxM and Myc (Singhania et al., 2011). This element affects CycB activity by helping CycB production as shown in Figure 3.12. This element is one of a number of differences between our model and Faure et al. (2006) model, as they did not include this element in their model. We assumed that TFB can be active in the presence of CycA. The Boolean functions that organise TFB activity can be presented in the form of BF 3.13.

TFB, CYCA (3.13)

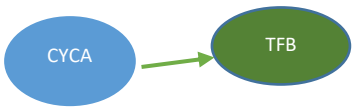


Figure 3.12: TFB activity regulation in the Boolean network model. The green node represents TFB while the other node represents TFB activator, CycA. The directed edge —> denotes activation.

An additional element is the Myc protein which can be found in our model and in biological studies by (I. Albert et al., 2008; Morgan, 2007). These researchers reported that Myc oncogene was one of the most important transcription factor in cell cycle. It takes a substantial part of the cell cycle processes, such as cell growth progression, programmed cell death (apoptosis) and cellular transformation. In our Boolean model, Myc is considered as an input and it can be constant; it can take either 0 when no growth factors have occurred, or 1 when growth factors have occurred and it is represented by BF 3.14.

So, in our model, Myc is the receiver of the growth factor signals; it can be “ON” when sensing the growth factors and is the starting point for our proposed Boolean network. When Myc receives growth factors, it fires the Boolean network which represents the cell cycle control system, as shown in Figure 3.13 and BF 3.14. We note that Faure et al. (2006) does not include this element in their model.

MYC = MYC (3.14)

Figure 3.13 Myc activity controller

We introduced another element, the SCF protein “SCF” = "Skp2-Cullin-Fbox"). This is an E3 ubiquitin ligase that plays a crucial role in degrading cell cycle proteins (Singhania et al., 2011). In addition, this element does not exist in Faure et al. (2006) model. This element has a crucial impact on cyclin activity, such as CycD and CycE. Therefore, our assumption for this element in our Boolean model is that SCF is active in the presence of CycE and inactive/absent in the presence of CDH1, since CDH1 can inhibit SCF activity (Alao, 2007), as shown in Figure 3.14. The Boolean functions which organise SCF activity are shown in BF 3.15.

SCF, !CDH1 & CYCE (3.15)

Figure 3.14: SCF activity regulation in the Boolean network model. The green node represents SCF while the other nodes represent SCF activator and deactivator. The directed edges —> or —| denote activation or inhibition, respectively.

Moving on to CDH1 and CDC20 (CDC20A and CDC20B) proteins, these are “auxiliary” proteins that combine with APC/C (anaphase-promoting complex/cyclosome, an E3 ubiquitin ligase) in order to direct its ubiquitin ligase activity towards specific substrates. The CDC20B-form of APC is active during anaphase when it causes the degradation of securin and Cyclin B (Singhania et al., 2011). The CDH1 form of APC is active during the G1

phase of cell cycle when it keeps Cyclin B levels very low. CDC20A form can cause Cyclin A degradation (Hilioti et al., 2001). Our assumptions agree with Faure et al. (2006) model for these proteins.

CDH1 depends on several conditions that work together to organise CDH1 activity and make it active. CDH1 is active in the absence of CycA and CycB and in the absence of CycE or in the presence of CDC20A and CDC20B, or by the presence of P27 and the absence of CycB. Otherwise, it will be inactive (Harper et al., 2002; B. Novak & J. J. Tyson, 2004), as shown in Figure 3.15. The Boolean functions that organise the CDH1 activity can be developed in the form of BF 3.16.

CDH1= (!CYCA & !CYCB) & !CYCE | (CDC20A & CDC20B)|(P27 &!CYCB) (3.16)

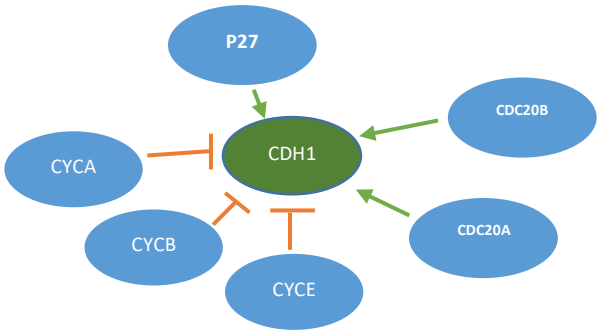


Figure 3.15: CDH1 activity regulation in the Boolean network model. The green node represents CDH1 while the other nodes represent CDH1 activators and deactivators. The directed edges —> or —| denote activation or inhibition, respectively.

CDC20A controls CycA activity and CDC20B controls CycB activity; we share this assumption with Singhania et al. (2011) model. But both CDC20A and CDC20B activities are managed depending on the presence of CycB (Harper et al., 2002; Singhania et al., 2011), as shown in Figure 3.16. The Boolean functions that organise CDC20A and CDC20B activity can be presented in the form of BF 3.17 and 3.18, respectively.

CDC20A = CYCB (3.17)

CDC20B = CYCB (3.18)

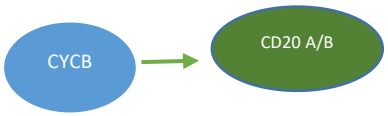


Figure 3.16: CDC20A/B activity regulation in the Boolean network model. The green node represents CDC20A or CDC20B while the other node represent the activators such as CycB. The directed edge —> denotes activation.

Our model involves a supplementary controlling element which is P27 protein. This protein keeps the cell in quiescent mode (G0), so it can be found at high levels during the early G1 phase. However, when growth factors are present, c-Myc activates CycD then the cell cycle is started and moved to new phase such as S phase which make P27 concentration decreases. These actions decrease the amount of P27 available and its activity, as shown in Figure 3.17. (Chu et al., 2008). In Faure et al. (2006) model, they assumed that P27 could become active only in the absence of a set of cyclins. We also depend on this knowledge to build the Boolean rules for P27 as shown in BF 3.19.

P27 = (!CYCD & (!CYCD & ! P27)) & (!CYCD & !CYCE) |(!CYCE & !CYCA) (3.19)

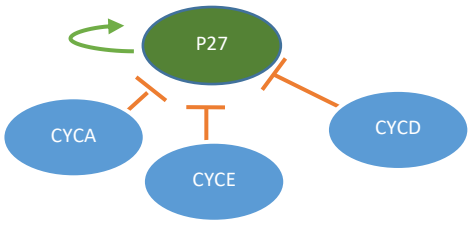


Figure 3.17: P27 activity regulation in the Boolean network model. The green node represents P27 while the other nodes represent P27 activator and deactivators. The directed edges —> or —| denote activation or inhibition, respectively.

An extra controlling element has been added to our model- Rb protein. For this protein, it has been assumed in Faure et al. (2006) model that Rb is expressed in the absence of a set of cyclins (B. Novak & J. J. Tyson, 2004). We assumed in our model the same for Rb activity

which requires CycD, CycE and CycA to be absent as shown in Figure 3.18 (Iwamoto et al., 2011). This invests our model with the ability to describe the transition process from late M phase to G0 phase and the transition from G0 to the early G1 phase; and the corresponding rules can be represented in the form of BF 3.20.

RB, !CYCD & (!CYCE & !CYCA) (3.20)

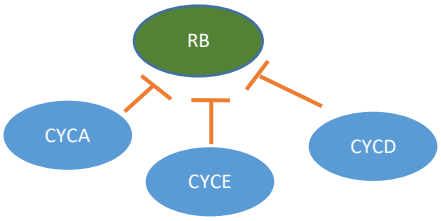


Figure 3.18: Rb activity regulation in the Boolean network model. The green node represents RB while the other nodes represent RB deactivators. The directed edges —| denote inhibition.

Finally, when all the above described system development processes are complete, we get the new mammalian cell cycle control system as shown in Figure 3.19.

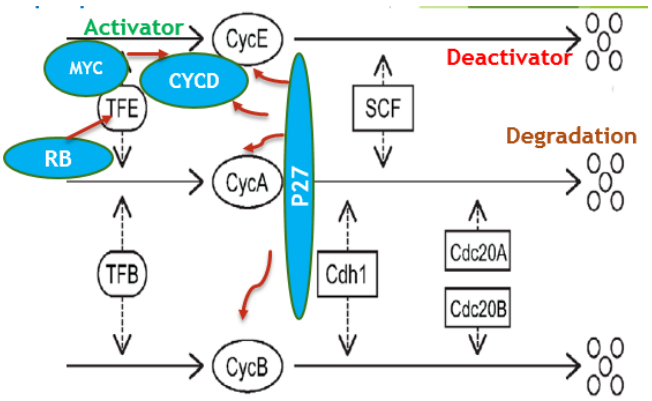


Figure 3.19: Proposed enhanced model of cell cycle controller (novel aspects are shown in blue). The red lines represent the new added impacts (activation or inhibition for the targeted node).

In addition, this novel model can be presented in Boolean form – computational and network interaction aspects of the model are as shown in Table 3.2 and Figure 3.20, respectively. We then use this model to answer following questions:

- **What extra insights can be gained by incorporating the new elements – or these elements in combination with others – in the network design?**
- **Can the model describe the temporal qualitative behaviour of the cell cycle system with greater accuracy or realism?**

These are important questions to raise and we provide worthwhile answers to these questions in this chapter.

Table 3.2 Proposed Boolean network with 13 proteins and their transfer functions

Protein	Transition Functions:
Myc	MYC = MYC
CDH1	CDH1= (!CYCA & !CYCB) & !CYCE | (CDC20A & CDC20B)|(P27 &!CYCB)
P27	P27 = (!CYCD & (!CYCD & ! P27)) & (!CYCD & !CYCE) |(!CYCE & !CYCA)
RB	RB = !CYCD & (!CYCE & !CYCA)
CYCD	CYCD = MYC & (!SCF & (!CDC20A & !CDC20B))
TFE	TFE = (!RB & !CYCA & !CYCB) | (P27 & !RB & !CYCB)
SCF	SCF = !CDH1 & CYCE
CYCE	CYCE = TFE |(!SCF & !P27)
CYCA	CYCA = (TFE & !P27) | (!CDC20A & !CDH1)
TFB	TFB = CYCA
CYCB	CYCB = TFB | (!CDH1 & !CDC20B)
CDC20A	CDC20A = CYCB
CDC20B	CDC20B = CYCB

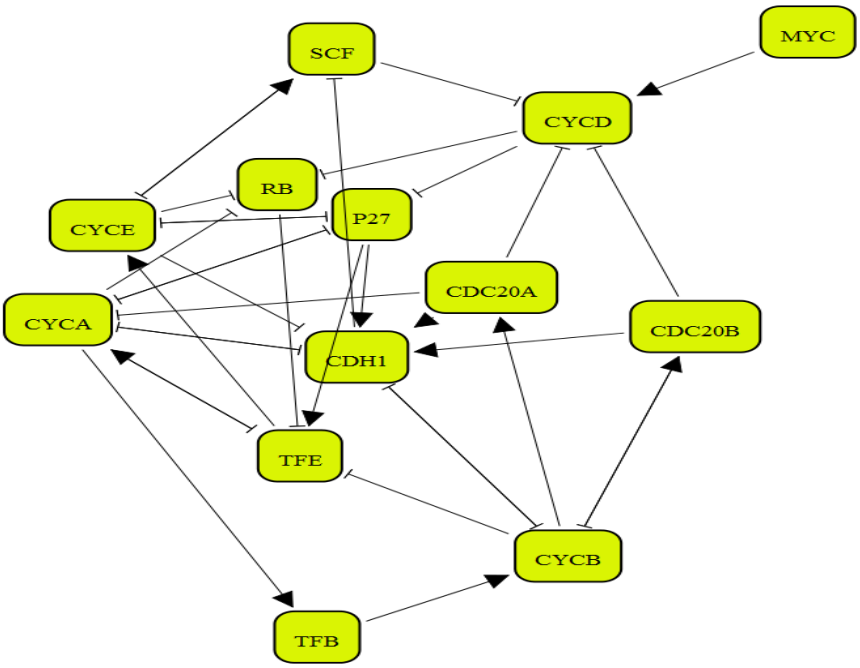


Figure 3.20: Proposed Boolean Model generated by BooleSim simulator (Matthias Bock, 2014). In this graph, each node represents a protein and double headed arrows with directed edges —> or —| denote activation or inhibition, respectively. We use double headed arrows to reduce the appearance of the number of arrows and to keep the graph simple and clear. In our model, we have around 43 connections among the 13 network nodes, which represent the biological interactions between system components

The proposed network is also used to probe more into model features, such as attractors, in cell cycle. In addition, the model can describe the temporal qualitative behaviour of the cell cycle system. It can also show how perturbations may change the system behaviour that can reveal system resilience and vulnerabilities as well as potential causes of cell cycle malfunction and potential avenues for recovery.

3.4 Model Simulation

We studied the system dynamics using the proposed Boolean network and we relied on some of the salient features of Boolean networks, such as Boolean attractors. Boolean network attractors usually represent steady activation states for biological components. In addition, it can represent specific cellular phenotypes (Fauré et al., 2006; Li et al., 2004).

Once we have characterised the system attractors, we can study the dynamics and robustness of the system. Furthermore, we can examine the steady states of the elements by comparing them with existing literature and answer the question:

- **Can the model properly explain steady state dynamics in cell cycle progression?**

Models need tools and software to transfer the theoretical ideas into the design of the model. Most logical models, such as Boolean networks, are built on a set of rules, hypotheses and assumptions from literature, and they attempt to represent and produce realistic dynamic behaviour. Much of the existing software available can be used to model the dynamic behaviour of biological systems, such as the BooleanNet (I. Albert et al., 2008), BoolNet (Müssel et al., 2010), SimBoolNet (Zheng et al., 2010), ChemChains (Helikar & Rogers, 2009) and Odefy package (Krumsiek, Pölsterl, Wittmann, & Theis, 2010).

Herein presents a full investigation of the cell cycle dynamics from the simulations of the proposed Boolean network. In the coming section of the chapter, we explain the methods used to determine the initial conditions for the simulations.

3.4.1 Determination of the Initial Conditions for the Model

Prior knowledge of the system can be very useful in model simulation; especially when there exist many unknowns or uncertainty. This can include the knowledge of the initial states of the system. In Boolean models, the concentration of each node is represented as a binary variable; each variable can have one of two qualitative states – active/present or inactive/absent states. Concentration or activity levels above a threshold indicate active state (1) and below the threshold indicate inactive state (0). The inactive state for any variable in the Boolean model did not mean that the variable was completely absent. It means that the concentration of the variable was below the threshold level.

In many cases, it is hard to have an adequate knowledge of the initial levels of each variable; therefore, the general approach in modelling Boolean networks is to use an

exhaustive search (Section 3.1.6). In this kind of update, all possible states in the network state space are treated as starting conditions.

However, in some cases, the states of some or the majority of the network nodes are known *a priori.* In this case, a need to employ heuristic search (Section 3.1.6.) arises to benefit from this knowledge in the analysis of the attractors and their basin (catchment of initial state vectors that flow into specific attractors). Sections 3.4.3 and 3.4.5 investigate the use of exhaustive search and heuristic search in synchronous and asynchronous updates.

3.4.2 System Dynamics of the Cell Signalling Network

In Chapter 2 and Section 3.3, we presented the development of the Boolean network and information related to the model, especially the biological knowledge. Our knowledge is based on current and recently available information from the experimental literature. The model developed consists of 13 proteins with 56 interactions. Each protein is represented by a node, while the interactions are represented by the edges as explained in Sections 3.1.1 and 3.3.3.

Simulating cyclin synthesis, degradation and cell cycle phases with our developed network required us to involve a set of elements. These elements include cyclins that lead the cell cycle phases, and the other elements that control protein production and destruction; these elements have been explained in Section 3.3. Our model consists of four sets of proteins as shown in Figure 3.21. The first set represents cyclin products, while the second set represents transcription factors required for protein synthesis. The third set represents protein degradation factors responsible for the destruction of the existing cyclins. The fourth set represents Rb and P27. These four sets work together as one to control cell cycle system and transition between phases.

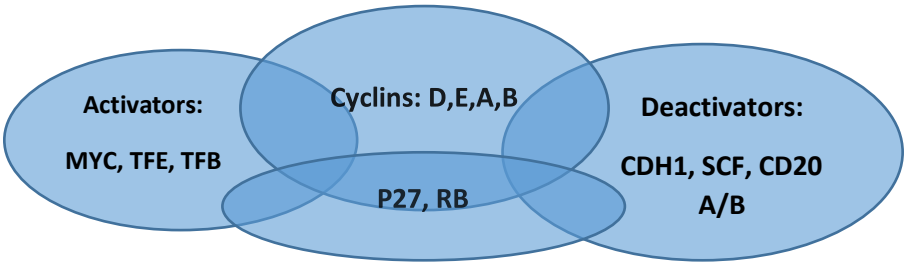


Figure 3.21: Cell cycle controller sub-sets

Regarding the timing for the interactions between system elements simulated in our Boolean model, the arrows represent the interaction between system components as shown in Figure 3.20; these interactions have different time scales and different binding constants and rates. However, in our model, which is similar to the models in literature, we are looking for model simplicity to represent the dynamic behaviour of the system. Therefore, in the first stage of model simulation with synchronous update, we assumed that all the nodes are the same – meaning that the activation and degradation of elements, which are considered fast events, are treated similar to production of proteins such as cyclins. Cyclin production is considered slow as protein synthesis takes hours. In asynchronous update, all the nodes have the same timing as well, but the randomisation feature in this type of update represents slow events in better way than synchronous update.

3.5 Modelling Dynamics of Cell Cycle Signalling – The Synchronous Model

The main objective of this study was to unravel the qualitative characteristics of the dynamic behaviour of cell cycle system. This investigation can be done using synchronous and asynchronous Boolean approaches. This is the initial part of our study, which aims to provide a significant qualitative study of global dynamics of the system. The results from the synchronous approach should be compared with current and existing models as these models are part of literature review and provided the base for model advancements in our study. The aim is to estimate how our results are compatible with other studies, such as the Singhania et al (2011) and Faure et al (2006) models. The comparison includes steady-state behaviour of attractors, and whether they represent simple or complex, limit cycle or

fixed point attractors, and comparing the attractors with Faure et al (2006) model attractors. It is also aimed to explore how the developed Boolean network reflect the global behaviour of cell cycle system; especially the transition from phase to phase.

To validate our results, we compared them with the results from Faure et al (2006) model. Therefore, we needed to simulate Faure et al (2006) model using both updating methods and obtain correct results. Then results from the two models were compared.

Synchronous Model Simulation

The synchronous Boolean approach depends on updating all elements (nodes) in the Boolean network simultaneously at each time point. The updating state for each node is controlled by Boolean rules that were defined previously in this chapter. Aspects of dynamical behaviour, such as attractors along with their basin of attraction, have been investigated using synchronous Boolean network models of cell cycle and other signalling processes (Dubrova & Teslenko, 2011; Fauré et al., 2006; Garg, 2009).

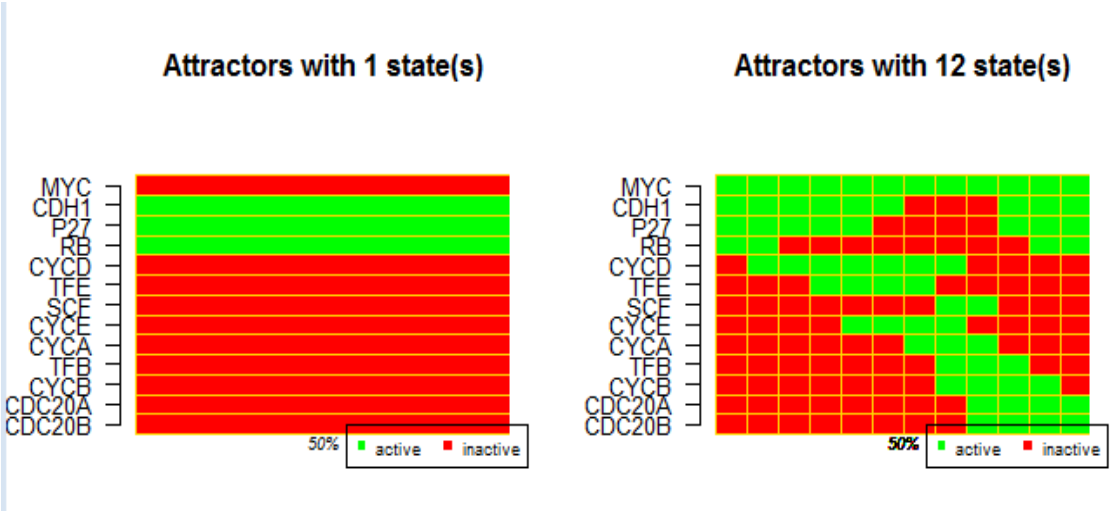
Our developed network contains a constant node (c-Myc) which represents the firing node. In addition, this node does not have any input links. Each node in the network must have one of two states, either active “ON” or inactive “OFF.” Each node must change its state between one of these two states. Concentration measurements can determine the activity of the element. Based on these an appropriate threshold value is determined; typically, concentration at the half activation level of the element is taken as the threshold beyond which the element is active; otherwise, the element is inactive.

The simulation of the network we developed and the investigations were conducted on an effective software package - BoolNet for R programming environment (Hopfensitz et al., 2013). We study the attractors in our proposed network dynamics using two different scenarios: the heuristic search and exhaustive search.

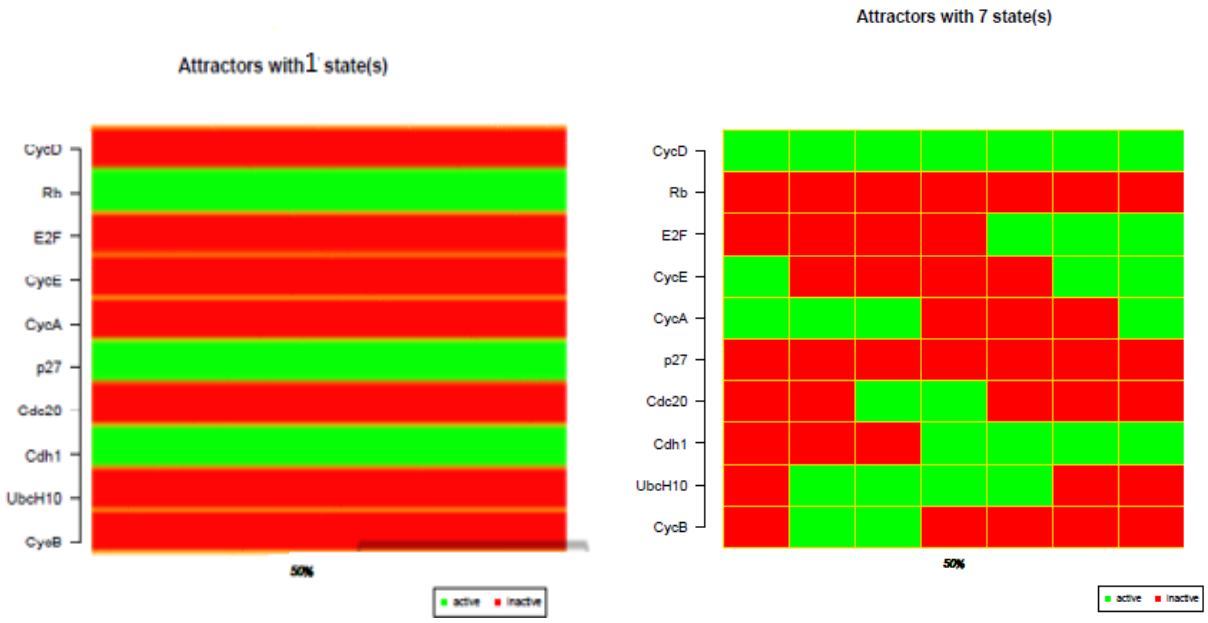
3.5.1 Exhaustive Search Scenario within the Synchronous Update

In order to assess whether our network can reproduce the same global biological behaviour of cell cycle system, we assumed here that we lack knowledge of initial states of the proteins. Therefore, we used all possible initial conditions where nodes were allowed to choose a random initial state for each simulation. The number of trial random initial conditions is 2^n, which equals 2^{13} = 8,192 where 13 is the number of network components.

According to the simulation results, the system completes the entire cell cycle phases with a complete description of the cyclin oscillations as shown in Figure 3.22a. The results showed that all 8,192 random initial conditions fall into two attractors. Half the random initial conditions falls automatically into Attractor 1 with one state (steady state) with 4,092 vector basin of attraction (Figure 3.22a left image). This attractor represents the quiescent (G0) state where there is no growth factor present to start cell cycle. We found with this attractor that only CDH1, P27 and RB were active and the other elements were inactive. This means that these active elements keep cells in G0 state waiting for growth factors signals and this is supported by existing biological knowledge.



(a)



(b)

Figure 3.22: Visualisation of state changes in attractors. a) The Boolean attractors from our model using 2^{13} random initial conditions and b) Boolean attractors from Faure et al. (2006) model using 2^{10} random initial conditions. Columns in the images represent consecutive states of attractors. At the bottom, percentage of states leading to the attractor is given; green cells represent active state while red cells represent inactive state. The simulation used Boolnet for R-language package.

The other half of the initial conditions produced a second limit cycle attractor with 12 states with 4,096 basin vectors (Figure 3.22a right image); this attractor represented remarkably correctly cell cycle phases with cyclin oscillations. This means that when the growth factor is present all these trajectories converge and are directed towards a unique dynamical limit cycle; this shows simplicity and potentially the advantage of synchronous updating. This attractor represented cell cycle phases depending on the changing states of cyclin products in the presence of growth factor – the oscillation of cyclins in the cell cycle phases and other cell cycle controller species activity. Our developed model shows the correct biological behaviour for cell cycle elements, especially cyclin activity. It showed how each cyclin leads and moves the cell cycle from phase to phase. Our model also properly described cyclin synthesis, degradation, and the activity of other system components over cell cycle phase transitions.

Moreover, the supplementary simulation used Faure et al. (2006) model and we simulated this model using 2^{10} random initial conditions with synchronous update. The results revealed two attractors (Figure 3.22b). The first was a steady state attractor that represented the G0 state in the absence of growth factors (Figure 3.22b left image). The second was a limit cycle attractor with seven states describing dynamic behaviour of cell cycle system. These results are reported in literature by Faure et al. (2006). However, Figure 3.22b shows that the cell cycle attractor in Faure et al. (2006) model does not show system activity properly. Cell cycle stages or phases are neither clear nor properly continuous. In addition, some of the element behaviours were not presented correctly, for instance, CycE, which became active at the beginning, then became silent, and then became active again at a later stage – this is contrary to CycE behaviour in cell cycle, where it becomes active just at the early stage of cell cycle phases. But CycE start its activity only late in G1 phase until the end of S phase; it helps the cell to move from G1 phase to S phase as shown in Figure 3.22a.

The dynamic trajectory of the system can also be observed from model simulation as a cell responds to a large number of different initial conditions. Figure 3.23 describes the transition states of the trajectories of our developed cell cycle signalling network over the simulation time. In the figure, each dot represents a state vector of the system at a specific time step during simulation.

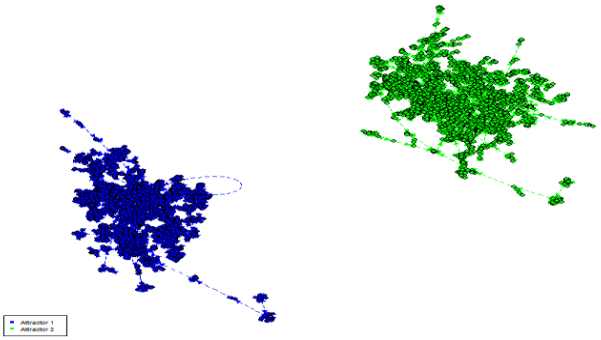
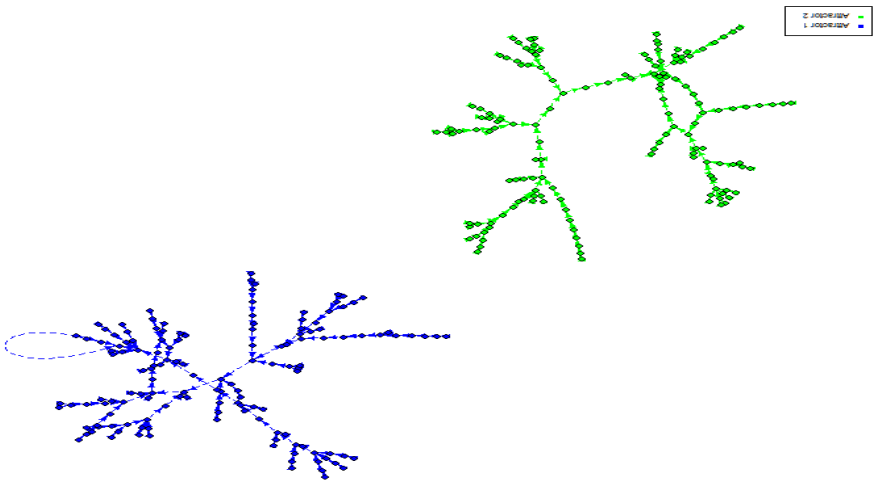


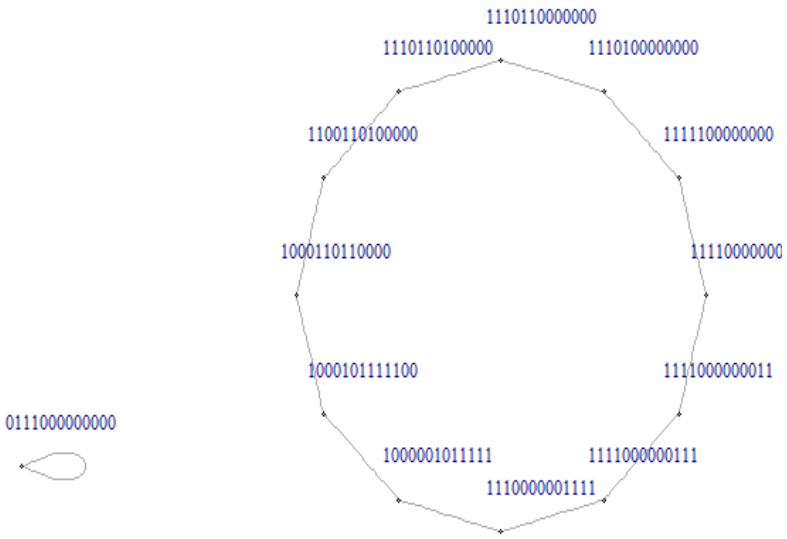
Figure 3.23: The trajectories of state transition from our model obtained by using 213 initial random conditions. The trajectories leading to the limit cycle attractor corresponding to cell cycle is

represented by the green graph while those leading to the fixed attractor representing G0 state is represented by the blue graph.

The state vectors are 13 digit binary numbers with values 0 or 1 corresponding to the states of the 13 nodes of the Boolean network. These trajectory graphs were generated using a force-directed graph-drawing algorithm from (Kawai, 1989).The simulation showed that the attractor states repeat themselves with every initial condition; this is a recurrent cycle of states (attractor). As it is difficult to ascertain attractor behaviour from all possible initial conditions as shown in Figure 3.23, we carried out another simulation using only 200 different random initial conditions and we still noted that half of them automatically fell into the same fixed-point attractor (steady state) as before and the other half of random initial conditions fell into the same previously found 12 state-limit cycle attractor (periodic self-loop) as shown in Figure 3.24a. Figure 3.24b shows the transition states of the system elements in each attractor over time; the state for each element is either 0 or 1.



(a)



(b)

Figure 3.24: Transition states obtained from 200 random initial conditions for system elements in each attractor over time. a) State transitions in limit cycle attractor representing cell cycle (green graph) and the state of the fixed-point attractor representing G0 state (blue graph); b) values of transition states for system elements in each attractor over time

3.5.2 Heuristic Search Scenario within the Synchronous Update

This method depends on the values of the initial conditions that can be known from previous knowledge. It is always preferable to start the simulation with relevant initial conditions if possible. In the literature, all sources of knowledge agree that early in the G1 phase, when growth factors are present, there are some elements that need to be and are active. For instance, CDH1 as well as P27 and Rb are active at the beginning of cell cycle (Sangahi, 2011). The other elements are inactive or absent. Accordingly, we started the simulation in our model with these initial conditions - Myc, CDH1, P27 and Rb in active form with all having values of 1. The synchronous Boolean approach depended on updating all elements (nodes) in the Boolean network simultaneously at each time point. The updating

state for each node was controlled by Boolean rules that were defined earlier in the last section.

The result of this scenario provided one attractor with 12 states (limit cycle) similar to the one found in exhaustive search; these states represent a recurrent cycle that explains the oscillation of cyclins and cell cycle phase transitions as shown in Table 3.3 and Figure 3.25. Here, state 1 shows the G0 state for the new born cell. The biological elements that are active in the G0 state are - Myc, CDH1, P27 and Rb. The next state demonstrates the production of protein CycD due its promoter Myc becoming active in the previous state. State 3 shows how CycD activity affects Rb by making it inactive. In the next state (time step 4), only one element becomes active, which is TFE - transcription factor for Cyclin E. In the next state (time step 5), the new element CycE is produced by TFE. Then in state 6, the model simulates the late G1 phase. State 6 mimics the transition from G1 phase to S phase (DNA synthesis phase); researchers call this the “boundary phase”; in this state, P27 is made silent by cyclins such as CycD and CycE. The next transition state, (time step 7), reveals that Cyclin A becomes active and the continuous production of CycE makes CDH1 inactive. State 8 represents the continuous production of CycE that activates SCF, but TFE then becomes silent. Moreover, CycA keeps its production and activity and other elements, such as TFB and Cyclin B, become active. The next state (time step 9), SCF remains active and inhibits CycE and make it silent. CycA, TFB, and CycB are also active. In addition, CycB activates CDC20A and CDC20B; in this state, CycD becomes inactive due to CDC20A, CDC20B and SCF being active which degrade CycD. All these activities represent the G2 phase.

Table 3.3 For known initial conditions, simulation results shows a limit cycle attractor similar to the one found in exhaustive search

	MYC	CDH1	P27	RB	CYCD	TFE	SCF	CYCE	CYCA	TFB	CYCB	CDC20A	CDC20B
1	1	1	1	1	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	0	0	0	0	0	0	0
4	1	1	1	0	1	1	0	0	0	0	0	0	0
5	1	1	1	0	1	1	0	1	0	0	0	0	0
6	1	1	0	0	1	1	0	1	0	0	0	0	0
7	1	0	0	0	1	1	0	1	1	0	0	0	0
8	1	0	0	0	1	0	1	1	1	1	1	0	0
9	1	0	0	0	0	0	1	0	1	1	1	1	1
10	1	1	1	0	0	0	0	0	0	1	1	1	1
11	1	1	1	1	0	0	0	0	0	0	1	1	1
12	1	1	1	1	0	0	0	0	0	0	0	1	1

Figure 3.25: Visualisation of the state changes in the cell cycle attractor. The columns represent consecutive states of the attractor. Green denotes active state ("ON") while red denotes inactive state ("OFF"). The visualisation reveals the changes in states of system elements in each phase over the simulation period of cell cycle.

The next state (time step 10) shows CDH1 becomes active again, this because CycA and CycB together inhibit CDH1 but this condition is no more presence since CycA become inactive by P27 in the previous state, also, CDC20A/B further activates CDH1. In addition, in this state, SCF becomes silent due to CDH1 being active and CycE is silent. However, TFB would still be active promoting CycB synthesis which keeps CDC20A and CDC20B active (the mitosis phase). In State 11 (time step 11), the system shows the events that occur at a later time in the mitosis phase. For instance, TFB becomes silent to reduce the amount of CycB production/activity, but Rb becomes active again in this phase. Finally, in the last 12^{th} time step, simulation shows that CycB becomes silent. This is because CDC20B and CDH1 works together to degrade CycB, and this represents the final state when the cell divides into daughter cells and moves into the G0 phase. This time step reveals transition from M phase to G0 phase in each new cell. The new cell now either enters cell cycle and follows the limit cycle attractor or stays in G0 phase until further notice.

This simulation results revealed realistic dynamics of cell division process in mammalian cell cycle. Both exhaustive and heuristic search based simulations show identical limit cycle attractor for cell cycle. Both scenario simulations revealed cell cycle phases and how cyclin

oscillations lead cell cycle. These results provided evidence that our model was robust even with different random initial conditions. This means that extensive perturbation to initial conditions can still sustain the limit cycle attractor in cell cycle which also points to the robust design of cell cycle. The synchronous simulation captured the main dynamics of the cell cycle system, such as cyclin localisation; these results agree with experimental literature (Fauré et al., 2006; Iwamoto et al., 2011; Singhania et al., 2011). They also agree with the results that were previously published in mathematical models (Ali Abroudi, 2017; Iwamoto et al., 2011). Precisely, our results are compatible with the biological knowledge that was discovered from models in literature, but these models in the literature provide only a part of cell cycle knowledge since each of them miss one or more elements, but our model provides new comprehensive biological knowledge concerned with cell cycle controller system.

However, this model is really useful in describing cell cycle behavior, but synchronous models cannot provide time delays between biological interactions since each change in state required only one time step; and all the nodes are treated in the same way. This makes a distinction between synchronous Boolean model results and actual cell cycle. Therefore, this model results have limitations and asynchronous updating can address these limitations.

The next section investigates asynchronous update scheme in detail.

3.6. Modelling Dynamics of Cell Cycle Signalling – Asynchronous Boolean Approach

In regard to synchronous updating scheme, it depended on applying Boolean functions for all elements simultaneously with respect to the previous state vector of the system. This idea is used to describe system dynamics. In addition, synchronous model was considered a deterministic model due to the possibility of the model starting at given initial conditions will always converge to the same end state after the same time steps (Kauffman, 1993).

Although the Boolean model with synchronous updating scheme showed realistic progression of cell cycle events, it could not capture the real temporal dynamics due to being unable to incorporate realistic time delays in the update schedule. Each event needs

suitable timing with the correct sequence of events. Therefore, asynchronous update was considered an alternative solution since it can introduce time delays for the interactions of system elements.

In this section, we discuss asynchronous update scheme that was used in the literature review and in our system, and show how it can enable representation of temporal dynamics of cell cycle in the Boolean network. We also investigate the system attractors from asynchronous update, explain time delays in our system, and compare results from asynchronous update experiments with the synchronous update model from the last section.

3.6.1 Asynchronous Updating Scheme for the Boolean Network

Generally, biological interactions that describe the dynamic behaviour of biological systems have time scales. The time scales associated with biological interactions can vary from fractions of a second to hours. For instance, signalling activities such as kinase/phosphatase reactions have time scales from fractions of a second to seconds, while transcriptional events and cellular growth can take hundreds of seconds or sometimes even longer (Papin et al., 2005). In particular biological scenarios, some elements/nodes in the biological network must have their states updated less frequently than others. These kinds of biological elements, represented by node(s), should not have a prompt response to biological signals; they may need longer times to complete their reactions. Therefore, asynchronous models have been developed to represent various time scales (Thomas, 1973), which add more variability to the update order. Later, various prominent asynchronous models have been added, such as the ones made by Chaves et al. (2005)); Harvey and Bossomaier (1997); Thakar, Pilione, Kirimanjeswara, Harvill, and Albert (2007) ; and R. Zhang et al. (2008).

Due to the limited availability of biological information, asynchronous updating scheme has three assumptions that any asynchronous Boolean network adheres to and they are:

I. In a single discrete time step, only one node can make a transition.
II. If the network has any element/node without any regulators that control its activities, this kind of node should remain in the identical state during simulation.

In asynchronous update, nodes are represented in the same way as in synchronous update. Any node in the system becomes either a 0/"OFF" or a 1/"ON". Changing their states depend on a set of Boolean functions. The main difference between synchronous and asynchronous update is that these Boolean functions turn nodes active with different time delays.

In some asynchronous updates, nodes become lined up in a specific order with a predetermined sequence for their updates where only one node changes its state at a given time in each update Saadatpour, Albert, and Albert (2010). However, in this part of the simulation, we randomly selected an element at each new time step to be updated. The reason for this choice, as mentioned before, is that available knowledge of time delays is limited. For instance, kinetic data that can help us specify the exact length of each state transition are scarce.

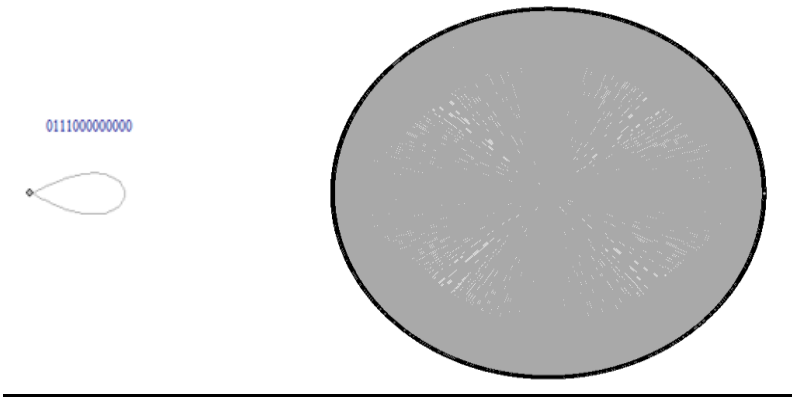
In this part of the research, we simulate our cell cycle signalling network with asynchronous updating method discussed above. We maintain the same network topology, Boolean transition functions, and the assumptions made in simulating the synchronous model, but a randomly selected node updates its state at each time step.

As in synchronous update, asynchronous model can also be simulated either with exhaustive search using 2^n random initial conditions or with heuristic search using known initial conditions from literature. We intend to utilise both to gain more knowledge.

3.6.2 Asynchronous Update – Exhaustive Search Scenario

In the previous simulation, we investigated the dynamic behaviour of our developed cell cycle system using the synchronous Boolean update. There we revealed that our developed model could be considered as an efficacious model to mimic cell cycle transition system even without time delays. Since the results and the network were robust, we use the results from synchronous analysis to compare with asynchronous results. Furthermore, we also simulate Fauré et al. (2006) model with the asynchronous update for comparison.

With the exhaustive search using 2^{13} random initial conditions, asynchronous simulation provided us with a very complicated result as shown in Figure 3.26; this made it difficult to interpret and investigate. The reason was that the states can be reached from other states. This can be a normal result since asynchronous update can provide complex attractors. Our results in Figure 3.26a showed that there could be two attractors; the first was a simple steady state attractor, which represented G0 state, and the second was a complicated attractor which contain cell cycle attractor. Further, Figure 3.26b shows how the states can be reached from other states in both steady attractor and complex attractor from our asynchronous Boolean model, as noted, this figure shows a small part of state transition from a huge and complex states transition that can be found inside the complex attractor. Hence, we simulated Faure et al. (2006) model with asynchronous update using 2^{10} random initial conditions. We gained two attractors that were compatible with our results, but the main difference was that there was greater clarity in the nature of the complex attractor due to the smaller number of initial conditions as shown in Figure 3.26c. The cell cycle complex attractor have a complicated order of reactions. Although randomised update has captured a version of time delays making it more realistic than synchronous update in concept the complexity of the attractors indicated that there needed to be a simpler approach or proper time delays integrated into the update.

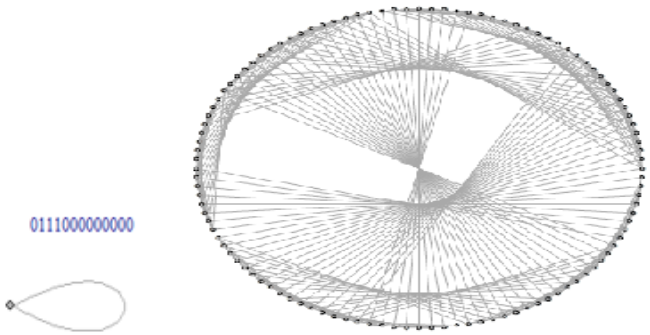


(a)

```
[[1]]
IGRAPH DN-- 1 1 --
+ attr: name (v/c)
+ edge (vertex names):
[1] 0111000000000->0111000000000

[[2]]
IGRAPH DN-- 3941 23483 --
+ attr: name (v/c)
+ edges (vertex names):
 [1] 1111111111111->1111111101111 1111111111111->1111110111111 1111111111111->1111101111111 1111111111111->1111011111111
 [5] 1111111111111->1110111111111 1111111111111->1101111111111 1011111111111->1011111101111 1011111111111->1011101111111
 [9] 1011111111111->1011011111111 1011111111111->1010111111111 1011111111111->1001111111111 1011111111111->1111111111111
[13] 1101111111111->1101110111111 1101111111111->1101101111111 1101111111111->1101011111111 1101111111111->1100111111111
[17] 1001111111111->1001101111111 1001111111111->1001011111111 1001111111111->1000111111111 1001111111111->1101111111111
[21] 1110111111111->1110111101111 1110111111111->1110110111111 1110111111111->1110101111111 1110111111111->1110011111111
[25] 1110111111111->1100111111111 1010111111111->1010111101111 1010111111111->1010101111111 1010111111111->1010011111111
[29] 1010111111111->1000111111111 1010111111111->1110111111111 1100111111111->1100110111111 1100111111111->1100101111111
+ ... omitted several edges
```

(B)



(c)

Figure 3.26: Representation of asynchronous attractors. a) The two attractors from our asynchronous Boolean model update obtained from 2^{13} random initial conditions; and b) the states transition inside the complex attractor from our asynchronous Boolean model; and c) The two attractors from Faure et al. (2006) asynchronous Boolean model update from 2^{10} random initial conditions.

Therefore, for simplicity of investigation and extracting more knowledge from the dynamic behaviour from asynchronous update, we decided to use heuristic search. The next section reveals details of heuristic search with asynchronous update.

3.6.3 Asynchronous Update – Heuristic Search Scenario

Here we selected the same known initial conditions that were used in the synchronous update for the heuristics search. In addition, for simulation and analysis, we used an open source Odefy Matlab package that helped simulate asynchronous update using specific initial conditions. The network revealed longer simulation cycles of 81 discrete time points, which is longer than total time steps in synchronous update. This is because the asynchronous attractors are expected to be longer than synchronous attractors due to randomised update of only one individual node at a time.

After running several experiments using asynchronous update, we obtained several different results. The variation in simulation results was due to the randomisation in the selection of nodes to be updated in asynchronous update. Therefore, we adopted the results most compatible with biological knowledge of the cell cycle. Consequently, the results from asynchronous analysis revealed realistic dynamic behaviour for cell cycle transition between phases. This was comparable with synchronous results, but in the asynchronous results, the time scale was longer than in the synchronous update. For instance, in the synchronous update the system components needed just 12 time steps to complete cell cycle with known initial conditions, whereas in asynchronous update, the cell cycle transition required approximately 81 time steps as shown in Figure 3.27.

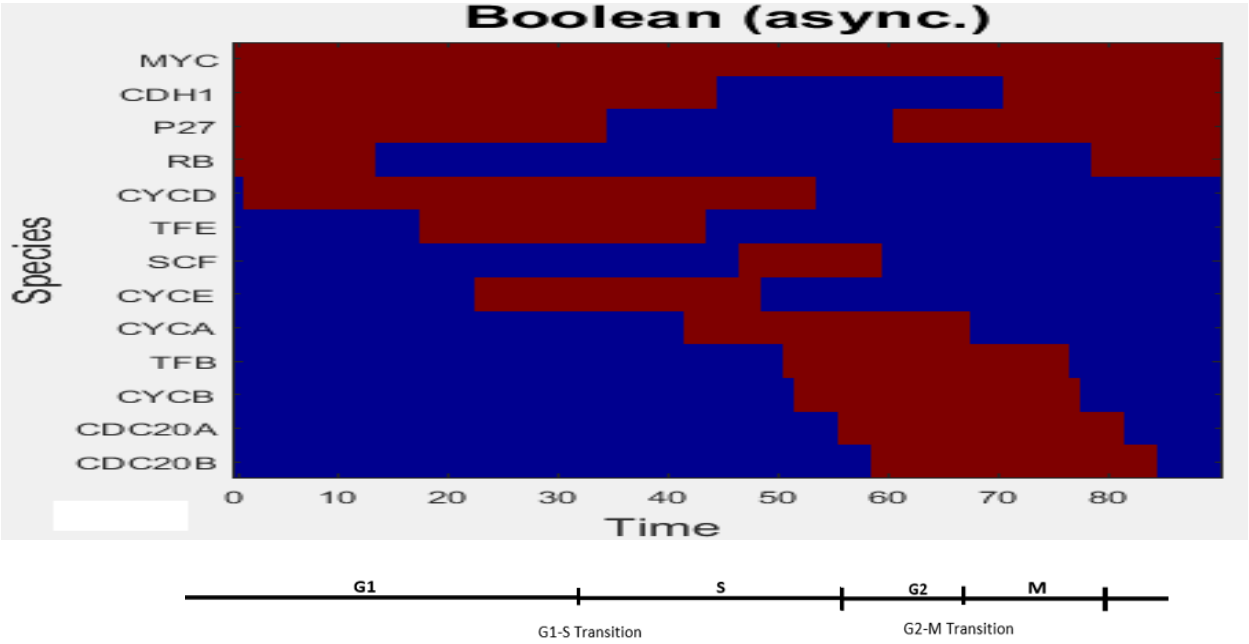


Figure 3.27: Visualisation of the state changes in attractor in asynchronous update. Red indicates active ("ON") states for individuals proteins over time, while blue indicates inactive ("OFF") States. The simulation was carried out on Odefy-Matlab open source program.

The results showed that majority of the network nodes were either in the "ON" or "OFF" state within the cycle of state dynamics. Similar results were observed in the synchronous Boolean model but with different time steps.

We observe that asynchronous update can describe cell cycle phase transitions well. In addition, it revealed correct biological interactions and cyclin oscillations. Moreover, asynchronous update showed time delays for biological interactions between system elements in a realistic way. For instance, TFE takes a major part in activating either Cyclin E or Cyclin A. In synchronous update simulation, we observed that when TFE became active Cyclin E became active in the next time step. However, in asynchronous update, the element's activity became different; when TFE became active, we noticed that time delays were observed before Cyclin E became active, and Cyclin A followed this regulatory behaviour as well.

The asynchronous Boolean model was considered to be a realistic model but needed prior knowledge to explain and investigate system behaviours. In the asynchronous model, each

simulation can provide a different result; this makes modellers take a long time to choose the most appropriate results. An alternative solution could be to use synchronous update while incorporating time variables to control the time of firing of the rules; this can provide realistic results. In the next section, we explain this method.

3.7. Synchronous Update of Boolean Model with a Time Variable

In this model, we implemented a synchronous update with time delays. When a node is turned active, different time delays (steps) are required to activate its targets. Updating scheme with time delays requires realistic knowledge of time delays in the absence of which random time delays become an option, but this kind of implementation is very computationally intensive and can lead to complex results, due to incorrect implementation of rules, that may or not provide useful insights. In this study, therefore, a constrained version of this updating scheme was used as a compromise.

We mainly applied discrete time delays for cyclins to become active since it is considered a slow biological reaction. Slow reactions of cyclin production require extra time steps and so a single time step is not enough as in synchronous update and random update may miss the target. For example, after becoming active TFE should produce CycE in a number time-steps and not after a single time step as in the normal synchronous update.

Furthermore, we used both exhaustive and heuristic approaches to generating initial conditions for the simulation. However, the exhaustive approach turned out to be very costly in terms of processing time as it required a high-performing device to handle the complexity of processing time. For instance, at the beginning, we ran the simulation with 2^{13} random initial conditions, and then discovered the timing issue, so we ran it with a smaller number of 100 random initial conditions. We obtained a set of attractors as shown in Figure 3.28; one of these attractors represented G0 state, and another attractor represented cell cycle system with 20 states. This attractor mimicked the time delays in biological interactions within cell cycle. For instance, the time delays obviously appear in this attractor; it shows system element activities during cell cycle phase transitions

properly - each cyclin has several time steps to be active in contrast to typical synchronous update which needs just one time step. In addition, this attractor explains how cyclin products lead cell cycle phases; this attractor is further explained in the heuristic search since heuristic search provides this attractor as well. The rest of the attractors were either spurious or attractors that can expose some missing or undiscovered knowledge or diseases such as cancer.

Figure 3.28: Attractors from synchronous update with time delays for cyclins showing visualisation of state changes within attractors. Columns represent consecutive states, green denotes active state ("ON") and red indicates inactive state ("OFF"). This visualisation reveals changes in system element states in each phase of the simulation.

Another attempt was undertaken by employing heuristic knowledge of initial conditions in simulation with time delays for cyclins. The same initial state (G0 state) that was used in the first synchronous and asynchronous update simulations was used again to run the simulation. It provided the same results for the attractor as was discovered by the exhaustive search with limited number of initial conditions as shown in Figure 3.29. This attractor showed cell cycle system activity; it also revealed the movement from phase to phase with time delays. For instance, in this attractor, state 1 represents the early G1 phase when growth factors are present, this makes Myc, CDH1, P27 and Rb active. In state 2,

CycD starts its activity and affects RB and makes it silent; in state 3 the early mentioned elements are still active and in state 4, only TFE becomes active and join other active elements. TFE will not activate CycE in the coming state, as CycE should wait few time steps to be active – thus we mimic the biological time delays between CycE and their activator. Therefore, in state 8, CycE becomes active. The next state shows that P27 becomes silent since several cyclins products become active which affects its activity. In state 10, CycA becomes active and CDH1 becomes silent. CycA waits several time steps to be active even though TFE has become active earlier and wait for P27 and CDH1 to become silent. In state 11, SCF becomes active and join all active elements, but TFE becomes silent and join other silent elements. In state 12, SCF does not silence CycE in the next state similar to typical synchronous update, CycE still active. Further, in this state TFB becomes active, TFB waits few time steps to be active although its activator CycA has become active earlier. However, in this state CycD becomes silent. State 13 has no any major or minor change in system element activity. In state 14, CycB becomes active. In state 15 CDC20A/B becomes active while CycE becomes silent, as we can note CycE become silent after several time steps even their inhibitor SCF earlier became active. In state 16, CycA becomes silent while CDH1 and P27 become active. State 17, Rb becomes active again. In state 18, TFB becomes silent while other active elements are still active. State 19 has no changes in system element activity. In the last state, CycB becomes silent. This state represents the end of mitosis phase. However, in this investigation we analysed this attractor from timing aspects since the biological conditions and relations are previously explained. As we can see almost all system elements have a time delay to become active in contrast to synchronous update requiring only one time step for state transition.

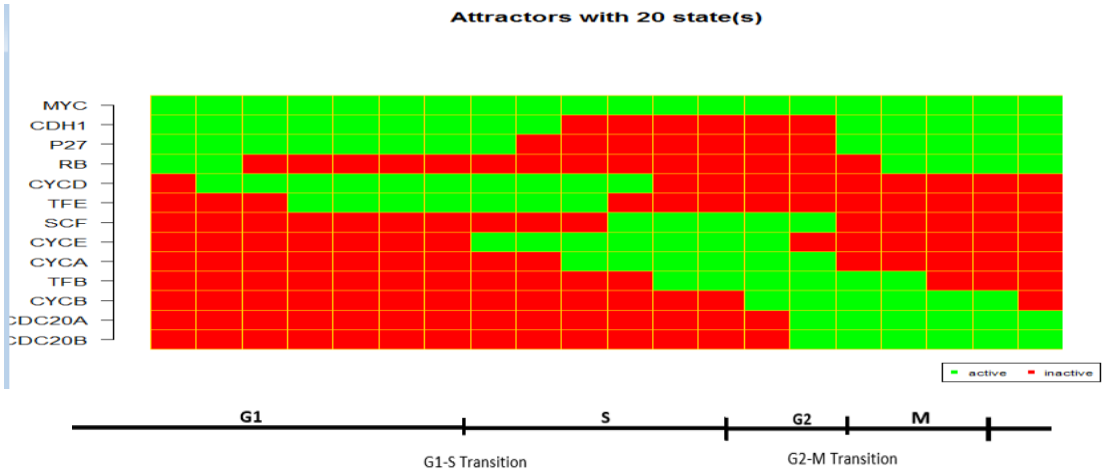


Figure 3.29: Attractor with 20 states from asynchronous update with time delays for cyclin production and heuristic initial conditions

Considering the results from the 6 analyses (synchronous, asynchronous update (random) and specified synchronous time delays) for exhaustively or heuristically selected initial conditions, we conclude that our Boolean model of cell cycle (i) produces realistic cell cycle stages in all cases, (ii) provides a greater level of temporal resolution and reality to cell cycle stages with the introduction of time delays for cyclins in asynchronous update compared to other updates, (iii) reveals that the system is robust against perturbations to initial conditions as evidenced by the similarity of results for random initial conditions (which represent perturbations to the actual initial conditions) and heuristic (known) initial conditions for all scenarios. The robustness is further investigated in the next section.

3.8 Analysis of Time Evolution of Network Events

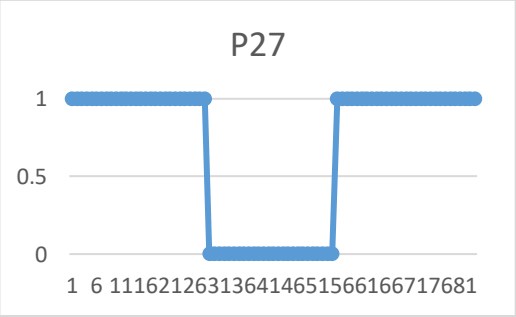
Previously we presented results from various simulations on our system to highlight attractors, cell cycle stages that are invariant to update method and initial conditions and robustness of the system to initial conditions. Here, we reveal more results and additional investigations related to other important aspects such as time evolution of cell cycle systems in synchronous and asynchronous updates.

3.8.1 Analysis of Time Evolution of Network Events – Asynchronous Update

While asynchronous update with a large number of random initial conditions was difficult to analyse, we chose the exhaustive search nonetheless. We chose the same known initial conditions used in the case of synchronous update with heuristic initial conditions.

The network showed simulation cycles of 81 discrete time points for heuristic initial conditions. The total time steps in this simulation was longer than that in synchronous update. This was because the asynchronous update with randomly updating one element at a time was expected to be longer than the synchronous update where all elements are updated simultaneously. We found a number of different results for different initial conditions due to random selection of elements for updating. For this investigation we selected the results that most realistically represent cell cycle phases.

Figure 3.30 shows the temporal evolution of the state of individual elements and the time required for their activation or inactivation over 81 time steps of asynchronous update. It presents cyclins such as Cyclin D, E, A and B that control cell cycle stages. Additionally, it shows Rb and P27 that control cyclin activities. Further, it presents transcription factors needed for cyclin syntheses, such as TFE and TFB. Finally, it reveals the behaviour of cyclin degraders CDC02A, CDC20B and CDH1.



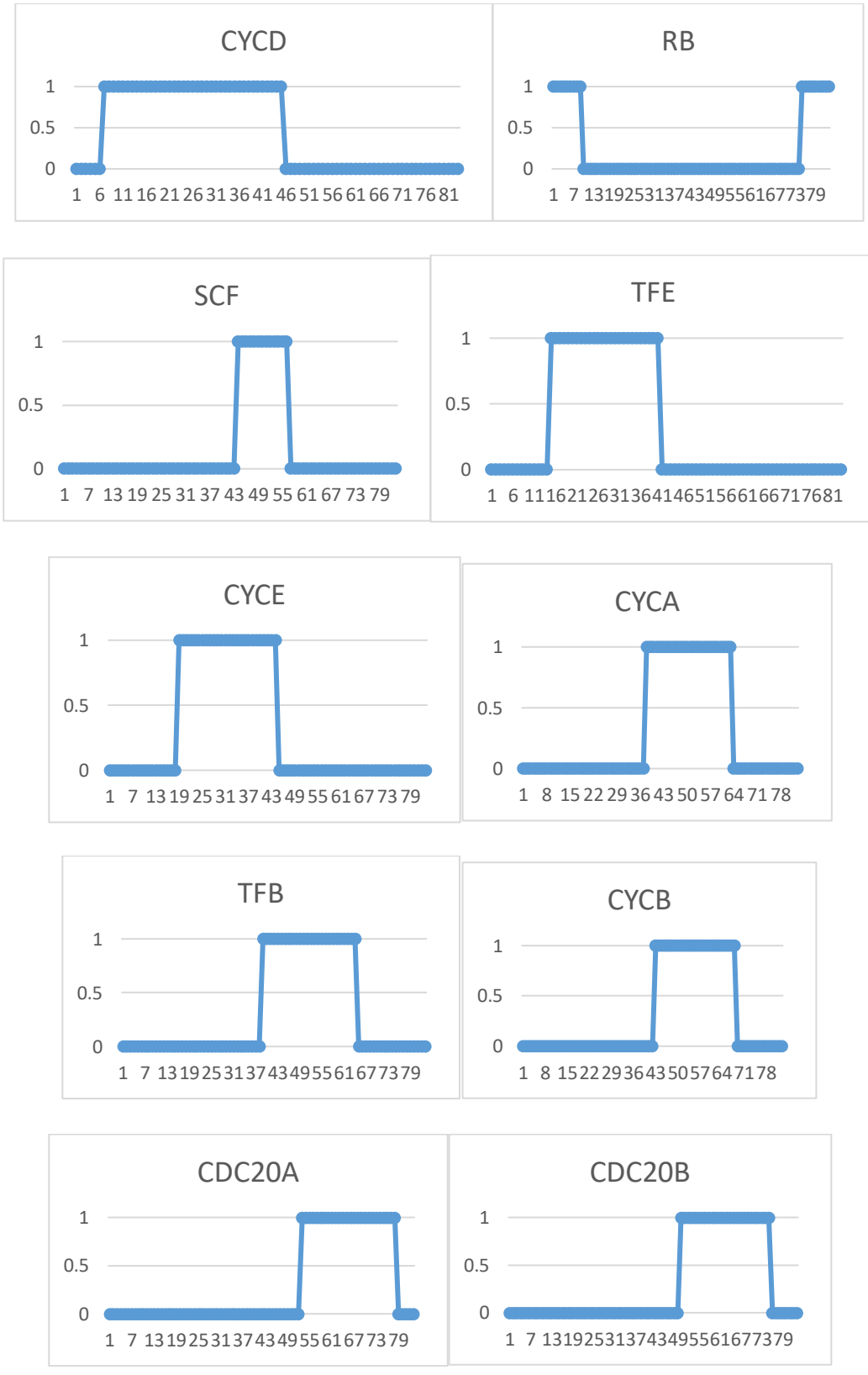


Figure 3.30: Temporal evolution of the state of individual elements and the required time steps for their activation/inactivation in asynchronous update.

We noticed from the previous figures that asynchronous update provided more realistic behaviour than synchronous update. This is because asynchronous update incorporates time delays for system activities, such as cyclins activators, inhibitors.

The asynchronous update takes more than one time step to activate or inhibit the interactions of system components. For instance, Figure 3.31 represents the controlling process for CycD. CycD is one of the cyclins that should be available early in the cell cycle. Therefore, CycD should be active sometime after Myc becomes active. Figure 3.30 shows that when Myc becomes active after a few time steps at the beginning CycD becomes active as well. This leads the cell cycle transition into G1 phase. CycD activity can be reduced and eliminated when its degraders become active, *i.e.,* SCF and CDC20A/B. The asynchronous update showed that more than one time step is needed to inhibit CycD by SCF and CDC20BA/B when they become active. These were the most important activations and inhibitions that managed CycD activity. Consequently, we can summarise this process by saying that CycD becomes active in time step 5, several time steps after Myc is activated. The inhibitor SCF becomes active in time step 43, and it takes more than one time step to affect CycD activity; CycD becomes silent in time step 47. CDC20A and CDC20B become active later, in time steps 52 and 51, respectively. These kept CycD silent until the end of cell cycle. Thus asynchronous update provides realistic descriptions of temporal evolution of CycD and the elements affecting its activity, relative to each other. How these timings compare with actual timing of cell cycle events need further investigation.

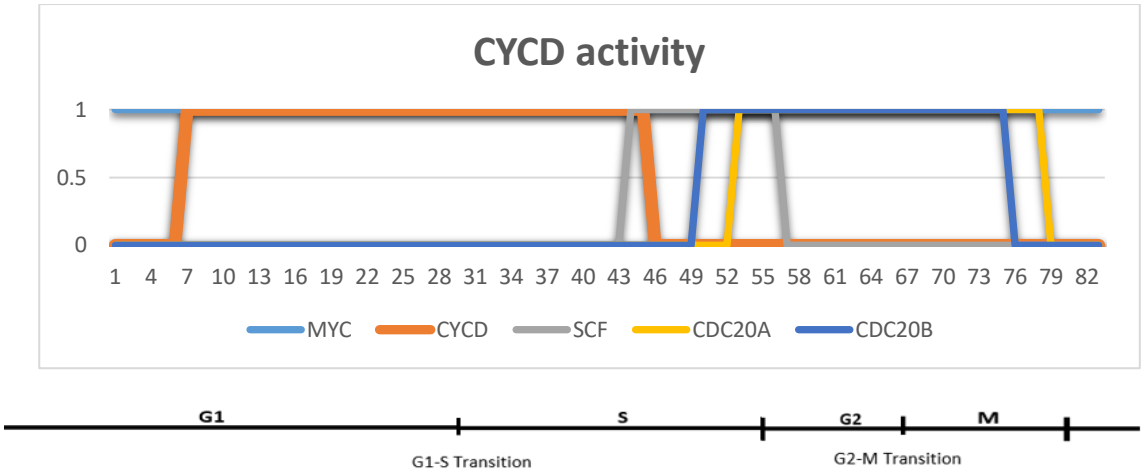


Figure 3.31: Temporal evolution of CycD and its controllers in asynchronous update

Figure 3.32 presents temporal evolution of CycE activity in relation to its activators and inhibitors. For instance, TFE activates CycE after several time steps and inhibitors also require several time steps to inhibit CycE activity. For example, TFE becomes active in time step 15 and activates CycE in time step 19. P27 and SCF also took more than one time step to affect CycE activity, while synchronous update required only one time step for this, either through activation or inhibition. For example, SCF becomes active in time step 44 and impacts CycE in time step 46. P27 then becomes active in time step 55 to keep CycE silent until the end of cell cycle.

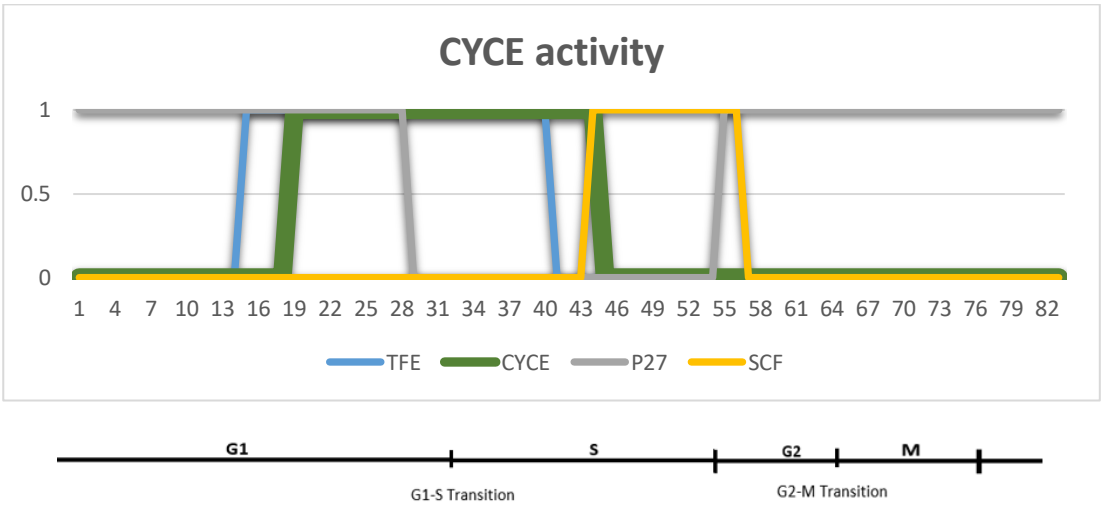


Figure 3.32: Temporal evolution of CycE and its controllers in asynchronous update

Likewise, CycA activity is appropriately mapped by synchronous update. CycA is the most important regulator of S phase of cell cycle, and it should be available in S phase. It works with other cyclins to move the cell from S phase to G2 phase (the second gap phase). Consequently, CycA should be active sometime after TFE becomes active on time step 15; this TFE first synthesises CycE prior to synthesis of CycA. Further, CycA activity is inhibited by P27 and CDH1. The asynchronous update demonstrated the relative time delays between these events as shown in Figure 3.33. For instance, P27 becomes inactivated in time step 29 (see also Figure 3.32) and later CDH1 becomes inactive in time step 35. Therefore, after several time steps CycA becomes active in time step 38 and this leads to cell cycle transition in G2 phase. However, CycA activity goes down when its degraders CDC20A and CDH1 become active. At time step 52, CDC20A becomes active and then CDH1 is activated in time step 61. After a few time steps from here CycA gets degraded. These activations and inhabitations control CycA movement. Figure 3.32 characterises the processes of CycA control showing the time delays between the activators and inhibitors.

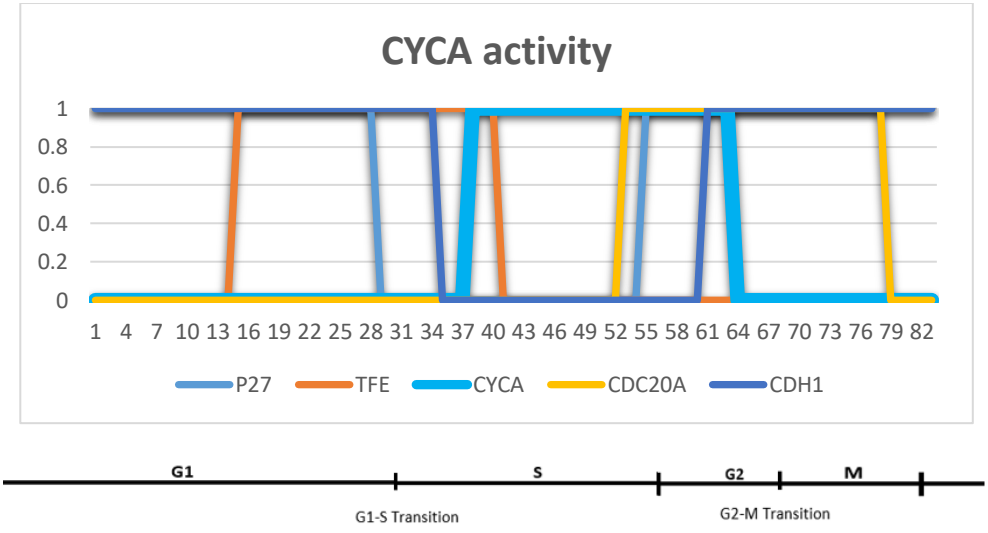


Figure 3.33: Temporal evolution of CycA and its controllers in asynchronous update

CycB is the main regulator in M phase; it should be active after TFB becomes active since it synthesises CycB. So when TFB becomes active CycB also becomes active after a delay of few time steps, as shown in Figure 3.34. Moreover, the asynchronous update revealed the time delays for CycB inhibition by CDC20B and CDH1. CycB is inhibited by CDC20B and CDH1

when they work together to inhibit it. The CDC20B inhibitor becomes active in time step 51, and later on in time step 62, CDH1 becomes active, followed by CycB inactivation in time step 69, which was inhibited by both CDC20B and CDH1.

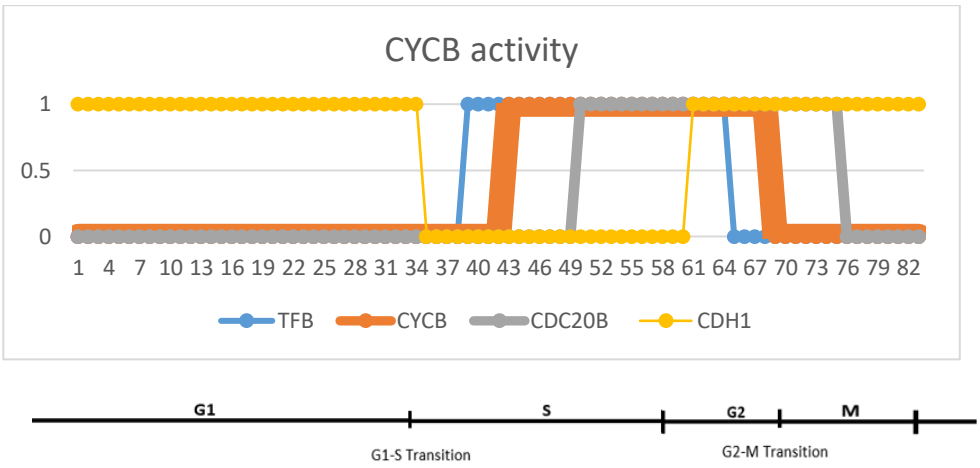
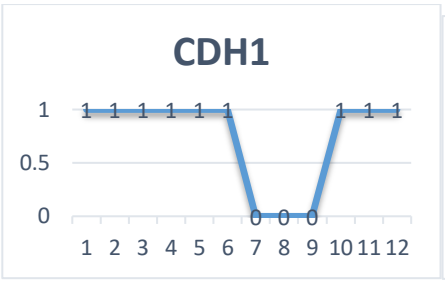
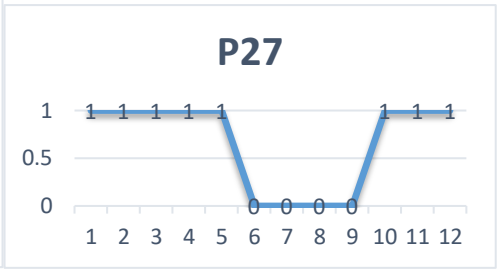


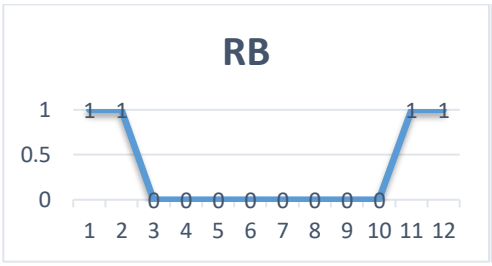
Figure 3.34: Temporal evolution of CycB and its controllers in asynchronous update

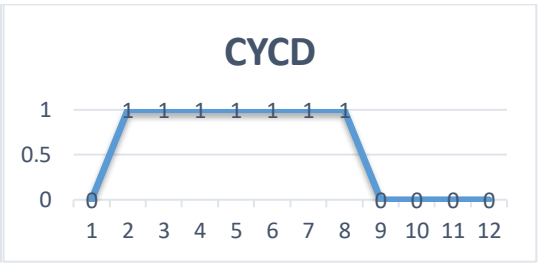
3.8.2 Time Evolution of Network Events in the Synchronous Update

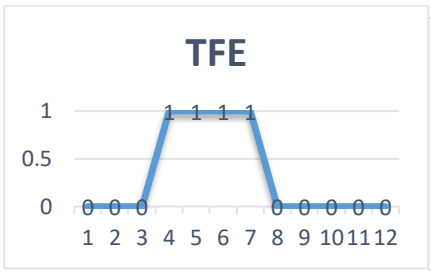
As demonstrated previously, the core functionality of our model was to explain the dynamic behaviour of cyclins and other parts of the cell cycle controller system. Therefore, we mainly focused on the behaviour of cyclins since each cyclin leads a phase in the cell cycle. To simplify our investigation, we used known initial conditions from biological literature.. Figure 3.35 demonstrates the temporal variability in the activity of a set of elements that represents the major functional events in the cell cycle system. It shows individual activations/inhibitions and the corresponding required time steps for each element during the simulation. The simulation timeline is 12 time steps for the synchronous update.

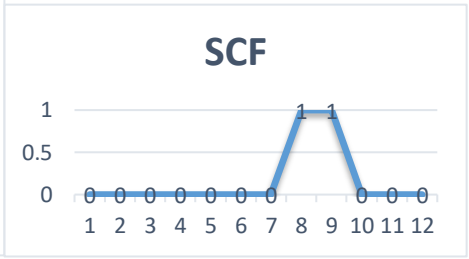

CDH1
1
0.5
0
1 1 1 1 1 1 0 0 0 1 1 1
1 2 3 4 5 6 7 8 9 10 11 12
Time step

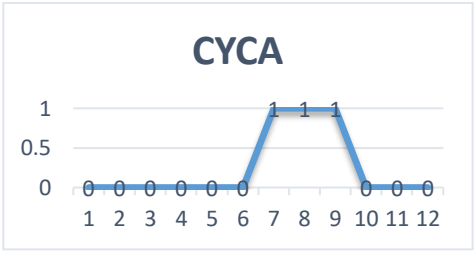

P27
1
0.5
0
1 1 1 1 1 0 0 0 0 1 1 1
1 2 3 4 5 6 7 8 9 10 11 12
Time step

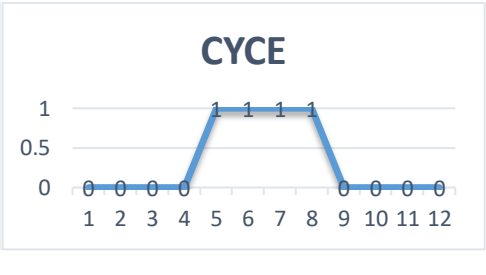

RB
1
0.5
0
1 1 0 0 0 0 0 0 0 1 1
1 2 3 4 5 6 7 8 9 10 11 12
Time step

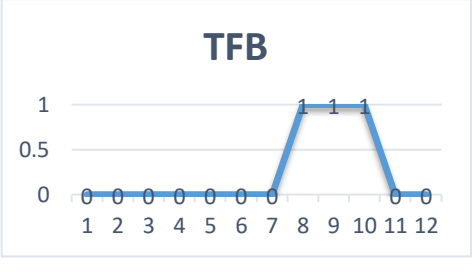

CYCD
1
0.5
0
0 1 1 1 1 1 1 1 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step

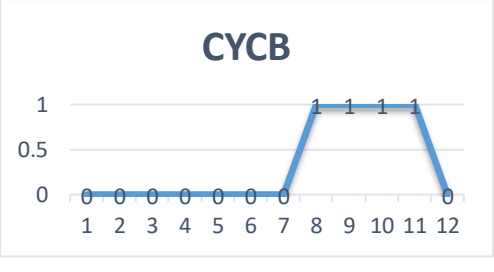

TFE
1
0.5
0
0 0 0 1 1 1 1 0 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step


SCF
1
0.5
0
0 0 0 0 0 0 0 1 1 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step


CYCA
1
0.5
0
0 0 0 0 0 0 1 1 1 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step


CYCE
1
0.5
0
0 0 0 0 1 1 1 1 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step


TFB
1
0.5
0
0 0 0 0 0 0 0 1 1 1 0 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step


CYCB
1
0.5
0
0 0 0 0 0 0 0 1 1 1 1 0
1 2 3 4 5 6 7 8 9 10 11 12
Time step

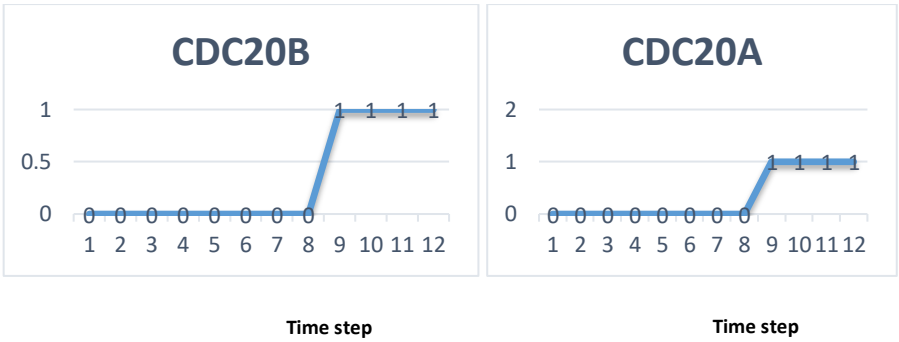


Figure 3.35: Individual activations/inhibitions and the corresponding required time for each element in the synchronous model.

We also explored cyclin control activity in the synchronous update. Figure 3.36 represents the controlling process for CycD. As described earlier, CycD should be available early in the cell cycle when the Myc becomes active. Figure 3.36 shows that when Myc becomes active at time step 1, CycD becomes active and leads the cell cycle transition into G1 phase in the next time step 2. CycD activity can be reduced and eliminated when its degraders SCF and CDC20A/B become active. At time step 8, SCF becomes active and then CDC20A/B become active as well at time step 9; therefore, they inhibit CycD activity at time step 9. CDC20A/B both keep CycD silent for the rest of the cell cycle. These were the most important activations and inhabitations processes managed by the CycD activity.

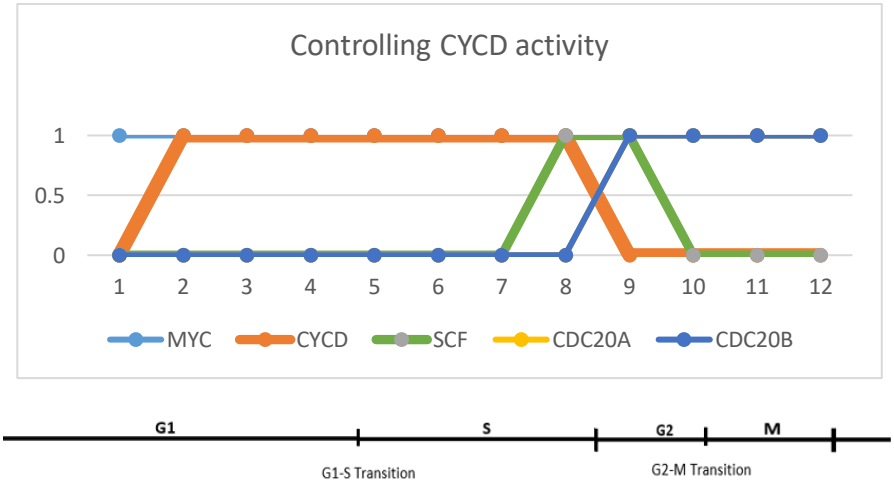


Figure 3.36: Temporal evolution of CycD and its controllers in synchronous update

Figure 3.37 characterises the controlling process for CycE which is the controller of G1 phase of cell cycle. As such, it should be available early in the cell cycle. As demonstrated earlier, CycE works with CycD to move the cell from G1 to S phase. Therefore, CycE should be active when its transcription factor TFE becomes active. Figure 3.37 shows that when TFE becomes active at time step 4, then at the next 5th time step CycE becomes active and leads the cell cycle transition into S phase. However, CycE activity can be reduced and eliminated when its degrader and inhibitor SCF and P27 become active. At time step 8, SCF becomes active and then at time step 9 CycE activity is inhibited. Moreover, P27 becomes active at time step 10 to keep CycE at bay for the rest of cycle. These activation and inhibition processes manage CycE activity as needed to enable it to control cell cycle.

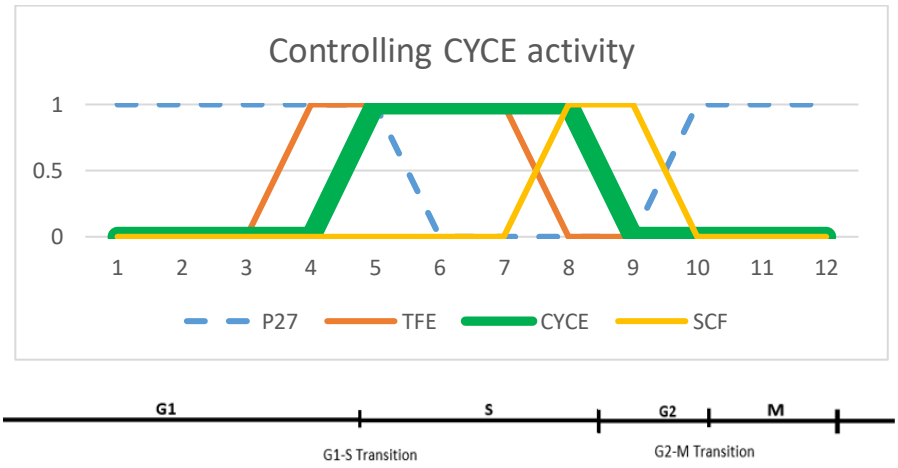


Figure 3.37: Temporal evolution of CycE and its controllers in synchronous update

Figure 3.38 characterises the control process for CycA, the main controller of S phase and works with other cyclins to move the cell from phase S to G2 phase, which is the second gap phase. As such, it should be available in S and G2 phases. Consequently, CycA should be active when its transcription factor TFE becomes as shown in Figure 3.38. Moreover, CycA activity requires another two factors P27 and CDH1 to be silent. P27 becomes silent at time step 6 while CDH1 becomes silent at time step 7. After that, CycA becomes active and leads the cell cycle transition into G2 phase at time step 7. However, CycA activity can be inhibited when its degraders CDC20A and CDH1 become active. At time step 9, CDC20A becomes active and then at time step 10 CycA activity is inhibited. CDH1 also becomes

active at time step 10 to keep CycA silent for the rest of the cell cycle. These activation and inhibition processes manage the movement of CycA to enable it to carry out its function.

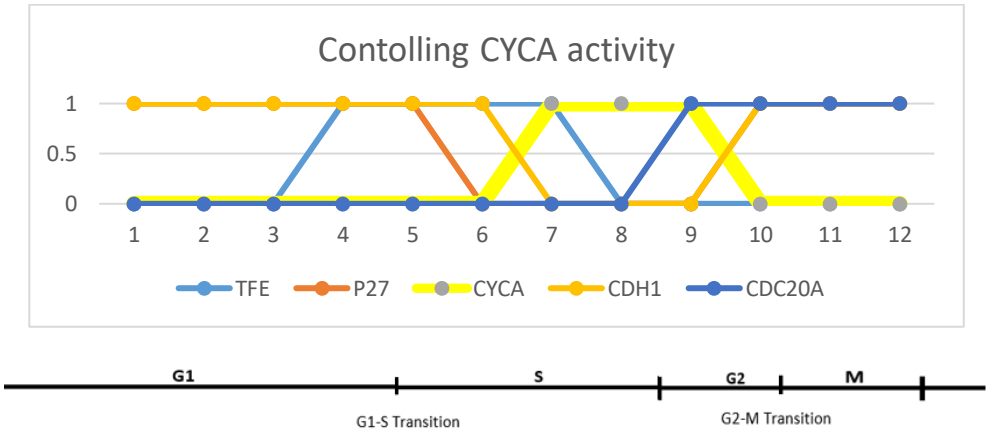
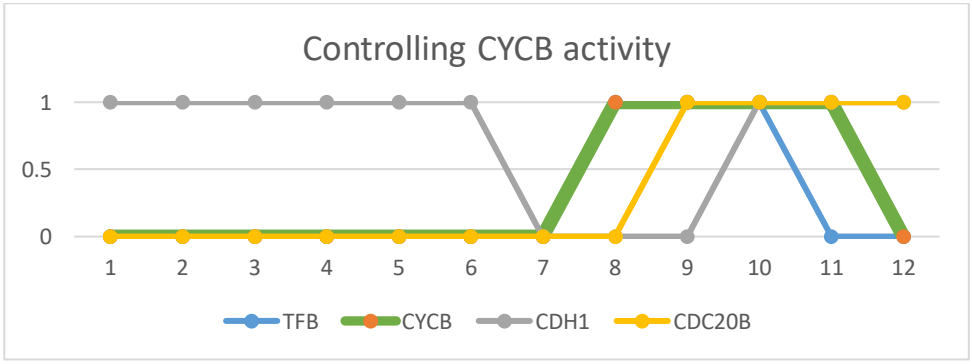


Figure 3.38: Temporal evolution of CycA and its controllers in synchronous update

Figure 3.39 characterises controlling processes for CycB. Synchronous update has clarified CycB activity. Specifically, CycB should be active when TFB becomes active. However, CycB activity requires another two factors CDC20B and CDH1 to be silent. As shown in Figure 3.39, when CDH1 becomes silent at time step 7, CycB becomes active and leads the M phase transition at time step 8. However, CycB activity can be inhibited when its degraders CDC20B and CDH1 become active. CDC20B becomes active at time step 9. However, this is not enough to inhibit CycB so at time step 10, CDH1 becomes active. It works together with CD20B to inhibit CycB activity. These activation and inhibition processes manage CycB levels to enable it to do its control function.



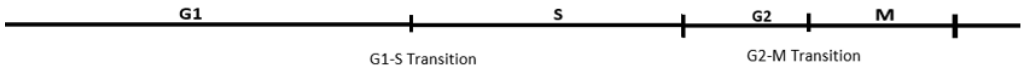
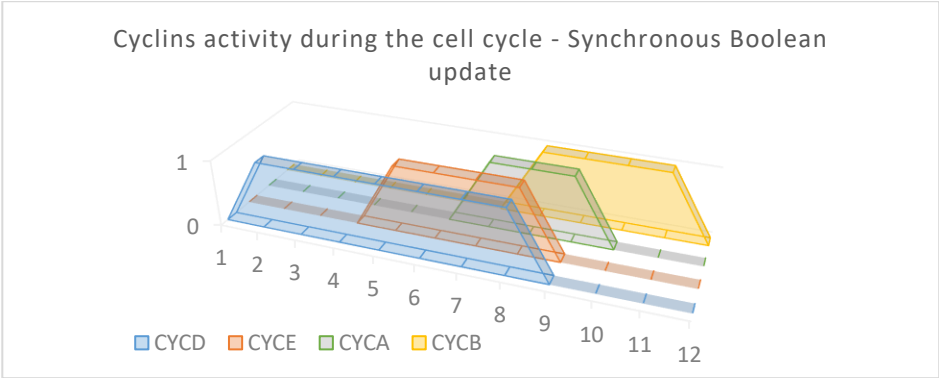


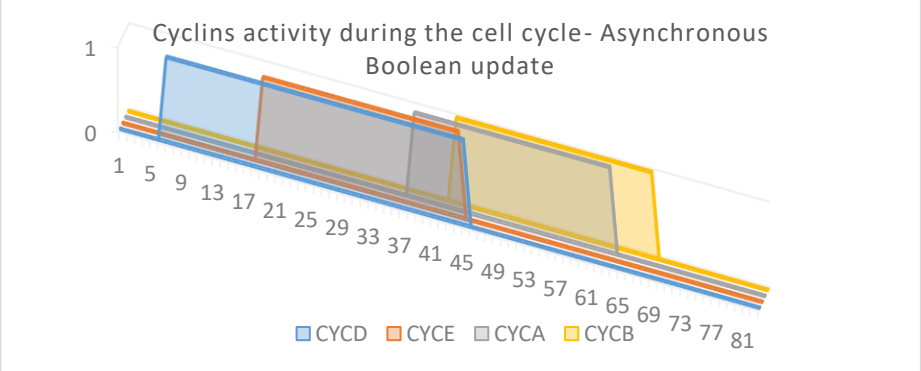
Figure 3.39: Temporal evolution of CycB and its controllers in synchronous update

3.8.3 Cyclin Activity in Cell Cycle with Synchronous and Asynchronous Updates Compared With ODE Model

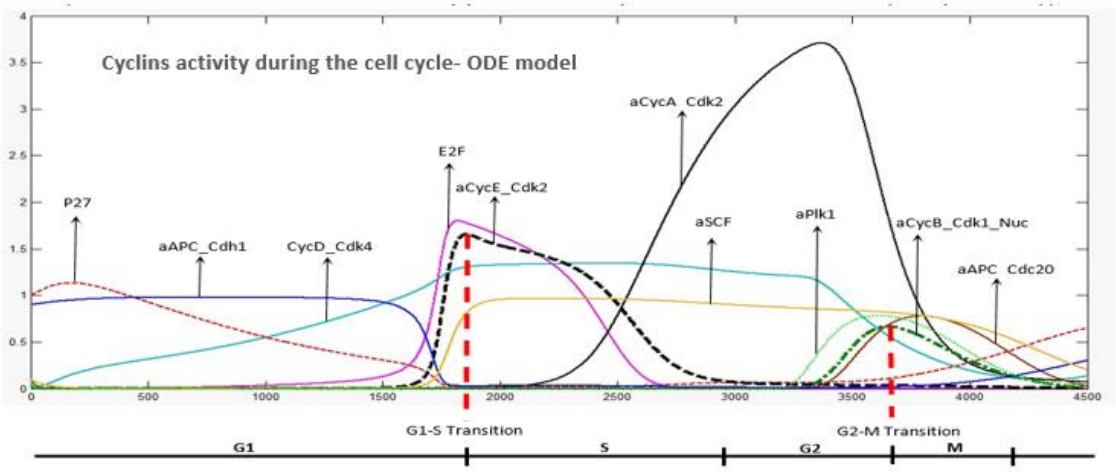
The cell cycle system as described previously consists of four phases. These phases are well-ordered and controlled by cyclins. The literature shows that each cyclin can control each phase or can work with other cyclins to move the cell from phase to phase. Figures 3.40a and 3.40b show how our Boolean system results characterised the long-time behaviour of cell cycle system depending on cyclin activity, and how cyclins can lead cell cycle phases in the mammalian cell cycle system similar with realistic result from ODE Model (Ali Abroudi, 2017) as shown in Figure 3.40.c. This was revealed especially when we used the synchronous and asynchronous updates with the same initial conditions



(a)



(b)



(c)

Figure 3.40: Cyclin activities in cell cycle phases [1] (Ali Abroudi, 2017). (a) Synchronous and (b) Asynchronous updates, and (c) ODE model results. One cyclin or more than one can lead a particular cell cycle phase.

The synchronous update explained the abundance of cyclins and their activity in cell cycle over 12 time steps of simulation as shown in Figure 3.40a. Asynchronous update showed time delays in system interactions and therefore these results can be considered more realistic. Time for this simulation was 81 time steps as shown in Figure 3.40b. However, our model can explain cell cycle activities and behaviour either with synchronous or asynchronous updates; the main difference is that asynchronous update presented the activities with temporal reality incorporating time delays. Consequently, the results from the discrete model can describe the dynamic behaviour of the cell cycle system approximating realistic continuous results from the ODE model (Figure 3.40c in comparison with Figure 3.40b). The results from our discrete model and ODE model are compatible with the biological knowledge in terms of how each cyclin leads cell cycle phases.

[1] Reprinted from Elsevie, Journal of Theoretical Biology,429, Ali Abroudi,Sandhya Samarasinghe,Don Kulasiri, A comprehensive complex systems approach to the study and analysis of mammalian cell cycle control system in the presence of DNA damage stress, Page No 25, Sep 21, 2017, with permission from Elsevier, License Number 4464060603162 License date Nov 08, 2018.

3.8.4 Comparison of Asynchronous and Synchronous Attractors

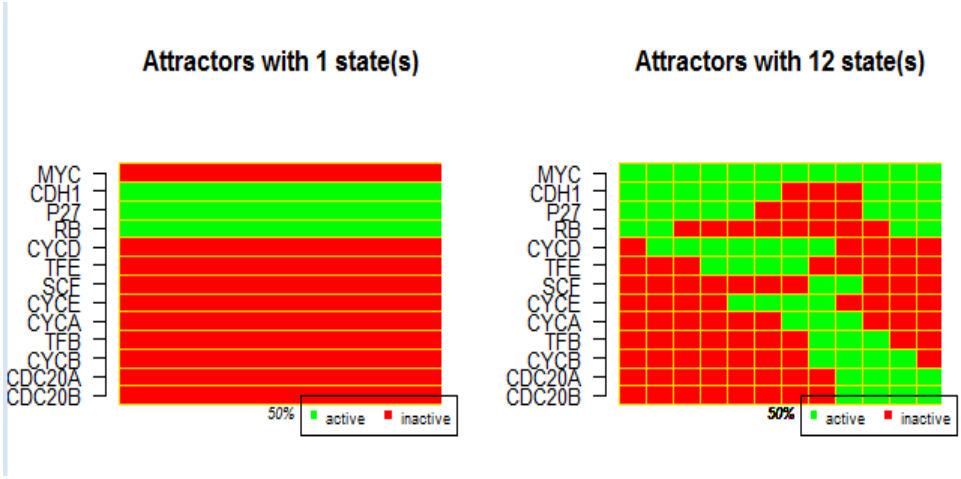
The results from simulation experiments from our model were compatible with the most popular existing models such as Faure et al. (2006) that was also compatible with the studies of Novak (2006) and Tyson (2004) that explained cell cycle events. However, our model has been developed with new and novel biological knowledge. Further, our model provides better results than current models in representing the dynamic behaviour of the cell cycle controller elements. For instance, our model shows the realistic order of all system element activities.

In the simplest synchronous update, we acquired two attractors as shown in Figure 3.41a. The first one was in a stable state with only CDH1, P27 and Rb being active in the absence of growth factors. This state was reached from all the other states lacking Myc activity, and thus, it corresponded to G0 or the quiescent state. In contrast, the second attractor on the right side of Figure 3.41a showed that in the presence of Myc, all corresponding trajectories lead to a unique limit cycle consisting of a sequence of 12 successive states. This 12-state cycle appeared with both exhaustive and heuristic (Figure 3.41b) initial conditions.

These attractors were also discovered in the asynchronous update. For instance, an exhaustive search within asynchronous update provided two attractors as shown in Figure 3.41c; however, despite complexity, asynchronous update provided two attractors, as shown in Figure 3.41c. The first one was in a steady state, while the second one was a complex attractor which hard to be investigated. Furthermore, the heuristic search within asynchronous update provided an attractor that describe cell cycle controller activities. This attractor is more realistic since it represented time delays in system activities as shown in Figure 3.41d.

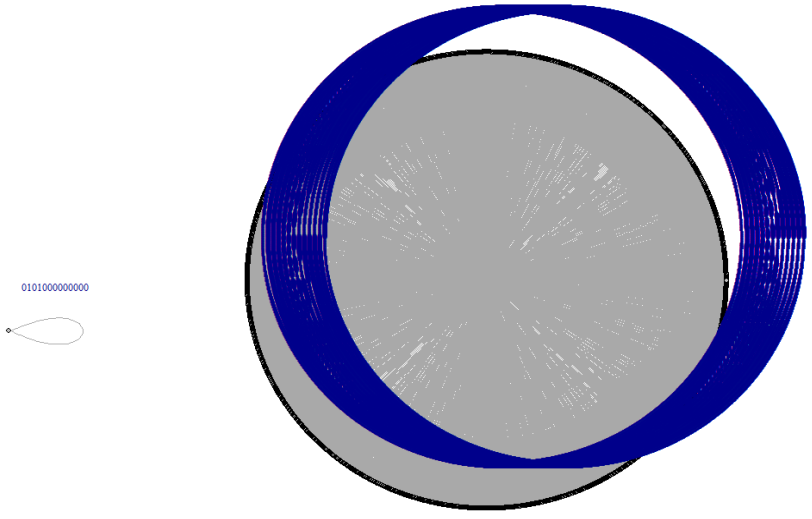
In the case where random time delays were applied on system components, synchronous update also provided promising results revealing realistic temporal evolution cell cycle over 20 time steps as shown in Figure 3.41e; even this model was very demanding computationally. Since this update is biologically more realistic than either standard

synchronous or asynchronous update, the temporal progression could also be closer to biological reality.

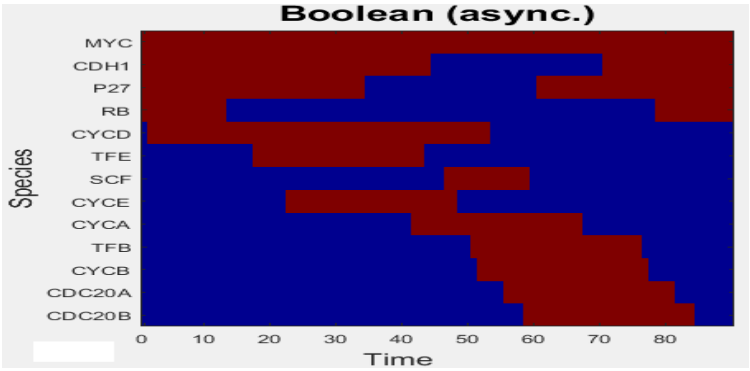


(a)

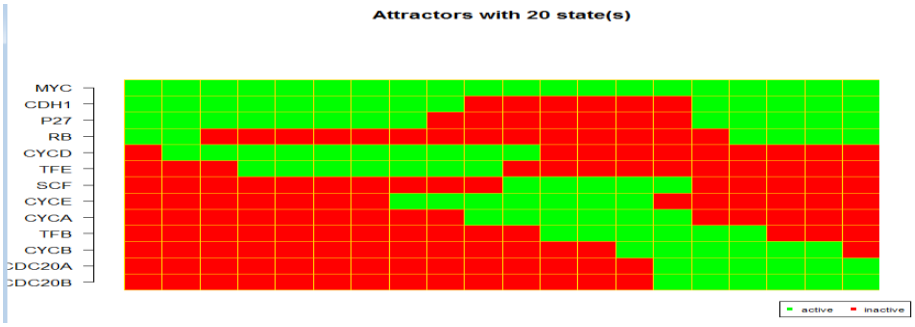
(b)



(c)



(d)



(e)

Figure 3.41 Asynchronous and synchronous attractors from model simulation. (a) Synchronous update with exhaustive (2^{13}) initial conditions; (b) Synchronous update with known (heuristic) initial conditions; (c) Asynchronous update with exhaustive (2^{13}) initial conditions.All the right part represents one complex attractor. The ring on the right side represents complex attractor for all possible random initial conditions 2^{13}, the blue ring represent the values for each state (the labels) and each states is represented by dot. Therefore, the black ring represents the dots for each state. The gray lines represents the states transitions between states.(d) Asynchronous update with known (heuristic) initial conditions; and (e) attractor with 20 states using asynchronous update with random time delays applied to cyclins only

It is clear from the last part of the simulation that both synchronous and asynchronous updates are comparable in describing the phases of cell cycle transitions. Moreover, both showed the correct biological interactions and cyclin oscillations. They both gave the same

results; however, synchronous update using time delays produced a realistic temporal evolution of cell cycle showing a longer time horizon.

Our enhancement for the current models provides better results, especially, the realistic structure and dynamic behaviour for the cell cycle components. We can conclude that our system can represent cell cycle stages and dynamics well and the attractors found correspond to actual processes in various cell cycle stages. Simulations with a large number of initial conditions revealed that systems is robust against the perturbations to initial conditions in all the scenarios studied. A comprehensive robustness study is presented and investigated in the next section.

3.9 Network Robustness, Temporal Dynamics and Sensitivity Assessment

This part assesses the robustness of our developed Boolean network. Boolean networks that simulate biological networks are presumed to be robust when facing a small amount of noise or change in the system. From this perspective, it is important for network models to be tested for robustness against various noises such as element failures. Here we used artificial noise such as a random noise.

Random noise can be applied by using perturbations to the current state of the network in simulation. We appreciatively employed BoolNet package due to its inclusion of the functions that support these robustness assessments.

We applied noise to the current state of the network and *used perturbation trajectories function* to measure its impact on network states. Specifically, we generated a set of initial states and then created a set of perturbed copies from these states. We made these perturbed copies by randomly flipping bits before measuring the impacts of these flips on the dynamic behaviour of the network. For example, we generated 100 states and another 100 copies with a one-bit flip, then performed a single state transition for each state. Then we measured the fraction of different bits (normalised Hamming distance) between each

state and the corresponding perturbed copy. Boolean network robustness can be assessed with the assumption that more robust networks yield a lower Hamming distance. Our simulation results showed a very small Hamming distance of 0.079 which indicates that the model is robust. In addition, we compared our results with the most popular Boolean model available in the literature, Faure et al. (2006) model, which was considered a very robust model with Hamming distance of 0.107 as shown in Table 3.4.

Further, we studied another measure of robustness- the effect of perturbation on the long-term behaviour of the system by comparing the attractors reached from the initial states with those reached from their perturbed copies using normalized Hamming distance. Results revealed a difference of 0.73 and the same measure for Faure et al. (2006) model was 0.93 as shown in Table 3.4. Our models shows a greater level of robustness against changes to attractor states due to perturbed initial conditions. Thus our results show a greater agreement with literature assumption that small changes in the state must not affect the long-term behaviour of the network, so most of the attractors should be the same.

Table 3.4 Robustness of the Boolean model comparison with previous model

Robustness Measure	Our model	Faure et al. (2006) model
Impact on system states based on Hamming Distance	0.079	0.107
Impact on long-term behaviour of system (Attractors)	0.73	0.93

Moreover, we studied the average sensitivity of element behaviour to perturbed initial states as shown in Table 3.5. This measure only assesses single transition functions, and we counted the number of successor states for a specific element for the original initial states and their corresponding perturbed copies and estimated the average difference for the

two conditions. For instance, the measured average sensitivity of the transition function of one of our system elements Cyclin E had a value of 0.11. We then compared it with Faure et al. (2006) model which had the same average sensitivity of 0.11 for CycE. Individual sensitivity comparison between the two models was also carried out as shown in in Table 3.5 and Figure 3.42.

Table 3.5 Individual transition function sensitivity comparative study

Average sensitivity of transition function	Our Model	Faure et al., 2006 Model
Myc	0.03	NA
CDH1	0.09	0.09
P27	0.05	0.1
Rb	0.14	0.06
CycD	0.04	0.06
TFE	0.11	0.12
SCF	0.09	NA
CycE	0.11	0.11
CycA	0.17	0.08
TFB	0.04	NA
CycB	0.10	0.12
CDC20A	0.05	0.09
CDC20B	0.12	0.09

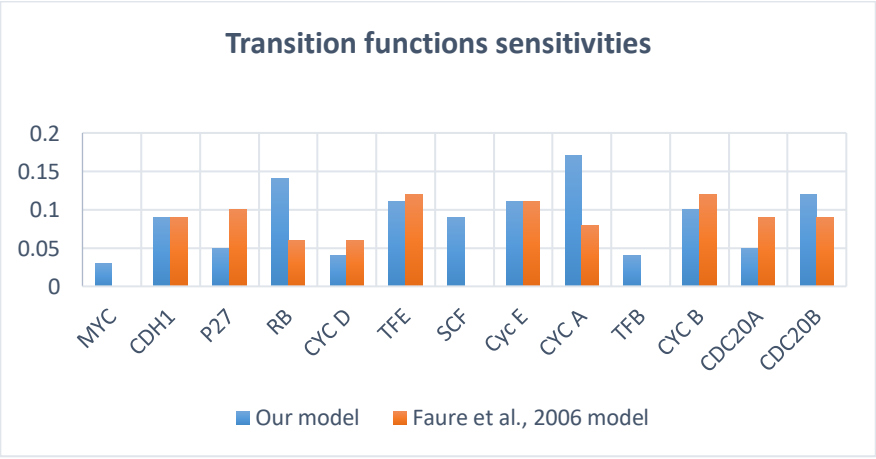


Figure 3.42: Transition function sensitivity for individual species in the network. Blue columns indicate our model results and red columns indicate those for Faure et al. (2006) model.

3.10. Mutation Study

Boolean network used to modelling biological network such as human signalling network; It helps to investigate human diseases and therapeutic (Saadatpour et al., 2011).

In this section we can specify how each element can effects on the global behaviour of cell cycle model. Applying systematically perturbation on the system by tuning the state of specific node from active to inactive (or vice versa). By doing this kind of perturbation we mimic the either biological genetic knock out to exposes the pharmacological inhibition of a particular node or gene overexpress. In addition, this can support to investigate human diseases, also this provides us a new knowledge about our system for example which elements have a critical effect on network efficiency.

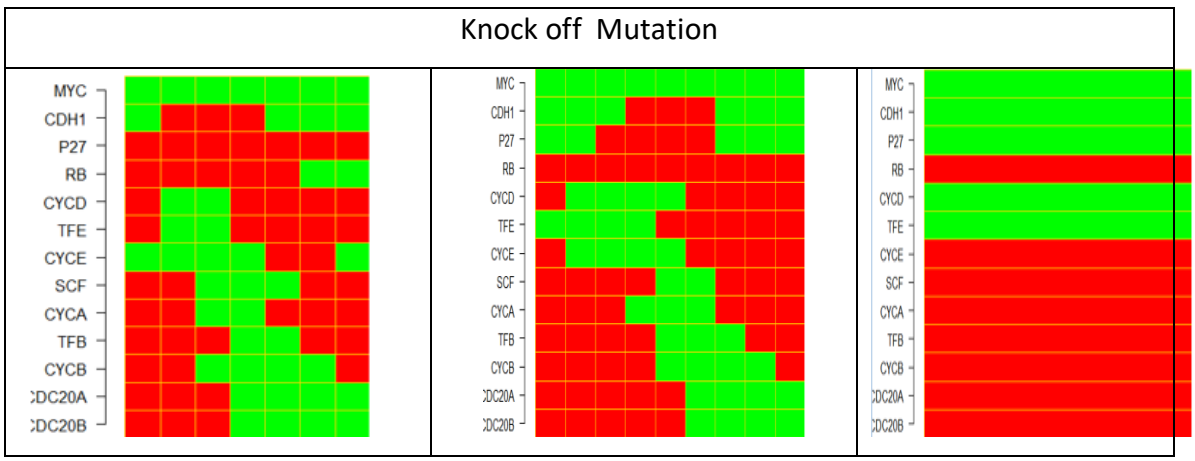
Herein, we studied the effect of mutation in elements on system behaviour. We fixed the value of Myc to 1 and kept it active for all mutations to mimic consistent presence of growth factor that activated Myc to fire up the Boolean network. Then, we applied individual mutations to all system elements.

We applied two types of mutation: 1) a knock-off mutation by fixing the values of the selected element to be 0 for the whole simulation, and then 2) an overexpressed mutation by fixing the value of the selected element to 1 for the whole simulation. In this kind of mutation study, we investigate the effects of mutation in individual elements in the network. First, the chosen element was knocked off for the whole simulation. We then repeated the simulation by overexpressing the same chosen element and collected corresponding results for the system. These mutation tests were conducted on all 12 elements one by one, which represented the components of the whole system. We then investigated and compared results with available literature to draw our conclusions.

The initial states for the system used in these investigations were the known (heuristic) states used previously in this study and this state vector represents the early G1 phase of cell cycle. The next sub-section investigates the individual knock off mutations in the Boolean cell cycle controller model and the sub-section that follows investigates individual overexpression mutations.

3.10.1 Knock off Mutations

Figure 3.43 summarises all the knock-off mutation results. When P27 was knocked off, the system provided us with cell cycle phase transitions, but all cyclin activities had one fewer time step than usual as shown in Figure 3.43a; this provides 7 states not 12 states as normal cell cycle transitions. Moreover, the Rb knock-off mutation showed an acceptable cell cycle global behaviour and cyclin oscillations, but with 9 states and CycE was become active with CycD at the same time, this is not precise result as shown in Figure 3.43b. In addition, knock-off mutation on Cyclin E for the entire simulation affect the system efficiency, when Cyclin E that has been knocked off the system had not the ability to describe cell cycle phases and cyclin oscillations as shown in Figure 3.43c, the simulation provides just one state. In addition, knocking off SCF has a negative impact on both of Rb and P27, which made it inactive for the entire of simulation. Further, Cyclin E becomes active for all the simulation and the cell cycle for this mutation has just seven states as shown in Figure 3.43d.



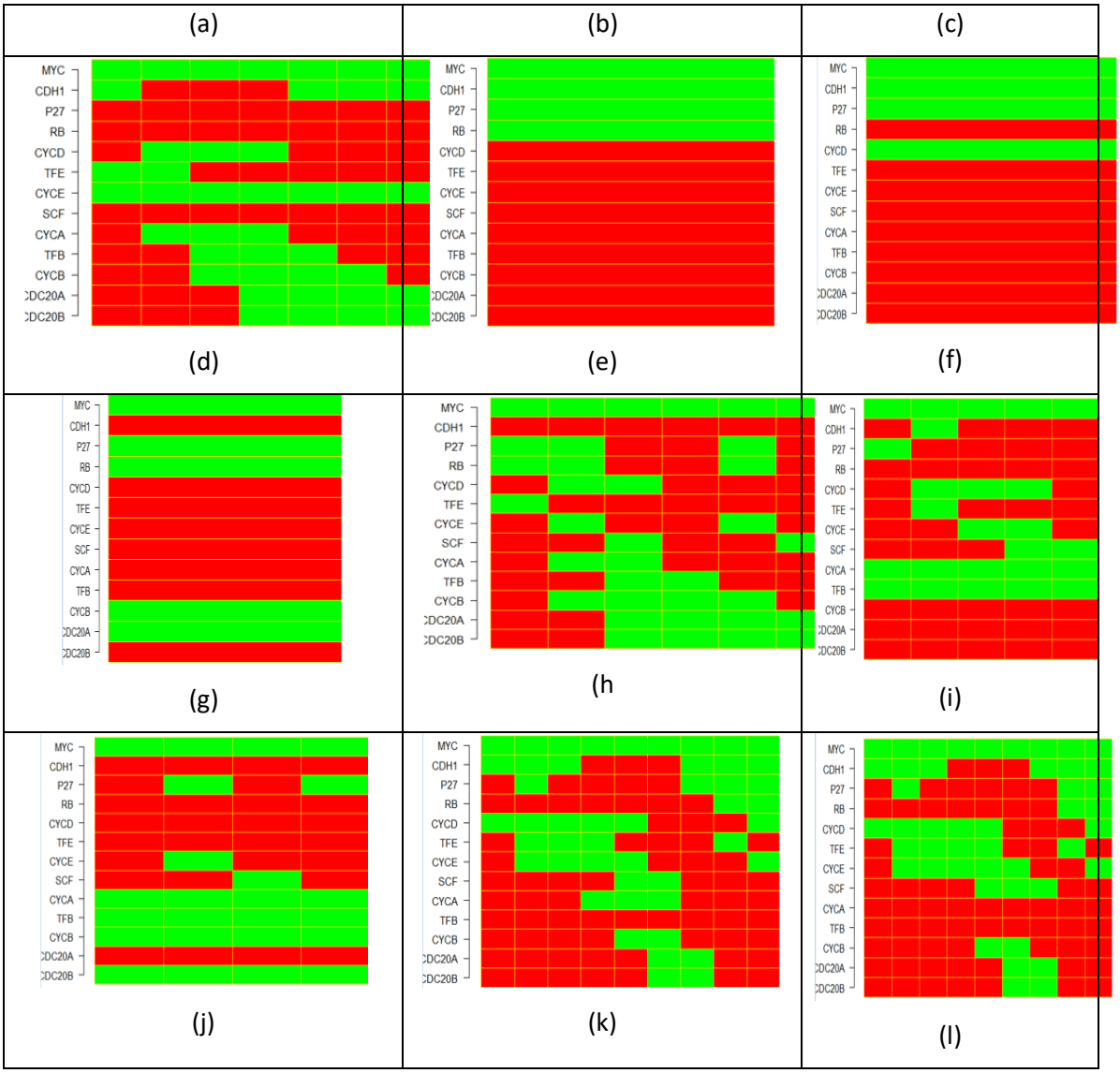


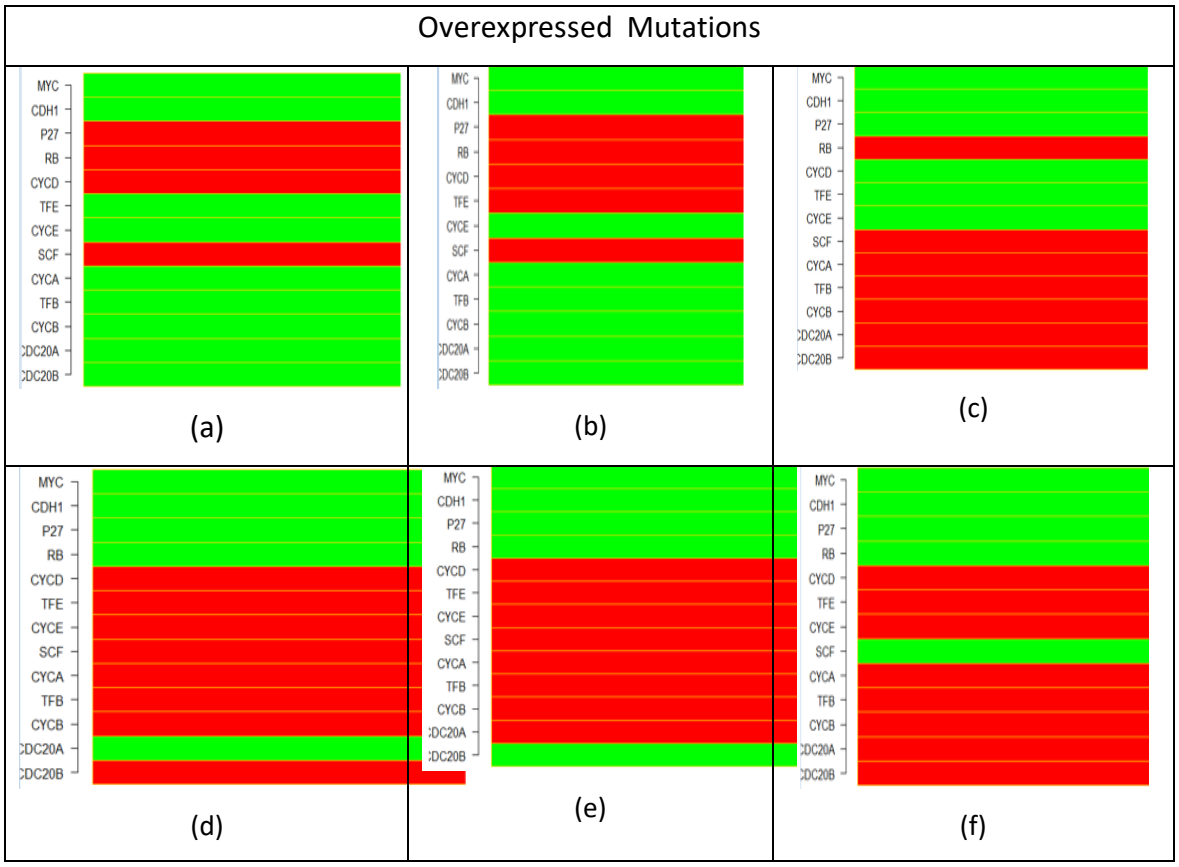
Figure 3.43: knock off mutation from model simulation. (a) Knock off P27; (b) Knock off RB; (c) Knock off CycE; (d) Knock off SCF; (e) Knock off CycD; (f) Knock off TFE; (g) Knock off CDC20B; (h))Knock off CDH1; (i) Knock off CycB; (j) Knock off CDC20A; (k) Knock off TFB and (l) Knock off CycA.

Another inference from this type of knock-off mutations was that several elements, when knocked off, the cell cycle did not go anywhere, such as knock-off Cyclin D, TFE and CDC20B as shown in Figure 3.43e, f and g, respectively. Furthermore, some other elements, when knocked off such as CDH1 CycB, CDC20A, TFB and CycB respectively, produced inconsistent cell cycle activities as shown in Figure 3.43h, I, j, k and l respectively.

3.10.2 Overexpressed Mutations

The second type of mutation was overexpressed mutations as shown in Figure 3.44. This figure presents concise results from overexpressed mutation experiments. consequently, Overexpressing most of system elements forced the cell cycle to not go anywhere; these include overexpressing TFE, CycA, P27, CDC20A, CDC20B, SCF and Rb as shown in Figure 3.44a, b, c, d, e, f and g respectively. . In addition, overexpressing CycB and TFB produced just two states then the system is not go anywhere as shown in Figure 3.44h and I respectively.

Furthermore, some system elements when were overexpressed, the system worked in the incorrect way. For instance, overexpressing CDH1, CycE and CycD for the entire simulation retained inconsistent global system dynamics as shown in Figure 3.44j, k and l respectively.



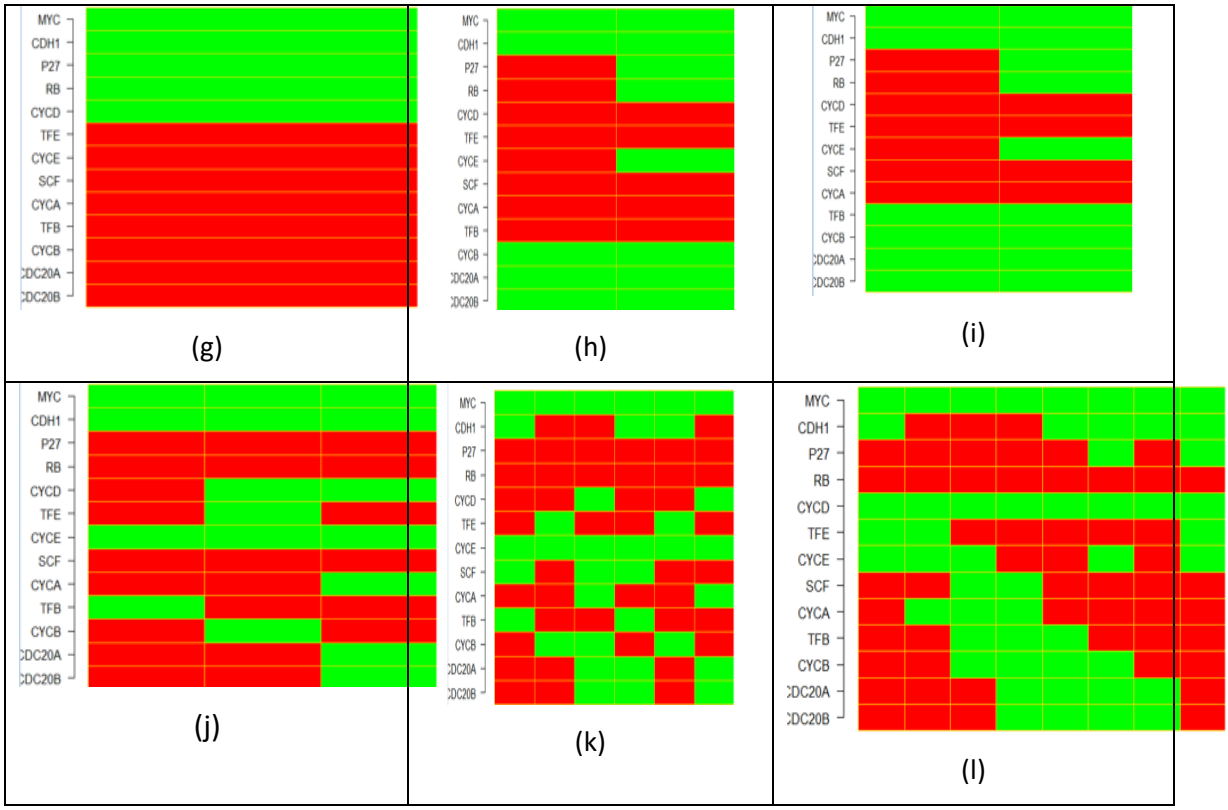


Figure 3.44: Overexpressed mutation from model simulation. (a))Overexpressed TFE; (b) Overexpressed CycA; (c) Overexpressed P27; (d) Overexpressed CDC20A; (e) Overexpressed CDC20B; (f) Overexpressed SCF; (g) Overexpressed RB; (h)) Overexpressed CycB; (i) Overexpressed TFB; (j) Overexpressed CDH1; (k) Overexpressed CycE and (l) Overexpressed CycD.

Finally, we present another scenario for mutation simulation. This scenario did not depend on a specific initial condition to start the system but used a very large number of random initial conditions. The number of random initial conditions was 2^{11} (2048). We found that the simulation results were the same as those for original known (heuristic) initial states of cell cycle.

Indeed, our work provided a roadmap for how Boolean network modelling can be used as a predictive tool to uncover the dynamic patterns of a biological system under various perturbations. This qualitative model can be used to investigate Biological mutations such as protein function loss and protein overexpression as mentioned previously. We did individual element mutation studies but multiple element mutations can also be studied with the model. Thus we have utilized our model to introduce and investigate a new

individual biological mutation study for cell cycle controller system while previous models have not been used to study or investigate any biological mutation for cell cycle controller system.

3.11 Chapter Summary

With our proposed model, we assessed the strength of the logical approach to model complex biological systems. This was undertaken using a Boolean network which was considered the simplest form of logical model. Our developed model was used to study the dynamics of the cell cycle control system.

We introduced a new Boolean model that can mimic the fluctuations or oscillations of Cyclins in cell cycle progression. The model included the most important elements, such as cyclins, and their activators and inhibitors. Our Boolean network with 13 proteins can capture the essential aspects of cell cycle and show cell cycle phases depending on the changes in cyclins since each cyclin lead a cell cycle phase. The new model has also revealed localisation of system elements (*i.e.*, element abundances, such as Cyclins abundance) and the fluctuations of cyclins in cell cycle.

Our model was inspired by Singhania et al. (2011)generic model which consisted of nine elements to control Cyclin activities that control cell cycle phases. Their model was built without regulating system activities and interactions using any Boolean rules or functions. Singhania et al. (2011) model depended on the assumption that the elements interacted and changed their state in a sequence of states and values for these states were given to the model externally as Boolean rules were absent in their model. In addition, their model did not include all Cyclins; it missed some elements, such as Cyclin D that controls G1 phase in cell cycle and affects the behaviour of other elements. Therefore, we enhanced this model in several ways; first, we boosted this model by increasing the size of this system. We added four new and significant biological elements to the system components - Myc,

CycD, P27 and RB; these elements have been selected assiduously from literature to improve the system efficiency.

Our model consisting of these new 13 rules can explain cell cycle phases and system dynamics (temporal evolution of element activities). It can be used to study diseases, such as cancer. We made a comprehensive individual element mutation study and some of these mutations may be indicative of diseases. The second enhancement is that we developed a comprehensive Boolean model with new Boolean rules. These developed rules controlled all biological interactions in cell cycle phases. This set of Boolean rules that represents system activities was developed from biological knowledge and a literature review of Boolean models.

We compared our model with one of the most popular Boolean models, mammalian cell cycle control model developed by (Fauré et al., 2006)). Our model can localise all Cyclins in different cell cycle phases while Faure model cannot localise cyclins and other elements. Our model can also show the transition states, such as local transition from G1 phase to S, G2 and M phase until the end of cell cycle in contrast to the aforementioned model.

We have demonstrated that our proposed Boolean network was robust and resilient in design. The biological states of the attractor represented cell cycle phases and the dynamic change in oscillation of Cyclins in cell cycle phases. The model we developed relied on two crucial benchmark models (Fauré et al., 2006; Singhania et al., 2011); these models did not include checkpoints in their implementations. They studied the major components of the cell cycle, which were often able to capture the essential behaviour of cell cycle system. Our model did not involve cell cycle checkpoints either.

Mainly, our results were investigated from a qualitative point of view, where the order of activity (“OFF” or “ON”) was consistent with the available data. Further, our model provided a visual assessment of cell cycle dynamics; it revealed cell phenotypes.
This information was provided in qualitative form and demonstrated cell phenotypes such as the cell states.

Next chapter presents and investigates a semi-quantitative model developed by converting this Boolean model, a qualitative model, into a continuous fuzzy logic model. It explores how the qualitative knowledge from this discreet model can be used to develop the novel semi-quantitative model.

Chapter 4. A Novel Semi-Quantitative Cell Cycle Controller Model – A Fuzzy Logic Model

Mathematical and computational modelling approaches are becoming influential tools to analyse and discover knowledge hidden in biological systems. Generally, there is a close correspondence between these modelling techniques and laboratory experiments on signalling and regulation of networks (Wittmann et al., 2009). However, as mentioned earlier, use of these modelling techniques, such as the ODE models, is limited by the scarcity of data which in some case are available only in qualitative form. Furthermore, these methods are considered crisp methods since they utilize crisp (exact) values such as concentration for the entities in the system. The scarcity of such crisp values and the vagueness of uncertain data can and must be addressed by other methods.

These limitations prompted researchers to carry out research on alternative methods that can make greater use of the advantages offered by continuous and other discrete models. These attempts will greatly aid in understanding complex systems. Logic-based models, such as fuzzy logic, can be promising in modelling biological systems. This is because logic models make it simpler and easier to represent system components, interactions, behaviour and events. In addition, logical models can generate accurate qualitative results from system dynamics in a discrete form. This can provide a simplified view of system activities that are useful for gaining insight into overall behaviour.

There is the potential for fuzzy logic to replace both Boolean and ODE models in modelling cell cycle system. Here, it is one of our research objectives to attempt to develop a new fuzzy logic model to enhance the Boolean model presented in the previous chapter. A description of the development process and other related information is given in detail in this chapter.

This chapter explores the following questions:

- **Is it possible to transform the qualitative model in the previous chapter into a fuzzy logic model?**
- **If fuzzy logic relies on prior knowledge and vague information with missing data, such as the kinetic parameters, can it be flexible like the Boolean model and be accurate like the ODE model in order to simulate the continuous behaviour of the cell cycle controller system?**
- **Can the cell cycle controller in the form of a fuzzy logic model provide the expected result and improvements? Are the semi-continuous results from a fuzzy model better and more flexible than discrete results from the Boolean model to investigate and mimic cell cycle controller system?**
- **What are the best features of fuzzy logic method that can be utilised to make a fuzzy logic model a close equivalent of an ODE model?**
- **Can a fuzzy logic model handle imprecise biological information?**

It is hypothesised that a thorough investigation can answer these questions and enable the development of a fuzzy logic model for continuous description of biological systems in a way that approximates ODE models while making it easier to analyse and gain insights into the mammalian cell cycle controller system.

Information in this chapter is organised into several sections including fuzzy logic theory, principles and fuzzy sets. Furthermore, we explore the potential of fuzzy logic to explain the behaviour of complex biological systems. In addition, we investigate more into fuzzy inference systems to identify how they can be made to mimic ODE models, and then we compare the outcomes from both fuzzy and ODE models simulating the behaviour of the mammalian cell cycle controller system to explore the benefits of the fuzzy model.

4.1 What is a Fuzzy Logic System?

Fuzzy logic allows variables to be continuous and more finely valued than other discrete models such as Boolean models. It allows the possibility to consider all the intermediate levels of what is modelled within a qualitative modelling framework. In addition, it can show the relative activity of elements resembling real biology.

L. A. Zadeh (1965) Introduced fuzzy logic and fuzzy theory as well as fuzzy sets and this transformed the two-valued [0, 1] representation into that of an interval [0, 1].In this new representation, vague and uncertain data are represented using fuzzy sets (L. A. Zadeh, 1965). In fuzzy set theory, the total range of a particular variable (universe of discourse) in the interval [0, 1] is populated by a number of overlapping fuzzy subsets each representing a membership function of the variable which is predominantly active in a specific region of the range. A particular value of the variable can then belong to one or several membership functions with a degree of membership between [0, 1]; this allows continuous representation of system. This method is the opposite of the traditional theory, such as crisp set theory, where an element can be a part of the set or not.

The main difference between fuzzy logic and other methods is that fuzzy logic depends on linguistic terms more than numbers. For instance, “John is very tall;” in this example, the linguistic term “very” depicts the proportion (fuzzy subset) of the fuzzy variable “tall.” As noted in this example, and by extension, a fuzzy sets depend on linguistic expressions to represent a problem and find solutions. Reasoning with fuzzy sets emulates human reasoning and decision-making thus creating a solution framework for vaguely defined systems.

4.1.1 Fuzzy Logic in Biology and Bioinformatics

Bioinformatics is a combination of computer science, biology, chemical and physical principles and tools that are used to model large biological data sets. For example, high-throughput technologies, such as DNA microarrays, provide large amounts of data rapidly.

These data contain rich information embedded in high levels of complexity to be analysed by biological data analysis (Torres & Nieto, 2006).

Currently, fuzzy logic is one of the available solutions to problems in bioinformatics and biology. For instance, it is used to study differences between polynucleotides (Torres & Nieto, 2003). Fuzzy adaptive resonance theory has also been used to evaluate experimental expression data (Tomida, Hanai, Honda, & Kobayashi, 2002). Fuzzy logic is used to predict the subcellular locations of proteins from their dipeptide composition; here, the modeller depends on a specific algorithm, such as fuzzy k-nearest neighbour algorithm. Furthermore, some studies have used fuzzy logic to cluster a large number of genes from microarray data (Belacel, Čuperlović-Culf, Laflamme, & Ouellette, 2004) and used genetic fuzzy systems for DNA sequencing (Cordón, Gomide, Herrera, Hoffmann, & Magdalena, 2004).

Biology researchers are interested in studying biological systems; and many use protein and gene regulatory networks (GRN) to understand cellular functions and diseases. These networks can help specify the temporal or spatial expression patterns of a set of genes (Helgason & Jobe, 2003). Although various methods have been constructed to understand GRN and develop hypotheses that are related to the interactions of genes, few of them have been related to the mammalian cell cycle (Helgason, Malik, Cheng, Jobe, & Mordeson, 2001).

Therefore, a fuzzy inference system based on fuzzy logic concepts is used to develop a cell cycle control system in this study. It is hypothesised that analysis of biological data and modelling biological systems become easier using fuzzy logic-based modelling and analysis approaches.

4.2 Features of Fuzzy Logic

Biological knowledge is currently imprecise and fuzzy systems can handle this kind of knowledge. Therefore, choosing fuzzy systems as computational tools is considered a good decision. Fuzzy logic can make variables continuous within a qualitative form. As we have the aim of converting the Boolean model in the previous chapter into a fuzzy model, we need to know how to transform the Boolean discrete representation into a flexible semi-continuous representation using fuzzy logic. For this reason, the next sub-section explains how the Boolean network can be mathematically transformed into semi-continuous models using fuzzy logic theory.

4.2.1. Transforming Boolean Models into Fuzzy Models

Boolean models represent the state of components in a discrete form- either "OFF"' or "ON." These models are often able to capture the essential behaviour of a network. However, these models cannot provide intermediate values of variables, such as concentration of proteins in a signalling system. Mathematically, many standardised methods have been introduced to transform discrete Boolean models into continuous models, such as *HillCube*, which uses the multivariate polynomial interpolation technique and incorporate Hill kinetics. However, fuzzy logic can work with qualitative knowledge in the literature and as stated before it allows variables to be more continuous or multivalued than discrete models, such as the Boolean network (Wittmann et al., 2009).

The state of each node in Boolean models can be "ON" if the concentration of the protein is above the specified threshold. Boolean models can be generalised by using fuzzy logic. Therefore, in fuzzy logic, the concept of a node being "ON" or "OFF" can be eased; this is done through the degree of membership (DOM) μ as represented in Equation 4.1:

$$0 \leq \mu X_i(\bar{x}_i) \leq 1 \quad (4.1.)$$

Eq. 4.1 states that the level of species X_i belongs to the membership function μ with a degree of membership between 0 and 1. This equation is applied to each species X_i, and their concentration $0 = \bar{x}_i = 1$ can identify the degree of activation "ON". There are two standard ways to generalise Boolean operators "AND", "OR" and "NOT" (Wittmann et al., 2009) as explained in Eqs 4.2 and 4.3:

Min-Max Logic (Wittmann et al., 2009) (4.2.)

$$X_i \wedge X_j \longmapsto \min\left(\mu_{x_i}(\bar{x}_i), \mu_{x_j}(\bar{x}_j)\right)$$

$$X_i \vee X_j \longmapsto \max(\mu_{x_i}(\bar{x}_i), \mu_{x_j}(\bar{x}_j))$$

$$\neg X_i \longmapsto 1 - \mu_{x_i}(\bar{x}_i)$$

Product-Sum Logic (Wittmann et al., 2009) (4.3.)

$$X_i \wedge X_j \longmapsto \mu_{x_i}(\bar{x}_i).\mu_{x_j}(\bar{x}_j)$$

$$X_i \vee X_j \longmapsto \mu_{x_i}(\bar{x}_i) + \mu_{x_j}(\bar{x}_j))$$

$$\neg X_i \longmapsto 1 - \mu_{x_i}(\bar{x}_i)$$

Product sum requires normalising the unit intervals. The reason is that $\mu_{x_i}(\bar{x}_i) + \mu_{x_j}(\bar{x}_j)$ can be greater than 1. However, both min-max logic and product-sum logic can be used to construct fuzzy function $\bar{B}_i$ from Boolean function Bi.

In this sub-section, we provided the mathematical conversion of Boolean aspects into fuzzy logic aspects where values of the variables become more flexible than discrete. The next

sub-section provides more details on fuzzy logic aspects, such as fuzzy sets and principles of fuzzy rules.

4.2.2 Main Differences between Traditional Theory, the Fuzzy Set and Membership Function

The traditional theory in terms of crisp theory can be extended; this extension can be interpreted as fuzzy theory. So, what is the main difference between them? This can be explained using an example of thinking and decision making in our daily life. For example, to specify the height X of a person as a member or not in the set *tall (A*), the membership $\mu_A(x)$ of X in A can be shown as in Equation 4.4:

$$\begin{aligned} &\mu_A(x) = 1 && \text{if x is completely in A;} && (4.4.)\\ &\mu_A(x) = 0 && \text{if x is not in A;} \\ &0 < \mu_A(x) < 1 && \text{if x is partly in A.} \end{aligned}$$

As we can notice, the membership value can be in the interval 0 and 1. So, any person taller than say 180cm is considered tall and can be classified by the crisp values x|x >/ 180 cm as shown in Figure 5.1a. The same situation can be represented by fuzzy set method using a degree of membership function as shown in Figure 5.1b. It describes a fuzzy set of over a universe of discourse of x by membership function μ A(x): X-> [0, 1]. This means that every element of x has a degree of membership μ A(x) in A in the interval [0, 1].

Mathematically, a membership function can be presented as a curve that specifies how each point in the input space is mapped into a degree of membership in the interval [0, 1].

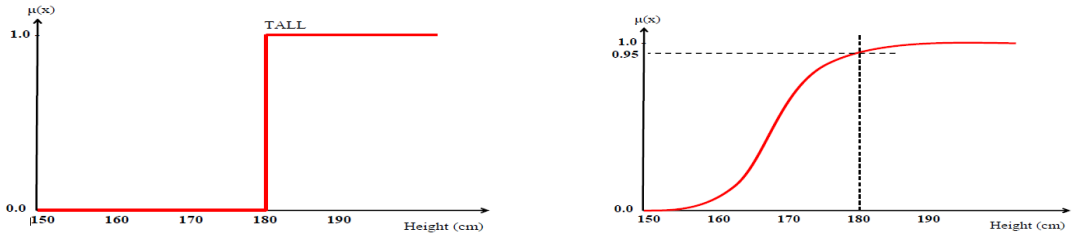


Figure 4.1: Difference between crisp and fuzzy representations of height: a) Representation of a tall person in crisp form and b) Representation of a tall person in continuous fuzzy form

Experts and software engineers can specify this design depending on its suitability to represent the situation in an efficient and simple way by selecting appropriate membership functions for the fuzzy sets(J. S. R. S. JANG, C.T., 1995; MATHWORKS, 2012) . The selection of an efficient fuzzy set can be made by identifying a suitable universe of discourse and by specifying a convenient membership function. This needs good knowledge of the human experts (J. S. R. S. JANG, C.T., 1995).

Using the previous example of a person's height, the fuzzy set representing height can be classified into three subsets: short, average and tall (further details related to fuzzy sets are given in Sub-section 4.2.3). The representation of these fuzzy sub-sets by a degree of membership function can be demonstrated as shown in Figure 4.2. Here, the three fuzzy subsets are represented by Trapazoidal (for short and tall) and triangular (for average) membership functions. As we see from this figure, the person with a height of up to 184 cm can be a member of both the average and tall subsets with a value of the degree of membership of 0.1 and 0.4, respectively.

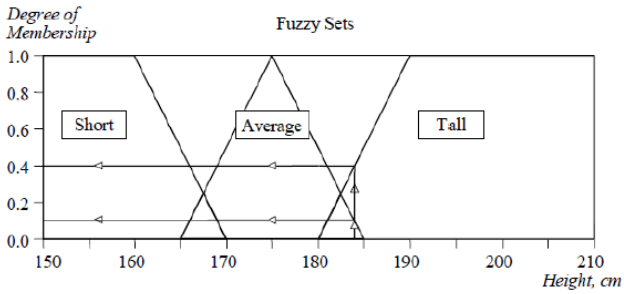


Figure 4.2: Height membership functions. Height consists of three fuzzy sets, which are short, average, and tall. The figure also shows how a person with a height of 184 cm can fall with a specific degree of membership function into both average and tall fuzzy sets.

In this figure, we demonstrated the meaning of fuzzy sets, membership functions and degree of membership in fuzzy subsets to represent the height of a person. Next we discuss how to reason with this information using fuzzy reasoning.

4.2.3 Linguistic Variables and Fuzzy Rules in Fuzzy Reasoning

L.A. Zadeh (1975) had introduced linguistic variables; he interpreted the variables values in linguistic expressions more than in numerical values(L.A. Zadeh, 1975). For instance, as described earlier, when we say John is tall, this indicates that height represents a linguistic variable and tall is the linguistic value in this sense. This linguistic expression can be part of an "IF-THEN" conditional statement called *fuzzy reasoning* (Negnevitsky, 2005) as shown in Equation 4.5:

IF antecedent(s) THEN consequent(s) (4.5)

In Eq. 4.5 the first part represents the antecedent that involves input in fuzzy form and the second part represents the output or consequence also in fuzzy form. In case we have more than one antecedent, fuzzy operators can be used to connect them together. There are three different operators that are used: "AND" operator, called fuzzy intersection or

conjunction, the “OR” operator, called fuzzy union or disjunction, and “NOT” operator, called fuzzy complement.

The following example employs the use of these operators. For instance, if we have two fuzzy sets, A and B, with membership functions $\mu_A(x)$ and $\mu_B(x)$

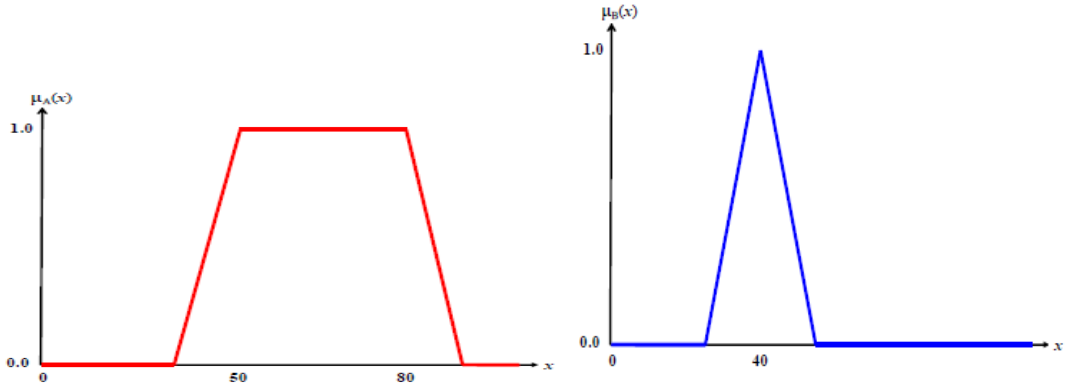
The intersection or conjunction between fuzzy sets A and B can be defined as:

$$\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu B(x)) \qquad (4.6.)$$

The union of two fuzzy sets A and B can be defined as:

$$\mu_A(x) \vee_B (x) = \max(\mu A(x), \mu B(x)) \qquad (4.7.)$$

The complement of fuzzy sets A can be defined as shown in Figure 4.3.



Fuzzy set A Fuzzy set B

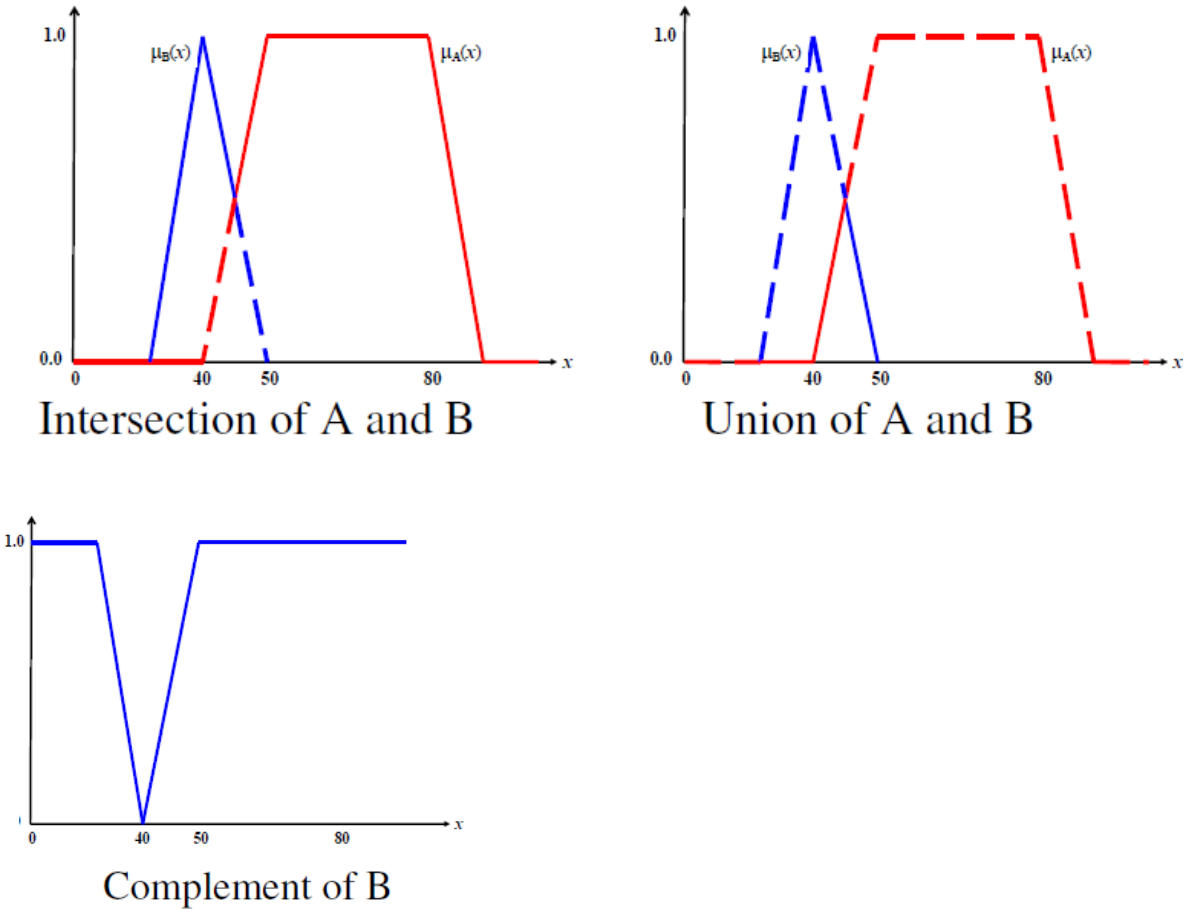


Figure 4.3: Use of fuzzy logic operators on fuzzy sets (Fuzzy set A and Fuzzy set B). The results for intersection and union are represented by the dotted line while the complement is represented in the last figure.

After setting up all the antecedents and connecting them together using suitable fuzzy operators, these parts should be processed together. The final result can include multiple parts that are aggregated into a single output of a fuzzy set (Negnevitsky, 2005); this output must be processed by defuzzification to convert it to a crisp value. Defuzzification is a process that extracts crisp values from fuzzy sets (J. S. R. S. JANG, C.T., 1995). Concerning how fuzzy sets, fuzzy rules, fuzzification, and defuzzification work in a single process is discussed in more detail in the fuzzy inference system in Section (4.3.).

In this section, we explained the aspects of fuzzy reasoning. In the next sections, we demonstrate the most important aspects in the design of fuzzy sets; then, we explain how fuzzy sets can characterise the states of biological entities.

4.2.4 Fuzzy Set Design

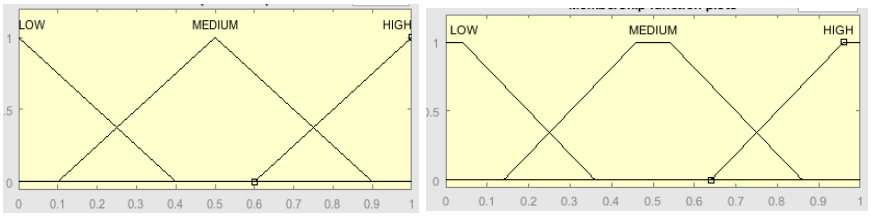
The design of fuzzy sets involves determining several properties such as shapes and parameter of fuzzy sets and the domain of discourse for the fuzzy sets. To design suitable fuzzy sets, several aspects must be taken into consideration due to their ability to influence system performance.

- **Type of biological property:** type of biological entity and its property can specify the fuzzy set design. For instance, fuzzy design for protein concentration must incorporate different levels of protein concentration, such as low, medium, and high concentration levels.
- **Functional considerations:** fuzzy design must reflect the functional behaviour of the biological entity. For example, the design for the fuzzy set must represent changes in concentration when system elements interact with each other.
- **Range of data observed**: the range of observed data should be compatible to the domain of discourse because it leads the design of the fuzzy set. For example, it should represent the absence of a protein or a low, medium, or high concentration of system elements. The level of abstraction defines the quantitative behaviour of the system, especially when the modeller uses more fuzzy sets to improve granularity of discretisation. However, in some cases, modellers can achieve the desired quantitative behaviour with only few fuzzy sets.
- **Detection limits and precision:** when concentration is quite low or extremely high, measurement methods cannot capture these concentrations, so fuzzy sets such as trapezoids can be used to cover a broad range of extremes and imprecisions. See first and last fuzzy sets in Figure 4.2)

Basically, each function μ: D-> [0, 1] is a fuzzy set included in a domain of discourse called D. These fuzzy sets are designed in several shapes. Here, we are investigating the most common fuzzy shapes such as triangle, trapezoid and Gaussian as shown in Figure 4.4.

The nature of the entity specifies the shape that is suitable to achieve desired results(Windhager, 2013). For example, triangle fuzzy sets shown in Figure 4.4a are used when results are quite sensitive to state changes, such as when small variations in the states of effectors can be propagated to their targets. Also, Trapezoid fuzzy sets shown in Figure 4.4b can be utilized to specify regions where the precise value of a state is not essential or irrelevant for a systems. The last common shape is Gaussian fuzzy set shown in Figure 4.4c. This kind of shape is typically used to represent noisy data. These data are centred on a specific (mean) value, such as when representing the change in the fold extracted from gene expression data.

(a) (b)



(c)

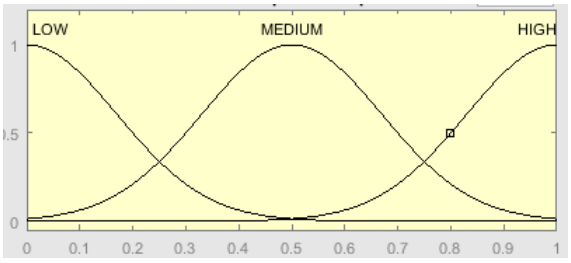


Figure 4.4: Most common shapes for fuzzy set representation. a) Triangular fuzzy sets with low, medium and high categories; b) Rectangular fuzzy sets with low, medium and high categories; c) Gaussian fuzzy sets with low, medium and high categories. The domain of discourse (x-axis) represents the range of the expected observations which in this example is the interval [0, 1].

These fuzzy sets need a real number (R) as the domain of discourse. For example, a biological entity's property, such as protein concentration, can generally be mapped into

the domain of discourse with real numbers; and this provides a quantitative result. Next section investigates how fuzzy sets interpret protein concentrations.

4.2.5 Fuzzy Sets Describe the States of Biological Entities

Every element in a biological system, such as a protein, has a specific state concerning its properties. For example, a protein has a location in a cell or its compartments. In addition, its concentration shows the current phase in the cell cycle. These properties can be measured and exposed from experiments. These properties provide us with qualitative or quantitative results that describe the state of an entity, such as protein measurements from fluorescence, which provide intensity levels. These measurements must be converted and estimated into molar concentrations. Another example is the studies that provide visual assessments of cells, which can indicate cell phenotypes. This information is qualitative.

Indeed, biological system behaviour can be explained using their qualitative description with the help of IF-THEN rules using 'natural language' expressions. This description does not need specific mathematical functions. For example, protein concentration can be described in terms of low, medium and high. The values for protein concentration must be normalised to intervals such as [0, 1]. The linguistic values of low, medium, and high can be mapped inside the interval [0, 1] as shown in Figure 4.5.

The above concentration interval can be divided into three different intervals, such as [0, 0.25], [0.25; 0.75), and [0.75; 1]. In Figure 4.5, the left part demonstrates the representation of the concentration in crisp set, and the input value (0.749) would be considered as high since it is located in the high crisp set. The right part of the same figure shows the fuzzy set representation of concentration. This makes the intervals more natural; for example, if we have a protein concentration value of (0.749), then fuzzy sets

can work with it in a more flexible manner, as we cannot say it is high, but it is high to a degree such as rather high.

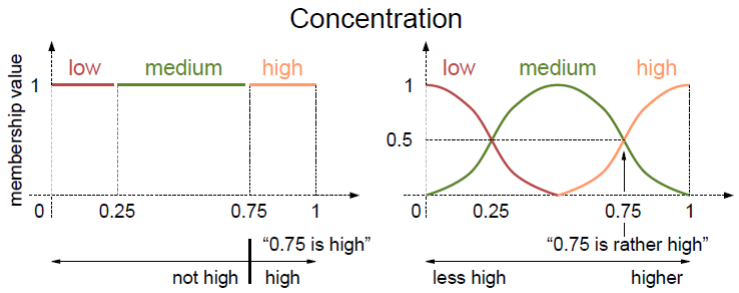


Figure 4.5: Main difference between crisp and fuzzy representation of concentration. The left part represents the crisp set while the right side represents the fuzzy sets. They represents species concentrations in three levels of low, medium and high. Also shown is an example of mapping a specific concentration value, such as (0.75), into membership functions.

In experimental measurements, we mostly get measurements of protein concentrations as shown in the left part of Figure 4.6 (Windhager, 2013). The domain of discourse can have values between [0, 100] in (%) unit. This time course measurement is represented in the fuzzy sets as shown in the right part of Figure 4.6, where the interval [0,100] is discredited using three fuzzy sets to represent the states of the concentrations. The fuzzy sets are low, medium and high must be interpreted according to the domain of discourse D. the membership function μ of a fuzzy set represents all states within the interval [0, 1]. Each x has a specific membership value. This value determines the degree of membership of x in the fuzzy set as represented in Equation 4.9 where D is the discourse.

$$\mu: D \rightarrow [0;1] \qquad (4.9.)$$

The right part of Figure 4.6 also shows the mapping process between the y-axis that represents the membership values and the x-axis that represents the relative concentration. The mapping process called fuzzification, this process is responsible for mapping crisp values into membership values.

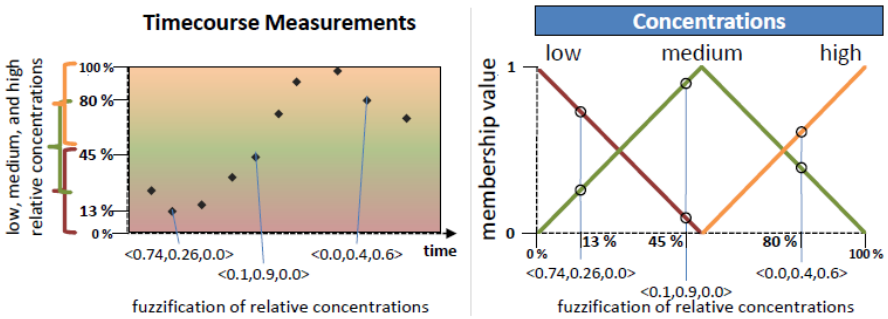


Figure 4.6: Mapping time course measurements in fuzzy logic: The left part shows some measurements for protein concentrations. These measurements occur in an unspecified time interval. In addition, the domain of discourse consists of all real numbers in the interval [0, 100] (% unit). Furthermore, fuzzy sets can discretise the aforementioned interval into three subsets. These subsets describe the protein concentration as low (μ low), medium (μ medium) and high (μ high) as shown in the right part of the figure. The fuzzy sets should map the relative concentration values on the x-axis to the membership values on the y-axis.

The fuzzification process produces a fuzzy results which typically have to be transformed into crisp output. To convert the fuzzy result into crisp values a defuzzification process should be performed. Precisely, defuzzification is the process of transforming the fuzzified result into a single crisp value with respect to a fuzzy set. There are several defuzzification methods are available such as center of sums method (COS), Maxima method, and center of gravity (COG)/ Centroid of area method (COA).

The most popular method is center of gravity, this method produces a crisp value based on center of gravity of the fuzzy set. In this method, the over-all area of the membership function distribution utilized to signify the combined control action is separated into a number of sub-areas. The area and the center of gravity or centroid of each sub area is computed and then the sum of entire sub areas must be taken to get the defuzzified value for a discrete fuzzy set.

In COG the defuzzified value for the discrete membership function is denoted by x* as represented by Equation 4.10:

$$x^* = \frac{\sum_{i=1}^{n} x_i \cdot \mu(x_i)}{\sum_{i=1}^{n} \mu(x_i)} \quad (4.10.)$$

Where x_i indicates to the sample element, μ(xi) is the membership function, and n indicates the number of the elements in the sample.

Further, the continuous membership function for x* is represented by Equation 4.11

$$x^* = \frac{\int x\, \mu_A(x)\, dx}{\int \mu_A(x)\, dx} \quad (4.11.)$$

All the information mentioned above is integrated into a fuzzy inference system details of which are presented in the next section.

4.3. Fuzzy Inference System

Essentially, the fuzzy inference system became popular for several reasons. For example, it offers the capability to convert data into knowledge, which makes it more understandable by people. Moreover, the system's behaviour can be represented using a set of rules that are constructed with expert human knowledge.

4.3.1. Fuzzy Inference System Functionality

Asmuni (2008) provides a detail description of fuzzy inference system (FIS); simply put, it is a process of mapping of an input to an output using fuzzy set theory. Fuzzy inference has several functional blocks, each representing a main stage. These stages are called fuzzification, inference, and defuzzification, as shown in Figure 4.7 (Asmuni, 2008). Initial data for the system is typically crisp values representing input and output. In the fuzzy inference system, the fuzzification process transforms all crisp input values into a fuzzy membership depending on how the membership functions are defined.

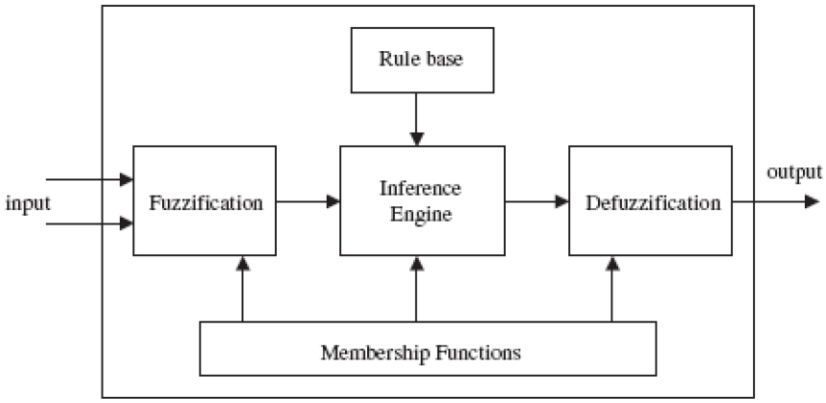


Figure 4.7: Fuzzy inference system with five functional blocks.

The fuzzy inference engine employs logical rules from the rule base block. It should also employ suitable fuzzy operators in order to process rules and reach an output. This stage utilises fuzzy logic operators such as “AND”, “OR” and “NOT” to reason with fuzzified inputs. Furthermore, the rules are processed and aggregated in this part. The output is reached in fuzzified form at the conclusion of fuzzy reasoning. Finally, fuzzy inference is responsible for transforming the final fuzzy output into a crisp output; this is undertaken using a specific defuzzification method. In summary, it can be seen from the above figure that the membership functions block is connected to most other blocks inside the fuzzy inference system and all blocks work together in a single process called fuzzy inference. More details of this proves are provided in the upcoming sections.

4.3.2. Fuzzy Inference Process

The essential part of fuzzy inference system is the fuzzy inference process (MATHWORKS, 2015), shown for an example system in Figure 4.8 that explains the process flow inside the fuzzy inference engine. For instance, we have a system that consists of two inputs, two IF-THEN rules, and one output. The process flow commences from the two inputs (crisp values) in the lower left of the figure. Then, these values go through a fuzzification process. The rows represent system rules that contain domain knowledge of the system and after

rules are processed the THEN part of the rules provides a fuzzified output. These are then defuzzified and averaged as shown in the figure into a crisp output.

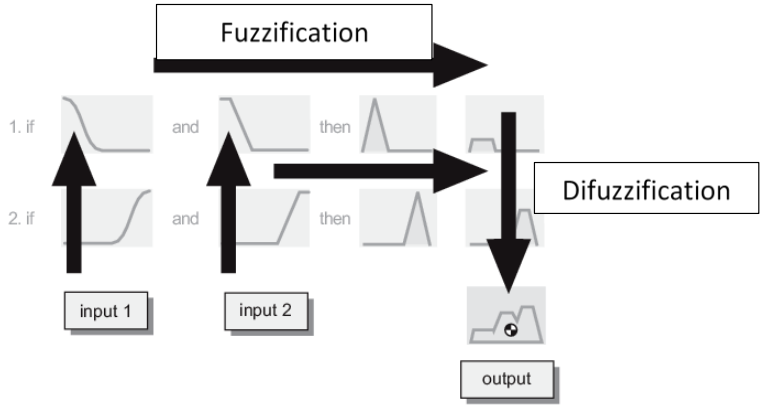


Figure 4.8: Fuzzy inference system process. Starting with crisp values, process goes through fuzzification of inputs into linguistic variables, followed by processing rules to provide fuzzy output that is defuzzified and aggregated into a crisp output.

There are two common methods used in fuzzy inference systems - Mamdani method and the Sugeno method (Negnevitsky, 2005). These methods rely on the use of simple language, such as in IF-THEN rules, to explain the essential system response as a function of several linguistic variables. The main difference between the two is in the specification of the consequent part (output). For instance, in Mamdani method, the consequent part is represented by fuzzy sets. Conversely, in the Sugeno method, the consequent part is represented by real numbers that can be either linear or constant; this means that the output takes on a different form to that in the Mamdani method - it is a singleton output membership function equivalent to the weighted average of each rule's output(Mamdani & Assilian, 1975). For example, a fuzzy rule of the first-order Sugeno method can take the form of Equation 4.12.

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z = ax + by + c \qquad (4.12.)$$

In this example, A and B are the fuzzy sets in the antecedent and the consequent is represented by mathematical linear function ax+by+c. The output is a function which can be constant when a = b = 0. This is known as the zero-order Sugeno model. However, the Sugeno model faces limitation in specifying effective coefficients. There is no good intuitive method for determining the coefficients in the Sugeno model.

In our model, we employ Mamdani method for the fuzzy inference system that is developed. Any Mamdani fuzzy inference system typically has the same functional blocks shown in Figure 4.7. If we have two crisp values (x1, y1) as inputs for the Mamdani fuzzy inference system, then the process should go through these steps:

- The first step in the Mamdani system is mapping the crisp inputs x1 and y1 into membership functions of the relevant fuzzy sets and determining the degree to which these inputs belong to each of fuzzy sets as shown in Figure 4.9. This figure shows fuzzy sets representing inputs and demonstrates how the crisp input values are mapped into membership values.

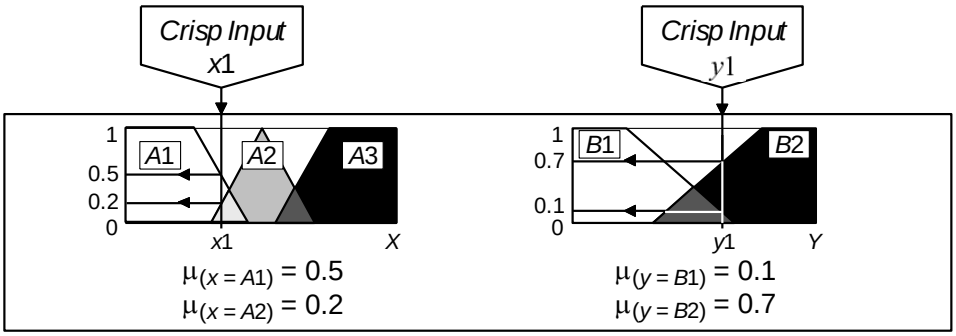


Figure 4.9: Mapping the crisp input values X1 and Y1 into fuzzy sets

The first membership function represents the X variable and consists of three subsets, A1, A2 and A3 respectively. The domain of discourse for input X is from 0 to X and includes all possible values of the variable X. The particular input X1 belongs to A1 fuzzy set with 0.5 membership and A2 with 0.2 membership. For variable Y, there are two membership

functions B1 and B2 with domain of discourse ranging from 0 to Y, which contains all possible values for Y. The value Y1 belongs to B1 fuzzy set with 0.1 membership and B2 with 0.7 membership.

- The second step is rule evaluation. In this step, the fuzzified inputs values μ(x=A1) = 0.5, μ(x=A2) = 0.2, μ(y=B1) = 0.1 and μ(y=B 2) = 0.7 are applied to the antecedent part in the fuzzy rules as shown in Figure 4.10. In this figure, each fuzzified input value is applied to the relevant fuzzy rule. Furthermore, in the case where we have a fuzzy rule that has multiple antecedents, the fuzzy operators ("AND" or "OR") are used to acquire a single value that represents the result of antecedent evaluation. In this example, we have three rules: in the first rule antecedent is compounded with an "OR" operator and "AND" operator joins the second rule. The result of antecedent processing is then mapped to the consequent membership function- the conclusion part of the fuzzy rule.

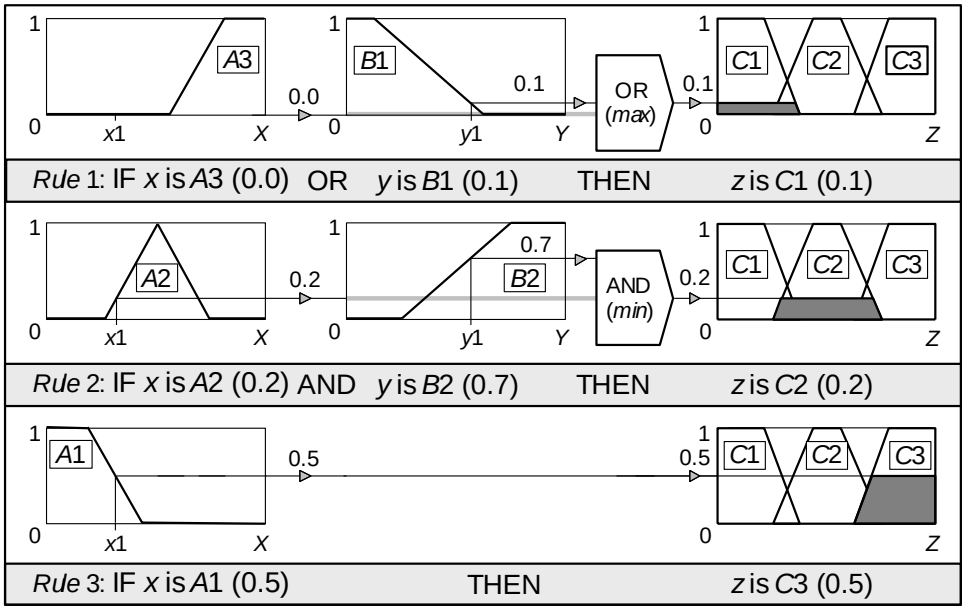


Figure 4.10: Mapping inputs into membership functions to obtain fuzzified inputs, fuzzy inference rule base and fuzzy inferencing

As shown in Figure 4.10, rule number one is

IF x is A3 OR y is B1 THEN z is C1 **(4.13)**

It consists of two parts that are grouped by the "OR" operator. In Rule 1, the condition part relating to x refers to fuzzy set A3 but the degree of membership of input value x1 in this set is 0. The second condition, which is related to y, refers to fuzzy set B1; so the degree of membership of y1 in this set is 0.1. As demonstrated before in Section 4.2.2, to evaluate the disjunction in the rule antecedents, we use "OR" fuzzy operation. In this case, fuzzy expert system typically utilises fuzzy Union operation to get the maximum value as shown in Equation (4.14):

$$\mu_{A\cup B}(x) = max[\mu_A(x), \mu_B(x)] \quad \textbf{(4.14)}$$

The result of this rule is the maximum value in the antecedent part and equals to 0.1. This is mapped to the relevant fuzzy set C1 of output z according to the rule resulting in membership of 0.1 in C1.

The second rule is

IF x is A2 AND y is B2 THEN z is C2 ***(4.17)***

This comprises two parts grouped by the "AND" operator. The first condition is related to x that belongs to fuzzy set A2, and the degree of membership of input value x1 in this set 0.2. The second condition is associated with y belonging to fuzzy set B2 and the degree of membership of input value y1 in this set 0.7. In order to evaluate the conjunction of the rule antecedents, "AND" fuzzy operator is used to find intersection represented by the minimum value as in Equation 4.18:

$$\mu_{A\cap B}(x) = \boldsymbol{min}[\mu_A(x), \mu_B(x)] \qquad \textbf{(4.18)}$$

Accordingly, the result of this rule is that the minimum value for the antecedent part equals to 0.2. Therefore, the consequent part z is C2 take membership value of 0.2.

The last rule is

IF x is A1 THEN z is C3 (4.19)

This rule has condition part related to fuzzy set A1 in and the degree of membership of input value x1 in this set 0.5. The consequent part relates fuzzy set C3 in z and so the resulting output is 0.5.

Next, aggregation process applies to all outputs from all rules. Here, all consequent membership functions from the rules are combined into an overall membership representation as shown in Figure 4.11.

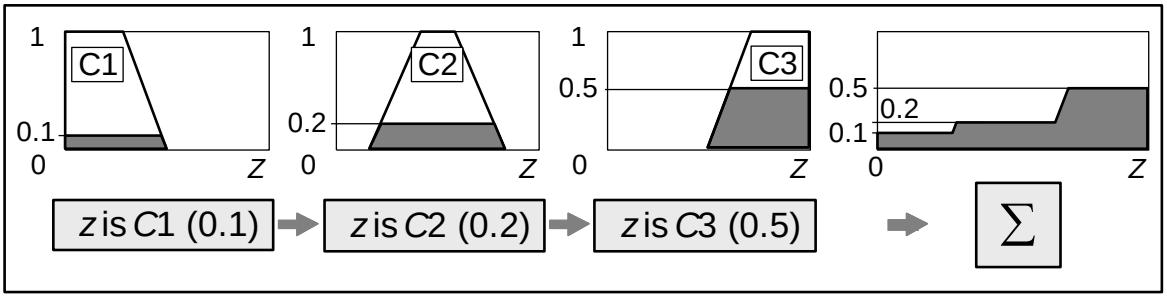


Figure 4.11: Aggregation process for consequents of fuzzy rules in the fuzzy inference system

Finally, the last step in the fuzzy inference process is defuzzification. Fuzziness affords benefits for evaluating the rules but the final output of a fuzzy system should be converted

to a crisp value. The input of the defuzzification method is the aggregated output fuzzy set and the output should be a single number.

In the literature, we can find several types of defuzzification methods, such as the mean of maxima (MOM), centre of gravity (COG), and the bisector of the area (BOA) (Cox, 1999). The most popular defuzzification method is the centre of gravity (COG). COG finds the point where a vertical line slices the aggregate set into two equal masses as shown in Figure 4.12.

The COG defuzzification method finds the centre point in the fuzzy set within a specific interval [a, b]. A reasonable estimate of this can be obtained by computing it over a sample of points. Therefore, we used COG method in the fuzzy model developed in this study. Further information concerning our model is given in Section 4.4.

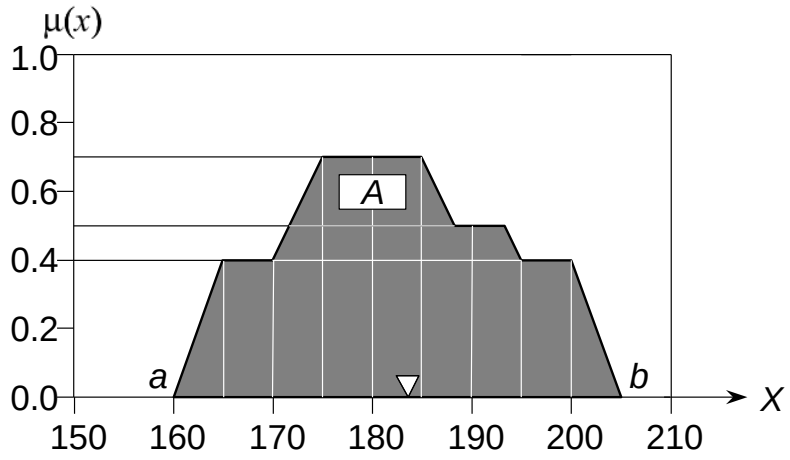


Figure 4.12: Operation of Centre of gravity (COG) method

4.4. Fuzzy Model of Cell Cycle Controller:

A promising computational model can be developed by transforming the qualitative knowledge and descriptions into an executable model. This can be done by adopting an appropriate mathematical representation of the discrete information. This includes

determining the most natural terms to represent states of all the entities in the system which then allows expression of the qualitative knowledge in natural language to represent the relationships (mathematical functions) existing in the system in a qualitative form to describe system function.

In this part of the chapter, we investigate the development of a fuzzy model of cell cycle controller. The development follows the process described above as discussed next.

4.4.1. Developing Fuzzy Logic Procedure:

Fuzzy approaches can play a fundamental role in merging biological processes with logical techniques for reconstructing the underlying biological system such as the cell cycle controller system. The nonlinear behaviour of proteins can also be explained by fuzzy models. This could be achieved by exploiting the extracted knowledge from the specialists or specialised knowledge bases, and then converting the data about protein relations into appropriate new fuzzy logic models. Fuzzy models use human reasoning to construct system rules for a complex system such as the biological system. This helps build comprehensive knowledge systems about regulatory mechanisms to explain their behaviour in an intuitive way using the semantics of fuzzy logic (Kong & Kosko, 1992).

Developing a powerful fuzzy system still requires a standard development procedure. Furthermore, when there is a need to develop a significant fuzzy system, the developer and biological specialists should work together with respect to the following procedures:

- Determine the case study problem and analyse the purpose/goal for the new model.
- Specify the linguistic variables that need to be represented by fuzzy sets in antecedent and consequent parts of rule. This should cover system activity or

observations. For example, fuzzy sets must describe protein concentrations as low, medium or high.

- Construct the fuzzy rules: these rules should appropriately control the activity of the system variables and interpret system behaviour including the interactions of the system components. For example, fuzzy rules in our case study should control the biological interactions between the species in the system. Furthermore, since the concentration levels are considered as indications of the activity of biological species, each change in individual protein level has an impact on other species in the system; so these rules must expose the impact of the change of protein levels on other species.
- Determine and employ appropriate methods of fuzzification, fuzzy inference and defuzzification.
- Evaluate the outcome of the developed fuzzy system and enhance the system outcome if required. This is the case where the fuzzy system produces an unacceptable result.

The implementation of this development procedure could provide an efficacious fuzzy system to explain, analyse and predict the behaviour of complex systems such as cell cycle controllers.

4.4.2. Properties Cell Cycle Controller Fuzzy Model

This section demonstrates the main properties of cell cycle controller in the fuzzy logic model. It also shows how we can develop a new semi-quantitative system. This system is deduced from the discrete model we developed in the previous chapter and its outcomes will be compared with the results of the ODE continuous model.

Each species behaviour is represented by a fuzzy inference system. We derive fuzzy inference for each cell cycle species from the Boolean model we developed (Chapter 3). In

addition, each fuzzy inference model mimics the ODE model functionality to interpret the relationships between the species activator(s) and degrade(s). For instance, the nonlinear behaviour of proteins can be explained by a fuzzy inference system similar to the ODE model. This depends on the effect of the concentration levels of a protein on activation or suppression of the level of a target protein. However, the developed fuzzy models use linguistic fuzzy reasoning using linguistic knowledge descriptions for species in the cell cycle model rather than the mathematical calculations as used in ODE equations.

Basically, each fuzzy inference system (FIS) has a set of input variable(s) and output variable(s). Any fuzzy inference system would have at least a single input variable and output variable; this is called the single input-output fuzzy inference system. Other kinds of fuzzy inference can have a set of inputs and one-output variable; this is called the multi-input single output system. Here the input variables from any fuzzy inference system represent the activators, degraders, or inhibitors of a specific element. The output variable represents the current state of an element affected by inputs as mimicked by this inference system. Here, all the fuzzy model's input variables are the same as those used in our Boolean network.

Although it is possible to extract best features from existing modelling platforms such as fuzzy logic toolbox for Matlab, which can provide approximate and realistic results, it is advisable to carry out experiments to choose the right shapes, number, locations and domain of discourse of fuzzy sets. In addition, it is necessary to choose optimal approaches to rule representation such as “AND”, “OR” operators as well as methods of aggregation and defuzzification.

The selection of a method should be based on which approach provides more accurate results. For example, we created a fuzzy inference system that mimicked the behaviour of SCF. This fuzzy inference system has two inputs and one output as shown in Figure 4.13. The first input is CycE and the second input is CDH1, and the universe of discourse for

protein concentration levels is then defined depending on observations from the available ODE model results for the same system. Furthermore, the fuzzy rules for this system have been built on the relations between these species and the impacts of the two inputs (CycE and CDH1) on SCF.

Table 4.1 shows the protein concentration measurements that are used as input data for the ODE model to calculate SCF values as outputs; this provided the SCF value. Also, these same input values would be used as inputs for our fuzzy experiments, and the results from the SCF fuzzy inference system will be compared with the SCF value from the ODE model. This is the validation process for the developed fuzzy model. The results from the SCF fuzzy inference model closest to the ODE results can help us select the optimal method for the inference system.

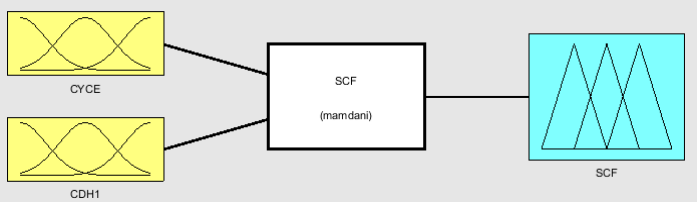


Figure 4.13: SCF fuzzy inference system structure: Two input variables and one output variable

We carried out six experiments as shown in Table 4.2. In each experiment, we made several attempts to make changes to the system, such as methods for processing “AND” or “OR” operators, implications, aggregation and defuzzification. We also experimented with different membership function shapes, such as rectangular, trapezoidal and sigmoidal. These changes were carried out within a heuristic design framework. We used our previous knowledge to design the systems that produce the most accurate results that should be close to the realistic values. Furthermore, we checked whether or not the shape and fuzzy set borders of the membership functions were appropriate for the system’s species. Finally, we selected the sigmoid membership function as the best since this shape can represent protein concentration levels, such as low, medium, and high, in the fuzzy sets more accurately, and describe the nonlinear behaviour of protein concentration measurements and activity. Moreover, the fuzzy parameters for each fuzzy set are adjusted

heuristically (manually) by using trial and error to achieve the best representation of species behaviour based on our biological knowledge.

Table 4.1 Example of ODE model input and output protein concentration measurements

ODE model- result		
CycE	Cdh1	SCF
0.001	0.9	0.1

Table 4.2 Six different experiments to specify the best method for designing the fuzzy inference system**.** In each experiment, we used the same input values for CYCE and CDH1 but we changed the methods used to calculate the SCF values

Fuzzy Logic Model – Experiment 1		**Fuzzy Logic Model – Experiment 2**		**Fuzzy Logic Model – Experiment 3**		**Fuzzy Logic Model – Experiment 4**		**Fuzzy Logic Model – Experiment 5**		**Fuzzy Logic Model – Experiment 6**	
AND Method	MIN	AND Method	MIN	AND Method	MIN	AND Method	PROD	AND Method	PROD	AND Method	PROD
OR Method	MAX	OR Method	MAX	OR Method	MAX	OR Method	PROBOR	OR Method	PROBOR	OR Method	PROBOR
Implication	MIN	Implication	MIN	Implication	MIN	Implication	PROD	Implication	PROD	Implication	MIN
Aggregation	MAX	Aggregation	MAX	Aggregation	SUM	Aggregation	SUM	Aggregation	SUM	Aggregation	SUM
Defuzzification	CENTROID	Defuzzification	MOM	Defuzzification	CENTROID	Defuzzification	CENTROID	Defuzzification	MOM	Defuzzification	CENTROID
CYCE	0.001	CYCE	0.001	CYCE	0.001	CYCE	0.001	CYCE	0.001	CYCE	0.001
CDH1	0.9	CDH1	0.9	CDH1	0.9	CDH1	0.9	CDH1	0.9	CDH1	0.9
SCF	**0.0874**	**SCF**	**0**	**SCF**	**0.088**	**SCF**	**0.0874**	**SCF**	**0**	SCF	**0.0893**

From the results in Table 4.2, we proposed methods used in Experiment 6 for our whole model.

Henceforth, the development process of the fuzzy logic model utilises this method from Experiment 6. The methods are product sum method and the centre of gravity method since they provide more accurate results and are close to realistic values. For each species in the system we developed a single fuzzy inference system. All fuzzy inference systems share the same principles in computation.

Additionally, we validate all individual fuzzy inferences systems which represent the cell cycle controller with realistic input values. These input values were extracted from the ODE model (Ali Abroudi, 2017). There were 390 records of these values representing the most precise continuous values of protein concentration over cell cycle phases. In this validation process, we make sure that each fuzzy inference system predicts the correct values for protein concentration that closely approximate result from the ODE model.

4.4.3 Fuzzy Inference Model for SCF, Design, Results and Discussion:

SCF Design

The SCF fuzzy inference system (FIS) mentioned above has two input variables and one output variable. SCF FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise SCF activity are shown in the BF in Eq. 3.15.

SCF, !CDH1 & CYCE (3.15)

SCF FIS works in an analogous manner to the SCF ODE model which is represented in Appendix 2, Equation 4.20:

The SCF (FIS) reflects the same functionality as that of the ODE model, but the fuzzy system can work without the need for kinetic parameters. In this way, the SCF fuzzy system can cover the limitations of the SCF ODE model. Both rely on the protein concentration levels to activate or degrade SCF protein level. The SCF fuzzy inference model should interpret the relations between SCF activator and degrader.

SCF FIS interprets the behaviour of SCF over the cell cycle phases as shown in the fuzzy rules in Table 4.3. The linguistic fuzzy reasoning is expected to provide accurate results using these linguistic knowledge descriptions for the SCF species in the cell cycle model as an alternative to using mathematical descriptions in ODE equations.

Table 4.3 Fuzzy rules for the SCF fuzzy model. The first column represents the rule number; and the second and third columns represent system inputs while the last column represents SCF status. Each row represents a rule that describes the impact of inputs on the SCF value. For instance, in Rule 1, when CycE concentration is low (low availability) SCF level also goes down and so on for the rest of the rules.

Rule number	CYCE	CDH1	SCF
1	L	-	L
2	-	L	M
3	L	L	L
4	H	-	H
5	L	L	L
6	-	H	L
7	-	L	H

Furthermore, the output variable of the SCF FIS model can describe SCF activity or concentration by employing three fuzzy sets (low, medium and high) as shown in Figure 4.14. and the fuzzy membership functions for inputs are presented in the next subsections.

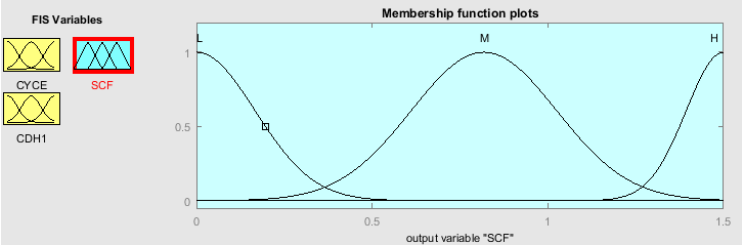


Figure 4.14: SCF output variable membership function design. This shows fuzzy sets, low, medium and high, and the universe of discourse [0, 1.5].

The fuzzy inference system development processes described previously were employed to develop the SCF FIS model. This model was validated with a realistic data set of around 390 records. These input values obtained from ODE model represent the most precise and continuous values of CycE and CDH1 over mammalian cell cycle. These values were used in conjunction with the rules (Table 4.3) in the SCF fuzzy inference system to compute the values of SCF over cell cycle phases as shown in Figure 4.15.

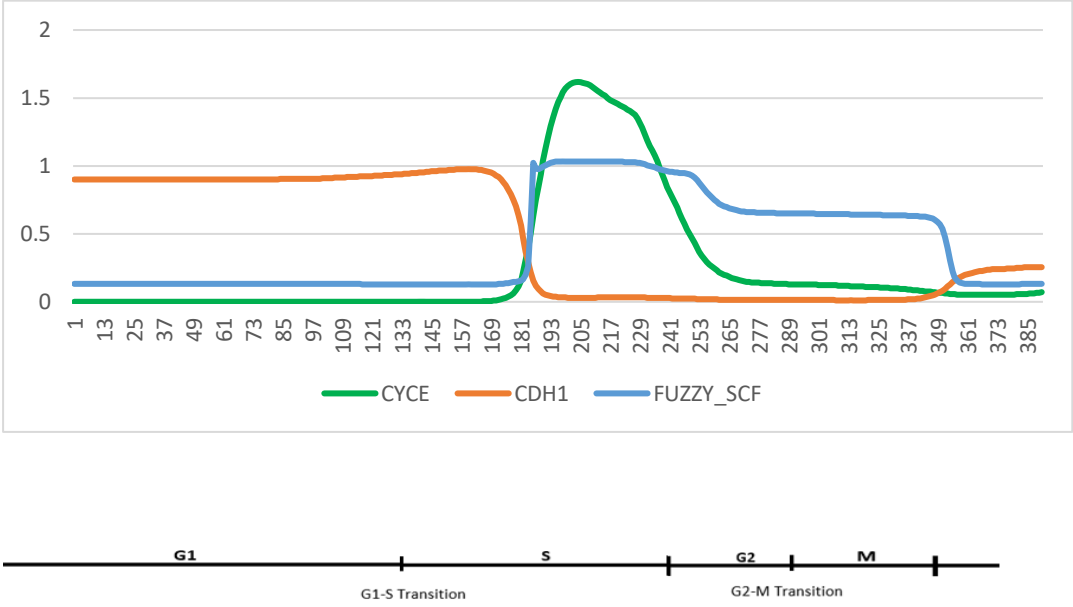


Figure 4.15: Fuzzy model outcome mimicking SCF activity over cell cycle. The green line represents CycE activity and the orange line represents CDH1 activity. These two are the input variables in the system. The blue line represents crisp values for SCF activity from the fuzzy system.

The figure above shows that at the inception point of the cell cycle and early in the G1 phase, CDH1 protein concentration is found at the highest level. This prevents SCF from accumulating. Moreover, CycE, activator of SCF, has a low level at the beginning of the cell cycle. Later in G1 phase, CDH1 protein level decreases gradually while CycE increases dramatically. These changes in activator and degrader cause SCF levels to go up for a period during cell cycle. The SCF protein level as shown in Figure 4.15 goes up from being low at the G1 phase to medium and then a high level in the S and G2 phases; this is considered the maximum for that protein level. This is due to the SCF activator being available while the degrader being at the lowest point of protein concentration. This arrangement keeps SCF level steady. However, when CDH1 becomes active again in M phase, its concentration increases leading to reduction of SCF concentration gradually to zero as shown in Figure 4.15.

SCF fuzzy system results were compared to those from the corresponding ODE model and a close agreement was found for the pattern of behaviour as shown in Figure 4.16. The whole range of SCF activities from the beginning to end and the temporal changes agree well with the ODE results. However, there is some variation between the two in terms of exact values. Our aim is to provide a simpler and easier model than the ODE model and is a close approximation of it. Therefore, this SCF could be enhanced to provide better results. Therefore, in the next Chapter, we attempt to optimise this model using one of the most popular autonomous optimisation methods, particle swarm optimisation (PSO).

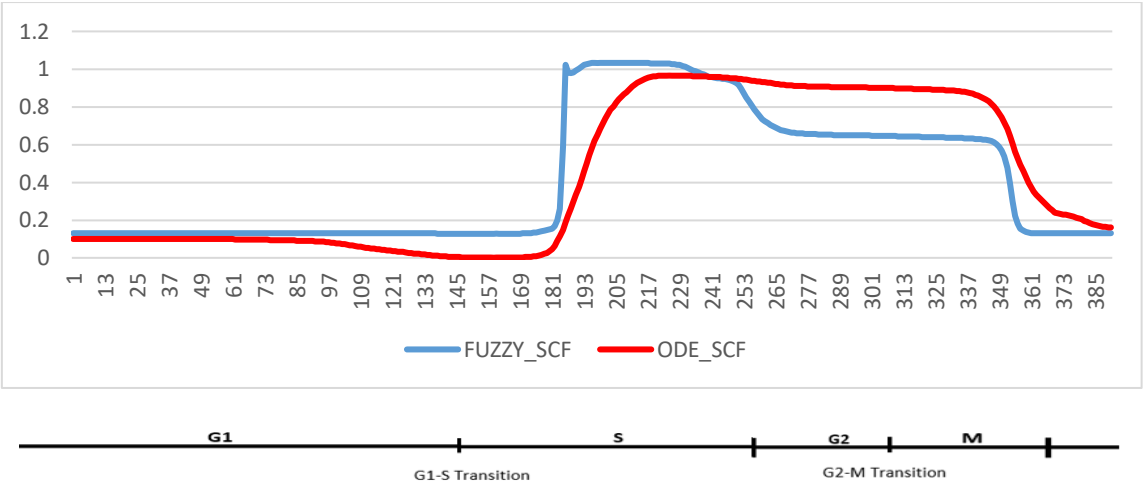


Figure 4.16: Comparison between SCF fuzzy model and ODE model outcomes for concentration of SCF in cell cycle. Protein concentration values are over the cell cycle phases. The blue line shows SCF values from the fuzzy model and red line shows SCF values from the ODE model.

4.4.4 CDH1 Fuzzy Inference Model, Design, Results and Discussion

- **FIS Design for CDH1**

CDH1 fuzzy inference system (FIS) has a set of input variables and an output variable (multiple inputs and single output (MISO)). As demonstrated earlier, CDH1 has activators and degraders similar to other cell cycle controller components. Here, the CDH1 fuzzy inference system consists of seven input variables and one output variable as shown in Figure 4.17. The input variables represent the activator(s) and degraders of CDH1 (FIS), while the output variable is the CDH1. CDH1 FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise CDH1 activity are shown in the BF in Eq.3.16.

CDH1= (!CYCA & !CYCB) & !CYCE | (CDC20A & CDC20B)|(P27 &!CYCB) (3.16)

Furthermore, CDH1 fuzzy inference system works in a similar way to the corresponding ODE equation which is represented in Appendix 2 (Equation 4.21).

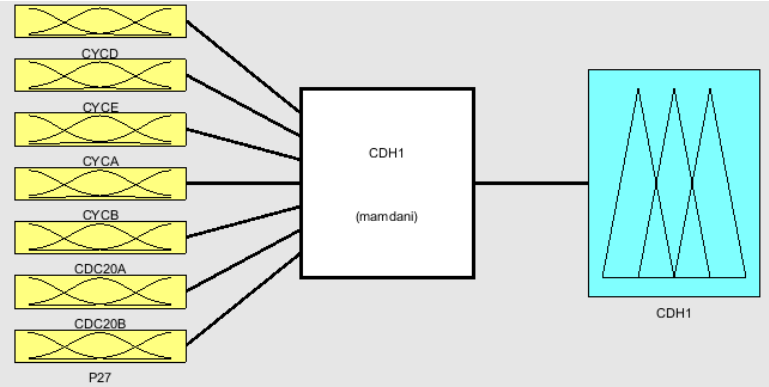


Figure 4.17: CDH1 fuzzy inference system structure. Seven input variables with one output variable

In addition, CDH1 FIS has a set of fuzzy rules that control the relations between the CDH1 and its activator and deactivator elements as shown in Table 4.4. These rules interpret the behaviour of CDH1 over cell cycle activity.

Table 4.4: Rules for the CDH1 fuzzy model. The first column represents the rule number, while the rest of the columns represent system inputs except the last column which represents the CDH1 status. Each row signifies a rule that describes the impact of inputs on CDH1 value. For instance, Rule 5 explains the fact that cyclins are not active while CDH1 is active (antagonist relation between cyclins and CDH1). This means that when CDH1 is at high concentration and high activity, cyclins are found in low concentrations, and so on for the rest of the rules.

Rule number	CYCD	CYCE	CYCA	CYCB	CDC20A	CDC20B	P27	CDH1
1	H	-	-	-	-	-	L	L
2	-	H	-	-	-	-	L	L
3	-	-	-	M	-	-	-	L
4	-	-	L	L	H	H	L	M
5	L	L	L	L	-	-	-	H
6	-	-	-	M	-	-	-	L

Each variable involved in the fuzzy inference system was described by fuzzy sets and associated membership functions. Fuzzy inference system provides an efficient and flexible

way to design and update the membership functions for the FIS variables as shown in Figure 4.18. In this section, we investigate only the CDH1 since the rest of the input elements are investigated individually in the coming sections. However, all of them share the fuzzy set format (Low, Medium, and High).

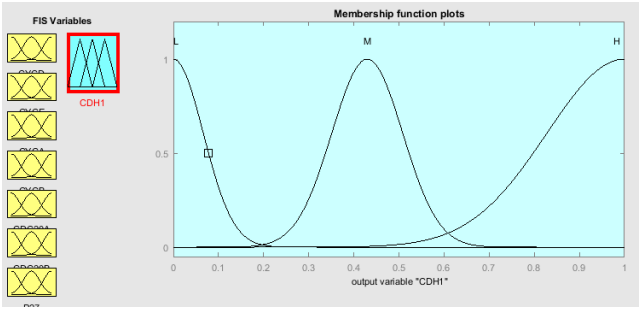


Figure 4.18: CDH1 output variable membership function design. This shows the fuzzy sets as low, medium and high and the universe of discourse. The left part shows the fuzzy sets for the inputs variables for the CDH1 FIS. The fuzzy membership functions for inputs are presented in the next subsections.

The dynamic behaviour can generally be explained by temporal variation of protein concentration levels. Consequently, the CDH1 output variable has three fuzzy subsets (low (L), medium (M), and high (H)). The universe of discourse for CDH1 is the interval [0, 1]. This universe of discourse includes all possible values for CDH1 concentration that can appear over the simulation.

- **CDH1 FIS Results and Discussion**

Both the ODE model and our developed fuzzy model depend on concentration levels of activators or degraders to determine the level of CDH1 protein concentration. Figure 4.19 shows the results from the FIS for CSH1. At the beginning of cell cycle, the CDH1 protein concentration is at a peak. Then, accumulation of cyclins over G1-S phases decrease CDH1

abundance. For example, CycD accumulation in mid-late G1 slightly inhibits CDH1 activity and lowers its concentration as shown in Figure 4.19.

In addition, CycE is considered as one of the most important inhibitors of CDH1 since they have a high antagonism. In late G1 phase, CDH1 protein level decreases steadily while CycE increases dramatically. This makes CDH1 level fall to a low after G1 phase as shown in Figure 4.19. Furthermore, CDH1 should stay at a low level since another cyclin, CycA, would be accumulating in the middle of the cell cycle (S-G2 phases). Another three species keep CDH1 concentrations at a low level in the M phase; these are CDC20A, CDC20B, and CycB. But the concentration of cyclins and other degraders decrease at the end of M phase. As a result, CDH1 protein starts to accumulate again with a rising level as shown in Figure 4.19.

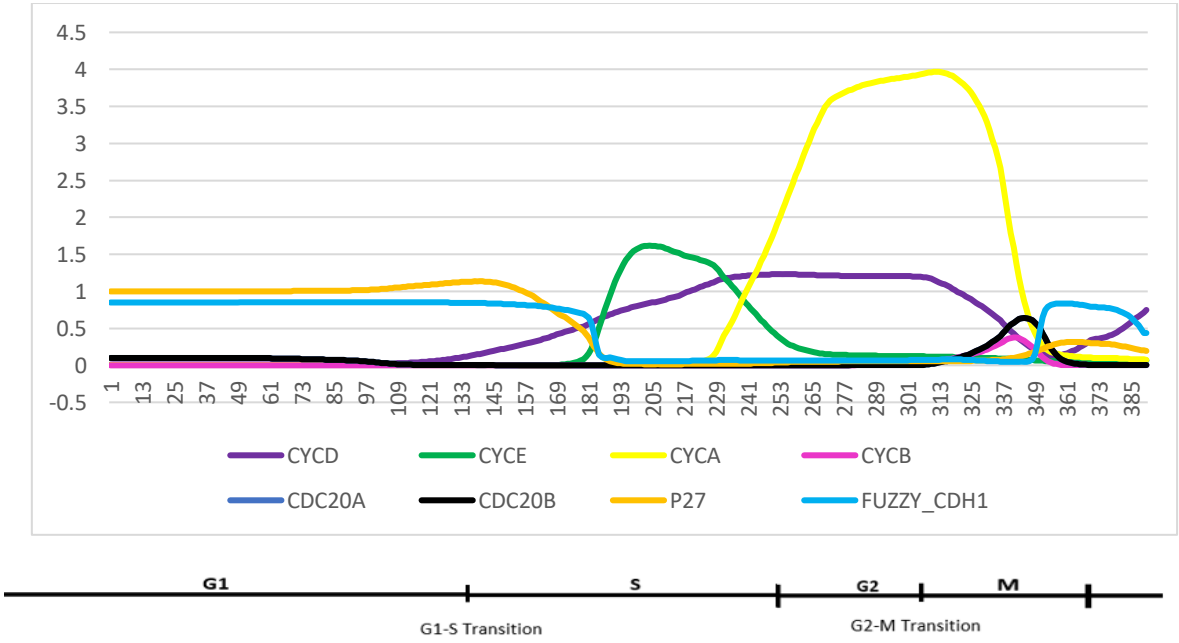


Figure 4.19: CDH1 fuzzy model outcome to mimic CDH1 activity in cell cycle. The blue line represents CDH1 activity (protein concentration) with crisp values from the fuzzy system. The other lines represent activity of the input variables.

CDH1 fuzzy system delivers an approximately similar result to ODE model as shown in Figure 4.20. At the start, CDH1 level from the fuzzy system is similar to ODE results as shown in this Figure. However, there is a substantial difference between the two at the end of M phase when the cell completes cell cycle. The fuzzy model provides the appropriate cyclic behaviour for CDH1 during the simulation; this makes the cell start a new cycle with the close peak level that was observed in the previous cycle while the ODE model failed to observe this starting behaviour in the new cycle. Indeed, our simpler CDH1 model provides results that approximate for the most part ODE model results.

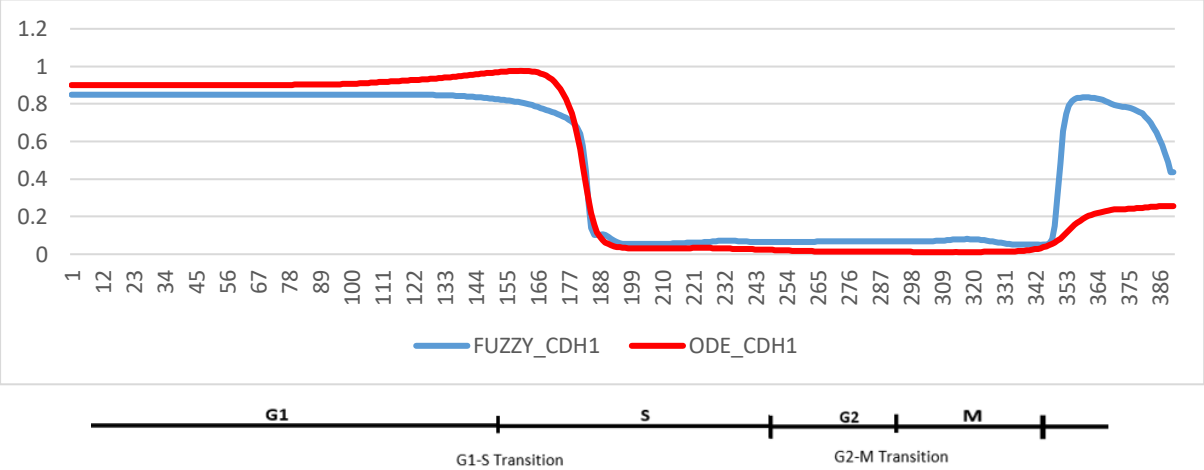


Figure 4.20: A comparison between CDH1 fuzzy model and the ODE model outcomes for CDH1 concentration in cell cycle over the cell cycle phases. The blue line shows CDH1 from the fuzzy system while red line shows CDH1 results from the ODE model.

4.4.5 P27 Fuzzy Inference Model, Design, Results and Discussion:

- **P27 FIS Design:**

The P27 fuzzy inference system has three input variables and one output. These three inputs impact P27 activity over the cell cycle phases, and they are P27 transcription factor, activator and inhibitor. The output variable signifies the values of P27 over the simulation as shown in

Figure 4.21. This output variable provides all the predictable approximate values for P27 over the cell cycle phases.

P27 FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise CDH1 activity are shown in the BF in Eq. 3.19.

P27 = (!CYCD & (!CYCD & ! P27)) & (!CYCD & !CYCE) |(!CYCE & !CYCA) (3.19)

As confirmed earlier, each fuzzy system can mimic a specific ODE equation. Therefore, the P27 fuzzy model mimics ODE model which is represented in Appendix 2, Equation 4.22, which estimates the activity of P27 in cell cycle phases.

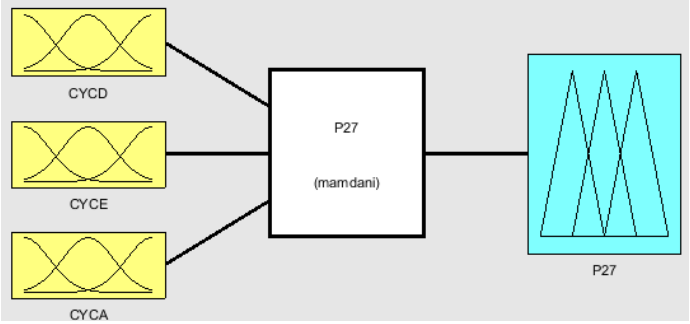


Figure 4.21: P27 fuzzy inference system structure. The three input variables with one output variable

Principally, P27 fuzzy inference system works in a like manner to the ODE model to interpret p27 relationship to its activator and degrader. P27 FIS has a set of fuzzy rules to explain the dynamic behaviour of P27 as shown in Table 4.5. These sets of fuzzy rules can reveal the relationships of each input variable(s) to the output variable relying on protein concentrations levels, and these rules were extracted from and are compatible with biological knowledge. For example, fuzzy rules invert the relationship between CycD, CycE, CycA, and P27 over the cell cycle. Furthermore, fuzzy rules demonstrate how cyclin protein levels have a negative

effect on P27 activity. The rise in cyclin protein production would increase cyclins levels. As a result, these species reduce P27 abundance in the cell cycle system.

Table 4.5: Fuzzy rules for the P27 fuzzy model. The first column represents the rule number, while the rest of the columns represent system inputs, except the last column, which represents P27 status. Each row indicates a rule that describes the impact of the inputs on P27. These rules interpret the biological knowledge related to P27 and other species in cell cycle. For instance, Rule 1 describes the fact that when cyclins are not active (low level), then P27 would be active at a high level. The second rule reflects the decrease in P27 concentration level when CycD starts accumulating and binding with P27,and so on for the rest of the rules.

Rule number	CYCD	CYCE	CYCA	P27
1	L	L	L	H
2	M	L	L	M
3	H	L	L	L
4	H	H	L	L
5	-	-	M	L

Concerning the P27 membership function design, P27 concentration values are classified into three fuzzy sets (low (L), medium (M), and high (H)) as shown in Figure 4.22. The fuzzy membership functions for inputs are presented in the other subsections

Based on the membership functions, the universe of discourse for this variable has values inside the interval [0, 1.2]. This interval denotes P27 concentrations that correspond to realistic biological observations. The borders for fuzzy sets (fuzzy set parameters) are also adjusted carefully by using heuristic design to provide the most accurate results. This form of fuzzy sets can provide approximate continuous results to ODE models for the dynamic behaviour of P27 over cell cycle system.

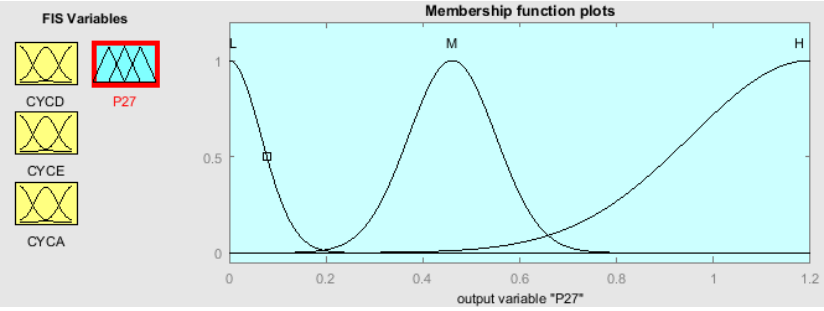


Figure 4.22: The membership function design of P27 output variable. It shows the fuzzy sets as low, medium and high over the universe of discourse.

- **P27 FIS Results and Discussion**

The P27 FIS model interpreted all P27 activity over cell cycle in a very realistic manner as shown in Figure 4.23. At the start of the cell cycle, P27 protein concentration can be found at the highest level. The main task for P27 is to keep the new born cell in G0 until the growth factor arrives to start cell cycle. However, cyclin abundance over cell cycle phases can influence P27 abundance and activity. For example, from the FIS results shown in Figure 4.23, CycD in G1 started synthesis and accumulated sharply to reach a peak, which affected P27 activity and reduced the available P27 protein level. In addition, CycE has a negative impact on P27 activity. At a late point in the G1 phase, P27 protein level decreases gradually to reach the lowest point. Thereafter, it remains at the same low level for most of the cell cycle. This is because P27 should stay at a low level while another cyclin (CycA) accumulates in the middle of the cell cycle. When concentration of cyclin species that control P27 activity become low, P27 level starts to rise and accumulate again. This increases P27 level gradually at the end of cell cycle as shown in the Figure 4.23.

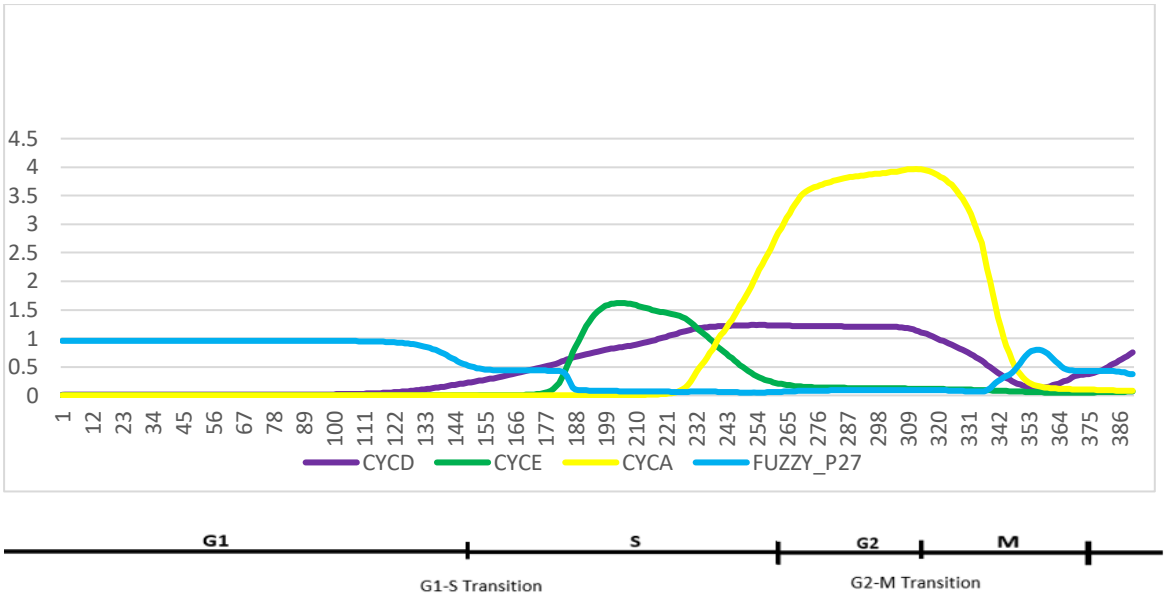


Figure 4.23: P27 fuzzy model results that mimic P27 activity during cell cycle. The blue line represents P27 activity (protein concentration) in crisp values from the fuzzy system. The other lines represent the input-variable activity over cell cycle phases.

The P27 fuzzy system offers realistic results close to those from the ODE mode (Figure 4.24). Regarding P27 activity, the onset and the change of P27 over cell cycle phases have been described similarly to ODE as shown in Figure 4.24. Hence, our attempts aimed at providing a simpler and easier model with accurate results comparable to ODE models have been successful. We realised that our P27 FIS model is similar to the last CDH1 FIS model in that P27 FIS represents initial p27 condition at the beginning of the next cell cycle better than the ODE model which failed to represent this behaviour as shown in Figure 4.24.

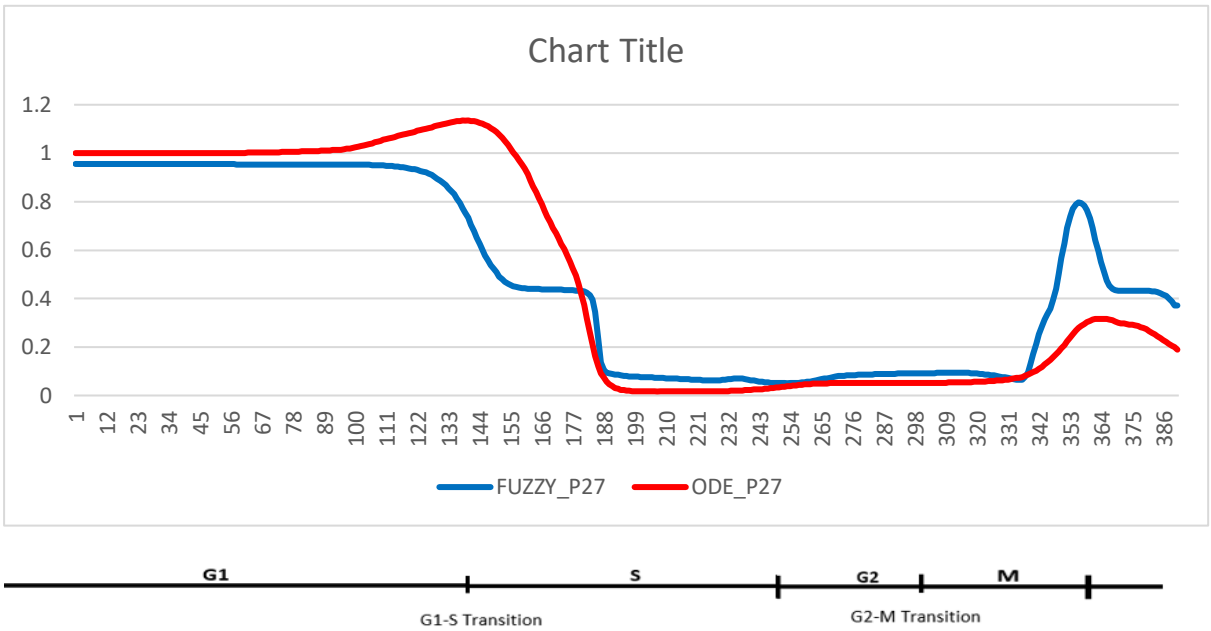


Figure 4.24: Comparison between P27 fuzzy model outcome and P27 ODE model outcome for P27 activity in cell cycle. The blue line indicates p27 from the fuzzy system whereas the red line represents p27 activity from the ODE model.

4.4.6. CycD Fuzzy Inference Model, Design, Results and Discussion:

- **CycD FIS Design**

The CycD fuzzy inference system is considered a multi-input and single output system. It has four input variables and one output. These four inputs influence CycD activity over the cell cycle phases. The output variable represents the values of CycD over the simulation as shown in Figure 4.25. CycD FIS was derived from the Boolean model we developed (Chapter 3), The Boolean functions which organise CycD activity are shown in BF 3.8.

CYCD, MYC & (!SCF & (!CDC20A & !CDC20B)) (3.8)

Essentially, CycD FIS aims to mimic the ODE equation model for CycD over the cell cycle, shown in Appendix 2, equation 4.23.

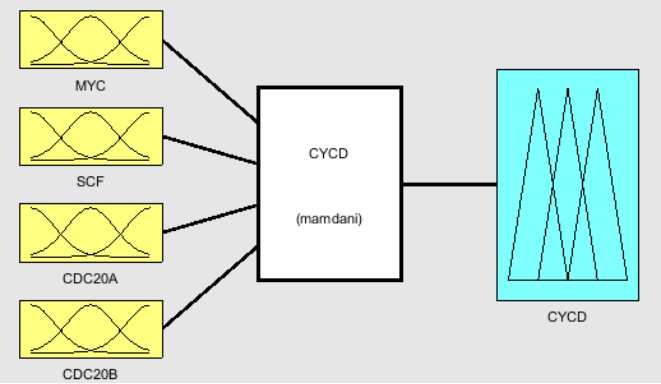


Figure 4.25: CycD fuzzy inference system structure. Four input variables with one output variable.

Basically, CycD membership functions consist of three fuzzy sets (low (L), medium (M) and high (H)), and the universe of discourse for this variable is the interval [0, 1.4] as shown in Figure 4.26. The fuzzy membership functions for inputs are presented in the other subsections. These values represent all possible CYCD concentration values; these values are generated from the ODE model. The parameters for fuzzy sets are also tuned carefully to provide the most accurate results. With this form of fuzzy sets and parameter adjustment, it is hypothesised that a close result or the same patterns of dynamic behaviour for CycD over the cell cycle system as the ODE model would eventuate.

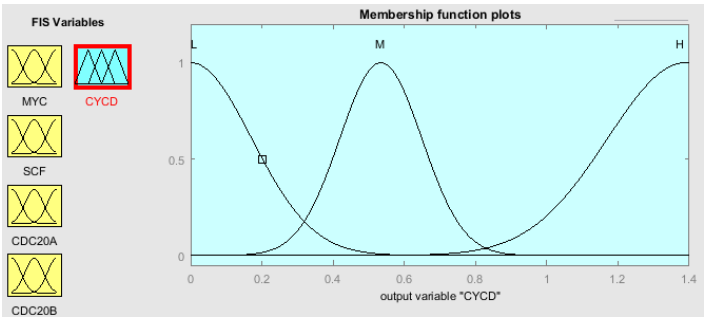


Figure 4.26: CycD output variable membership function design. It shows the fuzzy sets; low, medium and high over the universe of discourse [0, 1.4].

CycD FIS also has a set of fuzzy rules to elucidate the dynamic behaviour of CycD as shown in Table 4.6. This set of fuzzy rules can capture the relationship between input variable and the output variable. For example, the fuzzy rules demonstrate how protein levels have either a positive or negative impact on CycD activity. Increased levels of SCF, CDC20A, and CDC20B can reduce CycD abundance in the cell cycle system whereas increased levels of MYC increases it. Accordingly, fuzzy rules reveal the Inverse relationship between MYC and SCF, CDC20A, and CDC20B over time in the cell cycle system.

Table 4.6: Fuzzy rules for CycD fuzzy model. The first column represents the rule number, while the rest of the columns represent the system inputs except the last column, which represents CycD status. Each row indicates a rule that defines the effect of the inputs on CycD. For instance, Rule 2 explains the fact when growth factors are present and activate MYC to a high level, CycD reaches a high level in the absence of other degraders (*i.e.*, low level of SCF, CDC20A and CDC20B), and so on for the rest of the rules.

Rule number	MYC	SCF	CDCC20A	CDC20B	CycD
1	M	L	L	L	M
2	H	L	L	L	H
3	H	H	H	H	L
4	L	L	L	L	L
5	L		H	H	L
6		L	M	M	L
7		H	L	L	H
8		H	M	M	L
9	!L	L	L	L	L

- **CycD FIS Results and Discussion**

The CYCD (FIS) functionality is shown in Figure 4.27 and it follows the results from the CycD ODE model. Both depend on protein concentration levels to activate or degrade CycD and

influence its abundance. The outcome from the CycD FIS model is acceptable. For example, supporting the available knowledge presented earlier, at the start of the cell cycle CYCD level is at the lowest (Figure 4.27). Then, when MYC becomes active, this causes CycD synthesis/production to start and CycD accumulates gradually. As a consequence, CycD moves the cell from the G1 phase to the S phase. In this stage of the cell cycle, CycD remains at a high concentration level. However, the CycD abundance is affected by SCF, CDC20A, and CDC20B that reduce the available CycD protein as shown in Figure 4.27. At late G2 phase, CycD protein level decreases gradually to the lowest level as shown in Figure 4.27 so that it remains low at the start of the next cycle.

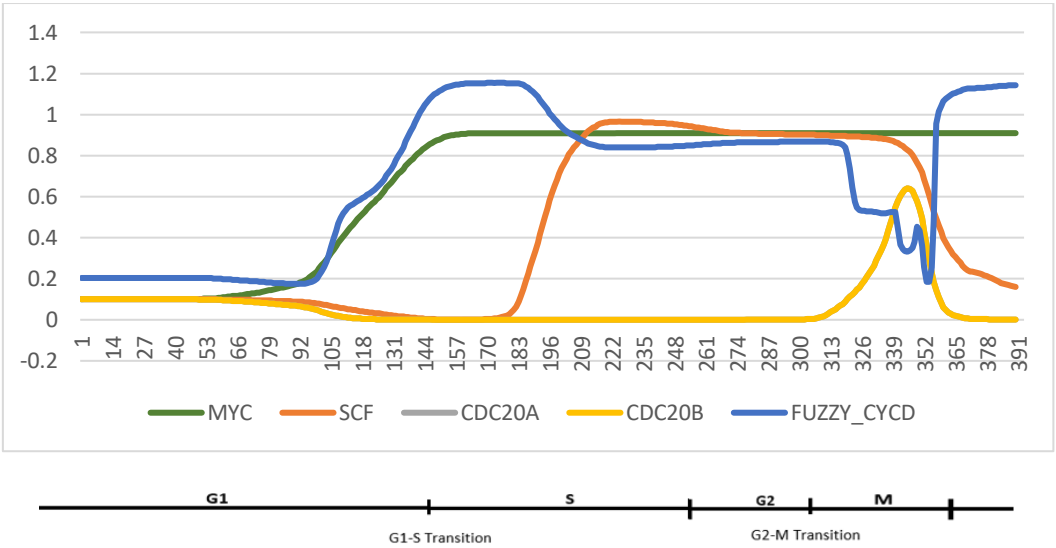


Figure 4.27: CycD fuzzy model results simulating CycD activity over the cell cycle. The blue line represents CycD activity (protein concentration) in crisp values from the fuzzy system. The other lines represent input variables values and activity over cell cycle phases.

CycD fuzzy system provides generally similar behaviour to that of ODE model as shown in Figure 4.28. The start and the change of CycD over the cell cycle phases have been described similar to ODE. However, there are some differences in CycD abundance level and therefore FIS system can be improved to provide better results. So, we attempted to optimise this model by using particle swarm optimisation (PSO) in a similar way to the SCF FIS model; further information on this can be found in the next chapter.

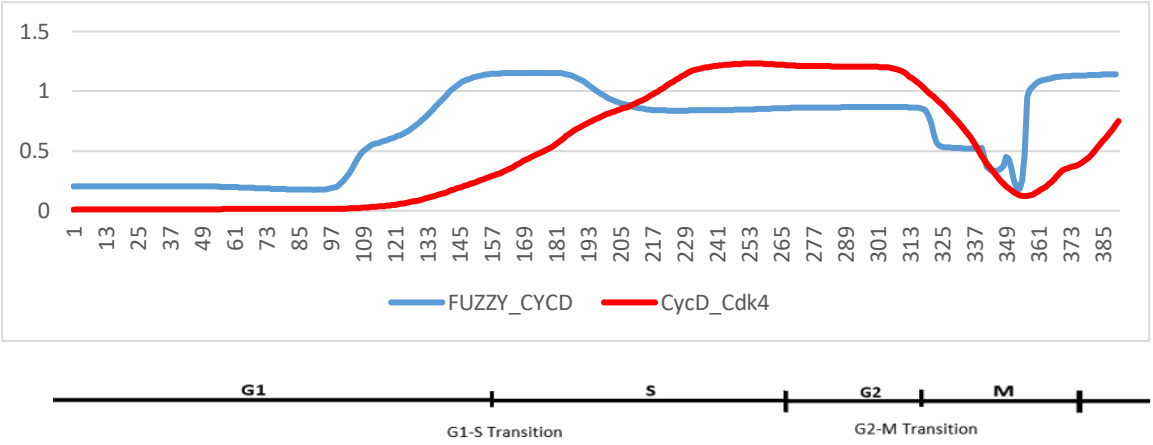


Figure 4.28: Comparison between CycD fuzzy model outcome and CycD ODE model outcome for CycD concentration throughout the cell cycle. The blue line indicates fuzzy system computed CycD whereas the red line show ODE model results for CycD.

4.4.7 TFE Fuzzy Inference Model, Design, Results And Discussion:

- **TFE (FIS) Design**

The TFE FIS model has four input variables and one output variable. Rb, CycA, CycB and P27 are the input variables while TFE is the output variable as shown in Figure 4.29. TFE FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise TFE activity are shown in the BF in Eq. 3.12.

TFE, (!RB & !CYCA & !CYCB) | (P27 & !RB & !CYCB) (3.12)

TFE FIS has been designed to mimic the TFE ODE model which is represented in Appendix 2, Equation 4.24.

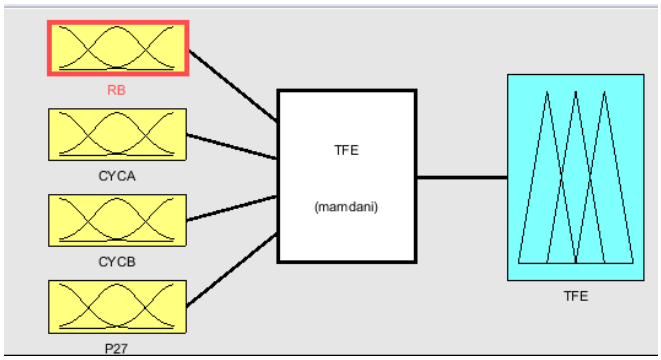


Figure 4.29: TFE fuzzy inference system structure. There are four input variables with one output variable.

The TFE membership functions have been designed in a similar way to those for other species we have studied so far. They consist of three fuzzy subsets (low (L), medium (M) and high (H)) over the universe of discourse [0, 1.8] covering all possible TFE values in cell cycle as shown in Figure 4.30. The fuzzy membership functions for inputs are presented in the other subsections.

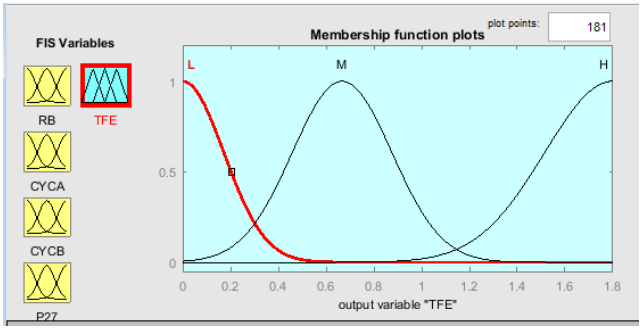


Figure 4.30: TFE output variable membership function design. This shows the fuzzy sets, Low, medium over the universe of discourse [0, 1.8].

Furthermore, this FIS model involves a set of fuzzy rules, which explain the biological interactions between TFE and its TFE activator(s) and degrader(s) as shown in Table 4.7. These fuzzy rules capture system element activity in terms of the status of the protein concentration levels for each input variable and their influences on the target TFE protein concentration level.

Table 4.7: Fuzzy rules for TFE fuzzy model. The first column represents the rule number, while the rest of the columns represent the system inputs except the last column, which represents the TFE status. Each row indicates a rule that defines the effects of inputs on TFE value. For instance, Rule 2 explains the fact when RB and P27 are at a moderate level and in the absence of degraders CycA and CycB, TFE remains low; and so on for the rest of the rules.

Rule number	RB	CYCA	CYCB	P27	TFE
1	H	L	L	H	L
2	M	L	L	M	L
3	L	L	L	L	H
4	L	H	L	L	L
5	M	-	M	L	L
6	L	M	L	L	M
7	-	M	-	-	L

- **TFE Results and Discussion**

The simulation shows that RB negatively impact TFE since it is the strongest inhibitor of TFE; therefore, at the beginning of cell cycle, RB has high concentration levels, while the rest of cyclins are at a low level. This leaves TFE at a low level at the beginning of cell cycle as shown in Figure 4.31. However, the purpose of CycD accumulation at the beginning of cell cycle is to form CycD_Cdk4 that in conjunction with the other cyclins in the system facilitates the release of TFE from RB. This leads to accumulation of TFE and TFE induced production of CycE and CycA as shown in Figure 4.31. When released, TFE keeps gradually increasing until the maximum accumulation but when the degraders, such as CycA, CycB and P27, start their accumulation, they degrade TFE levels over time as shown in Figure 4.31.

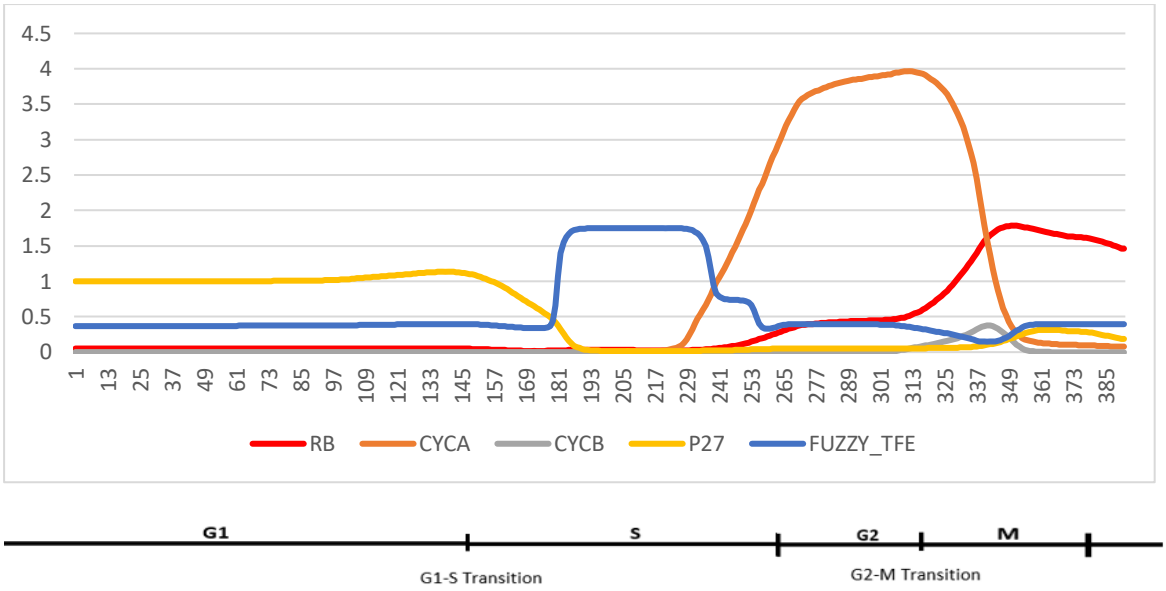


Figure 4.31: TFE fuzzy model results simulating TFE activity over the cell cycle. The blue line represents TFE activity (protein concentration) in crisp values from the fuzzy system. The other lines represent input-variable values and activity over cell cycle phases.

The previous figure shows that the TFE fuzzy model is an effective model since it provides the expected behaviour of all elements resulting from their interactions in the system. It represents the dynamic behaviour of TFE over the cell cycle well. There is high level of agreement between the fuzzy and ODE models for the TFE dynamic behaviour as shown Figure 4.32. However, there is a small variation between the two in terms of exact values.

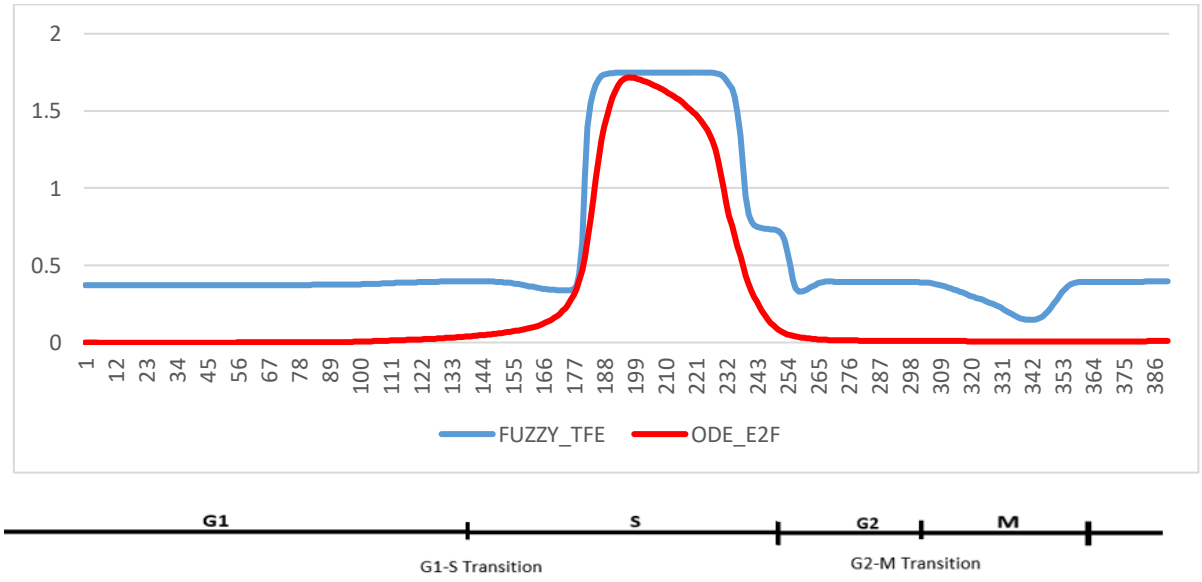


Figure 4.32: Comparison between TFE fuzzy model outcome and TFE ODE model outcome for TFE concentration and activity over the cell cycle. The blue line shows TFE from the fuzzy system whereas the red line represents TFE from the ODE model.

4.4.8 CycE Fuzzy Inference Model, Design, Results and Discussion:

- **CycE FIS Design**

The CycE protein was represented by its fuzzy inference system model. This model mimics CycE activity; it has three input variables and one output variable as shown in Figure 4.33. These input variables are the activators and degraders of CycE. As elucidated before, TFE is the activator of CycE while both SCF and P27 are degraders of CycE.

CycE FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise CycE activity are shown in the BF in Eq. 3.9.

CYCE, TFE | (!SCF & !P27) (3.9)

This fuzzy model should approximate the CycE ODE model which is represented in Appendix 2, Equation 4.25:

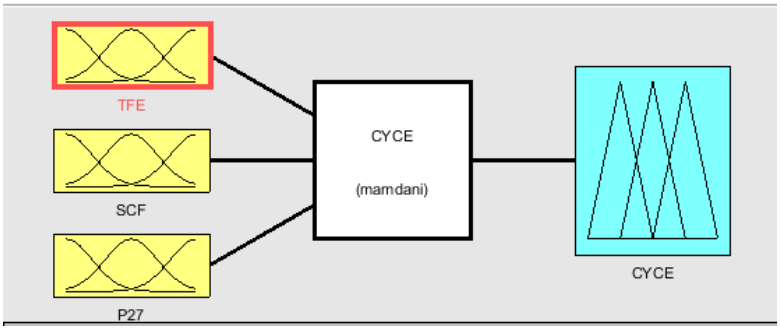


Figure 4.33: CycE fuzzy inference system structure. The three input variables and one output variable.

Comparable to other developed model designs, CycE variable was represented by three fuzzy sets for CycE concentration levels of low, medium and high. The membership functions are as shown in Figure 4.34. As before, model development involves a manual setup of fuzzy

parameters over the domain of discourse. The fuzzy membership functions for inputs are presented in the other subsections.

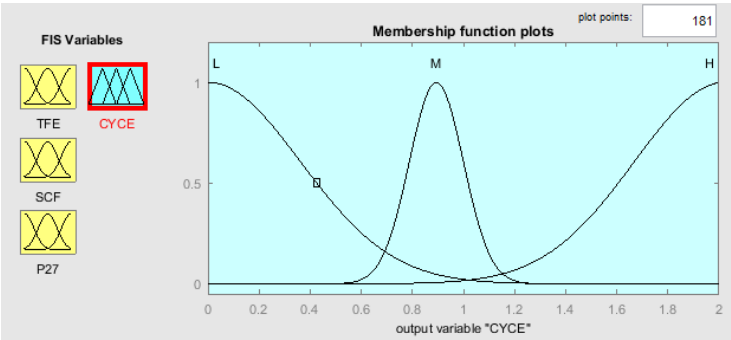


Figure 4.34: CycE output variable membership function design. It shows the fuzzy sets, low, medium and high over the universe of discourse.

CycE inference system also has a set of fuzzy rules to mimic CycE abundance over cell cycle. These rules depict the variable interactions that change species concentration levels as shown in Table 4.8.

Table 4.8: Fuzzy rules for the CycE fuzzy model. The first column represents the rule number, while the rest of the columns represent system inputs except for the last column, which represents CycE status. Each row indicates a rule that defines the effects of the inputs on CYCE. For instance, Rule 1 explains the fact that at early G1 phase, high level of P27 keeps CycE at a low level; and so on for the rest of the rules in the table.

Rule number	TFE	SCF	P27	CYCE
1	-	-	H	L
2	L	L	H	L
3	M	L	L	M
4	H	L	L	H
5	M	-	L	H
6	-	H	L	L
7	-	L	H	L
8	L	L	L	L
9	L	-	-	L
10	M	-	-	M
11	H	-	-	H

- **CycE FIS Results and Discussion**

The simulation revealed CycE activity over the cell cycle system as shown in Figure 4.35. For example, P27 should be reduced to a low level after initiating cell cycle since it keeps the cell in G0 as it prevents CycE production. TFE is also the activator for CycE production during the cell cycle. The TFE concertation should be at a low level at the beginning of the cell cycle but when TFE starts to accumulate, it causes CycE to be accumulated quickly. However, other CycE degraders become active appropriately to bring CycE level down. For instance, SCF concentration steadily decreases CycE protein level as shown in Figure 4.35.

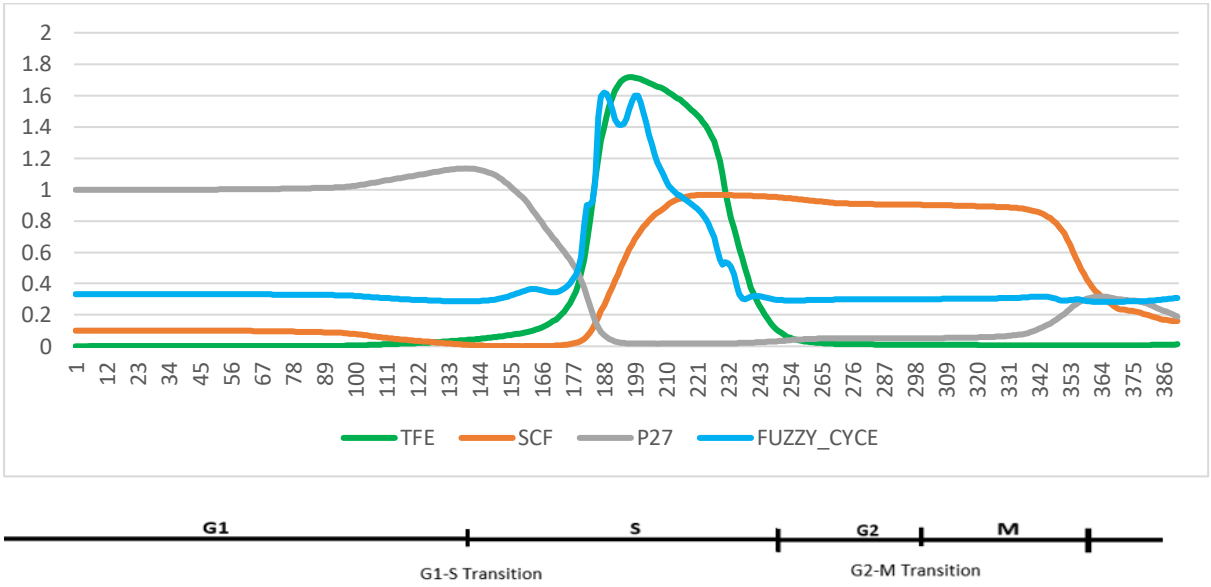


Figure 4.35: The CycE fuzzy model results simulating CycE activity through the cell cycle. The blue line represents CycE activity (concentration) in crisp values given by the fuzzy system. The other lines represent input variables and their activity over cell cycle phases

The results from the CycE fuzzy model and the ODE model were also compared. Indeed, the simpler fuzzy model can provide satisfactory results comparable to the ODE in describing the dynamic behaviour and protein concentrations as shown in Figure 4.36.

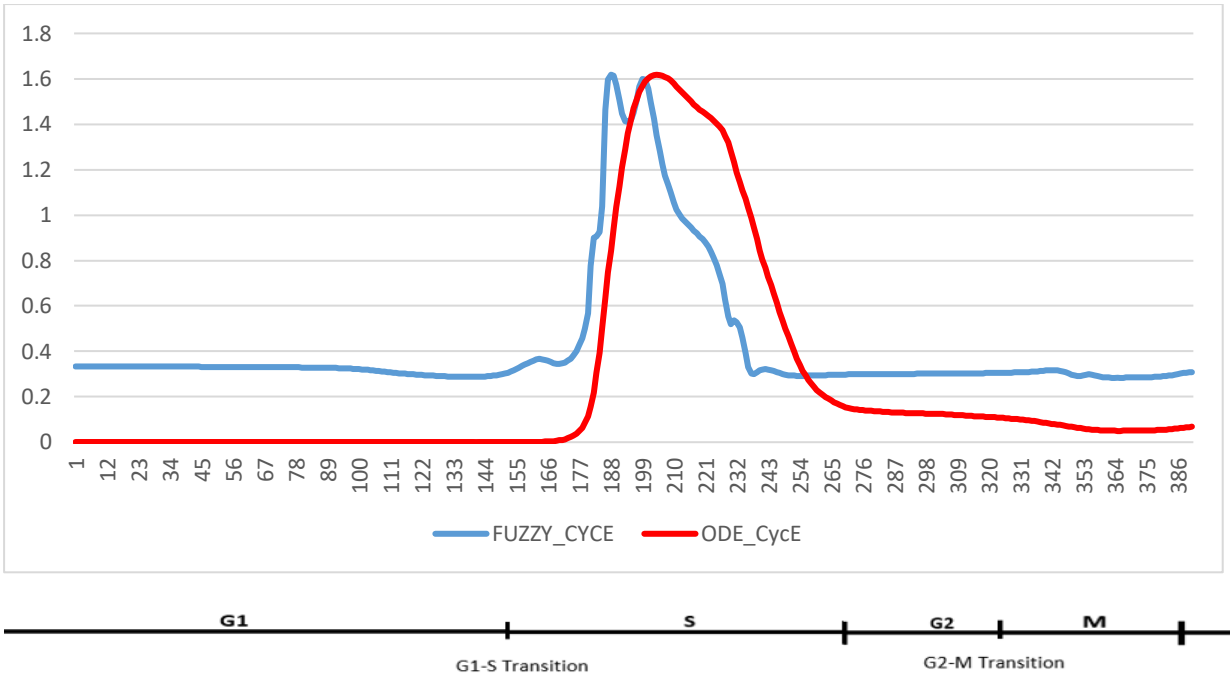


Figure 4.36: Comparison between CycE fuzzy model and CycE ODE model outcomes for CycE concentration and activity over the cell cycle. The blue line shows CycE from the fuzzy system whereas the red line represents CycE from the ODE model.

4.4.9 CycA Fuzzy Inference Model, Design, Results and Discussion:

- **CycA Design**

CycA activity has also been modelled using fuzzy inference system. The fuzzy inference system consists of four input variables and one output variable as shown in Figure 4.37. The input variables represent the activator and degraders of CycA activity over the cell cycle. It is known that the TFE variable has a positive impact on CycA abundance while rest of the elements, CDC20A, P27 and CDH1, negatively impact CycA abundance. Precisely, CycA FIS was derived from the Boolean model we developed (Chapter 3), The Boolean functions which organise CycA activity are shown in in BF 3.10.

CYCA, (TFE & !P27) | (!CDC20A & !CDH1) (3.10)

Correspondingly, this fuzzy inference model should mimic the CycA ODE model which is represented in Appendix 2, Equation 4.26.

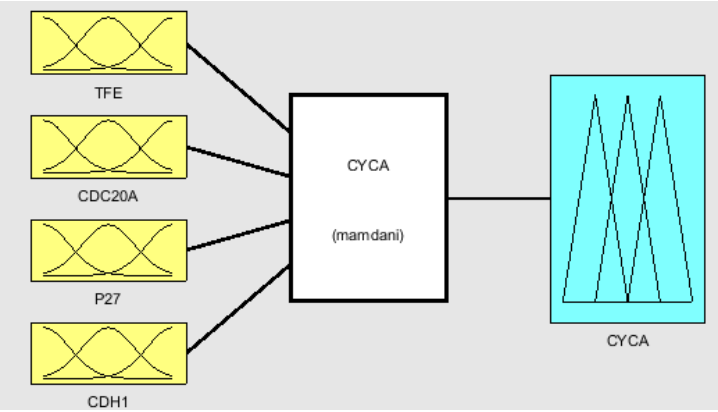


Figure 4.37: CycA fuzzy inference system structure. The four input variables with one output variable

The description for CycA protein concentration level is similar to that used in other system components. The CycA membership functions include three fuzzy sets (low, medium and high) as shown in Figure 4.38. The fuzzy membership functions for inputs are presented in the other subsections.

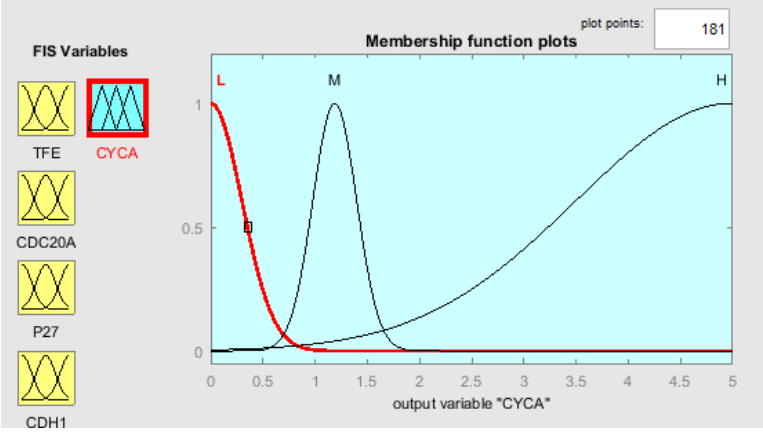


Figure 4.38: CycA membership function design. It shows the fuzzy sets, low, medium and high over the universe of discourse.

CycA fuzzy inference system rules describe the entire relationship between CycA and its activators and degraders. These rules are presented in Table 4.9.

Table 4.9: Fuzzy rules for the CycA fuzzy model**.** The first column characterises the rule number, while the rest of the columns characterise system inputs excluding the last column, which characterises the CycE status. Each row specifies a rule that describes the effect of inputs on CYCA value. For example, Rule 1 explains the fact that at the early G1 phase when P27 and CDH1 are at a high level while TFE is at a low level, CycA remains at a low level; and so on for the rest rules in the table.

Rule number	TFE	CDC20A	P27	CDH1	CYCA
1	L		H	H	L
2	M	L	L	L	H
3		H			L
4	H	NOT L	NOT L	NOT L	H
5	L		M	M	L
6			M	M	L

- **CycA FIS Results and Discussion**

Simulation results for the dynamic behaviour of CycA FIS elements are shown in Figure 4.39. For example, as described previously, both CDH1 and P27 have prevented CycA protein production at the beginning of the cell cycle, where CDH1 concentration level is at the highest as shown in Figure 3.49. P27 also keeps the new born cell in the G0 phase until receiving the growth factor and other elements to initiate cell cycle. For this reason, P27 should be at the highest level too at the beginning. Our fuzzy rules in the model are built on the aforementioned facts concerning the protein concentration levels and interactions.

Furthermore, we assumed that to start CycA production, both CDH1 and P27 should gradually reduce their concentration level from high to medium and, finally, to the lowest level of

concentration. This way, we are mimicking the transition from the G0 to the G1 phase. Afterwards, TFE can be active and stimulate the production of CycA protein, while CDC20 is available in the lowest amount. CycA steadily accumulates in the cell when TFE becomes active, but when CDH1 reaches the lowest point and is no longer active, CycA increases intensely and reaches the uppermost peak. However, the degrader of CycA, CDC20A, would be active after a while. This degrader would decrease CycA abundance steadily until it reaches the lowest level in the latter part of cell cycle.

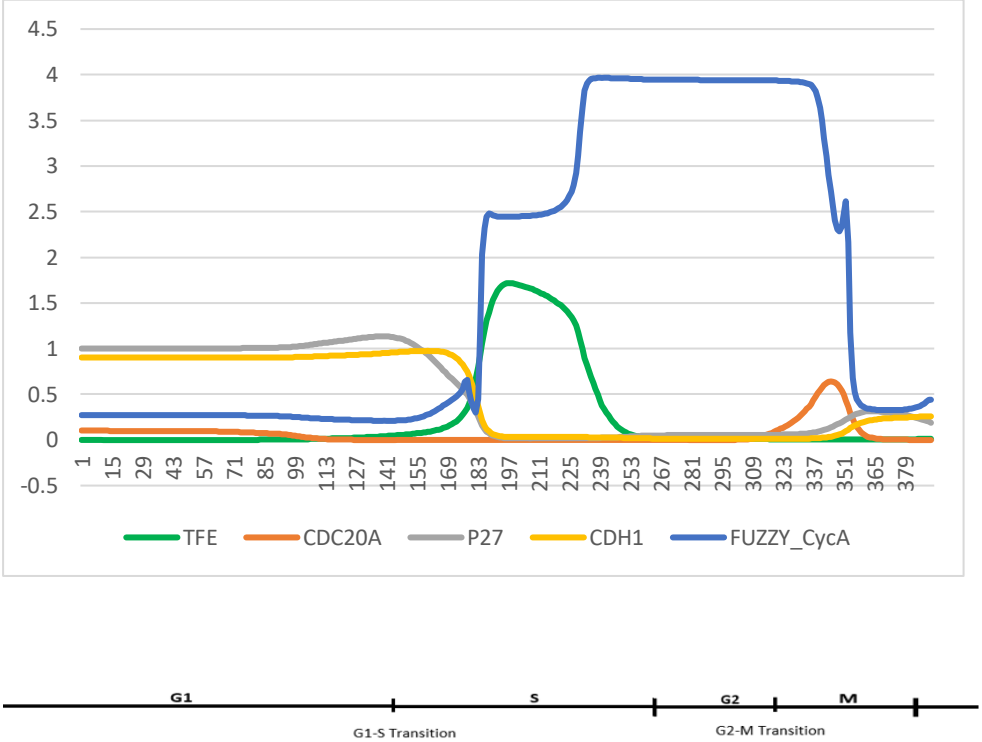


Figure 4.39: CycA fuzzy model simulating CycA activity through the cell cycle. The blue line represents CycA activity (concentration) in crisp values computed from the fuzzy system. The other lines represent input-variable values and activity over cell cycle phases

A comparison has been established between CycA FIS model results and CycA ODE results as shown in Figure 4.40. Although there is general agreement between the two models, in order for this FIS model to achieve better results, it is highly recommended that it be enhanced. The enhancement of this model can be done using PSO optimisation method; further details are explained in the next chapter.

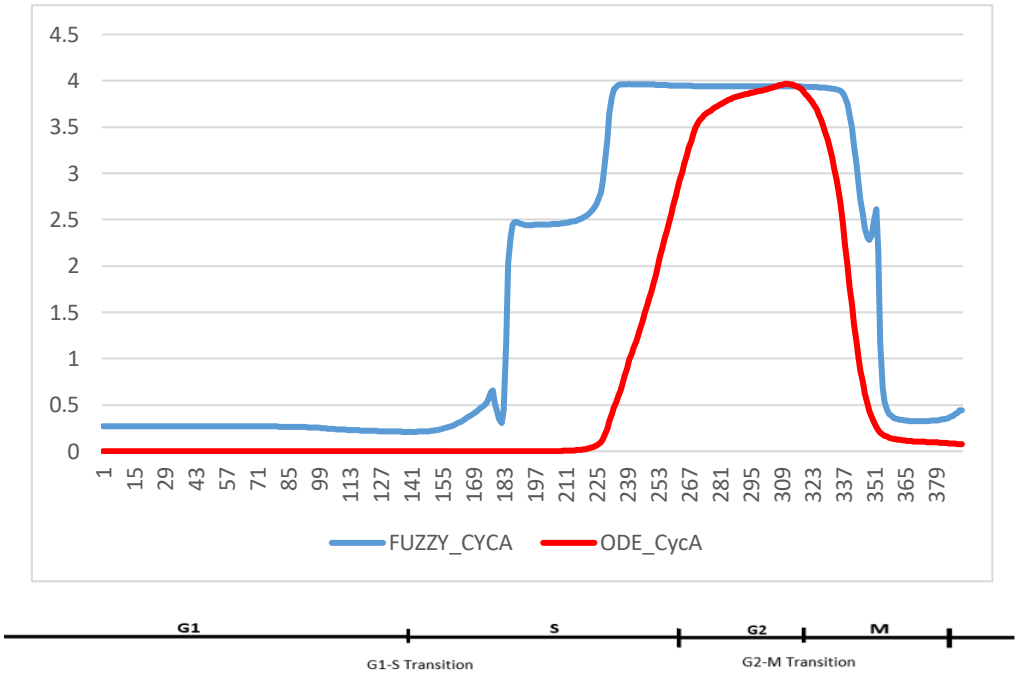


Figure 4.40: A comparison between CycA fuzzy model outcomes and CycA ODE model outcomes for CycA concentration and activity over the cell cycle. The blue line shows fuzzy system computed CycA values whereas the red line represents CycA from ODE model.

4.4.10 TFB Fuzzy Inference Model, Design, Results and Discussion:

- **TFB FIS Design**

The fuzzy inference system for TFB has a single input and a single output; the input variable is CycA while the output variable is TFB as shown in Figure 4.41. TFB FIS was derived from the Boolean model we developed (Chapter 3), The Boolean functions which organise TFB activity are shown in the BF in Eq. 3.13.

TFB, CYCA (3.13)

This fuzzy inference model should imitate the TFB ODE model which is represented in Appendix 2, Equation 4.27.

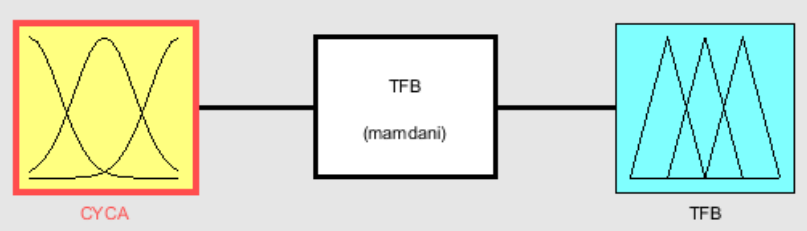


Figure 4.41: TFB fuzzy inference system structure. The single input variable and one output variable.

As confirmed earlier, we depend on heuristic design, so a heuristic design has been applied to design the TFB membership function and the domain of discourse. The design of the TFB variable has three fuzzy subsets (low, medium and high) as shown in Figure 4.42. The fuzzy membership functions for inputs are presented in the other subsections.

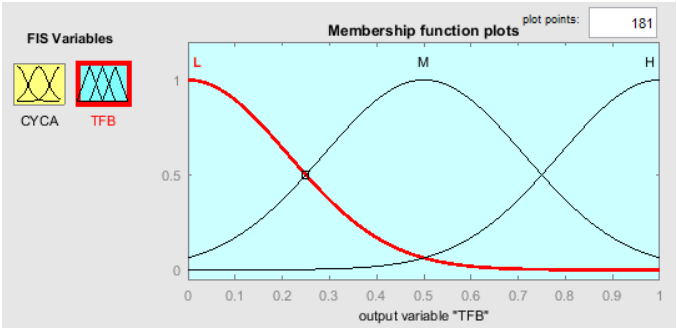


Figure 4.42: TFB output variable membership function design. It shows the fuzzy sets; low, medium and high, and the universe of discourse.

TFB FIS has a set of fuzzy rules which organise the TFB activity as presented in Table 4.10. TFB activity is related to CycA concentration level, and for this reason, the abundance of CycA can affect TFB production level as shown in Figure 4.42.

Table 4.10: TFB fuzzy rule set**.** The first column shows rule numbers in the system**.** The second column represents the input variables of the system, while the last column represents the output variable, TFB. Each row represents a different fuzzy rule, which explains how the input variables control TFB protein level in cell cycle. These rules describe species activity in cell cycle.

Rule numbers	Input 1 CycA	Output TFB
1	L	L
2	M	M
3	H	H

- **TFB FIS Results and Discussion**

Overall, the design information presented here generates an efficient fuzzy inference system to simulate the behaviour of the TFB as shown in Figure 4.43. As we can see, the fuzzy model reproduces the biological behaviour of TFB based on the biochemical reaction between the input variable CycA and TFB.

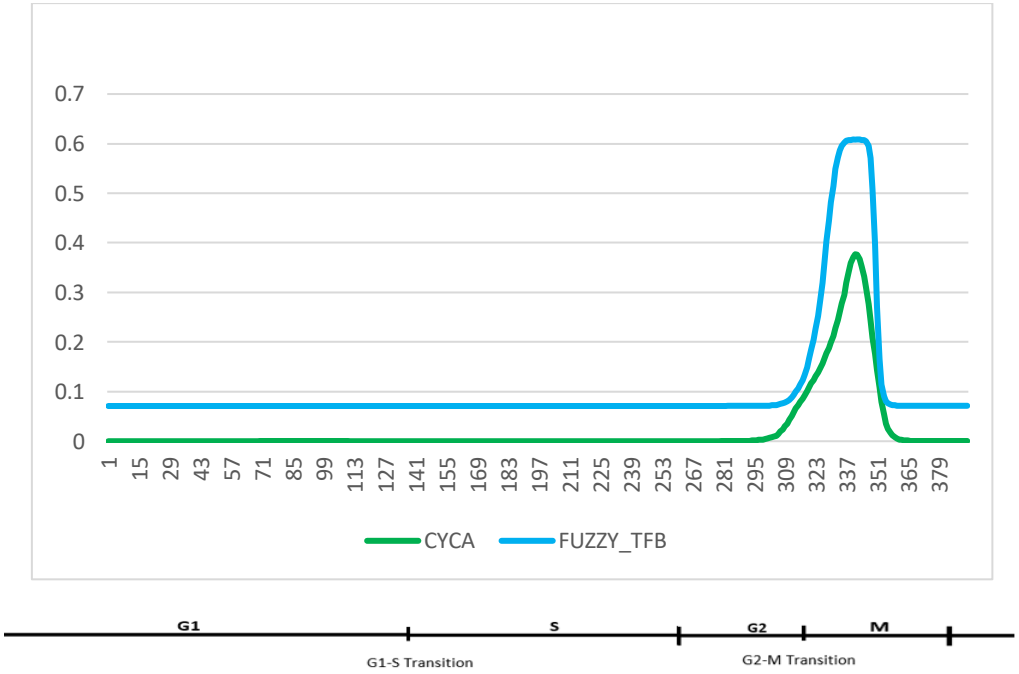


Figure 4.43: TFB fuzzy model results simulating TFB activity through the cell cycle. The green line represents CycA activity. The blue line represents TFB activity - this represents the computed crisp values from the fuzzy system.

Furthermore, the output of this TFB FIS model is considered satisfactory as it is very consistent with ODE model as shown in Figure 4.44.

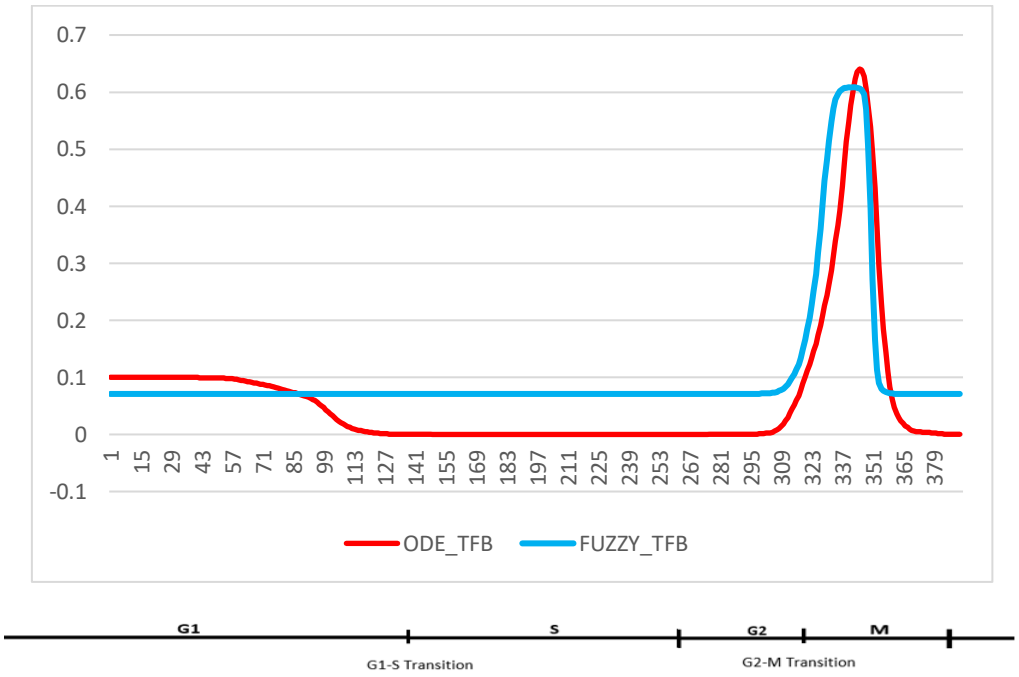


Figure 4.44: Comparison of TFB fuzzy model outcomes and TFB ODE model outcomes for TFB concentration and activity over the cell cycle. The blue line shows fuzzy system generated TFB values whereas the red line represents TFB from ODE model.

4.4.11 CycB Fuzzy Inference Model

- **CycB FIS Design**

The fuzzy inference system for CycB comprises three input variables and one output variable, as shown in Figure 4.45. The input variables represent the activator and degraders of CycB activity over the cell cycle. The TFB has a positive impact on CycB production while the rest of the elements, CDC20B and CDH1, negatively impact CycB abundance. CycB FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise CycB activity are shown in the BF in Eq. 3.11.

CYCB, TFB | (!CDH1 & !CDC20B) (3.11)

The main aim of this fuzzy model is to provide a CycB profile comparable to that from the ODE model which is presented in Appendix 2, Equation 4.28.

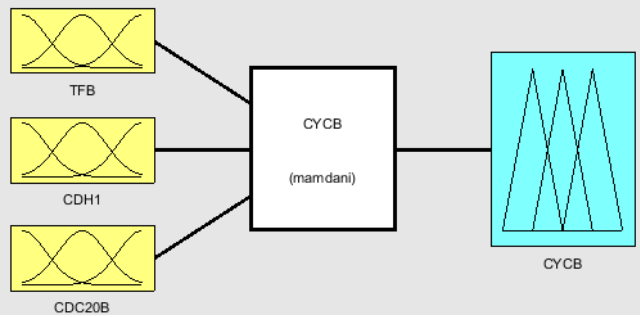


Figure 4.45: CycB fuzzy inference system structure with three input variables and one output variable.

Concerning the design of CycB membership functions, we created three-fuzzy sets (low, medium and high) to describe the activity (protein concentration) for this variable. Depending on prior knowledge, the fuzzy parameters for the fuzzy sets are adjusted manually; this also includes the domain of discourse of the variable, as shown in figure 4.46. The fuzzy membership functions for inputs are presented in the other subsections.

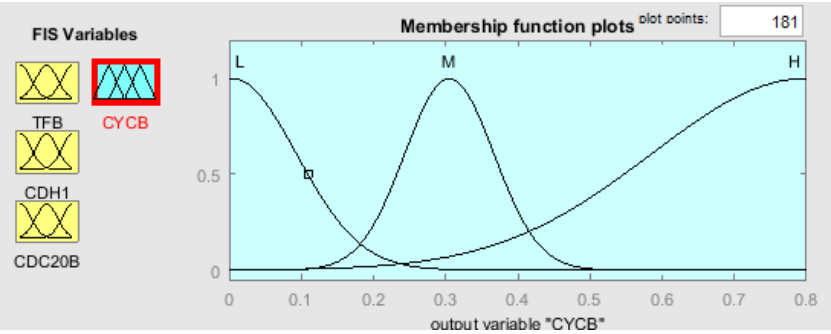


Figure 4.46: CycB (output) membership function design. This shows the fuzzy sets, Low, medium and high, and the universe of discourse [0, 0.8].

This fuzzy model also has a set of fuzzy rules. Our assumptions regarding the fuzzy rules are that they are built on the biological facts mentioned previously concerning protein concentration levels and interactions. Table 4.11 shows the CycB fuzzy inference system rules which describe the entire relationship between CycB and its activator and degraders.

Table 4.11: CYCB fuzzy rule set**.** The first column shows the number of rules in the system. The second column represents the input variables in the system while the last column represents the output variable CYCB. Each row represents a different fuzzy rule, which explains how the input variables can control CycB protein levels in cell cycle. These rules describe species activity in cell cycle.

Rule number	TFB	CDH1	CDC20B	CYCB
1	-	-	H	L
2	L	L	H	L
3	M	L	L	M
4	H	L	L	H

- **CycB FIS Results and Discussion**

Simulation results from the CycB FIS model are shown in Figure 4.47. Fuzzy rules we established showed that both CDH1 and CDC20B have prevented CycB protein production in the late phase of the cell cycle. We assumed that to start CycB production both CDH1 and CDC20B should be at their lowest concentrations. When TFB is active, it stimulates the production of CycB protein while CDC20 is available in the lowest amount and CycB levels rise quickly to its upper limit. Once CycB has reached the highest level, it activates CDC20B for its own degradation. This degrader would steadily decrease CycB abundance. Subsequently, CDH1 would be active again and work with CD20B to degrade CycB to the lowest level as shown in Figure 4.47.

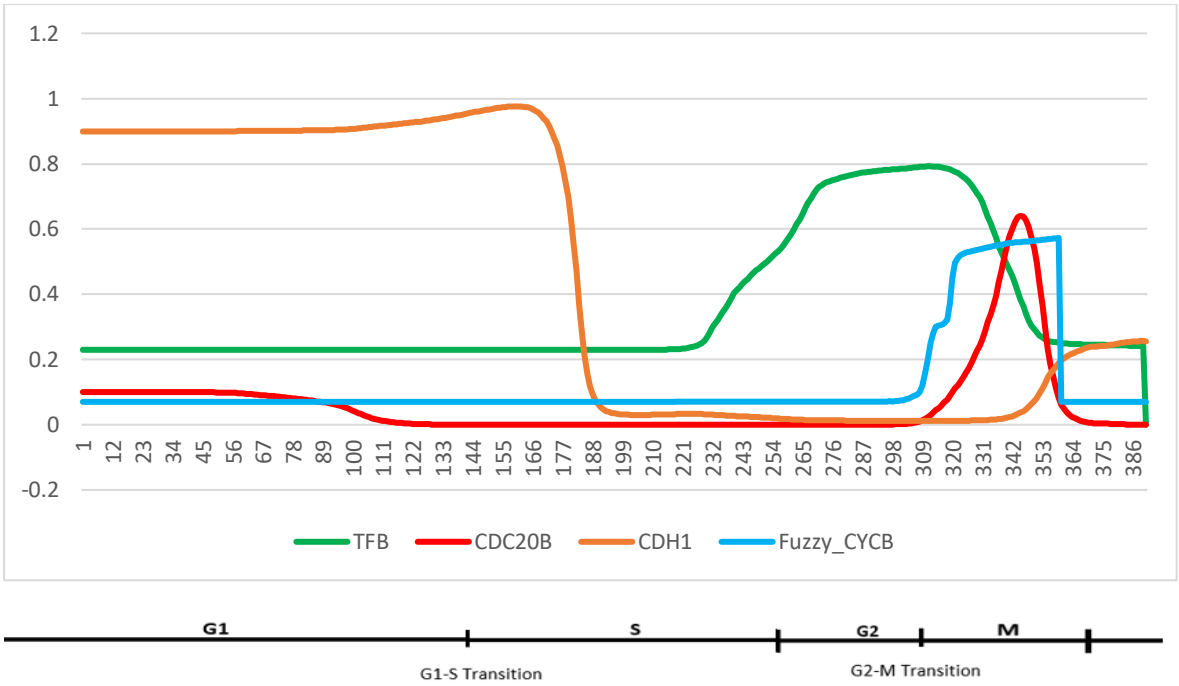


Figure 4.47: CycB fuzzy model results simulating CycB activity through the cell cycle. Green line represents TFB which activates CycB in M phase. The orange line represents CDH1 activity while the red line represents CDC20B activity to degrade CycB. These lines represent the system input variables. The blue line represents the outcome of the fuzzy CYCB model (CycB level) in crisp values.

Thus, CycB FIS produced realistic CycB activity over the cell cycle phases. For instance, there is a silent mode of CycB in the first and middle phases of the cell cycle, and the increase in CycB concentration becomes pronounced afterwards. Moreover, the results from this model and ODE model were compared and they are in good agreement overall as shown in Figure 4.48.

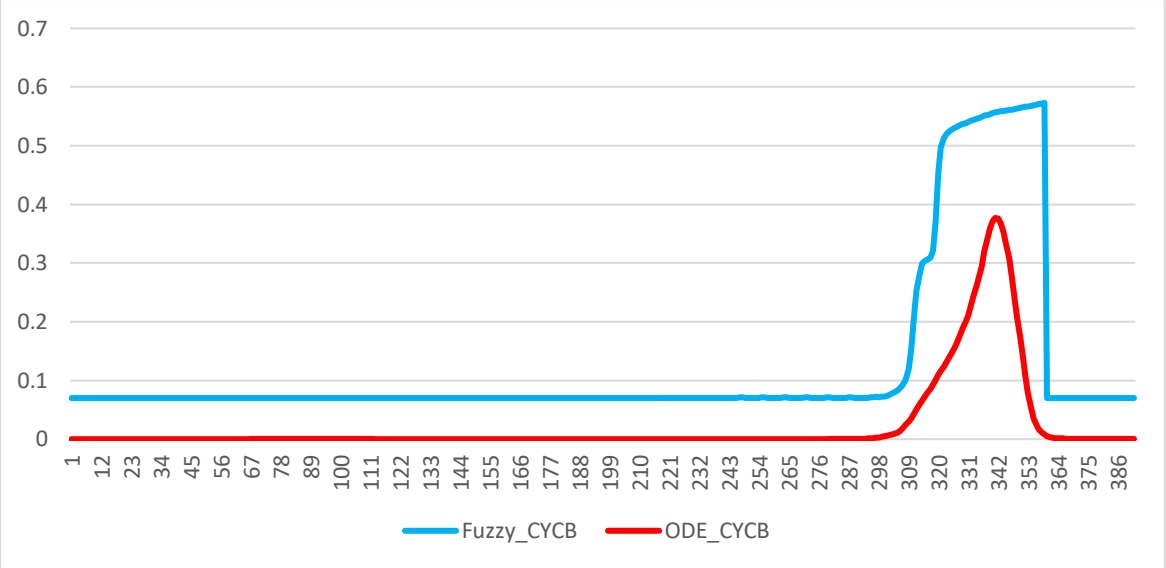


Figure 4.48: Comparison between CycB fuzzy model outcomes and CycB ODE model outcomes for CycB concentration and activity over the cell cycle. The blue line shows fuzzy system computed CycB values whereas the red line represents CycB from ODE model.

4.4.12 RB Fuzzy Inference Model:

- **RB FIS Design**

Here, the RB fuzzy inference system is considered as a multi-input system with a single output. It consists of three input variables (CycD, CycE and CycA) and one output as shown in Figure 4.49. The literature has reported that these three inputs influence RB activity in cell cycle. The output variable represents the values of RB over the simulation. RB FIS was derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise RB activity are shown in the BF in Eq. 3.20

RB, !CYCD & (!CYCE & !CYCA) (3.20)

RB FIS is expected to mimic the behaviour of RB in cell cycle similar to the RB ODE model which is presented in Appendix 2 (Equation 4.29).

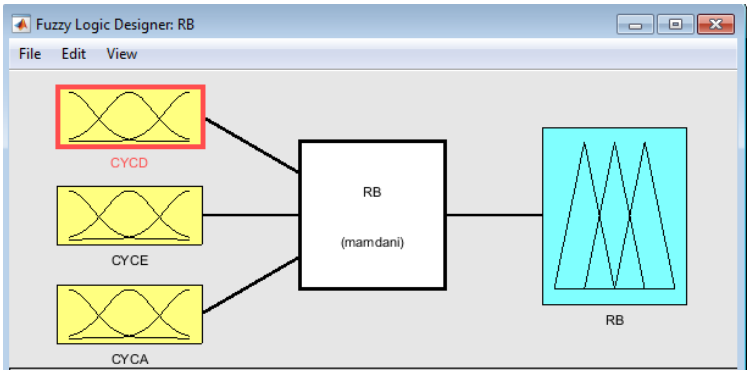


Figure 4.49: RB fuzzy inference system structure - three-input variables and one output variable.

Depending on the previous knowledge and using heuristic design, we aim to create the RB FIS. For instance, the membership function for RB has three-fuzzy sets (low (L), medium (M) and high (H)); and the universe of discourse for this variable is [0, 1.2]. These values signify the possible RB concentrations values; they are mined from realistic biological observations. The borders for fuzzy sets are also tuned judiciously to provide results with greater accuracy. This way, the tuned membership functions in FIS (Figure 4.50) can produce results for Rb similar to ODE. The fuzzy membership functions for inputs are presented in the other subsections.

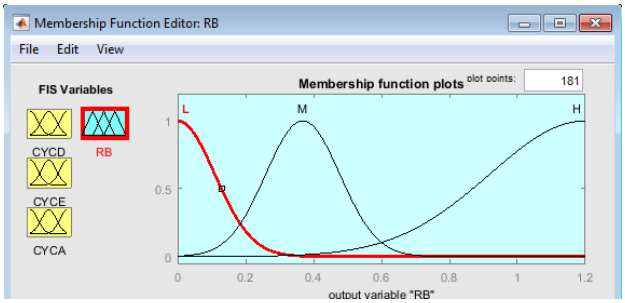


Figure 4.50: RB output variable membership function design. It shows the fuzzy sets; low, medium and high, and the universe of discourse [0, 1.2]

RB FIS also has a set of fuzzy rules to elucidate the dynamic behaviour for RB as shown in Table 4.12. This set of fuzzy rules capture the relationship between the input variables and the output variable.

Table 4.12: RB fuzzy rules set. The first column shows the number of rules in the system while the rest of the columns represent the input variables of the system, except the last column, which represents the output variable, RB. Each row represents a different fuzzy rule, which explains how the input variables control RB protein level in cell cycle. These rules describe species activity in cell cycle. For example, Rule 1 reflects the fact when cyclins are in low-level concentration, RB is at high level; this is the situation in G0 phase. The rest of the rules explain the relations between RB and the rest of the species.

Rule Number	Input 1 CYCD	Input 2 CYCE	Input 3 CYCA	Output RB
1	L	L	L	H
2	M	L	L	L
3	H	L	L	L
4	H	H	L	L
5	-	-	M	L

For instance, the fuzzy rules presented in the above table reveal the inverse relationship between the inputs CycD, CycE, CycA and RB over time in cell cycle system. Furthermore, fuzzy rules have demonstrated how cyclins protein levels negatively impacts RB activity. The increments in cyclins protein production increase cyclins levels that then reduce RB abundance. Cyclins affect both P27 and RB in a similar manner over cell cycle phases.

- **RB FIS Results and Discussion**

RB fuzzy inference works similarly to ODE model to infer the relationship between RB and activators and degraders. It can be seen that the RB (FIS) functionality produces the same functionality as the ODE model since both depend on protein concentration levels to activate or degrade RB protein abundance. For instance, as verified before, at the beginning cell cycle, RB protein level is at the highest level as shown in Figure 4.51.

The main task of RB is to prevent cell cycle initiation. This can be done by inhibiting TFE activity which produces CycE that initiates cell cycle. However, the abundance of cyclins over cell cycle phases can affect RB abundance and its activity. For instance, in G1 CycD starts synthesis and accumulates to a very high level, which affects RB activity and reduces the available RB protein level. CycE also negatively impacts RB activity. In late G1 phase, RB protein level decreases gradually to the lowest level and remains so over the rest of the cell cycle as shown in Figure 4.51. This is because another cyclin (CycA) is accumulated in the middle of the cell cycle that further suppresses Rb. On the other hand, when the cyclins species concentrations that control RB activity become low due to their degradation, the RB protein starts to accumulate again and increase gradually, especially in the transition from M phase to G0 phase, in the new born cell as shown in Figure 4.51.

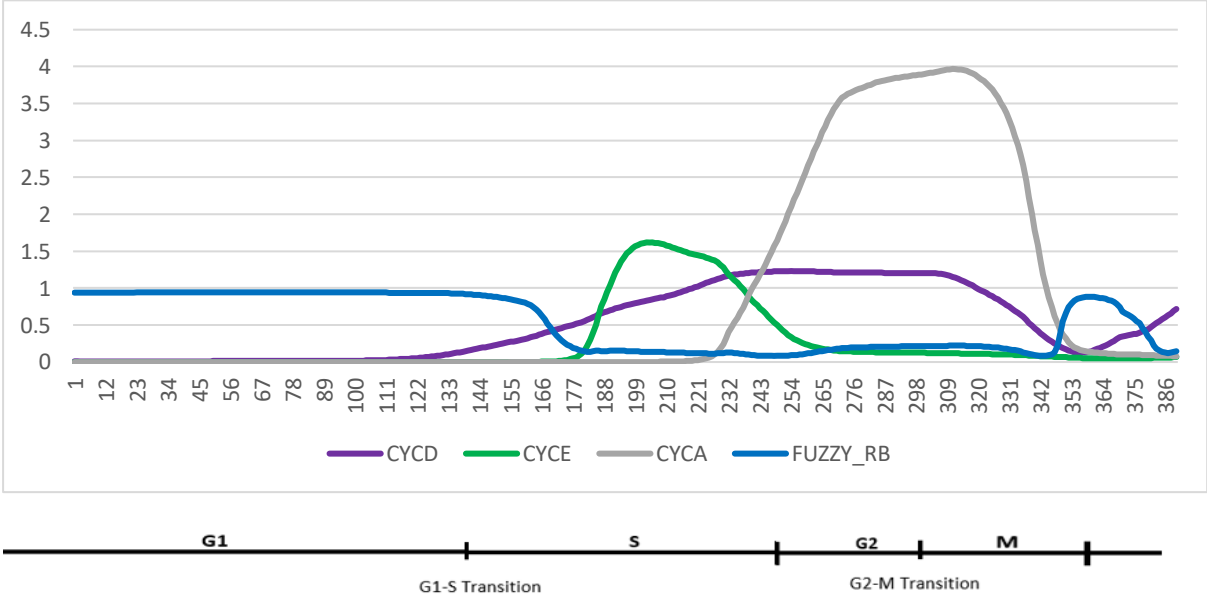


Figure 4.51: RB fuzzy inference system activity over cell cycle simulation. The blue line represents RB activity (protein concentration) in crisp values from the RB fuzzy system. The other lines represent input variables values and activity over cell cycle phases.

Finally, a comparative study has been done between the RB FIS model and RB ODE model. The aim of this comparison is to study the efficiency of the fuzzy model. The model is expected to mimic the RB behaviour in the same way as the popular ODE model. Indeed, the simulation

showed that our RB FIS model has the ability to mimic the behaviour of RB over the cell cycle phases as shown in Figure 4.52. In addition, our fuzzy model better represents the cyclic behaviour of RB from first cell cycle to the next (beginning and end values), whereas RB ODE model fails to produce this cyclic behaviour, as shown in Figure 4.52. For instance, RB should be highly active at the beginning and end of the simulation. Our fuzzy model shows that RB is at high level at the beginning and at the end of the cell cycle. The fuzzy model starts the second cycle with a high level of RB (cyclic behaviour). The RB ODE model shows that RB is inactive at the beginning of the cell cycle and is at high level at the end of cell cycle as shown in Figure 4.52. Thus the ODE has difficulty representing RB correctly at the beginning.

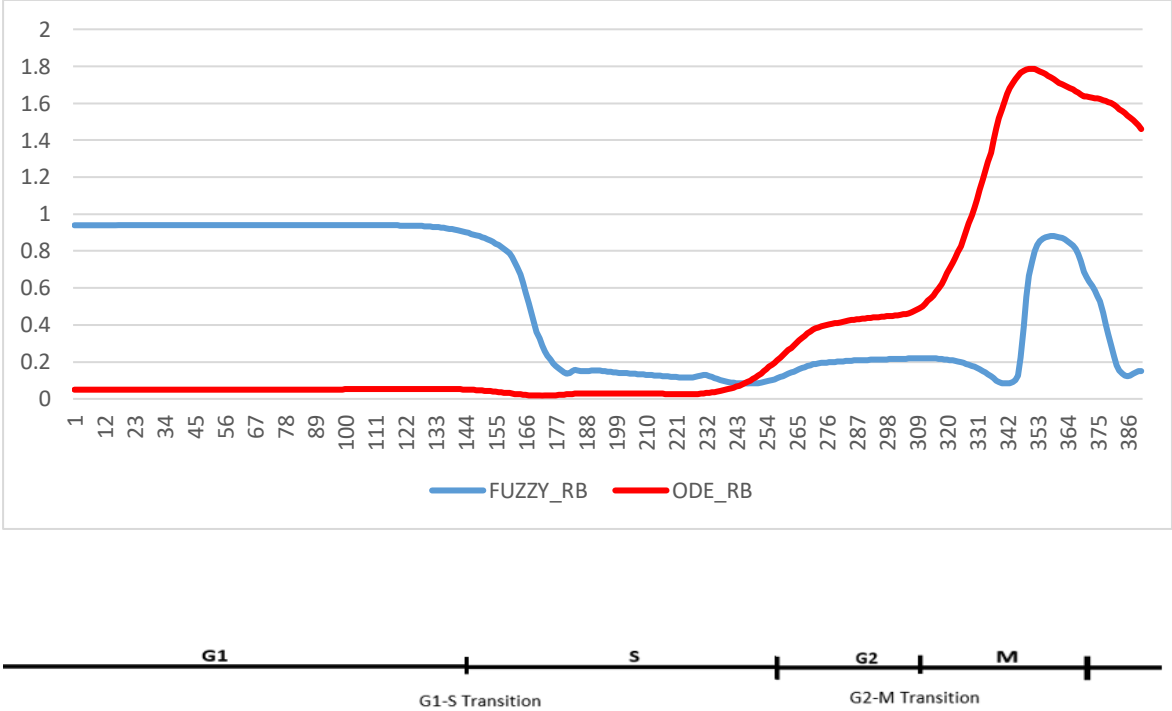


Figure 4.52: RB protein activity over cell cycle phases. The blue line shows fuzzy system generated RB whereas the red line represents Rb from ODE model.

4.4.13 CDC20A and CDC20B Fuzzy Inference Model:

- **CDC20 A/B FIS Design**

As demonstrated before, CDC20 species have a strong negative impact on Cyclin A and B, Concerning the CDC20 in our design, we rely on the biological facts from popular models, such as Singhania (2011). For CDC20, the assumption is that it should separate into two species:

CDC20A which deactivates CycA, and CDC20B, which deactivates CycB, but both CDC20A and CDC20B are dependent on CycB for their activation. Thus same principles were used in the design of for both proteins.

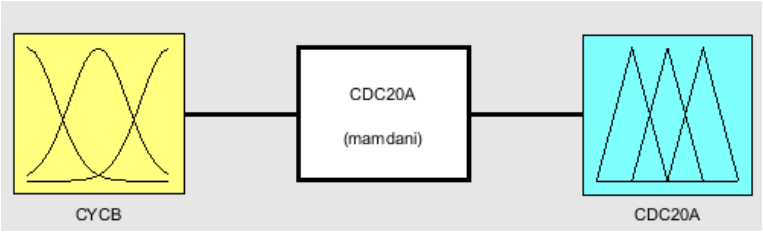
Consequently, our assumptions regarding the fuzzy inference systems for CDC20A/B are the same. So, to reduce the redundancy in analysis and discussion, we investigate both of them in the same manner in this section.

The CDC20A/B fuzzy inference system has a single input and a single output; the input variable is CycB, while the output variable is CDC20A/B as shown in Figure 4.53. CDC20A/B FIS's were derived from the Boolean model we developed (Chapter 3). The Boolean functions which organise CDC20A/B activities are shown in the BFs in Eqs 3.17 and 3.18, respectively.

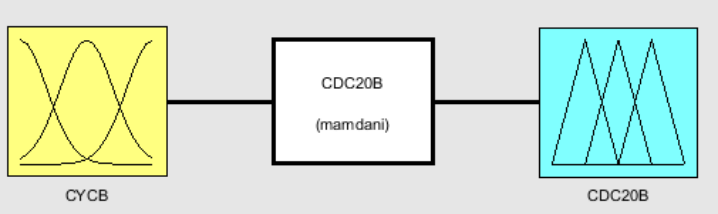
CDC20A = CYCB (3.17)

CDC20B = CYCB (3.18)

CDC20 behaviour has been mimicked by ODE model which is presented in Appendix 2, Equation 4.30. The intuitive and simple fuzzy inference system is expected to mimic CDC20A/B behaviour.



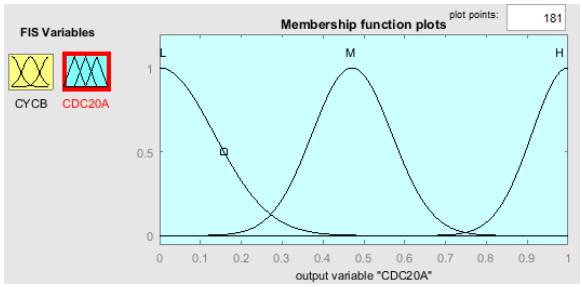
a. CDC20A fuzzy inference system structure - A single input variable and single output variable



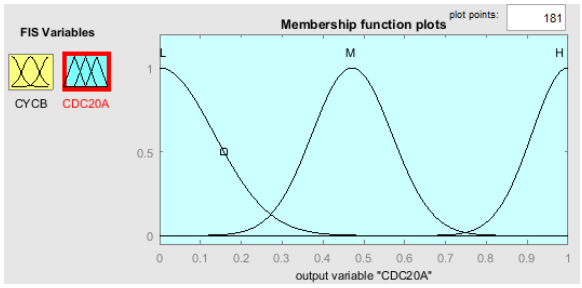
b. CDC20B fuzzy inference system structure -A single input variable and single output variable

Figure 4.53: CDC20A/B fuzzy inference system structure

The design of the membership function for CDC20A/B include three fuzzy subsets (low, medium and high) as shown in Figure 4.54. The fuzzy membership functions for inputs are presented in the other subsections. The fuzzy rules organise the CDC20A/B activity through these membership functions. As discussed, CDC20A/B activity is controlled by CycB concentration level. For this reason, the abundance of CycB affects CDC20A/B level.



a. CDC20A variable membership function design



b. CDC20B variable membership function design

Figure 4.54: The design of CDC20A/B output variables membership function. It shows the fuzzy sets low, medium and high and the universe of discourse [0, 1].

In addition, these fuzzy systems have a set of fuzzy rules as presented in Table 4.13.

Table 4.13: CDC20A/B Fuzzy rules sets

Rule number	CYCB	CDC20A	**Rule number**	CYCB	CDC20B

1	L	L	**1**	L	L
2	M	M	**2**	M	M
3	H	H	**3**	H	H

- **CDC20A/B FIS Results and Discussion**

This simulation offered appropriate results as shown in Figure 4.55. As can be seen, as long as CycB is not active CDC20A/B would stay at a low level. This is the situation in the cell cycle phases until the middle of the G2 phase. However, in late G2 phase, CycB concentration gradually increases, which affects CDC20A/B, since it is considered to be their activator. In the middle of the M phase, when CycB concentration decreases, then CDC20A/B would also decrease. These biological events are successfully interpreted by the intuitive CDC20A/B FIS models as shown in Figure 4.55.

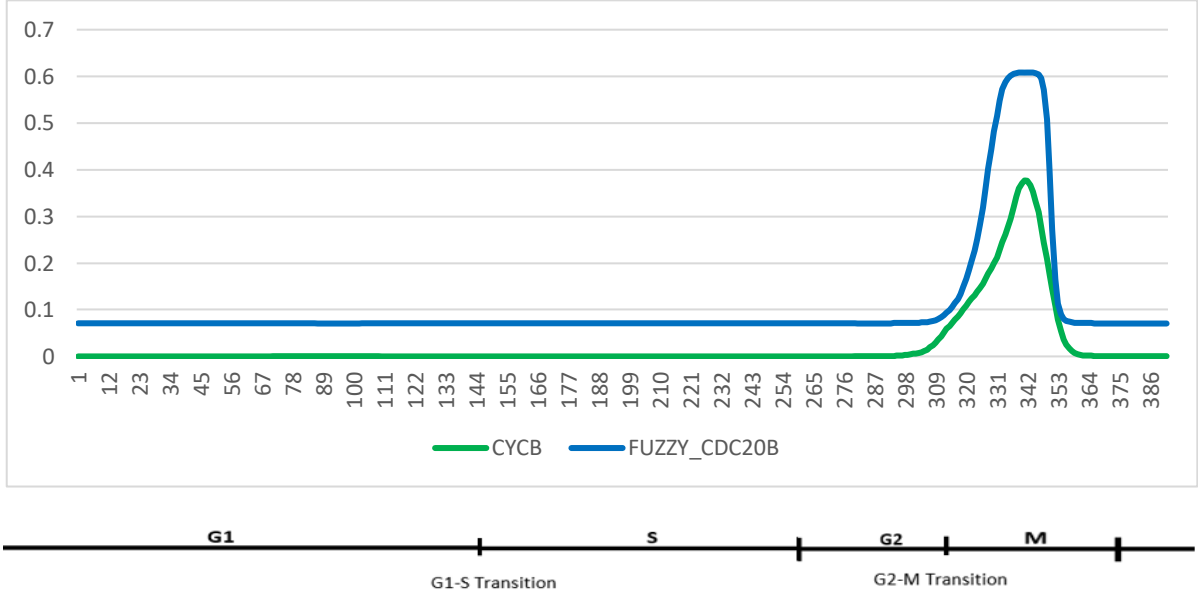


Figure 4.55: CDC20A/B fuzzy inference system activity over cell cycle simulation. The blue line represents fuzzy CDC20A/B activity protein concentration) in crisp values from the fuzzy system. The other line represents the input variable CycB.

Furthermore, a comparison was undertaken between CDC20A/B FIS models and CDC20A/B ODE model. A high level of agreement was found as shown in Figure 4.56.

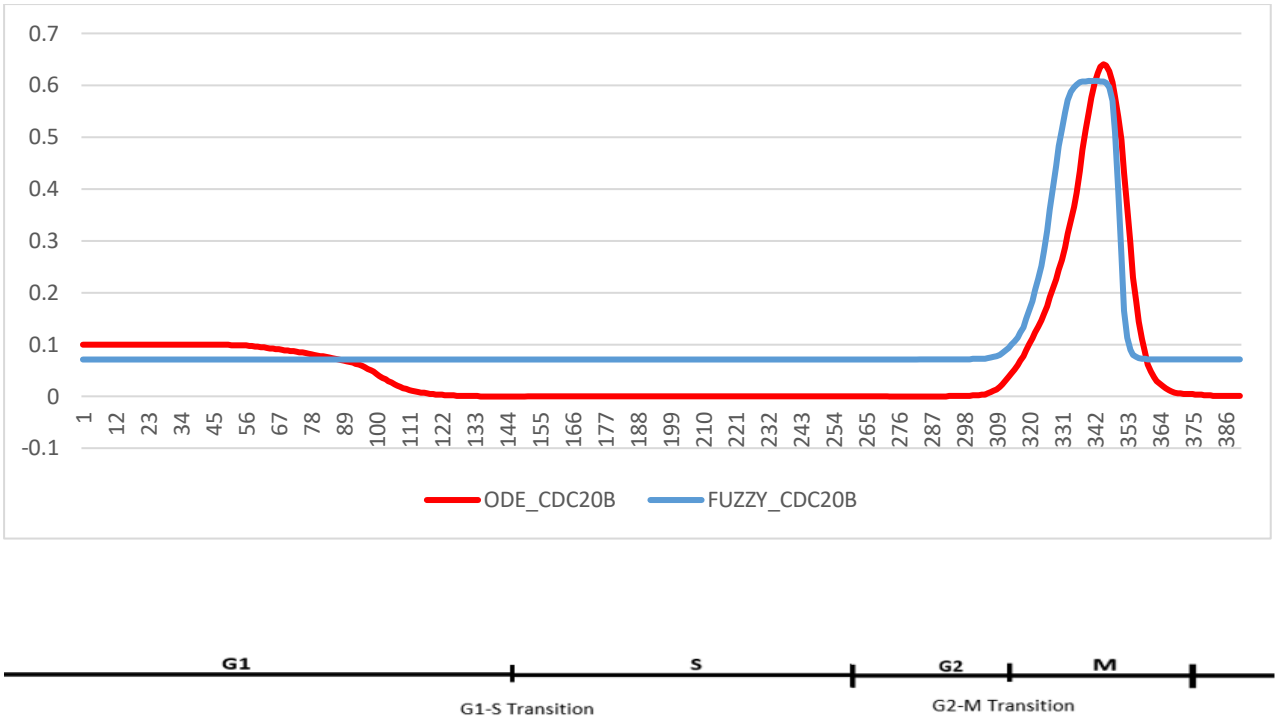


Figure 4.56: Comparison between CDC20A/B fuzzy model outcomes and CDC20 ODE model outcomes for CDC20A/B concentration and activity all over the cell cycle. The blue line shows fuzzy system generated CDC20A/B values over cell cycle simulation. The red line represents CDC20A/B from the ODE model.

4.5 Chapter Summary

In this chapter, we presented and investigated a highly promising method to mimic the mammalian cell cycle control system. We introduced and investigated fuzzy logic features; we also explained how it can become a significantly effective method to bridge the limitations in current discrete (Boolean etc.) and continuous (ODE etc.) models. Indeed, fuzzy logic proves that it can be considered as a novel method to mimic cell cycle controller system activity and dynamic behaviour by computing protein concentrations and describing the nonlinear behaviour of species over the cell cycle, using a fuzzy inference system in the modelling process. Furthermore, we developed a set of simple and intuitive fuzzy inference systems to mimic each cell cycle species. These fuzzy models are shaped depending on prior knowledge mined from realistic biological observations to design them in the best possible way. Each fuzzy inference model mimics the behaviour of one element of the cell cycle controller system over cell cycle phases. Earlier in the literature, each element in the cell cycle controller has been presented by an ODE model, since the ODE model is the most accurate and popular method used to mimic these element activity.

Thus each species was modelled by the fuzzy inference system model and validated with the existing ODE model but, as demonstrated previously, ODE models require kinetic parameters that are scarce. Accordingly, fuzzy inference system has the advantage of mimicking the continuous behaviour of these elements without the need for kinetic parameter values, and even under the conditions of missing and vague knowledge. This is achieved by creating local functions (membership functions) to represent variable as fuzzy variables that are processed by If- THEN rules to provide continuous outputs. Furthermore, these intuitive fuzzy inference systems can provide similar insights to ODE with greater simplicity and less effort. For instance, fuzzy inference models, such as RB, P27, and CDH1, have better represented the cyclic behaviour of these proteins in cell cycle while corresponding ODE model showed some difficulty creating this behaviour.

Essentially, the developed fuzzy inference systems provided the expected results to describe the temporal behaviour of protein concentrations over the cell cycle phases. We revealed that the results from the semi-continuous FIS models are comparable to the results from the continuous ODE model. However, we discovered that few of the developed fuzzy models (SCF, CycD and CycA FIS) show small variance from the ODE model. However, our aim was to provide simpler and easier approximate models that closely resemble the behaviour of ODE models as an alternative to ODE models and that was largely achieved by the FIS models. However, the above mentioned remaining few variant models can be improved to obtain more accurate results in order to make the whole model system a close equivalent of the ODE system. For this reason, we attempted to optimise the stated models by using one of the most autonomous optimisation methods - particle swarm optimisation (PSO). These attempts are discussed in the following chapter.

Chapter 5: Tuning fuzzy model with Particle Swarm Optimisation (PSO)

In the previous chapter, we investigated the fuzzy inference models we developed. As a consequence of manual tuning the models, they provided accurate results, which were compatible with the available biological knowledge (C.-C. Chen & Zhong, 2008; Iwamoto et al., 2011; Massagué, 2004). Furthermore, we compared our results with the most accurate and realistic results from the corresponding ODE model. The comparison revealed good agreement; however, several models showed a small variation in their results compared to the ODE model results. Therefore, in this chapter, a computational optimisation technique is used to improve the results.

In essence, most computational techniques have been inspired by existing and well known complex systems (Omizegba & Adebayo, 2009). For instance, artificial neural networks were considered to be a simplified model of the human brain. Furthermore, genetic algorithms were inspired by human evolution. In this chapter we investigate an additional biological social system inspired method called the Swarm Intelligence System, which can be used in computational methods. The Swarm Intelligence System is a collection of behaviours for a set of individuals who are interacting with their environment and each other. In addition, Swarm Intelligence has intelligent behaviour that is created by means of some agents, such as birds, ants and fish (Kenndy & Eberhart, 1995). The most popular examples of Swarm Intelligence in the computational intelligence area are the Ant Colony Optimisation (ACO) and the Particle Swarm Optimisation (PSO). ACO was developed on the basis of ants' behaviour and was used to optimise many discrete problems. The concept of Particle Swarm Optimisation (PSO) was developed to simulate a simplified social system and as an optimizer. Further details, including PSO concepts and the method used, are explained in the next sections.

5.1 PSO with Evolutionary Computation

Evolutionary computation is inspired by biological evolution; it is a family of algorithms for global optimisation, and most of the sub-fields of soft computing and artificial intelligence use these algorithms. In addition, evolutionary computation is considered as part of a family of errors and problem solvers and is based on the stochastic optimisation character.

Typically, evolutionary techniques go through a similar general procedure; the steps of this procedure are described below:

1. The initial population should be generated randomly.
2. Each subject should have a fitness value. This fitness value depends on the distance to the optimum.
3. Reproduction of the population depended on fitness values.
4. The stop condition - if requirements are met, then stop, otherwise go back to Step 2.

Generally, Particle Swarm Optimisation is comparable to other common evolutionary computational techniques, such as the Genetic Algorithm (GA) (Akkar et al., 2015). The system in PSO commences with a population. This population consists of random solutions, and a search process is employed to find the optimal one by updating the generations. Furthermore, PSO, like GA, needs fitness values to evaluate the population; but neither system guarantees success.

However, the main difference between PSO and GA is that PSO does not use evolutionary operators. For example, PSO has no crossover or mutation operators, such as the ones used in GA. Furthermore, the potential solutions in PSO are called "particles"; further details of the particles in PSO are clarified in the next section, Section 5.2. These particles update themselves with an internal velocity. These particles also have a memory, which is important for the algorithm. Moreover, information sharing among population members is completely different. For example, GA provides the chromosomes with the ability to share information between each other; this makes the entire population work as one group, which moves to the optimal area. However, PSO uses either g-Best or l-Best (as explained in the next section) to provide information to others. This mechanism is called "a one-way information sharing

mechanism." In most cases, compared to GA, all particles in the PSO tend to find the best solution rapidly (Akkar et al., 2015).

In addition, PSO is easier to implement than GA and needs fewer parameters to adjust (Akkar et al., 2015). These advantages make PSO applicable in many areas, such as function optimisation, fuzzy system optimisation, and any areas where GA can be applied. This inspired the researcher to involve PSO in the model refinement process as it could provide solutions relatively easily. The next section highlights the most important features of PSO.

5.2 The Basics of PSO

This technique was developed by Dr Eberhart and Dr Kennedy; it is a population-based stochastic optimisation technique (Kenndy & Eberhart, 1995). The authors were inspired by the social behaviour shown in existing biological systems, such as birds flocking and fish schooling. Therefore, PSO is an intelligent swarm method that uses some intelligent agents called "particles" to reach a higher level of intelligence. The main functionality of PSO is based on mimicking the behaviour of birds flocking (Clerc & Kennedy, 2002). For example, if we have a group of birds that are searching for food in a specific area in a random way, and the search area has only one piece of food. The birds do not know exactly where the food is, but they do know how far away the food in each iteration is. Therefore, what is the premium strategy to discover where the food is? The most efficient way is to follow the bird that is considered to be the nearest to the food.

The example above explains how the learning method in PSO for solving the optimisation problem works. In the PSO algorithm (Clerc & Kennedy, 2002; Kenndy & Eberhart, 1995), each single solution is called either an "agent," "bird," or "particle" in the search space. The search space represents all possible solutions for the optimisation problem. In addition, all the particles should have fitness values; these values are evaluated by the fitness function to be optimised. Particles should also have velocities, which direct the particles' acts of flying. Also, the flying process inside the problem space –the search process – for these particles would be accomplished by following the current optimum particles.

PSO is initialised by a group of random particles. These particles represent solutions in the search space; the algorithm then starts a search for the optimal by updating generations. In each iteration, each particle should update itself based on the two best values. The first value, which controls the updating scheme, is the best solution (fitness), achieved so far. This value is called “Personal Particle Best” (p-Best). Each particle remembers the position they were in when it had its best result so far; and, for this reason, this value should be stored. The second best value, which is tracked by the PSO optimiser, is the “Global Best” value (g-Best). G-Best is the best value obtained so far by any particle in the population, as shown in Figure 5.1.

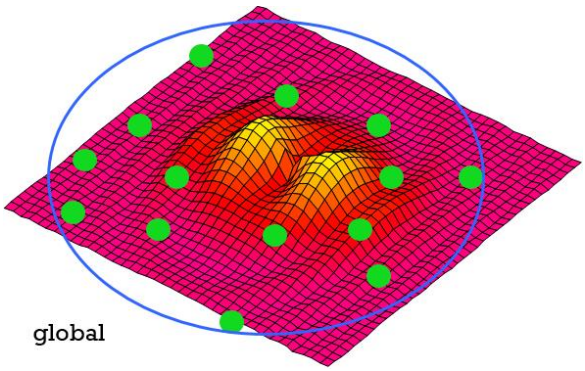


Figure 5.1: The global values within the population. The pink area represents the search space, while the orange peaks represent the solution area. The green nodes represent the particles in the swarm, while the blue circle indicates the global values of the swarm

In addition, particles in a swarm cooperate*;* they exchange information between each other about what they have discovered from the places that have been already visited. In the case when a particle takes part in a population with its topological neighbours, a particle knows the fitness values of those that are in its neighbourhood and uses the position of the one with the best fitness value. The particle can have a local best value called (l-Best), as shown in Figure 5.2. Accordingly, these personal (p-Best) and global (g-Best) positions are used to adjust the particle's velocity to achieve the new optimum solution (position), as shown in Equations 5.1 and 5.2.

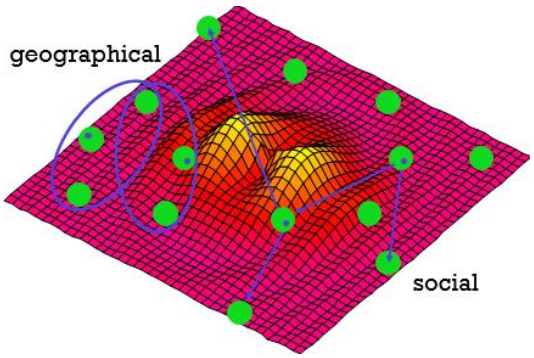


Figure 5.2: Particles and neighbours in the population**.** The pink area represents the search space, while the orange peaks represent the solution area. The blue circles represent the neighbouring particles in the swarm, while the blue lines show the social influence and information exchanges between the particles in the swarm (social experience); this makes the particle follow the best neighbour's direction.

The next section explains the dependency of each particle on the values of p-Best and g-Best for their movements. It also explains how each particle's movement attempts are processed in the search space to seek out the optimal solution in PSO.

5.3 The Mathematical Model for PSO

In PSO, particles adjust their positions and movements according to a "psychosocial compromise" between what an individual is satisfied with, and what society estimates. Mathematical representation of PSO can be clarified in this example: Suppose we have a particle called *i* that has a position denoted by a vector called *xi*, which is also a member in the search space, as shown in Figure 5.3. This vector has a discrete time step denoted by (*t*), and it shows the number of iterations of the algorithm. In addition, each particle has a velocity, which is denoted by vector *vi(t)*. The velocity vector describes the movement of the particle *i* in the sense of direction, distance, and size of the steps. Both positions of the vector *xi(t)* and the velocity vector *vi(t)* are in the same search space, which means they have the same dimension. Therefore, in Figure 5.3, we have a particle in time step *t*, which is located in position *xi(t)* and moves in the direction of vector *vi(t)*.

Furthermore, each particle has a memory to save the best-won position or the best-owned experience, which is called "Personal Best" (p-Best). This p-Best for particle *i* is denoted by the vector *Pi(t)*. In addition, in Figure 5.3, we have another experience with a common best

value among members of the swarm, which is denoted by *g(t)*. We need to be aware that this is not denoted with *i* because it belongs to the whole swarm called "Global Best" (g-Best).

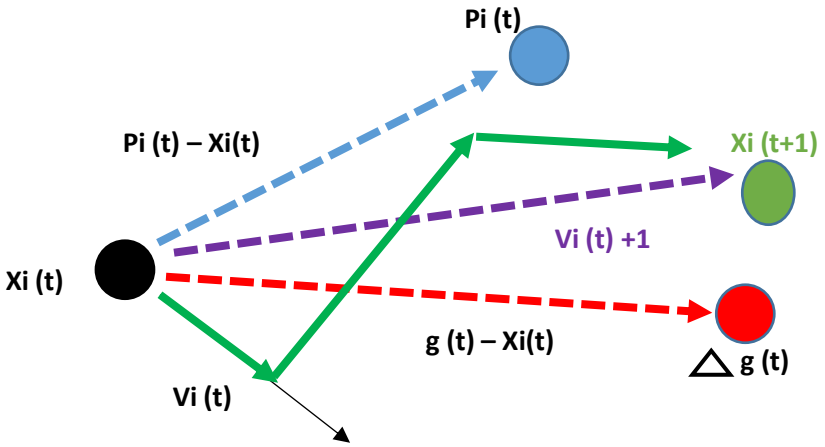


Figure 5.3: Simple model of the particle moving within the search space. The black node, *Xi(t)*, is the current position vector of particle *i*., while *Vi(t)* is the velocity vector, which is represented by the black arrow. The blue node represents *Pi(t)*, which is the personal best (p-Best vector for particle *i*) and represents the previous experience of the particle *i*, while the blue arrow shows the distance between the current position and the p-Best. The red node, *g(t)*, represents the global best (g-Best vector), and the red arrow represents the distance between the g-Best and the current position of the particle *i*. The green node, *xi(t+1)*, represents the new position of the current particle *i*, which is the new best solution, while the green arrows show the calculation for the three aforementioned vectors (black, blue, and red arrows) for the movement to the new position. The purple arrow, *Vi (t+1)*, represents the distance vector between the new and the previous positions for the particle *i* in the next time step.

According to Figure 5.3, mathematical model of PSO is very simple in defining these concepts in every iteration of PSO. Consequently, in each iteration the position and velocity of each particle is updated according to this simple mechanism. Each particle in the swarm must follow this mechanism.

Specifically, in updating with the mechanism of PSO, we first define a vector from the current particle position to be the p-Best and another vector that connects with the current position as the g-Best. So, if we want to define an equation for the vector between the current position and the p-Best, it would be the beginning point, *xi(t),* subtracted from the endpoint, *Pi(t)* (the blue arrow in Figure 5.3). Also, the vector between the current position and the g-Best would be the beginning point xi(t) subtracted from *g(t)* (the red arrow in Figure 5.3), while the velocity vector *vi(t)* is the black arrow in Figure 5.3.

Each particle should use these three vectors to move towards a new position. At the beginning, the particle moves in the direction of *vi(t)* velocity vector, then it moves towards the vector that connects *xi(t)* to *pi(t)*. After that, it moves towards the vector that connects *xi(t)* with the g-Best. Hence, the newly-updated position would be *xi(t+1),* the green node as shown in the previous figure, also the addition to the three vectors used from the beginning of the first vector to the end of the third vector defines the new velocity vector *vi(t+1)*. As we can see, the new position (the green node in Figure 5.3) is created according to the previous velocity as well as the p-Best and g-Best values. Consequently, this is a better location due to using previous decisions about the movement of this particle *Vi(t)*, the previous experience of the particle itself (p-Best), and the previous experience of the whole swarm (g-Best).

In each iteration, both the velocity and the position of each particle is changed by using the values and equations of the best positions. The particle can find this best position and so can other particles in the neighbourhood. Every particle regulates its travelling speed dynamically; this regulation would correspond to the flying experiences of the particle and its swarm. The particle update rule for the postioin is repesented in Eq. 5.1

$$p = p + v \tag{5.1}$$

and the violocity update equation is represented in Eq.5.2

$$v = v + c_1 * rand * (pBest - p) + c_2 * rand * (gBest - p) \tag{5.2}$$

where *p*: particle's position, *v*: path direction , c_1: weight of local information , c_2: weight of global information, *pBest*: best position of the particle, *gBest*: best position of the swarm, *rand*: random variable, stands for a random number between (0, 1). This random number changes with each iteration.

Meanwhile PSO is iterative technique, the equation for iterative update of the PSO particles in each time step can be written as follows (Akkar et al., 2015):

The position of a particle at time t+1 is given by:

$$xi(t+1) = xi(t) + vi(t+1) \tag{5.3}$$

However, the second part of the previous equation, *vi(t+1)* vector, consists of the sum of three vectors shown in Figure 5.3 - previous velocity vector, vector that connects *xi(t)* to *Pi(t)*, and a third vector which connects *xi(t)* to the g-Best as follows:

$$Vi(t+1) = w\ vi(t) + C1\ (pi(t) - xi(t)) + C2(g(t) - xi(t)) \qquad (5.4)$$

inertia + ***Personal influence*** + ***Social influence***

Where *w* is a real value coefficient (inertia weight), *p* is the personal best position of the particle, while *v* represents the particle velocity or path direction, and C_1 and C_2 represent the learning factors. C_1 can be considered as the personal best value, while C_2 can be considered as the neighbourhood's best value. These factors typically have values of $C_1=C_2=2$. C_1 is the weight of local information, while C_2 is the weight of global information. Furthermore, in Eq. 5.4, the part that includes C_1 and p-Best represents the personal influence; this part improves the individual and helps the particle return to the previous best position, which is better than the current one, while, the second part, with C_2 and g-Best, represents the social influence. It makes the particle track the best neighbour's direction. *p-Best* is the best position of the particle, while *g-Best* is the best position of the swarm, as demonstrated previously.

The previous equation, which explains PSO particles' velocity in each iteration, can be generalised as follows:

Vi(t+1) = diversification (inertia) + intensification *(5.5)*

$w\ vi(t)$ + $C1\ (pi(t) - xi(t)) + C2(g(t) - xi(t))$

Diversification searches for new solutions and explores the regions with potentially the best solutions, while the inertia forces the particle to transfer in the same direction and with the same velocity in the search space. As for the intensification part, it discovers the previous solutions and explores the best solution in a given region.

5.4 PSO Algorithm Pseudo Code

PSO functionality, i.e., equations of motion for particles, can be organised in a general pseudo code. The pseudo code describes the principles and functionality of the PSO algorithm in a high-level description. Therefore, the structural convention for PSO is as follows:

```
For each particle                                                          // line 1
   Initialise particle                                                     // line 2
END                                                                        // line 3

DO                                                                         // line4
   For each particle                                                       // line 5
      Calculate fitness value                                              // line 6
      If the fitness value is better than the best fitness value (p-best) in history   // line 7
         Set current value as the new p-best                               // line 8
   END                                                                     // line 9

   Choose the particle with the best fitness value of all the particles as the g-best   // line 10
   For each particle                                                       // line 11
      Calculate particle velocity according equation (5.4)                 // line 12
      Update particle position according equation (5.3)                    // line 13
   END                                                                     // line 14
While maximum iterations or minimum error criteria is not attained         // line 15
```

Figure 5.4: The pseudo code for the particle swarm optimisation

The first part – lines 1 to 3 – of the pseudo code describes the initialisation of the random population of PSO. The second part – lines 4 to 9 – defines the process of calculating the fitness function for each particle. Furthermore, it also explains how to select and update the p-Best of each particle. Line 10 shows how to select and update the g-Best value for the entire swarm. The next section – lines 11 to 14 – explains the updating process for velocity and position of each particle. The last part of the pseudo code, line 15, explains the stopping condition for the optimisation process. The stopping condition would be either the maximum number of iterations or the minimum value for the error.

The next section provides more details about PSO and how effective it would be in optimising the fuzzy model parameters.

5.5 Tuning Fuzzy Membership Function Parameters by PSO

We developed our Mamdani fuzzy inference systems based on the available human expert knowledge to control various activities of mammalian cell cycle. Throughout the development process, we discovered that the fuzzy rules need to be set up in a way that reflects the activities of the species. A good understanding of the activities of the species helps to derive accurate fuzzy rules. Furthermore, we adjusted the fuzzy membership functions (MFs) manually, based on our expertise and knowledge; this type of design of membership function is called "heuristic design." Even though the fuzzy inference systems developed provided acceptable results with few exceptions, we sought more precise results from our fuzzy inference systems. Further, these manual adjustments were time-consuming. We also sometimes faced the need to carry out several additional experiments for each membership function (MF) that were tiresome.

Several techniques have been reported in the literature to overcome the difficulties of manual adjustment for MFs. One such method is adaptive neural-network-based fuzzy inference system (ANFIS) (J.-S. Jang, 1993). This technique uses an adaptive neural network (NN) to learn the mapping between system inputs and outputs, and then create a Sugeno-type fuzzy system based on the neural network. Another example is adjusting the membership function parameters automatically using a method such as GA. GA has been used to tune MFs in electrical engineering fuzzy systems (Mucientes, Moreno, Bugarín, & Barro, 2007; Pratihar, Deb, & Ghosh, 1999).

As explained previously, PSO is more efficient and flexible than other techniques, such as GA; this has made PSO more popular and more applicable in engineering applications and for modelling systems (Kwok, Ha, Nguyen, Li, & Samali, 2006). Consequently, we use PSO to automatically tune the membership functions of our Mamdani fuzzy inference systems. Using PSO, the aim is to minimise the difference between all our fuzzy inference systems' results and ODE results. It is expected that PSO would enhance our results and make them more precise and more convincing.

The difference between actual and predicted values is the "error". The function that calculates the error is called the "fitness function". The most popular error function is the mean-squared error (MSE) as shown below in Equation 5.6:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(\hat{X}_i - X_i\right)^2 \qquad (5.6)$$

Accordingly, we employed the MSE function as a fitness function in the implementation of PSO, as PSO was similar to other evolutionary computational techniques such as GA, which needed a fitness function.

Consequently, the proper adjustment of MFs for each fuzzy inference system (FIS) should provide a new and better MFs design with more precise results. Furthermore, the results from each FIS will have minimised the differences from the target ODE results. The next sub-section explains the procedure for tuning the parameters of membership functions using PSO.

5.5.1 Procedure for Tuning MFs

The approach used to employ PSO for adjusting the membership function parameters in fuzzy inference systems in our study is shown in Figure 5.5. As shown, each particle should represent the parameters of the membership functions for the input and output variables of the fuzzy inference system. In the last chapter – Chapter 4 – we observed that some fuzzy systems have to be optimised due to their results being slightly different from the original ODE results. Even so, their biological activities were elucidated very well. In this chapter, we optimise these fuzzy inference systems, including SCF, CycD, and CycA. Each of these systems would follow the same optimisation procedure.

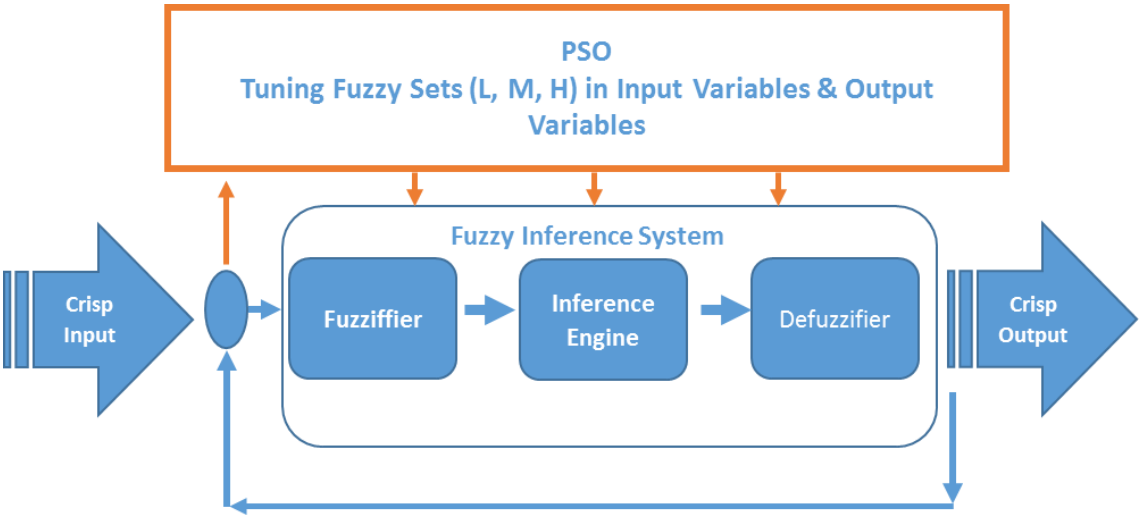


Figure 5.5: Tuning the fuzzy set parameters using PSO

Almost all the fuzzy inference systems developed consisted of multi-inputs and single-output; we call them MISO. To optimise these MISO fuzzy systems, we follow these assumptions in our implementation:

- We use the same fuzzy rules that already exist inside the fuzzy inference system that we intend to optimise. Therefore, the optimisation of each fuzzy inference system would be applied to fine tune the membership function parameters.
- If we have an inference system with *m* number of inputs, then, the number of fuzzy sets for these input variables would be *m1*, *m2*, *m3*, ... *mn*, as shown in Table 5.1.

Each input and output variable in these systems has Gaussian-shaped membership functions. For this kind of fuzzy set shape, the parameters are the *mean* and the *standard deviation* for each fuzzy set as shown in Figure 5.6. This means that each fuzzy set has two parameters as shown in Table 5.1. Fundamentally, almost all fuzzy inference systems that we established for each species have membership functions in the Gaussian form.

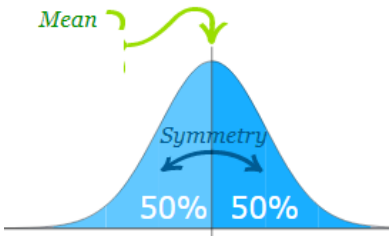


Figure 5.6: Representation of the general Gaussian shape with mean and standard deviation

- We can specify the size of the parameters of the Gaussian membership function of the input variables by Equation 5.7:

$$\sum_{i=1}^{n}(2mi) \qquad (5.7)$$

This represents the dimensions of the particles for the input variables where *n* indicates the number of input variables and *m* indicates the number of fuzzy sets. In addition, particle dimensions for the output variable can be represented by Equation (5.8):

$$\sum_{i=1}^{n}(2t) \qquad (5.8)$$

Where *n* indicates the number of output variables and *t* indicates the number of fuzzy sets.

Table 5.1 General representation of PSO particles for Gaussian membership function parameters in the MISO system. The first column illustrates the system variables, such as the input and output variables. In this column, the number signifies the input variable. The first row describes the mean and standard deviation for each membership function. These rows are for the n input variables and one output variable. The input variables have 1 to m membership functions and the output variable has 1 to t of membership functions. Finally, in the last column, 2m means that two positions have been used until the m variables**.**

	C	σ	C	σ	.....	C	σ	
Input Variable number 1	X11	X11	X12	X12	.....	X1m	X1m	2m1
Input Variable number 2	X21	X21	X22	X22	.....	X2m	X2m	2m2
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
Input Variable number n	Xn1	Xn1	Xn2	Xn2	.....	Xnm	Xnm	2mn
Output Variable	Y1	Y1	Y2	Y2	.....	Yt	Yt	2t

- In PSO, the aforementioned parameters are represented as a particle that flies in the search space searching for the global best fitness. Furthermore, each simulation starts with an initialised set of parameters.

- When the optimisation method adjusts the parameters, they should be used to examine the efficiency of the fuzzy logic system. When the membership functions undergo change, the inputs and outputs of the membership functions would become different.

This procedure is repeated until the goal is achieved, either by finding the minimum error or by reaching the end of iterations. In our optimisation, we specified the stopping condition using the minimum error.

Our main idea in exploiting PSO is to use particles to adjust the MF parameters. Therefore, in the PSO, each particle was implemented to signify the parameters in MFs. The optimisation is applied to MFs representing both the input and output variables in each fuzzy inference system.

PSO is implemented in MatLab to optimise the fuzzy sets parameters as shown in Figure 5.7. Each particle is expected to accurately represent membership function parameters of input and output variables (Section 5.5.1); therefore, in our implementation, we treated all fuzzy set parameters as variables in the PSO algorithm, where each particle contains *x* values indicative of the total number of fuzzy set parameters. For each set of variables, we assessed the performance by measuring the error using the MSE fitness function. The implemented program reads the input values from the source file and stores the values in the variables. It then uses the same variables to evaluate the fuzzy system and calculate the error using the MSE fitness function. As shown in Figure 5.7, the PSO implementation attempts to minimise the mean square error between fuzzy output results and targeted ODE output results.

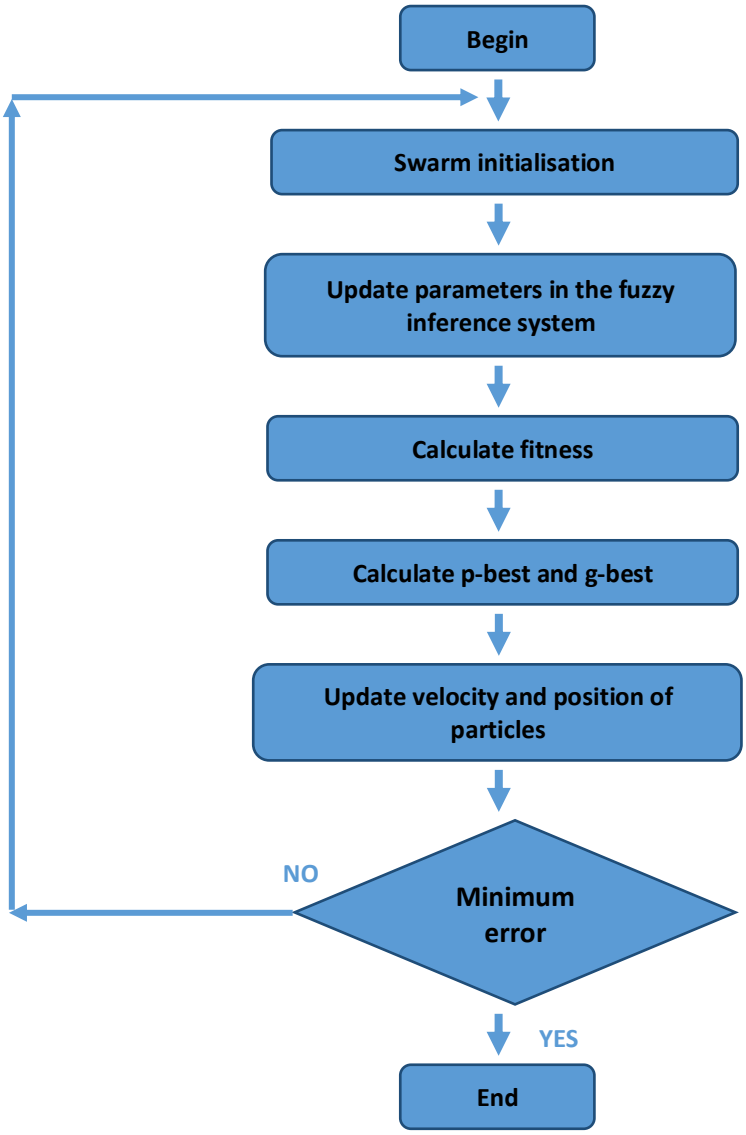


Figure 5.7: PSO Procedure to tune fuzzy membership function parameters in the fuzzy inference systems

Concerning the number of particles in the PSO population, most literature report that typically 20-40 particles are enough, but for complex optimisation, suggestion is to use a larger number, such as 100 particles, in order to get more accurate results. Therefore, we assumed that the size of the population was 100 particles. As mentioned before, the stopping condition is set to the minimum value for the error. For other PSO factors such as *C1*, *C2*, and the inertia factor, we used the optional default values for them, the most commonly used parameters of PSO algorithm. For instance, the aacceleration factors (c_1 and c_2): 2 to 2.05 (Alam, 2016).

The next section investigates the optimisation of the aforementioned fuzzy models.

5.6 Optimisation Results and Discussion

Here, we investigate the optimisation of the fuzzy inference systems for SCF, CycD, and CycA. The optimisation aims to get new and more precise results than those we achieved in Chapter 4. The results from the newly optimised models better reflect protein concentrations and species activity for each element in cell cycle phases. For all optimisation attempts, we used the assumptions and methods as explained in the following sections.

5.6.1 Optimising SCF Fuzzy Inference System

The SCF fuzzy inference system components have been explained in the last chapter, in Section 4.2. However, we applied the optimisation for this inference system to adjust the fuzzy set parameters. As demonstrated previously, this fuzzy inference system has two input variables and one output variable. Each variable has three fuzzy sets in the form of Gaussian function. According to Equations 5.6 and 5.7, the number of parameters for this system is 18 parameters. These parameters need to be tuned. Therefore, in PSO implementation, each particle would have 18 dimensions.

By completing the MFs tuning procedure using the optimisation procedure described in Section 5.5.1, we achieved optimum results for the 18 parameters that provided minimum value for the fitness function. As can be seen in Figure 5.8., the search was terminated after 80 iterations since the stopping condition was achieved. Furthermore, as we can see, no reduction in error was observed. From the figure we can see that the best MSE fitness function value is 0.000780685; this is the minimum value for the error between the optimised fuzzy inference system results and the target ODE system results.

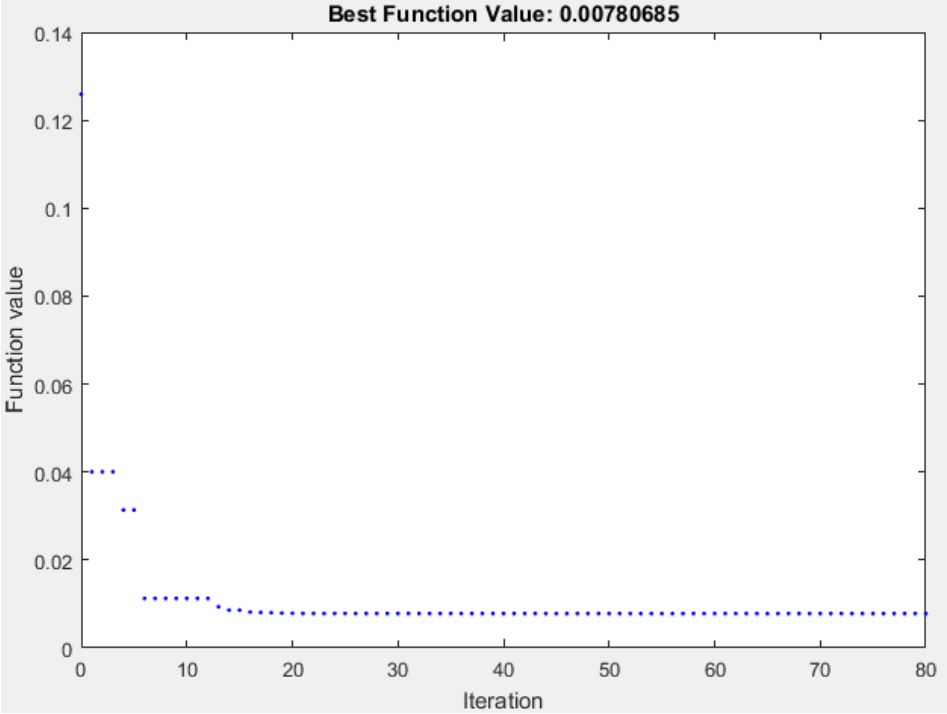
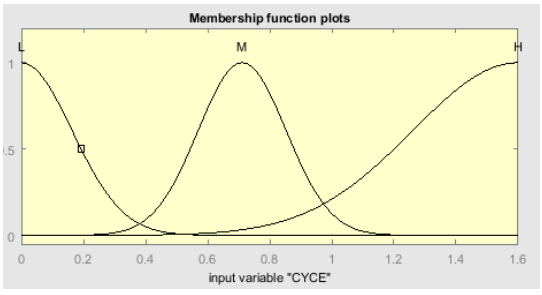
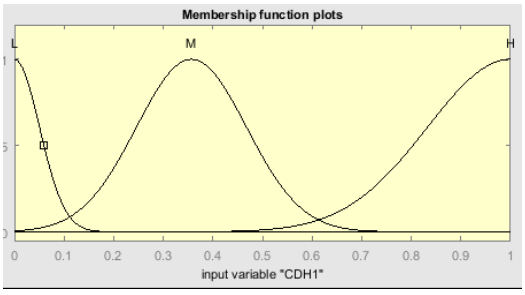
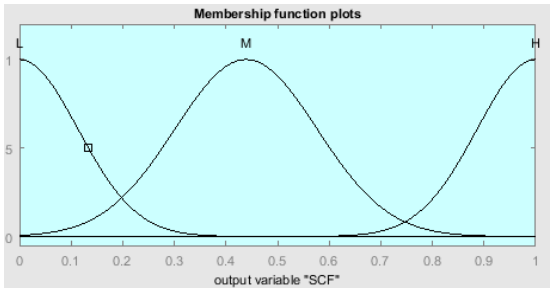


Figure 5.8: The reduction of error during PSO for the SCF fuzzy inference system

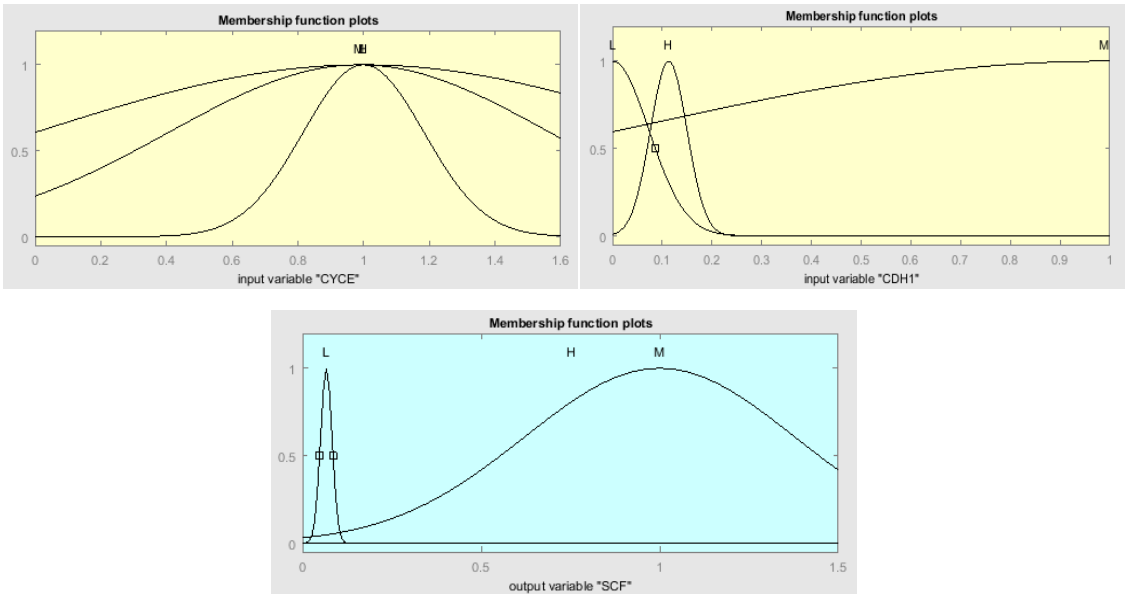
PSO optimisation has tuned the MFs for the input and output variables of the SCF model. As shown in Figure 5.9.b, the membership functions of the fuzzy sets have been modified significantly compared to the manually tuned configuration 5.9.a. Also, the values of the optimised fuzzy set parameters are quite different from manually tuned values. Table 5.2 represents the values of the parameters – pre- and post-optimisation - for the input variables and the output variable for the SCF model. This optimisation provides a new fuzzy inference system for SCF with the same fuzzy rules but with optimised fuzzy sets.







(a)



(b)

Figure 5.9: Pre- and post-optimisation of membership functions (shapes and parameters) of the 2 input variables and one output variable in the SCF fuzzy inference system. (a) Parameters, shapes, and values of the heuristically optimised membership functions; (b) Parameters, shapes, and values of the post-optimised membership functions

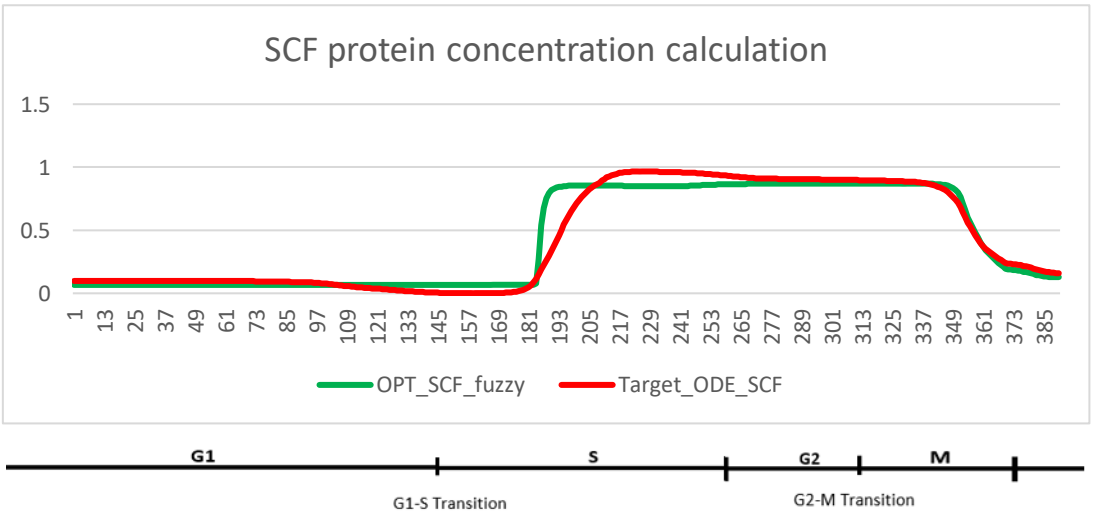
Table 5.2: Heuristically optimised and post-optimisation memberships function parameter values for input and output variables of SCF fuzzy inference system

Input Variable 1	**CYCE Fuzzy Set Parameters Pre-Optimisation**			**CYCE Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.1636 2.08e-17]	[0.142 0.7111]	[0.3397 1.6]	[1 1]	[0.5804 0.9866]	[0.184 0.9998]
Input Variable 2	**CDH1 Fuzzy Set Parameters Pre-Optimisation**			**CDH1 Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.05055 6.94e-18]	[0.111 0.3568]	[0.1674 1]	[0.07317 1e-05]	[0.9715 0.9897]	[0.03676 0.1131]
Output Variable 1	**SCF Fuzzy Set Parameters Pre-Optimisation**			**SCF Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H

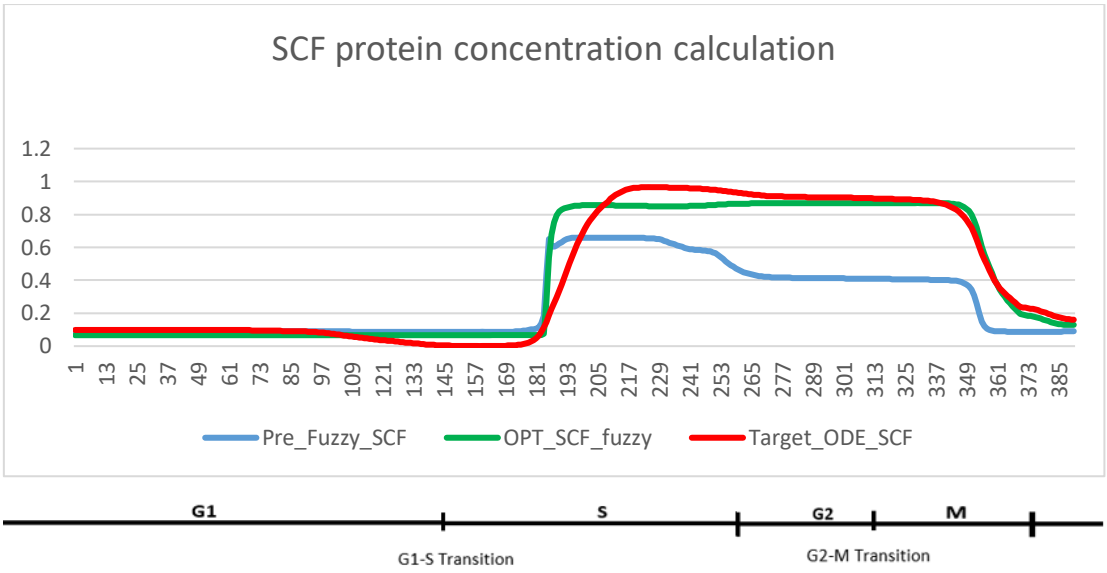
	[0.1668 1.041e-17]	[0.207 0.8181]	[0.1062 1.5]	[0.0161 0.06675]	[0.3788 1]	[6.54e-05 0.3301]

The new optimised fuzzy inference system was used to calculate the values of SCF over the cell cycle simulation and validate whether the new model can explain SCF activity in cell cycle. We employed the same data set as inputs as used in Section 4.4, and here we obtained new results that were more precise and closer to the target ODE results, as shown in Figure 5.10a. Furthermore, Figure 5.10b explores the variation between the results from the PSO optimised fuzzy model, the heuristically designed model (pre-optimised), and the ODE target model results.

Figure 5.10b elucidates the enhancements attained by PSO optimisation of the SCF fuzzy inference model. The results showed how the newly optimised model (green line) can provide closer results to the target values (red line), and also how the optimised model has outperformed the heuristically designed model (blue line).



(a)



(b)

Figure 5.10: SCF concentrations and activity results over cell cycle for the pre-optimised (heuristic) fuzzy model, PSO optimised fuzzy model and ODE model. (a) Optimised SCF fuzzy model results (green line) and the target ODE model results (red line) showing that the difference between the two has been remarkably reduced in PSO optimisation; and (b) Comparison between the heuristic (blue line) and PSO optimised (green line) SCF fuzzy model results and ODE results (red line). A remarkable improvement in SCF concentration can be seen after PSO optimisation.

5.6.2 Optimising CycD Fuzzy Inference System

We applied the PSO optimisation to CycD fuzzy inference system as well to tune the fuzzy set parameters. As explained earlier (Chapter 4), this fuzzy inference system has four input variables and one output variable. Each variable has three fuzzy sets of Gaussian shape. From Equations 5.6 and 5.7, the number of parameters for this system is 30. These parameters need to be adjusted. Hence, in PSO implementation, each particle has 30 dimensions.

Optimisation on the CycD model reached minimum error in 460 iterations, as shown in Figure 5.11. This figure shows the fitness function value for the given optimal solution (MSE= 0.018464). This is the minimum value for the error between the optimised fuzzy inference system's results and the target ODE system results.

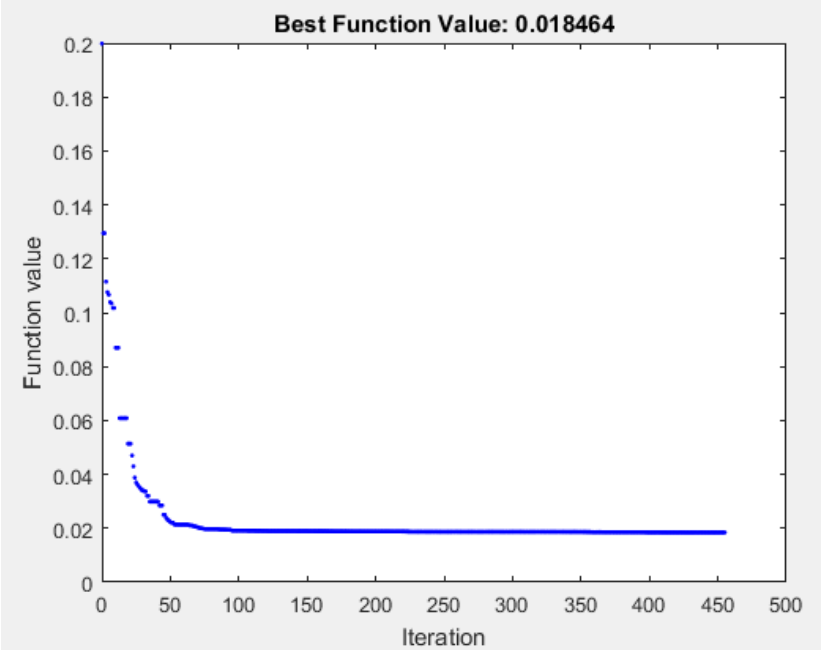
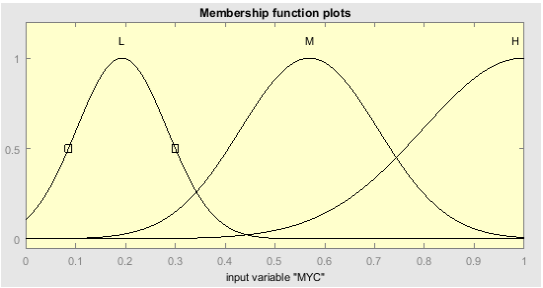
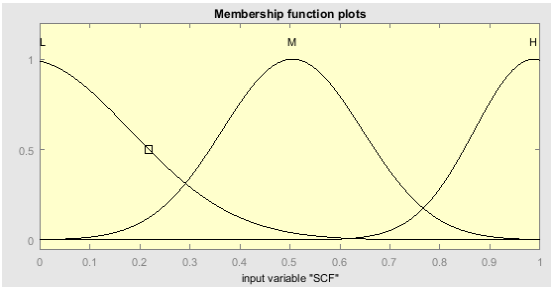
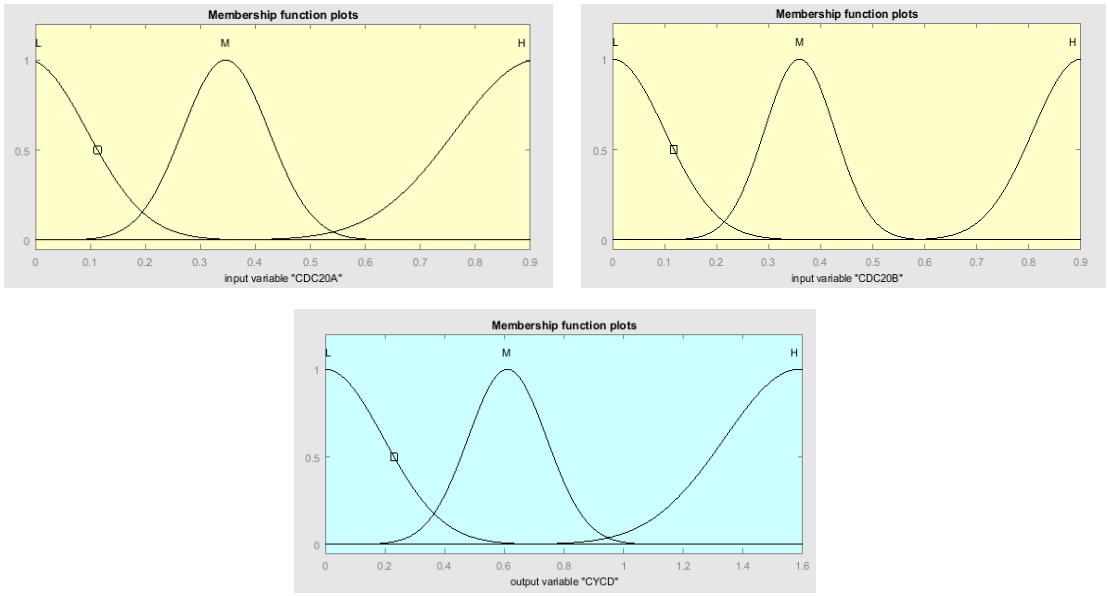


Figure 5.11: The reduction of error during PSO for the CycD fuzzy inference system

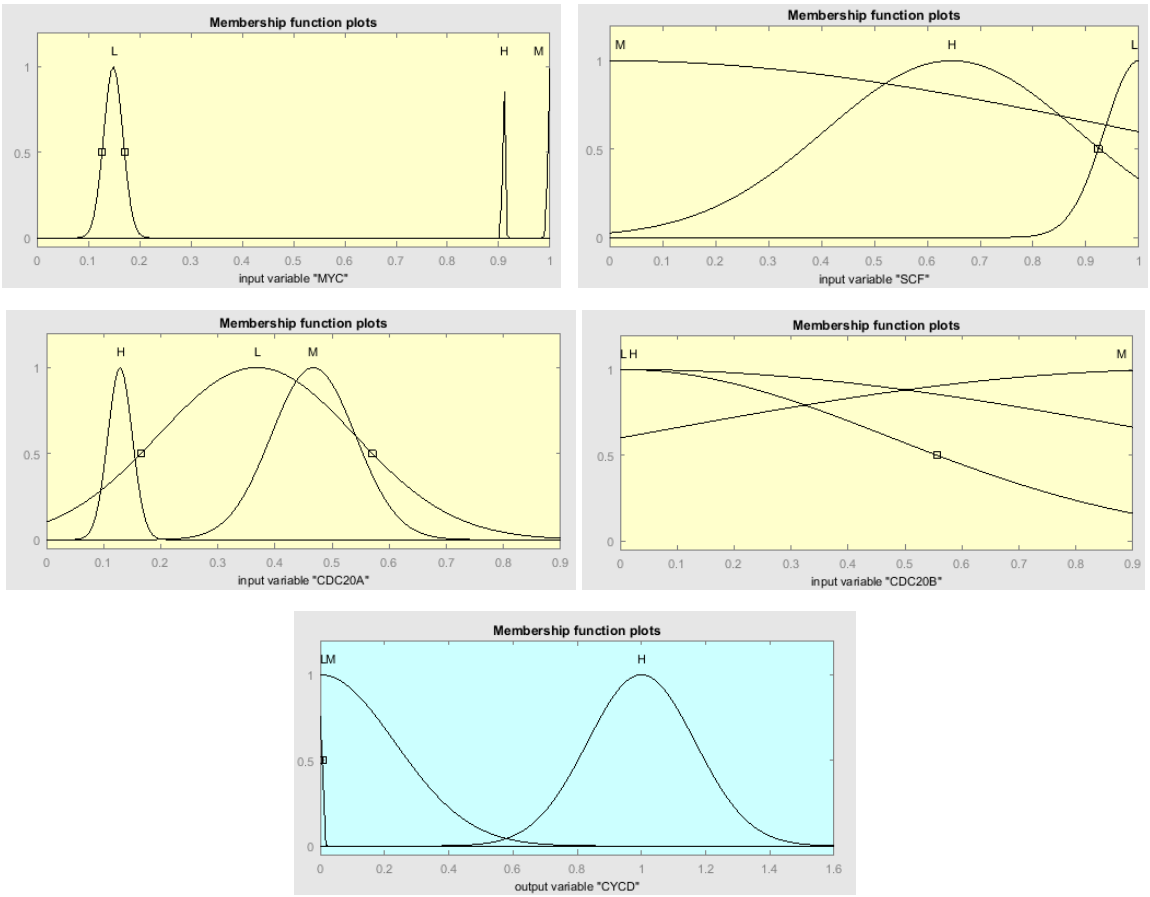
The optimisation progressively adjusted the MFs of the input and output variables of the CycD fuzzy inference system. The newly-adjusted fuzzy set parameters are shown in Figure 5.12. Furthermore, Table 5.3 presents the values for the parameters of the pre- and post-optimised functions for the input variables and the output variable. The values of the optimised fuzzy set parameters are all more diverse than those that were adjusted manually. This optimisation provides a new fuzzy inference system for CycD with identical rules but with new optimised fuzzy sets.







(a)



(b)

Figure 5.12: Heuristically optimised (Pre-) and post-optimised membership functions of the input and output variables in the CycD fuzzy inference system (a) Parameters, shapes and values of the heuristically optimised (pre-optimised) membership functions; (b) Parameters, shapes, and values of the post-optimised membership.

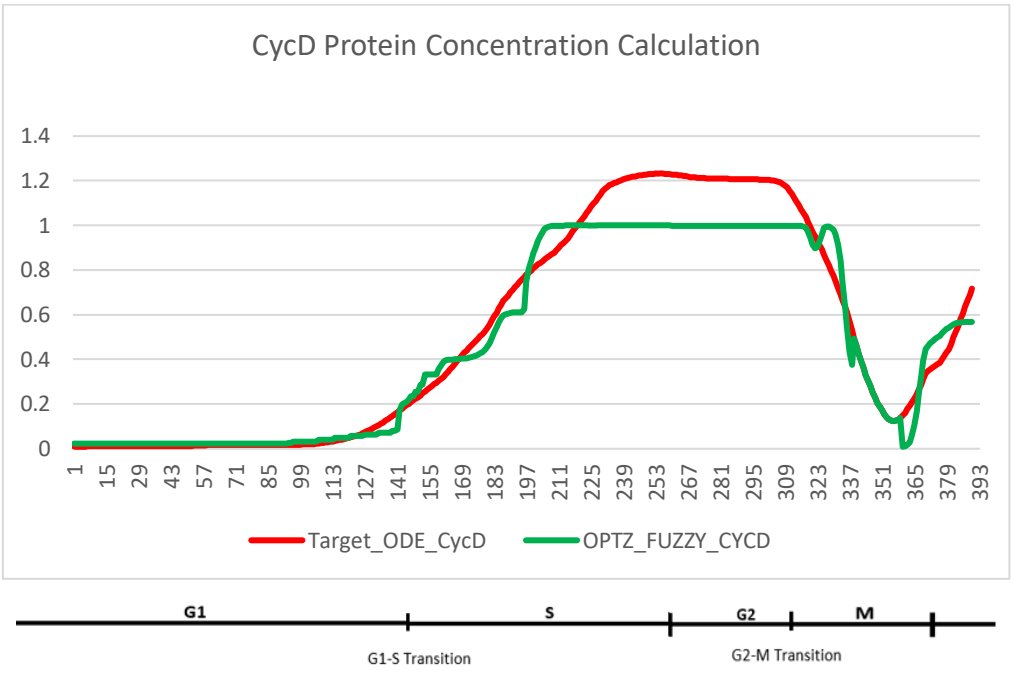
Table 5.3: Heuristically optimised (Pre-) and post-optimisation memberships function parameters values for input and output variables of the CycD fuzzy inference system

Input Variable 1	**MYC Fuzzy Set Parameters Pre-Optimisation**			**MYC Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.0911 0.193]	[0.138 0.569]	[0.2016 0.998]	[0.01866 0.1488]	[0.003481 0.9996]	[0.002386 0.9098]
Input Variable 2	**SCF Fuzzy Set Parameters Pre-Optimisation**			**SCF Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.2101 -0.0291]	[0.1402 0.505]	[0.1182 0.987]	[0.06521 1]	[0.982 0.003733]	[0.2382 0.6458]
Input Variable 3	**CDC20A Fuzzy Set Parameters Pre-Optimisation**			**CDC20A Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.108 -0.01386]	[0.0785 0.348]	[0.1496 0.913]	[0.1722 0.3678]	[0.07373 0.4671]	[0.02136 0.1288]
Input Variable 4	**CDC20B Fuzzy Set Parameters Pre-Optimisation**			**CDC20B Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.1001 1.04e-17]	[0.0677 0.3596]	[0.09201 0.9]	[0.4684 0.004563]	[0.9901 1]	[0.9911 0.002583]
Output Variable 1	**CycD Fuzzy Set Parameters Pre-Optimisation**			**CycD Fuzzy Set Parameters Post-Optimisation**		
	L	M	H	L	M	H
	[0.1956 0.0003806]	[0.1319 0.6103]	[0.2509 1.589]	[0.004199 0.003097]	[0.231 0.000921]	[0.1679 1]

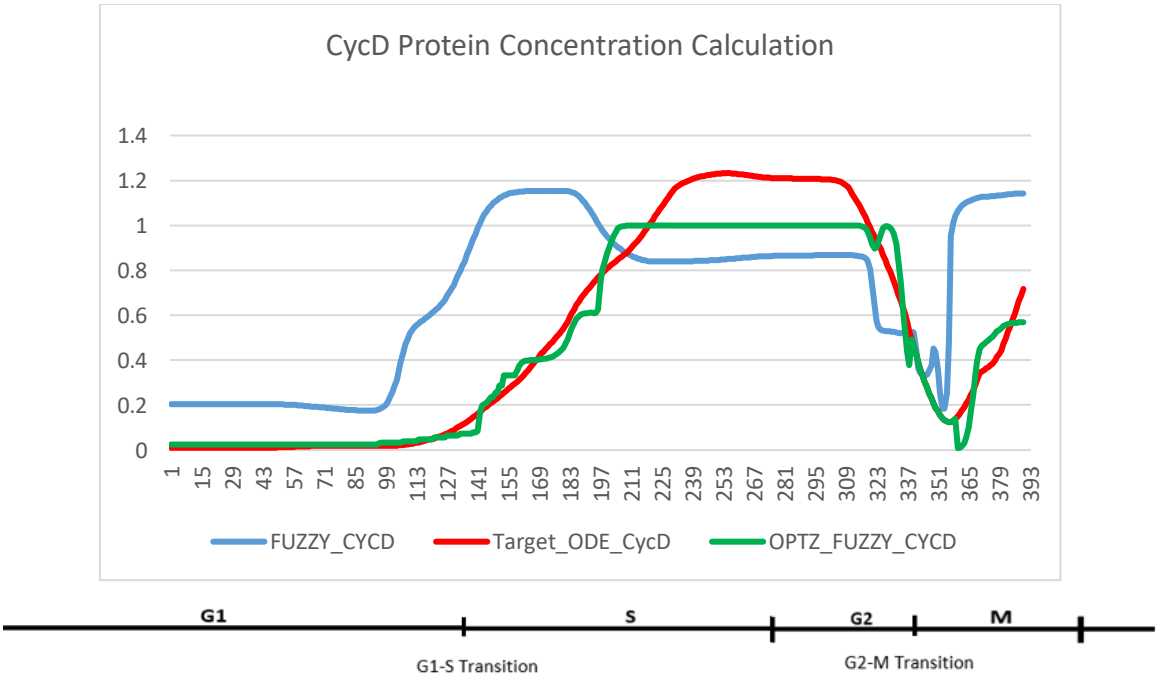
The newly optimised CycD fuzzy inference system was used to predict the protein concentration and activity of CycD over the cell cycle simulation. Furthermore, by using a similar data set as had been used in Section 4.11 for inputs, new optimised model was also validated. New results were more accurate and closer to the target ODE results as shown in Figure 5.13a. Furthermore, Figure 5.13b examines the differences between the results from the three systems - the optimised fuzzy model, the heuristically-designed model (pre-optimised) and the ODE target model. The figure shows a remarkable improvement from the PSO optimised CycD fuzzy inference model.

Figure 5.13b also interprets the improvement that we achieved using PSO optimisation for the CycD fuzzy inference model. The results revealed how the new optimised model (green line) can offer closer results to the desired (red line); likewise, the optimised model results are better than those from the original heuristic model (blue line). There is still a slight

difference between the optimised and ODE results but the improvement gained is quite significant.



(a)



(b)

Figure 5.13: CycD model results over cell cycle simulation from the heuristically-optimised fuzzy model, PSO optimised fuzzy model and ODE model. (a) Difference between the optimised fuzzy

model results (the green line) and the target ODE model results (the red line); (b) Comparison between all three model results. The green line represents the optimised fuzzy model, blue line represents the heuristically-optimised fuzzy model, and the red line represents the target ODE model.

5.6.3 Optimising CycA Fuzzy Inference System

Here, we exploited PSO optimisation to fine tune the fuzzy set of parameters in the CycA fuzzy inference system. As shown previously, this fuzzy inference system consists of four input variables and one output variable. Each variable has three fuzzy sets of Gaussian shape. From Equations 5.6 and 5.7, the total of parameters for this system is 30, so it is essential for these parameters to be tuned. Hence, in PSO implementation, each particle has 30 dimensions.

Optimising the CycA model via PSO also provides significant results. The minimum error (0.288982) was obtained in 1030 iterations as shown in Figure 5.14. However, the minimum error similar to the final value is reached far sooner than 1030 iterations.

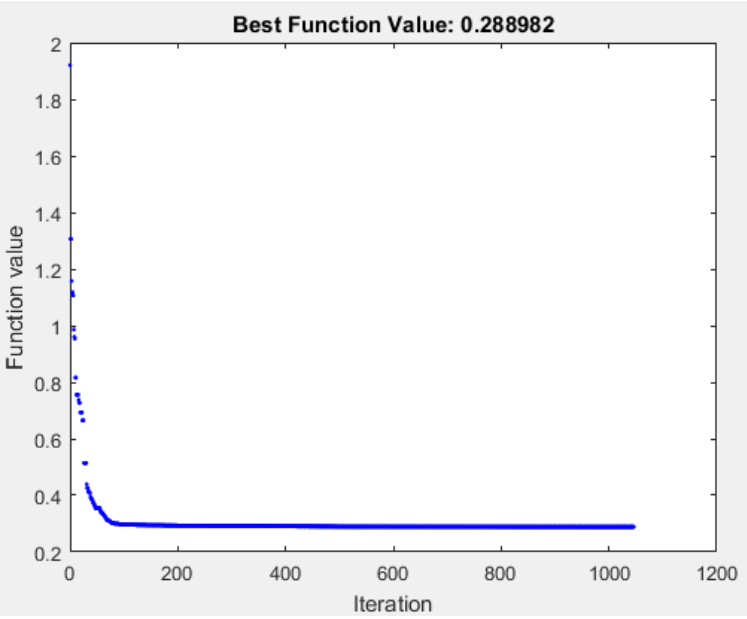
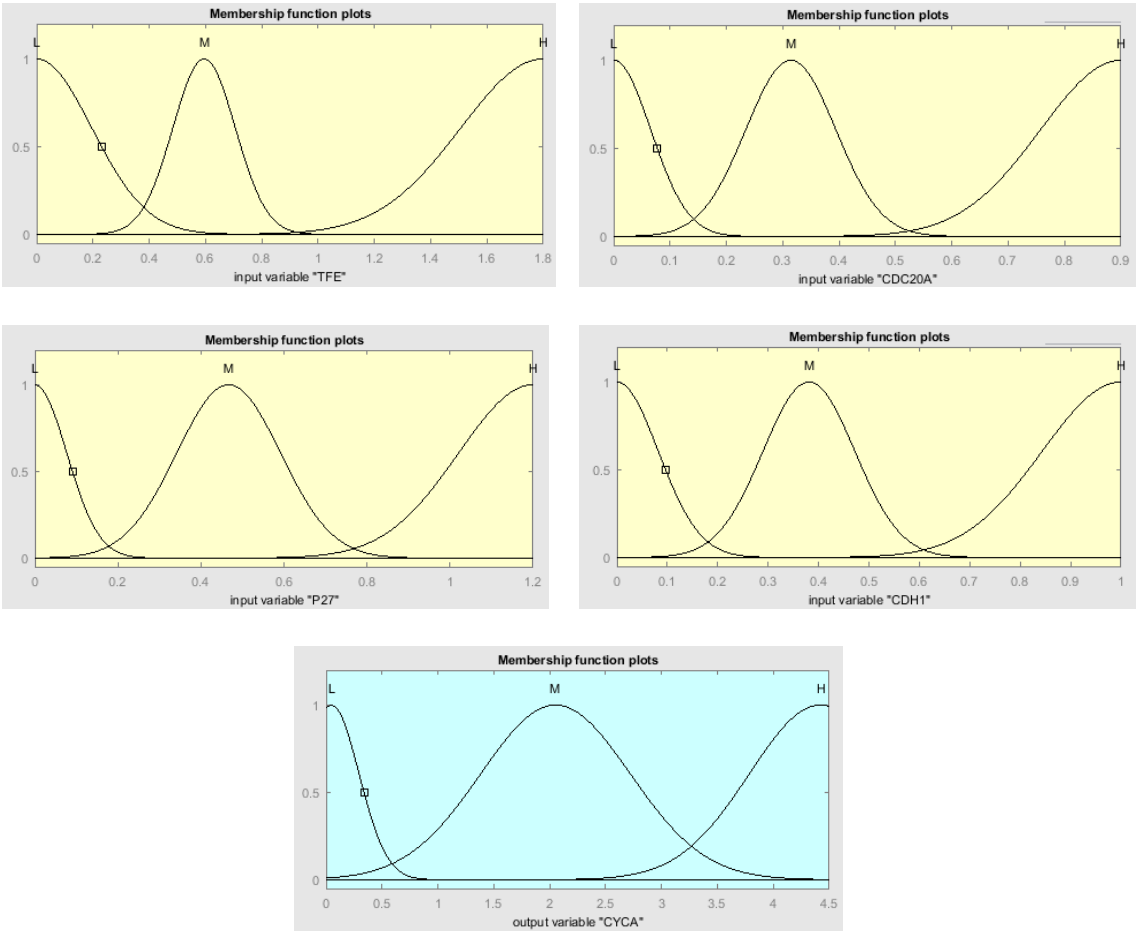


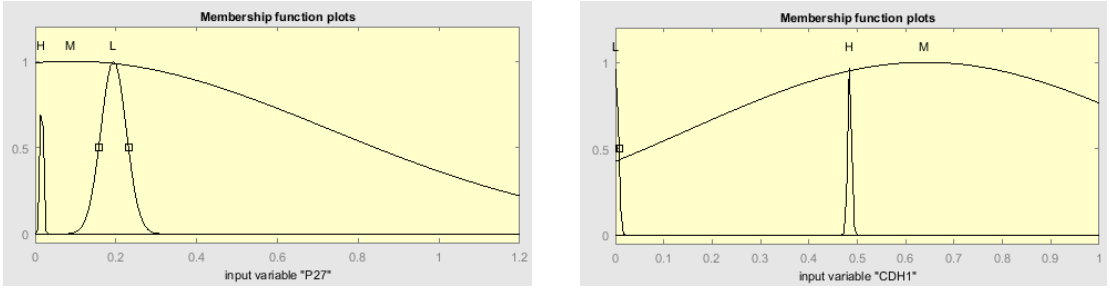
Figure 5.14: The reduction of error during PSO optimisation of the CycD fuzzy inference system

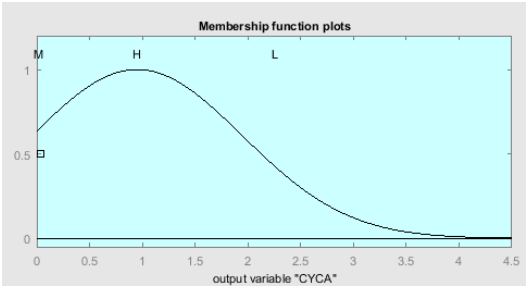
As expected, PSO optimised the MFs for the input and output variables of the CycA fuzzy inference system. The newly-adapted fuzzy sets are shown in Figure 5.15 and the new

parameter are presented in Table 5.4. This table presents the values of the parameters from the heuristically optimised and PSO optimised systems for the input variables and the output variable. We note that the values for the optimised fuzzy set parameters are quite different from the ones that were adjusted manually. This optimisation provides a new fuzzy inference system for CycA with comparable rules but with different optimised fuzzy sets.



(a)





(b)

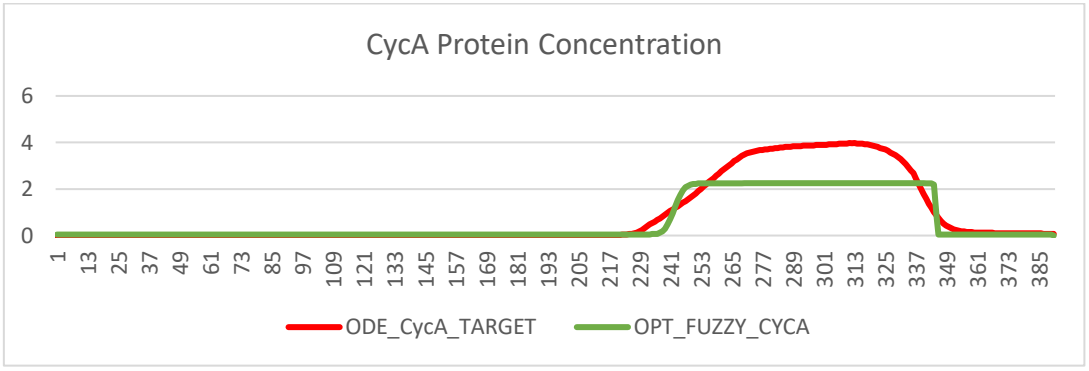
Figure 5.15: Heuristically optimised and PSO optimised fuzzy membership functions for the input and output variables in the CycA fuzzy inference system. (a) Heuristically optimised membership functions for the input variables and the output variable; (b) PSO optimised membership functions for the input variables and the output variable

Table 5.4: Heuristically optimised and PSO optimised fuzzy membership function parameter values for input and output variables of the CycD fuzzy inference system

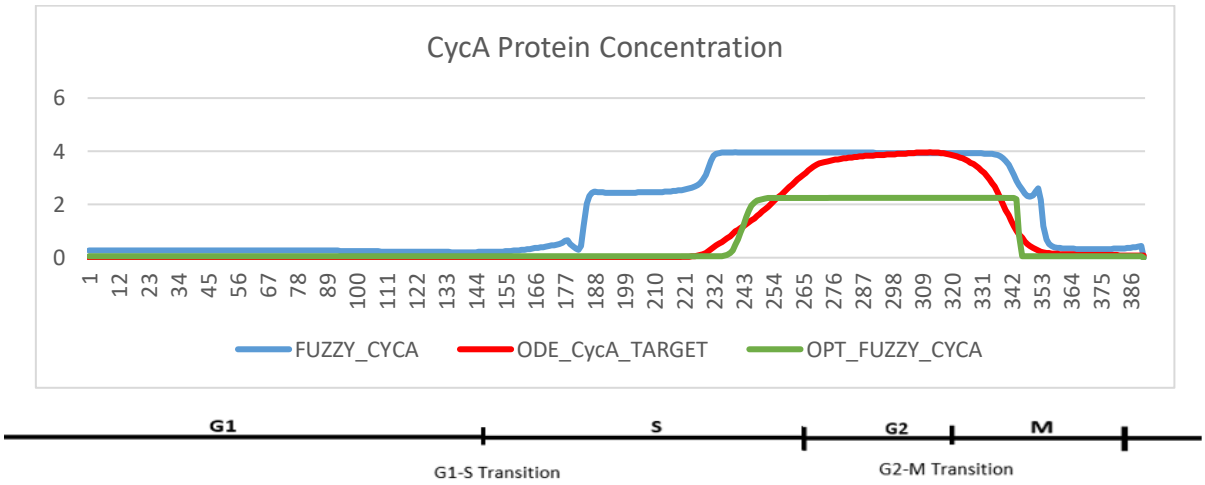
Input Variable 1	TFE Fuzzy Set Parameters Pre-Optimisation			TFE Fuzzy Set Parameters Post-Optimisation		
	L	M	H	L	M	H
	[0.1962 3.47e-17]	[0.111 0.595]	[0.2932 1.8]	[0.8432 0.0007668]	[0.5906 0.009638]	[0.4923 0.002487]
Input Variable 2	CDC20A Fuzzy Set Parameters Pre-Optimisation			CDC20A Fuzzy Set Parameters Post-Optimisation		
	L	M	H	L	M	H
	[0.06572 1.04e-17]	[0.0799 0.3143]	[0.1426 0.9]	[0.6141 0.9954]	[0.02334 0.9675]	[0.992 0.4593]
Input Variable 3	P27 Fuzzy Set Parameters Pre-Optimisation			P27 Fuzzy Set Parameters Post-Optimisation		
	L	M	H	L	M	H
	[0.07684 2.08e-17]	[0.125 0.4667]	[0.1793 1.2]	[0.03137 0.1943]	[0.6391 0.08892]	[0.00361 0.0164]
Input Variable 4	CDH1 Fuzzy Set Parameters Pre-Optimisation			CDH1 Fuzzy Set Parameters Post-Optimisation		
	L	M	H	L	M	H
	[0.08201 6.94e-18]	[0.091 0.381]	[0.1562 1]	[0.004926 0.001388]	[0.4903 0.64]	[0.003841 0.4843]
Output Variable 1	CycA Fuzzy Set Parameters Pre-Optimisation			CycA Fuzzy Set Parameters Post-Optimisation		
	L	M	H	L	M	H
	[0.2477 0.0476]	[0.669 2.05]	[0.6348 4.42]	[1e-05 0.9774]	[0.00179 0.0331]	[0.999 0.9502]

The newly optimised CycA fuzzy inference system was validated by predicting CycA activity over cell cycle simulation using the same data set that was used in Section 4.4. New results were closer to the target ODE results, as shown in Figure 5.16a. However, the main difference was the peak point between the optimised model results and the target results, while the activity pattern was the same for both models. Figure 5.16b examines the differences

between the results from the optimised fuzzy model, the heuristically optimised model and the ODE target model. The figure clearly elucidates the advancements made by the PSO optimisation. Results here revealed that the newly optimised model (green line) provided closer results to the desired (red line) and also that the optimised model was better than the heuristically optimised model (blue line).



(a)



(b)

Figure 5.16: CycA protein concentration and activity over cell cycle simulation from the heuristically optimised fuzzy model, PSO optimised fuzzy model and the ODE model. (a) Difference between the results from PSO optimised fuzzy model (green line) and the target ODE model (red line) for CycA concentration; and (b) Comparison between the three models. The green line represents the PSO optimised fuzzy model results, while the blue line represents the heuristically optimised fuzzy model results and the red line represents the target ODE model results.

5.7 Chapter Summary

Adjusting fuzzy set membership function parameter values heuristically (manually) is generally time consuming in most cases. This is due to the trial and error nature of the tuning process which can also be exhausting and frustrating. For this reason, in this chapter, we introduced an autonomous method to adjust membership functions in the design of fuzzy inference systems for mammalian cell cycle controller species. This method is called particle swarm optimisation (PSO). PSO was used to adjust parameters in some of the models developed heuristically, whose results were relatively different from the target ODE results. The results have revealed that the tuned fuzzy inference system provided more precise results than those optimised manually.

This chapter introduced a new contribution to the modelling field of mammalian cell cycle. We employed an intelligent system, PSO, to enhance the fuzzy computational approach that we used in Chapter 4 to model the mammalian cell cycle system. Indeed, the optimised fuzzy inference system provided a significant improvement and more accurate results in comparison with the same heuristically optimised fuzzy inference system. This chapter has provided a new methodological contribution similar to the contributions of the previous chapters (Chapters 3 and 4) where new methods or extensions were employed to model the complex mammalian cell cycle controller. With the successful application of PSO, this chapter offers an approach to automate the development of fuzzy inference systems for modelling cell cycle based on simple if then rules involving binary logic.

Chapter 6. Summary, Conclusions, Contributions and Future Research

In this thesis, we studied, analysed, designed and modeled a real complex system to provide advanced solutions to support varied decision making and prediction processes using powerful and novel computation methods for environmental and biological systems. The researcher has chosen a case study from a complex biological system; specifically, mammalian cell cycle which is a system that performs controlled cell division using specific gene-protein interaction networks in such a way that avoids abnormal cell proliferation leading to cancer. Cell cycle system is regulated by a set of elements constituting mammalian cell cycle controller. Therefore, in our study we involved the essential controllers of mammalian cell cycle. Precisely, this study reveals cell regulators that control cell transitions from one phase to another in cell division. It has been stated in the literature that most of the studies investigate simple biological systems such as yeast cell cycle while mammalian cell cycle system is less studied in the research domain due to the complicity of mammalian cells.

Therefore, the primary purpose of this study was to study the structural and dynamic analysis of mammalian cell cycle controller system. Basically, protein signaling pathway is studied that is involved in mammalian cell cycle controller to understand how these proteins communicate through biochemical interactions to achieve the global system output, and control cell transitions from one phase to another in mammalian cell division.

Furthermore, this research aimed to study and model interactions and events between cell cycle controller components. Cell components such as proteins and genes have complex interactions taking place at different speeds. It has been found that understanding the complex behavior among these components is a very challenging task. Therefore, to model this issue first, we have to understand the major components of the system. For instance, Cyclins are important elements in cell cycle system due to cyclin proteins having the most significant influence on the regulators of all the major events of the eukaryotic cell cycle. For example, Cyclins are the regulatory components of the Cyclin-Dependent Kinases (CDKs) that control all the major events of the eukaryotic cell cycle. There are four important classes of

cyclins in mammalian cells; these control cell cycle phases. Also, cell proliferation processes depend on the correct and fixed order of chromosome replication and separation and cell division during cell cycle.

This chapter summarizes the chapters of the thesis following the order of the research objectives outlined in Chapter 1. Herein, in each section, an elucidation of each objective and a summary of the results are provided. The end of each section highlights the conclusions and contributions made under each objective giving suggestions or directions for future research as appropriate.

6.1 Reconstruction of the mammalian cell cycle controller from the Available Information

6.1.1 Summary of the Results

The extended work consists of 13 network nodes (proteins) and 43 biological interactions in total. The Fauré et al. (2006) network had 9 nodes and 22 interactions, while Singhania et al. (2011) network had 9 nodes and 10 Boolean states considered as interactions. The two aforementioned networks (the Faure 2006 network and Singhania 2011 network) and our network share 9 nodes while almost all the interactions are updated in our model. Based on these modifications, we found that the extended cell cycle controller model provides improved results. For instance, each phase in mammalian cell cycle is controlled by a set of species such as cyclins and their transcriptions factors; we believe that the expansion of the components of the network and their connections with other network elements have incorporated the biological interactions in a better way than in previous models. It improves the coordination of transition of each phase of the cell cycle. Indeed, the developed model reveals the orderly sequence of regulatory protein activity along the cell cycle, contrary to the aforementioned models which failed to represent the circular behaviour of cell cycle phase transitions and the correct activities of system species over cell cycle. Also, regulatory

components that are newly added for regulating mammalian cell cycle system have meaningfully contributed to mammalian cell cycle phenomena.

6.1.2 Conclusion

- The protein and gene interaction network of mammalian cell cycle controller has been studied, modeled and analyzed in order to understand the dynamic events and changes in this system.
- I developed a new biological network with 13 proteins which mimic cell cycle controller. Further, it is represented by a qualitative- Boolean model that consists of 13 proteins to study the oscillations of the cyclins because cyclins behaviour has been studied scarcely in mammalian cells. The selection of cyclins are due to cyclins (Cyc D, Cyc E, Cyc A and Cyc B) being a family of proteins that control the progression of cells through the cell cycle by activating cyclin-dependent kinases (Cdk).
- The extended mammalian cell cycle controller network has been significantly enhanced with the new amendments.

6.1.3 Contribution and Future Directions

Here a new biological network was developed which mimic the most important regulatory items of the mammalian cell cycle controller. This work enhanced the exiting models by expanding the mammalian cell cycle controller network with novel biological findings to investigate it using a systems biology approach, and to meet the crucial current need in the mammalian cell cycle system research field to bridge the gaps in our understanding.

We cannot assert that our developed network can be considered as a complete network since the biological knowledge is incomplete; there are still missing species and relations that can be discovered to complete the understanding. For instance, in the previous models, they assumed that some of the elements were directly linked in the cell cycle network, but few years later researchers have discovered new intermediate biological elements and relations. In this regard, we trust that employing any new elements that may be found in future and updating their relations to the network should provide more significant biological results.

6.2 Study of Network Dynamics and Comparing Results with the Existing Network Results (Synchronous Boolean Model)

The addition of novel information was expected to provide improved results compared to those in the literature such as Fauré et al. (2006) model and Singhania et al. (2011) model as the new elements completed the missing links and assumptions in the previous models of the cell cycle system controller. Indeed, we assessed the system dynamics with synchronous update Boolean model to answer the questions in below.

- What are the dynamics of the mammalian cell cycle controller network?
- What are the improvements made by the new amendments?
- What are the important components of the network to regulate cell cycle phase transitions?
- Is the developed cell cycle controller network considered a robust and resilient network?

6.2.1 Summary of the Results

Our model has improved the representation of the dynamic behavior of cell cycle control system while preserving the core characteristics of the cell cycle controller system found in previous models. Concerning the comparative analysis of our model and previous models, both models (Fauré et al. (2006) model and our model) display common results with regard to cell cycle system species' activity over cell cycle phase transitions when simulated by applying either exhaustive search or heuristic search algorithms. Further, in terms of the Boolean attractor to explain biological phenomena, the two models agree on the number of attractors - both models provide two attractors. But, our model provides better results in revealing individual element behaviour and periodic behaviour of cell cycle. Specifically, the first attractor is a steady state attractor that represents the G0 phase when no growth factors are present; we shared the same result with Faure et al. (2006) model to specify which elements are active. In this case, just few elements should be active such as CDH1, P27, and

RB. The second attractor with 12 states (limit cycle- cell cycle- transitions phases) shows the dynamic behaviour of the cell cycle system and these results broadly agree with literature. Consequently, this is considered a good outcome since the additional new nodes in our developed Boolean model have not disagreed with system regulation captured by the available models in the literature. However, although the second attractor in Fauré et al. (2006) model is the cell cycle attractor, it does not show system activity properly, i.e., cell cycle stages are not clearly or continuously represented.

Furthermore, we applied an additional mutation study on system elements to specify which element can affect the system significantly. Also, we studied the long-term behaviour of our system and evaluated system behaviour by making a comparison between the attractors reached from the initial states and their perturbed copies. It shows how each cyclin lead and move the cell cycle from phase to phase. Our model also shows cyclin synthesis, degradation, and other system component activity over cell cycle phase transitions.

6.2.2 Conclusions

- The developed synchronous Boolean cell cycle controller model explains the biological phenomena in a creative way. It explains the impact of the presence or absence of growth factor signals to establish cell cycle for the new born mammalian cells. In addition, it shows the correct dynamic behaviour for cell cycle elements; especially cyclins activity; our model explains and supports the facts about how each cyclin leads the cell cycle phase transitions. This includes the biological interactions between cyclins and transcription factors and other regularity components.
- Our model reveals the activities of all species that regulate the cell cycle system.
- Our model is identified as robust and resilient against biological noise. The Boolean network that simulated the cell cycle controller revealed that it was robust when facing a small amount of noise. For instance, we measured the impact of the noise when we applied it to the current state of the network.

- In this work we studied the long term behaviour of the developed model and the results agreed with those in the literature.

6.2.3 Contributions and Future Directions

Based on the above findings, we believe that the developed model in this thesis reveals the state transition patterns in the attractors more descriptive of the cell cycle system as our developed model involved novel, more accurate and realistic regulatory mechanisms in the system than other models. For illustration, in the Fauré et al. (2006) model some of the element behaviours are not presented correctly – for example, CycE was shown to become active at the beginning, then silent, and then become active again at a later stage. Furthermore, the mutation study shows that the proposed model can be considered as a good qualitative prediction model. It can be utilised to predict the impact of any biological mutation on cell cycle controller such as protein loosing function and protein over expressions. This can help study cell cycle diseases such as cancer.

However, the developed process relied on two bench mark models, and they have not included cell cycle checkpoints. Neither has our model. Therefore, future research should include additional species that represent the activity of checkpoints to control DNA damage.

6.3 Modelling the cell cycle controller Network with an Asynchronous Boolean Approach

Difficulty in mimicking the realistic behaviour of biological events is considered as a drawback of synchronous Boolean models, specifically, mimicking the state transitions of the network elements. Therefore, the researcher decided to apply another updating method on the - Boolean network - asynchronous update - to represent the times delays between biological interactions.

The asynchronous update was applied on the Boolean model of the cell cycle controller to answer the following questions.

- What are the main differences in the result between the synchronous and synchronous update?
- What are the cell cycle dynamics from the asynchronous approach?
- How does this update reflect the realistic biological behaviour of the mammalian cell cycle controller?

6.3.1 Summary of the Results

The asynchronous update revealed the correct biological interactions and cyclin oscillation with time delays. Basically, we applied the exhaustive search and heuristic search and the results were compatible with the available results in the literature such as the result from Fauré et al. (2006) model. The exhaustive search provided two attractors, the first revealed the steady state attractor which represented the G0 phase, while the second attractor was a complex attractor. With the heuristic search algorithm, asynchronous update provided more realistic behaviour than the synchronous update. Furthermore, asynchronous update revealed the time delays between species activities by using randomization in the selection of proteins to update their states from active to inactive and vice versa.

6.3.2 Conclusions

- The asynchronous model correctly captured the temporal behaviour of most of the events in the cell cycle controller system.
- The asynchronous model also showed resilience and robustness of the cell cycle control system that is consistent with the results in the literature.

6.3.3 Contributions and Future Directions

The asynchronous model provided realistic results that are compatible with biological knowledge, but it needs prior knowledge to explain and investigate system behaviour since

the randomisation in each simulation can provide different results. Therefore, we believe that there should be an alternative method which can provide consistent results but without randomization update. For instance, the future work could use the synchronous update with the incorporation of actual timing to control the time of firing of the rules; this can provide more realistic results.

6.4 Modelling the Cell Cycle Controller Network with Synchronous Update with Time Variables Boolean Model

The Synchronous update is easy to explain and investigate while asynchronous update is complex to explain and investigate. Here, in this model we depend on the simplicity of synchronous update and incorporate realistic time delays between cyclin activation and deactivation and activity of other system elements.

Involving timing with biological interactions to reflect the time delays, this attempt can answer the following questions:

- What are the dynamics of this cell cycle controller network?
- What is the advancement achieved from this model?
- How important are the cyclins to the maintenance of system behaviour?

6.4.1 Summary of the Results

The main difference between the prior synchronous model that we developed and this model is that when a node turns active, different time steps (delays) are required to activate each of its targets. Here, timing is incorporated into Boolean interactions; specifically, the cyclin nodes take extra time steps to be active. However, this technique is time consuming in computing with exhaustive search. Therefore, the researcher utilised a heuristic search to evaluate this model. This model provides realistic behaviour with time delays to activate or inactivate cyclins nodes and other cell cycle controller species.

6.4.2 Conclusion

- The model correctly captured the temporal signalling events in the cell cycle control system, such as cyclins and their transcription factors.

- The model generated biologically realistic steady state dynamics of the cell cycle control system. It provides four attractors. The first is a steady states attractor which represents G0 phase. The second attractor is a limit cycle that represents the entire cell cycle phases with time delays, while the rest of the attractors are spurious which could indicate missing or undiscovered knowledge.

- The result from this model is compatible with biological knowledge in the literature. It provides the steady state attractor G0 and full cell cycle transition.

6.4.3 Contributions and Future Directions

Our contribution from this model is cooperating time delays that properly describe the slow biological interactions inside the cell cycle controller system into the network updating method. We believe that this model revealed cell cycle activity (e.g., cyclins activities) more realistically than previous models. However, future studies should investigate the spurious attractors and attempt to link them with biological research and undiscovered knowledge.

6.5 Modelling the Cell Cycle Regulatory Network with a semi Continuous Approach

The literature reported that the most popular and accurate models used in modelling cell cycle signaling network are continuous models. Continues models are based on mathematical models such as ordinary differential equations that provide accurate continuous solutions but their parameters need estimation from data which are limited. Therefore, discrete models such as Boolean that can use limited data and still be helpful in modelling and understating complex systems have become more popular. However, Boolean model provides only qualitative analysis and results since it assumes that each node in the network takes only one

state either: 1 (‘ON’/’active’) or 0 (‘OFF’/’inactive’); this assumption cannot clearly reveal the temporal development of the dynamical events. Further, Boolean models cannot show the intermediate states for protein concentrations. Hence, it was expected to enhance the discrete Boolean model with a continuous model, but the available continuous models depend on kinetic parameters which are limited or not available. Also, available biological knowledge is generally imprecise. These draw backs inspired us to select an alternative method that covers the shortcoming of the aforementioned method. The alternative method is simple, similar to Boolean model, and has the ability to work with ambiguous and missing data; we select fuzzy logic because it can cover the shortcoming of both discrete and continuous models.

This model attempted to answer the following questions:

- Can we transform the qualitative model – Boolean model - into a fuzzy logic model?
- If fuzzy logic relies on the prior knowledge and vague information even with missing data such as the kinetic parameters, then can it work with a flexible approach like the Boolean model and be accurate like the ODE model in order to simulate the continuous behaviour of cell cycle control system?
- Can the cell cycle controller in the form of a fuzzy logic model provide the promised results? Are the semi-continuous results from the fuzzy model better and more flexible/continuous than the discrete results from the Boolean model to investigate and mimic cell cycle controller system?
- What are the best features of a fuzzy logic methodology that we must utilise to consider the developed fuzzy logic model an accurate model that is similar to ODE model?
- The biological knowledge is imprecise; Can the Fuzzy systems handle this kind of knowledge.

6.5.1 Summary of the Results

A semi quantitative representation was derived from the developed Boolean model. This work proposed an intuitive and simple fuzzy logic model to provide a semi quantitative

representation of a system and its dynamics. Each element in the cell cycle controller is represented as a fuzzy inference system (FIS). These fuzzy inference systems have been employed to compute the proteins concertation levels and reveal element activities. According to results, fuzzy logic model provides approximate continuous dynamic for the discrete events using limited available data. Furthermore, this model produces realistic concentrations of proteins during each cell cycle event as well as how the system operates as a whole.

Finally, the entire fuzzy inference systems have been validated by utilising a data set that is available in the literature (Ali Abroudi, 2017). Then we accomplished an individual comparison study to make sure that the developed fuzzy inference systems can mimic each element interactions, behaviours and provide accurate results for protein synthesis and degradations. Indeed, the semi continuous results from the fuzzy inference system have been compatible with results from a recent continuous ODE model Ali Abroudi (2017) model. Further, in some cases our fuzzy inference system provided better result than the aforementioned ODE model. For instance, it describes the periodic behaviour of the entire set of system components from one cell cycle to the next, while the ODE model failed to describe the periodic behaviour for some elements such as P27, RB and CDH1.

6.5.2 Conclusions

- The transformation of the Boolean model into a semi continuous model by using fuzzy logic provided accurate results. It reproduced biologically-observed behaviour of the cell cycle controller signalling system.

- Fuzzy inference system can be considered as a useful approach to describe the nonlinear behaviour of protein activities such as protein synthesis and degradation.
- The fuzzy logic model was able to work with missing knowledge such as kinetic parameters and provide accurate results. For instance, it relies on human reasoning rather than exact numbers to compute protein concentration levels and describe species activities.

6.5.3 Contribution and Future Directions

Our contribution in this objective is the transformation of the Boolean model of the cell cycle system into a semi continuous model to better understand the dynamics of the system. This was a successful attempt to reveal the nonlinear behaviour of the cell cycle controller and describe the continuous behaviour without needing kinetic parameters such as ODE models. Also, the proposed model provided in some cases a better results than the recent ODE continuous model - Ali Abroudi (2017) model. We believe that the simplicity and the flexibility of fuzzy logic make it an attractive alternative method to ODE models in modeling cell cycle. However, the design of the fuzzy inference system relies on heuristic design and manual adjusting of the sensitive membership function parameters; this can work well for some fuzzy inference models in the system but some other fuzzy inference models may require more enhancements to get accurate results. Therefore, we used an artificial intelligent technique - particle swarm optimization PSO - method to enhance the functionality of these models. Furthermore, we believe that development a full fuzzy logic network can provide more insight for the cell cycle controller system, this is one of our future work.

6.6 Enhancing Biological Fuzzy Inference Models by Employing Particle Swarm Optimization

Our entire set of fuzzy inference models have been built and designed based on prior knowledge. Further, the computing was based on human/biological reasoning instead of typical ordinary differential equations. Also, developing the fuzzy rules was an easy, while tuning the fuzzy parameters was more complex since we adjusted them manually. Adjusting the fuzzy parameters is an extremely sensitive task as it can make the outcome either very close to the realistic results or quite vary from them. With careful tuning, the outcomes from the developed fuzzy inference systems agreed well with the biological evidence available for cell cycle controller system induced cyclin oscillations and other species activities. However, we noticed that few models needed more enhancement to provide accurate results similar

to the continuous model used for comparison. We believed that the best way to enhance the functionality of these models was to tune the fuzzy parameters automatically.

This investigation attempted to answer the following questions:

- What is the best way to enhance the functionality of the developed fuzzy inference models?
- What are the best features of the selected optimization method, the Particle Swarm Optimisation (PSO)?
- What advancement can be achieved by using PSO?
- Does the optimization make the result more accurate?

6.6.1 Summary of the Results

We automatically adjusted the fuzzy parameters of the fuzzy inference systems. We used a novel way to implement a specific programing code to integrate an intelligence method based on a biological social system - PSO - to tune the fuzzy parameters automatically. Consequently, the outcome from fuzzy models that adjusted parameters automatically were more accurate than the results from the manually adjusted models.

6.6.2 Conclusions

- The behavior of cell cycle controller fuzzy inference models and outcomes are sensitive to small changes in the adjustment to fuzzy parameters.
- The automatic tuning of fuzzy parameters is better than manual adjusting since it required less efforts but, it needs an efficient approach to do the fine tuning. Therefore, this study integrated and imported a method that is widely used in the engineering filed into complex biological systems modeling.

- The automatically adjusted fuzzy inference systems, such as CycD, CycA and SCF fuzzy inferences models, provided better results than the same fuzzy inference systems that were adjusted and designed manually.

6.6.3 Contributions and Future Directions

One of the main contributions of this part is related to modelling mammalian cell cycle. This is considered as a novel modelling attempt in the field of cell cycle modelling. The researcher employed an intelligent systems approach, PSO, to empower fuzzy logic computational approach that we used to model the mammalian cell cycle system. This technique was imported from the electrical and industrial engineering modelling field and integrated into cell cycle modelling.

Concerning future research, we intend to extend our developed cell cycle controller network. This extension should include up-to-date discovered elements. Furthermore, it can involve more primary elements that perform other biological activities such as protein P25 which play roles in cell cycle checkpoints and DNA damage. In this study, we did not investigate DNA damage and cell cycle checkpoint element activities. Checkpoints will be one of our future works that we intend to implement as an extension of our research. Certainly, the models which this thesis is based on such as Singhania et al. (2011) model and Fauré et al. (2006) model did not involve checkpoints in their implementations; similar our study also only focused on the major controllers of the cell cycle system activity like Cyclin flows over cell cycle. Therefore, this expansion will update the entire models that we developed such as Boolean model and fuzzy logic model. Further, PSO would play a crucial part in enhaning these computational models.

Concerning future studies, the researcher studied and investigated the most important regulators of cell cycle such as cyclins. Cyclin can be used in the investigation of human diseases. For instance, cyclin A overexpression correlates with poor prognosis in breast cancer. Also, the presence of cyclin D is required to maintain tumor growth in ErbB2-induced

mammary carcinomas (breast cancer). Furthermore, the abundance of cyclin E transcript and protein is increased in carcinomas of the lung, gastrointestinal tract, and breast in addition to lymphomas and leukemia. The new model proposed in this thesis can be used to examine the localization of system elements (element abundances such as abundance of cyclins in cell cycle) and can mimic the fluctuation of cyclins during cell growth and division.

Appendix

Appendix 1

Boolean model utilized in the study (in both the Synchronous and asynchronous Approaches) is given in this Appendix 1.

This model is an assortment of statements collected on node dependencies from literature as elucidated in Chapter 3 and expresses as Boolean rules according to:

Targets, Factors such as:

MYC,MYC

CDH1,(!CYCA & !CYCB) & !CYCE | (CDC20A & CDC20B)|(P27 &!CYCB)

P27,(!CYCD & (!CYCD & ! P27)) & (!CYCD & !CYCE) |(!CYCE & !CYCA)

RB,!CYCD & (!CYCE & !CYCA)

CYCD, MYC & (!SCF & (!CDC20A & !CDC20B))

TFE,(!RB & !CYCA & !CYCB) | (P27 & !RB & !CYCB)

SCF,!CDH1 & CYCE

CYCE,TFE|(!SCF & !P27)

CYCA,(TFE & !P27) | (!CDC20A & !CDH1)

TFB,CYCA

CYCB, TFB| (!CDH1 & !CDC20B)

CDC20A,CYCB

CDC20B,CYCB

Appendix 2

This Appendix includes all the ODE models related to the mammalian cell cycle regulator (Abroudi et al. (2017).

$$\frac{d[aSCF]}{dt} = k_{92} \cdot [aCycE\_Cdk2] \cdot [iSCF] - k_{91} \cdot [aSCF] \cdot [aAPC\_Cdh1], \text{(4.20)}$$

Where $\frac{d[aSCF]}{dt}$ represents the changes in the concentration levels of SCF over time. This ODE equation consists of activator CycE and inhibitor CDH1 of SCF. The process also depends on kinetic parameter values, k92 and k91, relating to SCF production and degradation, respectively, to define the changes in SCF protein level over time.

We also need to be aware that each ODE equation consists of an activator and a deactivator and a set of kinetic parameters. These components control the changes in protein concentration levels over the cell cycle time. However, the most evident bottleneck is the lack of availability of kinetic parameter values. ODE representation is considered more complex continuous form (Chapter 2) compared to Boolean representation but the required set of kinetic parameter values are very hard to find for signalling networks. In the rest of the Appendix, ODE functionality, including kinetic parameters, activators and inhibitors will not be explained since each ODE model follows the same basics and principles in mimicking biological events.

$$\frac{d[Cdh1]}{dt} = k_{145} \cdot [iAPC\_Cdh1] - k_{144} \cdot ([aCycB\_Cdk1\_Nuc] + [aCycA\_Cdk2] + [aCycE\_Cdk2]) \cdot [aAPC\_Cdh1], \quad (4.21)$$

$$\frac{d[p27]}{dt} = k_{49} + k_{47} \cdot [p27\_CycD\_Cdk4] + k_{50} \cdot [p27\_CycE\_Cdk2] + k_{51} \cdot [p27\_aCycA\_Cdk2] - (k_{48} \cdot [CycD\_Cdk4] + k_{52} \cdot [aCycE\_Cdk2] + k_{53} \cdot [aCycE\_Cdk2] + k_{54} \cdot [aCycA\_Cdk2] + k_{55} \cdot [aCycA\_Cdk2]) \cdot [p27], \quad (4.22)$$

$$\frac{d[CycD]}{dt} = k_{39} + k_{40} \cdot [acMyc] + k_{41} \cdot [CycD\_Cdk4] - (k_{44} + k_{42} \cdot [Cdk4] + k_{43} \cdot [aSCF] + k_{147} \cdot [aAPC\_Cdc20]) \cdot [CycD] \quad (4.23)$$

$$\frac{d[E2F\_pRb]}{dt} = k_{56} \cdot [E2F] \cdot [pRb] - (k_{57} \cdot [CycD\_Cdk4] + k_{58} \cdot [p27\_CycD\_Cdk4] + k_{59} \cdot [p21\_CycD\_Cdk4]) \cdot [E2F\_pRb], \quad (4.24)$$

$$\frac{d[CycE]}{dt} = k_{70} \cdot [E2F] + k_{72} \cdot [iCycE_{Cdk2}] - (k_{73} + k_{71} \cdot [Cdk2] + k_{74} \cdot [aSCF]) \cdot [CycE \quad (4.25)$$

$$\frac{d[CycA]}{dt} = k_{75} \cdot [E2F] + k_{76} \cdot [aBMyb] + k_{77} \cdot [NFY] + k_{78} \cdot [iCycA\_Cdk2] - (k_{79} + k_{80} \cdot [Cdk2] + k_{81} \cdot [aAPC\_Cdc20] + k_{82} \cdot [aAPC\_Cdh1]) \cdot CycA, \quad (4.26)$$

$$\frac{d[TFB]}{dt} = k_{96} \cdot [aCycA\_Cdk2] - k_{97} \cdot [TFB], \quad (4.27)$$

$$\frac{d[CycB]}{dt} = k_{123} \cdot [NFY] + [iCycB\_Cdk1\_Cyto] \cdot (k_{124} + k_{125} \cdot [Gadd45]) - (k_{126} + k_{127} \cdot [Cdk1] + (k_{128} \cdot [aAPC\_Cdc20] + k_{129} \cdot [aAPC\_Cdh1])) \cdot [CycB], \quad (4.28)$$

$$\frac{d[pRb]}{dt} = (k_{60} \cdot [aCycE\_Cdk2] + k_{61} \cdot [aCycA\_Cdk2]) \cdot [E2F\_pRbPP] - k_{64} \cdot [pRbPPP], \quad (4.29)$$

$$\frac{d[aAPC\_Cdc20]}{dt} = (k_{143} \cdot [aCycB\_Cdk1\_Nuc] + k_{148} \cdot [aPlk1]) \cdot [iAPC\_Cdc20] - k_{142} \cdot [aAPC\_Cdh1] \cdot [aAPC\_Cdc20], \quad (4.30)$$

Appendix 3

This Appendix includes all the Fuzzy inference models related to the mammalian cell cycle regulator.

```
[System]
Name='CDC20A'
Type='mamdani'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=3
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCB'
Range=[0 0.9]
NumMFs=3
```

```
MF1='L':'gaussmf',[0.157 3.47e-18]
MF2='M':'gaussmf',[0.0822 0.486095238095238]
MF3='H':'gaussmf',[0.069765719309373 0.9]

[Output1]
Name='CDC20A'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.133689542637929 6.94e-18]
MF2='M':'gaussmf',[0.0977 0.470899470899471]
MF3='H':'gaussmf',[0.0887518812470284 1]

[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1

Name='CDC20B'
Type='mamdani'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCB'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.0958698270607015 0.00736]
MF2='M':'gaussmf',[0.113 0.416]
MF3='H':'gaussmf',[0.0857410579338381 0.895]

[Output1]
Name='CDC20B'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.08763 8.328e-18]
MF2='M':'gaussmf',[0.1156 0.6084]
MF3='H':'gaussmf',[0.04988 1.2]

[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1

[System]
Name='CDH1_A'
Type='mamdani'
Version=2.0
NumInputs=7
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
```

```
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCD'
Range=[0 1.5]
NumMFs=3
MF1='L':'gaussmf',[0.240978209208704 1.39e-17]
MF2='M':'gaussmf',[0.179206899860772 0.664]
MF3='H':'gaussmf',[0.242865590987121 1.53]

[Input2]
Name='CYCE'
Range=[0 1.6]
NumMFs=3
MF1='L':'gaussmf',[0.174358126196694 2.08e-17]
MF2='M':'gaussmf',[0.135 0.702645502645503]
MF3='H':'gaussmf',[0.206713242398143 1.6]

[Input3]
Name='CYCA'
Range=[0 3.8]
NumMFs=3
MF1='L':'gaussmf',[0.388486082724335 0]
MF2='M':'gaussmf',[0.557 1.64095238095238]
MF3='H':'gaussmf',[0.738550464959449 3.8]

[Input4]
Name='CYCB'
Range=[0 0.8]
NumMFs=3
MF1='L':'gaussmf',[0.0871790630983469 1.04e-17]
MF2='M':'gaussmf',[0.124706504125888 0.341]
MF3='H':'gaussmf',[0.132116724489247 0.8]

[Input5]
Name='CDC20A'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.106726945803389 6.94e-18]
MF2='M':'gaussmf',[0.0887518812470285 0.5]
MF3='H':'gaussmf',[0.113467595012024 1]

[Input6]
Name='CDC20B'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.0685299336211232 6.94e-18]
MF2='M':'gaussmf',[0.10222419213202 0.418]
MF3='H':'gaussmf',[0.120208244220659 1]

[Input7]
Name='P27'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.114591036546796 2.08e-17]
MF2='M':'gaussmf',[0.154751824531844 0.467]
MF3='H':'gaussmf',[0.160427451165515 1.2]

[Output1]
```

```
Name='CDH1'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.0662830505515782 6.94e-18]
MF2='M':'gaussmf',[0.0786 0.429783068783069]
MF3='H':'gaussmf',[0.171886554820194 1]

[Rules]
3 0 0 0 0 0 1, 1 (1) : 1
0 3 0 0 0 0 1, 1 (1) : 1
0 0 0 2 0 0 0, 1 (1) : 1
0 0 1 1 3 3 1, 2 (1) : 1
1 1 1 1 0 0 0, 3 (1) : 1

System]
Name='CYCB'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=6
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='TFB'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.102120835510821 1.04e-17]
MF2='M':'gaussmf',[0.062 0.354142857142857]
MF3='H':'gaussmf',[0.144586925525222 0.9]

[Input2]
Name='CDH1'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.104480062733844 6.94e-18]
MF2='M':'gaussmf',[0.0865 0.430973544973545]
MF3='H':'gaussmf',[0.171886554820194 1]

[Input3]
Name='CDC20B'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.100098640748231 1.04e-17]
MF2='M':'gaussmf',[0.0859 0.376619047619048]
MF3='H':'gaussmf',[0.144586925525222 0.9]

[Output1]
Name='CYCB'
Range=[0 0.8]
NumMFs=3
MF1='L':'gaussmf',[0.0889789512500367 0.00423]
MF2='M':'gaussmf',[0.0618 0.38337037037037]
MF3='H':'gaussmf',[0.157281814868152 0.8]

[Rules]
```

```
1 0 0, 1 (1) : 1
2 0 0, 2 (1) : 1
3 0 0, 3 (1) : 1
0 0 1, 3 (1) : 1
0 1 3, 1 (1) : 1
0 3 0, 1 (1) : 1

[System]
Name='CYCE_4'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=11
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='TFE'
Range=[0 1.8]
NumMFs=3
MF1='L':'gaussmf',[0.18 -0.0278095238095239]
MF2='M':'gaussmf',[0.11526510146766 0.9]
MF3='H':'gaussmf',[0.305351409151169 1.8]

[Input2]
Name='SCF'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.131442659568384 6.94e-18]
MF2='M':'gaussmf',[0.0842581151079385 0.5]
MF3='H':'gaussmf',[0.1112 1]

[Input3]
Name='P27'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.0984134784460721 2.08e-17]
MF2='M':'gaussmf',[0.101109738129526 0.6]
MF3='H':'gaussmf',[0.163123710848969 1.2]

[Output1]
Name='CYCE'
Range=[0 2]
NumMFs=3
MF1='L':'gaussmf',[0.3617 2.6e-17]
MF2='M':'gaussmf',[0.1057 0.894]
MF3='H':'gaussmf',[0.356 2.025]

[Rules]
0 0 3, 1 (1) : 1
1 1 3, 1 (1) : 1
2 1 1, 2 (1) : 1
3 1 1, 3 (1) : 1
0 2 1, 2 (1) : 1
0 3 1, 1 (1) : 1
0 1 3, 1 (1) : 1
```

```
1 1 1, 1 (1) : 1
1 0 0, 1 (1) : 1
2 0 0, 2 (1) : 1
3 0 0, 3 (1) : 1

[System]
Name='P27_A'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCD'
Range=[0 1.4]
NumMFs=3
MF1='L':'gaussmf',[0.0927962707722095 1.39e-17]
MF2='M':'gaussmf',[0.165 0.466925925925926]
MF3='H':'gaussmf',[0.335010265669163 1.4]

[Input2]
Name='CYCE'
Range=[0 1.6]
NumMFs=3
MF1='L':'gaussmf',[0.235477075514214 -0.00847]
MF2='M':'gaussmf',[0.149 0.546206349206349]
MF3='H':'gaussmf',[0.386463887961744 1.6]

[Input3]
Name='CYCA'
Range=[0 3.8]
NumMFs=3
MF1='L':'gaussmf',[0.8069 0]
MF2='M':'gaussmf',[0.8068 1.9]
MF3='H':'gaussmf',[0.8069 3.8]

[Output1]
Name='P27'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.0660583622446237 2.08e-17]
MF2='M':'gaussmf',[0.0902 0.460587301587302]
MF3='H':'gaussmf',[0.246707761036044 1.2]

[Rules]
1 1 1, 3 (1) : 1
2 1 1, 2 (1) : 1
3 1 1, 1 (1) : 1
3 3 1, 1 (1) : 1
0 0 2, 1 (1) : 1

[System]
Name='RB_3'
Type='mamdani'
```

```
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCD'
Range=[0 1.4]
NumMFs=3
MF1='L':'gaussmf',[0.187165359693101 1.39e-17]
MF2='M':'gaussmf',[0.0811 0.54137037037037]
MF3='H':'gaussmf',[0.281534448613992 1.4]

[Input2]
Name='CYCE'
Range=[0 1.6]
NumMFs=3
MF1='L':'gaussmf',[0.159978074551606 2.08e-17]
MF2='M':'gaussmf',[0.134961278455292 0.597]
MF3='H':'gaussmf',[0.260638436067223 1.6]

[Input3]
Name='CYCA'
Range=[0 3.8]
NumMFs=3
MF1='L':'gaussmf',[0.251971489065406 0.00994]
MF2='M':'gaussmf',[0.637800228121051 1.52]
MF3='H':'gaussmf',[0.798272616947956 3.81]

[Output1]
Name='RB'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.109171593678819 3.17e-05]
MF2='M':'gaussmf',[0.109 0.368206349206349]
MF3='H':'gaussmf',[0.281759136920946 1.2]

[Rules]
1 1 1, 3 (1) : 1
2 1 1, 1 (1) : 1
3 1 1, 1 (1) : 1
3 3 1, 1 (1) : 1
0 0 2, 1 (1) : 1

[System]
Name='TFB'
Type='mamdani'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
```

```
DefuzzMethod='centroid'

[Input1]
Name='CYCA'
Range=[0 3.8]
NumMFs=3
MF1='L':'gaussmf',[0.8069 0]
MF2='M':'gaussmf',[0.8068 1.9]
MF3='H':'gaussmf',[0.8069 3.8]

[Output1]
Name='TFB'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.2123 6.939e-18]
MF2='M':'gaussmf',[0.2123 0.5]
MF3='H':'gaussmf',[0.2123 1]

[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1

[System]
Name='TFE_EX_B'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=7
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='RB'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.084932180028802 2.08e-17]
MF2='M':'gaussmf',[0.106 0.35868253968254]
MF3='H':'gaussmf',[0.2548 1.2]

[Input2]
Name='CYCA'
Range=[0 3.8]
NumMFs=3
MF1='L':'gaussmf',[0.379947927060064 0]
MF2='M':'gaussmf',[0.286 1.46772486772487]
MF3='H':'gaussmf',[0.8069 3.8]

[Input3]
Name='CYCB'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.104480062733844 6.94e-18]
MF2='M':'gaussmf',[0.107 0.415343915343915]
MF3='H':'gaussmf',[0.2123 1]
```

```
[Input4]
Name='P27'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.114591036546796 2.08e-17]
MF2='M':'gaussmf',[0.19 0.603507936507937]
MF3='H':'gaussmf',[0.2548 1.2]

[Output1]
Name='TFE'
Range=[0 2]
NumMFs=3
MF1='L':'gaussmf',[0.191 2.311e-17]
MF2='M':'gaussmf',[0.2317 0.7411]
MF3='H':'gaussmf',[0.3168 2]

[Rules]
3 1 1 3, 1 (1) : 1
2 1 1 2, 1 (1) : 1
1 1 1 1, 3 (1) : 1
1 3 1 1, 1 (1) : 1
2 0 2 1, 1 (1) : 1
1 2 1 1, 2 (1) : 1
0 0 2 0, 1 (1) : 1

[System]
Name='CYCA_optmized'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=6
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='TFE'
Range=[0 1.8]
NumMFs=3
MF1='L':'gaussmf',[0.0438358723682758 0.350818098616905]
MF2='M':'gaussmf',[0.999782444622033 0.00747231938859937]
MF3='H':'gaussmf',[0.053563457703846 0.269395528027957]

[Input2]
Name='CDC20A'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.00253885515902562 0.554169242737569]
MF2='M':'gaussmf',[0.994382706810766 0.746809360168231]
MF3='H':'gaussmf',[0.0846367469078191 0.999999869810367]

[Input3]
Name='P27'
Range=[0 1.2]
NumMFs=3
MF1='L':'gaussmf',[0.0674452811568626 0.211057148539961]
MF2='M':'gaussmf',[0.117969901647764 0.74283332551062]
```

```
MF3='H':'gaussmf',[0.998114974980433 0.34777897930689]

[Input4]
Name='CDH1'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.9999999985794 0.853622539699502]
MF2='M':'gaussmf',[0.999973886232124 0.810925393097 12]
MF3='H':'gaussmf',[0.999985070365868 0.490643917685763]

[Output1]
Name='CYCA'
Range=[0 5]
NumMFs=3
MF1='L':'gaussmf',[0.0213394062321226 1.51523342162408e-05]
MF2='M':'gaussmf',[0.984034505265254 0.999968077144573]
MF3='H':'gaussmf',[0.999999418503115 0.0400599322050034]

[Rules]
1 0 3 3, 1 (1) : 1
2 1 1 1, 3 (1) : 1
0 3 0 0, 1 (1) : 1
3 -1 -1 -1, 3 (1) : 1
1 0 2 2, 1 (1) : 1
0 0 2 2, 1 (1) : 1

[System]
Name='CYCD__optimized'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=1
NumRules=8
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='MYC'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.0186619326121608 0.148808855408691]
MF2='M':'gaussmf',[0.00348141977304808 0.999561778460741]
MF3='H':'gaussmf',[0.0023857876583823 7 0.909763903746813]

[Input2]
Name='SCF'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.0652101978693894 0.999959905626557]
MF2='M':'gaussmf',[0.981984393422502 0.0037329107268 0879]
MF3='H':'gaussmf',[0.238215901043482 0.645798957133665]

[Input3]
Name='CDC20A'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.172244711641824 0.367796475120247]
```

```
MF2='M':'gaussmf',[0.0737345483190596 0.467077442908999]
MF3='H':'gaussmf',[0.0213589405564776 0.128849308959119]

[Input4]
Name='CDC20B'
Range=[0 0.9]
NumMFs=3
MF1='L':'gaussmf',[0.468372103636873 0.004563282985519]
MF2='M':'gaussmf',[0.990099286578992 0.999997497254048]
MF3='H':'gaussmf',[0.991100519967061 0.0025826580508018]

[Output1]
Name='CYCD'
Range=[0 1.6]
NumMFs=3
MF1='L':'gaussmf',[0.00419871343604425 0.003097363364918]
MF2='M':'gaussmf',[0.231255190110146 0.000921316377010458]
MF3='H':'gaussmf',[0.167872327565564 0.999998433224413]

[Rules]
1 1 1 1, 1 (1) : 1
2 1 1 1, 2 (1) : 1
3 1 1 1, 3 (1) : 1
3 3 2 2, 2 (1) : 1
3 3 3 3, 1 (1) : 1
3 2 2 2, 1 (1) : 1
3 1 2 2, 1 (1) : 1
3 1 2 2, 1 (1) : 1

[System]
Name='SCF_optz'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=6
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='sum'
DefuzzMethod='centroid'

[Input1]
Name='CYCE'
Range=[0 1.6]
NumMFs=3
MF1='L':'gaussmf',[1 0.999999995583782]
MF2='M':'gaussmf',[0.580426508896776 0.986631710057269]
MF3='H':'gaussmf',[0.183986325940716 0.999818181039404]

[Input2]
Name='CDH1'
Range=[0 1]
NumMFs=3
MF1='L':'gaussmf',[0.0731739283401547 1e-05]
MF2='M':'gaussmf',[0.971524400094879 0.989680990010237]
MF3='H':'gaussmf',[0.0367643109907039 0.113092004577116]

[Output1]
Name='SCF'
```

```
Range=[0 1.5]
NumMFs=3
MF1='L':'gaussmf',[0.0160957093496636 0.0667510017410857]
MF2='M':'gaussmf',[0.378790253816283 0.999999984794546]
MF3='H':'gaussmf',[6.54019745187982e-05 0.330104315291503]

[Rules]
1 0, 1 (1) : 1
0 1, 2 (1) : 1
1 1, 1 (1) : 1
3 0, 3 (1) : 1
0 3, 1 (1) : 1
0 1, 3 (1) : 1
```

References

Adhikary, S., & Eilers, M. (2005). Transcriptional regulation and transformation by Myc proteins. *Nature Reviews Molecular Cell Biology, 6*(8), 635-645.

Aguda, B., & Tang, Y. (1999). The kinetic origins of the restriction point in the mammalian cell cycle. *Cell proliferation, 32*(5), 321-335.

Akkar, H. A., A-Amir, A. N., & Saleh, M. S. (2015). Tuning of Fuzzy Logic Controller for a DC Motor Based on Particle Swarm Optimization.

Alam, M. N. (2016). Particle Swarm Optimization: Algorithm and its Codes in MATLAB. *Department of Electrical Engineering, IIT, Roorkee*.

Alao, J. P. (2007). The regulation of cyclin D1 degradation: roles in cancer development and the potential for therapeutic invention. *Molecular cancer, 6*(1), 1.

Alberch, P. (1994). Kauffman, SA The origins of order. Self-organization and selection in evolution. Oxford University Press (1993). Price:£ 17.95 (pb),£ 51.00 (hb). ISBN: 0-19-505811-9 (hb) and 0-19-507951-5 (pb). *Journal of Evolutionary Biology, 7*(4), 518-519.

Albert, I., Thakar, J., Li, S., Zhang, R., & Albert, R. (2008). Boolean network simulations for life scientists. *Source code for biology and medicine, 3*(1), 1-8.

Albert, R., & Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *J Theor Biol, 223*(1), 1-18.

Alberts, B. (2008). *Molecular Biology of the Cell: Reference edition*: Garland Science.

Alberts, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., . . . Walter, P. (2013). *Essential cell biology*: Garland Science.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). Molecular Biology of the Cell Taylor & Francis. *New York*.

Alfieri, R., Barberis, M., Chiaradonna, F., Gaglio, D., Milanesi, L., Vanoni, M., . . . Alberghina, L. (2009). Towards a systems biology approach to mammalian cell cycle: modeling the entrance into S phase of quiescent fibroblasts after serum stimulation. *BMC bioinformatics, 10*(12), 1.

Ali Abroudi, S. S., Don Kulasiri. (2017). A comprehensive complex systems approach to the study and analysis of mammalian cell cycle control system in the presence of DNA damage stress. *ScienceDirect*.

Alm, E., & Arkin, A. P. (2003). Biological networks. *Current opinion in structural biology, 13*(2), 193-202.

Andries.P.E. (2002). Computational intelligence. *John Wiley & Sons, Ltd*.

Ang, X. L., & Harper, J. W. (2004). Interwoven ubiquitination oscillators and control of cell cycle transitions. *Sci STKE, 242*, pe31.

Asmuni, H. (2008). *Fuzzy methodologies for automated university timetabling solution construction and evaluation.* University of Nottingham.
Barrett, J. C., & Kawasaki, E. S. (2003). Microarrays: the use of oligonucleotides and cDNA for the analysis of gene expression. *Drug Discovery Today, 8*(3), 134-141.
Bartek, J., Bartkova, J., & Lukas, J. (1996). The retinoblastoma protein pathway and the restriction point. *Current opinion in cell biology, 8*(6), 805-814.
Bartek, J., & Lukas, J. (2001). Pathways governing G1/S transition and their response to DNA damage. *FEBS letters, 490*(3), 117-122.
Bean, J. M., Siggia, E. D., & Cross, F. R. (2006). Coherence and timing of cell cycle start examined at single-cell resolution. *Molecular cell, 21*(1), 3-14.
Behaegel, J., Comet, J.-P., Bernot, G., Cornillon, E., & Delaunay, F. (2015). A hybrid model of cell cycle in mammals. *Journal of bioinformatics and computational biology*, 1640001.
Behl, C., & Ziegler, C. (2014a). Cell Aging: Molecular Mechanisms and Implications for.
Behl, C., & Ziegler, C. (2014b). Cell Cycle: The Life Cycle of a Cell *Cell Aging: Molecular Mechanisms and Implications for Disease* (pp. 9-19): Springer.
Belacel, N., Čuperlović-Culf, M., Laflamme, M., & Ouellette, R. (2004). Fuzzy J-Means and VNS methods for clustering genes from microarray data. *Bioinformatics, 20*(11), 1690-1701.
Bell, S. P., & Dutta, A. (2002). DNA replication in eukaryotic cells. *Annual review of biochemistry, 71*(1), 333-374.
Besson, A., Dowdy, S. F., & Roberts, J. M. (2008). CDK inhibitors: cell cycle regulators and beyond. *Developmental cell, 14*(2), 159-169.
Braastad, C. D., Zaidi, S. K., Montecino, M., Lian, J. B., van Wijnen, A. J., Stein, J. L., & Stein, G. S. (2004). Architectural Organization of the Regulatory Machinery for Transcription, Replication, and Repair: Dynamic Temporal-Spatial Parameters of Cell Cycle Control. *Cell Cycle and Growth Control: Biomolecular Regulation and Cancer, Second Edition*, 15-92.
Braunewell, S., & Bornholdt, S. (2007). Superstability of the yeast cell-cycle dynamics: ensuring causality in the presence of biochemical stochasticity. *J Theor Biol, 245*(4), 638-643.
Bruce, Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2010). Molecular Biology of the Cell. New York: Garland Science; 2008. *Classic textbook now in its 5th Edition*.
Bruggeman, F. J., & Westerhoff, H. V. (2007). The nature of systems biology. *TRENDS in Microbiology, 15*(1), 45-50.
Büchsenschuß, J. (2014). Im Anfang war das Wort… *Auf der Suche nach der idealen Schildkröte* (pp. 7-30): Springer.
Calder, M., Gilmore, S., Hillston, J., & Vyshemirsky, V. (2010). Formal methods for biochemical signalling pathways *Formal Methods: State of the Art and New Directions* (pp. 185-215): Springer.
Chaves, M., Albert, R., & Sontag, E. D. (2005). Robustness and fragility of Boolean models for genetic regulatory networks. *J Theor Biol, 235*(3), 431-449.
Chen, C.-C., & Zhong, S. (2008). Inferring gene regulatory networks by thermodynamic modeling. *BMC genomics, 9*(Suppl 2), S19.
Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F. R., Novak, B., & Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell, 15*(8), 3841-3862.
Chiorino, G., & Lupi, M. (2002). Variability in the timing of G 1/S transition. *Mathematical Biosciences, 177*, 85-101.
Chu, I. M., Hengst, L., & Slingerland, J. M. (2008). The Cdk inhibitor p27 in human cancer: prognostic potential and relevance to anticancer therapy. *Nature Reviews Cancer, 8*(4), 253-267.
Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation, 6*(1), 58-73.

Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., & Magdalena, L. (2004). Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems, 141*(1), 5-31.
Csikász-Nagy, A. (2009). Computational systems biology of the cell cycle. *Briefings in Bioinformatics, 10*(4), 424-434.
Davidich, M. I., & Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PloS one, 3*(2), e1672.
De Backer, P., De Waele, D., & Van Speybroeck, L. (2010). Ins and outs of systems biology vis-à-vis molecular biology: continuation or clear cut? *Acta biotheoretica, 58*(1), 15-49.
De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology, 9*(1), 67-103.
Dubrova, E., & Teslenko, M. (2011). A SAT-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 8*(5), 1393-1399.
Düvel, K., Yecies, J. L., Menon, S., Raman, P., Lipovsky, A. I., Souza, A. L., . . . Cleaver, S. (2010). Activation of a metabolic gene regulatory network downstream of mTOR complex 1. *Molecular cell, 39*(2), 171-183.
Evans, T., Rosenthal, E. T., Youngblom, J., Distel, D., & Hunt, T. (1983). Cyclin: a protein specified by maternal mRNA in sea urchin eggs that is destroyed at each cleavage division. *cell, 33*(2), 389-396.
Fauré, A., Naldi, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics, 22*(14), e124-e131.
Ferrell, J. E., Tsai, T. Y.-C., & Yang, Q. (2011). Modeling the cell cycle: why do certain circuits oscillate? *cell, 144*(6), 874-885.
Florens, L., Washburn, M. P., Raine, J. D., Anthony, R. M., Grainger, M., Haynes, J. D., . . . Tabb, D. L. (2002). A proteomic view of the Plasmodium falciparum life cycle. *Nature, 419*(6906), 520-526.
Garg, A. (2009). *Implicit methods for modeling gene regulatory networks.* EPFL.
Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., & De Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics, 24*(17), 1917-1925.
Gérard, C., & Goldbeter, A. (2012). The cell cycle is a limit cycle. *Mathematical Modelling of Natural Phenomena, 7*(06), 126-166.
Gérard, C., & Goldbeter, A. (2012). Entrainment of the mammalian cell cycle by the circadian clock: modeling two coupled cellular rhythms. *PLoS Comput Biol, 8*(5), e1002516.
Gilbert, D. (1974). The nature of the cell cycle and the control of cell proliferation. *Biosystems, 5*(4), 197-206.
Gilbert, D., Fuß, H., Gu, X., Orton, R., Robinson, S., Vyshemirsky, V., . . . Dubitzky, W. (2006). Computational methodologies for modelling, analysis and simulation of signalling networks. *Briefings in Bioinformatics, 7*(4), 339-353.
Glass, L., & Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J Theor Biol, 39*(1), 103-129.
Hanahan, D., & Weinberg, R. A. (2000). The hallmarks of cancer. *cell, 100*(1), 57-70.
Harper, J. W., Burton, J. L., & Solomon, M. J. (2002). The anaphase-promoting complex: it's not just for mitosis any more. *Genes & development, 16*(17), 2179-2206.
Hartwell, L. H., Culotti, J., Pringle, J. R., & Reid, B. J. (1974). Genetic control of the cell division cycle in yeast. *science, 183*(4120), 46-51.
Hartwell, L. H., & Weinert, T. A. (1989). Checkpoints: controls that ensure the order of cell cycle events. *science, 246*(4930), 629-634.
Harvey, I., & Bossomaier, T. (1997). *Time out of joint: Attractors in asynchronous random boolean networks.* Paper presented at the Proceedings of the Fourth European Conference on Artificial Life.

Hay, E. D. (1991). Collagen and other matrix glycoproteins in embryogenesis *Cell biology of extracellular matrix* (pp. 419-462): Springer.
Helgason, C. M., & Jobe, T. H. (2003). Perception based reasoning and fuzzy cardinality provide direct measures of causality sensitive to initial conditions in the individual patient. *International Journal of Computational Cognition, 1*(2), 79-104.
Helgason, C. M., Malik, D., Cheng, S.-C., Jobe, T. H., & Mordeson, J. N. (2001). Statistical versus fuzzy measures of variable interaction in patients with stroke. *Neuroepidemiology, 20*(2), 77-84.
Helikar, T., & Rogers, J. A. (2009). ChemChains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists. *BMC systems biology, 3*(1), 58.
Helin, K. (1998). Regulation of cell proliferation by the E2F transcription factors. *Current opinion in genetics & development, 8*(1), 28-35.
Hilioti, Z., Chung, Y.-S., Mochizuki, Y., Hardy, C. F., & Cohen-Fix, O. (2001). The anaphase inhibitor Pds1 binds to the APC/C-associated protein Cdc20 in a destruction box-dependent manner. *Current Biology, 11*(17), 1347-1352.
Hopfensitz, M., Müssel, C., Maucher, M., & Kestler, H. A. (2013). Attractors in Boolean networks: a tutorial. *Computational Statistics, 28*(1), 19-36.
Huang, S. (1999). Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine, 77*(6), 469-480.
Ilsley, G. R., Luscombe, N. M., & Apweiler, R. (2009). Know your limits: assumptions, constraints and interpretation in systems biology. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics, 1794*(9), 1280-1287.
Irons, D. (2009). Logical analysis of the budding yeast cell cycle. *J Theor Biol, 257*(4), 543-559.
Iwamoto, K., Hamada, H., Eguchi, Y., & Okamoto, M. (2011). Mathematical modeling of cell cycle regulation in response to DNA damage: exploring mechanisms of cell-fate determination. *Biosystems, 103*(3), 384-391.
Jang, J.-S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics, 23*(3), 665-685.
JANG, J. S. R. S., C.T. (1995). Neuro-fuzzy modeling and control. *In The Proceeding*

of the IEEE, 83(3).
Kapuy, O., He, E., Uhlmann, F., & Novák, B. (2009). Mitotic exit in mammalian cells. *Molecular systems biology, 5*(1), 324.
Kar, S., Baumann, W. T., Paul, M. R., & Tyson, J. J. (2009). Exploring the roles of noise in the eukaryotic cell cycle. *Proceedings of the National Academy of Sciences, 106*(16), 6471-6476.
Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol, 22*(3), 437-467.
Kawai, T. K. a. S. (1989). An algorithm for drawing general undirected graphs. *Information*

Processing Letters, 31, (l), 7–15 (1989), Information

Processing Letters, 31, (l), 37–15 (1989).
Kenndy, J., & Eberhart, R. (1995). *Particle swarm optimization.* Paper presented at the Proceedings of IEEE international conference on neural networks.
Kitano, H. (2002). Computational systems biology. *Nature, 420*(6912), 206-210.
Knoblauch, M., Hibberd, J. M., Gray, J. C., & van Bel, A. J. (1999). A galinstan expansion femtosyringe for microinjection of eukaryotic organelles and prokaryotes. *Nature biotechnology, 17*(9), 906-909.
Kohn, K. W. (1998). Functional capabilities of molecular network components controlling the mammalian G1/S cell cycle phase transition. *Oncogene, 16*(8), 1065-1075.
Kong, S.-G., & Kosko, B. (1992). Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE transactions on Neural Networks, 3*(2), 211-223.

Koonin, E. V., & Wolf, Y. I. (2008). Genomics of bacteria and archaea: the emerging dynamic view of the prokaryotic world. *Nucleic acids research, 36*(21), 6688-6719.
Kremling, A., & Saez-Rodriguez, J. (2007). Systems biology—an engineering perspective. *Journal of biotechnology, 129*(2), 329-351.
Krumsiek, J., Pölsterl, S., Wittmann, D. M., & Theis, F. J. (2010). Odefy-from discrete to continuous models. *BMC bioinformatics, 11*(1), 233.
Kwok, N., Ha, Q., Nguyen, T., Li, J., & Samali, B. (2006). A novel hysteretic model for magnetorheological fluid dampers and parameter identification using particle swarm optimization. *Sensors and Actuators A: Physical, 132*(2), 441-451.
Lähdesmäki, H., Shmulevich, I., & Yli-Harja, O. (2003). On learning gene regulatory networks under the Boolean network model. *Machine learning, 52*(1-2), 147-167.
Lamartine, J. (2006). The benefits of DNA microarrays in fundamental and applied bio-medicine. *Materials Science and Engineering: C, 26*(2), 354-359.
Lee, T. I., & Young, R. A. (2000). Transcription of eukaryotic protein-coding genes. *Annual review of genetics, 34*(1), 77-137.
Li, F., Long, T., Lu, Y., Ouyang, Q., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America, 101*(14), 4781-4786.
Liang, S., Fuhrman, S., & Somogyi, R. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures.
Ling, H., Kulasiri, D., & Samarasinghe, S. (2010). Robustness of G1/S checkpoint pathways in cell cycle regulation based on probability of DNA-damaged cells passing as healthy cells. *Biosystems, 101*(3), 213-221.
Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies, 7*(1), 1-13.
Martirosyan, A., Figliuzzi, M., Marinari, E., & De Martino, A. (2016). Probing the limits to microRNA-mediated control of gene expression. *PLoS Comput Biol, 12*(1), e1004715.
Massagué, J. (2004). G1 cell-cycle control and cancer. *Nature, 432*(7015), 298-306.
MATHWORKS. (2012). Foundation of fuzzy logic, http://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html [18/02/2012].
MATHWORKS. (2015). Fuzzy Inference Process, https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html.
Matthias Bock, T. S., Chaitanya Talnikar, Edda Klipp. (2014). BooleSim: an interactive Boolean network simulator. *Bioinformatics, Volume 30*(Issue 1, 1), Pages 131–132.
Meyyappan, M., Atadja, P., & Riabowol, K. (1996). Regulation of gene expression and transcription factor binding activity during cellular aging. *Neurosignals, 5*(3), 130-138.
Morgan, D. O. (2007). *The cell cycle: principles of control*: New Science Press.
Morris, M. K., Melas, I., & Saez-Rodriguez, J. (2013). Construction of cell type-specific logic models of signaling networks using CellNOpt *Computational Toxicology* (pp. 179-214): Springer.
Mucientes, M., Moreno, D. L., Bugarín, A., & Barro, S. (2007). Design of a fuzzy controller in mobile robotics using genetic algorithms. *Applied Soft Computing, 7*(2), 540-546.
Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics, 26*(10), 1378-1380.
Negnevitsky. (2005). Artificial Intelligence: a guide to intelligent systems. *Pearson*

Education Limited, Essex, England.
Nikolov, D., & Burley, S. (1997). RNA polymerase II transcription initiation: a structural view. *Proceedings of the National Academy of Sciences, 94*(1), 15-22.
Noel, V., Grigoriev, D., Vakulenko, S., & Radulescu, O. (2012). Hybrid models of the cell cycle molecular machinery. *arXiv preprint arXiv:1208.3854*.
Noël, V., Vakulenko, S., & Radulescu, O. (2013). A hybrid mammalian cell cycle model. *arXiv preprint arXiv:1309.0870*.

Novák, B., Sible, J. C., & Tyson, J. J. (2003). Checkpoints in the cell cycle. *eLS*.
Novak, B., & Tyson, J. J. (1993). Numerical analysis of a comprehensive model of M-phase control in Xenopus oocyte extracts and intact embryos. *Journal of cell science, 106*(4), 1153-1168.
Novak, B., & Tyson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *J Theor Biol, 230*(4), 563-579. doi:10.1016/j.jtbi.2004.04.039
Novak, B., & Tyson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *J Theor Biol, 230*(4), 563-579.
Omizegba, E. E., & Adebayo, G. E. (2009). *Optimizing fuzzy membership functions using particle swarm algorithm.* Paper presented at the Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on.
OpenStax-CNX. (2013). Control of the Cell Cycle.
Orlando, D. A., Lin, C. Y., Bernard, A., Wang, J. Y., Socolar, J. E., Iversen, E. S., . . . Haase, S. B. (2008). Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature, 453*(7197), 944-947.
Papin, J. A., Hunter, T., Palsson, B. O., & Subramaniam, S. (2005). Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology, 6*(2), 99-111.
Pardee, A. (1989). G1 events and regulation of cell proliferation. *Science, 246*(4930), 603-608. doi:10.1126/science.2683075
Pardee, A. B. (1989). G1 events and regulation of cell proliferation. *science, 246*(4930), 603-608.
Pinheiro, D., & Sunkel, C. (2012). Mechanisms of cell cycle control.
Planas-Silva, M. D., & Weinberg, R. A. (1997). The restriction point and control of cell proliferation. *Current opinion in cell biology, 9*(6), 768-772.
Pratihar, D. K., Deb, K., & Ghosh, A. (1999). A genetic-fuzzy approach for mobile robot navigation among moving obstacles. *International Journal of Approximate Reasoning, 20*(2), 145-172.
Reinders, J., Lewandrowski, U., Moebius, J., Wagner, Y., & Sickmann, A. (2004). Challenges in mass spectrometry-based proteomics. *Proteomics, 4*(12), 3686-3703.
Roeder, R. G. (1996). The role of general initiation factors in transcription by RNA polymerase II. *Trends in biochemical sciences, 21*(9), 327-335.
Russo, A. A., Jeffrey, P. D., Patten, A. K., Massagué, J., & Pavletich, N. P. (1996). Crystal structure of the p27Kip1 cyclin-dependent-kinase inhibitor bound to the cyclin A-Cdk2 complex. *Nature, 382*(6589), 325-331.
Saadatpour, A., Albert, I., & Albert, R. (2010). Attractor analysis of asynchronous Boolean models of signal transduction networks. *J Theor Biol, 266*(4), 641-656.
Saadatpour, A., Wang, R.-S., Liao, A., Liu, X., Loughran, T. P., Albert, I., & Albert, R. (2011). Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. *PLoS Comput Biol, 7*(11), e1002267.
Sahin, Ö., Fröhlich, H., Löbke, C., Korf, U., Burmester, S., Majety, M., . . . Thieffry, D. (2009). Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC systems biology, 3*(1), 1.
Sel'kov, E. (1969). [2 alternative autooscillatory stationary states in thiol metabolism--2 alternative types of cell multiplication: normal and neoplastic]. *Biofizika, 15*(6), 1065-1073.
Shin, Y.-J., & Bleris, L. (2010). Linear control theory for gene network modeling. *PloS one, 5*(9), e12785.
Singhania, R., Sramkoski, R. M., Jacobberger, J. W., Tyson, J. J., & Beard, D. (2011). A hybrid model of mammalian cell cycle regulation. *PLoS Comput Biol, 7*(2), e1001077.
Srividhya, J., & Gopinathan, M. (2006). A simple time delay model for eukaryotic cell cycle. *J Theor Biol, 241*(3), 617-627.
Stein, G. S., & Pardee, A. B. (2004). *Cell cycle and growth control: Biomolecular regulation and cancer*: John Wiley & Sons.

Steuer, R. (2004). Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations. *J Theor Biol, 228*(3), 293-301.
Swain, M. T., Mandel, J. J., & Dubitzky, W. (2010). Comparative study of three commonly used continuous deterministic methods for modeling gene regulation networks. *BMC bioinformatics, 11*(1), 459.
Tashima, Y., Hamada, H., Okamoto, M., & Hanai, T. (2008). Prediction of key factor controlling G1/S phase in the mammalian cell cycle using system analysis. *Journal of bioscience and bioengineering, 106*(4), 368-374.
Thakar, J., Pilione, M., Kirimanjeswara, G., Harvill, E. T., & Albert, R. (2007). Modeling systems-level regulation of host immune responses. *PLoS Comput Biol, 3*(6), e109.
Thomas, R. (1973). Boolean formalisation of genetic control circuits. *J Theor Biol, 42*, 565–583.
Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *J Theor Biol, 153*(1), 1-23.
Tomida, S., Hanai, T., Honda, H., & Kobayashi, T. (2002). Analysis of expression profile using fuzzy adaptive resonance theory. *Bioinformatics, 18*(8), 1073-1083.
Torres, A., & Nieto, J. J. (2003). The fuzzy polynucleotide space: basic properties. *Bioinformatics, 19*(5), 587-592.
Torres, A., & Nieto, J. J. (2006). Fuzzy logic in medicine and bioinformatics. *BioMed Research International, 2006*.
Trimarchi, J. M., & Lees, J. A. (2002). Sibling rivalry in the E2F family. *Nature Reviews Molecular Cell Biology, 3*(1), 11-20.
Tyson, J. J., Novak, B., Chen, K., & Val, J. (1995). Checkpoints in the cell cycle from a modeler's perspective *Progress in cell cycle research* (pp. 1-8): Springer.
Vander Heiden, M. G., Cantley, L. C., & Thompson, C. B. (2009). Understanding the Warburg effect: the metabolic requirements of cell proliferation. *science, 324*(5930), 1029-1033.
Vellai, T., & Vida, G. (1999). The origin of eukaryotes: the difference between prokaryotic and eukaryotic cells. *Proceedings of the Royal Society of London B: Biological Sciences, 266*(1428), 1571-1577.
Vijesh, N., Chakrabarti, S. K., & Sreekumar, J. (2013). Modeling of gene regulatory networks: A review. *Journal of Biomedical Science and Engineering, 6*(02), 223.
Walworth, N. C. (2000). Cell-cycle checkpoint kinases: checking in on the cell cycle. *Current opinion in cell biology, 12*(6), 697-704.
Wang, R.-S., Saadatpour, A., & Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Physical biology, 9*(5), 055001.
Wawra, C., Kühl, M., & Kestler, H. A. (2007). Extended analyses of the Wnt/β-catenin pathway: Robustness and oscillatory behaviour. *FEBS letters, 581*(21), 4043-4048.
Weis, M. C., Avva, J., Jacobberger, J. W., & Sreenath, S. N. (2014). A data-driven, mathematical model of mammalian cell cycle regulation. *PloS one, 9*(5), e97130.
Wheaton, K., Atadja, P., & Riabowol, K. (1996). Regulation of transcription factor activity during cellular aging. *Biochemistry and cell biology, 74*(4), 523-534.
Wille, J. J., Pittelkow, M. R., Shipley, G. D., & Scott, R. E. (1984). Integrated control of growth and differentiation of normal human prokeratinocytes cultured in serum-free medium: Clonal analyses, growth kinetics, and cell cycle studies. *Journal of cellular physiology, 121*(1), 31-44.
Windhager, L. (2013). *Modeling of dynamic systems with Petri nets and fuzzy logic.* lmu.
Wittmann, D. M., Krumsiek, J., Saez-Rodriguez, J., Lauffenburger, D. A., Klamt, S., & Theis, F. J. (2009). Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC systems biology, 3*(1), 98.
Wolkenhauer, O., Kolch, W., & Cho, K.-H. (2004). Mathematical systems biology: Genomic cybernetics *Computation in Cells and Tissues* (pp. 305-325): Springer.

Wolkenhauer, O., Ullah, M., Kolch, W., & Cho, K.-H. (2004). Modeling and simulation of intracellular dynamics: choosing an appropriate framework. *NanoBioscience, IEEE Transactions on, 3*(3), 200-207.

Xie, F., Liu, T., Qian, W.-J., Petyuk, V. A., & Smith, R. D. (2011). Liquid chromatography-mass spectrometry-based quantitative proteomics. *Journal of Biological Chemistry, 286*(29), 25443-25449.

Xu, X., Wang, L., & Ding, D. (2004). Learning module networks from genome-wide location and expression data. *FEBS letters, 578*(3), 297-304.

Yao, G., Lee, T. J., Mori, S., Nevins, J. R., & You, L. (2008). A bistable Rb–E2F switch underlies the restriction point. *Nature cell biology, 10*(4), 476-482.

Zadeh, L. A. (1965). Fuzzy Sets. *information and contol, L 8, 338--353 (1965)* (L 8, 338--353 (1965)), L 8, 338--353 (1965)

Zadeh, L. A. (1975). The concept of a lingusitic variable and its applications to approximate

reasoning - Part I;II ; and III. *Information Sciences, (8); (8) ; (9),*, 199–249;

301–357; 143–180.

Zámborszky, J., Hong, C. I., & Nagy, A. C. (2007). Computational analysis of mammalian cell division gated by a circadian clock: quantized cell cycles and cell size control. *Journal of biological rhythms, 22*(6), 542-553.

Zetterberg, A., & Larsson, O. (1985). Kinetic analysis of regulatory events in G1 leading to proliferation or quiescence of Swiss 3T3 cells. *Proceedings of the National Academy of Sciences, 82*(16), 5365-5369.

Zhang, L., Cheng, Y., & Liew, K. (2013). A mathematical analysis of DNA damage induced G2 phase transition. *Applied Mathematics and Computation, 225*, 765-774.

Zhang, R., Shah, M. V., Yang, J., Nyland, S. B., Liu, X., Yun, J. K., . . . Loughran, T. P. (2008). Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences, 105*(42), 16308-16313.

Zhang, Y., Qian, M., Ouyang, Q., Deng, M., Li, F., & Tang, C. (2006). Stochastic model of yeast cell-cycle network. *Physica D: Nonlinear Phenomena, 219*(1), 35-39.

Zheng, J., Zhang, D., Przytycki, P. F., Zielinski, R., Capala, J., & Przytycka, T. M. (2010). SimBoolNet—a Cytoscape plugin for dynamic simulation of signaling networks. *Bioinformatics, 26*(1), 141-142.