Formalization of Fourier Transform using HOL-Light

Adnan Rashid and Osman Hasan
School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan
{adnan.rashid,osman.hasan}@seecs.nust.edu.pk

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 report 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 reversal, time scaling and differentiation in time domain, and its relationships with Fourier cosine, Fourier sine and Laplace transforms.

Keywords: Continuous-time Systems, Theorem Proving, Higher-order Logic, HOL-Light, Fourier Transform

1 Introduction

Fourier Transform [1] is a transform method, which converts a time varying function to its corresponding ω-domain representation, where ω is its corresponding angular frequency [2]. 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.

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 reversal, time scaling 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 transform methods based analysis of continuous-time systems has been done using the paper-and-pencil based analytical technique. However, due to the involvement of the 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. The other traditional analysis techniques are the computer-based methods, which include the numerical methods and the symbolic techniques. Some of the computer tools involved in these analysis are MATLAB [3], Mathematica [4] and Maple [5]. 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 [6]. Moreover, they also contain some unverified symbolic algorithms in their core [7], 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 [8] 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, expressed in an appropriate logic. The involvement of mathematical reasoning in the formal verification process and the mathematical nature of the system model and the desired property guarantees the accuracy of the analysis. Formal methods have been widely used for the verification of software [9] and hardware [10] systems and the formalization (or mathematical modeling) of classical mathematics [11, 12].

Higher-order-logic theorem proving [13] 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 [14] and used them to analyze an Infinite Impulse Response (IIR) filter. Similarly, *Hira et al.* formalized the Laplace transform [6] 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 [6], 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 [15]. 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 [16]. 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 theorem proving based analysis 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. We use the HOL-Light theorem prover [17] 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.

The rest of the report 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. Finally, Section 5 concludes the report.

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 [17] 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 [18]. 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, arithmetics, real numbers, transcendental functions and multivariate analysis 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 [19] 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
/\	and	Logical *and*
\/	or	Logical *or*
~	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

2.2 Multivariable Calculus Theories in HOL-Light

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 [20]. 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 2.1. Cx and ii
```
⊢ ∀ a. Cx a = complex (a, &0)
⊢ 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 2.2. Re, Im, lift and drop
```
⊢ ∀ z. Re z = z$1
⊢ ∀ z. Im z = z$2
⊢ ∀ x. lift x = (lambda i.  x)
⊢ ∀ x. drop x = x$1
```

The function `Re` accepts a complex number and returns its real part. Here, the notation `z$i` represents the i^{th} 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. Similarly, `drop` takes a 1-dimensional vector and returns its single element as a real number.

Definition 2.3. Exponential, Complex Cosine and Sine Functions
```
⊢ ∀ x. exp x = Re (cexp (Cx x))
⊢ ∀ z. ccos z = (cexp (ii * z) + cexp (--ii * z)) / Cx (&2)
⊢ ∀ 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 [21].

Definition 2.4. Vector Integral and Real Integral
```
⊢ ∀ f i. integral i f = (@y. (f has_integral y) i)
⊢ ∀ 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 $\texttt{i} : \mathbb{R}^N \to \mathbb{B}$, 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. Similarly, the function real_integral accepts integrand function $\texttt{f} : \mathbb{R} \to \mathbb{R}$ and a set of real numbers $\texttt{i} : \mathbb{R} \to \mathbb{B}$ 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 $[\mathsf{a}, \mathsf{b}]$ and real_interval $[\mathsf{a}, \mathsf{b}]$, respectively.

Definition 2.5. 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 $\texttt{f} : \mathbb{R}^1 \to \mathbb{R}^M$ and a $\texttt{net} : \mathbb{R}^1 \to \mathbb{B}$, 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. The function has_vector_derivati ve defines the same relationship in the relational form. Similarly, the function real_deriv ative accepts a function $\texttt{f} : \mathbb{R} \to \mathbb{R}$ 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 2.6. 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 $\texttt{f} : \mathbb{A} \to \mathbb{R}^M$ and returns l of data-type $\mathbb{R}^M$, i.e., the value to which f converges at the given net.

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:

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

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 3.1. Fourier Transform
⊢ ∀ w f. fourier_transform f w = integral UNIV (λt. cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t)

The function fourier_transform accepts a complex-valued function $\texttt{f} : \mathbb{R}^1 \to \mathbb{R}^2$ 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 $\texttt{cexp} : \mathbb{R}^2 \to \mathbb{R}^2$ 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 $\texttt{ii} * \texttt{Cx w}$, it is first converted into a real number by using drop and then it is converted to data-type $\mathbb{R}^2$ using Cx. Next, we use the vector function integral (Definition 2.4) 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 $\texttt{UNIV} : \mathbb{R}^1$, 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 [2]. 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 3.2. Fourier Exists
⊢ ∀ f. fourier_exists f ⇔
    (∀ a b. f piecewise_differentiable_on interval [lift a, lift b]) ∧
        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 3.1. 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 3.1 and 3.2 and Theorem 3.1 to verify some of the classical properties of Fourier transform in HOL-Light. The verification of these properties 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 **??** of this paper.

4.1 Integrability and Limit Existence of the Improper Integral

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 4.1. Integrability of Integrand of Fourier Transform Integral
⊢ ∀ f w. fourier_exists f ⇒ (λt. cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t) integrable_on UNIV

The proof of Theorem 4.1 starts by writing the region of integration, i.e., the whole real line UNIV : $\mathbb{R}^1$, as a union of positive real line (interval $[0, \infty)$) and negative real line (interval $(-\infty, 0]$):

Subgoal 10.1. Integrability on Positive and Negative Real Line
(λt. cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t) integrable_on {t | &0 <= drop t} ∧
(λt. cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t) integrable_on {t | drop t <= &0}

The above subgoal consists of two conjuncts. We explain the proof process of the first conjunct only as the proof of the other conjunct is based on the same reasoning. Using some well-established results of multi-variable calculus, along with some extensive complex arithmetic reasoning, the first conjunct of Subgoal 10.1 becomes:

Subgoal 10.2. Integrability of the Fourier Integrand on Positive Real Line

```
(∀a. (λt.  cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t) integrable_on interval [lift (&0),a]) ∧
(∃l. ((λa.  integral (interval [lift (&0),lift a])
         (λt. cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t)) → l) at_posinfinity)
```

The first conjunct of the above subgoal can be discharged by using integrability, continuity and some other properties of the exponential function.

Next, we split the complex-valued integrand, $f(t)e^{-i\omega t}$, of the second conjunct of the Subgoal 10.2 into its corresponding real and imaginary parts:

Subgoal 10.3. Complex-valued Integrand after Splitting into Real and Imaginary Parts

```
∃l. ((λa. integral (interval [lift (&0),lift a])
         (λt. Cx (Re (cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t))) +
         ii ∗
         integral (interval [lift (&0),lift a])
         (λt. Cx (Im (cexp (--((ii ∗ Cx w) ∗ Cx (drop t))) ∗ f t)))) → l) at_posinfinity
```

Next, we converted the above vector integral into a real-valued integral for both of the real and imaginary parts and then chose complex (k - m, k' - m') as the value for the existentially quantified variable:

Subgoal 10.4. Equivalent Real Integrals of the Corresponding Vector Integrals

```
((λt. real_integral (real_interval [&0,t])
      (λx. Re (cexp (--((ii ∗ Cx w) ∗ Cx (drop (lift x)))) ∗ f (lift x)))) → k - m) at_posinfinity ∧
((λt. real_integral (real_interval [&0,t])
      (λx. Im (cexp (--((ii ∗ Cx w) ∗ Cx (drop (lift x)))) ∗ f (lift x)))) → k' - m') at_posinfinity
```

Both the conjuncts can now be discharged based on the formally verified Integral Comparison Test [22] for improper integrals, and some other properties of the integrals and exponentials, along with some complex arithmetic reasoning.

4.2 Linearity

Linearity property is frequently used for the analysis of systems having composition of subsystems, which accept different scaled inputs. For the input functions f and g of any two subsystems and two complex numbers α and β, which are acting as the scaling factors, it can be mathematically expressed as follows:

$$\mathcal{F}[\alpha f(t) + \beta g(t)] = \alpha F(\omega) + \beta G(\omega) \tag{2}$$

We verify this property as the following theorem:

Theorem 4.2. Linearity of Fourier Transform

```
⊢ ∀ f g w a b. fourier_exists f ∧ fourier_exists g ⇒
   fourier_transform (λx.  a ∗ f x + b ∗ g x) w = a ∗ fourier_transform f w + b ∗ fourier_transform g w
```

where `a` : $\mathbb{C}$ and `b` : $\mathbb{C}$ are arbitrary constants acting as the scaling factors. The two assumptions ensure the fourier existence condition for both of the functions `f` and `g`. The proof of above theorem is based on Definitions 3.1 and 3.2, the linearity property of integration and Theorem 4.1 along with some arithmetic reasoning.

4.3 Time Shifting

The time shifting property of Fourier transform 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). These are mathematically expressed by the following two equations:

$$\mathcal{F}[f(t + t0)] = F(\omega)e^{+i\omega t0} \tag{3}$$

$$\mathcal{F}[f(t - t0)] = F(\omega)e^{-i\omega t0} \tag{4}$$

where $t_0 : \mathbb{R}$ is an arbitrary constant. We verify the above properties as the following two theorems:

Theorem 4.3. Time Advance or Left Shifting

```
⊢ ∀ f w t0. fourier_exists f ⇒
   fourier_transform (λt.  f (t + t0)) w = fourier_transform f w ∗ cexp ((ii ∗ Cx w) ∗ Cx (drop t0))
```

Theorem 4.4. Time Delay or Right Shifting

```
⊢ ∀ f w t0. fourier_exists f ⇒
   fourier_transform (λt.  f (t - t0)) w = fourier_transform f w ∗ cexp (--((ii ∗ Cx w) ∗ Cx (drop t0)))
```

The proofs of Theorems 4.3 and 4.4 are mainly based on Definitions 3.1 and 3.2, Theorem 4.1, properties of integration and complex exponential, and complex arithmetic reasoning along with the translation of the universal set property, i.e., `∀ a.  IMAGE (λt.  a +  t) UNIV = UNIV`.

4.4 Frequency Shifting

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 is mathematically represented by the following equation.

$$\mathcal{F}[e^{+i\omega_0 t} f(t)] = F(\omega - \omega_0) \tag{5}$$

where $\omega_0 : \mathbb{R}$ is an arbitrary constant. Similarly, the frequency left shifting (frequency advance) shifts the frequency signal to the left on the frequency axis. Mathematically, it can be represented by the following expression:

$$\mathcal{F}[e^{-i\omega_0 t} f(t)] = F(\omega + \omega_0) \tag{6}$$

We verify the above theorems as follows:

Theorem 4.5. Frequency Right Shifting

```
⊢ ∀ f w w0. fourier_exists f ⇒
   fourier_transform (λt. cexp ((ii * Cx (w0)) * Cx (drop t)) * f t) w = fourier_transform f (w - w0)
```

Theorem 4.6. Frequency Left Shifting

```
⊢ ∀ f w w0. fourier_exists f ⇒
   fourier_transform (λt. cexp (--(ii * Cx (w0)) * Cx (drop t)) * f t) w = fourier_transform f (w + w0)
```

4.5 Modulation

The modulation property of Fourier transform 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. The modulation property is a variant of frequency shifting.

The cosine based modulation property can be expressed as follows:

$$\mathcal{F}[cos(\omega_0 t)f(t)] = \frac{F(\omega - \omega_0) + F(\omega + \omega_0)}{2} \tag{7}$$

where $\omega_0 : \mathbb{R}$ is an arbitrary constant. Similarly, the modulation property based on the sine function is given by the following mathematical relation:

$$\mathcal{F}[sin(\omega_0 t)f(t)] = \frac{F(\omega - \omega_0) - F(\omega + \omega_0)}{2i} \tag{8}$$

We verify both of these modulation properties in HOL-Light as follows:

Theorem 4.7. Cosine Based Modulation

```
⊢ ∀ f w w0. fourier_exists f ⇒
        fourier_transform (λt.  ccos (Cx w0 * Cx (drop t)) * f t) w =
       (fourier_transform f (w - w0) + fourier_transform f (w + w0)) / Cx (&2)
```

Theorem 4.8. Sine Based Modulation

```
⊢ ∀ f w w0. fourier_exists f ⇒
        fourier_transform (λt.  csin (Cx w0 * Cx (drop t)) * f t) w =
        (fourier_transform f (w - w0) - fourier_transform f (w + w0)) / (Cx (&2) * ii)
```

4.6 Time Reversal

The time reversal property of Fourier transform of a function f is given as:

$$\mathcal{F}[f(-t)] = F(-\omega) \tag{9}$$

We verify this property in HOL-Light as the following theorem:

Theorem 4.9. Time Reversal

```
⊢ ∀ f w. fourier_exists f ⇒ fourier_transform (λt.  f (--t)) w = fourier_transform f (--w)
```

The proof is mainly based on Definitions 3.1 and 3.2 and the reasoning used to verify the second conjunct of Subgoal 10.2.

4.7 Time Scaling

The time scaling property of Fourier transform of a function f is given as:

$$\mathcal{F}[f(at)] = \frac{1}{|a|}F(\frac{\omega}{a}) \tag{10}$$

where $a : \mathbb{R}$ 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. We verify this property in HOL-Light as the following theorem:

Theorem 4.10. Time Scaling

```
⊢ ∀ 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)
```

4.8 Differentiation in Time Domain

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 and is given as:

$$\mathcal{F}[\frac{d}{dt}f(t)] = i\omega F(\omega) \tag{11}$$

which can be formally verified in HOL-Light as follows:

Theorem 4.11. First Order Differentiation in Time Domain

```
⊢ ∀ f w. fourier_exists f ∧
  fourier_exists (λt.  vector_derivative f (at t)) ∧ (∀t.  f differentiable at t) ∧
  ((λt.  f (lift t)) → vec 0) at_posinfinity ∧ ((λt.  f (lift t)) → vec 0) at_neginfinity
   ⇒ fourier_transform (λt.  vector_derivative f (at t)) w = ii * Cx w * fourier_transform f w
```

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 ϵ $\mathbb{R}$. 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 Theorem 4.11 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.

The Fourier transform of a n-times continuously differentiable function f is given as follows:

$$\mathcal{F}[\frac{d^n}{dt^n}f(t)] = (i\omega)^n F(\omega) \tag{12}$$

The above property is the foremost foundational property for analysing higher-order differential equations based on the Fourier transform and is verified using the following theorem:

Theorem 4.12. Higher Order Differentiation in Time Domain

```
⊢ ∀ 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 (lift t)) → vec 0) at_posinfinity) ∧
  (∀k. k < n ⇒ ((λt. higher_vector_derivative k f (lift t)) → vec 0) at_neginfinity) ⇒
   fourier_transform (λt.  higher_vector_derivative n f t) w = (ii * Cx w) pow n * fourier_transform f w
```

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 k^{th} derivative of f and $f^{(0)} =$ f. Finally, the conclusion of Theorem 4.12 is the Fourier transform of n^{th} order derivative of the function. The proof of above theorem is mainly based on induction on variable n along with Theorem 4.11.

4.9 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:

$$F_c(\omega) = \int_{-\infty}^{+\infty} f(t)cos(\omega t)dt \tag{13}$$

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 4.13. Fourier Transform of Even Function

```
⊢ ∀ f w. fourier_exists f ∧ (∀t. f (--t) = f t) ⇒ 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:

$$F_s(\omega) = \int_{-\infty}^{+\infty} f(t) sin(\omega t) dt \tag{14}$$

If the input function is an odd function, i.e., $f(-t) = -f(t)$ for all $t \epsilon \mathbb{R}$, 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 4.14. Fourier Transform of Odd Function

```
⊢ ∀ f w. fourier_exists f ∧ (∀t. f (--t) = --f t)
                    ⇒ 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.

4.10 Relationship with Laplace Transform

By restricting the complex-valued function $f : \mathbb{R}^1 \rightarrow \mathbb{R}^2$ 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:

$$F(s) = \int_{0}^{\infty} f(t) e^{-st} dt, \ s \ \epsilon \ \mathbb{C} \tag{15}$$

A formalized form of the Laplace transform is as follows [6]:

Definition 4.1. Laplace Transform

```
⊢ ∀ s f. laplace_transform f s =
 lim at_posinfinity (λb. integral (interval [lift (&0),lift (b)]) (λt. cexp (--(s ∗ Cx(drop t))) ∗ 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 [6]:

Definition 4.2. Laplace Exists

```
⊢ ∀ s f.  laplace_exists f s ⇔
        (∀ b.  f piecewise_differentiable_on interval [lift (&0),lift b] ) ∧
        (∃ M a.  Re s > drop a ∧ exp_order f M a)
```

The function `exp_order` in the above definition has been formally defined as [6]:

Definition 4.3. Exponential Order Function

```
⊢ ∀ f M a.  exp_order f M a ⇔ &0 < M ∧
                  (∀ t.  &0 <= t ⇒ norm (f (lift t)) <= M ∗ exp (drop a ∗ t))
```

If the function f is causal, i.e., $f(t) = 0$ for all $t < 0$ and the real part of Laplace variable $\mathtt{s} : \mathbb{R}^2$ is zero, i.e., *Re s = 0*, then the Fourier transform of function f is equal to Laplace transform, i.e., $(\mathcal{F} f)(Im\ s) = (\mathcal{L} f)(s) \mid_{Re\ s\ =\ 0}$ [23].

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

Theorem 4.15. Relationship with Laplace Transform

```
⊢ ∀ f s. laplace_exists f s ∧ (∀t. t IN {t | drop t <= &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 4.1. Alternative Representation of Laplace Transform

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

The above lemma presents an alternative representation of the Laplace transform, given in Definition 4.1. This relationship 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.

4.11 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:

$$\begin{aligned} \textit{Differential Equation} &= \sum_{k=0}^{n} \alpha_k \frac{d^k y}{dt^k} \\ &= \alpha_n \frac{d^n y}{dt^n} + \alpha_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + ... + \alpha_1 \frac{d^1 y}{dt^1} + \alpha_0 y \end{aligned} \tag{16}$$

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:

$$\mathcal{F}\Big(\sum_{k=0}^{n} \alpha_k \frac{d^k y}{dt^k}\Big) = F(\omega)\ \sum_{k=0}^{n} \alpha_k (i\omega)^k \tag{17}$$

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

Definition 4.4. 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 m^{th} 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 as the following theorem in HOL-Light:

Theorem 4.16. 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 (lift t)) → vec 0) at_posinfinity) ∧
   (∀k. k < n ⇒ ((λt. higher_vector_derivative k f (lift t)) → 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 Theorem 4.12. The conclusion of Theorem 4.16 is the Fourier transform of a n^{th}-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 3.1, 4.2 and 4.12 along with the following important lemma about the Fourier existence of the differential equation.

Lemma 4.2. Fourier Existence of Differential Equation

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

4.12 Area under a function

The Fourier transform can be used to evaluate the area under a function f using the following formula:

$$\int_{-\infty}^{\infty} f(t)dt = F(0) \tag{18}$$

where $F(0)$ in the above equation is the Fourier transform of a function f at $\omega = 0$. We verify the above theorem in HOL-Light as follows:

Theorem 4.17. Area under a function f

⊢ ∀ f. fourier_exists f ⇒ integral UNIV f = fourier_transform f (&0)

This completes our formalization of Fourier transform and formal verification of some of its classical properties.

5 Conclusions

In this report, 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.

References

[1] R. N. Bracewell. *The Fourier Transform and its Applications.* McGraw-Hill, 1978.

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

[3] MATLAB. `http://www.mathworks.com/products/matlab/`, 2016.

[4] Stephen Wolfram. The Mathematica Book, Wolfram Media, 2003. 2, 2015.

[5] Maple. `http://www.maplesoft.com/products/maple/`, 2016.

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

[7] Antonio J Durán, Mario Pérez, and Juan L Varona. Misfortunes of a Mathematicians' Trio using Computer Algebra Systems: Can We Trust? *CoRR*, abs/1312.3270, 2013.

[8] Osman Hasan and Sofiène Tahar. Formal Verification Methods. *Encyclopedia of Information Science and Technology, IGI Global Pub*, pages 7162–7170, 2015.

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

[10] M. Gordon A. Camilleri and T. F. Melham. *Hardware Verification Using Higher-Order Logic.* University of Cambridge, Computer Laboratory, UK, 1986.

[11] Thomas C Hales. Introduction to the Flyspeck Project. *Mathematics, Algorithms, Proofs*, 5021:1–11, 2005.

[12] Jeremy Avigad and John Harrison. Formally Verified Mathematics. *Communications of the ACM*, 57(4):66–75, 2014.

[13] John Harrison. *Handbook of Practical Logic and Automated Reasoning.* Cambridge University Press, 2009.

[14] Umair Siddique, Mohamed Yousri Mahmoud, and Sofiene Tahar. On the Formalization of Z-Transform in HOL. In *Interactive Theorem Proving*, volume 8558 of *LNCS*, pages 483–498. Springer, 2014.

[15] Joseph W Goodman. *Introduction to Fourier Optics.* Roberts and Company Publishers, 2005.

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

[17] J. Harrison. HOL Light: A Tutorial Introduction. In *Formal Methods in Computer-Aided Design*, volume 1166 of *LNCS*, pages 265–269. Springer, 1996.

[18] L.C. Paulson. *ML for the Working Programmer.* Cambridge University Press, 1996.

[19] J. Harrison. Formalized Mathematics. Technical Report 36, Turku Centre for Computer Science, Finland, 1996.

[20] John Harrison. The HOL Light Theory of Euclidean Space. *Journal of Automated Reasoning*, 50(2):173–190, 2013.

[21] John Harisson. HOL-Light Transcendental Theory. `https://github.com/jrh13/hol-light/blob/master/Multivariate/transcendentals.ml`, 2016.

[22] Jie Xiao. *Integral and Functional Analysis.* Nova Publishers, 2008.

[23] Roland E Thomas, Albert J Rosa, and Gregory J Toussaint. *The Analysis and Design of Linear Circuits, Binder Ready Version.* John Wiley & Sons, 2016.