Formally Analyzing Continuous Aspects of Cyber-Physical Systems modeled by Homogeneous Linear Differential Equations

Muhammad Usman Sanwal[1] and Osman Hasan[2]

[1]Computational Biomodeling Laboratory
Turku Centre for Computer Science and Department of Computer Science
Abo Akademi University, Turku, Finland
muhammad.sanwal@abo.fi
[2]School of Electrical Engineering and Computer Science
National University of Sciences and Technology (NUST), Islamabad, Pakistan
osman.hasan@seecs.nust.edu.pk

Abstract. Traditionally, the continuous aspects of cyber-physical systems (CPS), usually modeled by differential equations, are analyzed using paper-and-pencil proof methods, computer based numerical methods or computer algebra systems. All these methods are error-prone and thus the analysis cannot be termed as accurate, which poses a serious threat to the accuracy of the cyber-physical systems. To guarantee the correctness of analysis, we propose to use higher-order-logic theorem proving to reason about the correctness of solutions of differential equations. This paper presents a formalization framework to express homogeneous linear differential equation of arbitrary order and formally verify their solutions within the sound core of the higher-order-logic theorem prover HOL4. In order to illustrate the usefulness of the proposed formalization, we utilize it to formally verify a couple of CPS used in the domain of bio-medicine, namely, a heart pacemaker and a fluid-filled catheter.

1 Introduction

Cyber-physical systems (CPS) [26] are characterized as computational systems, with software and digital and/or analog hardware components, that closely interact with their continuously changing physical surroundings. These days, CPS are widely being used and advocated to be used in a variety of applications ranging from ubiquitous consumer electronic devices, such as tele-operated health-care units and autonomous vehicles, to not so commonly used but safety-critical domains, such as tele-surgical robotics, space-travel and smart disaster response and evacuation. Due to the tight market windows or safety-critical nature of their applications, it has become a dire need to design error-free CPS and thus a significant amount of time is spent on ensuring the correctness of CPS designs.

Traditionally, physical and continuous aspects of a CPS are analyzed by capturing their behaviors by appropriate differential equations [33] and then solving these differential equations to obtain the required design constraints. This kind

of analysis can be done using paper-and-pencil proof methods or computer based numerical techniques. Whereas, the software and digital hardware components of a CPS are usually analyzed using computer based testing or simulation methods, where the main idea is to deduce the validity of a property by observing its behavior for some test cases. However, all the above mentioned analysis techniques, i.e., paper-and-pencil proof methods, numerical methods and simulation, cannot ascertain the absence of design flaws in a design. For example, paper-and-pencil proof methods are error prone due to the human error factor. Moreover, it is quite often the case that many key assumptions of the results obtained using paper-and-pencil proof methods are in the mind of the mathematician and are not documented. Such missing assumptions may also lead to erroneous CPS designs. Similarly, computer based numerical methods cannot attain 100% accuracy as well due to the memory and computation limitations and round-off errors introduced by the usage of computer arithmetics. Thus, given the above mentioned inaccuracies, these traditional techniques should not be relied upon for the analysis of CPS, especially when they are used in safety-critical areas, such as medicine and transportation, where inaccuracies in the analysis could result in design bugs that in turn may even lead to the loss of human life.

In the past couple of decades, formal methods [5] have been successfully used for the precise analysis of a variety of software, hardware and physical systems. The main principle behind formal analysis of a system is to construct a computer based mathematical model of the given system and formally verify, within a computer, that this model meets rigorous specifications of intended behavior. Two of the most commonly used formal verification methods are model checking [4] and higher-order-logic theorem proving [16]. Model checking is an automatic verification approach for systems that can be expressed as a finite-state machine. Higher-order-logic theorem proving, on the other hand, is an interactive approach but is more flexible in terms of tackling a variety of systems. The rigorous exercise of developing a mathematical model for the given system and analyzing this model using mathematical reasoning usually increases the chances for catching subtle but critical design errors that are often ignored by traditional techniques like paper-and-pencil based proofs or simulation.

Given the extensive usage of CPS in safety-critical applications, there is a dire need of using formal methods for their analysis. However, the frequent involvement of ordinary differential equations (ODEs) in their analysis is a main limiting factor in this direction. For example, ODEs are essential for modeling the motion of mechanical parts, analog circuits and control systems, which are some of the most common elements of any CPS. Thus, automatic state-based formal methods, like model checking, and automatic theorem provers cannot be used to model and analyze the true CPS models due to their inability to model continuous systems. This is the main reason why most of the formal verification work about CPS utilizes their abstracted discrete models (e.g., [30]). Hybrid model-checking and theorem proving based approaches, e.g, [1], have been generally used for analyzing systems that can be modeled as differential equations. Moreover, safety properties of such systems have also been formally verified using

differential invariants [24, 3] based on fixed point algorithms. Similarly, the Coq theorem prover has been used to formally verify the convergence of numerical solutions for a widely used partial differential wave equation [6]. Other notable higher-order-logic formalizations related to differential equations include verification of the convergence of numerical solutions for differential equations [6] and the approximate numerical solution of ordinary differential equations using the one-step method [19]. However, to the best of our knowledge, none of these formal approaches allow us to verify the solutions of differential equations.

These limitations can be overcome by using higher-order-logic theorem proving [13] for conducting the formal analysis of CPS since the high expressiveness of higher-order logic can be leveraged upon to model elements of continuous nature. However, the main challenge in this direction is the enormous human guidance required in the formal verification of CPS due to the non-decidable nature of higher-order logic. As a first step towards using a higher-order-logic theorem prover for formally verifying solutions of differential equations, we presented the formal reasoning support for the solutions of second-order homogeneous linear differential equations [23], i.e., a simple yet the most widely used class of differential equations, in [27]. In the current paper, we extend this work by presenting a formal definition that can be used to specify arbitrary order homogeneous linear differential equations. Moreover, we provide the formal verification of some mathematical facts, like a couple of general solutions of arbitrary order homogeneous linear differential equations and the quadratic formula, that allow us to reason about the correctness of the solutions of arbitrary order homogeneous linear differential equations in a very straightforward way. The prime advantage of these results is that they greatly minimize the user intervention for formal reasoning about differential equations and thus facilitate the usage of higher-order-logic theorem proving for verifying the solutions of differential equations for real-world industrial problems. In order to demonstrate the practical effectiveness and utilization of our formalization, we utilize it to analyze two CPS used in biomedical applications, i.e, a heart pacemaker and fluid-filled catheter.

Our formalization primarily builds upon the higher-order-logic formalization of the derivative function and its associated properties. This formalization is available in a number of theorem provers like HOL4 [14], PVS [8] and Coq [10]. Our work is based on Harrison's formalization [14] that is available in the HOL4 theorem prover [29]. The main motivations behind this choice is include the availability of formalized transcendental functions, which play a key role in the reported work, and the general richness of Harrison's real analysis related theories. Though, it is important to note here that the ideas presented in this paper are not specific to the HOL4 theorem prover and can be adapted to any other LCF style higher-order-logic theorem prover as well.

2 Related Work

Formal methods have been extensively used these days for analyzing CPS due to their ever increasing usage in various safety and financial-critical domains.

Zhang et. al [21] proposed to use formal specification for CPS in order to reduce the infinite set of test parameters in a finite set. Similarly, the aspect-oriented programming based on the UML and formal methods is utilized for QoS modeling of CPS in [20]. Moreover, in order to formally specify CPS along with their continuous aspects, a combination of formal methods Timed-CSP, ZimOO and differential (algebraic) equations is used in [32]. Even though such rigorous formal specifications allow us to catch bugs in the early stages of the design but they do not guarantee error-free analysis due to the informal nature of the analysis.

For formal verification of CPS, model-checking has been frequently explored. For example, Akella [2] proposed to discretize the events causing the change of flow and thus model the CPS as a deterministic state model with discrete values of flow within its physical components. This model is then used to formally verify insecure interactions between all possible behaviors of the given CPS using model checking. Similarly, Bu et. al [7] used hybrid model checking for verifying CPS. However, this verification is also not based on true continuous models of the system and instead a short-run behavior of the model is observed by providing numerical values of various parameters in order to reduce the state-space. A statistical model checker has been recently utilized to analyze some aspects of CPS [9]. However, this approach also suffers from the classical model checking issues, like the state-space explosion and inability to reason about generic mathematical relations. Thus the model checking approach, even though is capable of providing exact solutions, is quite limited in terms of handling true continuous models of CPS and thus various abstractions [30] have to be used for attaining meaningful results. The accuracy of the analysis is thus compromised, which is undesirable in the case of analyzing safety-critical applications of CPS.

Hybrid theorem provers, like KeYmaera [25], have also been used to verify CPS. However, these theorem provers use the support of computer algebra systems when it comes to solving differential equations and thus the solutions obtained cannot be completely trusted due to the presence of unverified symbolic manipulation algorithms in computer algebra systems.

Higher-order-logic theorem proving is capable of overcoming all the above mentioned problems. Atif et. al [22] used the HOL4 theorem prover for the probabilistic analysis of cyber-physical transportation systems. However, their focus was only on the formal verification of probabilistic aspects of CPS and they did not tackle the continuous aspects, especially the ones that require to be modeled by ODEs, which is the main focus of the current paper.

3 Derivatives in HOL4

In this section, we give a brief introduction to the formalization of the derivative function in HOL4 function to facilitate the understanding of the rest of the paper. Harrison [14] formalized the *real number theory* along with the fundamentals of calculus, such as real sequences, summation series, limits of a function and derivatives and verified most of their classical properties in HOL4. The limit of

a function f, which takes a real number and returns a real number, is defined in HOL4 using the operator $\rightarrow$ as follows [14]:

Definition 1: `⊢ ∀ f y0 x0. (f → y0)(x0) = ∀e. 0 < e ⇒`
`∃d. 0 < d ∧ ∀x. 0 < |x - x0| ∧ |x - x0| < d ⇒`
`|f(x) - y0| < e`

where `(f → y0)(x0)` can be written mathematically as $lim_{(x \rightarrow x0)} f(x) = y0$, i.e., the function `f` approaches `y0` as its real number argument approaches `x0`. Based on this definition, the derivative of a function f is defined as follows [14]:

Definition 2: `⊢ ∀ f l x.(f diffl l) x =`
`((λ h.(f (x + h) - f x) / h)→ l)(0)`

Definition 2 provides the derivative of a function f at point x as the limit value of $\frac{f(x+h)-f(x)}{h}$ when h approaches 0, which is the standard mathematical definition of the derivative function. Now, the differentiability of a function f is defined as the existence of its derivative [14].

Definition 3: `⊢ ∀ f x. f differentiable x = ∃l. (f diffl l) (x)`

A functional form of the derivative, which can be used as a binder, is also defined using the Hilbert choice operator `@`as follows [14]:

Definition 4: `⊢ ∀ f x. deriv f x = @l. (f diffl l) x`

The function `deriv` accepts two parameters f and x and returns the derivative of function f at point x.

The above mentioned definitions associated with the derivative function have been accompanied by the formal verification of most of their classical properties, such as uniqueness, linearity and composition [14]. Moreover, the derivatives of some commonly used transcendental functions have also been verified. For example, the derivative of the Exponential function has been verified as follows:

Theorem 1: `⊢ ∀ g m x. ((g diffl m) x ⇒`
`((λx. exp (g x)) diffl (exp (g x) * m)) x)`

where `exp x` represents the exponential function e^x and $(\lambda \mathtt{x.f(x)})$ represents the lambda abstraction function which accepts a variable x and returns $f(x)$. We build upon the above mentioned formalization to develop formal reasoning support for homogeneous linear differential equations and the details of our work are given in the next two sections.

4 Homogeneous Linear Differential Equations

An n$^{\texttt{th}}$-order homogeneous linear ordinary differential equation can be mathematically expressed as follows:

$$p_n(x)\frac{d^n y(x)}{dx} + p_{n-1}(x)\frac{d^{n-1} y(x)}{dx} + \cdots + p_0(x)y(x) = 0 \tag{1}$$

where $\frac{d^i f}{dx}$ denotes the i^{th} ordinary derivative of the function f with respect to variable x and terms $p_i(x)$ represent the coefficients of the differential equation defined over a function y. The equation is linear because (i) the function y and its derivatives appear only in their first power and (ii) the products of y with its derivatives are also not present in the equation. By finding the solution of the above equation, we mean to find functions that can be used to replace the function y in Equation (1) and satisfy it.

The first step in the proposed reasoning support framework is the ability to formalize homogeneous linear differential equation. We proceed in this direction by first formalizing an n$^{\texttt{th}}$-order derivative function as follows:

Definition 5: ⊢ (∀ f x. n_order_deriv 0 f x = f x) ∧
(∀f x n. n_order_deriv (n+1) f x = n_order_deriv n (deriv f x) x)

The function `n_order_deriv` accepts a positive integer n that represents the order of the derivative, the function f that represents the function that needs to be differentiated, and the variable x that is the variable with respect to which we want to differentiate the function f. It returns the n$^{\texttt{th}}$-order derivative of f with respect to x. Now, based on this definition, we can formalize the left-hand-side (LHS) of Equation (1) in HOL4 as the following definition.

Definition 6: ⊢ ∀ P y x. diff_eq_lhs P y x =
sum (0,LENGTH P) (λn. (EL n P) x * (n_order_deriv n y x))

The function `diff_eq_lhs` accepts a list P of coefficient functions corresponding to the p_i's of Equation (1), the differentiable function y, the order of differentiation n and the differentiation variable x. It utilizes the HOL4 functions `sum (0,n) f`, `EL n L` and `LENGTH L`, which correspond to the summation ($\sum_{i=0}^{n-1} f_i$), the n$^{\texttt{th}}$ element of a list L_n, and the length of a list $|L|$, respectively. It generates the LHS of a differential equation of `LENGTH(P)`$^{\texttt{th}}$ order with coefficient list P. It is important to note that the order of the differential equation has been inferred from the number of its coefficients in the above definition.

The linearity property of derivatives play a very important role in our work. We verified this property for *class* C^n functions, i.e., the functions for which the first n derivatives exist for all x as the following higher-order-logic theorem:

Theorem 2: ⊢ ∀ f g x a b.
(∀m x. m ≤ n⇒(λx. n_order_deriv m f x)differentiable x) ∧

```
(∀m x. m ≤ n⇒(λx. n_order_deriv m g x)differentiable x) ⇒
  (n_order_deriv n (λx. a * f x + b * g x) x =
    a * n_order_deriv n f x + b * n_order_deriv n g x)
```

where variables a and b represent constants with respect to variable x. The formal reasoning about Theorem 2 involves induction on variable n, which represents the order of differentiation, and is primarily based on the linearity property of the first order derivative function [14].

5 Solution of Homogeneous Linear Differential Equations

It is a well-known mathematical fact that if $y_1(x), y_2(x), \cdots, y_n(x)$ are independent solutions of Equation (1) then their linear combination

$$Y(x) = c_1 y_1(x) + c_2 y_2(x) + \cdots + c_n y_n(x) \tag{2}$$

also forms a solution of Equation (1), where $c_1, c_2, \cdots, c_n$ are arbitrary constants [33]. This result plays a vital role in solving differential equations as it allows us to find the solution of a differential equation if its n independent solutions are known. A particular case of interest arises when the coefficients p_i's of Equation (1) are constants in terms of the differentiation variable x. In this case, using the fact that the derivative of the exponential function $y = e^{rx}$ (with a constant r) is a constant multiple of itself $dy/dx = re^{rx}$, we can obtain the following solution:

$$Y(x) = c_1 e^{r_1 x} + c_2 e^{r_2 x} + \cdots + c_n e^{r_n x} \tag{3}$$

where $c_1, c_2, \cdots, c_n$ are arbitrary constants and $r_1, r_2, \cdots, r_n$ are the real and distinct roots of the characteristic equation

$$p_n r^n + p_{n-1} r^{n-1} + \cdots + p_0 = 0 \tag{4}$$

with constant p_i's [33]. The above mentioned results play a key role in solving homogeneous linear order differential equations and the main focus of this paper is the formal verification of these results, which in turn would facilitate formal reasoning about the correctness of solutions of differential equations in a higher-order-logic theorem prover.

We verified the first property, corresponding to Equation (2), as follows:

Theorem 3: ⊢ ∀ Y C P x.

```
(n_order_differentiable_fn_list Y (LENGTH P)) ∧
(n_order_diff_eq_soln_list Y P x) ⇒
  (diff_eq_lhs P (λx. linear_sol C Y x) x = 0)
```

where Y represents the list of solutions $y_1(x), y_2(x), \cdots, y_n(x)$ of the given differential equation, C represents the list of arbitrary constants $c_1, c_2, \cdots, c_n$, P represents the list of functions corresponding to the coefficients $p_1(x), p_2(x), \cdots, p_n(x)$ of the differential equation and x is the variable of differentiation. The first predicate in the assumptions of Theorem 3, i.e, `n_order_differentiable_fn_list`,

ensures that each element of the list Y is n^{th}-order differentiable, where n ranges from 0 to `LENGTH P`. It is defined in HOL4 recursively as follows:

Definition 7:
```
⊢ (∀ n. n_order_differentiable_fn_list [] n = True) ∧
  ∀ h t n. n_order_differentiable_fn_list (h::t) n =
  (∀m x. m ≤ n⇒(λx. n_order_deriv m h x)differentiable x) ∧
                        n_order_differentiable_fn_list t n
```

where `::` represents the list *cons* operator in HOL4.

The second predicate in the assumptions of Theorem 3, i.e., `n_order_diff_eq_soln_list`, ensures that each element of the list Y is a solution of the given differential equation with coefficients P. This predicate is recursively defined in HOL4 as:

Definition 8:
```
⊢ (∀ P x. n_order_diff_eq_soln_list [] P x = True) ∧
   ∀ h t P x. n_order_diff_eq_soln_list (h::t) P x =
   (diff_eq_lhs P h x = 0) ∧ n_order_diff_eq_soln_list t L x
```

Finally the function `linear_sol`, used in the conclusion of Theorem 3, models the linear solution represented by Equation (2) using the lists of solution functions Y and arbitrary constants C as follows:

Definition 9:
```
⊢ (∀ C x. linear_sol C [] x = 0) ∧
   ∀ C h t x. linear_sol C (h::t) x =
     EL (LENGTH C - LENGTH (h::t)) C * h x + linear_sol C t x
```

The recursive variable of Definition 9 is instantiated with the list Y in Theorem 3 and the expression `EL (LENGTH C - LENGTH (h::t)) C` picks the corresponding constant from list C for each y_i. Thus, using the functions `linear_sol` and `diff_eq_lhs`, we have formally verified the intended property in Theorem 3.

We verified Theorem 3 by performing induction on the the variable Y. The proof is primarily based on the linearity properties of the n^{th}-order derivative, (Theorem 2) and the summation function along with arithmetic reasoning.

The second property of interest, described using Equation (3), can be expressed in HOL4 as the following theorem:

Theorem 4:
```
⊢ ∀ C P R x.(∀n. n < LENGTH R ⇒ EL n R <> r) ∧
    (ch_eq_roots_list R (const_fn_list P) x) ⇒
    (diff_eq_lhs (const_fn_list P)
                (λx. linear_sol C (exp_list R) x) x = 0)
```

where C represents the list of arbitrary constants $c_1, c_2, \cdots, c_n$, P represents the list of constants corresponding to the coefficients $p_1, p_2, \cdots, p_n$ of the differential equation, R represents the list of roots $r_1, r_2, \cdots, r_n$ of the characteristic equation, given in Equation (4), and x is the variable of differentiation. The function `const_fn_list` used in the above theorem transforms a constant list to the corresponding constant function list recursively as follows:

Definition 10: ⊢ `(const_fn_list [] = []) ∧`
`(∀h t.const_fn_list (h::t) = (λ(x:real). h)::(const_fn_list t))`

The function `diff_eq_lhs` permits coefficients that are functions of the variable of differentiation but Theorem 4 is valid only for constant coefficients. Thus, using `const_fn_list` we provide the required type for the coefficient list of the function `diff_eq_lhs` while fulfilling the requirement of Theorem 4.

The assumption predicate, i.e, `ch_eq_roots_list`, recursively ensures that each element of the list R is a valid root of the characteristic equation, like the one given in Equation (4), with constant coefficients given by list P:

Definition 11: ⊢ `∀ P r x. ch_eq_root P r x =`
`(sum(0,LENGTH P)(λn.((EL n P x)) * (r pow n)) = 0) ∧`
`(∀ P x. ch_eq_roots_list [] P x = True) ∧`
`(∀ h t P x. ch_eq_roots_list (h::t) P x =`
`(ch_eq_root P h x) ∧ (ch_eq_roots_list t P x))`

The first function `ch_eq_root` ensures that its argument r is a valid root of the characteristic equation formed by coefficients given in list P. The function `ch_eq_roots_list` recursively calls function `ch_eq_root` for each entry of the looping variable and thus ensures that all the entries of the looping list are valid roots of the characteristic equation formed by coefficients given in list P.

Finally, the function `exp_list` is used in Theorem 4 to model a list of exponential functions that are used to form the solution of the main differential equation, like the one given in Equation (3). This function is defined as follows:

Definition 12: ⊢ `(exp_list [] = []) ∧`
`(exp_list (h::t) = (λx. exp (h * x)) :: (exp_list t))`

It is important to note that the function linear_sol is used to express the conclusion of Theorem 4 as has been then case for Theorem 3. This way, the formally verified result of Theorem 3 can be used in formally verifying Theorem 4. The formal reasoning about Theorem 4 is conducted by performing induction on variable Y and the reasoning is primarily based on Theorem 4 and the following lemma that allows us to express the left-hand-side of the step case of Theorem 4 in terms of real summation.

Lemma 1: ⊢ `∀ P h x. (diff_eq_lhs P (λx. exp (h * x)) x =`
`(exp (h * x) * (sum (0,LENGTH P) (λn. EL n P x * h pow n))))`

Now, If the roots of an characteristic equation are real and repeated then the solution of Equation (1) can be written as

$$Y(x) = c_1 e^{rx} + c_2 x e^{rx} + \cdots + c_n x^{n-1} e^{rx} \tag{5}$$

where $c_1, c_2, \cdots, c_n$ are arbitrary constants and r is the real and repeated root of the characteristic equation given below

$$p_n r^n + p_{n-1} r^{n-1} + \cdots + p_0 = 0 \tag{6}$$

The solution of Equation (1) , described using Equation (5), can be expressed in HOL4 as the following theorem:

Theorem 5: ⊢ ∀ C R r. (∀n. n < LENGTH R ⇒ EL n R = r) ∧
(∀m. m < LENGTH R ⇒ (diff_eq_lhs
(const_fn_list C) (λx. x pow m * exp (r * x)) x =0))
⇒ (diff_eq_lhs (const_fn_list C)
(λx. linear_sol C (polynomial_function R) x) x = 0)

Where C and R are lists of constants and roots, respectively.

The assumptions of Theorem 5 ensure that the roots of the characteristic equation are the same and equal to r and $e^{rx}, xe^{rx}, x^2e^{rx}, \cdots, x^{LENGTH\ R\ -1}e^{rx}$ are all solutions of the given differential equation. The conclusion of the theorem specifies that Equation 5 is a solution of the given differential equation using the functions `polynomial_function` and `linear_sol`. The function `linear_sol` is given in Definition 9 and the `polynomial_function` is defined as follows:

Definition 13: ⊢ (polynomial_function [] = []) ∧
(polynomial_function (h::t) =
(λx.(x pow (LENGTH t))*exp(h*x))::(polynomial_function t))

The formal reasoning about Theorem 5 is conducted by performing induction on variable R and the reasoning is primarily based on Theorem 3 and the following lemma that tells us that all derivatives of exponential with multiple of increasing power of x are differentiable.

Lemma 2: ⊢ ∀ R n h x. (λx. n_order_deriv n
(λx. x pow LENGTH R * exp(h * x)) x) differentiable x

Besides the above mentioned key results, we also verified the famous quadratic formula, which plays a vital role in reasoning about the characteristic equations of second degree and also provides some support for reasoning about characteristic equations of higher order. The quadratic formula is verified as follows:

Theorem 6: ⊢ ∀ a b c x. (a ≠ 0) ∧ (4 * a * c < b pow 2) ⇒
ch_eq_roots_list [((-b + sqrt (b pow 2 - 4 * a * c)) /
(2 * a));((-b - sqrt (b pow 2 - 4 * a * c)) / (2 * a))]
(const_fn_list [a; b; c]) x

where the functions `sqrt` and `pow` represent the square-root and square of a real number, respectively. The theorem essentially says that the roots of the characteristic equation $ax^2 + bx + c$ are given by the first list argument of the function `ch_eq_roots_list`. The assumption `(4 * a * c < b pow 2)` guarantees that the roots are always real.

The main benefit of the formalization presented above is that now building upon these results the formal verification of solutions of homogeneous linear differential equations would be done almost automatically as will be illustrated in the next section. It is worth while to point out that the major effort in our development was spent in finalizing the formal nomenclature, presented in the form of definitions in this paper, to represent homogeneous linear differential equations. The generic nature of these definitions allows us to represent almost all kinds of homogeneous linear differential equations. The formal verification of the theorems described in this section required human guidance but the simplifiers were a great help in this process. Our HOL proof script [28] is composed of over 1200 lines of code and the verification took about 300 man-hours.

6 Biomedical Applications

Biomedical applications are one of the most safety-critical applications of CPS as their bugs could eventually result in the loss of human lives. Differential equations form the core foundation of modeling almost all biomedical applications [11]. Due to a lack of formal reasoning support for differential equation solutions, most of the analysis of CPS used in biomedical applications with continuous components is done using informal analysis techniques so far. Our work tends to fill this gap and thus facilitates the usage of formal methods in this safety-critical domain. We present two case studies, i.e., the analysis of a heart pacemaker and a fluid-filled catheter, to illustrate the usefulness and effectiveness of our work.

6.1 Heart Pacemaker

Electronic heart pacemakers are widely used for supplementing or replacing heart's natural pacing mechanism in humans. The pacemaker specification has been proposed as a pilot project for the Verified Software Initiative [17].

Their main working principle is to store electrical energy in a capacitor and then discharge this energy in short pulses through the heart to provide it with the required sudden electrical stimulus. Besides the capacitor, they include a battery, which provides the energy source, and a switch to govern the charging and discharging of the capacitor. Figure 1 illustrates the connections between these main components and their working [11]. The capacitor is charged via the battery when the switch S is moved to position A, while the capacitor provides the short and intense pulses to the heart when the switch S is in position B.

Based on Figure 1, the behavior of an electronic heart pacemaker can be described in terms of the following differential equation [11]:

$$\frac{dV}{dt} + \frac{1}{RC}V = 0,\ V(0) = E \tag{7}$$

since the current through the capacitor (CdV/dt) equals the current through the heart (V/R), which behaves as a resistor R, when the switch S is in position B. Moreover, the capacitor is allowed to charge to its full capacity when the switch is

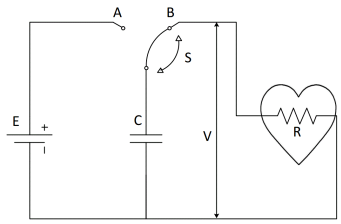

Fig. 1: Equivalent Circuit of an Electronic Pacemaker

in position A and thus we obtain the initial condition $V(0) = E$. This simplistic but realistic mathematical model of a heart pacemaker has been extensively used in the literature to analyze the underlying properties of interest (See e.g., [31, 11]. In this paper, we utilize our formalization described in the previous two sections to formally reason about the solution of Equation (7). We proceed by specifying the theorem stating the solution ($Ee^{-\frac{t}{RC}}$) of Equation (7) as follows:

Theorem 7: $\vdash \forall$ `R C C1 V E t.`

```
(diff_eq_lhs (const_fn_list [(1/(R*C)); 1])
(λx.linear_sol [C1] (exp_list [-(1/(R*C))]) x) t = 0)
```

The initial condition $V(0) = C1$ is implicitly contained in the above theorem as it is satisfied for the case $t = 0$. Thus, the theorem provides the general solution of the given differential equation and the value of the constant C for the particular solution.

Our formalized definitions facilitated the formal specification of the above theorem and the formally verified Theorem 4 allowed us to verify the above theorem in a few reasoning steps where we just had to provide the definitions of the functions used in Theorem 6 and some primitive list theory functions, like `EL` and `LENGTH`, along with invoking an automatic arithmetic simplifier. The straightforward reasoning process about the correctness of solution of the given differential equation in the sound environment of HOL4 clearly demonstrates the effectiveness of our work.

6.2 Fluid-Filled Catheter

As a second case study of our work, consider the dynamic analysis of a fluid-filled catheter, which allows physicians to measure the pressure of the internal organs and fluids of a human body without inserting a pressure transducer in the body. The main idea is to insert a long and small-bore fluid-filled tube or catheter in the body and thus bring the pressure of the pressure measuring site outside and then use a conventional pressure transducer to measure it. However, mechanical parameters like the mass of the catheter fluid and the friction of this fluid with the catheter wall may introduce some discrepancies in the pressure measurements. Therefore, it is very important to analyze the effects of such

mechanical parameters on the pressure measurements as a wrong reading may endanger a patient's life. A number of studies, e.g. [18, 12], have analyzed this aspect by considering the following second-order linear differential equation:

$$\frac{1}{\omega_n^2}\frac{d^2p}{dt^2} + \frac{2\zeta}{\omega_n}\frac{dp}{dt} + p = 0 \tag{8}$$

where p is the applied pressure, $\omega_n = \sqrt{k/\rho LA}$ represents the undamped natural angular frequency (radians per unit time) in terms of a constant k, catheter fluid density ρ, length L and cross-sectional area A, and $\zeta = c/2\sqrt{1/\rho kLA}$ is the damping factor with a constant c. Equation (8) allows us to find the pressure in response to any force function given that the coefficients ω_n and ζ are known. The solution of this equation can be formally verified as the following theorem:

Theorem 8: ⊢ ∀ rho A L k c C1 C2.
```
(sqrt(4 * rho * L * A * k)<c ∧ 0<rho ∧ 0<L ∧ 0<A ∧ 0<k ⇒
(diff_eq_lhs (const_fn_list
            [k / (rho * L * A); c / (rho * L * A); 1])
            (λx. linear_sol [C1; C2]
(exp_list [(-(c / (rho * L * A)) +
          sqrt ((c / (rho * L * A)) pow 2
        - 4 * (k / (rho * L * A)))) / 2;(-(c / (rho * L * A)) -
            sqrt ((c / (rho * L * A)) pow 2 -
            4 * (k / (rho * L * A)))) / 2]) x) x = 0))
```

The assumptions of the above theorem declare the relationships between the various parameters that are required for the solution to hold. This is one of strengths of the proposed theorem proving based verification as all the assumptions have to be explicitly stated besides the theorem for its formal verification. Thus, there is no chance of missing a critical assumption which often occurs in paper-and-pencil proof methods where there is no such guarantee that the mathematician who worked out the proof has written down all the assumptions.

Formal reasoning about Theorem 8 is primarily based on Theorems 4 and 6 along with some arithmetic rewriting, which can be done in an automatic manner using the HOL arithmetic simplifiers. The straightforward proof scripts for of Theorems 7 and 8 clearly indicate the usefulness of our foundational formalization presented in Sections 4 and 5 of this paper. Just like these case studies our formalization results can be utilized to formally reason about solution of any homogeneous linear differential equation and the results would be guaranteed to be correct due to the inherent soundness of theorem proving.

7 Conclusions

In this paper, we propose to use higher-order-logic theorem proving to analyze continuous aspects of CPS. Due to the high expressiveness of the underlying logic, we can formally model the continuous components of CPS while capturing

their true behavior and the soundness of theorem proving guarantees correctness of results. To the best of our knowledge, these features are not shared by any other existing CPS analysis technique. The main challenge in the proposed approach is the enormous amount of user intervention required due to the undecidable nature of the logic. We propose to overcome this limitation by formalizing the foundational mathematical theories so that these available results can be built upon to minimize user interaction. As a first step towards this direction, we presented the formalization of the solutions of any homogeneous linear differential equation in this paper. Based on this work, we are able to formally analyze the CPS used in a couple of biomedical systems.

The proposed approach opens the doors to many new directions of research. We are working on developing reasoning support for non-homogeneous linear differential equations. Moreover, the calculus theories available in HOL-Light [15] are based on multivariate real numbers and thus can model complex numbers. Our formalization can be ported in a very straight-forward manner to this formalization of complex numbers in HOL-Light, which would enable handling the formal analysis of CPS that can be modeled in the complex plane only.

References

1. E. Abraham-Mumm, M. Steffen, and U. Hannemann. Verification of Hybrid Systems: Formalization and Proof Rules in PVS. In *ICECCS*, pages 48–57, 2001.
2. R. Akella and B.M. McMillin. Model-Checking BNDC Properties in Cyber-Physical Systems. In *Computer Software and Applications Conference*, pages 660 –663, 2009.
3. A.Platzer and E.M. Clarke. Computing Differential Invariants of Hybrid Systems as Fixedpoints. *Formal Methods in System Design*, 35(1):98–120, 2009.
4. C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.
5. P.P. Boca, J.P. Bowen, and J.I. Siddiqi. *Formal Methods: State of the Art and New Directions*. Springer, 2009.
6. S. Boldo, F. Clment, J. Fillitre, M. Mayero, G. Melquiond, and P. Weis. Formal proof of a Wave Equation resolution Scheme: The Method Error. In *Interactive Theorem Proving*, volume 6127 of *LNCS*, pages 147–162. Springer, 2010.
7. L. Bu, Q. Wang, X. Chen, L. Wang, T. Zhang, J. Zhao, and X. Li. Towards Online Hybrid Systems Model Checking of Cyber-Physical Systems' Time-Bounded Short-Run Behavior. *SIGBED*, (2):7–10, 2011.
8. R. W. Butler. Formalization of the Integral Calculus in the PVS Theorem Prover. *Journal of Formalized Reasoning*, 2(1):1–26, 2009.
9. E. M. Clarke and P. Zuliani. Statistical model checking for cyber-physical systems. In *Automated Technology for Verification and Analysis*, volume 6996 of *LNCS*, pages 1–12. Springer, 2011.
10. L. Cruz-Filipe. *Constructive Real Analysis: a Type-Theoretical Formalization and Applications*. PhD thesis, University of Nijmegen, April 2004.
11. S. A. Glantz. *Mathematics for Biomedical Applications*. University of California Press, 1979.
12. S.A. Glantz and J. V. Tyberg. Determination of Frequency Response from Step Response: Application to Fluid-Filled Catheters. *The American Journal of Physiology*, 236:376–378, 1979.

13. M.J.C. Gordon. Mechanizing Programming Logics in Higher-0rder Logic. In *Current Trends in Hardware Verification and Automated Theorem Proving*, pages 387–439. Springer, 1989.
14. J. Harrison. *Theorem Proving with the Real Numbers*. Springer-Verlag, 1998.
15. J. Harrison. A HOL theory of Euclidean space. In *Theorem Proving in Higher Order Logics*, volume 3603 of *LNCS*, pages 114–129. Springer, 2005.
16. J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.
17. C.A.R. Hoare, J. Misra, G. T. Leavens, and N. Shankar. The Veried Software Initiative: A Manifesto. *ACM Comput. Survey*, 41(4):1–8, 2009.
18. J.O. Hougen, S.T. Hougen, and T.J. Hougen. Dynamics of Fluid-Filled Catheter Systems by Pulse Testing. In *Ind. Eng. Chem. Fundam*, volume 25, pages 462–470, 1986.
19. F. Immler and J. Holzl. Numerical Analysis of Ordinary Differential Equations in Isabelle/HOL. In *Interactive Theorem Proving*, volume 7406 of *LNCS*, pages 377–392. Springer, 2012.
20. J. Liu and L. Zhang. QoS Modeling for Cyber-Physical Systems using Aspect-Oriented Approach. In *Networking and Distributed Computing (ICNDC), 2011 Second International Conference on*, pages 154 –158, 2011.
21. L.Zhang, J. Hu, and W . Yu. Generating Test Cases for Cyber Physical Systems from Formal Specification. pages 97–103. Springerl, 2011.
22. A. Mashkoor and O. Hasan. Formal Probabilistic Analysis of Cyber-Physical Transportation Systems. In *ICCSA (3)*, pages 419–434, 2012.
23. K. Oduola, I. Sofimieari, and P. Nwambo. A Method for Solving Higher Order homogeneous Ordinary Differential Equations with Non-constant Coefficients. *Journal of Emerging Trends in Engineering and Applied Sciences*, 2(1):7–10, 2011.
24. A. Platzer. Differential dynamic Logics for Hybrid Systems. *Journal of Automated Reasoning*, 41(2):143–189, 2008.
25. A. Platzer and J. Quesel. KeYmaera: A Hybrid Theorem Prover for Hybrid Systems (System Description). In *Automated Reasoning*, volume 5195 of *Lecture Notes in Computer Science*, pages 171–178. Springer, 2008.
26. R. Rajkumar, I. Lee, L. Sha, and J. J. Stankovic. Cyber-Physical Systems: The next Computing Revolution. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 731 –736, 2010.
27. M. U. Sanwal and O. Hasan. Formal Verification of Cyber-Physical Systems: Coping with Continuous Elements. In *Computational Science and Its Applications - Part I*, volume 7971 of *LNCS*, pages 358–371. Springer, 2013.
28. M.U. Sanwal. Formal Reasoning about Homogeneous Linear Differential Equations. http://save.seecs.nust.edu.pk/students/usman/lde.html, 2013.
29. K. Slind and M. Norrish. A Brief Overview of HOL4. In *Theorem Proving in Higher-order Logics*, volume 5170 of *LNCS*, pages 28–32. Springer, 2008.
30. R. A. Thacker, K. R. Jones, C. J. Myers, and H. Zheng. Automatic Abstraction for Verification of Cyber-Physical Systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 12–21. ACM, 2010.
31. H. Zhang, J.H. Liu, and A.V. Holden. Computing the Age-Related Dysfunction of Cardiac Pacemaker. In *Computers in Cardiology*, volume 33, pages 665–668, 2006.
32. L. Zhang. Aspect Oriented Formal Techniques for Cyber Physical Systems. *journal of software*, 7(4):823–834, 2012.
33. D.G. Zill, W.S. Wright, and M.R. Cullen. *Advanced Engineering Mathematics*. Jones and Bartlett Learning, fourth edition, 2009.