Formal Analysis of Continuous-time Systems using Fourier Transform

Adnan Rashid

School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Islamabad, Pakistan

Osman Hasan

School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Islamabad, Pakistan

Abstract

To study the dynamical behaviour of the engineering and physical systems, we often need to capture their continuous behaviour, which is modeled using differential equations, and perform the frequency-domain analysis of these systems. Traditionally, Fourier transform methods are used to perform this frequency domain analysis using paper-and-pencil based analytical techniques or computer simulations. However, both of these methods are error prone and thus are not suitable for analyzing systems used in safety-critical domains, like medicine and transportation. In order to provide an accurate alternative, we propose to use higher-order-logic theorem proving to conduct the frequency domain analysis of these systems. For this purpose, the paper presents a higher-order-logic formalization of Fourier transform using the HOL-Light theorem prover. In particular, we use the higher-order-logic based formalizations of differential, integral, transcendental and topological theories of multivariable calculus to formally define Fourier transform and reason about the correctness of its classical properties, such as existence, linearity, time shifting, frequency shifting, modulation, time scaling, time reversal and differentiation in time domain, and its relationships with Fourier Cosine, Fourier Sine and Laplace transforms. We use our proposed formalization for the formal verification of the frequency response of a generic n-order linear system, an audio equalizer and a MEMs accelerometer, using the HOL-Light theorem prover.

Key words: Frequency Response, Continuous-time Systems, Theorem Proving, Higher-order Logic, HOL-Light, Fourier Transform.

Email addresses: `adnan.rashid@seecs.nust.edu.pk` (Adnan Rashid),
`osman.hasan@seecs.nust.edu.pk` (Osman Hasan).

1. Introduction

Fourier Transform (Bracewell, 1978) is a transform method, which converts a time varying function to its corresponding ω-domain representation, where ω is its corresponding angular frequency (Beerends et al., 2003). This transformation allows replacing the differentiation and integration in time domain analysis to multiplication and division operators in the frequency domain, which can be easily manipulated. Moreover, the ω-domain representations of the differential equations can also be used for the frequency response analysis of the corresponding systems. Due to these distinguishing features, Fourier transform has been widely used for analyzing many continuous-time systems, such as signal (Papoulis, 1977; Gaydecki, 2004), and image (Dougherty, 2009) processing algorithms, analog circuits (Thomas et al., 2016), communication systems (Ziemer and Tranter, 2006; Du and Swamy, 2010), medical sciences (Bracewell, 1978; Dougherty, 2009), mechanical systems (Oppenheim et al., 1996) and optics (Gaskill, 1978; Stark, 2012).

The first step in the Fourier transform based analysis of a continuous-time system, is to model the dynamics of the system using a differential equation. This differential equation is then transformed to its equivalent ω-domain representation by using the Fourier transform. Next, the resulting ω-domain equation is simplified using various Fourier transform properties, such as existence, linearity, frequency shifting, modulation, time shifting, time scaling, time reversal and differentiation. The main objective of this simplification is to either solve the differential equation to obtain values for the variable ω or obtain the frequency response of the system corresponding to the given differential equation. The frequency response can in turn be used to analyze the dynamics of the system by studying the impact of different frequency components on the intended behaviour of the given system. The information sought from this analysis plays a vital role in analyzing reliable and performance efficient engineering systems.

Traditionally, the analysis of continuous-time systems, using transform methods, has been done using the paper-and-pencil based analytical technique. However, due to the highly involved human manipulation, the analysis process is error prone, especially when dealing with larger systems, and hence an accurate analysis cannot be guaranteed. Moreover, this kind of manual manipulation does not guarantee that each and every assumption required in the mathematical analysis is written down with the analysis. Thus, some vital assumptions may not accompany the final result of the analysis and a system designed based on such a result may lead to bugs later on. For example, the Air France Flight 447 crashed in 2009, which resulted in 228 deaths, was attributed to the faulty warning system consisting of speed sensors. These sensors gave wrong/invalid reading about the speed of the airplane, which led to the crash. A more rigourous analysis of the warning system could have prevented this incident.

Computer-based methods including the numerical methods and the symbolic techniques, provide a more scalable option to analyze larger systems. Some of the computer tools involved in these analysis are MATLAB (MATLAB, 2017), Mathematica (Wolfram, 2015) and Maple (Maple, 2017). The numerical analysis involves the approximation of the continuous expressions or the continuous values of the variables due to the finite precision of computer arithmetic, which compromises the accuracy of the analysis. Moreover, it involves a finite number of iterations, depending on the computational resources and computer memory, to judge the values of unknown continuous parameters, which introduces further inaccuracies in the analysis as well. Similarly, the symbolic tools cannot

assure absolute accuracy as they involve discretization of integral to summation while evaluating the improper integral in the definition of Fourier transform (Taqdees and Hasan, 2013). Moreover, they also contain some unverified symbolic algorithms in their core (Durán et al., 2013), which puts another question mark on the accuracy of the results. Given the widespread usage of the continuous-time systems in many safety-critical domains, such as medicine and transportation, we cannot rely on these above-mentioned analysis methods as the analysis errors could lead to disastrous consequences, including the loss of human lives.

Formal methods (Hasan and Tahar, 2015) are computer based mathematical techniques that involve the mathematical modeling of the given system and the formal verification of its intended behaviour as a mathematically specified property, which is expressed in an appropriate logic. This verification of the properties of the underlying system is based on mathematical reasoning. Moreover, the mathematical nature of the system model and the desired property guarantees the accuracy of formal analysis. Formal methods have been widely used for the verification of software (Schumann, 2001) and hardware (A. Camilleri and Melham, 1986) systems and the formalization (or mathematical modeling) of classical mathematics (Hales, 2005; Avigad and Harrison, 2014).

Higher-order-logic theorem proving (Harrison, 2009) is a widely-used formal verification method that has been extensively used to completely analyze continuous systems by leveraging on the high expressiveness of higher-order logic and the soundness of theorem proving. *Umair et al.* formalized the Z-transform (Siddique et al., 2014) and used them to analyze an Infinite Impulse Response (IIR) filter. Similarly, *Hira et al.* formalized the Laplace transform (Taqdees and Hasan, 2013) and used their formalization to analyze a Linear Transfer Converter (LTC) circuit. However, the formalization of Z-transform can only be utilized for the discrete-time system analysis. On the other hand, the formalization of Laplace transform can be used to reason about the solutions of ordinary differential equations and the transfer function analysis of the continuous-time systems (Taqdees and Hasan, 2013), but is only limited to causal functions, i.e., the functions that fulfill the condition: $f(x) = 0$ for all $x < 0$. However, many physical and engineering systems exhibit the non-causal continuous behaviors, involving functions with infinite extent. For example, in optics, the optical image of a point source of light may be described theoretically by a Gaussian function of the form e^{-x^2}, which exists for all x (Goodman, 2005). Another example is the rate of flow of water out of a tap at the bottom of a bucket of water, which can be modeled using e^{-kt}, where t ranges over the whole real line (Thibos, 2013). Fourier transform can cater for the analysis involving both continuous and non-causal functions and thus can overcome the above-mentioned limitations of Z and Laplace transforms.

With the objective of extending the scope of the analysis based on theorem proving, to cover non-causal functions, we present a higher-order-logic based formalization of Fourier transform in this paper. In particular, we formalize the definition of Fourier transform in higher-order logic and use it to verify the classical properties of Fourier transform, such as existence, linearity, time shifting, frequency shifting, modulation, time scaling, time reversal, differentiation, and its relations to Fourier Cosine, Fourier Sine and Laplace transforms. These foundations can be built upon to reason about the analytical solutions of differential equations or frequency responses of the physical systems, as depicted in Fig. 1. The user of the proposed formal analysis framework is required to develop a formal model of the given system, by using its corresponding differential equation. Similarly, the

desired frequency response behavior from the given system can be captured as a proof goal (theorem) that can be formed using this behavior along with the formal model of the given system. The proof goal can then be verified based on the above-mentioned formalization of Fourier transform within the sound core of the HOL-Light theorem prover, which is an interactive theorem proving environment for conducting proofs in higher-order logic. The availability of the above-mentioned formally verified properties decreases the manual user interaction and thus effort required while performing the formal Fourier transform based analysis of a system. Besides the above-mentioned foundational formalization of Fourier transform, we also use these results to verify a relationship of frequency response of a generic n-order system (Adams, 2012). This generic relationship can be specialized to facilitate the reasoning process of the formal frequency response analysis of any specific order system. In order to illustrate the practical utilization of the proposed formalization, we present a formal analysis of an audio equalizer (Adams, 2012) and a MEMs accelerometer (Kaajakari et al., 2009), which are extensively used in communication systems and many safety critical systems, respectively. We use the HOL-Light theorem prover (Harrison, 1996b) for the proposed formalization in order to build upon its comprehensive reasoning support for multivariable calculus. Particularly, the proposed formalization heavily relies upon the formalization of differential, integration, topological and transcendental theories of multivariable calculus.
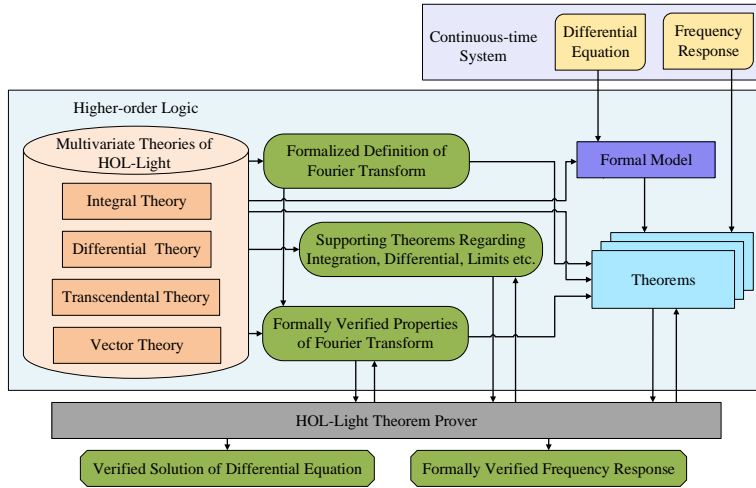

Fig. 1. Proposed Framework

The rest of the paper is organized as follows: Section 2 provides a brief introduction about the HOL-Light theorem prover and the multivariable calculus theories of HOL-Light. Section 3 presents the formalization of the Fourier transform definition and the conditions required for its existence. We provide the verification of the classical properties of Fourier transform in Section 4. Section 5 presents the Fourier transforms of some commonly used functions. We present the formal analysis of generic n-order system in Section 6. Section 7 provides the verification of the frequency response of an audio equalizer and a MEMs accelerometer. Section 8 presents the discussion, which highlights upon the main challenges faced in the proposed formalization. Finally, Section 9 concludes the paper.

4

2. Preliminaries

In this section, we present an introduction to the HOL-Light theorem prover and an overview about the multivariable calculus theories of HOL-Light, which provide the foundational support for the proposed formalization.

2.1. HOL-Light Theorem Prover

HOL-Light (Harrison, 1996b) is an interactive theorem proving environment for conducting proofs in higher-order logic. The logic in the HOL-Light system is represented in the strongly-typed functional programming language ML (Paulson, 1996). A theorem is a formalized statement that may be an axiom or could be deduced from already verified theorems by an inference rule. A theorem consists of a finite set Ω of Boolean terms, called the assumptions, and a Boolean term S, called the conclusion. Soundness is assured as every new theorem must be verified by applying the basic axioms and primitive inference rules or any other previously verified theorems/inference rules. A HOL-Light theory is a collection of valid HOL-Light types, constants, axioms, definitions and theorems. Various mathematical foundational concepts have been formalized and saved as HOL-Light theories. The HOL-Light theorem prover provides an extensive support of theorems regarding, Boolean algebra, arithmetic, real numbers, transcendental functions and multivariate analysis such as differential, integration, vectors and topology, in the form of theories, which are extensively used in our formalization. In fact, one of the primary reasons to chose the HOL-Light theorem prover for the proposed formalization was the presence of an extensive support of multivariable calculus theories. There are many automatic proof procedures and proof assistants (Harrison, 1996a) available in HOL-Light, which help the user in concluding a proof more efficiently.

Table 1 presents the standard and HOL-Light representations and the meanings of some commonly used symbols in this paper.

Table 1. HOL-Light Symbols

HOL-Light Symbols	Standard Symbols	Meanings
$/\backslash$	and	Logical *and*
$\backslash/$	or	Logical *or*
$\sim$	not	Logical *negation*
==>	$\longrightarrow$	Implication
<=>	=	Equality in Boolean domain
!x.t	$\forall x.t$	For all $x : t$
?x.t	$\exists x.t$	There exists $x : t$
λx.t	$\lambda x.t$	Function that maps x to $t(x)$
num	$\{0, 1, 2, \ldots\}$	Positive Integers data type
real	All Real numbers	Real data type
SUC n	$(n+1)$	Successor of natural number
&a	$\mathbb{N} \to \mathbb{R}$	Typecasting from Integers to Reals
abs x	$|x|$	Absolute function
EL n l	*element*	n^{th} element of list l

5

A N-dimensional vector is represented as a $\mathbb{R}^{\mathbb{N}}$ column matrix with each of its element as a real number in HOL-Light (Harrison, 2013). All of the vector operations are thus performed using matrix manipulations. A complex number is defined as a 2-dimensional vector, i.e., a $\mathbb{R}^2$ column matrix. All of the multivariable calculus theorems are verified in HOL-Light for functions with an arbitrary data-type $\mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{M}}$.

Some of the frequently used HOL-Light functions in our work are explained below:

Definition 1. *Cx and ii*
$\vdash \forall$ `a. Cx a = complex (a, &0)`
$\vdash$ `ii = complex (&0, &1)`

`Cx` is a type casting function from real ($\mathbb{R}$) to complex ($\mathbb{R}^2$). It accepts a real number and returns its corresponding complex number with the imaginary part equal to zero, where the `&` operator type casts a natural number ($\mathbb{N}$) to its corresponding real number ($\mathbb{R}$). Similarly, `ii` (iota) represents a complex number having the real part equal to zero and the magnitude of the imaginary part equal to 1.

Definition 2. *Re, Im, lift and drop*
$\vdash \forall$ `z. Re z = z$1`
$\vdash \forall$ `z. Im z = z$2`
$\vdash \forall$ `x. lift x = (lambda i. x)`
$\vdash \forall$ `x. drop x = x$1`

The function `Re` accepts a complex number and returns its real part. Here, the notation `z$i` represents the ith component of vector `z`. Similarly, `Im` takes a complex number and returns its imaginary part. The function `lift` accepts a variable of type $\mathbb{R}$ and maps it to a 1-dimensional vector with the input variable as its single component. It uses the `lambda` operator in HOL to construct a vector based on its components (Harrison, 2013). Similarly, `drop` takes a 1-dimensional vector and returns its single element as a real number. In order to make the functions `lift` and `drop` better understandable to a non-HOL user, we use $\overline{x}$ and $\underline{x}$ as the equivalent symbols for `lift x` and `drop x`, respectively.

Definition 3. *Exponential, Complex Cosine and Sine Functions*
$\vdash \forall$ `x. exp x = Re (cexp (Cx x))`
$\vdash \forall$ `z. ccos z = (cexp (ii * z) + cexp (--ii * z)) / Cx (&2)`
$\vdash \forall$ `z. csin z = (cexp (ii * z) - cexp (--ii * z)) / (Cx (&2) * ii)`

The complex exponential and real exponentials are represented as `cexp` $: \mathbb{R}^2 \to \mathbb{R}^2$ and `exp` $: \mathbb{R} \to \mathbb{R}$ in HOL-Light, respectively. Similarly, the complex cosine `ccos` and complex sine `csin` functions are formally defined in terms of `cexp` using the Euler's formula (Harisson, 2017).

Definition 4. *Vector Integral and Real Integral*
$\vdash \forall$ `f i. integral i f = (@y. (f has_integral y) i)`
$\vdash \forall$ `f i. real_integral i f = (@y. (f has_real_integral y) i)`

The function `integral` represents the vector integral and is defined using the Hilbert choice operator `@` in the functional form. It takes the integrand function `f`, having an arbitrary type $\mathbb{R}^N \to \mathbb{R}^M$, and a vector-space `i` : $\mathbb{R}^N \to \mathbb{B}$, which defines the region of convergence as $\mathbb{B}$ represents the boolean data type, and returns a vector $\mathbb{R}^M$ which is the integral of `f` on `i`. The function `has_integral` represents the same relationship in the relational form. It is a predicate, which accepts the integrand function, integral value and the region of integration, and returns `T` if the integral of the integrand function over the region of integration is equal to the integral value. Where as, the function `integral` accepts the integrand function, its region of integration and returns the value of the integral over the given region of the integration using the Hilbert choice operator `@`. Similarly, the function `real_integral` accepts the integrand function `f` : $\mathbb{R} \to \mathbb{R}$ and a set of real numbers `i` : $\mathbb{R} \to \mathbb{B}$ and returns the real-valued integral of the function `f` over `i`. The region of integration, for both of the above integrals can be defined to be bounded by a vector interval $[a, b]$ or real interval $[a, b]$ using the HOL-Light functions `interval` $[\mathtt{a}, \mathtt{b}]$ and `real_interval` $[\mathtt{a}, \mathtt{b}]$, respectively.

Definition 5. *Vector Derivative and Real Derivative*
⊢ ∀ `f net. vector_derivative f net = (@f'. (f has_vector_derivative f') net)`
⊢ ∀ `f x. real_derivative f x = (@f'. (f has_real_derivative f') (atreal x))`

The function `vector_derivative` takes a function `f` : $\mathbb{R}^1 \to \mathbb{R}^M$ and a `net` : $\mathbb{R}^1 \to \mathbb{B}$, which defines the point at which `f` has to be differentiated, and returns a vector of data-type $\mathbb{R}^M$, which represents the differential of `f` at `net`. Moreover, depending on the usage of the definition, `net` can be specified as (`at a within s`) or (`at a`), which can be a point `a` : $\mathbb{R}^1$ of a set `s` : $\mathbb{R}^1 \to \mathbb{B}$ or point `a` : $\mathbb{R}^1$, respectively, where the function `f` has to be differentiated. The function `has_vector_derivative` defines the same relationship in the relational form. Similarly, the function `real_derivative` accepts a function `f` : $\mathbb{R} \to \mathbb{R}$ and a real number `x`, which is the point at which `f` has to be differentiated, and returns a variable of data-type $\mathbb{R}$, which represents the differential of `f` at `x`. The function `has_real_derivative` defines the same relationship in the relational form.

Definition 6. *Limit of a function*
⊢ ∀ `f net. lim net f = (@l. (f → l) net)`

The function `lim` accepts a `net` with elements of arbitrary data-type $\mathbb{A}$ and a function `f` : $\mathbb{A} \to \mathbb{R}^M$ and returns `l` of data-type $\mathbb{R}^M$, i.e., the value to which `f` converges at the given `net`. Moreover, `net` can take `at_posinfinity` or `at_neginfinity` to model the positive infinity or negative infinity, respectively.

We build upon the above-mentioned fundamental functions of multivariable calculus to formalize the Fourier transform in the next section.

3. Formalization of Fourier Transform

The Fourier transform of a function $f(t)$ is mathematically defined as:

$$\mathcal{F}[f(t)] = F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt, \ \omega \ \epsilon \ \mathbb{R} \tag{1}$$

7

where f is a function from $\mathbb{R}^1 \to \mathbb{C}$ and ω is a real variable. The limit of integration is from $-\infty$ to $+\infty$. We formalize Equation (1) in HOL-Light as follows:

Definition 7. *Fourier Transform*
⊢ ∀ w f. fourier_transform f w =
                    integral UNIV (λt. cexp (--((ii * Cx w) * Cx <u>t</u>)) * f t)

The function `fourier_transform` accepts a complex-valued function `f` : $\mathbb{R}^1 \to \mathbb{R}^2$ and a real number `w` and returns a complex number that is the Fourier transform of `f` as represented by Equation 1. In the above function, we used complex exponential function `cexp` : $\mathbb{R}^2 \to \mathbb{R}^2$ because the return data-type of the function `f` is $\mathbb{R}^2$. To multiply `w` with `ii`, we first converted `w` into a complex number $\mathbb{R}^2$ using `Cx`. Similarly, the data-type of `t` is $\mathbb{R}^1$ and to multiply it with `ii * Cx w`, it is first converted into a real number <u>t</u> by using `drop` and then it is converted to data-type $\mathbb{R}^2$ using `Cx`. Next, we use the vector function `integral` (Definition 4) to integrate the expression $f(t)e^{-i\omega t}$ over the whole real line since the data-type of this expression is $\mathbb{R}^2$. Since the region of integration of the vector integral function must be a vector space therefore we represented the interval of the integral by `UNIV` : $\mathbb{R}^1$, which represents the whole real line.

The Fourier transform of a function f exists, i.e., the integrand of Equation 1 is integrable, and the integral has some converging limit value, if f is piecewise smooth and is absolutely integrable on the whole real line (Rashid and Hasan, 2016; Beerends et al., 2003; Rashid and Hasan, 2017). A function is said to be piecewise smooth on an interval if it is piecewise differentiable on that interval. The Fourier existence condition can thus be formalized in HOL-Light as follows:

Definition 8. *Fourier Exists*
⊢ ∀ f. fourier_exists f ⇔
    (∀ a b. f piecewise_differentiable_on interval [ā, b̄]) ∧
            f absolutely_integrable_on UNIV

In the above function, the first conjunct expresses the piecewise smoothness condition for the function `f`. Whereas, the second conjunct represents the condition that the function `f` is absolutely integrable on the whole real line. Next, we present a very important property of the Fourier existence as follows:

Theorem 9. Linearity of Fourier Existence
⊢ ∀ f g a b. fourier_exists f ∧ fourier_exists g ⇒
                        fourier_exists (λx. a * f x + b * g x)

where `a` : $\mathbb{C}$ and `b` : $\mathbb{C}$ are arbitrary constants acting as the scaling factors. The proof of above theorem is based on the linearity properties of integration, limit and piecewise differentiability.

4. Formal Verification of Fourier Transform Properties

In this section, we use Definitions 7 and 8 and Theorem 9 to verify some of the classical properties of Fourier transform and its relationship with various transforms, like Fourier Cosine, Fourier Sine, and Laplace transforms and Fourier transform of a nth-order differential equation in HOL-Light. The verification of these properties and

the relationships not only ensures the correctness of our definitions but also plays a vital role in minimizing the user intervention and time consumption in reasoning about Fourier transform based frequency domain analysis of continuous-time systems, as will be depicted in Section 7 of this paper.

4.1. Properties of Fourier Transform

The existence of the improper integral of Fourier Transform is a pre-condition for most of the arithmetic manipulations involving the Fourier transform. This condition is formalized in HOL-Light as the following theorem:

Theorem 10. Integrability of Integrand of Fourier Transform Integral

```
⊢ ∀ f w. fourier_exists f ⇒
          (λt. cexp (--((ii * Cx w) * Cx t)) * f t) integrable_on UNIV
```

The proof of Theorem 10 is based on splitting the region of integration, i.e., the whole real line $\mathtt{UNIV} : \mathbb{R}^1$, as a union of positive real line (interval $[0, \infty)$) and negative real line (interval $(-\infty, 0]$). Next, we split the complex-valued integrand, $f(t)e^{-i\omega t}$, into its corresponding real and imaginary parts. In this process, we need the integrability of the integrand, which can be derived by the piecewise differentiability of `fourier_exists` condition. Finally, some theorems regarding integration, integrability, continuity and some properties of the transcendental functions are used to conclude the proof of Theorem 10.

Next, we verified some of the classical properties of Fourier transform, given in Table 2.

Table 2: Properties of Fourier Transform

Mathematical Form	Formalized Form
Linearity	
$\mathcal{F}[\alpha f(t) + \beta g(t)] =$ $\alpha F(\omega) + \beta G(\omega)$	⊢ ∀ f g w a b. fourier_exists f ∧ fourier_exists g ⇒ fourier_transform (λt. a * f t + b * g t) w = a * fourier_transform f w + b * fourier_transform g w
Time Shifting (Time Advance and Time Delay)	
$\mathcal{F}[f(t + t0)] = F(\omega)e^{+i\omega t0}$	⊢ ∀ f w t0. fourier_exists f ⇒ fourier_transform (λt. f (t + t0)) w = fourier_transform f w * cexp ((ii * Cx w) * Cx t0)
$\mathcal{F}[f(t - t0)] = F(\omega)e^{-i\omega t0}$	⊢ ∀ f w t0. fourier_exists f ⇒ fourier_transform (λt. f (t - t0)) w = fourier_transform f w * cexp (--((ii * Cx w) * Cx t0))
Frequency Shifting (Right and Left Shifting)	
$\mathcal{F}[e^{+i\omega_0 t} f(t)] = F(\omega - \omega_0)$	⊢ ∀ f w w0. fourier_exists f ⇒ fourier_transform (λt. cexp ((ii * Cx w0) * Cx t) * f t) w = fourier_transform f (w - w0)

9

$\mathcal{F}[e^{-i\omega_0 t} f(t)] = F(\omega + \omega_0)$	⊢ ∀ f w w0. fourier_exists f ⇒ fourier (λt. cexp (--(ii * Cx w0) * Cx <u>t</u>) * f t) w = fourier f (w + w0)

Modulation (Cosine and Sine Based Modulation)

$\mathcal{F}[cos(\omega_0 t) f(t)] =$ $\dfrac{F(\omega - \omega_0) + F(\omega + \omega_0)}{2}$	⊢ ∀ f w w0. fourier_exists f ⇒ fourier_transform (λt. ccos (Cx w0 * Cx <u>t</u>) * f t) w = (fourier_transform f (w - w0) + fourier_transform f (w + w0)) / Cx (&2)
$\mathcal{F}[sin(\omega_0 t) f(t)] =$ $\dfrac{F(\omega - \omega_0) - F(\omega + \omega_0)}{2i}$	⊢ ∀ f w w0. fourier_exists f ⇒ fourier_transform (λt. csin (Cx w0 * Cx <u>t</u>) * f t) w = (fourier_transform f (w - w0) - fourier_transform f (w + w0)) / (Cx (&2) * ii)

Time Scaling

$\mathcal{F}[f(at)] = \dfrac{1}{|a|} F(\dfrac{\omega}{a})$	⊢ ∀ f w a. fourier_exists f ∧ ∼(a = &0) ⇒ fourier_transform (λt. f (a % t)) w = (Cx (&1) / Cx (abs a)) * fourier_transform f (w / a)

Time Reversal

$\mathcal{F}[f(-t)] = F(-\omega)$	⊢ ∀ f w. fourier_exists f ⇒ fourier_transform (λt. f (--t)) w = fourier_transform f (--w)

First-order Differentiation

$\mathcal{F}[\dfrac{d}{dt} f(t)] = i\omega F(\omega)$	⊢ ∀ f w. fourier_exists f ∧ fourier_exists (λt. vector_derivative f (at t)) ∧ (∀t. f differentiable at t) ∧ ((λt. f <u>t</u>) → vec 0) at_posinfinity ∧ ((λt. f <u>t</u>) → vec 0) at_neginfinity ⇒ fourier_transform (λt. vector_derivative f (at t)) w = ii * Cx w * fourier_transform f w

Higher-order Differentiation

$\mathcal{F}[\dfrac{d^n}{dt^n} f(t)] = (i\omega)^n F(\omega)$	⊢ ∀ f w n. fourier_exists_higher_deriv n f ∧ (∀t. differentiable_higher_derivative n f t) ∧ (∀k. k < n ⇒ ((λt. higher_vector_derivative k f <u>t</u>) → vec 0) at_posinfinity) ∧ (∀k. k < n ⇒ ((λt. higher_vector_derivative k f <u>t</u>) → vec 0) at_neginfinity) ⇒ fourier_transform (λt. higher_vector_derivative n f t) w = (ii * Cx w) pow n * fourier_transform f w

Area Under a Function	
$\int_{-\infty}^{\infty} f(t)dt = F(0)$	$\vdash \forall$ `f. fourier_exists f` $\Rightarrow$ `integral UNIV f = fourier_transform f (&0)`

The first property is *linearity*, which is frequently used for the analysis of systems having composition of subsystems and accept different scaled inputs.

Next, we verified the *time shifting* property, which is usually used to evaluate the Fourier transform of the function f that is shifted over some constant value of time. The time shifting of the function f can be towards the left of the origin of the time axis (time advance) or towards the right side of the origin of the time-axis (time delay). Its mathematical and the formalized form is given in Table 2.

The *frequency shifting* property of Fourier transform is usually used to evaluate the Fourier transform of multiplication of the function f with the exponential function. It basically shifts the frequency domain representation of f to a certain portion of the frequency spectrum, which is desired for the corresponding frequency analysis. Similar to the time shifting, the frequency shifting is of two types. The frequency right shifting (frequency delay) shifts the frequency signal to the right on the frequency axis and the frequency left shifting (frequency advance) shifts the frequency signal to the left on the frequency axis. The mathematical and the formalized forms of both versions of the frequency shifting are given in Table 2.

The next entry in Table 2 presents a variant of frequency shifting, called the *modulation* property, which is usually used to evaluate the Fourier transform of multiplication of the function f with the cosine and sine functions. This property forms the basis of the Amplitude Modulation (AM) in communication systems. The multiplication of the sinusoidal functions (carrier signals) with the function f in time-domain shifts the frequency components to the portion of the frequency spectrum that is desired for a particular signal transmission.

Next, we verified the *time scaling* property of Fourier transform of a function f, as given in Table 2. Here $a : \mathbb{R}$ is an arbitrary constant. If $|a| < 1$, then the function $f(at)$ represents the function f compressed by a factor of a and its resulting frequency spectrum will be expanded by the same factor. Similarly, in the case of $|a| > 1$, the function $f(at)$ is expanded by the factor a and its corresponding frequency spectrum will be compressed by the same factor. The next property is the *time reversal* property, which is a special case of time scaling property, under the condition $a = -1$.

The Fourier transform of the *differential of a function f* is a very important property that enables us to evaluate the frequency spectrum of the derivative of a function f using the Fourier transform of f. Its mathematical and formalized form is presented in Table 2. In its formalized form, the first two assumptions ensure that the Fourier transforms of the function `f` and its derivative $\frac{df}{dt}$ exist. The third assumption models the condition that the function `f` is differentiable at every $t \epsilon \mathbb{R}$. The last two assumptions represent the condition that $\lim_{t \to \pm\infty} f(t) = 0$. Finally, the conclusion provides the Fourier transform of the first order derivative of the given function. The proof of this property involves a significant amount of arithmetic reasoning along with the integration by parts and the fact $f(t)e^{-i\omega t}|_{-\infty}^{\infty} = (\lim_{B \to \infty} f(B)e^{-i\omega B} - \lim_{A \to -\infty} f(A)e^{-i\omega A}) = 0$ and the integrability of the Fourier integrand on the positive and negative real lines.

11

The next property is the *Fourier transform of a n-times continuously differentiable function f*, which is the foremost foundational property for analysing higher-order differential equations based on the Fourier transform. In its formalized form, the first assumption ensures the Fourier transform existence of `f` and its first `n` higher-order derivatives. Similarly, the second assumption ensures the differentiability of `f` and its first `n` higher-order derivatives on `t` ϵ $\mathbb{R}$. The next two assumptions model the condition $\lim_{t \to \pm\infty} f^{(k)}(t) = 0$ for each $k = 0, 1, 2, ..., n-1$, where $f^{(k)}$ denotes the kth derivative of `f` and $f^{(0)} = $ `f`. Finally, the conclusion is the Fourier transform of nth order derivative of the function. Its proof is mainly based on induction on variable `n` along with Fourier transform of the first order derivative of the given function. The Fourier transform can be used to evaluate the *area under a function f*, as given in the final entry of Table 2.

4.2. Relationship with Various Transforms

This section presents the relationship of Fourier transform with various transforms, which include Fourier Cosine, Fourier Sine and Laplace transforms.

4.2.1. Relationship with Fourier Cosine and Fourier Sine Transforms

The Fourier transform of the even and odd function enables us to relate the Fourier transform to Fourier Cosine and Fourier Sine transforms. The Fourier Cosine transform is mathematically expressed by the following indefinite integral:

$$F_c(\omega) = \int_{-\infty}^{+\infty} f(t)cos(\omega t)dt \tag{2}$$

If the input function is an even function, i.e., $f(-t) = f(t)$ for all t ϵ $\mathbb{R}$, then its Fourier transform is equal to its Fourier Cosine transform.

We verify the even function property as the following theorem:

Theorem 11. Fourier Transform of Even Function
$\vdash \forall$ `f w. fourier_exists f` $\wedge$ `(`$\forall$`t. f (--t) = f t)`
$\Rightarrow$ `fourier_transform f w = fourier_cosine_transform f w`

In the above theorem, the two assumptions ensure the Fourier existence of `f` and model the even function condition, respectively. The conclusion presents the relationship of Fourier transform to Fourier Cosine transform.

Next, the Fourier Sine transform is mathematically expressed as:

$$F_s(\omega) = \int_{-\infty}^{+\infty} f(t)sin(\omega t)dt \tag{3}$$

If the input function is an odd function, i.e., $f(-t) = -f(t)$ for all t ϵ $\mathbb{R}$, then its Fourier transform is equal to its Fourier Sine transform.

The odd function property is verified in HOL-Light as the following theorem:

Theorem 12. Fourier Transform of Odd Function
$\vdash \forall$ `f w. fourier_exists f` $\wedge$ `(`$\forall$`t. f (--t) = --f t)`
$\Rightarrow$ `fourier_transform f w = --ii * fourier_sine_transform f w`

In the above theorem, the first assumption presents the condition of the Fourier existence of the function `f`, whereas the second assumption models the odd function condition.

4.2.2.   Relationship with Laplace Transform

By restricting the complex-valued function $f : \mathbb{R}^1 \to \mathbb{R}^2$ and the variable $s : \mathbb{R}^2$ for Laplace Transform, we can find a very important relationship between Fourier and Laplace transforms. The Laplace transform of a function f is given by the following equation:

$$F(s) = \int_0^\infty f(t)e^{-st}dt, \ s \ \epsilon \ \mathbb{C} \tag{4}$$

A formalized form of the Laplace transform is as follows (Taqdees and Hasan, 2013):

Definition 13. *Laplace Transform*
```
⊢ ∀ s f. laplace_transform f s =
        lim at_posinfinity (λb. integral (interval [&0, b̄])
                                         (λt. cexp (--(s * Cx t̲)) * f t))
```

The Laplace transform of a function f exists, if the function `f` is piecewise smooth and of exponential order on the positive real line. The existence of the Laplace transform has been formally defined as follows (Taqdees and Hasan, 2013; Rashid and Hasan, 2017):

Definition 14. *Laplace Exists*
```
⊢ ∀ s f. laplace_exists f s ⇔
        (∀ b. f piecewise_differentiable_on interval [&0, b̄]) ∧
        (∃ M a. Re s > a̲ ∧ exp_order f M a)
```

The function `exp_order` in the above definition has been formally defined as (Taqdees and Hasan, 2013; Rashid and Hasan, 2017):

Definition 15. *Exponential Order Function*
```
⊢ ∀ f M a. exp_order f M a ⇔ &0 < M ∧
                    (∀ t. &0 <= t ⇒ norm (f t̄) <= M * exp (a̲ * t))
```

If the function f is causal, i.e., $f(t) = 0$ for all $t < 0$ and the real part of Laplace variable $s : \mathbb{R}^2$ is zero, i.e., $Re\ s = 0$, then the Fourier transform of function f is equal to Laplace transform, i.e., $(\mathcal{F}f)(Im\ s) = (\mathcal{L}f)(s)\,|_{Re\ s\ =\ 0}$ (Thomas et al., 2016).

The above relationship is verified in HOL-Light as follow:

Theorem 16. Relationship with Laplace Transform
```
⊢ ∀ f s. laplace_exists f s ∧
  (∀t. t IN {t | t̲ <= &0} ⇒ f t = vec 0) ∧ (∀t. Re s = &0)
                        ⇒ fourier_transform f (Im s) = laplace_transform f s
```

The first assumption of above theorem ensure the existence of the Laplace transform. The next two assumptions ensure that `f` is a causal function and the real part of the Laplace variable `s` is zero. The proof of the above theorem is mainly based on the integrability of the Fourier integrand on the positive and negative real lines, properties of the complex exponential, and the following important lemma:

Lemma 17. Alternative Representation of Laplace Transform

```
⊢ ∀ f s. laplace_exists f s ⇒
        laplace_transform f s =
        integral {t | &0 <= t} (λt. cexp (--(s * Cx t)) * f t)
```

The above lemma presents an alternative representation of the Laplace transform, given in Definition 13. This alternate representation of Laplace transform as well as the formalization of Fourier transform, given in Definition 7, is better than the formal definition (Definition 13), presented in (Taqdees and Hasan, 2013), which involves the notion of the limit. As the HOL-Light definition of the integral function implicitly encompasses infinite limits of integration, so we do not need to involve the notion of limit. Hence, this alternate representation covers the region of integration, i.e., $[0, \infty)$, as `{t | &0 <= drop t}` and is equivalent to the definition of Laplace transform given by Definition 13. Similarly, the region of integration for Fourier transform, i.e., $(-\infty, \infty)$ is modeled as `UNIV`. This relationship (Theorem 17) can facilitate the formal reasoning process of Laplace transform related properties and thus can be very useful towards the formalization of inverse Laplace transform function and verification of its associated properties. Moreover, the formal definition of the Fourier transform presented as Definition 7 considerably simplifies the reasoning process in the verification of its properties.

4.3. Differential Equation

Differential equations are widely used to mathematical model the complex dynamics of a continuous-time system and hence characterize the behavior of the system at each time instant. A general linear differential equation can be mathematically expressed as follow:

$$
\begin{aligned}
Differential\ Equation &= \sum_{k=0}^{n} \alpha_k \frac{d^k y}{dt^k} \\
&= \alpha_n \frac{d^n y}{dt^n} + \alpha_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + ... + \alpha_1 \frac{d^1 y}{dt^1} + \alpha_0 y
\end{aligned}
\tag{5}
$$

where n is the order of the differential equation and α_i represents the list of the constant coefficients. The Fourier transform of the above n^{th}-order differential equation is given by the following mathematical expression:

$$
\mathcal{F}\left(\sum_{k=0}^{n} \alpha_k \frac{d^k y}{dt^k} \right) = Y(\omega) \sum_{k=0}^{n} \alpha_k (i\omega)^k
\tag{6}
$$

We formalize the above differential equation using the following definition in HOL-Light:

Definition 18. *Differential Equation of Order n*

```
⊢ ∀ n lst f t. differential_equation n lst f t =
                vsum (0..n) (λk. EL k lst * higher_order_derivative k f t)
```

The function `differential_equation` accepts the order of the differential equation n, a list of constant coefficients `lst`, a differentiable function `f` and the differentiation

variable `t`. It utilizes the functions `vsum n f` and `EL m lst`, which return the vector summation $\sum_{i=0}^{n} f_i$ and the mth element of a list `lst`, respectively, to generate the differential equation corresponding to the given parameters.

Next, we verify the Fourier transform of a linear differential equation, which is expected to be the most widely used result of our formalization as depicted in Sections 6 and 7, and is given by the following theorem in HOL-Light.

Theorem 19. Fourier Transform of Differential Equation of Order n

```
⊢ ∀ f lst w n. fourier_exists_higher_deriv n f ∧
     (∀t. differentiable_higher_derivative n f t) ∧
     (∀k. k < n ⇒
         ((λt. higher_vector_derivative k f t̄) → vec 0) at_posinfinity) ∧
     (∀k. k < n ⇒
         ((λt. higher_vector_derivative k f t̄) → vec 0) at_neginfinity)
             ⇒ fourier_transform (λt. differential_equation n lst f t) w =
         fourier_transform f w * vsum (0..n) (λk. EL k lst * (ii * Cx w) pow k)
```

The set of the assumptions of the above theorem is the same as that of property named *Higher Order Differentiation in Time Domain*, given in Table 2. The conclusion of Theorem 19 is the Fourier transform of a nth-order linear differential equation. The proof of above theorem is based on induction on variable `n`. The proof of the base case is based on simple arithmetic reasoning and the step case is discharged using Theorems 9, *linearity* and *Higher Order Differentiation in Time Domain* properties, along with the following important lemma about the Fourier existence of the differential equation.

Lemma 20. Fourier Existence of Differential Equation

```
⊢ ∀ n lst f. fourier_exists_higher_deriv n f
                 ⇒ fourier_exists (λt. differential_equation n lst f t)
```

5. Fourier Transform of Some Commonly Used Functions

In this section, we present Fourier transform of some functions that are commonly used for the analysis of the physical and engineering systems in various domains, i.e., signal processing, analog circuits, optical systems and communication systems etc.

5.1. Fourier Transform of a Rectangular Pulse

The rectangular pulse is characterized by having a constant value over a range of values on the time axis. It is represented by the following mathematical expression.

$$f(t) = \begin{cases} 1, & |t| \leq T_1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

It has a constant value 1 inside the interval $[-T_1, T_1]$, whereas it is 0 outside this interval over the whole real line. For the value of $T_1 = 0.5$, it is known as the unit gate function. The Fourier transform of the rectangular pulse in given by the following equation.

$$F(\omega) = 2\,T_1\,sinc(\omega\,T_1)$$
$$= 2\,T_1\,\frac{sin(\omega\,T_1)}{\omega\,T_1} \tag{8}$$

where $sinc(\omega\,T_1)$ is the sinc function, which is the multiplication of a sinusoidal function $sin(\omega\,T_1)$ with the monotonically decreasing function $\dfrac{1}{\omega\,T_1}$, which makes it a continuously decreasing sinusoidal function. It approaches to value 0 at both endpoints of the axis. i.e., $-\infty$ and $+\infty$. It is also known as the interpolation or filtering function. We model the rectangular pulse and the sinc function using the following HOL-Light functions:

Definition 21. *Rectangular Pulse*
```
⊢ rect_pulse T1 =
            (λt. if t IN {t | --T1 <= t ∧ t <= T1}
                then Cx (&1)
                else Cx (&0))
```

Definition 22. *Sinc Function*
```
⊢ sinc T1 w = csin (Cx w * Cx T1) / (Cx w * Cx T1)
```

The Fourier transform of the rectangular pulse is represented as the following HOL-Light theorem:

Theorem 23. Fourier Transform of Rectangular Pulse
```
⊢ ∀ T1 w. &0 < T1 ∧ ∼(w = &0)
            ⟹ fourier_transform (rect_pulse T1) w =
                                Cx (&2) * Cx T1 * sinc T1 w
```

We verified the above theorem using Definitions 7, 21 and 22, and the properties of the integration along with some arithmetic reasoning.

5.2. Fourier Transform of Unilateral Negative Complex Exponential

The unilateral negative complex exponential is given by the following mathematical expression:

$$f(t) = \begin{cases} 0, & t < 0 \\ e^{-ct}, & t \geq 0 \end{cases} \tag{9}$$

where c is a positive real constant, which makes the function e^{-ct} an exponentially decaying function. The Fourier transform of the unilateral complex exponential function is as below:

$$F(\omega) = \frac{1}{c + i\omega} \tag{10}$$

We formally model the unilateral negative complex exponential by the following HOL-Light function:

Definition 24. *Unilateral Negative Complex Exponential*
```
⊢ unilat_neg_cexp c =
            (λt. if t IN {t | &0 <= t}
                 then cexp (--Cx c * Cx t)
                 else Cx (&0))
```

We verified the Fourier transform of the unilateral negative complex exponential as the following theorem.

Theorem 25. Fourier Transform of Unilateral Negative Exponential
```
⊢ ∀ c w. &0 < c ∧ ∼(Cx c + ii * Cx w = Cx (&0))
            ⟹ fourier_transform (unilat_neg_cexp c) w =
                                        Cx (&1) / (Cx c + ii * Cx w)
```

5.3. Fourier Transform of Bilateral Complex Exponential

The Fourier transform of the bilateral complex exponential is given by the following mathematical equation:

$$\mathcal{F}[e^{-|t|}] = \frac{2}{1 + \omega^2} \tag{11}$$

We verified its Fourier transform as the following theorem:

Theorem 26. Fourier Transform of Bilateral Complex Exponential
```
⊢ ∀ w. ∼(Cx (&1) - ii * Cx w = Cx (&0)) ∧
       ∼(Cx (&1) + ii * Cx w = Cx (&0))
            ⟹ fourier_transform (λt. cexp (--Cx (abs t))) w
                                Cx (&2) / (Cx (&1) + Cx w pow 2)
```

The verification process of the above theorem starts by rewriting with the definition of Fourier transform. Next, we split the region of integration, i.e., $(-\infty, +\infty)$, into $(-\infty, 0]$ and $[0, +\infty)$, as a result, we obtain two integrals with the same integrand and the respective regions of integration. We rewrite the resultant subgoal with the definition of absolute value of a real number to replace its value. i.e., $|t| = -t$ in interval $(-\infty, 0]$ and $|t| = t$ in interval $[0, +\infty)$. Next, these two integrals are evaluated using the properties of integration along with the complex arithmetic reasoning to conclude the proof of Theorem 26.

5.4. Fourier Transform of Finite Duration Sinusoidal Tone-burst

The sinusoidal tone-burst occurring for a finite duration $-T_1$ to T_1 is mathematically defined as follows:

$$f(t) = \begin{cases} sin\omega_0 t, & |t| \leq T_1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The Fourier transform of the above sinusoidal tone-burst is given by the following mathematical expression:

$$F(\omega) = -iT_1\{sinc((\omega - \omega_0)T_1) - sinc((\omega + \omega_0)T_1)\} \tag{13}$$

The frequency spectrum corresponding to the above sine wave is inversely proportional to its duration $2T_1$. i.e., with the increase in the duration $2T_1$, it results into a narrower frequency line spectrum and vice versa.

We defined Equation 12 as the following HOL-Light function:

Definition 27. *Finite Duration Sinusoidal Tone-burst*
```
⊢ ∀ T1 w0. sine_tone_burst T1 w0 =
                      (λt. csin (Cx w0 * Cx t) * rect_pulse T1 t)
```

The Fourier transform of the above sinusoidal tone-burst is represented as the following theorem:

Theorem 28. Fourier Transform of Finite Duration Sinusoidal Tone-burst
```
⊢ ∀ T1 w w0. &0 < T1 ∧
             ∼(w - w0 = &0) ∧ ∼(w + w0 = &0)
                ⟹ fourier_transform (sine_tone_burst T1 w0) w =
                   --ii * Cx T1 * (sinc T1 (w - w0) - sinc T1 (w + w0))
```

We start the verification process of the above theorem by rewriting with Definitions 7, 27 and 21, and the definition of complex sine `csin` (Definition 3), which results into a subgoal having vector integral of the linear combination of the `cexp` functions over interval $[-T_1, T_1]$, which is verified based on the properties of integration along with the arithmetic reasoning.

5.5. Fourier Transform of Damped Unilateral Sinusoidal Function

The damped unilateral sinusoidal function is basically a product of a decaying complex exponential function with the periodic sinusoidal function and is given as below:

$$f(t) = \begin{cases} 0, & t < 0 \\ e^{-ct} sin\omega_0 t, & t \geq 0 \end{cases} \tag{14}$$

where c is a positive real constant. The Fourier transform of the damped unilateral sinusoidal function is represented by the following equation:

$$F(\omega) = \frac{w_0}{(c + i\omega)^2 + w_0{}^2} \tag{15}$$

We formalized Equation 14 as the following HOL-Light function:

Definition 29. *Damped Unilateral Sinusoidal Function*
```
⊢ damped_unilat_sine c w0 =
        (λt. if t IN {t | &0 <= t}
             then cexp (--Cx c * Cx t) * csin (Cx w0 * Cx t)
             else Cx (&0))
```

Its Fourier transform has been verified as the following theorem:

Theorem 30. Fourier Transform of Damped Unilateral Sinusoidal Function
```
⊢ ∀ c w w0. &0 < c ∧
            ∼(Cx c + ii * Cx (w - w0) = Cx (&0)) ∧
```

$$\sim(\text{Cx c} + \text{ii} * \text{Cx (w + w0)} = \text{Cx (\&0)}) \wedge$$
$$\sim((\text{Cx c} + \text{ii} * \text{Cx w}) \text{ pow } 2 + \text{Cx w0 pow } 2 = \text{Cx (\&0)})$$
$$\Longrightarrow \text{fourier\_transform (damped\_unilat\_sine c w0) w} =$$
$$\text{Cx w0} / ((\text{Cx c} + \text{ii} * \text{Cx w}) \text{ pow } 2 + \text{Cx w0 pow } 2)$$

The proof process of the above theorem involves Definitions 7 and 29 along with some properties about the complex exponential functions.

This completes our formalization of Fourier transform in HOL-Light. The source code of our formalization is available for download (Rashid, 2017) and can be utilized for further development and the analysis of continuous-time systems.

6. Formal Analysis of a Generic n-order System

In this section, we present the formal modeling and the frequency response analysis of a generic n-order Linear Time Invariant (LTI) system. A generic n-order LTI system (Adams, 2012) presents a relationship between an input signal $x(t)$ and the output signal $y(t)$ and its dynamics are modeled using a higher-order differential equation. Due to the generic nature of its differential equation based model and its corresponding frequency response analysis, we can use it for the modeling and frequency analysis of any real-world application, which certainly eases out the formal reasoning based analysis of these system and is illustrated in the next section. Fig. 2 provides the block diagram representation of a generic n-order system, which is primarily composed of the addition, the scalar multiplication and the integration operations (Girod et al., 2001).

The generalized linear differential equation, with constant coefficient, describing the input-output relationship for this generic n-order system is mathematically expressed as (Adams, 2012):

$$\sum_{k=0}^{n} \beta_k \frac{d^k}{dt^k} y(t) = \sum_{k=0}^{m} \alpha_k \frac{d^k}{dt^k} x(t), \quad m \le n \tag{16}$$

where $y(t)$ in the above equation is the output and $x(t)$ is the input to the system. The constants α_k and β_k are the coefficients of the input and the output differentials of order k, respectively. The greatest index n of the non-zero coefficient β_n determines the order of the underlying system. The corresponding frequency response of the system is given by the following mathematical expression:

$$\frac{Y(\omega)}{X(\omega)} = \frac{\sum_{k=0}^{m} \alpha_k (i\omega)^k}{\sum_{k=0}^{n} \beta_k (i\omega)^k} \tag{17}$$

In order to verify the above frequency response of the given system, we first model the corresponding differential equation as the following HOL-Light function:

Definition 31. *Differential Equation of n-order LTI System*
$\vdash \forall$ n outlst y m inlst x t. diff_eq_n_order_sys m n inlst outlst x y t $\Leftrightarrow$
    differential_equation n outlst y t = differential_equation m inlst x t

Next, we verified the frequency response, given in Equation 17, of the generic n-order system as the following HOL-Light theorem.
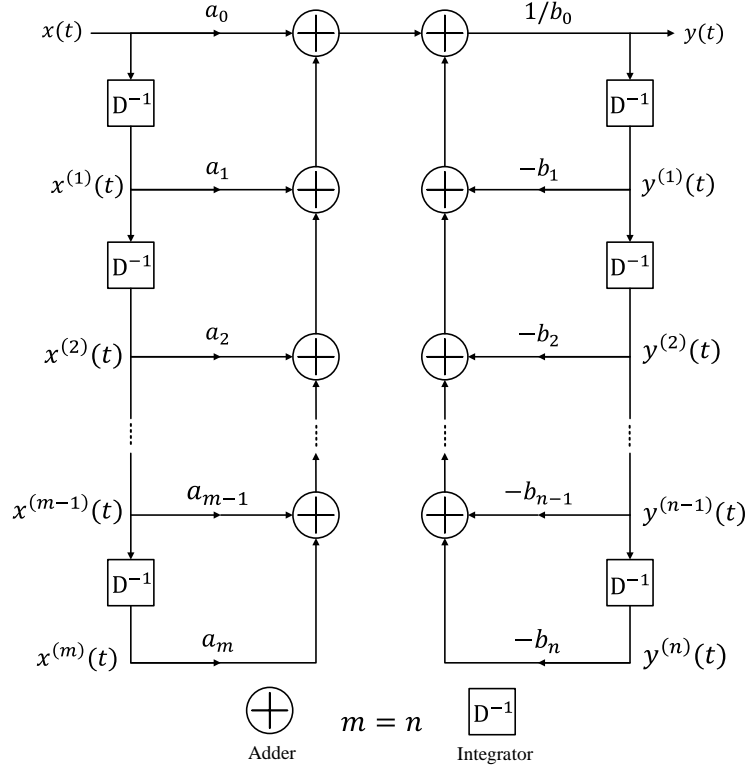
Fig. 2. Block Diagram Representation of a Generic n-order System

Theorem 32. Frequency Response of n-order LTI System

```
⊢ ∀ y x m n inlst outlst w.
  (∀t. differentiable_higher_derivative n y t) ∧
  (∀t. differentiable_higher_derivative m x t) ∧
  fourier_exists_higher_deriv n y ∧ fourier_exists_higher_deriv m x ∧
  (∀k. k < n ⇒
      ((λt. higher_vector_derivative k y t̄) → vec 0) at_posinfinity) ∧
  (∀k. k < n ⇒
      ((λt. higher_vector_derivative k y t̄) → vec 0) at_neginfinity) ∧
  (∀k. k < m ⇒
      ((λt. higher_vector_derivative k x t̄) → vec 0) at_posinfinity) ∧
  (∀k. k < m ⇒
      ((λt. higher_vector_derivative k x t̄) → vec 0) at_neginfinity) ∧
  (∀t. diff_eq_n_order_sys m n inlst outlst x y t) ∧
  ∼(fourier_transform x w = Cx (&0)) ∧
  ∼(vsum (0..n) (λt. Cx (EL t outlst) * (ii * Cx w) pow t) = Cx (&0))
                ⇒ fourier_transform y w / fourier_transform x w =
                      vsum (0..m) (λt. Cx (EL t inlst) * (ii * Cx w) pow t) /
                      vsum (0..n) (λt. Cx (EL t outlst) * (ii * Cx w) pow t)
```

20

The first two assumptions ensure that the functions y and x are differentiable up to the nth and m^{th} order, respectively. The next assumption represents the Fourier transform existence condition upto the nth order derivatives of function y. Similarly, the next assumption ensures that the Fourier transform exists up to the mth order derivative of the function x. The next two assumptions represent the condition $\lim_{t \to \pm\infty} y^{(k)}(t) = 0$ for all $k = 0, 1, ..., n - 1$, i.e., $\lim_{t \to \pm\infty} y^{(n-1)}(t) = 0, ... , \lim_{t \to \pm\infty} y^{(0)}(t) = \lim_{t \to \pm\infty} y(t) = 0$, where $y^{(k)}(t)$ is the kth derivative of y with respect to t. The next two assumptions provide the condition $\lim_{t \to \pm\infty} x^{(k)}(t) = 0$ for each $k = 0, 1, ..., m - 1$. The next assumption represents the formalization of Equation 16 and the last two assumptions provide some interesting design related relationships, which must hold for constructing a reliable continuous-time system. Finally, the conclusion of the above theorem represents the frequency response given by Equation 17. The proof of Theorem 32 was very straightforward and mainly based on Theorem 19, along with some arithmetic reasoning, thanks to our foundational formalization presented in the previous sections. The verification of this theorem is very useful as it greatly simplifies the verification of the frequency response of any real-world application as illustrated in the next section.

7. Applications

In this section, to illustrate the utilization of our foundational formalization for analyzing real-world continuous systems, we present a formal analysis of an audio equalizer and a MEMs accelerometer. To the best of our knowledge, these systems could not have been verified while capturing their continuous behavior in the true form by any other existing computer-based analysis technique.

7.1. Formal Analysis of an Audio Equalizer

An audio equalizer (Tan and Jiang, 2007) is an electronic circuit that adjusts the balance between different frequency components within an audio signal. The block diagram of a 3-channel audio equalizer is illustrated in Fig. 3.
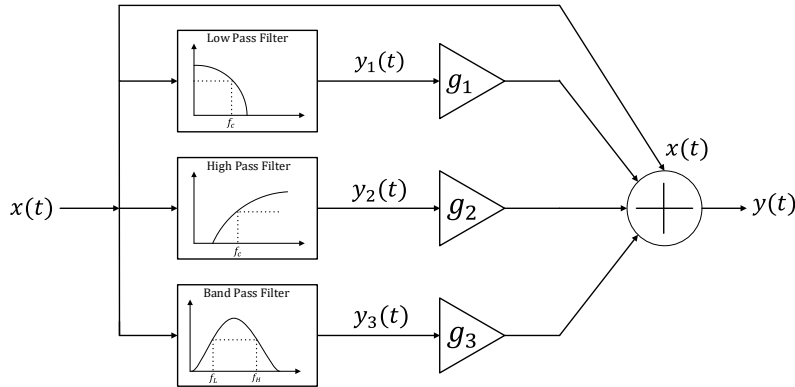

Fig. 3. Block Diagram of Audio Equalizer

21

It mainly consists of three different filters, which are low-pass, high-pass and bandpass, allowing a certain range of the frequency to pass on. The low-pass filter allows the passage of signals having frequency lower than then the cutoff frequency ($\omega_c = 1/2\pi f_c$), whereas, the high-pass filter passes the signal with a frequency higher than the cutoff frequency. Whereas the bandpass filter passes the signal having frequency components in a certain range only, as shown in Figure 3. After each filtering stage, some signal amplification with gain (g_i) is applied in order to enhance the quality of the signal. Being a major component of an audio equalizer, we verify the frequency response of each of the individual filter. Here, we only present the formal verification of the frequency response of the bandpass filter only due to space restrictions and the verification of rest of the filters can be found in the proof script (Rashid, 2017).

In order to verify the frequency response of the bandpass filter, we first model its corresponding differential equation, which is given by the following HOL-Light function:

Definition 33. *Differential Equation of Bandpass Filter*
⊢ ∀ `wc. outlst_de_bpf wc = [wc pow 2; &2 * wc; &1]`
⊢ ∀ `wc. inlst_de_bpf wc = [&0; wc]`
⊢ `diff_eq_BP_FILTER inlst_de_bpf outlst_de_bpf x y t wc ⇔`
        `differential_equation 2 (outlst_de_bpf wc) y t =`
        `differential_equation 1 (inlst_de_bpf wc) x t`

where the function `diff_eq_BP_FILTER` accepts the function variables x and y and the lists of coefficients (`inlst_de_bpf` and `outlst_de_bpf`) and returns the corresponding differential equation of the bandpass filter.

Next, the frequency response of the bandpass filter is mathematically expressed as:

$$
\begin{aligned}
\frac{Y(\omega)}{X(\omega)} &= \frac{\omega_c}{i\omega + \omega_c} \times \frac{i\omega}{i\omega + \omega_c} \\
&= \frac{\omega_c(i\omega)}{(i\omega)^2 + 2\omega_c(i\omega) + (\omega_c)^2}
\end{aligned}
\tag{18}
$$

We verified the above frequency response as the following HOL-Light theorem:

Theorem 34. Frequency Response of Bandpass Filter
⊢ ∀ `y x w wc. &0 < wc ∧`
  (∀`t. differentiable_higher_derivative 2 y t`) ∧
  (∀`t. differentiable_higher_derivative 1 x t`) ∧
  `fourier_exists_higher_deriv 2 y ∧ fourier_exists_higher_deriv 1 x ∧`
  (∀`k. k < 2 ⇒`
      ((λ`t. higher_vector_derivative k y` $\overline{\mathtt{t}}$) → `vec 0`) `at_posinfinity`) ∧
  (∀`k. k < 2 ⇒`
      ((λ`t. higher_vector_derivative k y` $\overline{\mathtt{t}}$) → `vec 0`) `at_neginfinity`) ∧
  ((λ`t. x` $\overline{\mathtt{t}}$) → `vec 0`) `at_posinfinity ∧`
  ((λ`t. x` $\overline{\mathtt{t}}$) → `vec 0`) `at_neginfinity ∧`
  (∀`t. diff_eq_BP_FILTER inlst_de_bpf outlst_de_bpf x y t wc`) ∧
  ∼(`fourier_transform x w = Cx (&0)`) ∧
  ∼((`ii * Cx w) pow 2 + Cx (&2) * Cx wc * ii * Cx w + Cx wc pow 2 = Cx (&0)`)
      ⇒ `fourier_transform y w / fourier_transform x w =`

22

```
Cx wc * ii * Cx w /
    ((ii * Cx w) pow 2 + Cx (&2) * Cx wc * ii * Cx w + Cx wc pow 2)
```

The first assumption ensures that the variable corresponding to the cutoff frequency (`wc`) cannot be negative or zero. The next two assumptions ensure that the functions `y` and `x` are differentiable up to the second and first order, respectively. The next two assumptions represent the Fourier transform existence condition up to the second and first order derivatives of the functions `y` and `x`, respectively. The next two assumptions represent the condition $\lim_{t \to \pm\infty} y^{(k)}(t) = 0$ for each $k = 0, 1$. The next two assumptions provide the condition $\lim_{t \to \pm\infty} x(t) = 0$. The next assumption represents the formalization of the corresponding differential equation and the last two assumptions provide some interesting design related relationships, which must hold for constructing a reliable bandpass filter. Finally, the conclusion of the above theorem represents the frequency response given by Equation 18. The proof is based on Theorem 32, along with some arithmetic reasoning.

7.2. Formal Analysis of MEMs Accelerometer

An accelerometer is an electromechanical device that is used for the measurement of the static and the dynamic accelerations, i.e., the acceleration due to gravity anywhere on the earth and the acceleration due to the motion or vibration of an object according to the theory of relatively. It uses the sensors, which further make use of the environmental physical parameters, i.e., pressure, temperature, light and force. Micro-Electro-Mechanical systems (MEMs) based accelerometer (Kaajakari et al., 2009) are widely used as accelerometers. They are smaller in size, utilizing low power and thus due to these features, are integrated in a variety of applications, such as aircrafts (Kuznetsov et al., 2011), airbag deployment (Galvin et al., 2001), robotic telepresence (Hung et al., 2004), handheld computing gadgets (Fennelly et al., 2012), natural disaster measurement devices (Hsieh et al., 2014) and automated external defibrillators (AEDs) (Eggers et al., 2016). Due to their wide usage in the safety critical domains, the accuracy of their frequency response analysis is of utmost importance. A typical MEMs accelerometer is depicted in Fig. 4a whereas its mechanical lumped model (Haykin and Van Veen, 2007) is illustrated in Fig. 4b.

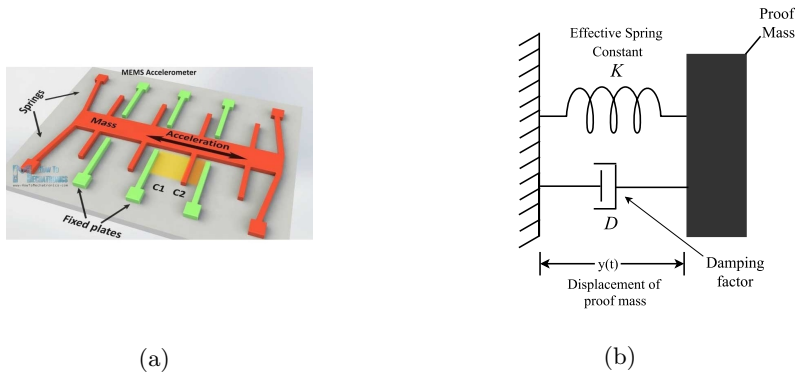(a)   (b)

Fig. 4. MEMs Accelerometer (a) Design (b) Mechanical Lumped Model

The differential equation modeling the dynamical behaviour of the MEMs accelerometer can be expressed as (Haykin and Van Veen, 2007):

$$\frac{d^2 y(t)}{dt^2} + \frac{D}{M}\frac{dy(t)}{dt} + \frac{K}{M}y(t) = u(t), \tag{19}$$

In the above equation, M is the proof mass, whereas, K is the effective spring constant and D represents the damping factor, which affects the dynamic movement of the proof mass as shown in Figure 4b. All of these are design parameters of the underlying system and can have positive values only. Similarly, $x(t)$ is the external acceleration due to motion of the proof mass, whereas $y(t)$ is the displacement of the corresponding mass.

The corresponding frequency response of the MEMs accelerometer is given as follows:

$$\frac{Y(\omega)}{U(\omega)} = \frac{1}{(i\omega)^2 + \dfrac{D}{M}(i\omega) + \dfrac{K}{M}} \tag{20}$$

In order to verify its frequency response, we first model the corresponding differential equation as the following HOL-Light function:

Definition 35. *Differential Equation of MEMs Accelerometer*
⊢ ∀ K D M. outlst_de_ma K D M = [K / M; D / M; &1]
⊢ inlst_de_ma = [&1]
⊢ diff_eq_MEMs_ACC inlst_de_ma outlst_de_ma u y t K D M ⇔
    differential_equation 2 (outlst_de_ma K D M) y t =
    differential_equation 0 inlst_de_ma u t

where the function diff_eq_MEMs_ACC accepts the function variables x and y and the lists of coefficients (inlst_de_ma and outlst_de_ma) and returns the corresponding differential equation of the MEMs accelerometer.

Next, we verify its frequency response as the following theorem in HOL-Light:

Theorem 36. Frequency Response of MEMs Accelerometer
⊢ ∀ y u w K D M. &0 < M ∧ &0 < D ∧ &0 < K ∧
 (∀t. differentiable_higher_derivative 2 y t) ∧ (∀t. u differentiable at t) ∧
 fourier_exists_higher_deriv 2 y ∧ fourier_exists u ∧
 (∀k. k < 2 ⇒
   ((λt. higher_vector_derivative k y $\bar{t}$) → vec 0) at_posinfinity) ∧
 (∀k. k < 2 ⇒
   ((λt. higher_vector_derivative k y $\bar{t}$) → vec 0) at_neginfinity) ∧
 (∀t. diff_eq_MEMs_ACC inlst_de_ma outlst_de_ma u y t K D M) ∧
 ∼(fourier u w = Cx (&0)) ∧
 ∼((ii * Cx w) pow 2 + Cx (D / M) * ii * Cx w + Cx (K / M) = Cx (&0))
 ⇒ fourier y w / fourier u w =
   Cx (&1) / ((ii * Cx w) pow 2 + Cx (D / M) * ii * Cx w + Cx (K / M))

The first three assumptions ensure that the variables corresponding to proof mass (M), spring constant (K) and damping factor (D) cannot be negative or zero. The next assumption ensures that the function y is differentiable up to the second order. Similarly, the next assumption represents the differentiability condition for the function u. The next assumption represents the Fourier transform existence condition up to the second

order derivatives of the function y. Similarly, the next assumption provides the Fourier transform existence condition of the function u. The next two assumptions represent the condition $\lim_{t \to \pm\infty} y^{(k)}(t) = 0$ for each $k = 0, 1$, i.e., $\lim_{t \to \pm\infty} y^{(1)}(t) = 0$ and $\lim_{t \to \pm\infty} y^{(0)}(t) = \lim_{t \to \pm\infty} y(t) = 0$, where $y^{(k)}$ is the kth derivative of y. The next assumption represents the formalization of Equation 19 and the last two assumptions provide some interesting design related relationships, which must hold for constructing a reliable MEMs accelerometer. Finally, the conclusion of the above theorem represents the frequency response, given by Equation 20. The proof is based on Theorem 32 along with some arithmetic reasoning.

Besides the above-mentioned audio equalizer and a MEMs based accelerometer applications, we also used the proposed formalization to formally verify the frequency response of a drug therapy model, which can be useful towards finding out a particular amount of dosage of drug lidocaine that has to be supplied to a particular person having the Ventricular arrhythmia and the details about its verification can be found in proof script (Rashid, 2017).

8. Discussion

The distinguishing feature of our proposed formalization as compared to traditional analysis methods is that all of the theorems verified are of generic nature, i.e., all of the variables and functions are universally quantified and can thus be specialized in order to obtain the results for some given values. Moreover, all of the required assumptions are guaranteed to be explicitly mentioned along with the theorem due to the inherent soundness of the theorem proving approach. Similarly, the verification of the frequency response described in Section 6, is of a generic n-order system, can be specialized in order to formally analyse any real-world system as presented in Section 7. Whereas, in the computer simulation techniques, we have to model each of the system individually. Moreover, the high expressiveness of the higher-order logic enables us to model the differential equation and the corresponding frequency response in their true continuous form, whereas, in model checking they are mostly discretized and modeled using a state-transition system, which may compromise the accuracy of the analysis.

The above-mentioned formalization is done interactively. However, we tried to automate the proof processes by making some simplification tactics. We develop a tactic `ASM_REAL_SIMP_TAC`, which simplifies an expression involving real arithmetics using all of the assumptions of the theorem. We also develop a tactic `ASM_COMPLEX_SIMP_TAC`, which simplifies a complex expression involving arithmetic operations using all of the assumptions of a theorem. We developed some more simplification tactics that can be found in our proof script (Rashid, 2017).

The major difficulty faced during the formalization was the unavailability of detailed proofs for the properties of Fourier transform in literature. The available paper-and-pencil based proofs were found to be very abstract and missing the complete reasoning about the steps. The other challenge in reported formalization was that some of the assumptions of the properties of the Fourier transform were not mentioned in the literature as in the case of the first-order and higher-order differentiation properties, the assumptions 4 and 5 of theorem of *First-order Derivative*, presented in Table 2, and the assumptions 3 and 4 of the *Higher-order Derivative*, presented in Table 2, were absent from most of the

Table 3. Verification Detail for Each Theorem

Formalized Theorems	Proof Lines	Man-hours
Theorem 10 (Integrability of Improper Integral)	880	87
Table 2: Linearity	115	13
Table 2: Time Shifting	190	22
Table 2: Frequency Shifting	24	4
Table 2: Modulation	98	11
Table 2: Time Scaling and Time Reversal	200	25
Table 2: First Order Differentiation in Time Domain	355	29
Table 2: Higher Order Differentiation in Time Domain	110	15
Table 2: Area under a function	12	1
Theorems 11 and 12 (Relationship with Fourier Cosine and Fourier Sine Transforms)	296	35
Theorem 16 (Relationship with Laplace Transform)	162	26
Theorem 19 (Fourier Transform of Differential Equation of Order n)	170	25
Theorems 23, 25, 26, 28 and 30 (Fourier Transform of Some Commonly Used Functions)	275	35
Theorem 32 (Frequency Response of n-order LTI System)	65	12
Theorem 34 (Frequency Response of an Audio Equalizer)	90	10
Theorem 36 (Frequency Response of MEMs Accelerator)	63	6

analysis books. The effort involved in the verification of individual theorem in the form of proof lines and the man-hours is presented in Table 3.

The proof process for the formal verification of Theorems 32, 34 and 36 took only 218 lines and 28 man-hours and was very simple and straightforward compared to the reasoning process of Theorem 10, *first-order differentiation, higher-order differentiation* properties presented in Table 2 and Theorem 19, which involves more effort and user interaction. This clearly illustrates the benefits of our foundational formalization, presented in Section 4 of this paper. Moreover, the man-hours are calculated based on two factors. The first factor includes the number of lines of code per hour by a person with an average expertise and the second factor is the difficulty of the proof. For example, the proof lines for Theorem 32 and 36 are almost same, whereas the man-hours for both of the theorems are different, i.e., the man-hours for Theorem 32 are double in number with respect to man-hours for Theorem 36.

9.   Conclusions

In this paper, we proposed a formalization of Fourier transform in higher-order logic in order to perform the frequency domain analysis of the continuous-time systems. We presented the formal definition of Fourier transform and based on it, verified its classical properties, namely existence, linearity, time shifting, frequency shifting, modulation, time reversal, time scaling, differentiation and its relationship to Fourier Cosine, Fourier Sine and Laplace transforms. We also presented the formal verification of some commonly used functions. Next, we provided the formal verification of the frequency response of a generic n-order system. Lastly, in order to demonstrate the practical effectiveness of the proposed formalization, we presented a formal analysis of an audio equalizer and a MEMs accelerometer.

In the future, we aim to verify the two-dimensional Fourier transform (Bracewell, 1978), which is frequently applied for the frequency-domain analysis of many optical systems, electromagnetic theory and image processing algorithms. We also plan to formalize the inverse Fourier transform and verification of its properties, which would be very helpful to reason about the solutions to differential equations. This formalization can be further used in our project on system biology (Rashid et al., 2017), to formally analyze the differential equations corresponding to the reaction kinetic models of the biological systems.

Acknowledgements

References

A. Camilleri, M. G., Melham, T. F., 1986. Hardware Verification Using Higher-Order Logic. University of Cambridge, Computer Laboratory, UK.

Adams, M., 2012. Continuous-time Signals and Systems. ISBN:1347386708.

Avigad, J., Harrison, J., 2014. Formally Verified Mathematics. Communications of the ACM 57 (4), 66–75.

Beerends, R. J., Morsche, H. G., Van den Berg, J. C., Van de Vrie, E. M., 2003. Fourier and Laplace Transforms. Cambridge University Press.

Bracewell, R. N., 1978. The Fourier Transform and its Applications. McGraw-Hill.

Dougherty, G., 2009. Digital Image Processing for Medical Applications. Cambridge University Press.

Du, K.-L., Swamy, M. N. S., 2010. Wireless Communication Systems: from RF Subsystems to 4G Enabling Technologies. Cambridge University Press.

Durán, A. J., Pérez, M., Varona, J. L., 2013. Misfortunes of a Mathematicians' Trio using Computer Algebra Systems: Can We Trust? CoRR abs/1312.3270.

Eggers, P. E., Eggers, E. A., Bailey, B. Z., 2016. Wearable Apparatus, System and Method for Detection of Cardiac Arrest and Alerting Emergency Response. US Patent 20,160,174,857.

Fennelly, J., Ding, S., Newton, J., Zhao, Y., 2012. Thermal MEMs Accelerometers Fit Many Applications. Sensor Magazin, 18–20.

Galvin, G. J., Davis, T. J., MacDonald, N. C., 2001. Microelectromechanical Accelerometer for Automotive Applications. US Patent 6,199,874.

Gaskill, J. D., 1978. Linear Systems, Fourier Transforms, and Optics. Linear Systems, Fourier Transforms, and Optics by Jack D. Gaskill John Wiley and Sons 1.

Gaydecki, P., 2004. Foundations of Digital Signal Processing: Theory, Algorithms and Hardware Design. Institution of Engineering and Technology.

Girod, B., Stenger, A., Rabenstein, R., 2001. Signals and Systems. John Wiley.

Goodman, J. W., 2005. Introduction to Fourier Optics. Roberts and Company Publishers.

Hales, T. C., 2005. Introduction to the Flyspeck Project. Mathematics, Algorithms, Proofs 5021, 1–11.

Harisson, J., 2017. HOL-Light Transcendental Theory. https://github.com/jrh13/hol-light/blob/master/Multivariate/transcendentals.ml.

Harrison, J., 1996a. Formalized Mathematics. Tech. Rep. 36, Turku Centre for Computer Science, Finland.

Harrison, J., 1996b. HOL Light: A Tutorial Introduction. In: Formal Methods in Computer-Aided Design. Vol. 1166 of LNCS. Springer, pp. 265–269.

Harrison, J., 2009. Handbook of Practical Logic and Automated Reasoning. Cambridge University Press.

Harrison, J., 2013. The HOL Light Theory of Euclidean Space. Journal of Automated Reasoning 50 (2), 173–190.

Hasan, O., Tahar, S., 2015. Formal Verification Methods. Encyclopedia of Information Science and Technology, IGI Global Pub, 7162–7170.

Haykin, S., Van Veen, B., 2007. Signals and Systems. John Wiley & Sons.

Hsieh, C. Y., Wu, Y. M., Chin, T. L., Kuo, K. H., Chen, D. Y., Wang, K. S., Chan, Y. T., Chang, W. Y., Li, W. S., Ker, S. H., 2014. Low Cost Seismic Network Practical Applications for Producing Quick Shaking Maps in Taiwan. Terrestrial, Atmospheric & Oceanic Sciences 25 (5).

Hung, P. D., Hung, V. V., Quang, N. V., Vinh, T. Q., Thuy, N. P., 2004. Telepresence Virtual Control System for Robot using MEMS-based Sensors. In: International Conference on Mechatronic Technology. pp. 379–384.

Kaajakari, V., et al., 2009. Practical MEMS: Design of Microsystems, Accelerometers, Gyroscopes, RF MEMS, Optical MEMS, and Microfluidic systems. Small Gear Publishing.

Kuznetsov, A., Abutidze, Z., Portnov, B., Galkin, V., Kalik, A., 2011. Development of MEMS Sensors for Aircraft Control Systems. Gyroscopy and Navigation 2 (1), 59–62.

Maple, 2017. `http://www.maplesoft.com/products/maple/`.

MATLAB, 2017. `http://www.mathworks.com/products/matlab/`.

Oppenheim, A. V., Willsky, A. S., Hamid Nawab, S., 1996. Signals and Systems, 2nd Edition. Prentice Hall Processing series. Prentice Hall, Inc.

Papoulis, A., 1977. Signal Analysis. Vol. 2. McGraw-Hill.

Paulson, L., 1996. ML for the Working Programmer. Cambridge University Press.

Rashid, A., 2017. Formal Analysis of Continuous-time Systems using Fourier Transform. `http://save.seecs.nust.edu.pk/projects/factsft/`.

Rashid, A., Hasan, O., 2016. On the Formalization of Fourier Transform in Higher-order Logic. In: International Conference on Interactive Theorem Proving. Vol. 9807 of LNCS. Springer, pp. 483–490.

Rashid, A., Hasan, O., 2017. Formalization of Transform Methods using HOL Light. In: Conference on Intelligent Computer Mathematics. Vol. 10383 of LNAI. Springer, pp. 319–332.

Rashid, A., Hasan, O., Siddique, U., Tahar, S., 2017. Formal Reasoning about Systems Biology using Theorem Proving. PloS ONE 12 (3), e0180179.

Schumann, J. M., 2001. Automated Theorem Proving in Software Engineering. Springer Science & Business Media.

Siddique, U., Mahmoud, M. Y., Tahar, S., 2014. On the Formalization of Z-Transform in HOL. In: Interactive Theorem Proving. Vol. 8558 of LNCS. Springer, pp. 483–498.

Stark, H., 2012. Application of Optical Fourier Transforms. Elsevier.

Tan, L., Jiang, J., 2007. Fundamentals of Analog and Digital Signal Processing. AuthorHouse.

Taqdees, S. H., Hasan, O., 2013. 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, pp. 744–758.

Thibos, L. N., 2013. Fourier Analysis for Beginners. Available online at `http://www.opt.indiana.edu/VSG/Library/FourierBook/title.html`.

Thomas, R. E., Rosa, A. J., Toussaint, G. J., 2016. The Analysis and Design of Linear Circuits, Binder Ready Version. John Wiley & Sons.

Wolfram, S., 2015. The Mathematica Book, Wolfram Media, 2003 2.

Ziemer, R., Tranter, W. H., 2006. Principles of Communications: System Modulation and Noise. John Wiley & Sons.