2025 11th International Conference on Mechatronics and Robotics Engineering

Formalization of Biological Circuit Block Diagrams for formally analyzing Biomedical Control Systems in pHRI Applications

Adnan Rashid
School of Electrical Engg. and Computer Science
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
adnan.rashid@seecs.edu.pk

Saed Abed*
Computer Engg. Depertment, College of Engg. and Petroleum
Kuwait University
Kuwait City, Kuwait
s.abed@ku.edu.kw
*Corresponding Author

Osman Hasan
School of Electrical Engg. and Computer Science
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
osman.hasan@seecs.edu.pk

***Abstract*—The control of Biomedical Systems in Physical Human-Robot Interaction (pHRI) plays a pivotal role in achieving the desired behavior by ensuring the intended transfer function and stability of subsystems within the overall system. Traditionally, the control aspects of biomedical systems have been analyzed using manual proofs and computer based analysis tools. However, these approaches provide inaccurate results due to human error in manual proofs and unverified algorithms and round-off errors in computer-based tools. We argue using Interactive reasoning, or frequently called theorem proving, to analyze control systems of biomedical engineering applications, specifically in the context of Physical Human-Robot Interaction (pHRI). Our methodology involves constructing mathematical models of the control components using Higher-order Logic (HOL) and analyzing them through deductive reasoning in the HOL Light theorem prover. We propose to model these control systems in terms of their block diagram representations, which in turn utlize the corresponding differential equations and their transfer function-based representation using the Laplace Transform (LT). These formally represented block diagrams are then analyzed through logical reasoning in the trusted environment of a theorem prover to ensure the correctness of the results. For illustration, we present a real-world case study by analyzing the control system of the utrafilteration dialysis process.**

***Index Terms*—Biomedical Systems, Control Aspects, Physical Human Robot Interaction, Automated Reasoning, Laplace Transform, Deductive Reasoning.**

I. Introduction

Biomedical systems encompass a wide range of technologies and processes used in healthcare, including medical devices, diagnostic tools, monitoring equipment, and life-supporting machines, as well as processes such as dialysis or drug delivery systems. These systems interact directly with the human body and are often responsible for life-sustaining functions, making their reliability and safety paramount. Given that even minor malfunctions or inaccuracies in these systems can lead to severe consequences, such as incorrect diagnoses, improper treatment, or even loss of life, ensuring their correctness is critical. Biomedical systems must adhere to stringent safety and performance standards to minimize risks to patients. The complexity of these systems, often involving continuous data processing, real-time decision-making, and interactions with biological processes, further emphasizes the need for rigorous verification. Any failure in the design or operation of biomedical systems can compromise patient safety, highlighting the importance of thorough testing, validation, and formal verification methods to guarantee their dependability and effectiveness in clinical settings.

Nowadays principles from engineering, mathematics, and physics have been extensively employed to analyze biomedical systems, such as prosthetic eyes, artificial limbs, pacemakers, dental implants, and dialysis machines. In these systems, control components are crucial for achieving desired behaviors by regulating the functioning of different modules, ensuring smooth operation of the entire system. For instance, automatic anesthesia controllers administer drugs to patients to prevent overdosing, and automatic control of hemodialysis offers better treatment for end-stage renal disease patients. Similarly, control components in medical robots assist in rehabilitating disabled patients.

Analyzing these biomedical systems requires modeling their dynamics to understand interactions between different system modules. These dynamic behaviors are typically represented using differential equations that capture relationships between various parameters, such as input signals and corresponding system responses. The Laplace Transform (LT) is subsequently applied to transform these differential equations from the time

This work was supported and funded by Kuwait University, Research Project No. (EO 01/23)

979-8-3315-0928-6/25/$31.00 ©2025 IEEE

domain into their frequency-domain equivalents. These models form the basis for performing Transfer Function (TF) and stability analyses of the control components in biomedical systems.

Historically, the control aspects of biomedical systems have been analyzed using manual proofs and computer-based symbolic and numerical methods [1]. However, manual approaches are prone to human error, particularly when handling large and complex systems. There's also a risk of overlooking critical assumptions necessary for accurate mathematical analysis, potentially leading to erroneous results. Computer-based methods rely on tools like Maple, MATLAB, and Mathematica, which may contain unverified algorithms or produce incorrect results due to unproven symbolic procedures or numerical approximations.

The increasing use of robotic systems in biomedical engineering, especially in applications involving Physical Human-Robot Interaction (pHRI), necessitates rigorous verification techniques to ensure safety and reliability. pHRI systems, which involve direct or indirect physical contact between humans and robots, are especially prone to safety risks. Therefore, ensuring the correct operation of control systems in these applications is crucial. Therefore, given the safety-critical nature of these biomedical systems, the above-mentioned conventional approaches cannot be relied upon.

Formal verification techniques [2] are computer-assisted mathematical methods commonly used to rigorously model and analyze complex systems [3], [4]. One of the most widely adopted formal methods is Interactive Theorem Proving (ITP) [2], which employs suitable logics—such as propositional, predicate, or higher-order logic—to build mathematical models of systems. These models are then verified against their intended specifications through deduction and reasoning within a theorem prover, ensuring the soundness and completeness of the analysis. While ITP has been applied to Laplace Transform-based analyses in various engineering domains, including automated vehicle platoons [5], linear analog filters [6], unmanned submersible vehicles [7], image processing filters [8], cyber physical systems [9], to our knowledge, it has not yet been utilized for formal verification of control systems in biomedical engineering, particularly in pHRI.

This paper introduces a formal analysis approach based on interactive theorem proving for analyzing control aspects in biomedical systems. Specifically, we propose to develop formal models capturing the dynamics of control components using differential equations in higher-order logic. We then transform these equations into their frequency-domain representations using the Laplace Transform. These transformed representations are subsequently used to conduct transfer function (TF) and stability analyses of the control components within the HOL Light theorem prover. In this step, we require the block diagrams representations that allow us to establish a relationship of time-domain dynamics of these systems with their corresponding frequency domain representations. The block diagrams facilitate the breakdown of complex biomedical control systems into simpler, analyzable components, enabling systematic verification and control design.

We introduced a formalization approach using higher-order logic for block diagram representations of biological circuits in [10]. This framework was applied to rigorously analyze protein activation and repression, as well as autoactivation, along with phase lead and lag controllers. These components are commonly employed in cancer cell identification and multi-input receptors for accurate disease diagnosis. We developed a similar formalization of block diagrams in this current paper but the focus of this block diagram formalization is for analyzing the control aspects of biomedical systems. So despite being block diagram representations the context is totally disjoint. We chose HOL Light for our formalization because of its comprehensive libraries for multivariate calculus and LT. For illustration purposes, we formally verify the Ultrafiltration dialysis process, which is a process of removing a volume of excess fluid from a patient by separating the small particles and macromolecules from the body fluid water.

II. Preliminaries

A. HOL Light

Theorem proving is a formal verification method used to verify mathematical theorems through computer programs known as theorem provers. This approach involves creating a formal model of a system and verifying its properties using higher-order logic, which offers greater expressiveness by allowing additional quantifiers. The undecidable nature of the underlying higher-order logic leads to the involvement of significant human interaction for proof construction, making these tools interactive theorem provers. They are particularly suitable for mathematical analysis based on multivariate calculus. HOL Light is a widely-used interactive theorem prover, implemented in OCaml [11], and designed to automate mathematical proofs. It has a minimal trusted core (about 400 lines of OCaml code), which has been verified through the CakeML project [12].

B. Multivariable Calculus Theories of HOL Light

HOL Light offers extensive support for system analysis through its multivariable calculus theories, like integral [1], derivatives [2], vectors [3], transcendental [4] and topology [5]. All theorems in these theories are formally verified for generic functions, i.e., functions with data type $\mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}^{\mathbb{M}}$. Some of these functions that are directly used in our proposed methodology for the formal verification of control aspects in biomedical systems are described below.

Definition 1. Integral of a Vector-valued Function
$\vdash_{def}$ ∀g x. integ x g =

[1] https://github.com/jrh13/hol-light/blob/master/Multivariate/integration.ml

[2] https://github.com/jrh13/hol-light/blob/master/Multivariate/derivatives.ml

[3] https://github.com/jrh13/hol-light/blob/master/Multivariate/vectors.ml

[4] https://github.com/jrh13/hol-light/blob/master/Multivariate/transcendentals.ml

[5] https://github.com/jrh13/hol-light/blob/master/Multivariate/topology.ml

```
          (@i.(g has_integ i) x)
```

The above-mentioned function returns the integral of g in the region of integration x: $\mathbb{R}^\mathbb{N} \rightarrow \mathbb{B}$, which represents a vector space. The function has_integ represents the integral relationship in relational form. The Hilbert choice operator @: $(\alpha \rightarrow$ bool$) \rightarrow \alpha$ provides the value of the integral, if it exists.

Definition 2. Derivative of a Vector-valued Function

```
⊢def ∀g n. vec_der g n =
     (@g'.(g has_vec_der f') n)
```

The above function provides the differential of the function g at a point n: $\mathbb{R}^N \rightarrow \mathbb{B}$. It returns a vector of type $\mathbb{R}^\mathbb{M}$, which represents the derivative of g at n.

C. Laplace transform Theories of HOL Light

The LT of a vector-valued continuous-time function is expressed as [13]:

$$\mathcal{L}[f(t)] = F(s) = \int_0^\infty f(t)e^{-st}dt, \quad s \in \mathbb{C} \qquad (1)$$

Here, s represents a complex variable. Additionally, $\mathbb{C}$ represents the data type of a complex number, which corresponds to the data type $\mathbb{R}^2$ in HOL Light, essentially a column matrix with two elements. The LT (Equation (1)) has been formalized in HOL Lightas detailed in [13]:

Definition 3. Laplace Transform

```
⊢def ∀s g. ltfm g s =
      integ {x| &0 ≤ x}
          (λx. cexp (--(s * Cx x)) * g x)
```

where cexp : $\mathbb{C} \rightarrow \mathbb{C}$ models the complex-valued exponential function. Similarly, t represents the HOL Light function drop t that converts a one-dimensional vector t to a real number, and the operator typecasts a natural number into a real number. Similarly, t | 0 ≤ t defines the positive real line, representing the integration limit from 0 to ∞. In the above definition, the function ltfm returns the LT of f as per Equation (1) by utilizing the vector function integral.

The Laplace transform exists for piecewise smooth functions as well as the functions for which the exponential order is on the positive real line. These conditions are met if the given function is piecewise differentiable within the given interval. We formally captured this as follows [13]:

Definition 4. Existence of the Laplace Transform

```
⊢def ∀s g. lexst g s ⇔
  (∀b. g pcws_diff_on
                         interval [&0,b]) ∧
  (∃M a. Re s > a ∧ eord g M a)
```

The function Re returns the real part of its input complex number. The function eord models the exponential-order condition to ensure that the LT exists and its formalization is detailed in [13], [14]:

Definition 5. Exp Order

```
⊢def ∀g M a. eord g M a ⇔
 &0 < M ∧
 (∀t. &0 ≤ t ⇒ ||g t|| ≤ M * exp (a * t))
```

Here, $||\vec{x}||$ denotes the norm of the vector. Likewise, $\bar{t}$ refers to the HOL Light function lift t, which converts a real number t into a one-dimensional vector.

We utilized the formal definitions of the Laplace transform and its existence condition to verify the commonly used properties such as linearity, time and frequency shifting, time scaling, differentiation and integration in time domain in [15].

III. Formalization of Block Diagram Representations

In biomedical systems involving human-robot interactions (pHRI), biological functions are often governed by intricate interactions between various physiological components. Block diagrams help break down these systems into distinct functional units or modules, each representing specific processes, such as fluid regulation, filtration, or control mechanisms in human-robot collaboration.

We now introduce our formal definitions for the essential building blocks (subsystems) of these block diagram representations. These definitions allow for the formal modeling of block diagrams for any biomedical system or process in the s-domain and enable the determination of the transfer function from its block diagram. The proposed formalization is primarily inspired by the block diagrams used in control systems [16] and our previous formal modeling of block diagram representations [10].

Serial Configuration The transfer function for the series connection of two components (subsystems), as shown in Figure 1, in a biomedical system, such as an ultrafiltration dialysis process—representing fluid transfer between different parts of the patient like the arm, trunk, or leg, can be expressed as the multiplication of the individual transfer functions as shown in Figure 1. We model in a generic way ((*N*) components) as follows:

Definition 6. Series Configuration

$\vdash_{def} \forall \mathsf{A_i}.\ \mathsf{ser}\ [\mathsf{A_1}; \mathsf{A_2}; ...; \mathsf{A_N}] = \prod_{\mathsf{i=1}}^{\mathsf{N}} \mathsf{A_i}$

The above-mentioned function returns the overall transfer function of an arbitrary number of block connected in series, which it takes their individual transfer functions as input in the form of a list. The net transfer function is calculated by multiplying all the individual transfer functions.

Summation Junction It acts as an addition module, as illustrated in Figure 2, that adds the transfer functions of all the individual components.

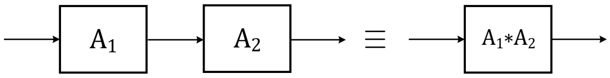

Fig. 1: Series Configuration

Definition 7. Summation
$\vdash_{def} \forall A_i.$ summ $[A_1; A_2; ...; A_N] = \sum_{i=1}^{N} A_i$

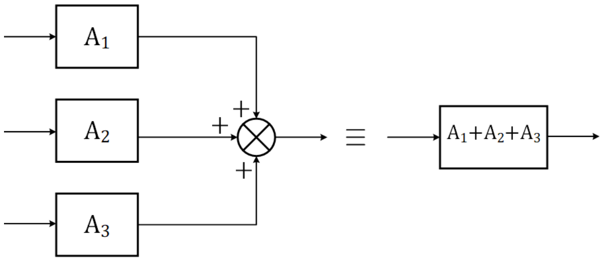

Fig. 2: Summation Junction

Pickoff The pickoff point configuration represents the branching of a component into parallel components (Figure 3). We formally model this as follows:

Definition 8. Pickoff
$\vdash_{def} \forall \alpha\ A_i.$ pick_point $[A_1; A_2; ...; A_N] =$
$[\alpha * A_1; \alpha * A_2; ...; \alpha * A_N]$

The above-mentioned function returns the equivalent transfer functions arranged as a list of components by using the transfer functions of the first and the rest of the components, represented as a list of complex numbers.

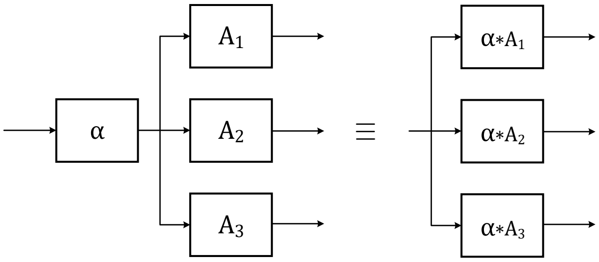

Fig. 3: Pickoff Point

Feedback The feedback configuration serves as the essential representation for modeling the closed-loop response (Figure 4). Owing to the inclusion of the feedback signal, this configuration is predominantly expressed as an infinite summation of branches comprising of serially connected components. We formally model this as:

Definition 9. Branch
$\vdash_{def} \forall \alpha\ \beta$ n. branch $\alpha\ \beta$ n =
$\prod_{i=0}^{n}$ series_comp $[\alpha; \beta]$

The above function returns a complex number representing the transfer function of the n^{th} branch by processing the forward path transfer function α, the feedback signal, the transfer function β and the number of the branches (n).

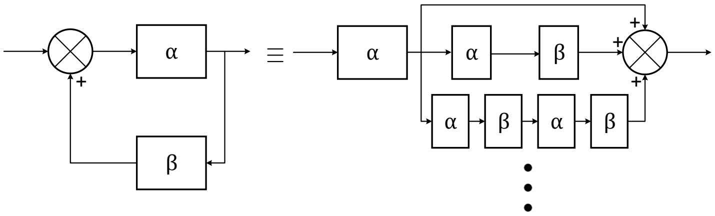

Fig. 4: Feedback

Now, just like the other blocks, we formally model the feedback block as follows:

Definition 10. Feedback
$\vdash_{def} \forall \alpha\ \beta.$ feedback_block $\alpha\ \beta$ =
series_comp $[\alpha; \sum_{k=0}^{\infty}$ branch $\alpha\ \beta$ k$]$

The above function accepts the forward (α) and the feedback path (β) transfer functions. It calculates the overall transfer function by creating a series network that combines the final forward path transfer function with the summation of all potential infinite branches.

IV. Formal Verification of the Ultrafiltration Dialysis Process

Ultrafiltration [17], [18] is a process used in kidney dialysis to remove excess fluid and waste products from the blood, particularly for patients with kidney failure or chronic kidney disease. It mimics the function of the kidneys by filtering blood through a semi-permeable membrane. It allows a passage of fluid across a semipermeable (allowing some substances) membrane that is driven by a pressure gradient between the blood and effluent compartments. Since ultrafiltration ensures a desirable level of body fluid by removing its excessive amount during dialysis so its formal verification is of utmost importance for ensuring the correct functionality.

Figure 5 provides a block (flow) diagram of the ultrafiltration dialysis process for a kidney patient. It is composed of three modules, such as arms, trunk and legs. The volumes of fluid in each of the modules arms, trunk and legs are represented by $V_A(t)$, $V_T(t)$ and $V_L(t)$, respectively. Moreover, the quantity of fluid transferred between these modules are controlled by the transfer constants k_{TA} (between trunk and arms), k_{TL} (between trunk and legs), k_A (between arms and trunk) and k_L (between legs and trunk). The quantity of the fluid removed from the trunk is represented by $UFR(t)$.

The dynamical model of the fluid transfer between arms and trunk is mathematically expressed as:

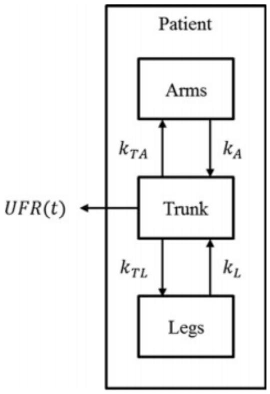

Fig. 5: Ultrafiltration Dialysis Process

$$\frac{dV_A(t)}{dt} = -k_A V_A(t) + k_{TA} V_T(t) \tag{2}$$

In order to model the dynamics of the fluid transfer, we formalize the generic linear differential equation as follows:

Definition 11. Linear Differential Equation of Order n
$\vdash_{def}$ `∀k f t. diff_eq_nord n lst f t =`
`vsum (0..n) (λi. EL i [` α_1 `;` α_2 `;...;` α_i `] *`
`higher_order_derivative i f t)`

The above function takes the order of the diff. equation `n`, the coefficients list `l`, a function `f` and its variable of differentiation `t`. It outputs the n^{th} order differential equation.

Now, we formalize the dynamics of fluid transfer (Equation (2)) as follows:

Definition 12. Dynamics of the Fluid Transfer between Arms and Trunk
$\vdash_{def}$ `∀kA. olst_diff_eq_at kA = [-- Cx kA;`
`Cx (&1)]`
$\vdash_{def}$ `∀kTA. ilst_diff_eq_at kTA = [Cx kTA]`
$\vdash_{def}$ `diff_eq_at VT VA t kA kTA ⇔`
`diff_eq_nord 1 (olst_diff_eq_at kA) VA t`
`=`
`diff_eq_nord 0 (ilst_diff_eq_at kTA) VT`
`t`

We now model the corresponding block diagram representation using our formalization in HOL Light as:

Definition 13. Block Diagram Representation of Fluid Transfer between Arms and Trunk
$\vdash_{def}$ `∀kA kTA.`
`blk_diag_rep_at kA kTA = ser`
$\left[\texttt{Cx kTA;}\ \frac{\texttt{Cx(\&1)}}{\texttt{s + Cx kA}}\right]$

Next we verify the overall transfer function for this system as follows:

Theorem 1. Fluid Transfer between Arms and Trunk
$\vdash_{thm}$ `∀kA kTA s. [A]: s + Cx kA ≠ Cx (&0)`
`⇒ blk_diag_rep_at kA kTA =`
$\frac{\texttt{Cx\&1}}{\texttt{s + Cx kA}}$

We now formally verify the transfer function derived from Theorem 1, grounded in the dynamical model, as follows:

Theorem 2. Model Implies Transfer Function
$\vdash_{thm}$ `∀kA kTA VT VA s.`
`A1: &0 < kA ∧ A2: &0 < kTA ∧`
`A3: ∀t. diff_hderivative VT VA t ∧`
`A4: lexst_hderivative VT VA ∧`
`A5: zero_init_conditions VT VA ∧`
`A6: (∀t. diff_eq_at VT VA t kA kTA) ∧`
`A7: (ltfm VT s ≠ Cx (&0)) ∧`
$[A_8]: \left(\frac{\texttt{Cx(\&1)}}{\texttt{s + Cx kA}} \neq \texttt{Cx(\&0)}\right)$
$\Rightarrow \frac{\texttt{ltfm VA s}}{\texttt{ltfm VT s}} = \frac{\texttt{Cx(\&1)}}{\texttt{s + Cx kA}}$

Assumptions `A1-A2` capture the positivity constraints on the transfer constants `kTA` and `kA`. The next 2 assumptions specify the prerequisites for the existance of the first derivative and Laplace transform of `VA` and `VT`, respectively. Assumption `A5` establishes the initial initial conditions for both VA and VT. Assumption `A6` describes the dynamic behavior of fluid transfer between the arms and trunk. The next 2 assumptions validate the denominator of the transfer function presented in the theorem's conclusion. Ultimately, the conclusion presents the transfer function for the fluid transfer between the arms and trunk.

The theorem-proving-based analysis presented in this work offers significant advantages, making it a rigorous and reliable method for analyzing complex systems like ultrafiltration dialysis. The main strength of the proposed methodology is its ability to explicitly specify all conditions underlying the analysis. Without these assumptions, it would be impossible to validate the correctness of a theorem. This strict requirement guarantees that every proven theorem is sound, making the analysis highly robust and trustworthy. Additionally, the theorems are universally quantified, meaning they are generalizable and can be specialized for any set of values. This flexibility allows the results to be adapted to a wide range of scenarios and configurations, offering great utility in practical applications.

Another key advantage of our approach is its ability to handle the continuities inherent in many real-world systems, such as fluid transfer in ultrafiltration dialysis. Other formal methods, such as SAT-based methods or state exploration based model checking approach, struggle to accurately capture continuous behaviors due to their reliance on discrete-state models. Consequently, theorem proving emerges as a superior formal method for analyzing continuous systems, providing a more accurate representation of the system dynamics.

However, one drawback of this method is the human-interactive nature of the proof process. Theorem proving often requires significant manual intervention, which can be time-consuming and cumbersome, particularly for complex proofs. While this human interaction ensures rigor and flexibility, it also introduces a barrier to efficiency. Fortunately, tactics can be developed to automate certain parts of the proof process, as demonstrated by the automatic tactic TF_TAC_UF used in this work. These tactics can streamline some of the more repetitive or standard aspects of the proofs, reducing the burden on the user while maintaining the overall integrity and soundness of the analysis.

Thus, the proposed theorem-proving approach offers a powerful and reliable method for analyzing biomedical systems, providing both rigor and flexibility unmatched by other formal methods, though it requires ongoing efforts to improve automation.

V. Conclusions

We provide a formal modelling of block diagram representations of biological circuits, specifically for the analysis of biomedical control systems in pHRI. By utilizing higher-order logic (HOL) theorem proving, we have successfully formalized the components and interactions within these systems, providing a systematic and rigorous method for verifying their behavior. This approach is particularly valuable for systems involving complex physiological processes, as it allows for precise analysis and ensures correctness in critical applications.

One key application of this methodology is demonstrated in the formal analysis of the ultrafiltration dialysis process. By modeling the fluid dynamics and transfer functions between different body compartments using block diagrams, we provided a comprehensive framework to verify the system's functionality. The formalization of the dialysis process showcases the practical importance of our approach, ensuring the correct functionality of this life-saving procedure.

References

[1] J Fernández, C Galindo, J Barbacho, and A Luque. Automatic Control Systems in Biomedical Engineering, 2018.

[2] O. Hasan and S. Tahar. Formal Verification Methods. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 7162–7170. IGI Global, 2015.

[3] S. M. Beillahi, U. Siddique, and S. Tahar. Formal Analysis of Power Electronic Systems. In *International Conference on Formal Engineering Methods*, pages 270–286. Springer, 2015.

[4] U. Siddique, V. Aravantinos, and S. Tahar. Formal Stability Analysis of Optical Resonators. In *NASA Formal Methods*, pages 368–382. Springer, 2013.

[5] A. Rashid, U. Siddique, and O. Hasan. Formal Verification of Platoon Control Strategies. In *Software Engineering and Formal Methods*, pages 223–238. Springer, 2018.

[6] S. H. Taqdees and O. Hasan. Formally Verifying Transfer Functions of Linear Analog Circuits. *IEEE Design & Test*, 34(5):30–37, 2017.

[7] A. Rashid and O. Hasan. Formal Analysis of Linear Control Systems using Theorem Proving. In *Int. Conf. on Formal Engineering Methods*, volume 10610 of *LNCS*, pages 345–361. Springer, 2017.

[8] A. Rashid, S. Abed, and O. Hasan. Formal Analysis of 2D Image Processing Filters using Higher-order-logic Theorem Proving. *Journal on Advances in Signal Processing*, 53, 2022.

[9] A. Rashid and O. Hasan. Formal Analysis of the Continuous Dynamics of Cyber-physical Systems using Theorem Proving. *Journal of System Architecture*, 112, 2021.

[10] S. Abed, A. Rashid, and O. Hasan. Formal Reasoning about Synthetic Biology using Higher-order-logic Theorem Proving. *Systems Biology*, 14(5):271–283, 2020.

[11] M. Abrams and et al. A History of OCaml. http://ocaml.org/learn/history.html, 2015.

[12] J. Harrison. Towards self-verification of HOL Light. In *Int. Joint Conf. on Automated Reasoning*, pages 177–191. Springer, 2006.

[13] A. Rashid and O. Hasan. Formalization of Lerch's Theorem using HOL Light. *Journal of Applied Logics—IFCoLog Journal of Logics and their Applications*, 5(8):1623–1652, 2018.

[14] S. H. Taqdees and O. Hasan. Formalization of Laplace Transform Using the Multivariable Calculus Theory of HOL-Light. In *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 8312 of *LNCS*, pages 744–758. Springer, 2013.

[15] A. Rashid. Formal Analysis of the Continuous Dynamics of Cyber-physical Systems using Theorem Proving. http://save.seecs.nust.edu.pk/projects/facdcpstp/, 2018.

[16] C. H. Houpis and S. N. Sheldon. *Linear Control System Analysis and Design with MATLAB*. CRC Press, 2013.

[17] W. L Henrich. *Principles and Practice of Dialysis*. Lippincott Williams & Wilkins, 2012.

[18] J. T Daugirdas, P. G Blake, and Todd S. *Handbook of Dialysis*. Lippincott Williams & Wilkins, 2012.