IET Systems Biology

Research Paper Submitted in IET Systems Biology

IET Journals
The Institution of Engineering and Technology

ISSN 1751-8857
doi: 0000000000
www.ietdl.org

Formal Reasoning about Synthetic Biology using Higher-order-logic Theorem Proving

Sa'ed Abed[1*], *Adnan Rashid*[2], *Osman Hasan*[2]

[1] *Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Kuwait*
[2] *School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan*
* *E-mail: s.abed@ku.edu.kw*

Abstract: Synthetic Biology is an interdisciplinary field that uses well-established engineering principles, ranging from electrical, control and computer systems, for performing the analysis of the biological systems, such as biological circuits, pathways, controllers and enzymes. Controllers play a pivotal role in regulating different modules and components of a biological system to ensure its smooth functionality. Conventionally, the analysis of these biological systems, i.e., the genetic circuits and their associated controllers is performed using paper-and-pencil proofs and computer simulations methods. However, these methods cannot ensures accurate results due to their inherent limitations such as proneness to human error, approximations of results, round-off errors and the involvement of unverified algorithms in the core of the underlying tools, providing such analyses. We propose to use higher-order-logic theorem proving as a complementary technique for analyzing linear biological systems and thus overcome the above-mentioned issues. The proposed approach is primarily based on developing a mathematical model of the genetic circuits and the bio-controllers used in synthetic biology based on higher-order logic and analyzing it using deductive reasoning in an interactive theorem prover. The involvement of the logic, mathematics and the deductive reasoning in this method ensures the accuracy of the analysis. The main idea is to, first, model the continuous dynamics of the genetic circuits and their associated controllers using differential equations. These dynamics are generally obtained from the reaction-based models of these systems and thus requires the notion of reaction kinetics. The next step is to obtain the systems' transfer function from their corresponding block diagram representations. Finally, the transfer function based analysis of these differential equation based models is performed using the Laplace transform. The transfer function is further used for performing the stability analysis of these systems. To illustrate the practical utilization of our proposed framework, we formally analyze the genetic circuits of activated and repressed expressions and autoactivation of protein, and phase lag and lead controllers, which are widely used in cancer-cell identifiers and multi-input receptors for precise disease detection.

1 Introduction

Currently, several engineering principles [1] are being commonly applied to analyze biological systems [2], signaling pathways [3] (molecules set interacting to control different tasks of a cell) and biological circuits [4] (cell biological portion imitating the logical functionality executed in electrical circuits) etc. For instance, control systems laws [5] are used for performing the analysis of biological systems, i.e., genetic circuits and bio-controllers. The merger of such interdisciplinary arenas into synthetic biology [6, 7] permits designing and investigating these systems in an effective way.

Controllers in synthetic biology, generally called as bio-controllers [8], are considered as a major part of a biological system and are used to regulate the functionality of different modules and components of the underlying system. The feedback controllers incorporate an extra feedback from the output to improve the system's robustness and reduce its sensitivity to noise. They are frequently utilized in genetic circuits. Analyzing such systems needs modeling their dynamics, demonstrating the interaction of their different modules/components, using differential equations. These differential equations are generally obtained from the reaction-based models of these systems. This requires the notion of reaction kinetics and is mainly based on complex-valued vectors and matrices. Then, the transfer functions which provide the dynamics in the frequency domain of these systems are extracted from their block diagram representations [9], based on foundational control system principles. Finally, the Laplace transform is utilized for performing the transfer function based (the frequency domain) analysis of these systems based on their dynamical (differential equation based) models.

Conventionally, the analysis of these biological systems, i.e., biological circuits, pathways, networks and controllers, is performed using the paper-and-pencil proof [10] and computer based numerical [11] and symbolic [12] methods. However, the paper-and-pencil proof methods based analysis is error prone due to the highly involved human manipulation, especially when analyzing larger systems, and hence cannot guarantee an accurate analysis. Moreover, this kind of manual manipulation does not ensure that every assumption necessary for the mathematical analysis is explicitly identified in the analysis. Thus, there is always a chance of missing some vital assumptions in the final result of the analysis and a system design based on such a result may not lead to the same results as predicted in the analysis. In the same way, the computer-based numerical and symbolic techniques involve various tools, such as, MATLAB, Mathematica and Maple for analyzing the dynamics of these systems. The numerical methods based analysis includes approximation of the continuous values of the variables due to the finite precision of computer arithmetic and thus compromises the accuracy of the analysis. Furthermore, due to the availability of the limited computational resources and computer memory, it involves a finite iterations number to obtain the values of unknown continuous parameters and presents more inaccuracies in the analysis. Also, the symbolic tools have some unproven symbolic procedures in their core and consequently cannot ascertain complete correctness and accuracy of the analysis results.

For example, the simplification of the mathematical expression $\frac{x^2-1}{x-1}$ in a symbolic tool Mathematica yields $(x+1)$ without explicitly highlighting the case when $x = 1$, which returns an indeterminate form [13]. Therefore, the concerned safety-critical nature of biological systems such that these traditional approaches cannot be trusted for their accurate analysis.

Formal methods [14] are computer-based mathematical methods that are being extensively used for the accurate modeling, specification and verification of many complex real-world systems [15–17]. They are classified into two categories, i.e., model checking [18] and higher-order-logic theorem proving [19]. Model checking is based on constructing a state-space model of the underlying system and verifying its intended temporal behavior by rigorous state-space exploration. It has been widely used in the synthetic biology field for formally investigating the biological circuits and their related feedback controllers. Yordanov et al. [20] proposed a novel approach for formally verifying the synthetic genetic circuits. The authors developed a state-space model of the overall system by constructing the individual components based on the experimentally obtained characterization data and used it for formally analyzing a genetic inverter and a genetic NOR gate. Similarly, Bartocci et al. [21] provided an approach for designing the synthetic biological circuits by specifying their behavior in terms of signal temporal logic (STL). Moreover, the authors utilized their proposed framework for synthesizing a half adder. Madsen et al. [22] used statistical model checking for formally analyzing genetic circuits. In particular, they proposed a framework, which converts a genetic circuit model into its stochastic model, i.e., continuous-time Markov chain, which is further used to conduct the transient Markov chain analysis of these circuits. Finally, they utilized their proposed framework to formally analyze a genetic toggle switch. All the model checking based analyses, presented above, suffer from the inherent state-space explosion problem [23] and hence are not appropriate for analyzing larger systems.

Higher-order-logic theorem proving includes developing a mathematical model of the system based on higher-order logic and verifying the interested properties using deductive reasoning inside the sound core of a theorem prover. The involvement of a formal model (specified in highly expressive higher-order logic) and its related formally specified properties, along with the correct nature of theorem proving, confirms the soundness and completeness of the analysis. It has also been used to formal reason about the biological systems and molecular pathways. The authors in [24, 25] offered a formalization of Zsyntax using HOL theorem prover. Also, they used their proposed platform to formally reason and analyze the TP53 degradation pathway and Glycolytic leading from D-Glucose to Fructose-1,6-bisphosphate. Recently, Abed et al. [26] performed the Laplace transform based analysis of the genetic circuits, i.e., the activated and the repressed expressions of protein using the HOL Light theorem prover. However, their proposed approach does not cater for the reaction kinetic based dynamical models of the biological circuits and bio-controllers. Similarly, Rashid et al. [15] proposed a platform using HOL Light, which offers the formal support for the reaction kinetic based dynamical analysis of the biological systems. However, their proposed platform considers the reaction-based models of the real-valued species (reactant and products).

In this work, the higher-order-logic theorem proving [19] will be used to formally reason about the genetic circuits and bio-controllers used in synthetic biology, incorporating their reaction kinetic based dynamical models for the complex-valued species (reactants and products) as shown in Figure 1. The first step is to model the continuous dynamics of the genetic circuits and their associated controllers using differential equations. These continuous dynamics are generally obtained from reaction-based model of these systems. This requires the notion of the reaction kinetics and a transformation of the reaction-based model to its corresponding differential equation model. The transfer function of these systems that can be obtained from their corresponding block diagram representations is needed in the next step. Finally, the Laplace transform is utilized for performing the transfer function based analysis of the differential equation based models of these circuits, which is further used for the stability analysis of these systems. To demonstrate the practical efficiency and usefulness of our proposed framework, we formally analyze the genetic circuits of activated and repressed expressions and autoactivation of protein, and phase lag and lead controllers using the HOL Light theorem prover.

The main contributions of the paper are:

- Formalization of the reaction kinetics involving formal models of the biological pathways and reactions for the complex-valued biological entities (reactants and products)
- Formalization of the complex-valued matrices that are required for the formalization of the reaction kinetics
- Formalization of the transformation from a biological reaction to the continuous dynamics (differential equations based models) of systems
- Formalization of the stability of the biological circuits and controllers using the HOL Light theorem prover
- Formal verification of the genetic circuits of activated and repressed expressions and autoactivation of protein, and phase lag and lead controllers using HOL Light

2 Preliminaries

We provide a brief introduction to the HOL Light theorem prover, its theories of multivariate calculus and the Laplace transform and the reaction kinetic based modeling of the biological systems in this section.

2.1 Theorem Proving and HOL Light Theorem Prover

Theorem proving is a broadly adopted formal verification technique that includes building the proofs of the mathematical theorems using a computer based program (called *theorem prover*) [27]. Theorem proving systems have been usually applied to formally reason and analyze the properties of both hardware and software systems [16, 17]. Based on the expressiveness needing, these properties are expressed as theorems using propositional, first-order or higher-order logic, i.e. the higher-order logic offers more expressiveness by permitting extra quantifiers. Furthermore, it is appropriate for conducting the mathematical analysis based on theories of multivariate calculus and the Laplace transform.

HOL Light [28], which is a widely used interactive theorem prover, is utilized for developing and building formal proofs of mathematical concepts represented as theorems. It is implemented in Objective CAML (OCaml), a functional programming language, with the purpose of automating the mathematical proofs [29]. It has a very small logical kernel containing the OCaml code of approximately 400 lines and supports terms, types, axioms, inference rules and theorems, which can be verified using the built-in axioms, inference rules or already existing theorems.

2.2 Multivariable Calculus and Laplace Transform Theories

HOL Light provides a wide support to analyze systems using theories of multivariable calculus and Laplace transform. Some definitions from HOL Light's theory of the Laplace transform is given in Table 1. The definitions include the Laplace transform, existence of the Laplace transform and the exponential-order condition. For more details, readers are advised to refer to [30, 31]. The proposed formal analysis of the biological circuits and bio-controllers is primarily based upon this formalization of Laplace transform.

2.3 Reaction Kinetics Based Models

Reaction kinetics [32] involves the interaction of the biological processes with each other and the effect of reactions, exhibiting different rates, on the corresponding processes. The rate of a reaction captures the evolution of the concentration of the species (reactants and products) over time. A process is a chain of reactions, commonly known as pathway, and the study of the rate of these reactions provides the rate of the overall process. Generally, biological reactions are categorized as irreversible or reversible.

Analyzing a biological process requires its kinetic reaction based model, which consists of a set of k species (reactants and products), $S = \{S_1, S_2, S_3, \ldots, S_k\}$ and a set of l reactions, $Re = \{Re_1, Re_2, Re_3, \ldots, Re_n\}$.

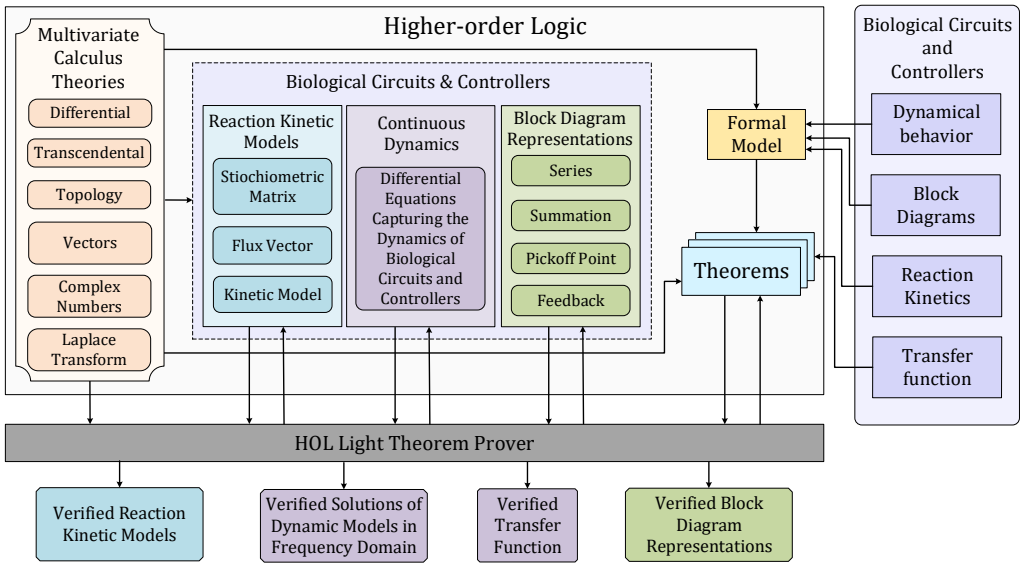

Fig. 1: Proposed Framework

Table 1 Laplace Transform

Mathematical Form	Formalized Form
Laplace Transform	
$\mathcal{L}[f(t)] = F(s) = \int_0^\infty f(t)e^{-st}dt,\ s \in \mathbb{C}$	$\vdash_{def}$ ∀s f. laplace_transform f s = integral {t | &0 ≤ drop t} (λt. cexp (−(s ∗ Cx t)) ∗ f t)
Laplace Existence	
f is piecewise smooth and is of exponential order on the positive real line	$\vdash_{def}$ ∀s f. existence_of_laplace f s ⇔ (∀b. f piecewise_differentiable_on interval [&0,b]) ∧ (∃M a. Re s > a ∧ expon_order_condit f M a)
Exponential-order Condition	
There exist a constant a and a positive constant M such that $\|f(t)\| \leq Me^{at}$	$\vdash_{def}$ ∀f M a. expon_order_condit f M a ⇔ &0 < M ∧ (∀t. &0 ≤ t ⇒ ||f (lift t)|| ≤ M ∗ exp (a ∗ t))

An irreversible reaction $Re_j, \{1 \leq j \leq l\}$ is generally described as follows:

$$Re_j : c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{k,j}S_k \xrightarrow{k_j} \acute{c}_{1,j}S_1 + \acute{c}_{2,j}S_2 + \ldots + \acute{c}_{k,j}S_k \quad (1)$$

Similarly, a reversible reaction $Re_j, \{1 \leq j \leq l\}$ can be written as follows:

$$Re_j : c_{1,j}S_1 + c_{2,j}S_2 + \ldots + c_{k,j}S_k \underset{k_j{}^r}{\overset{k_j{}^f}{\rightleftarrows}} \acute{c}_{1,j}S_1 + \acute{c}_{2,j}S_2 + \ldots + \acute{c}_{k,j}S_k \quad (2)$$

The coefficients $c_{1,j}, c_{2,j}, \ldots, c_{k,j}, \acute{c}_{1,j}, \acute{c}_{2,j}, \ldots, \acute{c}_{k,j}$ are the non-negative integers providing the stoichiometries of the species (reactants and products) taking part in the reaction. The non-negative integers k_j, and $k_j{}^f$ and $k_j{}^r$ are the kinetic rate constants of the irreversible, and the forward and reverse kinetic rate constants of the reversible reactions, respectively [33]. A reversible reaction can be divided into two irreversible reactions with the forward and reverse kinetic rate constants representing the kinetic rate constants of the first and the second reaction, respectively.

The dynamics of the biological systems are generally expressed by a set of ordinary differential equations as a counterpart to the above-mentioned reaction-based models. The development of the system is defined by analyzing the concentration change of the species, i.e., their time derivatives which can be mathematically represented as follows:

$$\frac{d[S_i]}{dt} = \sum_{j=1}^{n} s_{i,j}v_j \quad (3)$$

where $s_{i,j}$ represents the stoichiometric coefficient of the species S_i in reaction R_j (i.e., $s_{i,j} = \acute{c}_{i,j} - c_{i,j}$). The parameter v_j represents the flux of the reaction R_j.

For all the reactions from 1 to l, it becomes a vector as $v = (v_1, v_2, \ldots, v_l)^T$ and the corresponding system of ordinary differential equations can be written in the vectorial form as follows:

$$\frac{d[\mathbf{S}]}{dt} = N\mathbf{v} \quad (4)$$

where $[\mathbf{S}] = (S_1, S_2, \ldots, S_n)^T$ is vector of the concentration of all of the species taking part in the reaction and N is the stoichiometric matrix of order $k \times l$.

3 Related Work

Feedback controllers have been widely designed and implemented in the area of synthetic biology such as genetic circuits and molecular pathways. Chen et al. [34] designed molecular control circuits using DNA-based technology, which are widely used for integrated sensing, computation and actuation purposes. Their design is based on implementing several building-block reaction types and combining them into a network realizable at the molecular level. However, the controllers associated with these circuits do not incorporate the feedback for the underlying system. To cater for this issue, Yordanov et al. [35] proposed a computational framework for designing feedback control circuits constructed from nucleic acids. This is based on expressing the control and signal processing circuits as the

reaction based models and development of the controllers such as proportional-integrator (PI) controller. Moreover, it allows the analysis of various nucleic acid circuits such as (deoxyribonucleic acid) DNA and (ribonucleic acid) RNA, using the visual DNA Strand Displacement (DSD) tool, which is widely used for analyzing the computational devices implemented using DNA strand displacement. Similarly, Rosenfeld et al. [36] incorporated the negative feedback for analyzing the synthetic gene circuits. In particular, the authors studied the important characteristics of the negative autoregulatory circuit, such as rise-time and stability. Their analysis revealed the shorter rise-time and increased stability of the gene expression. Gardne et al. [37] proposed a genetic toggle switch, which is a synthetic bistable gene-regulatory network and is widely used in biocomputing and gene therapy. The authors conducted an analysis for predicting the conditions that are necessary for bistability of the corresponding network. The controllers considered in all the above-mentioned works are design specific and their size is also restricted to only a few genes. Recently, Harris et al. [8] designed genetic circuits based on the feedback control and its associated biological phase lag controller of generic nature. Moreover, the authors considered the reaction based models of these circuits and conducted the frequency domain analysis of the corresponding systems using Laplace transform. All the research contributions, presented above, use the traditional analysis techniques and their associated tools such as visual DSD and Matlab. Thus, these analyses suffer from their inherent limitations such as approximation, round off errors, limited computational resources and the unverified numerical algorithms present in the core of these tools, and thus cannot ensure the accurate analysis of the genetic circuits and their associated controllers.

Formal methods have been used as a complementary approach for analyzing the biological systems. In particular, model checking has been widely used in the synthetic biology field to formally analyze the biological circuits and their associated feedback controllers. Yordanov et al. [20] proposed a novel approach for formally verifying the synthetic genetic circuits. The authors developed a state-space model of the overall system by constructing the individual components based on the experimentally obtained characterization data. Moreover, to demonstrate the practical utilization of their proposed approach, they formally analyzed a genetic inverter and a genetic NOR gate using their intended properties specified in linear temporal logic (LTL). Similarly, Bartocci et al. [21] provided an approach for designing the synthetic biological circuits by specifying their behavior in terms of signal temporal logic (STL). Moreover, the authors utilized their proposed framework for synthesizing a half adder. Madsen et al. [22] used statistical model checking for formally analyzing genetic circuits. In particular, they proposed a framework, which converts a genetic circuit model into its stochastic model, i.e., continuous-time Markov chain, which is further used to conduct the transient Markov chain analysis of these circuits. Finally, they utilized their proposed framework to formally analyze a genetic toggle switch. All the model checking based analyses, presented above, suffer from the state-space explosion problem and therefore are not suitable for handling larger systems.

Higher-order logic theorem proving has also been used to formal reason about the biological systems and molecular pathways. Ahmad el at. [24, 25] proposed a formalization of Zsyntax using HOL theorem prover. Moreover, they utilized their proposed framework for formally analyzing the TP53 degradation pathway and Glycolytic leading from D-Glucose to Fructose-1,6-bisphosphate. Recently, Abed et al. [26] performed the Laplace transform based analysis of the genetic circuits, i.e., the activated and the repressed expressions of protein using the HOL Light theorem prover. However, their proposed approach does not cater for the reaction kinetic based dynamical models of the biological circuits and bio-controllers. Similarly,the authors in [15] built a framework, using HOL Light, to provide the formal support for the reaction kinetic based dynamical analysis of the biological systems. The developed framework considers the reaction-based models of the real-valued species (reactant and products). Whereas, for performing the Laplace transform based analysis of the genetic circuits and their associated bio-controllers, we need to formalize the species, contributing to their dynamical behaviour, as complex-valued functions. This requires formalizing the notion of reaction kinetics for the complex-valued reactants and products, which is the main scope of this paper.

4 Formalization of Reaction Kinetics for Complex-valued Species

In this section, we present the higher-order-logic formalization of the reaction kinetics for the complex-valued reactants and products. In reaction kinetic based modeling, a biological system (a pathway or a network) is modeled by a set of biological reactions, which are generally categorised as of two types, such as reversible and irreversible. We formally model this fact by using the inductive enumerating data-type feature of HOL Light as follows:

Definition 1.
```
define_type "type_of_reaction =
                irreversible | reversible"
```

A generic model of the biological reaction consists of lists of reactants and products, and its corresponding kinetic rate constants such as forward, and forward and reverse, for the case of irreversible and reversible reactions, respectively. In HOL Light, we can use the available types (e.g., Real ($\mathbb{R}$), Complex ($\mathbb{C}$)) to abbreviate new types. Therefore, we use the feature of type abbreviation in HOL Light to define new types for a biological reaction as follows:

Definition 2.
```
new_type_abbrev ("reactant",':((ℂ × ℕ))')
new_type_abbrev ("product",':((ℂ × ℕ))')
new_type_abbrev ("kinetic_rates",':((ℂ × ℂ))')
new_type_abbrev ("biological_reaction",':(type_of_reaction ×
        ((reactant) list × (product) list × kinetic_rates)')
new_type_abbrev ("biological_reactions",':
                        (biological_reaction) list')
```

The type reactant is a pair, capturing a single reactant of the biological reaction in such a way that its first element represents the concentration of the complex-valued reactant, whereas its second element models its stoichiometry. Similarly, the type kinetic_rates models the kinetic rate constants of the biological reaction. For an irreversible reaction, the first element of the pair captures its kinetic rate constant and the second element is zero. Whereas, for a reversible reaction, the pair models the forward and reverse kinetic rate constants, respectively.

To obtain a dynamical model of a biological system from its reaction based model, we require finding out its flux vector and the stoichiometric matrix. We formalize the flux of a single reaction, based on the *law of mass action* in HOL Light as follows:

Definition 3.
```
⊢def ∀t Rt Pd k.
    flux_single_react ((t,Rt,Pd,k):biological_reaction) =
      if t = irreversible then flux_irrever Rt Pd k
                          else flux_rever Rt Pd k
```

where flux_irrever and flux_rever provide the flux of the irreversible and reversible reactions, respectively, which are chosen using the conditional expression on the type of the reaction and are formalized as follows:

Definition 4.
```
⊢def ∀t Rt Pd k.
    flux_irrever ((t,Rt,Pd,k):biological_reaction) =
                                    FST k ∗ mk_flux Rt
⊢def ∀t Rt Pd k. flux_rever ((t,Rt,Pd,k):biological_reaction) =
                FST k ∗ mk_flux Rt - SND k ∗ mk_flux Pd
```

where FST and SND in the above definition accept the rate constants of the reaction as a pair ($\mathbb{C}$,$\mathbb{C}$) and return its first and second element, respectively. Similarly, mk_flux accepts a list of reactants R or products P and transforms the concentrations in to their exponent form according to the law of mass action, where the concentration of a molecule is implemented as the base while the corresponding stoichiometry as the exponent [38].

Table 2 Classical Properties of the Complex-valued Matrices

Name	Properties
Commutativity of matrix addition	$\vdash_{thm}$ ∀X Y. X + Y = Y + X
Associativity of matrix addition	$\vdash_{thm}$ ∀X Y Z. X + (Y + Z) = (X + Y) + Z
Associativity of matrix scalar multiplication	$\vdash_{thm}$ ∀a b X. a %%% (b %%% X) = (a ∗ b) %%% X
Associativity of matrix vector multiplication	$\vdash_{thm}$ ∀X Y x. X ∗∗ Y ∗∗ x = (X ∗∗ Y) ∗∗ x
Left Distributivity of matrix scalar multiplication	$\vdash_{thm}$ ∀X b c. (b ± c) %%% X = b %%% X ± c %%% X
Right Distributivity of matrix scalar multiplication	$\vdash_{thm}$ ∀X Y c. c %%% (X ± Y) = c %%% X ± c %%% Y
Left Additive identity	$\vdash_{thm}$ ∀X. cmat 0 + X = X
Right Additive identity	$\vdash_{thm}$ ∀X. X + cmat 0 = X
Left Multiplicative identity	$\vdash_{thm}$ ∀X. cmat 1 ∗ X = X
Right Multiplicative identity	$\vdash_{thm}$ ∀X. X ∗ cmat 1 = X
Left Multiplicative of zero	$\vdash_{thm}$ ∀X. cmat 0 ∗ X = cmat 0
Right Multiplicative of zero	$\vdash_{thm}$ ∀X. X ∗ cmat 0 = cmat 0

Generally, a biological pathway is composed of a chain of reversible and irreversible reactions in an arbitrary sequence. We model the flux vector for a chain of biological reactions, using the flux of a single reaction (flux_sing_reaction), as follows:

Definition 5. $\vdash_{def}$ ∀BL.
flux_vector BL = vector (MAP flux_sing_reaction BL)

The function flux_vector takes a list of reactions (biological_reactions, Definition 2) and returns a vector ($\mathbb{R}^k$) containing the fluxes of all the reactions, where k captures the length of the list BL. Here, we use the HOL Light function MAP for mapping the flux of a single reaction, i.e., flux_sing_reaction on every element of the list BL.

Next, in order to formalize the stoichiometric matrix, we require the notion of the complex-valued matrices, which are formalized as part of our proposed framework and some of their verified properties are given in Table 2. The HOL Light operator %%% presents the multiplication of a complex number with a complex-valued matrix. Similarly, cmat 0 provides the complex-valued zero matrix of order $M \times N$. More details about the formalization of the complex-valued matrices can be found at [38].

Now, we formalize the notion of stoichiometric matrix, which is a collection of column vectors and is given as follows:

Definition 6. $\vdash_{def}$ ∀BL. stiochio_mat (BL:biological_reactions)
= ctransp (vector (MAP stiochio_mat_single_react BL))

The function stiochio_mat accepts a list of reactions (biological_reactions, Definition 2) and returns a matrix having m rows and k columns, where m and k represent the number of reactions and the species (reactant and products), respectively. The HOL Light function ctransp models the transpose of the complex-valued matrix. Similarly, stiochio_mat_single_react provides a vector ($\mathbb{R}^k$) corresponding to the column of the stoichiometric matrix. Moreover, the function MAP is used to apply it on every element of the list BL. More details about the formalization of the stoichiometric matrix can be found at [38].

Finally, in order to present the derivatives of the concentrations of the reactant and product in matrix form (left side of Equation (4)), we define concen_vector_deriv as follows:

Definition 7. $\vdash_{def}$ ∀BL t. concen_vector_deriv BL t =
vector (mk_concen_vect_deriv BL t)

where mk_concen_vect_deriv accepts a complex-valued list BL containing the concentrations of all the species (reactants and products) taking part in the reaction. It uses a HOL Light function vector_derivative, which represents the vector-valued derivative of a function and maps it to every element of the list BL.

We can utilize the formalization of the flux vector and stoichiometric matrix to develop a reaction kinetic model of an arbitrary biological system (pathway or network) consisting of any number of reactions and species. For example, a biological pathway consisting of a list of S biological species and R biological reactions can be represented by the following generic reaction kinetic model:

((concen_vector_deriv S t):$\mathbb{C}^M$) =
((stoichio_mat R):$\mathbb{C}^{N\,M}$) ∗∗ flux R

For illustration purposes, we apply our formalization of reaction kinetics, for the complex-valued biological entities (reactants and products), to develop reaction kinetic model and an equivalent differential equations based model of a simple metabolic pathway [39], as shown in Figure 2.

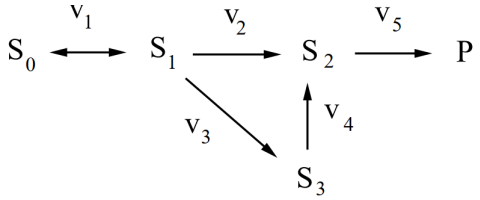

Fig. 2: Simple Metabolic Pathway [39]

The concentrations of the substrate and product are represented as S_0 and P, respectively. Whereas, S_1, S_2 and S_3 are the intermediate productions. All reactions except the production of S_1 are irreversible reactions. Moreover, each reaction is catalysed by an enzyme E_i with concentration e_i. The reaction rates are given as follows:

$$\begin{aligned} v_1 &= e_1([S_0] - s_1), \\ v_2 &= e_2 s_1, \\ v_3 &= e_3 s_1, \\ v_4 &= e_4 s_3, \\ v_5 &= e_5 s_2 \end{aligned}$$

The corresponding dynamics (differential equation model) is mathematically expressed as follows [39]:

$$\frac{d[S_1]}{dt} = e_1([S_0] - [S_1]) - (e_2 + e_3)[S_1] \tag{5}$$

$$\frac{d[S_2]}{dt} = e_2[S_1] + e_4[S_3] - e_5[S_2] \tag{6}$$

$$\frac{d[S_3]}{dt} = e_3[S_1] - e_4[S_3] \tag{7}$$

$$\tag{8}$$

We first formalize the reaction-based model of the simple metabolic pathway, using Definition 2, in HOL Light as follows:

Definition 8. $\vdash_{def}$ ∀S0 S1 S2 S3 P e1 e2 e3 e4 e5 t.
react_model_simp_meta_path
S0 S1 S2 S3 P e1 e2 e3 e4 e5 t =
[reversible,[S0 t, 1; S1 t, 0],

```
                    [S0 t, 0; S1 t, 1], e1, Cx (&1);
[irreversible,[S1 t, 1; S2 t, 0],
                    [S1 t, 0; S2 t, 1], e2, Cx (&0);
[irreversible,[S1 t, 1; S3 t, 0],
                    [S1 t, 0; S3 t, 1], e3, Cx (&0);
[irreversible,[S3 t, 1; S2 t, 0],
                    [S3 t, 0; S2 t, 1], e4, Cx (&0);
[irreversible,[S2 t, 1; P t, 0],
                    [S2 t, 0; P t, 1], e5, Cx (&0) ]
```

Next, we formally verify the transformation of the reaction-based model to its corresponding dynamical model in HOL Light as follow:

Theorem 1. $\vdash_{thm}$ ∀S0 S1 S2 S3 P e1 e2 e3 e4 e5 t.
(concen_vector_deriv [S1; S2; S3] t =
stoichio_matrix
(react_model_simp_meta_path
S0 S1 S2 S3 P e1 e2 e3 e4 e5 t) ∗∗
flux_vector
(react_model_simp_meta_path
S0 S1 S2 S3 P e1 e2 e3 e4 e5 t)) ⟺
(vector_derivative S1 (at t) =
e1 ∗ (S0 t - S1 t) - (e2 + e3) ∗ S1 t ∧
vector_derivative S2 (at t) =
e2 ∗ S1 t + e4 ∗ S3 t - e5 ∗ S2 t ∧
vector_derivative S3 (at t) = e3 ∗ S1 t - e4 ∗ S3 t)

The proof of the above theorem is based on the formal definitions of the stoichiometric matrix (stoichio_matrix, Definition 6), flux vector (flux_vector, Definitions 3-5) and derivatives of the concentration of species (concen_vector_deriv, Definition 7), properties of vectors and matrices alongwith some complex arithmetic reasoning.

We further use our formalization of reaction kinetics for formally analyzing the gene expression regulation as will be described later.

5 Formalization of Block Diagram Representations of Biological Circuits

In this section, we present our formalization of block diagram representations of the biological circuits. Mainly, the section provides the formal definitions corresponding to the fundamental building blocks (subsystems) of block diagram representations. These definitions enable us to formally model the block diagrams of a generic biological circuit in the s-domain and to find out the transfer function of any biological circuit from its block diagram representation. The presented formalization is basically inspired from the block diagrams of the control systems [40].

Configuration 1: Series Representation The transfer function of a sum of two components (subsystems) of a biological circuit, which can be any proteins or genes, in the case of a genetic circuit, is equal to the product of the transfer function of the individual components as depicted in Figure 3. We formalize this configuration for an arbitrary (N) number of components of a circuit as follows:

Definition 9. $\vdash_{def}$ ∀A_i. series_comp [A_1; A_2; ...; A_N] = $\prod_{i=1}^{N} A_i$

The function series_comp accepts the transfer functions of individual components of the circuit as a list of complex numbers and returns the transfer function of the overall circuit as a product of all individual transfer functions.

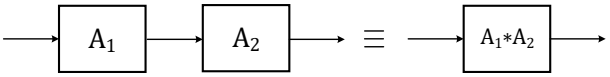

Fig. 3: Series Representation

Configuration 2: Summation Junction The summation junction for various components of a biological circuit is an addition module that provides the summation of the transfer functions of individual components as depicted in Figure 4. For an arbitrary number (N) of components of a circuit, having transfer functions represented by a list of complex numbers, we formalize this configuration as follows:

Definition 10. $\vdash_{def}$ ∀A_i. summ_jun [A_1; A_2; ...; A_N] = $\sum_{i=1}^{N} A_i$

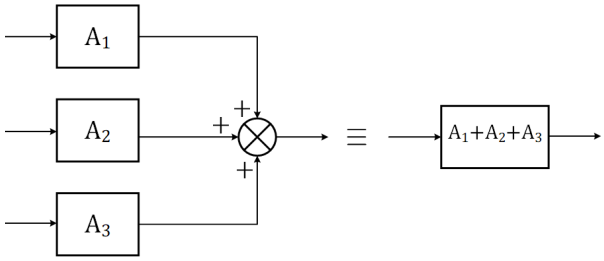

Fig. 4: Summation Junction

Configuration 3: Pickoff Point The pickoff point configuration represents the mapping of a component of a biological circuit to a parallel branch of components as shown in Figure 5. We model this configuration in HOL Light as follows:

Definition 11. $\vdash_{def}$ ∀α A_i. pick_point [A_1; A_2; ...; A_N] = [$\alpha * A_1$; $\alpha * A_2$; ...; $\alpha * A_N$]

The function pick_point takes the transfer function of the first component as a complex number and the transfer functions of the other components in parallel as a list of complex numbers, and returns the corresponding transfer functions corresponding to the equivalent block diagram representation as a list of complex numbers.

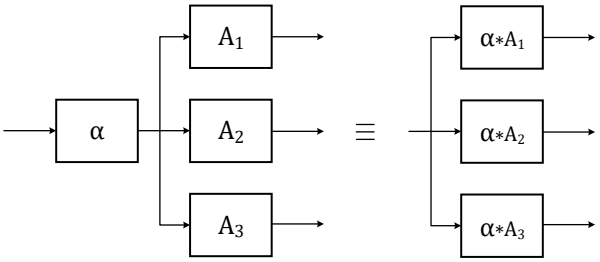

Fig. 5: Pickoff Point

Configuration 4: Feedback Block The feedback block configuration is the fundamental representation for modeling the closed loop controllers for the biological circuits as shown in Figure 6. Due to the presence of the feedback signal, it is primarily represented by an infinite summation of branches that consists of serially connected components, as shown in Figure 6. We formalize the transfer function of each branch as follows:

Definition 12. $\vdash_{def}$ ∀α β n. branch_tf α β n = $\prod_{i=0}^{n}$ series_comp [α; β]

The function branch_tf takes the forward path transfer function α (a protein or gene), the feedback path (feedback signal) transfer function β and the number of the branch (n), and returns a complex number representing the transfer function of the n^{th} branch.

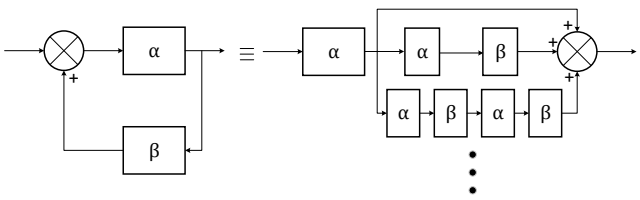

Fig. 6: Feedback Block

Now, we formalize the feedback block representation using our function branch_tf as follows:

Definition 13. $\vdash_{def} \forall \alpha\ \beta.$ feedback_block $\alpha\ \beta$ =
series_comp $[\alpha; \sum_{k=0}^{\infty}$ branch $\alpha\ \beta$ k$]$

The function feedback_block takes the transfer function of forward path (α) and the transfer function of feedback path (β) and returns the transfer function by forming the series network of the final forward path transfer function and the summation of all the possible infinite branches.

For illustration, we now apply our formalization of the block diagram representations to formally model and analyze a tunable synthetic gene oscillator [41], as shown in Figure 7.

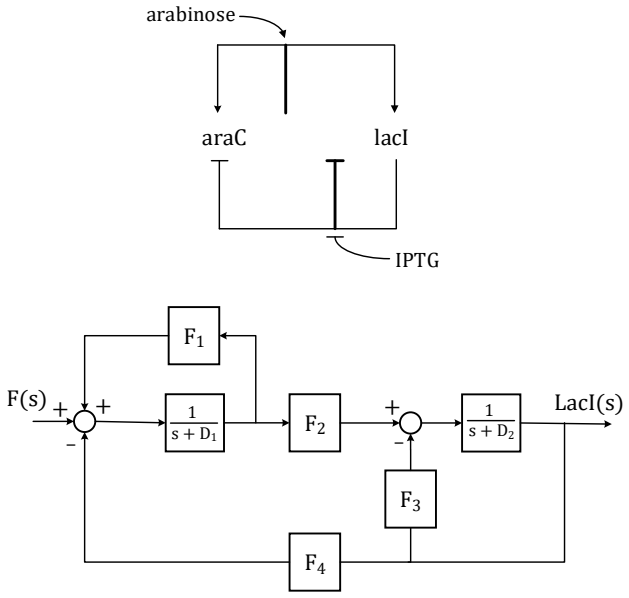

Fig. 7: Tunable Synthetic Gene Oscillator [41]

The genes *araC* and *lacI* have identical hybrid promoters that can be activated by *AraC* in the presence of arabinose and repressed by *LacI* in the absence of *IPTG*. Moreover, *araC* has a positive autoregulation, whereas, *LacI* has a negative autoregulation, as shown in Figure 7. We formalize the block diagram representation of the tunable synthetic gene oscillator, depicted in Figure 7, in HOL Light as follows:

Definition 14. $\vdash_{def} \forall$s D_1 D_2 F_1 F_2 F_3 F_4.
bdr_tsgo s D_1 D_2 F_1 F_2 F_3 F_4 =

$$\text{feedback\_block}\left(\text{series\_comp}\left[\text{feedback\_block}\left(\frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\ D_1}\right)\ (\text{Cx}\ F_1)\ ;\ \text{Cx}\ F_2;\ \text{feedback\_block}\left(\frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\ D_2}\right)\ (-\text{Cx}\ F_3)\right]\right)\ (-\text{Cx}\ F_4)$$

Next, we use the formalization of the block diagram representation of the tunable synthetic gene oscillator to formally verify its transfer function as the following HOL Light theorem:

Theorem 2. $\vdash_{thm} \forall$s D_1 D_2 F_1 F_2 F_3 F_4.
$[A_1]$: $(\text{s} + \text{Cx}\ (D_1 - F_1)) \neq \text{Cx}\ (\&0) \wedge$
$[A_2]$: $(\text{s} + \text{Cx}\ (D_2 + F_3)) \neq \text{Cx}\ (\&0) \wedge$
$[A_3]$: $((\text{s} + \text{Cx}\ (D_1 + F_1))(\text{s} + \text{Cx}\ (D_2 + F_3)) - \text{Cx}\ F_4) \neq \text{Cx}\ (\&0) \wedge$
$[A_4]$: $\left\| \frac{\text{Cx}\ F_1}{\text{s} + \text{Cx}\ (D_1 - F_1)} \right\| < \&1$
$[A_5]$: $\left\| \frac{\text{Cx}\ F_3}{\text{s} + \text{Cx}\ (D_2 + F_3)} \right\| < \&1$
$[A_6]$: $\left\| \frac{\text{Cx}\ (F_2 * F_4)}{(\text{s} + \text{Cx}\ (D_1 + F_1))(\text{s} + \text{Cx}\ (D_2 + F_3)) - \text{Cx}\ F_4} \right\| < \&1$
$[A_7]$: $(\text{s}^2 + \text{s} * \text{Cx}\ (D_1 + D_2 - F_1 + F_2) + \text{Cx}\ (D_1 * D_2) + \text{Cx}\ (D_1 * F_3) - \text{Cx}\ (D_2 * F_1) - \text{Cx}\ (F_1 * F_3) + \text{Cx}\ (F_2 * F_4)) \neq \text{Cx}\ (\&0)$
$\Rightarrow$ bdr_tsgo s D_1 D_2 F_1 F_2 F_3 F_4 =

$$\frac{\text{Cx}\ F_2}{\text{s}^2 + \text{s} * [\text{Cx}\ (D_1 + D_2 - F_1 + F_2)] + \text{Cx}\ (D_1 * D_2) + \text{Cx}\ (D_1 * F_3) - \text{Cx}\ (D_2 * F_1) - \text{Cx}\ (F_1 * F_3) + \text{Cx}\ (F_2 * F_4)}$$

Assumptions A_1-A_7 provide conditions on various parameters of the tunable synthetic gene oscillator. The conclusion of the above theorem presents the transfer function of the gene oscillator based on its block diagram representation. The proof of Theorem 2 is based on Definitions 9 and 13 along with some complex arithmetic reasoning.

Our formalization of the foundational configurations, presented above [38], enables us to formally model the block diagram representations of the generic biological circuits and their associated controllers as will be illustrated in the next section.

6 Formal Analysis of the Genetic Circuits and Controllers

Gene expression is a technique for transmitting information from the passive deoxyribonucleic acid (DNA) to the active proteins and is widely used in cellular biology. The process of this transmission is performed in two steps. In the first step, a section of DNA is read out into the ribonucleic acid (RNA) and is known as transcription. The second step, namely translation, involves conversion of a short strand of transcribed RNA into protein. This process is usually regulated in synthetic systems by transcription factors (TFs), which control the initiation rate of the transcription of a gene and its corresponding expression. TFs are of two types, namely activators and repressors. Activators increase the transcription rate, whereas the repressors inhibit transcription. We use our proposed formalization for formally analyzing the genetic circuits of the gene expression regulation, the activated and repressed expressions of protein, and phase lead and lag controllers.

6.1 Formal Dynamical Analysis

6.1.1 Gene Expression Regulation: The reaction-based model of the interaction between a TF, X (an activating or repressing TF) and the regulated gene G, is represented as follows [8]:

$$\mathrm{G} + \mathrm{nX} \underset{\mathrm{k}_r}{\overset{\mathrm{k}_f}{\rightleftharpoons}} G : X_n \tag{9}$$

where n captures the number of TF molecules binding to the gene and $G : X_n$ represents the gene-TF complex. Similarly, k_f and k_r provides the forward and reverse kinetic rates of the reaction, respectively. The corresponding dynamics (differential equation model) is mathematically expressed as follows [8]:

$$\frac{d[G]}{dt} = k_r[G : X_n] - k_f[G][X]^n \tag{10}$$

$$\frac{d[X]}{dt} = nk_r[G : X_n] - nk_f[G][X]^n \tag{11}$$

$$\frac{d[G : X_n]}{dt} = k_f[G][X]^n - k_r[G : X_n] \tag{12}$$

The above equation can be expressed in the vectorial form as follows:

$$\begin{bmatrix} \frac{d[G]}{dt} \\ \frac{d[X]}{dt} \\ \frac{d[G:X_n]}{dt} \end{bmatrix} = \begin{bmatrix} -1 \\ -n \\ 1 \end{bmatrix} \begin{bmatrix} k_f[G][X]^n - k_r[G : X_n] \end{bmatrix} \tag{13}$$

We first formalize the reaction-based model of the gene expression regulation in HOL Light as follows:

Definition 15. $\vdash_{def}$ ∀n G X GXn kf kr n t.
react_model_gene_expre_reg n G X GXn t =
[reversible,[G t, 1; X t, n; GXn t, 0],
[G t, 0; X t, 0; GXn t, 1], kf, kr]

Next, we formally verify the transformation of the reaction-based model to its corresponding dynamical model as the following HOL Light theorem:

Theorem 3. $\vdash_{thm}$ ∀n G X GXn t kr kf n.
(concen_vector_deriv [G; X; GXn] t =
stoichio_matrix
(react_model_gene_expre_reg G X GXn kf kr n t) ∗∗
flux_vector
(react_model_gene_expre_reg G X GXn kf kr n t)) ⟺
(vector_derivative G (at t) = kr ∗ GXn t - kf ∗ G t ∗ (X t)^n ∧
vector_derivative X (at t) =
Cx (&n) ∗ kr ∗ GXn t - Cx (&n) ∗ kf ∗ G t ∗ (X t)^n ∧
vector_derivative GXn (at t) = kf ∗ G t ∗ (X t)^n - kr ∗ GXn t)

The proof of the above theorem is based on the following lemmas for the verification of concen_vector_deriv, stoichio_matrix and flux_vector alongwith some complex arithmetic reasoning.

Lemma 1. $\vdash_{thm}$ ∀G X GXn t.
concen_vector_deriv [G; X; GXn] t =
vector [vector_derivative G (at t);
vector_derivative X (at t); vector_derivative GXn (at t)]

Lemma 2. $\vdash_{thm}$ ∀G X GXn kf kr n t.
stoichio_matrix
(react_model_gene_expre_reg G X GXn kf kr n t) =
vector [vector [- Cx (&1)]; vector [- Cx (&n)]; vector [Cx (&1)]]

Lemma 3. $\vdash_{thm}$ ∀G X GXn kf kr n t. [A]: (0 < n)
⇒ flux_vector
(react_model_gene_expre_reg G X GXn kf kr n t) =
vector [kf ∗ G t ∗ (X t)^n t - kr ∗ GXn t]

Lemma 4. $\vdash_{thm}$ ∀G X GXn kf kr n t. [A]: (n = 0)
⇒ flux_vector
(react_model_gene_expre_reg G X GXn kf kr n t) =
vector [kf ∗ G t - kr ∗ GXn t]

The dynamical model of the gene expression regulation is further used for analyzing the genetic circuits of the activated and repressed expressions of protein in the following sections.

6.1.2 Activated Expression of Protein: The genetic circuit of the activated expression of protein is depicted in Figure 8(a). It involves the interaction of the incoming activating TF A with its promoter and the regulation of the expression of gene Y producing the protein Y. The block diagram representation of the activated expression is shown in Figure 8(c) by a gain block of $+\gamma_x^*$ with $x = A$.

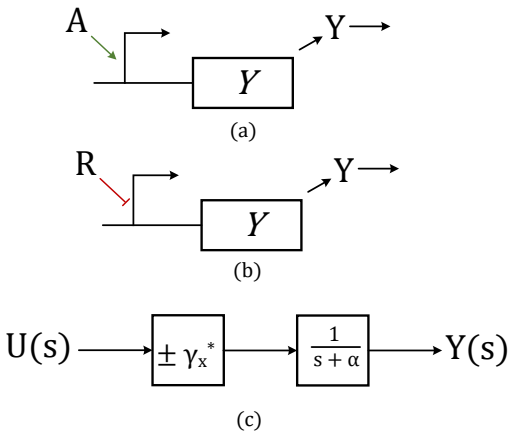

Fig. 8: (a) Genetic Circuit Diagram of Activated Expression (b) Genetic Circuit Diagram of Repressed Expression (c) Block Diagram Representation for a Single Gene with Activated and Repressed Expressions

The dynamical model of the activated expression of protein is mathematically expressed as the following linear differential equation [8]:

$$\frac{dy}{dt} + \alpha y = \gamma_A^* u \tag{14}$$

The corresponding transfer function is mathematically expressed as:

$$\frac{Y(s)}{U(s)} = \frac{\gamma_A^*}{s + \alpha} \tag{15}$$

In order to model the dynamical behaviour, we first model the linear differential equation of order n as:

Definition 16. $\vdash_{def}$ ∀k f t. differ_equat_order_n k l f t =
vsum (0..k) (λi. EL i [α_1;α_2;...;α_i] ∗
higher_order_derivative i f t)

The function differ_equat_order_n takes the order of the linear differential equation k, list of coefficients l, a differentiable function f and the differentiation variable t and returns the n^{th} order linear differential equation.

Now, we model the dynamical behaviour of the activated expression as the following HOL Light function [26]:

Definition 17. $\vdash_{def}$ ∀α. olst_de_ae α = [Cx α; Cx (&1)]
$\vdash_{def}$ ∀$\gamma_A{}^*$. ilst_de_ae $\gamma_A{}^*$ = [Cx $\gamma_A{}^*$]
$\vdash_{def}$ differ_equat_ae u y t α $\gamma_A{}^*$ ⇔
differ_equat_order_n 1 (olst_de_ae α) y t =
differ_equat_order_n 0 (ilst_de_ae $\gamma_A{}^*$) u t

To formally verify the transfer function of the activated expression based on its dynamical model, we first model its block diagram representation using our formalization in HOL Light as [26]:

Definition 18. $\vdash_{def}$ ∀α $\gamma_A{}^*$.
bdr_ae α $\gamma_A{}^*$ = series_comp [Cx $\gamma_A{}^*$; $\frac{Cx(\&1)}{s + Cx\ \alpha}$]

Next, we verify the transfer function of the activated expression based on its block diagram representation as the following HOL Light theorem [26]:

Theorem 4. $\vdash_{thm} \forall \alpha\ \gamma_A{}^*.$ [A]: $(s + Cx\ \alpha) \neq Cx\ (\&0)$
$\Rightarrow$ bdr_ae $\alpha\ \gamma_A{}^* = \dfrac{Cx\ \gamma_A{}^*}{s + Cx\ \alpha}$

The proof of the above theorem is based on Definition 9 along with some arithmetic reasoning. Now, we formally verify the transfer function, obtained from Theorem 4, based on the dynamical model as follows [26]:

Theorem 5. $\vdash_{thm} \forall \alpha\ \gamma_A{}^*$ y u s.
$[A_1] : \&0 < \gamma_A{}^* \wedge [A_2] : \&0 < \alpha \wedge$
$[A_3] :$ $\forall$t. differen_higher_derivat u y t $\wedge$
$[A_4] :$ existence_of_laplace_higher_derivat u y $\wedge$
$[A_5] :$ zero_ini_condit u y $\wedge$
$[A_6] :$ ($\forall$t. differ_equat_ae u y t $\alpha\ \gamma_A{}^*$) $\wedge$
$[A_7] :$ (laplace_transform u s $\neq$ Cx (&0)) $\wedge$
$[A_8] : \left(\dfrac{Cx(\&1)}{s + Cx\ \alpha} \neq Cx(\&0)\right)$
$\Rightarrow \dfrac{\text{laplace\_transform y s}}{\text{laplace\_transform u s}} = \dfrac{Cx\ \gamma_A{}^*}{s + Cx\ \alpha}$

Assumptions A_1-A_2 model the positivity conditions on circuit's parameters. Assumptions A_3-A_4 provide the differentiability and condition of the existence of the Laplace transform of the higher-order derivative of y up to order 1 and and the function u, respectively. Similarly, Assumption A_5 models the zero initial conditions for y and u. Assumption A_6 presents the dynamical behaviour of the activated expression. Assumptions A_7-A_8 ensure that the denominator of the transfer function, presented in the conclusion of the above theorem, provides a valid expression. Finally, the conclusion provides the transfer function of the activated expression. The proof of Theorem 5 is done almost automatically using the automatic tactic TFUN_TAC, which is developed as part of our proposed formalization.

6.1.3 Repressed Expression of Protein: Figure 8(b) depicts the genetic circuit of the repressed expression of protein. It involves the interaction of the incoming repressing TF A with its promoter and the regulation of the expression of gene Y producing the protein Y. The block diagram representation of the repressed expression is shown in Figure 8(c), by a gain block of $-\gamma_x^*$ with $x = R$. The dynamical model of the repressed expression of protein is mathematically expressed as [8]:

$$\frac{dy}{dt} + \alpha y = -\gamma_R^* u \tag{16}$$

The corresponding transfer function is mathematically expressed as:

$$\frac{Y(s)}{U(s)} = \frac{-\gamma_R^*}{s + \alpha} \tag{17}$$

We formally verified the block diagram representation of the repressed expression, its transfer function based on its block diagram and its dynamical model and the details about this verification can be found at [38].

6.1.4 Autoactivation of Gene: The activation of its own expression by a gene is known as autoactivation. The block diagram representation of the protein expression with autoactivation is depicted in Figure 9.

The dynamical model of the protein expression with autoactivation is mathematically expressed as the following linear differential equation [8]:

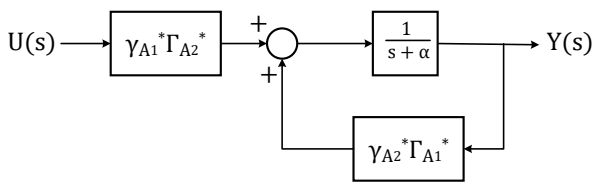

Fig. 9: Block Diagram Representation of Protein Expression with Autoactivation

$$\frac{dy}{dt} + (\alpha - \gamma_{A_2}^* \Gamma_{A_1}^*) y = \gamma_{A_1}^* \Gamma_{A_2}^* u \tag{18}$$

The corresponding transfer function is mathematically expressed as:

$$\frac{Y(s)}{U(s)} = \frac{\gamma_{A_1}^* \Gamma_{A_2}^*}{s + \alpha - \gamma_{A_2}^* \Gamma_{A_1}^*} \tag{19}$$

We model the dynamical behaviour of the protein expression with autoactivation as the following HOL Light function:

Definition 19. $\vdash_{def} \forall \alpha\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*.$ olst_de_aa $\alpha\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^* =$
[Cx α - Cx $\gamma_{A_2}{}^*$ * Cx $\Gamma_{A_1}{}^*$; Cx (&1)]
$\vdash_{def} \forall \gamma_{A_1}{}^*\ \Gamma_{A_2}{}^*.$ ilst_de_aa $\gamma_{A_1}{}^*\ \Gamma_{A_2}{}^* =$ [Cx $\gamma_{A_1}{}^*$ * Cx $\Gamma_{A_2}{}^*$]
$\vdash_{def}$ differ_equat_aa u y t $\alpha\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \gamma_{A_1}{}^*\ \Gamma_{A_2}{}^* \Leftrightarrow$
differ_equat_order_n 1 (olst_de_aa $\alpha\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*$) y t =
differ_equat_order_n 0 (ilst_de_aa $\gamma_{A_1}{}^*\ \Gamma_{A_2}{}^*$) u t

To formally verify the transfer function of the protein expression with autoactivation based on its dynamical model, we first model its block diagram representation using our formalization in HOL Light as:

Definition 20. $\vdash_{def} \forall$s $\alpha\ \gamma_{A_1}{}^*\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \Gamma_{A_2}{}^*.$
bdr_aa s $\alpha\ \gamma_{A_1}{}^*\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \Gamma_{A_2}{}^* =$
series_comp [Cx $\gamma_{A_1}{}^*$ * Cx $\Gamma_{A_2}{}^*$;
feedback_block $\left(\frac{Cx(\&1)}{s + Cx\ \alpha}\right)$ (Cx $\gamma_{A_2}{}^*$ * Cx $\Gamma_{A_1}{}^*$)]

Next, we verify the transfer function of the protein expression with autoactivation based on its block diagram representation as the following HOL Light theorem:

Theorem 6. $\vdash_{thm} \forall$s $\alpha\ \gamma_{A_1}{}^*\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \Gamma_{A_2}{}^*.$
$[A_1] : \&0 < \gamma_{A_2}{}^* \wedge [A_2] : \&0 < \Gamma_{A_1}{}^* \wedge$
$[A_3] : (s + Cx\ \alpha) \neq Cx\ (\&0) \wedge$
$[A_4] : (s + Cx\ \alpha - Cx\ \gamma_{A_2}{}^* * Cx\ \Gamma_{A_1}{}^*) \neq Cx\ (\&0) \wedge$
$[A_5] : \left|\left| \dfrac{Cx\ \gamma_{A_1}{}^* * Cx\ \Gamma_{A_2}{}^*}{s + Cx\ \alpha - Cx\ \gamma_{A_2}{}^* * Cx\ \Gamma_{A_1}{}^*} \right|\right| < \&1$
$\Rightarrow$ bdr_aa s $\alpha\ \gamma_{A_1}{}^*\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \Gamma_{A_2}{}^* =$
$\dfrac{Cx\ \gamma_{A_1}{}^* * Cx\ \Gamma_{A_2}{}^*}{s + Cx\ \alpha - Cx\ \gamma_{A_2}{}^* * Cx\ \Gamma_{A_1}{}^*}$

Assumptions A_1-A_5 provide various conditions on circuit's parameters. The conclusion of the above theorem presents the transfer function of the autoactivation of gene based on its block diagram representation. The proof of Theorem 6 is based on Definitions 9 and 13 along with some complex arithmetic reasoning. Now, we formally verify the transfer function, obtained from Theorem 6, based on the dynamical model as follows:

Theorem 7. $\vdash_{thm} \forall \alpha\ \gamma_{A_1}{}^*\ \gamma_{A_2}{}^*\ \Gamma_{A_1}{}^*\ \Gamma_{A_2}{}^*$ y u s.
$[A_1] : \&0 < \alpha \wedge [A_2] : \&0 < \gamma_{A_2}{}^* \wedge [A_3] : \&0 < \Gamma_{A_1}{}^* \wedge$
$[A_4] : \&0 < \gamma_{A_1}{}^* \wedge [A_5] : \&0 < \Gamma_{A_2}{}^* \wedge$
$[A_6] :$ $\forall$t. differen_higher_derivat u y t $\wedge$
$[A_7] :$ existence_of_laplace_higher_derivat u y $\wedge$

$[A_8]$: zero_ini_condit u y ∧
$[A_9]$: (∀t. differ_equat_aa u y t α $\gamma_{A_2}{}^*$ $\Gamma_{A_1}{}^*$ $\gamma_{A_1}{}^*$ $\Gamma_{A_2}{}^*$) ∧
$[A_{10}]$: (laplace_transform u s ≠ Cx (&0)) ∧
$[A_{11}]$: $\left(\frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\,\alpha - \text{Cx}\,\gamma_{A_2}{}^* * \text{Cx}\,\Gamma_{A_1}{}^*} \neq \text{Cx}(\&0)\right)$
$\Rightarrow \frac{\text{laplace\_transform y s}}{\text{laplace\_transform u s}} = \frac{\text{Cx}\,\gamma_{A_1}{}^* * \text{Cx}\,\Gamma_{A_2}{}^*}{\text{s} + \text{Cx}\,\alpha - \text{Cx}\,\gamma_{A_2}{}^* * \text{Cx}\,\Gamma_{A_1}{}^*}$

Assumptions A_1-A_5 capture the positivity conditions on circuit's parameters. Assumptions A_6-A_7 model the differentiability and condition of the existence of the Laplace transform of the higher-order derivative of y up to order 1 and the function u, respectively. Similarly, Assumption A_8 provides the zero initial conditions for y and u. Assumption A_9 models the dynamical behaviour of the activated expression. Assumptions A_{10}-A_{11} ensure that the denominator of the transfer function, presented in the conclusion of the above theorem, provides a valid expression. Finally, the conclusion presents the transfer function of the autoactivation. The proof of the above theorem is done almost automatically using the automatic tactic TFUN_TAC.

6.1.5 Phase Lag and Lead Controllers: The block diagram representation for phase lag and lead controllers is depicted in Figure 10. The variable γ_c^* is the promoter gain of the controllers, whereas, ${B_1}^*$ and ${B_2}^*$ are the gains of the downstream promoters. The variable α represents the degradation/dilution rate.

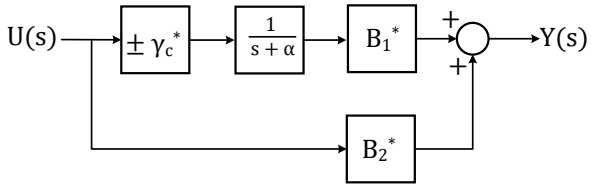

Fig. 10: Block Diagram of a Genetic Phase Lead (+) and Phase Lag (-) Controller

The dynamical model of the phase lag controller is mathematically expressed as [8]:

$$\frac{dy}{dt} + \alpha y = {B_2}^* \frac{du}{dt} + (\alpha B_2^* - \gamma_c^* B_1^*)u \tag{20}$$

The corresponding transfer function is given by the following mathematically expression:

$$\frac{Y(s)}{U(s)} = B_2^* \frac{s + \alpha - \frac{\gamma_c^* B_1^*}{B_2^*}}{s + \alpha} \tag{21}$$

We model the dynamical behaviour of the phase lag controller as the following HOL Light function:

Definition 21. $\vdash_{def}$ ∀α. olst_de_plagc α = [Cx α; Cx (&1)]
$\vdash_{def}$ ∀α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$. ilst_de_plagc α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$ =
[Cx α * Cx B_2^* − Cx γ_c^* * Cx B_1^*; Cx $B_2{}^*$]
$\vdash_{def}$ differ_equat_plagc u y t α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$ ⇔
differ_equat_order_n 1 (olst_de_plagc α) y t =
differ_equat_order_n 1 (ilst_de_plagc α $B_1{}^*$ $B_2{}^*$ $\gamma_c{}^*$) u t

To formally verify the transfer function of the activated expression based on its dynamical model, we first model its block diagram representation using our formalization in HOL Light as:

Definition 22. $\vdash_{def}$ ∀s α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$.
bdr_plagc s α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$ = summ_jun [series_comp [− Cx $\gamma_c{}^*$; $\frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\,\alpha}$; Cx $B_1{}^*$]; Cx $B_2{}^*$]

Next, we verify the transfer function of the phase lag controller based on its block diagram representation as follows:

Theorem 8. $\vdash_{thm}$ ∀s α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$.
[A]: (s + Cx α) ≠ Cx (&0)
⇒ bdr_plagc α $\gamma_c{}^*$ $B_1{}^*$ $B_2{}^*$ = $\frac{\text{Cx}\,B_2{}^* * (\text{s} + \text{Cx}\,\alpha) + \text{Cx}\,\gamma_c{}^* * \text{Cx}\,B_2{}^*}{(\text{s} + \text{Cx}\,\alpha)}$

The proof of the above theorem is mainly based on Definitions 9 and 10 along with some complex arithmetic reasoning. We also formally verified the transfer function of the phase lag controller based on its dynamical model. Similarly, we formalized the dynamical model and block diagram representation of the phase lead controller and formally verified its transfer function. The details about all these verification results can be found in our proof script [38].

6.1.6 Genetic Phase Lag Controller in Feedback with a Single Gene: The block diagram representation of the genetic phase lag controller in feedback with a single gene is depicted in Figure 11.

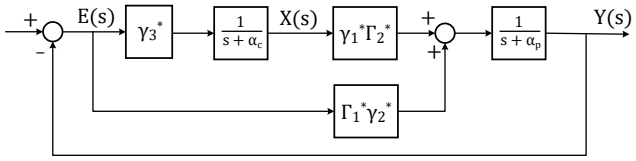

Fig. 11: Genetic Phase Lag Controller in Feedback with a Single Gene

The dynamical model of this phase lag controller is mathematically expressed as follows [8]:

$$\frac{d^2y}{dt^2} + (\alpha_c + \alpha_p)\frac{dy}{dt} + \alpha_c\alpha_p y = \Gamma_1^*\gamma_2^*\frac{de}{dt} + (\alpha_c\Gamma_1^*\gamma_2^* + \gamma_1^*\Gamma_2^*\gamma_3^*)e \tag{22}$$

The corresponding transfer function is mathematically expressed as [8]:

$$\frac{Y(s)}{U(s)} = \frac{s\Gamma_1^*\gamma_2^* + (\alpha_c\Gamma_1^*\gamma_2^* + \gamma_1^*\Gamma_2^*\gamma_3^*)}{s^2 + s(\alpha_c + \alpha_p) + \alpha_c\alpha_p} \tag{23}$$

We model the dynamical behaviour of the phase lag controller as the following HOL Light function:

Definition 23 $\vdash_{def}$ ∀α_c α_p. olst_de_plcnf α_c α_p =
[Cx α_c * Cx α_p; Cx α_c + Cx α_p; Cx (&1)]
$\vdash_{def}$ ∀α_c $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$.
ilst_de_plcnf α_c $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$ =
[Cx $\alpha_c{}^*$ * Cx $\Gamma_1{}^*$ * Cx $\gamma_2{}^*$ +
Cx $\gamma_1{}^*$ * Cx $\Gamma_2{}^*$ * Cx $\gamma_3{}^*$; Cx $\Gamma_1{}^*$ * Cx $\gamma_2{}^*$]
$\vdash_{def}$ differ_equat_plcnf α_c α_p $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$ u y t ⇔
differ_equat_order_n 2 (olst_de_plcnf α_c α_p) y t =
differ_equat_order_n 1
(ilst_de_plcnf α_c $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$) u t

To formally verify the transfer function of the phase lag controllers based on its dynamical model, we first model its block diagram representation using our formalization in HOL Light as:

Definition 24. $\vdash_{def}$ ∀s α_c α_p $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$.
bdr_plcnf s α_c α_p $\gamma_1{}^*$ $\gamma_2{}^*$ $\gamma_3{}^*$ $\Gamma_1{}^*$ $\Gamma_2{}^*$

$$= \text{series\_comp}\left[\text{summ\_jun}\left[\text{series\_comp}\left[\text{Cx}\,\gamma_3^*;\ \frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\,\alpha_c};\ \text{Cx}\,\gamma_1^* * \text{Cx}\,\Gamma_2^*\right];\ \text{Cx}\,\Gamma_1^* * \text{Cx}\,\gamma_2^*\right];\ \frac{\text{Cx}(\&1)}{\text{s} + \text{Cx}\,\alpha_p}\right]$$

Next, we verify the transfer function of the phase lag controller based on its block diagram representation as the following HOL Light theorem:

Theorem 9. $\vdash_{thm}$ $\forall$s α_c α_p γ_1^* γ_2^* γ_3^* Γ_1^* Γ_2^*.
[A_1] : (s + Cx α_p) $\neq$ Cx (&0) $\wedge$
[A_2] : (s + Cx α_c) $\neq$ Cx (&0) $\wedge$
[A_3] : (s^2 + s * (Cx α_c + Cx α_p) + Cx α_c * Cx α_p) $\neq$ Cx (&0)
$\Rightarrow$ bdr_plcnf s α_c α_p γ_1^* γ_2^* γ_3^* Γ_1^* Γ_2^* =

$$\frac{\text{s} * \text{Cx}\,\Gamma_1^* * \text{Cx}\,\gamma_2^* + (\text{Cx}\,\alpha_c * \text{Cx}\,\Gamma_1^* * \text{Cx}\,\gamma_2^* + \text{Cx}\,\gamma_1^* * \text{Cx}\,\Gamma_2^* * \text{Cx}\,\gamma_3^*)}{(\text{s}^2 + \text{s} * (\text{Cx}\,\alpha_c + \text{Cx}\,\alpha_p) + \text{Cx}\,\alpha_c * \text{Cx}\,\alpha_p)}$$

Assumptions A_1-A_3 provide conditions on various parameters of the phase lag controller. The conclusion of the above theorem presents the transfer function of the controller based on its block diagram representation. The proof of Theorem 9 is based on Definitions 9 and 10 along with some complex arithmetic reasoning. Now, we formally verify the transfer function, obtained from Theorem 9, based on the dynamical model as follows:

Theorem 10. $\vdash_{thm}$ $\forall$s u y α_c α_p γ_1^* γ_2^* γ_3^* Γ_1^* Γ_2^*.
[A_1] : &0 < α_c $\wedge$ [A_2] : &0 < α_p $\wedge$ [A_3] : &0 < Γ_1^* $\wedge$
[A_4] : &0 < Γ_2^* $\wedge$ [A_5] : &0 < γ_1^* $\wedge$ [A_6] : &0 < γ_2^* $\wedge$
[A_7] : &0 < γ_3^* $\wedge$
[A_8] : $\forall$t. differen_higher_derivat u y t $\wedge$
[A_9] : existence_of_laplace_higher_derivat u y $\wedge$
[A_{10}] : zero_ini_condit u y $\wedge$
[A_{11}] : ($\forall$t. differ_equat_plcnf α_c α_p γ_1^* γ_2^* γ_3^* Γ_1^* Γ_2^* u y t) $\wedge$
[A_{12}] : (laplace_transform u s $\neq$ Cx (&0)) $\wedge$
[A_{13}] : ((s^2 + s * (Cx α_c + Cx α_p) + Cx α_c * Cx α_p) $\neq$ Cx(&0))
$\Rightarrow$ $\frac{\text{laplace\_transform y s}}{\text{laplace\_transform u s}}$ =

$$\frac{\text{s} * \text{Cx}\,\Gamma_1^* * \text{Cx}\,\gamma_2^* + (\text{Cx}\,\alpha_c * \text{Cx}\,\Gamma_1^* * \text{Cx}\,\gamma_2^* + \text{Cx}\,\gamma_1^* * \text{Cx}\,\Gamma_2^* * \text{Cx}\,\gamma_3^*)}{(\text{s}^2 + \text{s} * (\text{Cx}\,\alpha_c + \text{Cx}\,\alpha_p) + \text{Cx}\,\alpha_c * \text{Cx}\,\alpha_p)}$$

Assumptions A_1-A_7 provide the positivity conditions on circuit's parameters. Assumptions A_8-A_9 capture the differentiability and condition of the existence of the Laplace transform of the higher-order derivatives of y and e up to order 2 and 1, respectively. Similarly, Assumption A_{10} models the zero initial conditions for y and u. Assumption A_{11} provides the dynamical behaviour of the activated expression. Assumptions A_{12}-A_{13} ensure that the denominator of the transfer function, presented in the conclusion of the above theorem, provides a valid expression. Finally, the conclusion presents the transfer function of the phase lag controller. The proof of the above theorem is done almost automatically using the automatic tactic TFUN_TAC.

7 Formal Stability Analysis

Stability of a biological system is its capability of returning to the equilibrium state after suffering from various disturbances [42]. Thus, a stable biological system ensures a stable response to a bounded input. Stability of the biological systems, such as biological circuits and bio-controllers, is based on their transfer functions that are obtained as a result of analyzing their dynamical behaviour.

Generally, the transfer function of a biological system is mathematically represented as:

$$\frac{Y(s)}{U(s)} = \frac{Numer(s)}{Denomin(s)} = \frac{p_m s^m + p_{m-1}s^{m-1} + ... + p_0}{q_n s^n + q_{n-1}s^{n-1} + ... + q_0} \quad (24)$$

where $Y(s)$ and $U(s)$ provide the Laplace transform of the output $y(t)$ and input $u(t)$ functions, respectively. Similarly, $Numer(s)$ and $Denomin(s)$ are complex-valued polynomials modeling the numerator and denominator of the transfer function. The equation $Denomin(s) = 0$ represents the characteristic equation and its roots are known as poles of the system. The orientation and placement of these poles in the complex plane provides vital information about the stability of the underlying system. If all poles of a system are located in the left half of the complex plane then it is said to be stable system [8].

We model the notion of stability of a biological system as follows:

Definition 25. $\vdash_{def}$ $\forall$F. is_stable_biosys F = {s | F s = Cx (&0) $\wedge$ Re s < &0} $\neq$ EMPTY

where is_stable_biosys takes the denominator of the transfer function corresponding to the dynamics of a biological circuit or controller, i.e., F:$\mathbb{C} \to \mathbb{C}$, and provides a stable system. Similarly, s:$\mathbb{C}$ captures the root of the characteristic equation. The conjunct F s = Cx (&0) models the characteristic equation. Similarly, Re s < &0 provides the condition that the poles of the system lie in the left half of the complex plane.

We formally verify the stability of the autoactivation of the gene corresponding to the transfer function, given in Equation (19), as the following HOL Light theorem:

Theorem 11. $\vdash_{thm}$ $\forall\alpha$ $\gamma_{A_2}^*$ $\Gamma_{A_1}^*$. [A:] $\gamma_{A_2}^*$ * $\Gamma_{A_1}^*$ < α
$\Rightarrow$ is_stable_biosys (s. s + Cx α − Cx $\gamma_{A_2}^*$ * Cx $\Gamma_{A_1}^*$)

Assumption A provides the necessary condition for the stability of the autoactivation. The proof of the above theorem is based on Definition 25 along with some complex arithmetic reasoning. Similarly, we formally verified the stability of the aggressive and repressed expressions of proteins and more details can be found at [38]. Finally, we formally verified the stability of the phase lag controller with negative feedback as the following HOL Light theorem:

Theorem 12. $\vdash_{thm}$ $\forall\alpha_c$ α_p. [A:] ((&0 < α_c + α_p $\wedge$
(((α_c + α_p)^2 − &4 * (α_c * α_p) < &0 $\vee$
(α_c + α_p)^2 − &4 * (α_c * α_p) = &0)) $\vee$
(&0 < (α_c + α_p)^2 − &4 * (α_c * α_p) $\wedge$
($\sqrt{(\alpha_c + \alpha_p)^2 - \&4 * (\alpha_c * \alpha_p)} < (\alpha_c + \alpha_p)$ $\vee$
−(α_c + α_p) < $\sqrt{(\alpha_c + \alpha_p)^2 - \&4 * (\alpha_c * \alpha_p)}$))))
$\Rightarrow$ is_stable_biosys (s. s^2 + s * (Cx α_c + Cx α_p) + Cx α_c * Cx α_p)

The formal reasoning for the proof of the above theorem is very similar to that of Theorem 12 and the details about the verification can be found at [38]. Due to the undecidable nature of the higher-order logic, the above-mentioned formal analysis involved manual interventions and human guidance. However, we developed the tactic TFUN_TAC [38] to automate the verification of the transfer functions of the biological circuits and their associated controllers. The details about this tactic and rest of the formalization can be found in our proof script [38].

8 Discussion

In this paper, we developed a distinguished formal analysis such that all of the proved theorems are of generic nature, i.e., all of the functions and variables are universally quantified and hence can be specialized based on the requirement of analyzing the biological circuits. On the other hand, in the case of computer based simulations and numerical methods, we require modeling each case individually. Furthermore, the inherent correctness of the theorem proving

Table 3 Verification Details for each Theorem

Formalized Theorems	Proof Lines	Man-hours
Theorem 3 (Transformation from Reaction-based Model to its Equivalent Dynamical Model)	75	8
Lemma 1 (Vector Containing Derivatives of the Concentration of Species)	14	$\frac{1}{2}$
Lemma 2 (Stoichiometric Matrix)	22	1
Lemma 3 (Flux Vector for the case of $n < 0$)	17	1
Lemma 4 (Flux Vector for the case of $n = 0$)	20	1
Theorem 4 (Transfer Function of Activated Expression)	7	1
Theorem 5 (Dynamical Model Implies Transfer Function)	2	$\frac{1}{4}$
Theorem 6 (Transfer Function of the Protein Expression with Autoactivation)	28	2
Theorem 7 (Dynamical Model Implies Transfer Function)	3	$\frac{1}{4}$
Theorem 8 (Transfer Function of the Phase Lag Controller)	9	1
Theorem 9 (Transfer Function of Phase Lag Controller in Feedback with a Single Gene)	18	2
Theorem 10 (Dynamical Model Implies Transfer Function)	2	$\frac{1}{4}$
Theorem 11 (Stable Autoactivation)	27	4
Theorem 12 (Stable Phase Lag Controller with Negative Feedback)	65	9

method confirms that all the needed assumptions are explicitly mentioned within the respective theorem statement. Also, because of the high expressiveness of the higher-order logic, our methodology allows us to model the dynamics of the biological circuits involving differential and derivative (Equations (14), (16), (18), (20), (22)) in their true form, whereas, in their corresponding model checking based analysis [20], they are discretized and modeled using a state-transition system, which compromises the accuracy and completeness of the corresponding analysis.

The effort spent in the verification of each theorem represented in the form of proof lines and the man-hours is presented in Table 3. The verification of Theorems 5, 7, 10 was done almost automatically, thanks to our automatic tactic `TFUN_TAC`. Note that the man-hours are based on the number of code lines in addition to the proof complexity. Therefore, lines number of the proof script do not have a direct relationship with the man-hours. For instance, the man-hours for the verification of Lemmas 3 and 4 are identical, while the proof lines for the former are less than that for the later.

Our proposed approach allows us to perform the reaction kinetic based analysis of an arbitrary biological system, i.e., a biological system having i species (reactants and products) and j reactions, where i and j can take any integer values, i.e., $1, 2, ...,$. Similarly, our proposed approach caters for a generic linear system of order n by performing the transfer function based analysis of the corresponding system, where, $n = 1, 2, 3, , ...,$. Moreover, the verification of the associated theorems is mainly based on the formal definitions along with some complex arithmetic reasoning. Therefore, the size of the formal model does not affect the effort (proof lines and man-hours) involved in the verification too much. Moreover, the proof-process of these theorems can be automated by writing some automatic tactics, like `TFUN_TAC`, which is used to perform the transfer function based analysis of any system of order n.

9 Conclusion

In this paper, we proposed a higher-order-logic theorem proving based framework to formally reason about synthetic biology, in particular, the biological circuits and their associated controllers. We first formalized the notion of the reaction kinetics, which provides the reaction-based model of any biological system. We also formalized the dynamical behaviour, based on reaction-based models, and the block diagram representations of the biological circuits and their associated controllers. Finally, we formally verified their transfer functions based on their dynamical models and their associated block diagram representations, and performed their stability analysis. We illustrated the practical effectiveness of our proposed approach by formally analyzing the activated and repressed expressions, and autoactivation of the protein, and phase lag and lead controllers. Our proposed approach focussed on the formal analysis of linear systems only. In future, we aim to extend our work to formally analyze the non-linear control systems in the context of biological systems. This idea is based on linearization of the non-linear models and our proposed framework can be directly used for their corresponding formal analysis. Moreover, we also plan to formally verify some more control systems properties of the biological circuits, such as, performance, robustness and sensitivity etc. Another future direction is to develop automatic tactics to automate the formal proofs of theorems describing various properties of the biological circuits and their associated controllers.

10 Acknowledgments

This work was supported and funded by Kuwait University, Research Project No. (EO 01/18).

11 References

1 Muschler, G.F., Nakamoto, C., Griffith, L.G.: 'Engineering Principles of Clinical Cell-based Tissue Engineering', *JBJS*, 2004, **86**, (7), pp. 1541–1558
2 Alon, U.: 'An Introduction to Systems Biology: Design Principles of Biological Circuits'. (Chapman and Hall/CRC, 2006)
3 Alberts, B.: 'Molecular Biology of the Cell'. (Garland Science, 2017)
4 Baldwin, G.: 'Synthetic Biology'. (John Wiley & Sons, 2012)
5 Ogata, K., Yang, Y.: 'Modern Control Engineering'. vol. 4. (London, 2002)
6 Andrianantoandro, E., Basu, S., Karig, D.K., Weiss, R.: 'Synthetic Biology: New Engineering Rules for an Emerging Discipline', *Molecular Systems Biology*, 2006, **2**, (1), pp. 2006–0028
7 Benner, S.A., Sismour, A.M.: 'Synthetic Biology', *Nature Reviews Genetics*, 2005, **6**, (7), pp. 533
8 Harris, A.W., Dolan, J.A., Kelly, C.L., Anderson, J., Papachristodoulou, A.: 'Designing Genetic Feedback Controllers', *IEEE Transactions on Biomedical Circuits and Systems*, 2015, **9**, (4), pp. 475–484
9 Nise, N.S.: 'Control Systems Engineering'. (John Wiley & Sons, 2007)
10 De.Zeeuw, R.A., Baker, G.B., Coutts, R.T., Vercruysse, A.: 'Evaluation of Analytical Methods in Biological Systems: Analysis of Biogenic Amines'. vol. 4. (Elsevier, 1982)
11 Haefner, J.W.: 'Modeling Biological Systems:: Principles and Applications'. (Springer Science & Business Media, 2005)
12 Gaylord, R.J., Wellin, P.R., Titus, B., McKay, S.R., Christian, W.: 'Computer Simulations with Mathematica: Explorations in Complex Physical and Biological Systems', *Computers in Physics*, 1996, **10**, (4), pp. 349–350
13 Harrison, J., Théry, L. 'Extending the HOL Theorem Prover with a Computer Algebra System to Reason about the Reals'. In: HOL Users' Group Workshop. (Springer, 1993. pp. 174–184
14 Clarke, E.M., Wing, J.M.: 'Formal Methods: State of the Art and Future Directions', *ACM Computing Surveys*, 1996, **28**, (4), pp. 626–643
15 Rashid, A., Hasan, O., Siddique, U., Tahar, S.: 'Formal Reasoning about Systems Biology using Theorem Proving', *PLOS ONE*, 2017, **12**, (7), pp. e0180179
16 Camilleri, A., Gordon, M., Melham, T.: 'Hardware Verification Using Higher-Order Logic'. (University of Cambridge, Computer Laboratory, 1986)
17 Schumann, J.M.: 'Automated Theorem Proving in Software Engineering'. (Springer Science & Business Media, 2001)
18 Grumberg, O., Clarke, E.M., Peled, D.. 'Model Checking'. (The MIT Press Cambridge, 1999
19 Hasan, O., Tahar, S. 'Formal Verification Methods'. In: Encyclopedia of Information Science and Technology, Third Edition. (IGI Global, 2015. pp. 7162–7170
20 Yordanov, B., Appleton, E., Ganguly, R., Gol, E.A., Carr, S.B., Bhatia, S., et al. 'Experimentally Driven Verification of Synthetic Biological Circuits'. In: Design, Automation & Test in Europe. (IEEE, 2012. pp. 236–241
21 Bartocci, E., Bortolussi, L., Nenzi, L. 'A Temporal Logic Approach to Modular Design of Synthetic Biological Circuits'. In: Computational Methods in Systems Biology. (Springer, 2013. pp. 164–177
22 Madsen, C., Myers, C.J., Roehner, N., Winstead, C., Zhang, Z. 'Utilizing Stochastic Model Checking to Analyze Genetic Circuits'. In: IEEE Symposium on

Computational Intelligence in Bioinformatics and Computational Biology. (IEEE, 2012. pp. 379–386
23 Clarke, E.M., Grumberg, O., Jha, S., Lu, Y., Veith, H. 'Progress on the State Explosion Problem in Model Checking'. In: Informatics. (Springer, 2001. pp. 176–194
24 Ahmad, S., Hasan, O., Siddique, U., Tahar, S. 'Formalization of Zsyntax to Reason About Molecular Pathways in HOL4'. In: Brazilian Symposium on Formal Methods. (Springer, 2014. pp. 32–47
25 Ahmad, S., Hasan, O., Siddique, U.: 'On the formalization of zsyntax with applications in molecular biology', *Scalable Computing: Practice and Experience*, 2015, **16**, (1), pp. 37–52
26 (, .
27 Harrison, J.: 'Handbook of Practical Logic and Automated Reasoning'. (Cambridge University Press, 2009)
28 Harrison, J. 'HOL Light: A Tutorial Introduction'. In: Formal Methods in Computer-Aided Design. vol. 1166 of *LNCS*. (Springer, 1996. pp. 265–269
29 Abrams, M., et al.. 'A History of OCaml'. (, 2015. `http://ocaml.org/learn/history.html`
30 Taqdees, S.H., Hasan, O. 'Formalization of Laplace Transform Using the Multivariable Calculus Theory of HOL-Light'. In: Logic for Programming, Artificial Intelligence, and Reasoning. vol. 8312 of *LNCS*. (Springer, 2013. pp. 744–758
31 Rashid, A., Hasan, O. 'Formalization of Transform Methods using HOL Light'. In: Conference on Intelligent Computer Mathematics. vol. 10383 of *LNAI*. (Springer, 2017. pp. 319–332
32 Pilling, M.J., Seakins, P.W., et al.: 'Reaction Kinetics'. (Oxford University Press, 1996)
33 Azimi, S., Iancu, B., Petre, I.: 'Reaction System Models for the Heat Shock Response', *Fundamenta Informaticae*, 2014, **131**, (3), pp. 299–312
34 Chen, Y.J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., et al.: 'Programmable Chemical Controllers Made from DNA', *Nature Nanotechnology*, 2013, **8**, (10), pp. 755
35 Yordanov, B., Kim, J., Petersen, R.L., Shudy, A., Kulkarni, V.V., Phillips, A.: 'Computational Design of Nucleic Acid Feedback Control Circuits', *ACS Synthetic Biology*, 2014, **3**, (8), pp. 600–616
36 Rosenfeld, N., Elowitz, M.B., Alon, U.: 'Negative Autoregulation Speeds the Response Times of Transcription Networks', *Journal of molecular biology*, 2002, **323**, (5), pp. 785–793
37 Gardner, T.S., Cantor, C.R., Collins, J.J.: 'Construction of a Genetic Toggle Switch in Escherichia Coli', *Nature*, 2000, **403**, (6767), pp. 339
38 Abed, S., Rashid, A., Hasan, O.. 'Formal Reasoning about Synthetic Biology using Higher-order-logic Theorem Proving'. (, 2020. `http://save.seecs.nust.edu.pk/frsbholtp/`
39 Ingalls, B.P.: 'Mathematical Modeling in Systems Biology: An Introduction'. (MIT press, 2013)
40 Houpis, C.H., Sheldon, S.N.: 'Linear Control System Analysis and Design with MATLAB'. (CRC Press, 2013)
41 Shin, Y.J., Bleris, L.: 'Linear Control Theory for Gene Network Modeling', *PloS One*, 2010, **5**, (9), pp. e12785
42 Walter, C.: 'Stability of Controlled Biological Systems', *Journal of Theoretical Biology*, 1969, **23**, (1), pp. 23–38