Formal Analysis of Discrete-Time Systems using z-Transform

Umair Siddique
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
muh_sidd@ece.concordia.ca

Mohamed Yousri Mahmoud
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
mo_solim@encs.concordia.ca

Sofiène Tahar
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
tahar@ece.concordia.ca

Abstract

The computer implementation of a majority of engineering and physical systems requires the discretization of continuous parameters (e.g., time, temperature, voltage, etc.). Such systems are then called discrete-time systems and their dynamics can be described by difference or recurrence equations. Recently, there is an increasing interest in applying formal methods in the domain of cyber-physical systems to identify subtle but critical design bugs, which can lead to critical failures and monetary loss. In this paper, we propose to formally reason about discrete-time aspects of cyber-physical systems using the z-Transform, which is a mathematical tool to transform a time-domain model to a corresponding complex-frequency domain model. In particular, we present the HOL Light formalization of the z-Transform and difference equations along with some important properties such as linearity, time-delay and complex translation. An interesting part of our work is the formal proof of the uniqueness of the z-Transform. Indeed, the uniqueness of the z-Transform plays a vital role in reliably deducing important properties of complex systems. We apply our work to formally analyze a switched-capacitor interleaved DC-DC voltage doubler and an infinite impulse response (IIR) filter, which are important components of a wide class of power electronics, control and signal processing systems.

1 Introduction

We observe many continuous-time natural phenomena in our every day life, for instance the speed of a car, the temperature of a city and heart-beat are time varying quantities. Even though continuous-time quantities permeate in nature, we also observe many discrete-time quantities, e.g., maximum and minimum temperature in a city, average speed of traffic vehicles and a stock market index. It is therefore indispensable to design engineering systems which can detect and process these phenomena to achieve different functionalities. However, the continuous-time quantities cannot be processed directly using digital computing machines, which are suitable to deal with the discrete-time quantities. In practice, a continuous-time quantity is converted to a corresponding sampled version which coincides with the original quantity at some instant in time [6]. For example, a continuous-time signal can be sampled into a sequence of numbers where each number is separated from the next in time by a sampling period of T seconds as shown in Figure 1.

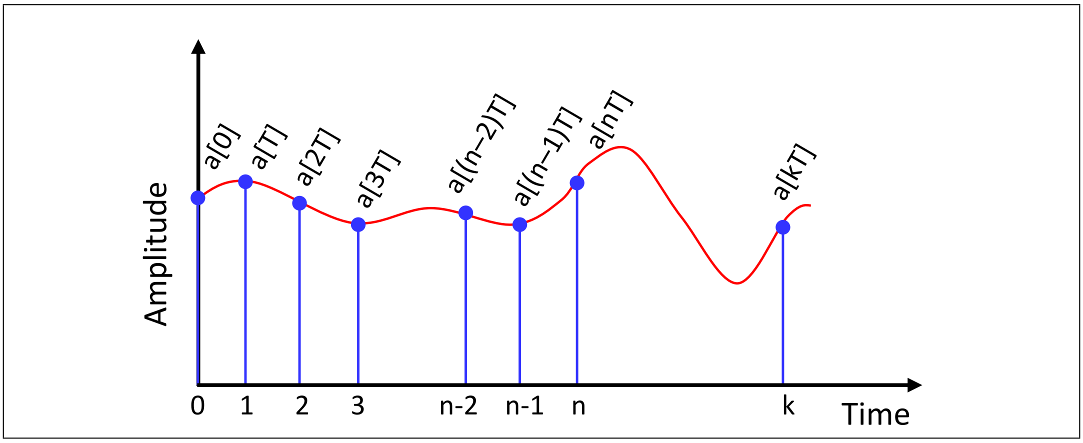

Figure 1: Sampling of a Continuous Signal

In general, the dynamics of engineering and physical systems are characterized by differential equations [33] and difference equations [7] in case of continuous-time and discrete-time, respectively. The complexity of these equations varies depending upon the corresponding system architecture (distributed, cascaded, hybrid etc.), the nature of input signals and the physical constraints. Transformation analysis is one of the most efficient techniques to mathematically analyze such complex systems. The main objective of transform method is to reduce complicated system models (i.e., differential or difference equations) into algebraic equations. The z-Transform [21] provides a mechanism to map discrete-time signals over the complex plane also called z-domain. This transform is a powerful tool to solve linear difference equations (LDE) by transforming them into algebraic operations in z-domain. Moreover, the z-domain representation of LDEs is also used for the transfer function analysis of corresponding systems. Due to these distinctive features, the z-Transform is one of the main core techniques available in physical and engineering system analysis software tools (e.g., MATLAB [20], Mathematica[19]) and is widely used in the design and analysis of signal processing filters [21], electronic circuits [7], control systems [8], photonic devices [5] and queueing networks [1].

The main idea of the z-Transform can be traced back to Laplace, but it was formally introduced by W. Hurewicz (1947) to solve linear constant coefficient difference equations [15]. Mathematically, the z-Transform can be defined as a function series which transforms a discrete time signal $f[n]$ to a function of a complex variable z, as follows:

$$X(z) = \sum_{n=0}^{\infty} f[n] z^{-n} \tag{1}$$

where $f[n]$ is a complex-valued function ($f : \mathbb{N} \to \mathbb{C}$) and the series is defined for those $z \in \mathbb{C}$ for which the series is convergent.

The first step in analyzing a difference equation (e.g., $x_{n+1} = kx_n(1 - x_n)$) using the z-Transform is to apply the z-Transform on both sides of a given equation. Next, the corresponding z-domain equation is simplified using various properties of the z-Transform, such as linearity, scaling and differentiation. The main task is to either solve the difference equation or to find a transfer function which relates the input and output of the corresponding system. Once the transfer function is obtained, it can be used to analyze some important aspects such as stability, frequency response and design optimization to reduce the number of corresponding circuit elements such as multipliers and shift registers.

Traditionally, the analysis of linear systems based on the z-Transform has been done using numerical computations and symbolic techniques [20, 19]. Both of these approaches, including paper-and-pencil proofs [21] have some known limitations like incompleteness, numerical errors and human-error proneness. In recent years, theorem proving has been actively used for both the formalization of mathematics (e.g., [11, 9]) and the analysis of physical systems (e.g., [30, 29]). For the latter case, the main task is to identify and formalize the underlying mathematical theories. In practice, four fundamental transformation techniques (i.e., the Laplace Transform (LT), the z-Transform (ZT), the Fourier Transform (FT), and the Discrete Fourier Transform (DFT)) are used in the design and development of linear systems. Interestingly, the Fourier transform and the Discrete Fourier transform can be derived from the Laplace Transform and the z-Transform, respectively. The formalization of the Laplace Transform and the Fourier Transform have been reported in [32] and [22] using the multivariate analysis libraries of HOL Light [12], with an ultimate goal of reasoning about differential equations and transfer functions of continuous systems. However, the formal proof of both the inverse Laplace Transform and the inverse Fourier Transform have not been provided in [32] and [22], which is necessary to reason about transformation from s-domain and ω-domain (where s and ω are LT and FT domain parameters, respectively) to the time-domain. The uniqueness and inverse of the z-Transform can be used to overcome this limitation by using the well-known Bilinear-Transformation of the z-domain and the s-domain [21]. The main relation amongst these four transformations is outlined in Figure 2.

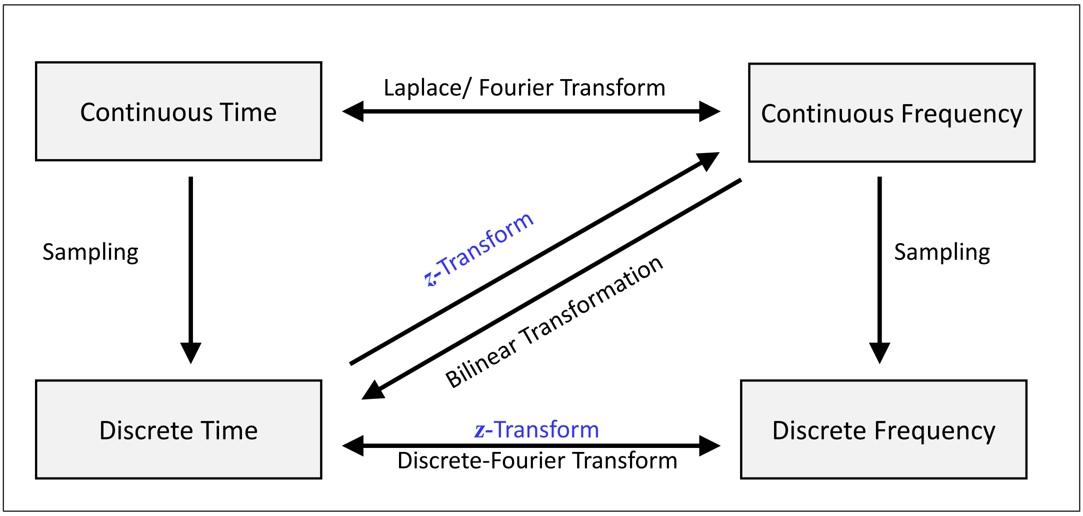

Figure 2: Discrete and Continuous Transformation Analysis

Nowadays, discrete-time linear systems are widely used in safety and mission critical domains (e.g., digital control of avionics systems and biomedical devices). We believe that there is a dire need for an infrastructure which provides the basis for the formal analysis of discrete-time systems within the sound core of a theorem prover. In this paper, we propose a formal analysis approach for the z-Transform based system models using a higher-order logic (HOL) theorem prover. The main idea is to leverage upon the high expressiveness of HOL to formalize Equation (1) and use it to verify classical properties of the z-Transform within a theorem prover. These foundations can be built upon to reason about the analytical solutions of difference equations or transfer functions. In [28], we presented the preliminary formalization of the z-Transform and its associated region of convergence (ROC). In this paper, however, we widen the scope by adding more interesting properties such as complex conjugation and initial value theorem of the z-Transform. We also provide the formally verified expressions for the z-Transform of commonly used mathematical functions (e.g., $\exp(x)$, $\sin(x)$ and $\cos(x)$). We then present the formalization of generic linear constant coefficient difference equations (LCCDE) along with the formal verification of corresponding z-Transform expressions by utilizing the key properties such as linearity of the z-Transform and ROC. A central part of the reported work is the formal proof of the uniqueness and inverse of the z-Transform, for which we also formalize the notion of an exterior region of a circle and its relation to ROC of the z-Transform. In order to demonstrate the practical effectiveness of the reported work, we present the formal analysis of a switched capacitor DC-DC power converter and an infinite impulse response (IIR) digital signal processing filter. The formalization reported in this paper has been developed in the latest version of the HOL Light theorem prover due to its rich multivariate analysis libraries [12]. The source code of our formalization is available for download [23] and can be utilized by other researchers and engineers for further developments and the analysis of more practical systems.

The rest of the paper is organized as follows: Section 2 describes some fundamentals of multivariate analysis libraries of the HOL Light theorem prover. Sections 3 and 4 present our HOL Light formalization of the z-Transform and the verification of its properties, respectively. Section 5 presents the formalization of difference equations and transfer functions. We describe the formal proof of the uniqueness of the z-Transform in Section 6. In Section 7, we present the analysis of a power-electronic DC-DC converter and IIR filter. Finally, Section 8 concludes the paper and highlights some future directions.

2 Preliminaries

In this section, we provide a brief introduction to the HOL Light formalization of some core concepts such as vector summation, summability, complex differentiation and infinite summation [12]. Our main intent is to introduce the basic definitions and notations that are used in the rest of the paper.

In the formalization of multivariate theory, an N-dimensional vector is represented as an $\mathbb{R}^N$ column matrix with individual elements as real numbers. All of the vector operations are then treated as matrix manipulations. Similarly, instead of defining a new type, complex numbers ($\mathbb{C}$) can be represented as $\mathbb{R}^2$. Most of the theorems available in multivariate libraries of HOL Light are verified for arbitrary functions with a flexible data-type of ($\mathbb{R}^M \to \mathbb{R}^N$). The injection from natural numbers to complex numbers can be represented by $\& : \mathbb{N} \to \mathbb{R}$. Similarly, the injection from real to complex numbers is done by $\texttt{Cx} : \mathbb{R} \to \mathbb{C}$. The real and imaginary parts of a complex number are represented by $\texttt{Re}$ and $\texttt{Im}$ both with type $\mathbb{C} \to \mathbb{R}$. The unary negation of x is represented as $-x$, where x can be real or a complex number.

The generalized summation over arbitrary functions is defined as follows:

Definition 1 (Vector Summation)**.**

$$\boxed{\vdash_{def}\ \forall \texttt{s f. vsum s f = (lambda i. sum s (}\lambda\texttt{x. f x\$i))}}$$

where $\texttt{vsum}$ takes two parameters $\texttt{s} : \texttt{A} \to \texttt{bool}$ which specifies the set over which the summation occurs and an arbitrary function $\texttt{f} : (\texttt{A} \to \mathbb{R}^{\texttt{N}})$. The function $\texttt{sum}$ is a finite summation over real numbers and accepts $\texttt{f} : (\texttt{A} \to \mathbb{R}^{\texttt{N}})$. For example, $\sum_{i=0}^{K} f(i)$ can be represented as $\texttt{vsum (0..K) f}$.

The traditional mathematical expression $\sum_{i=0}^{\infty} f(i) = L$ is defined in HOL Light as follows:

Definition 2 (Sums)**.**

$$\boxed{\begin{array}{l}\vdash_{def}\ \forall \texttt{s f l. (f sums l) s} \Leftrightarrow \\ \qquad\qquad \texttt{((}\lambda\texttt{n. vsum (s INTER (0..n)) f)} \longrightarrow \texttt{ l) sequentially}\end{array}}$$

where the types of the parameters are: $(\texttt{s} : \mathbb{N} \to \texttt{bool})$, $(\texttt{f} : \mathbb{N} \to \mathbb{R}^{\texttt{N}})$ and $(\texttt{L} : \mathbb{R}^{\texttt{N}})$.

We present the definition of the summability of a function $(\texttt{f} : \mathbb{N} \to \mathbb{R}^{\texttt{N}})$, which indeed represents that there exist some $(\texttt{L} : \mathbb{R}^{\texttt{N}})$ such that $\sum_{i=0}^{\infty} f(i) = L$.

Definition 3 (Summability)**.**

$$\boxed{\vdash_{def}\ \forall \texttt{f s. summable s f} \Leftrightarrow \ (\exists \texttt{l. (f sums l) s)}}$$

The limit of an arbitrary function can be defined as follows:

Definition 4 (Limit)**.**

$$\boxed{\vdash_{def}\ \forall \texttt{f net. lim net f = (}\varepsilon\texttt{l. (f} \longrightarrow \texttt{ l) net)}}$$

where the function $\texttt{lim}$ is defined using the Hilbert choice operator ε in the functional form. It accepts a $\texttt{net}$ with elements of arbitrary data-type $\texttt{A}$ and a function $(\texttt{f} : \texttt{A} \to \mathbb{R}^{\texttt{N}})$, and returns $(\texttt{L} : \mathbb{R}^{\texttt{N}})$ the value to which $\texttt{f}$ converges at the given $\texttt{net}$. In Definition 2, $\texttt{sequentially}$ represents a sequential net which describes the sequential evolution of a function, i.e., $f(i), f(i+1), f(i+2), \ldots,$ etc. This is typically used in the definition of an infinite summation. Note that $\texttt{nets}$ are defined as a bijective type in which domain is the set of two-parameter boolean functions, where we use the function $\texttt{mk\_net}$ to construct a net. The sequential nets are defined as $\texttt{mk\_net}\ \lambda\texttt{m n. m} \geq \texttt{n}$. According to this definition, we notice that the number $\texttt{a}$ that satisfies the property $\forall\texttt{n.}\ (\texttt{n} \geq \texttt{a})$, represents infinity. The continuous counterpart of the sequential net is `at_infinity`, which is defined as `mk_net` λ`x y`. `norm(x)` $\geq$ `norm(y)`. This is a generalized definition valid for any Euclidian space $\mathbb{R}^{\mathtt{N}}$. In case of real numbers, this simply reduces to `mk_net` λ`x y`. `x` $\geq$ `y`. The concept *tends to* ($\longrightarrow$) is formally defined as follows:

Definition 5.

$$\begin{array}{l}\vdash_{def} \forall \texttt{f l net. (f} \longrightarrow \texttt{ l) net} \Leftrightarrow \\ \qquad\qquad\qquad (\forall \texttt{e. \&0} < \texttt{e} \Rightarrow \texttt{eventually}(\lambda \texttt{x.dist (f x,l)} < \texttt{e) net)}\end{array}$$

We next present the definition of an infinite summation which is one of the most fundamental requirement in our development.

Definition 6 (Infinite Summation)**.**

$$\vdash_{def} \forall \texttt{f s. infsum s f} = (\varepsilon \texttt{l. (f sums l) s)}$$

where function `infsum` is defined using the Hilbert choice operator ε in the functional form. It accepts a parameter (`s` : `num` $\rightarrow$ `bool`) which specifies the starting point and a function (`f` : $\mathbb{N} \rightarrow \mathbb{R}^{\mathtt{N}}$), and returns (`L` : $\mathbb{R}^{\mathtt{N}}$) , i.e., the value at which infinite summation of `f` converges from the given `s`.

In some situations, it is very useful to specify infinite summation as a limit of finite summation (`vsum`). We proved this equivalence in the following theorem:

Theorem 1 (Infinite Summation in Terms of Sequential Limit)**.**

$$\vdash \forall \texttt{s f. infsum s f} = \texttt{lim sequentially}(\lambda \texttt{k.vsum (s INTER (0..k)) f)}$$

The differentiability of complex-valued functions is quite important in the development of the z-Transform, since it is the key element of proving uniqueness of the z-Transform. In HOL Light, a complex derivative is defined using the vector derivative as follows:

Definition 7 (Vector Derivative)**.**

$$\begin{array}{l}\vdash_{def} \forall \texttt{f f}' \texttt{ net. (f has\_complex\_derivative f}'\texttt{) net} \Leftrightarrow \\ \qquad\qquad\qquad \texttt{(f has\_derivative (}\lambda\texttt{x. f}' \texttt{ * x)) net}\end{array}$$

where a vector derivative (`has_derivative`) is defined as follows:

Definition 8 (Vector Derivative)**.**

$$\begin{array}{l}\vdash_{def} \forall \texttt{f f}' \texttt{ net. (f has\_derivative f}'\texttt{) net} \Leftrightarrow \\ \qquad\qquad\qquad \texttt{linear f}' \wedge \texttt{((}\lambda\texttt{y. inv (norm (y} - \texttt{netlimit net)) \%} \\ \qquad\qquad\qquad \texttt{(f y} - \texttt{(f (netlimit net)} + \\ \qquad\qquad\qquad\ \texttt{f}' \texttt{ (y} - \texttt{netlimit net))))} \longrightarrow \texttt{ vec 0) net}\end{array}$$

where `netlimit` of a `net` returns the supremum of the `net`.

The definition of a complex derivative can also be described in a functional form as follows:

Definition 9 (Complex Differentiation)**.**

```
⊢def ∀f x. complex_derivative f x =
        (εf′. (f has_complex_derivative f′) (at x))
```

This definition can further be generalized to formalize the concept of higher-order complex derivatives as described in the following definition:

Definition 10 (Higher-Order Complex Derivative)**.**

```
⊢def ∀f. higher_complex_derivative 0 f = f ∧
       (∀n. higher_complex_derivative (SUC n) f =
        complex_derivative (higher_complex_derivative n f))
```

Another important concept in complex analysis is holomorphic functions which are differentiable in the neighbourhood of every point in their domain. The formal definition of holomorphic functions in HOL Light is given as follows:

Definition 11 (Holomorphic Function)**.**

```
⊢def ∀f s. f holomorphic_on s ⇔
           (∀x. x IN s ⇒
           (∃f′. (f has_complex_derivative f′) (at x within s)))
```

3 Formalization of *z*-Transform

The unilateral z-Transform [16] of a discrete time function $f[n]$ can be defined as follows:

$$F(z) = \sum_{n=0}^{\infty} f[n] z^{-n} \tag{2}$$

where f is a function from $\mathbb{N} \to \mathbb{C}$ and z is a complex variable. Here, the definition that we consider has limits of summation from $n = 0$ to ∞. On the other hand, one can consider these limits from $n = -\infty$ to ∞ and such a version of the z-Transform is called two-sided or bilateral. This generalization comes at the cost of some complications such as non-uniqueness, which limits its practicality in engineering systems analysis. On the other hand, unilateral transform can only be applied to *causal* functions, i.e., $f[n] = 0$ for $\forall n. n < 0$. In practice, unilateral z-Transform is sufficient to analyze most of the engineering systems because their designs involve only causal signals [31]. For similar reasons, the authors of [32] formalized the unilateral Laplace transform rather than the bilateral version.

An essential issue of the z-Transform of $f[n]$ is whether the $F(z)$ even exists, and under what conditions it exists. It is clear from Equation (2) that the z-Transform of a function is an infinite series for each z in the complex plane or z-domain. It is important to distinguish the values of z for which the infinite series is convergent and the set of all those values is called the *region of convergence* (ROC). In mathematics and digital signal processing literature, different definitions of the ROC are considered. For example, one way is to express z in the polar form ($z = re^{j\omega}$) and then the ROC for $F(z)$ includes only those values of r for which the sequence $f[n]r^{-n}$ is absolutely summable. Unfortunately, to the best of our knowledge, this claim (i.e., absolute summability, e.g., [21]) is incorrect for certain functions, for example, $f[n] = \frac{1}{n}u[n-1]$ for which certain values of r result in convergent infinite series, but $x[n]r^{-n}$ is not absolutely summable.

Now, we have two distinct choices for defining the ROC: (1) values of z for which $F(z)$ is finite (or summable) and (2) values of z for which $x[n]z^{-n}$ is absolutely summable. Most of textbooks are not rigorous about the choice of the ROC and both of these definitions are widely used in the analysis of engineering systems. In this paper, we use the first definition of the ROC, which we can define mathematically as follows:

$$ROC = \{z \in \mathbb{C} : \exists k. \sum_{n=0}^{\infty} f[n]z^{-n} = k\} \tag{3}$$

In the above discussion, we mainly highlighted some arbitrary choices of using the definition of the z-Transform and its associated ROC. We formalize the z-Transform function (Equation 2) in HOL Light, as follows:

Definition 12 (z-Transform)**.**

```
⊢def ∀f z. z_transform f z = infsum (from 0) (λn. f n / z pow n)
```

where the `z_transform` function accepts two parameters: a function `f` $: \mathbb{N} \to \mathbb{C}$ and a complex variable `z` $: \mathbb{C}$. It returns a complex number which represents the z-Transform of `f` according to Equation (2).

We formalize the ROC of the z-Transform as follows:

Definition 13 (Region of Convergence)**.**

```
⊢def ∀f. ROC f = {z | ¬(z = Cx (&0)) ∧
                              summable (from 0) (λn. f n / z pow n)}
```

here, `ROC` accepts a function `f` $: \mathbb{N} \to \mathbb{C}$ and returns a set of non-zero values of variable z for which the z-Transform of `f` exists. In order to compute the z-Transform, it is mandatory to specify the associated ROC. We prove two basic properties of ROC which describe the linearity and scaling of the ROC, as follows:

Theorem 2 (ROC Linear Combination)**.**

```
⊢ ∀z a b f g.  z IN ROC f ∧ z IN ROC g  ⇒
               z IN ROC (λn. a * f n) INTER ROC (λn. b * g n)
```

Theorem 3 (ROC Scaling)**.**

```
⊢ ∀z a f. z IN ROC f ⇒  z IN ROC (λn. f n / a)
```

Theorem 2 describes that if z belongs to ROC f and ROC g then it also belongs to the intersection of both ROCs even though the functions f and g are scaled by complex parameters a and b, respectively. Similarly, Theorem 3 shows the scaling with respect to complex division by a complex number a.

4 Main Properties of the z-Transform

In this section, we use Definitions 12 and 13 to formally verify some of the classical properties of the z-Transform in HOL Light. The verification of these properties plays an important role in reducing the time required to analyze practical applications, as described later in Section 7.

4.1 Linearity of the z-Transform

The linearity of the z-Transform is a very useful property while handling systems composed of subsystems with different scaling inputs. Mathematically, it can be defined as:

If $\mathcal{Z}(f[n]) = F(z)$ and $\mathcal{Z}(g[n]) = G(z)$ then the following holds:

$$\mathcal{Z}(\alpha * f[n] \pm \beta * g[n]) = \alpha * F(z) \pm \beta * G(z) \tag{4}$$

The z-Transform of a linear combination of sequences is the same linear combination of the z-Transform of the individual sequences. We verify this property as the following theorem:

Theorem 4 (Linearity of z-Transform)**.**

```
⊢ ∀f g z a b. z IN ROC f ∧ z IN  ROC g ⇒
              z_transform (λx. a * f x + b * g x) z =
              a * z_transform f z + b * z_transform g z
```

where $\texttt{a} : \mathbb{C}$ and $\texttt{b} : \mathbb{C}$ are arbitrary constants. The proof of this theorem is based on the linearity of the infinite summation and Theorem 2.

4.2 Shifting Properties

The shifting properties of the z-Transform are mostly used in the analysis of digital systems and in particular in solving difference equations. In fact, there are two kinds of possible shifts: left shift ($f[n+m]$) or time advance and right shift ($f[n-m]$) or time delay. The main idea is to express the transform of the shifted signal (($f[n{+}m]$) or ($f[n-m]$)) in terms of its z-Transform ($F(z)$).

Left Shift of a Sequence: If $\mathscr{Z}(f[n])\ z = F(z)$ and k is a positive integer, then the left shift of a sequence can be described as follows:

$$\mathscr{Z}(f[n+k])\ z = z^k (F(z) - \sum_{n=0}^{k-1} f[n] z^{-n}) \tag{5}$$

We verify this theorem as follows:

Theorem 5 (Left Shift or Time Advance)**.**

```
⊢ ∀f z k. z IN ROC f ∧ 0 < k ⇒
           z_transform (λn. f (n + k)) z =
           z pow k * (z_transform f z −
                      vsum (0..k − 1) (λn. f n / z pow n))
```

The verification of this theorem mainly involves properties of complex numbers, summability of shifted functions and splitting an infinite summation into two parts as given by the following lemma:

Lemma 1 (Infsum Splitting)**.**

```
⊢ ∀f n m. summable (from m) f ∧ 0 < n ∧ m ≤ n ⇒
          infsum (from m) f = vsum (m..n−1) f + infsum (from n) f
```

Right Shift of a Sequence: If $\mathscr{Z}(f[n])\ z = F(z)$, and assuming $f(-n) = 0,\ \ \forall n = 1, 2, ..., m$, then the right shift or time delay of a sequence can be described as follows:

$$\mathscr{Z}(f[n-m])\ z = z^{-m} F(z) \tag{6}$$

We formally verify the above property as the following theorem:

Theorem 6 (Right Shift or Time Delay)**.**

```
⊢ ∀f z m. z IN ROC f ∧ is_causal f ⇒
          z_transform (λn. f (n−m)) z = z_transform f z / z pow m
```

Here, `is_causal` defines the causality of the function `f` in a relational form to ensure that $\texttt{f(n} - \texttt{m)} = \texttt{0}$, $\forall \texttt{m.n} < \texttt{m}$. The proof of this theorem also involves properties of complex numbers along with the following two lemmas:

Lemma 2 (Series Negative Offset).

```
⊢ ∀f k l. (f sums l) (from 0) ⇒  ((λn. f (n−k)) sums l) (from k)
```

Lemma 3 (Infinite Summation Negative Offset).

```
⊢ ∀f k. summable (from 0) f ⇒
        infsum (from 0) (λn. if k ≤ n then f (n−k) else vec 0) =
        infsum (from 0) f
```

As a direct application of the above results, we verify another important property called first-difference (which represents the difference between two consecutive samples of a signal), as follows:

Theorem 7 (First Difference).

```
⊢ ∀f z. z IN ROC f ∧ is_causal f ⇒
        z_transform (λn. f n − f (n − 1)) z =
        (Cx (&1) − z cpow Cx(−&1)) * z_transform f z
```

4.3 Scaling in the z-Domain or Complex Translation

The scaling property of the z-Transform is useful to analyze communication systems, such as the response analysis of modulated signals in z-domain. If $\mathcal{Z}(f[n])\ z = F(z)$, then two basic types of scaling can be defined as below:

$$\mathcal{Z}(h^n f[n])\ z = F(\frac{z}{h}) \tag{7}$$

$$\mathcal{Z}(\omega^{-n} f[n])\ z = F(\omega z) \tag{8}$$

If h is a positive real number, then it can be interpreted as shrinking or expanding of the z-domain. If h is a complex number with unity magnitude, i.e., $h = e^{j\omega_0}$, then the scaling corresponds to a rotation in the z-plane by an angle of ω_0. On the other hand, multiplication by ω^{-n} (Equation 8) shrinks the $z-$domain. Indeed, in the communication and signal processing literature, it is interpreted as frequency shift or translation associated with the modulation in the time-domain.

We verify the above theorems in HOL Light as follows:

Theorem 8 (Scaling in z-Domain)**.**

```
⊢ ∀f z h. inv h * z IN ROC f ∧ z IN ROC  f ⇒
          z_transform (λn. h cpow Cx (&n) * f n) z =
          z_transform (λn. f n) (inv h * z)
```

Theorem 9 (Scaling in z-Domain (Negative))**.**

```
⊢ ∀f z w. w * z IN ROC f ∧ z IN ROC f ⇒
          z_transform (λn. w cpow −Cx (&n) * f n) z =
          z_transform (λn. f n) (w * z)
```

4.4 Complex Differentiation

The differentiation property of the z-Transform is frequently used together with shifting properties to find the inverse transform. Mathematically, it can be expressed as:

$$\mathcal{Z}(n * f[n])\ z = -z * (\sum_{n=0}^{\infty} \frac{d}{dz}(f[n]z^{-n})) \tag{9}$$

We prove this property in the following theorem:

Theorem 10 (Complex Differentiation)**.**

```
⊢ ∀f z. &0 < Re z ∧ z IN ROC (Cx (&n) * f n) ⇒
         z_transform  (λn. Cx (&n) * f n) z = −z * infsum (from 0)
            (λn. complex_derivative (λz. f n * z cpow Cx (−&n)) z)
```

The proof of the above theorem requires the properties of complex differentiation, summability and complex arithmetic reasoning.

4.5 Complex Conjugation

The complex conjugation property provides the ease to manipulate the z-Transform of conjugated functions. The mathematical form of this property is as follows:

$$\mathcal{Z}(f^*[n])\ z = F^*(z^*) \tag{10}$$

where $f^*[n]$ represents the complex conjugate of function $f[n]$. The corresponding formal form of the complex conjugation is given as follows:

Theorem 11 (Complex Conjugation)**.**

```
⊢ ∀f z. cnj z IN ROC f ⇒
        z_transform (λn. cnj (f n)) z = cnj(z_transform f (cnj z))
```

4.6 The z-Transform of Commonly Used Functions

In real-world applications, the system is usually subject to a set of known input functions depending upon the dynamics and overall output response. It is quite handy to verify the z-Transform of such functions to simplify the reasoning while tackling practical applications using our formalization. In this regard, we verify the z-Transform expressions for most commonly used functions in signal processing and control systems. Table 1 summarizes these functions along with their mathematical form and corresponding z-Transform. In the following, we provide the formal definition and verification of the z-Transform of the Dirac-Delta function only whereas the verification of other functions can be found in the proof script [23].

Function Name	**Mathematical Notation**	**Z-Transform**
Dirac-Delta Function	$\delta[n-m]$	z^{-m}
Exponential	$\exp[-\alpha * n]$	$\frac{1}{1-\exp[-\alpha]z^{-1}}$
Complex Constant	a^n	$\frac{1}{1-az^{-1}}$
Sine	$\sin[\omega_0 n]$	$\frac{z^{-1}\sin[\omega_0]}{1-2z^{-1}\cos[\omega_0]+z^{-2}}$
Cosine	$\cos[\omega_0 n]$	$\frac{1-z^{-1}\cos[\omega_0]}{1-2z^{-1}\cos[\omega_0]+z^{-2}}$
Scaled Sine	$a^n \sin[\omega_0 n]$	$\frac{az^{-1}\sin[\omega_0]}{1-2az^{-1}\cos[\omega_0+a^2z^{-2}]}$
Scaled Cosine	$a^n \cos[\omega_0 n]$	$\frac{1-az^{-1}\cos[\omega_0]}{1-a2z^{-1}\cos[\omega_0+a^2z^{-2}]}$

Table 1: z-Transform of Commonly used Functions

Definition 14 (Dirac-Delta Function).

```
⊢def delta m = (λn. if n = m then Cx (&1) else Cx (&0))
```

Theorem 12 (The z-Transform of Dirac-Delta Function).

```
⊢ ∀z n. z_transform (delta m) z = inv z pow m
```

5 Formalization of Difference Equations

A difference equation characterizes the behavior of a particular phenomena over a period of time. Such equations are widely used to mathematically model complex dynamics of discrete-time systems. Indeed, a difference equation provides a formula to compute the output at a given time, using present and future inputs and past output as given in the following example:

$$y[k] - 5y[k-1] + 6y[k-2] = 3x[k-1] + 5x[k-2] \tag{11}$$

In the perspective of engineering systems, a difference equation is concerned with the generation of a sequence of control outputs $x[n]$ given a sampled sequence of the system inputs $y[n]$. Generally, it is important to determine the control output at a sample instance n based on the sampled system input at the sample instance n and a finite number of previous sampled outputs. Mathematically, it can be written as follows:

$$y[n] = f(x[n], x[n-1], x[n-2], \dots, x[n-m], y[n-1], y[n-2], ..., y[n-k]) \tag{12}$$

There is an infinite number of ways the $m + k - 1$ values on the right-hand side of the above equation can be combined to form $y(n)$. We consider the practical case where the right-hand side of the above equation involves a linear combination of the past samples of the outputs and control inputs, which can be described as follows:

$$y[n] = \sum_{i=1}^{N} \alpha_i y[n-i] + \sum_{i=0}^{M} \beta_i x[n-i] \tag{13}$$

where α_i and β_i are input and output coefficients. The output $y[n]$ is a linear combination of the previous N output samples, the present input $x[n]$ and M previous input samples. Here, α_i and β_i are considered as constants (either complex ($\mathbb{C}$) or real ($\mathbb{R}$)) due to which the Equation (13) is called Linear Constant Coefficient Difference Equation (LCCDE). For a given N^{th} order difference in terms of a function $f[n]$, its z-Transform is given as follows:

$$\mathcal{Z}(\sum_{i=0}^{N} \alpha_i f[n-i])\ z = F(z) \sum_{i=0}^{N} \alpha_i z^{-i} \tag{14}$$

Applying the z-Transform on both sides of Equation (13) results in an important mathematical form describing the relation among the coefficients of $x[n]$ and $y[n]$, called *transfer function* or *system function*, given as follows:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\displaystyle\sum_{i=0}^{M} \beta_i z^{-i}}{1 - \displaystyle\sum_{i=1}^{N} \alpha_i z^{-i}} \tag{15}$$

In order to build the reasoning support for LCCDE in HOL Light, we formalize the N^{th} difference as follows:

Definition 15 (N^{th} Difference).

```
⊢def ∀N alst f x. nth_difference alst f N x =
                 vsum (0..N) (λt. EL t alst * f (x − t))
```

The function `nth_difference` accepts the order (`N`) of the difference equation, a list of coefficients `alst`, function `f` and the variable x. It utilizes the functions `vsum s f` and `EL i L`, which return the vector summation and the i^{th} element of a list `L`, respectively, to generate the difference equation corresponding to the given parameters.

Next, we formalize a general LCCDE (i.e., Equation (13)) as follows:

Definition 16 (Linear Constant Coefficient Difference Equation (LCCDE)).

```
⊢def ∀y M x N n. LCCDE x y alist blist M N n ⇔
                y n = nth_difference alist y M n +
                      nth_difference blist x N n
```

Now equipped with these formal definitions, our next step is to verify the z-Transform of the N^{th}-difference (Definition 15) which is one of the most important results of our formalization.

Theorem 13 (z-Transform of N^{th}-Difference).

```
⊢ ∀f lst N z. z IN ROC f ∧ is_causal f ⇒
             z_transform (λx. nth_difference lst f N x) z =
             z_transform f z * vsum (0..N)
                               (λn. z cpow −Cx (&n) * EL n lst)
```

The proof of Theorem 13 is based on induction on the order of the difference and Theorems 2 and 4 along with the following important lemma about the summability of N^{th}-difference equation:

Lemma 4 (Summability of Difference Equation).

```
⊢ ∀N a_lst f. z IN ROC f ∧ is_causal f ⇒
             z IN  ROC (λx. nth_difference a_lst f N x)
```

In order to verify the transfer function of the LCCDE (Equation (15)), we need to ensure that the input and output functions should be causal as described in Section 3. Another important requirement is to ensure that there are no values of z for which the denominator is 0, such values are called poles of that transfer function. We package these conditions in the following definitions:

Definition 17 (Causal System Parameters).

```
⊢def is_causal_lccde x y ⇔  is_causal x ∧ is_causal y
```

Definition 18 (LCCDE ROC)**.**

```
⊢def ∀x y M alst. LCCDE_ROC x y M alst =
                 (ROC x) INTER (ROC y) DIFF
                 {z | Cx (&1) − vsum (1..M)
                     (λn. EL n alst * z cpow −Cx (&n)) =  Cx(&0)}
```

Here, the function is_causal_lccde takes two parameters, i.e., input and output, and ensures that both of them are causal. In Definition 18, LCCDE_ROC specifies the region of convergence of the input and output functions, which is indeed the intersection of ROC x and ROC y, excluding all poles of the transfer function. The function DIFF represents the difference of two sets, i.e., $A \setminus B = \{z \mid z \in A \wedge z \notin B\}$.

Next, we present the formal verification of the transfer function as given in Equation 15.

Theorem 14 (LCCDE Transfer Function)**.**

```
⊢ ∀x y alst blst M N.
        z IN LCCDE_ROC x y M alst ∧
        is_causal_lccde x y ∧
        (∀n. LCCDE x y alist blist M N n) ⇒
        z_transform y z / z_transform x z =
        vsum (0..N) (λn. z cpow −Cx (&n) * EL n blst) /
        (Cx (&1) − vsum (1..M) (λn. z cpow −Cx(&n) * EL n alst))
```

The first and second assumptions describe the region of convergence for LCCDE and the causality of the input and output. The last assumption gives the time-domain model of the LCCDE. The proof of this theorem is mainly based on the properties of the z-Transform such as linearity (Theorem 4), time-delay (Theorem 6) and summability of difference equation (Lemma 4). This is a very useful result to simplify the reasoning for the LCCDE of any order.

6 Uniqueness of the z-Transform

One of the most critical aspects of transformation based analysis of discrete-time systems is to be able to obtain the time-domain expressions from z-domain parameters. The inverse transformation is very important to reliably deduce the properties of the underlying system because the actual implementation is done in the time-domain. The inversion of bilateral z-Transform $X(z)$ to its corresponding time domain function $x[n]$ is not unique due to the existence of infinitely many ROCs for one function. However, the uniqueness of unilateral z-Transform (that we have formalized in our work) can be proved considering the nature of the ROC which is always the exterior region of a circle as shown in Figure 3. Mathematically, the uniqueness of the z-Transform can be described as follows:

$$\mathcal{Z}(f[n]) = \mathcal{Z}(g[n]) \Leftrightarrow f = g \tag{16}$$

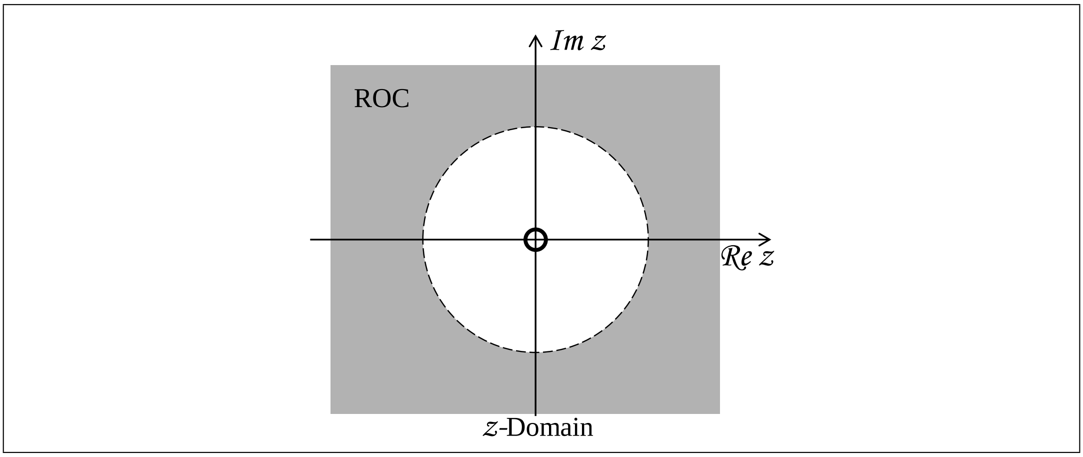

Figure 3: Region of Convergence (ROC) for Inverse z-Transform

The proof of the uniqueness of the z-Transform can be divided into two subgoals, i.e., forward and backward implications as follows:

$$f = g \Longrightarrow \mathcal{Z}(f[n]) = \mathcal{Z}(g[n]) \tag{17}$$

$$\mathcal{Z}(f[n]) = \mathcal{Z}(g[n]) \Longrightarrow f = g \tag{18}$$

The first subgoal is straight forward and can be proved by the definition of the z-Transform. However, the second subgoal requires the reconstruction of the original function $f[n]$ from the transformed function $\mathcal{Z}(f[n])$ or $(F(Z))$. There are two main methods for obtaining such reconstruction: First, a sequence that consists of the coefficients of the Laurent series of $F(z)$, which is given by the following equation [10]:

$$f(k) = \frac{1}{2\pi i} \oint_C F(z)^{k-1} dz \quad (k = 0, 1, 2, ...) \tag{19}$$

where the path of integration C is a circle of radius $r > \rho$ traversed in the anticlockwise direction. The second method involves the higher-order complex derivative of the infinite summation (i.e., the z-Transform $F(z)$) at origin ($z = 0$), given as follows [10]:

$$f(k) = \frac{1}{k!}(\frac{d^k}{dz^k}F(\frac{1}{z}))_{z=0} \quad (k = 0, 1, 2, ...) \tag{20}$$

Interestingly, the multivariate analysis libraries of HOL Light are rich enough to tackle both proofs involving the path integrals and the higher-order complex derivatives of a complex series. However, we have chosen the second methods due to the availability of some important lemmas in HOL Light as described in the sequel.

6.1 Formal Proof of Uniqueness

The unilateral z-Transform is unique for the ROC which forms an exterior of a circle excluding the centre as shown in Figure 3. We formally define the exterior region of a circle as follows:

Definition 19 (Exterior of Circle)**.**

```
⊢def ∀s. exterior_circle s ⇔
       (∃R. &0 < R ∧ (∀z. R < dist (z,Cx (&0)) ⇒  z IN s))
```

where `exterior_circle` accepts a set of complex elements ($\texttt{s} : (\texttt{real}^2 \rightarrow \texttt{bool})$) which forms an exterior region of a circle.

We verify three important properties describing the relation between the ROC of the z-Transform and the exterior of a circle.

- If the ROCs of the two functions f and g are exterior regions of a circle, then the intersection of their ROCs will also form an exterior circle.

```
⊢ ∀f g. exterior_circle (ROC f) ∧ exterior_circle (ROC g) ⇒
        exterior_circle (ROC f INTER ROC g)
```

- If a function f is summable, then its ROC will always form an exterior region of a circle.

```
⊢ ∀f. summable (from 0) f ⇒  exterior_circle (ROC f)
```

- If a function f is decaying over time, then its ROC will always be an exterior region of a circle.

```
⊢ ∀f c N. c < &1 ∧ (∀n. n ≥ N ⇒
          norm (f (SUC n)) ≤ c * norm (f n)) ⇒
          exterior_circle (ROC f)
```

We next prove the inverse transform function given in Equation 20.

Theorem 15 (Inverse z-Transform).

```
⊢ ∀f n. exterior_circle (ROC f) ⇒
        f n = higher_complex_derivative n
              (λz. z_transform f (inv z)) (Cx(&0)) / Cx(&(FACT n))
```

where the proof of Theorem 15 is done using the higher-order derivatives of a power series which is already available in HOL Light, as given in the following form:

Lemma 5 (Higher-Order Derivative of Power Series).

```
⊢ ∀f c r n k. &0 < r ∧ n IN k ∧
             (∀w. dist (w,z) < r ⇒
                 ((ı. c i * (w - z) pow i) sums f w) k) ⇒
             higher_complex_derivative n f z / Cx(&(FACT n)) = c n
```

Finally, we prove the uniqueness of the z-Transform based on Theorem 15 and Lemma 5 along with some complex arithmetic reasoning.

Theorem 16 (Uniqueness of the z-Transform).

```
⊢ ∀f g. exterior_circle (ROC f) ∧ exterior_circle (ROC g) ⇒
       (z_transform f = z_transform g ⇔  f = g)
```

6.2 Initial Value Theorem of the z-Transform

In many situations, it is desirable to compute the initial value of the function from its z-Transform. This is mainly achieved by using the famous initial value theorem of the z-Transform, which states that if the z-Transform of $x[k]$ is $X(z)$ and if $\lim_{z\to\infty} X(z)$ exists, then the initial value of $x[k]$ (i.e., $x[0]$) can be obtained from the following limit:

$$x(0) = \lim_{z\to\infty} X(z) \tag{21}$$

Theorem 17 (Initial Value Theorem).

```
⊢ ∀f. exterior_circle (ROC f) ⇒
      f 0 = lim at_infinity (λz. z_transform f z)
```

The proof of Theorem 17 is mainly based on the concepts about the differentiability, continuity and theory of holomorphic functions, as described in the following two lemmas (which are available in the HOL Light multivariate theory).

Lemma 6 (Complex Differentiability Implies Continuity).

```
⊢ ∀f x. f complex_differentiable at x ⇒  f continuous at x
```

Lemma 7 (Holomorphic Implies Differentiability).

```
⊢ ∀f s x. f holomorphic_on s ∧ open s ∧ x IN s ⇒
          f complex_differentiable at x
```

7 Applications

In order to illustrate the utilization and effectiveness of the reported formalization, we present the formal analysis of a couple of real-world applications namely power converters and digital filters which are widely used systems in the domain of power electronics and digital signal processing, respectively.

7.1 Formal Analysis of Switched-Capacitor Power Converter

In the last decade, very-large scale integrated (VLSI) systems industry has revolutionized many fields of physical sciences and engineering including communication, mobile devices and health-care. However, increased density of integrated chips resulted in high power dissipation which is known as energy crisis in VLSI industry. In order to overcome this issue, power management techniques can be applied at the system, circuit or device level depending on the system complexity and nature of the device operation. The system level power management techniques are used to identify optimal operating conditions by power sensing and power management. DC-DC converter [17] is one of the most important circuit level power management modules which convert an unregulated input DC voltage into an output voltage. Mainly, integrated DC-DC converters can be divided into three classes namely linear regulators, switch mode power converters and switched-capacitor power converters [17]. In this paper, we aim at formal modeling and analysis of switched-capacitor (SC) DC-DC converters due to their robustness and wide application domain [14].

7.1.1 Mathematical Modeling of SC Power Converter

In the design and modeling of any kind of power converter, it is very critical to obtain the transfer function (the input-output relation) to analyze the overall system design, system stability and desired power-gain. Generally, power electronics engineers obtain the power stage transfer function of switch mode and SC power converters using the z-Transform. Figure 4 outlines the system architecture of the interleaved SC power converter. The power stage is a cross-coupled voltage doubler

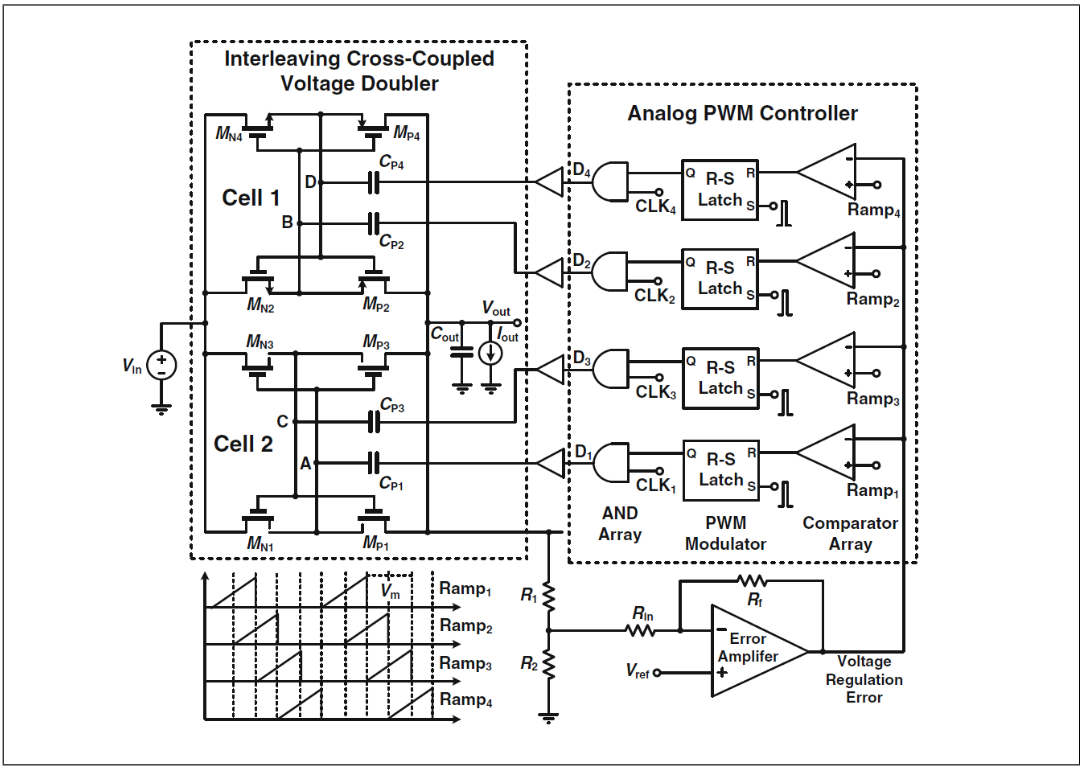

Figure 4: System Architecture of the Interleaved SC Power Converter [17]

that is regulated using an analog pulse-width modulation (PWM) controller. We can briefly describe its operation as follows: Initially, V_{out} is scaled down with the aid of a resistive voltage divider. This scaled voltage is then compared with the desired reference voltage V_{ref} and the corresponding voltage regulation error is determined and amplified by the error amplifier. The output of the error amplifier is then used to determine the output-input ratio of each charge pump sub-cell [17].

In order to derive the transfer function of the cross-coupled voltage doubler, Figure 5 describes the charge and discharge process of one charge pump cell. The charge pump operates in a full charge mode in which the current delivered by the pumping capacitors C_{Pi} at the end of each switching interval drops to a very low level, in comparison to its peak value. Since the two cross-coupled cells do not exchange charge or power at any instant during their operation, they can be modeled as separate elements. Finally, the overall operation can be modelled by the following six equations:

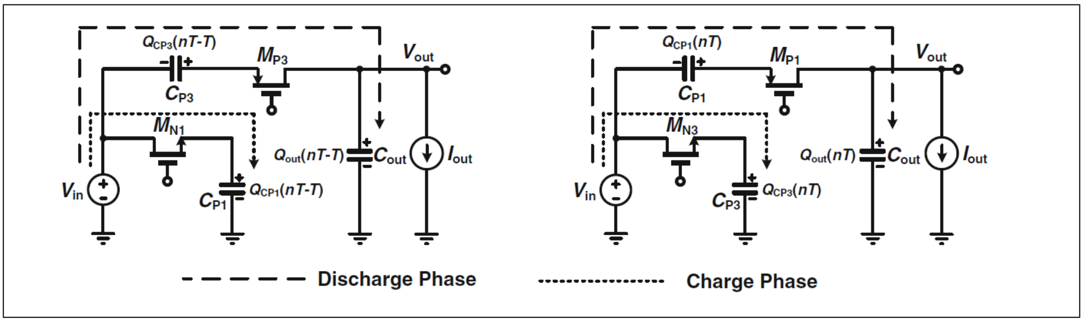

Figure 5: Charge and Discharge Phases for Interleaving SC Power Converter [17]

$$Q_1(n) = C_p(V_{out}(n) - V_{in}(n)) \tag{22}$$

$$Q_3(n) = C_p V_{in}(n) \tag{23}$$

$$Q_{out}(n) = C_{out} V_{out}(n) \tag{24}$$

$$Q_1(n-1) = C_p V_{in}(n-1) \tag{25}$$

$$Q_3(n-1) = C_p(V_{out}(n-1) - V_{in}(n-1)) \tag{26}$$

$$Q_{out}(n-1) = C_{out} V_{out}(n-1) \tag{27}$$

where Q_i, represents the charge stored at different nodes in the circuit, whereas V_{in} and V_{out} represent the input and output of the voltage doubler. The total charge transfer can be described as follows:

$$2 * (Q_1(n-1) - Q_1(n) + Q_3(n) - Q_3(n-1)) + Q_{out}(n-1) - Q_{out}(n) = \frac{T_s}{2}[\frac{V_{out}(n-1)}{R_{out}} + \frac{V_{out}(n)}{R_{out}}] \tag{28}$$

where R_{out} is output load resistor.

Using Equations ((22)-(28)) and the z-Transform results in the following transfer function:

$$\frac{V_{out}(z)}{V_{in}(z)} = \frac{4C_p(1+z^{-1})}{(2C_p - C_{out} + \frac{Ts}{2R_{out}})(\frac{(2C_p+C_{out}-\frac{Ts}{2R_{out}})}{(2C_p-C_{out}+\frac{Ts}{2R_{out}})} + z^{-1})} \tag{29}$$

Finally, letting $z = 1$ and $Ts = 0$, results in the DC conversion gain which should be consistent with the gain of an ideal voltage doubler, i.e., 2, as follows:

$$\left[\frac{V_{out}(z)}{V_{in}(z)}\right]_{z=1,Ts=0} = 2 \tag{30}$$

7.1.2 Formal Verification of the Transfer Function and DC Conversion Gain

Our main goal is to verify the transfer function of the voltage doubler (Equation (29) and the DC conversion gain (Equation (30), which are two critical requirements in the correct operation of the interleaved SC power converters. We formalize Equations ((22)-(28)) in HOL Light as follows:

Definition 20 (Voltage Doubler Model).

```
⊢def sc_voltage_doubler Q1 Q3 Qout Cp Cout Vin Vout ⇔
     (∀n. Q1 n = Cp * (Vout n − Vin n) ∧
     Q3 n = Cp * Vin n ∧ Qout n = Cout * Vout n ∧
     Q1 (n − 1) = Cp * Vin (n − 1) ∧
     Q3 (n − 1) = Cp * (Vout (n − 1) − Vin (n − 1)) ∧
     Qout (n − 1) = Cout * Vout (n − 1))
```

where the three variables `Q1`, `Q3` and `Qout` represent the values of the stored charge at different nodes. The parameters `Cp` and `Cout` represent the capacitors, whereas `Vin` and `Vout` represent the input and output voltages, respectively. The function `sc_voltage_doubler` returns the corresponding model of the voltage doubler corresponding to Equations ((22)-(27)). We next formally define the total transfer charge (Equation (28) as follows:

Definition 21 (Total Transfer Charge).

```
⊢def ∀Q1 Q3 Qout Ts Vout n Rout.
      total_charge_transfer Q1 Q3 Qout Vout Rout Ts n ⇔
      Cx(&2) * (Q1 (n − 1) − Q1 n + Q3 n − Q3 (n − 1)) +
      Qout (n − 1) − Qout n =
      Cx Ts / Cx(&2) * (Vout (n − 1) − Vout n) / Cx Rout
```

We next verify the transfer function of the SC Voltage doubler as follows:

Theorem 18 (SC Voltage Doubler Transfer Function).

```
⊢ ∀Q1 Q3 Qout n Vin Vout Cp Cout Ts Rout z.
     [A1] sc_voltage_doubler Q1 Q3 Qout Cp Cout Vin Vout ∧
     [A2] total_charge_transfer Q1 Q3 Qout Vout Rout Ts n ∧
     [A3] sc_parameters_constraints Cp Rout Cout Ts z ∧
     [A4] z IN ROC Vin ∧ z IN ROC Vout ∧
     [A5] is_causal Vin ∧ is_causal Vout ⇒
          transfer_function Vin Vout z =
          (Cx(&4) * Cp * (Cx(&1) + z cpow −Cx(&1))) /
          ((Cx(&2)* Cp − Cout + Cx Ts / (Cx(&2) * Cx Rout)) *
          ((Cx(&2)* Cp + Cout − Cx Ts / (Cx(&2) * Cx Rout)) /
          (Cx(&2) * Cp − Cout +
                    Cx Ts / (Cx(&2) * Cx Rout)) + z cpow −Cx(&1)))
```

where assumptions A1 and A2 describe the function of the SC voltage doubler and total transfer charge, respectively. The assumption A3 describes the constraints among the parameters of the SC voltage doubler so that the transfer function is well defined (i.e., there are no poles at which it becomes undefined). The assumptions A4 and A5 ensure that the input and output voltages are causal functions and form valid ROCs. The function `transfer_function` takes an input function `x`, an output function `y` and a z-domain parameter `z:complex` and returns the z-domain transfer function `z_transform y z / z_transform x z`.

Finally, we utilize Theorem 18 to verify the corresponding DC conversion gain of the voltage doubler configuration as follows:

Theorem 19 (SC Voltage Doubler Transfer Function).

```
⊢ ∀Q1 Q3 Qout n Vin Vout Cp Cout Rout.
     [A1] sc_voltage_doubler Q1 Q3 Qout Cp Cout Vin Vout ∧
     [A2] total_charge_transfer Q1 Q3 Qout Vout Rout (&0) n ∧
     [A3] sc_parameters_constraints Cp Rout Cout (&0) z ∧
     [A4] summable (from 0) Vin ∧ summable (form 0) Vout
     [A5] is_causal Vin ∧ is_causal Vout ⇒
          transfer_function Vin Vout Cx(&1) = Cx(&2)
```

In this application, we present the design of an interleaved cross-coupled SC voltage doubler, which is regulated using an analog PWM control scheme. We demonstrate the use of our formalization of the z-Transform and its properties by the formal modeling and verification of the SC interleaved cross-coupled SC voltage doubler. Similar analysis steps can be followed to analyze more converter configurations such as the monolithic SC power converter and the charge pump [17].

7.2 Formal Analysis of Infinite Impulse Response Filters

Digital filters are fundamental components of almost all signal processing and communication systems. The main functionality of such components are to: 1) limit a signal within a given frequency band; 2) decompose a signal into multiple bands; and 3) model the input-output relation of complicated systems such as mobile communication channels and radar signal processing. Digital filters can be used for the performance specifications which are very difficult to achieve by analog filters. Moreover, the functionality of digital filters can be controlled using software applications. Due to these features, such filters are widely used in adaptive filtering applications in telecommunications, speech recognition and biomedical devices.

An impulse response of a system describes its behavior under an external change (mathematically, this describes the system response when the Dirac-Delta function is applied as an input [21]). Infinite impulse response (IIR) filters have an impulse response function which is non-zero over an infinite length of time. In practice, IIR filters are implemented using the feedback mechanism, i.e., the present output depends on the present input and all previous input and output samples. Such an architecture requires delay elements due to the discrete nature of input and output signals. The highest delay used in the input and the output function is called the order of the filter. The time-domain difference equation describing a general M^{th} order IIR filter, with N feed forward stages and M feedback stages, is shown in Figure 6.

Mathematically, it can be described as:

$$y[n] - \frac{1}{\alpha_0} \sum_{i=1}^{M} \alpha_i y[n-i] = \sum_{i=0}^{N} \beta_i x[n-i] \tag{31}$$

where α_i and β_i are input and output coefficients (Note that, $\alpha_0 = 1$ in most practical situations [21]). In case of a time-invariant filter, α_i and β_i are considered constants (either complex ($\mathbb{C}$) or real ($\mathbb{R}$)) to obtain the filter response according to the given specifications.

Our main objective is to formally verify the frequency response of an IIR filter which is given as follows:

$$H(\omega) = \left[\frac{Num}{Den}\right] * \exp\left(j * Arg\left[\frac{Num}{Den}\right]\right) \tag{32}$$

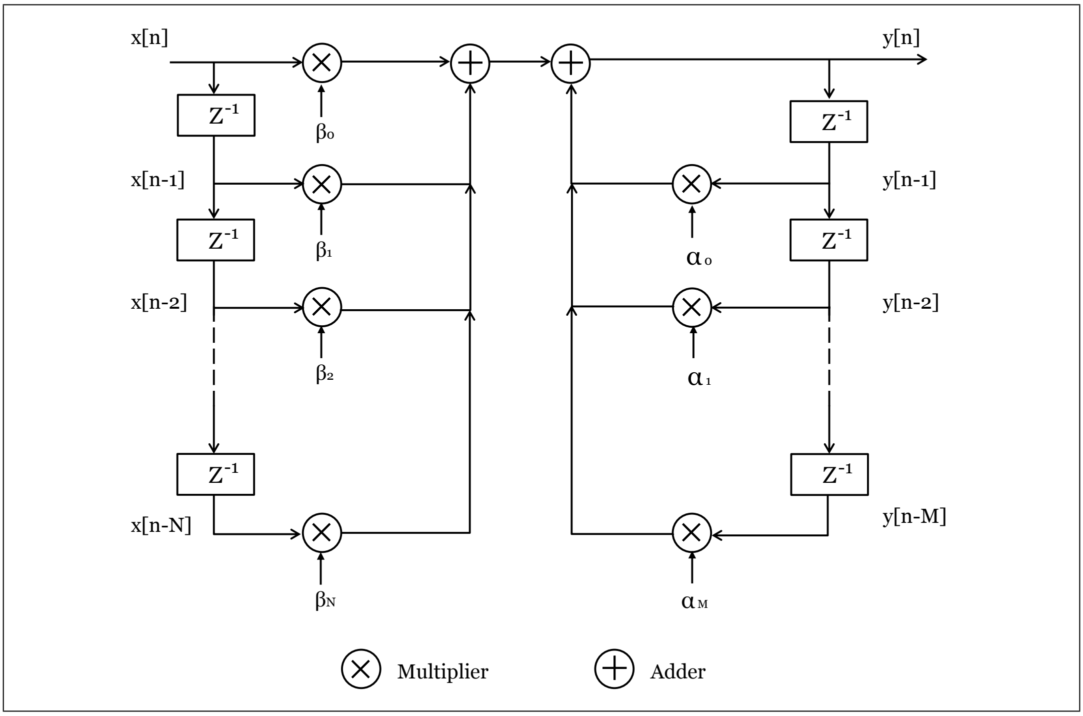

Figure 6: Generalized Structure of an M^{th} Order IIR Filter

where:

$$Num = \| (\sum_{i=0}^{N} \beta_i cos(i\omega)) - j(\sum_{i=0}^{N} \beta_i sin(i\omega)) \| \tag{33}$$

$$Den = \| (1 - \sum_{i=1}^{M} \alpha_i cos(i\omega)) + j(\sum_{i=1}^{M} \alpha_i sin(i\omega)) \| \tag{34}$$

Note that $\| \ . \ \|$ represents complex norm and $H(\omega)$ represents the complex frequency response of the filter. The function $Arg(z)$ represents the argument of a complex number [21]. Equation 32 can be derived from the transfer function $H(z)$ by mapping z on the unit circle, i.e., $z = exp(j * \omega)$. The parameter ω represents the angular frequency.

7.2.1 Formal Verification of the Frequency Response of the IIR Filter

Based on the above description of the IIR filter, our next move is to verify the frequency response (Equation (32)), which mainly involves two major steps, i.e., formal description of the model and the verification of the frequency response which is mainly based on the derivation of the transfer function. The difference equation (Equation (31)) describing the dynamics of IIR is similar to the LCCDE (i.e., Equation (13)). So we can model the IIR filter using the formalization of LCCDE as follows:

Definition 22 (IIR Model)**.**

```
⊢def ∀y M x N n. iir_model x y a_list b_list M N n =
                 LCCDE x y alist blist M N n
```

The function `iir_model` defines the dynamics of the IIR structure in a relational form. It accepts the input and output signals ($\mathtt{x}, \mathtt{y} : \mathbb{N} \rightarrow \mathbb{C}$), a list of input and output coefficients ($\mathtt{a\_lst}, \mathtt{b\_lst} : (\mathbb{C}(\mathtt{list}))$), the number of feed forward and feedback stages (`N`, `M`) and a variable `n`, which represents the discrete time.

We formally verify the frequency response of the filter given in Equation 32 as follows:

Theorem 20 (IIR Frequency Response)**.**

```
⊢ ∀x y N blst M w alst.
    cexp(j * w) IN LCCDE_ROC x y M alst ∧ is_causal_lccde x y ∧
    (∀n. iir_model x y alst blst M N n) ∧
    ¬(z_transform x (cexp (j * w)) = Cx(&0)) ⇒
    (let H = transfer_function x y (cexp (j * w)) and
    num_real = vsum (0..N) (λn. ccos (Cx(&n) * w) * EL n blst) and
    num_im = j * vsum (0..N) (λn. csin (Cx(&n) * w) * EL n blst) and
    denom_real = Cx(&1) − vsum (1..M)
                         (λn. ccos (Cx(&n) * w) * EL n alst) and
    denom_im = j * vsum (1..M) (λn. csin (Cx(&n) * w) * EL n alst) in
    H = Cx(norm (num_real − num_im) / norm (denom_real + denom_im)) *
        cexp(j * Cx(Arg((num_real − num_im) / (denom_real + denom_im)))))
```

where `cexp` and `Arg` represent complex exponential and argument of a complex number, respectively. The verification of the above theorem is mainly based on Theorem 14 and tedious complex analysis involving complex norms and transcendental functions.

This completes our formal analysis of a generalized IIR filter which demonstrates the effectiveness of the proposed theorem proving based approach to reason about practical discrete-time linear systems. The availability of the z-Transform properties greatly simplified the verification of the transfer function and frequency response. Moreover, Theorem 20 provides the generic results due to the universal quantification over the system parameters such as input and output coefficients (α_i and β_k, where

$i = 0, 1, 2, \ldots, M$ and $k = 1, 2, \ldots, N$), which is not possible in case of simulation based analysis of an IIR filter.

Thanks to the rich multivariate libraries of the HOL Light theorem prover, we have been able to formalize the z-Transform, which is an important tool to model discrete-time linear systems. The overall formalization reported in this paper consists of around 2000 lines of the HOL Light script. Indeed the underlying formalization of the z-Transform including its properties and the uniqueness took around 1700 lines of code, wheras the analysis of both applications took around 300 lines of code. The main contribution of formalizing the z-Transform in HOL can be seen as twofold: 1) To demonstrate the effectiveness of current state-of-the-art technology in theorem proving to formalize the fundamentals of engineering mathematics; 2) To build a formal framework which can be used to reason about the analytical properties of discrete-time systems in the time and frequency domain. Mostly the Laplace transform transfer functions are converted into z-domain to evaluate interesting properties and to obtain corresponding time-domain equations. The main reason behind this choice is the difficulty to obtain the inverse Laplace transform and issues about its uniqueness. In this perspective, the formalization of the z-Transform can also be used to analyze the continuous-time systems using the Biliear Transform, which is yet to be formalized in higher-order logic.

Note that the verification of the properties of the z-Transform had to be done in an interactive way due to the undecidable nature of higher-order logic. The main advantages of this long process are the accuracy of the verified results and digging out all the hidden assumptions, which are usually not mentioned in the textbooks and engineering literature. We believe that this is a one-time investment as the verification of applications becomes quite easy due to the availability of already verified properties of the z-Transform. As mentioned in [2], the availability of fundamental libraries of mathematics can attract mathematicians to use interactive theorem proving for verifying key lemmas in their work, so as in the case of engineers.

8 Conclusion and Future Directions

In this paper, we reported the formal analysis of discrete-time systems using the z-Transform which is one of the most widely used transform methods in signal processing and communication engineering. We leveraged upon the high expressiveness and the soundness of the HOL Light theorem prover to formalize the fundamental properties (e.g., time delay, time advance, complex translation and initial value theorem) of the z-Transform and linear constant coefficient difference equations. We also discussed and presented a proof of the uniqueness of the z-Transform which is required to transform z-domain expressions in the time-domain. Finally, in order to demonstrate the effectiveness of the developed formalization, we presented the formal analysis of a switched capacitor voltage doubler and a generalized infinite impulse response filter. Our reported work can be considered as a step towards an ultimate goal of using theorem provers in the design and analysis of systems from different engineering and physical science disciplines (e.g., signal processing, control systems, biology, optical and mechanical engineering).

In future, we plan to use the formalization of the z-Transform to verify the properties of photonic filters [5, 18] and discrete-time fractional order systems [27, 25]. In both these applications, our current formalization can be substantially used in its current state. However, the analysis of fractional order systems require more formalization of discrete fractional derivates based on the theory of special functions (e.g., Gamma Function [26, 13]). Another interesting direction is the development of a formal link between the z-Transform and the signal-flow-graph [24], which is a complementary technique to obtain the transfer functions of various engineering systems [4, 3]. Indeed such a formal link will provide a framework to use our formalization to reason about graphical models of signal processing and control systems often realized in MATLAB Simulink.

References

[1] A.S. Alfa. *Queueing Theory for Telecommunications - Discrete Time Modelling of a Single Node System.* Springer, 2010.

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

[3] S.M. Beillahi, U. Siddique, and S. Tahar. Formal Analysis of Power Electronic Systems. In *Formal Methods and Software Engineering*, volume 9407 of *Lecture Notes in Computer Science*, pages 270–286. Springer, 2015.

[4] S.M. Beillahi, U. Siddique, and S. Tahar. Formal Analysis of Engineering Systems Based on Signal-Flow-Graph Theory. In *Numerical Software Verification*, volume 10152 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2017.

[5] L.N. Binh. *Photonic Signal Processing: Techniques and Applications.* Optical Science and Engineering. Taylor & Francis, 2010.

[6] P.S.R. Diniz, E.A.B. Da Silva, and S.L. Netto. *Digital Signal Processing: System Analysis and Design.* Cambridge University Press, 2002.

[7] S. Elaydi. *An Introduction to Difference Equations.* Springer, 2005.

[8] S. Fadali and A. Visioli. *Digital Control Engineering: Analysis and Design.* Academic Press, 2012.

[9] G. Gonthier. Engineering Mathematics: The Odd Order Theorem Proof. In *Proceedings of the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 1–2. ACM, 2013.

[10] U. Graf. *Applied Laplace Transforms and z-Transforms for Scientists and Engineers: A Computational Approach using a Mathematica Package.* Birkhäuser Basel, 2012.

[11] T. Hales. *Dense Sphere Packings: A Blueprint for Formal Proofs*, volume 400 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 2012.

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

[13] J. Harrison. Formal Proofs of Hypergeometric Sums. *Journal of Automated Reasoning*, 55(3):223–243, 2015.

[14] A. Schultz J. Li, C.R. Sullivan. Coupled-Inductor Design Optimization for Fast-Response Low-Voltage DC-DC Converters. In *IEEE Applied Power Electronics Conference and Exposition*, volume 2, pages 817–823, 2002.

[15] E.I. Jury. *Theory and Application of the Z-Transform Method.* Wiley, 1964.

[16] B.P. Lathi. *Linear Systems and Signals.* Oxford University Press, 2005.

[17] D. Ma and R. Bondade. *Reconfigurable Switched-Capacitor Power Converters.* Springer, 2013.

[18] S. Mandal, K. Dasgupta, T.K. Basak, and S.K. Ghosh. A Generalized Approach for Modeling and Analysis of Ring-Resonator Performance as Optical Filter. *Optics Communications*, 264(1):97 – 104, 2006.

[19] Mathematica: Signal Processing Functions. `http://reference.wolfram.com/mathematica/guide/SignalProcessing.html`, 2018.

[20] MathWorks: Signal Processing Toolbox. `http://www.mathworks.com/products/signal/`, 2018.

[21] A.V. Oppenheim, R.W. Schafer, and J.R. Buck. *Discrete-Time Signal Processing.* Prentice Hall, 1999.

[22] A. Rashid and O. Hasan. On the Formalization of Fourier Transform in Higher-order Logic. In *Interactive Theorem Proving*, volume 9807 of *Lecture Notes in Computer Science*, pages 483–490. Springer, 2016.

[23] U. Siddique. Formal Anlysis of Discrete-Time Linear Systems. `http://hvg.ece.concordia.ca/projects/signal-processing/discrete-time.html`, 2018.

[24] U. Siddique, S.M. Beillahi, and S. Tahar. On the Formal Analysis of Photonic Signal Processing Systems. In *Formal Methods for Industrial Critical Systems*, volume 9128 of *Lecture Notes in Computer Science*, pages 162–177, 2015.

[25] U. Siddique and O. Hasan. Formal Analysis of Fractional Order Systems in HOL. In *Formal Methods in Computer-Aided Design*, pages 163–170. IEEE, 2011.

[26] U. Siddique and O. Hasan. On the Formalization of Gamma Function in HOL. *Journal of Automated Reasoning*, 53(4):407–429, 2014.

[27] U. Siddique, O. Hasan, and S. Tahar. Towards the Formalization of Fractional Calculus in Higher-Order Logic. In *Intelligent Computer Mathematics*, volume 9150 of *Lecture*

Notes in Computer Science, pages 316–324. Springer, 2015.

[28] U. Siddique, M.Y. Mahmoud, and S. Tahar. On the Formalization of Z-Transform in HOL. In *Interactive Theorem Proving*, volume 8558 of *Lecture Notes in Computer Science*, pages 483–498. Springer, 2014.

[29] U. Siddique and S. Tahar. On the Formal Analysis of Gaussian Optical Systems in HOL. *Formal Aspects of Computing*, 28(5):881–907, 2016.

[30] U. Siddique and S. Tahar. Formal Verification of Stability and Chaos in Periodic Optical Systems. *Journal of Computer and System Sciences*, 88:271 − 289, 2017.

[31] D. Sundararajan. *A Practical Approach to Signals and Systems.* Wiley, 2009.

[32] 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.

[33] X.S. Yang. *Mathematical Modeling with Multidisciplinary Applications.* John Wiley & Sons, 2013.

Received 21 September 2015