On the Formalization of Z-Transform in HOL

Umair Siddique, Mohamed Yousri Mahmoud, and Sofiène Tahar

Department of Electrical and Computer Engineering,
Concordia University, Montreal, Canada
{muh_sidd,mo_solim,tahar}@ece.concordia.ca

Abstract. System analysis based on difference or recurrence equations is the most fundamental technique to analyze biological, electronic, control and signal processing systems. Z-transform is one of the most popular tool to solve such difference equations. In this paper, we present the formalization of Z-transform to extend the formal linear system analysis capabilities using theorem proving. In particular, we use differential, transcendental and topological theories of multivariate calculus to formally define Z-transform in higher-order logic and reason about the correctness of its properties, such as linearity, time shifting and scaling in z-domain. To illustrate the practical effectiveness of the proposed formalization, we present the formal analysis of an infinite impulse response (IIR) digital signal processing filter.

1 Introduction

In general, dynamics of engineering and physical systems are characterized by differential equations [18] and difference equations [3] 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.), nature of input signals and physical constraints. Transformation analysis is one of the most efficient technique 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 those of algebraic equations. Z-transform [12] 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, z-domain representation of LDEs is also used for the transfer function analysis of corresponding systems. Due to these distinctive features, Z-transform is one of the main core techniques available in physical and engineering system analysis softwares (e.g., [11,10]) and is widely used in the design and analysis of signal processing filters [12], electronic circuits [3], control systems [4], photonic devices [9] and queueing networks [1].

The main idea of Z-transform can be traced back to Laplace, but it was formally introduced by W. Hurewicz (1947) to solve linear constant coefficient difference equations [7]. Mathematically, Z-transform can be defined as a function

G. Klein and R. Gamboa (Eds.): ITP 2014, LNAI 8558, pp. 483–498, 2014.

series which transforms a discrete time signal $f[n]$ into 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 ($f : \mathbb{N} \rightarrow \mathbb{R}$) 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 Z-transform is to apply Z-transform on both sides of a given equation. Next, the corresponding z-domain equation is simplified using various properties of 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 Z-transform has been done using numerical computations and symbolic techniques [11,10]. Both of these approaches, including paper-and-pencil proofs [12] 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 and the analysis of physical systems. For the latter case, the main task is to identify and formalize the underlying mathematical theories. In practice, four fundamental transformation techniques (i.e., Laplace transform (LT), Z-transform (ZT), Fourier transform (FT), and Discrete Fourier transform (DFT)) are used in the designing and analysis of linear systems. Interestingly, Fourier transform and Discrete Fourier transform can be derived from Laplace transform and Z-transform, respectively. Recently, the formalization of Laplace transform has been reported in [17] using the multivariate analysis libraries of HOL Light [6], with an ultimate goal of reasoning about differential equations and transfer functions of continuous systems. Nowadays, discrete-time linear systems are widely used in the safety and mission critical domains (e.g., digital control of avionics systems and biomedical devices). We believe that there is a dire need of an infrastructure which provides the basis for the formal analysis of discrete-time systems within the sound core of a theorem prover. To the best of our knowledge, so far Z-transform has not been formalized which is an important step towards formal analysis of discrete-time physical and engineering systems.

Our main objective is two-fold: firstly, we aim at extending theorem proving support for linear system analysis. Secondly, we plan to enrich the current foundations of optics formalization [13,15] to reason about futuristic photonic signal processing systems [2,9]. In this paper, we propose Z-transform based system analysis using a higher-order-logic theorem prover. The main idea is to leverage upon the high expressiveness of higher order logic to formalize Equation (1) and use it to verify the classical properties of Z-transform within a theorem prover. These foundations can be built upon to reason about the analytical solutions of difference equations or transfer functions. As a first step towards our ultimate

goal, we present in this paper the higher-order logic formalization of Z-transform and its associated region of convergence (ROC). Next, we present the formal verification of its most commonly used properties such as linearity, time delay, time advance and scaling in z-domain. Consequently, we present the formalization of linear constant coefficient difference equation along with the formal verification of its Z-transform by utilizing the above mentioned properties. In order to demonstrate the practical effectiveness of the reported work, we present the formal analysis of an infinite impulse response (IIR) digital signal processing filter.

Formalization reported in this paper has been developed in the HOL Light theorem prover due to its rich multivariate analysis libraries [6]. Another motivation of choosing HOL Light is the existing formalization of Laplace transform and photonic systems which are complementary to achieve our final objective of analyzing linear systems and integrated optics. The source code of our formalization is available for download [14] 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 Z-transform and the verification of its properties, respectively. In Section 5, we present the analysis of an IIR filter as illustrative practical application. Finally, Section 6 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 [5,6]. Our main intent is to introduce the basic definitions and notations that are going to be used in the rest of the paper.

In the vectors theory formalization, 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 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 ($\mathbb{R}^M \to \mathbb{R}^N$). Next, we present the definitions frequently used in our formalization.

First, generalized summation over arbitrary functions is defined as follows:

Definition 1 (Vector Summation)
$\vdash$ $\forall$ `s f. vsum s f = (lambda i. sum s (λ x. f x$i))`

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

Next, we present the formal definition of the traditional mathematical expression $\sum_{i=k}^{\infty} f(i) = L$, as follows:

Definition 2 (Sums)

```
⊢ ∀ s f L. (f sums L) s ⇔
           ((λ n. vsum (s ∩ (0..n)) f) → L) sequentially
```

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

Now, we define 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 $\sum_{i=k}^{\infty} f(i) = L$.

Definition 3 (Summability)

```
⊢ ∀ f s. summable s f ⇔ (∃ L. (f sums L) s)
```

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

Definition 4 (Limit)

```
⊢ ∀ f net. lim net f = (@L. (f → L) net)
```

The function `lim` is defined using the Hilbert choice operator @ in the functional form. It accepts a `net` with elements of arbitrary data-type `A` and a function $(\texttt{f} : \texttt{A} \to \mathbb{R}^{\texttt{N}})$, and returns $(\texttt{L} : \mathbb{R}^{\texttt{N}})$: , i.e., the value to which `f` converges at the given `net`. In this paper, we are considering only sequential nets, which describes the sequential evolution of a function, i.e. $f(i), f(i+1), f(i+2)$, ..., etc.

Next, we present the definition of an infinite summation which is one the most fundamental requirement in our development.

Definition 5 (Infinite Summation)

```
⊢ ∀ f s. infsum s f = (@L. (f sums L) s)
```

The function `infsum` is also defined using the Hilbert choice operator @ in the functional form. It accepts a parameter $(\texttt{s} : \texttt{num} \to \texttt{bool})$ which specifies the starting point and a function $(\texttt{f} : \mathbb{N} \to \mathbb{R}^{\texttt{N}})$, and returns $(\texttt{L} : \mathbb{R}^{\texttt{N}})$: , 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)

```
⊢ ∀ s f. infsum s f = lim sequentially (λ k. vsum (s ∩ (0..k)) f))
```

Next, we present the definition of complex differentiation as follows:

Definition 6 (Complex Differentiation)

```
⊢ ∀ f f' net. (f has_complex_derivative f') net ⇔
              (f has_derivative (λx. f' * x)) net
```

The function `has_complex_derivative` defines the complex derivative in a relational form. Here, $(\texttt{f} : \mathbb{C} \to \mathbb{C})$ and $\texttt{f'}:(\mathbb{C})$ represent a given function and the corresponding complex derivative at a given $(\texttt{net} : (\mathbb{C})\texttt{net})$, respectively. The function `has_derivative` is a generalized vector derivative. The above definition can also be described in a functional form as follows:

Definition 7 (Complex Differentiation)

```
⊢ ∀ f x. complex_derivative f x =
         (@f'. (f has_complex_derivative f')) (at x)
```

Note that, 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 `Cx` : $\mathbb{R} \to \mathbb{C}$. The real and imaginary parts of a complex number are represented by `Re` and `Im` both with type $\mathbb{C} \to \mathbb{R}$.

We build upon the above mentioned fundamentals to formalize Z-transform in the next section.

3 Z-Transform Formalization

The unilateral Z-transform [8] 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 $n = \infty$. On the other hand, one can consider these limits from $n = -\infty$ to $n = \infty$ and such a version of Z-transform is called two-sided or bilateral transform. 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 [16]. For similar reasons, in [17], the authors formalized the unilateral Laplace transform rather than the bilateral version.

An essential issue of 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 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 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 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., [12,16]) is incorrect for certain functions, for example, $f[n] = \frac{1}{n}u[n-1]$ for which certain values of z result in convergent infinite series, but $x[n]r^{-n}$ is not absolutely summable.

Now, we have two distinct choices for defining ROC: first, z values for which $F(z)$ is finite (or summable) and second, z values for which $x[n]z^{-n}$ is absolutely summable. Most of the textbooks are not rigorous about the choice of ROC and both of these definitions are widely used in the analysis of engineering

systems. In this paper, we use the first definition of ROC, which we can define mathematically as follows:

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

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

Definition 8 (Z-Transform)
```
⊢ ∀ f z. z_transform f z = infsum (from 0) (λ n. f n * z^-n)
```

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

Next, we present the formal definition of the ROC as follows:

Definition 9 (Region of Convergence)
```
⊢ ∀ f. ROC f = {z | summable (from 0) (λ n. f n * z^-n)}
```

Here, `ROC` accepts a function $\mathtt{f} : \mathbb{N} \to \mathbb{C}$ and returns a set of values of variable z for which the Z-transform of $f(n)$ is summable. In order to compute the Z-transform, it is mandatory to specify the associated ROC. Now, we present two basic properties of ROC as follows:

Theorem 2 (ROC Linear Combination)
```
⊢ ∀ z α β f g. z ∈ ROC f ∧ z ∈ ROC g ⟹
                z ∈ ROC (λn. α * f n) ∩ ROC (λn. β * g n)
```

Theorem 3 (ROC Scaling)
```
⊢ ∀ z α f. z ∈ ROC f ⟹ z ∈ ROC (λn. f n / α)
```

where 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 α and β, respectively. Similarly, Theorem 3 shows the scaling with respect to complex division by a complex number α.

4 Z-Transform Properties

In this section, we use Definitions 8 and 9 to formally verify some of the classical properties of Z-transform in HOL Light. The verification of these properties not only ensures the correctness of our definitions but also plays an important role in reducing the time required to analyze practical applications, as described later in Section 5.

4.1 Linearity of 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 $\mathcal{Z}(f[n])\ z = F(z)$ with $ROC = R_f$ and $\mathcal{Z}(g[n])\ z = G(z)$ with $ROC = R_g$, then the following holds:

$$\mathcal{Z}(\alpha * f[n] \pm \beta * g(n))\ z = \alpha * F(z) \pm \beta * G(z) \qquad ROC \supseteq R_f \cap R_g \tag{4}$$

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

Theorem 4 (Linearity of Z-Transform)

```
⊢ ∀ z f g α β. z ∈ ROC f ∩ ROC g ⟹
   z_transform (λ n. α * f n + α * g n) z =
                α * z_transform f z + β * z_transform g z
```

where $\alpha : \mathbb{C}$ and $\beta : \mathbb{C}$ are arbitrary constants.

The proof of these theorems are based on the linearity of infinite summation and Theorem 2.

4.2 Shifting Properties

The shifting properties of Z-transform are the most widely 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 $\mathcal{Z}(f[n])\ z = F(z)$ and m is a positive integer, then the left shift of a sequence can be described as follows:

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

We verify this theorem as follows:

Theorem 5 (Left Shift or Time Advance)

```
⊢ ∀ f z m. z ∈ ROC f ∧ (0 < m) ⟹
    z_transform (λ n. f (n + m)) z =
    z^m * (z_transform f z) - vsum (0..m - 1) (λ n. f n * z^-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 ∧ (m < n) ⟹
    infsum (from m) f = vsum (m..n - 1) f + infsum (from n) f
```

Right Shift of a Sequence: If $\mathcal{Z}(f[n])\ z = F(z)$, 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:

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

We formally verify the above property as the following theorem:

Theorem 6 (Right Shift or Time Delay)
```
⊢ ∀ f z m. z ∈ ROC f ∧ (∀ m. is_causal f m) ⟹
           z_transform (λ n. f (n - m)) z = z^m * (z_transform f z)
```

Here, `is_causal` defines the causality of the function `f` in a relational form to ensure that $\mathtt{f}(\mathtt{n} - \mathtt{m}) = 0,\ \forall \mathtt{m}.\mathtt{n} < \mathtt{m}$. 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) ⟹
           ((λ n. f (n - k)) sums l) (from k)
```

Lemma 3 (Infinite Summation Negative Offset)
```
⊢ ∀ f k. summable (from 0) f ⟹
         infsum (from 0) (λ n.if k ≤ n then f (n - k) else Cx(&0))
         = infsum (from 0) f
```

As a direct application of above results, we verify another important property called first-difference, as follows:

Theorem 7 (First Difference)
```
⊢ ∀ f. z ∈ ROC f ∧ (∀ m. is_causal f m) ⟹
  z_transform (λ n. f (n) - f(n-1)) z = (1 - z^-1) * (z_transform f z)
```

4.3 Scaling in *Z*-Domain

The scaling property of Z-transform plays an important role in the designing of 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:

$$\mathcal{Z}(Z_0^n f[n])\ z = F(\frac{z}{Z_0}) \tag{7}$$

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

If Z_0 is a positive real number, then it can be interpreted as shrinking or expanding of the z-domain. If Z_0 is a complex with unity magnitude, i.e., $z = e^{j\omega_0}$, then the scaling corresponds to a rotation in the z-plane by an angle of ω_0. Indeed, in 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 Z₀ z. z_transform (λ n. Z₀ pow n * f n) z = z_transform f (z / Z₀)
```

Theorem 9 (Scaling in z-Domain (Negative))
```
⊢ ∀ f ω z. z_transform (λ n. ω pow -- n * f n) z = z_transform f (ω * z)
```

The verification of above theorems mainly involves the properties of complex power.

4.4 Complex Differentiation

The differentiation property of 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. ≠ Cx(&0) ∧ &0 < Re z ∧ z ∈ (λ n. Cx (&n) * f n)
    ⟹ z_transform (λ n. Cx (&n) * f n) z =
  -z * infsum (from 0) (λ n. complex_derivative (λ z. f n * z pow -- n) z)
```

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

4.5 Difference Equation

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:

$$y[n] - y[n-1] = \sum_{i=0}^{M} \alpha_i f[n-i] \tag{10}$$

Here, M is called the order of difference equation and α_i represents the list of input coefficients. For a given M^{th} order difference equation in terms of a function $f[n]$, its Z-transform is given as follows:

$$\mathcal{Z}(\sum_{i=0}^{M} \alpha_i f[n-i])\ z = F(z) \sum_{i=0}^{M} \alpha_i z^{-n} \tag{11}$$

We formalize the difference equation as follows:

Definition 10 (Difference Equation)

```
⊢ ∀ N α_lst f x. difference_eq M α_lst f x =
    vsum (0..M) (λ t. EL t α_lst * f (x - t)* z^-n
```

The function `difference_eq` accepts the order (`M`) of the difference equation, a list of coefficients `α_lst`, a causal 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 verify the Z-transform of the difference equation which is one of the most powerful results of our formalization as will be demonstrated in Section 5.

Theorem 11 (Z-Transform of Difference Equation)

```
⊢ ∀ M α_lst f x. z ∈ ROC f ∧ z ≠ Cx(&0) ∧
    (∀ m. is_causal f m) ⟹
    z_transform (λx. difference_eq M α_lst f x) z =
    (z_transform f z) * (vsum (0..M) (λ n. EL n α_lst * z^-n))
```

We prove the above theorem by induction and using Theorems 2 and 4 along with the following important lemma about the summability of difference equation:

Lemma 4 (Summability of Difference Equation)

```
⊢ ∀ M α_lst f x. z ∈ ROC f ∧ (∀ m. is_causal f m)
    ⟹ z ∈ ROC (λx. difference_eq M α_lst f x)
```

This completes our formalization of the Z-transform and verification of its main properties, which to the best of our knowledge is the first one in higher-order logic. We believe that our formalization can be directly utilized in many applications such as economics, biology, signal processing and control engineering.

5 Application: Formal Analysis of Infinite Impulse Response Filter

In order to illustrate the utilization and effectiveness of the reported formalization, we apply it to analyze a real-world engineering system, i.e., an infinite impulse response filter [12].

Digital filters are fundamental components of almost all signal processing and communication systems. The main functionality of such components are: 1) to 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. The design and analysis of digital filters mainly involves three steps, i.e., the specification of the desired properties of the system, modeling using a causal discrete-time system and realization of overall structure (parallel, cascaded, etc.). Given the filter specifications in terms of frequency response, the first step is to model the filter

using constant coefficient difference equations. The next step is to express it in the form of transfer function using the Z-transform properties. Consequently, frequency response analysis and architectural optimization can be performed based on the given specifications.

An impulse response of a system describes its behaviour under an external change (mathematically, this describes the system response when the dirac-delta function is applied as an input [12]). 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 1.

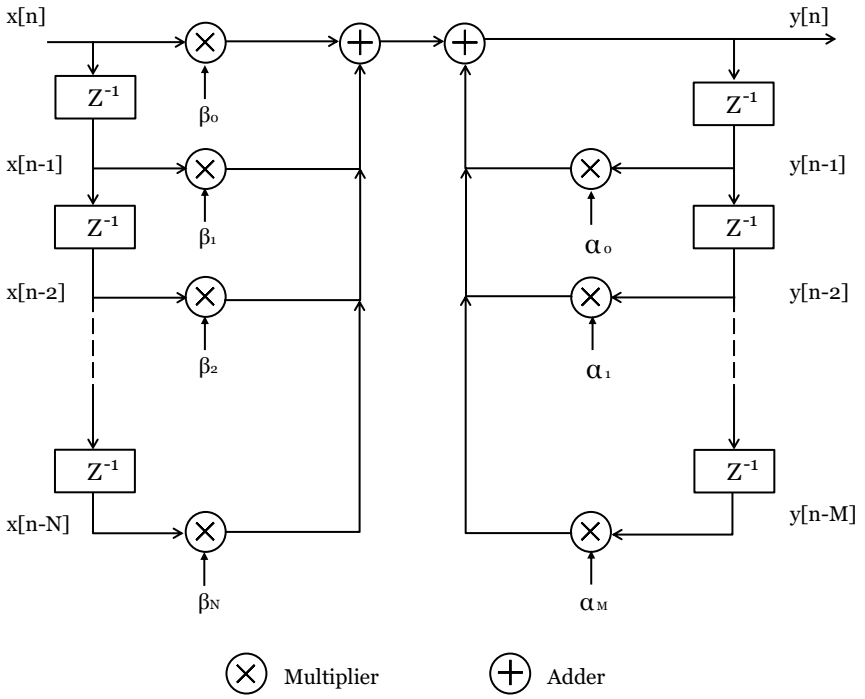

Fig. 1. Generalized Structure of an M^{th} Order IIR Filter

Mathematically, it can be described as:

$$y[n] = \sum_{i=1}^{M} \alpha_i y[n-i] + \sum_{i=0}^{N} \beta_i x[n-i] \tag{12}$$

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. 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 transfer function and frequency response of an IIR filter which are given as:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} \beta_i z^{-i}}{1 - \sum_{i=1}^{M} \alpha_i z^{-i}} \tag{13}$$

$$H(\omega) = \frac{\sqrt{(\sum_{i=0}^{N} \beta_i cos(i\omega))^2 + (\sum_{i=0}^{N} \beta_i sin(i\omega))^2}}{\sqrt{(1 - \sum_{i=1}^{M} \alpha_i cos(i\omega))^2 + (\sum_{i=1}^{M} \alpha_i sin(i\omega))^2}} * \exp(j * Arg(H(\omega))) \tag{14}$$

where $H(z)$ and $H(\omega)$ represent the filter's transfer function and complex frequency response, respectively. The function $Arg(z)$ represents the argument of a complex number [12]. Equation 14 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.

Based on the above description of the IIR filter, our next move is to conduct its formal analysis, which mainly involves two major steps, i.e., formal description of the model and underlying constraints followed by the formal verification of transfer function and frequency response. As a first step, we build the formal model of the IIR filter using Equation 12.

Definition 11 (IIR Model)

```
⊢ ∀ x y α_lst β_lst M N n. IIR_MODEL x y α_lst β_lst M N n ⇔
    y n = differen_eq α_lst y M n +
    difference_eq β_lst x N n ∧ HD α_lst = Cx(&0)
```

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

In order to model $\sum_{i=1}^{M} \alpha_i y[n-i]$ using our definition of difference equation, we added the constraint that the first element (i.e., HD α_lst) of the output coefficients should be 0.

According to the filter specification, we need to ensure that the input and output signals should be causal as described in Section 3. Another important requirement is to ensure that there are no values of z for which denominator is 0, such values are called poles of that transfer function. For the correct operation of the filter, the region of convergence (ROC) should not include any poles. We package these conditions in the following definitions:

Definition 12 (Causality Condition)

```
⊢ ∀ x y. is_causal_iir x y ⇔
    (∀ k. is_causal x k) ∧ (∀ k. is_causal y k)
```

Definition 13 (IIR FILTER ROC)

```
⊢ ∀ x y α_lst M.
    IIR_ROC x y α_lst M =
    z IN (ROC x ∩ ROC y) DIFF
    {z | (Cx(&1) - vsum (1..M) (λ n.EL n α_lst * z^-n) = Cx(&0)}
```

Here, the function is_causal_irr takes two parameters, i.e., input and output, and ensures that both of them are causal. In Definition 13, IIR_ROC specifies the region of convergence of IIR, 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., $A \setminus B = \{z : z \in A \wedge x \notin B\}$. Next, we present the formal verification of the transfer function as given in Equation 13.

Theorem 12 (IIR Transfer Function Verification)

```
⊢ ∀ x y α_lst β_lst M N.
    z ∈ IIR_ROC x y α_lst M ∧
    z ≠ Cx(&0) ∧ is_causal_iir x y ∧
     (∀ n. IIR_MODEL x y α_lst β_lst M N n) ⟹
    z_transform y z / z_transform x z = vsum (0..N) (λn.EL n β_lst * z^-n) / 1 − (vsum (1..M) (λn.EL n α_lst * z^-n))
```

The first and second assumptions describe the region of convergence for the IIR filter. The second assumption ensures the causality of the filter's input and output, and the last assumption gives the time-domain model of the given IIR filter. 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 as it greatly simplifies the reasoning for any given design of IIR. Moreover, this theorem can be used to reason about many important aspects such as stability and architectural optimization. For example, the stability of a given IIR design can be checked by ensuring that all poles of the transfer function lies inside the unit circle (i.e., their magnitude is less than 1).

Next, we verify the frequency response of the filter given in Equation 14 as follows:

Theorem 13 (IIR Frequency Response)

$$\vdash \forall\ \texttt{x y}\ \alpha\texttt{\_lst}\ \beta\texttt{\_lst M N.}$$
$$\texttt{z} \in \texttt{IIR\_ROC x y}\ \alpha\texttt{\_lst M} \wedge$$
$$\texttt{z = cexp(ii*}\omega\texttt{)} \wedge \texttt{is\_causal\_iir x y} \wedge$$
$$(\forall\ \texttt{n. IIR\_MODEL x y}\ \alpha\texttt{\_lst}\ \beta\texttt{\_lst M N n}) \Longrightarrow$$
$$\texttt{let H =}\ \frac{\texttt{z\_transform y z}}{\texttt{z\_transform x z}}\ \texttt{and}$$
$$\texttt{num\_real = vsum (0..N)}\ (\lambda\texttt{n.EL n}\ \beta\texttt{\_lst}\ *\ \texttt{ccos(n}*\omega\texttt{))}\ \texttt{and}$$
$$\texttt{num\_imag = -vsum (0..N)}\ (\lambda\texttt{n.EL n}\ \beta\texttt{\_lst}\ *\ \texttt{csin(n}*\omega\texttt{))}\ \texttt{and}$$
$$\texttt{den\_real = 1 - (vsum (1..M)}\ (\lambda\texttt{n.EL n}\ \alpha\texttt{\_lst}\ *\ \texttt{ccos(n}*\omega\texttt{)))}\ \texttt{and}$$
$$\texttt{den\_imag = vsum (1..M)}\ (\lambda\texttt{n.EL n}\ \alpha\texttt{\_lst}\ *\ \texttt{csin(n}*\omega\texttt{))}\ \texttt{in}$$
$$\texttt{H = Cx}(\frac{\texttt{sqrt}[(\texttt{num\_real})^2 + (\texttt{num\_imag})^2]}{\texttt{sqrt}[(\texttt{den\_real})^2 + (\texttt{den\_imag})^2]}) * \texttt{cexp(Arg(H))}$$

Where `sqrt`, `cexp` and `Arg` represent the real square root (over reals), 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.

Theorems 12 and 13 provide 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 $k = 1, 2, \ldots, N$). Next, we utilise these results to formally verify the transfer function and frequency response of a second order low-pass IIR filter as shown in Figure 2. The input and output coefficients are $[0.0605, 0.121, 0.0605]$ and $[1.94, -0.436]$, respectively. We model this structure as follows:

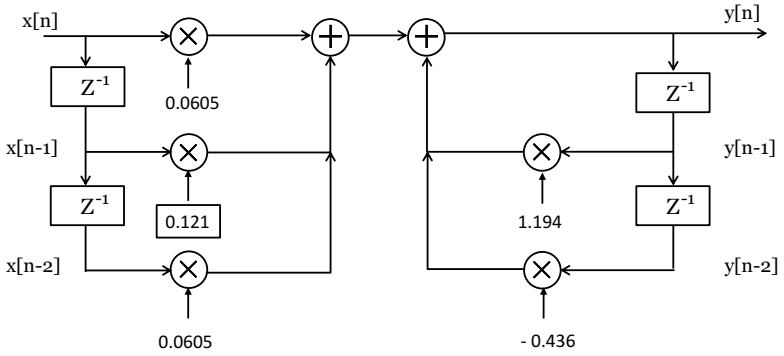

Fig. 2. Second Order Low-Pass IIR Filter

Definition 14 (Second Order IIR Model)

$$\vdash\ \alpha\texttt{\_lst} = [\texttt{Cx(\&0)}; \texttt{Cx}(\tfrac{\texttt{\&1194}}{\texttt{\&1000}}); -\texttt{Cx}(\tfrac{\texttt{\&436}}{\texttt{\&1000}})]$$

$$\vdash\ \beta\texttt{\_lst} = [\texttt{Cx}(\tfrac{\texttt{\&605}}{\texttt{\&10000}}); \texttt{Cx}(\tfrac{\texttt{\&121}}{\texttt{\&1000}}); \texttt{Cx}(\tfrac{\texttt{\&605}}{\texttt{\&10000}})]$$

$$\vdash\ \forall\ \texttt{x y. SECOND\_ORDER\_IIR\_MODEL x y}\ \alpha\texttt{\_lst}\ \beta\texttt{\_lst} \Leftrightarrow$$
$$\forall\ \texttt{n. y n = differen\_eq}\ \alpha\texttt{\_lst y 2 n + difference\_eq}\ \beta\texttt{\_lst x 2 n}$$

Here, SECOND_ORDER_IIR_MODEL accepts the input and output signals, a list of input and output coefficients (defined by α_lst, β_lst), and returns the difference equation describing the behaviour of the low-pass IIR filter.

Theorem 14 (Second Order Low-pass IIR Filter Transfer Function)
$\vdash \forall$ x y z. z $\in$ IIR_ROC x y α_lst 2 $\wedge$

z $\neq$ Cx(&0) $\wedge$ is_causal_iir x y $\wedge$

(SECOND_ORDER_IIR_MODEL x y α_lst β_lst) $\Longrightarrow$

$$\frac{\texttt{z\_transform y z}}{\texttt{z\_transform x z}} = \frac{\texttt{Cx}(\frac{\&605}{10000}) + \texttt{Cx}(\frac{\&121}{\&1000}) \ * \ \texttt{z}^{-1} + \texttt{Cx}(\frac{\&605}{\&10000}) \ * \ \texttt{z}^{-2}}{\texttt{Cx}(\&1) - \texttt{Cx}(\frac{\&1194}{\&1000}) \ * \ \texttt{z}^{-1} + \texttt{Cx}(\frac{\&436}{\&1000}) \ * \ \texttt{z}^{-2}}$$

The verification of the above theorem is based on Theorem 12.

This completes our formal analysis of a generalized IIR filter which demonstrates the effectiveness of the proposed theorem proving based approach to reason about discrete-time linear systems. The availability of the Z-transform properties greatly simplified the verification of the transfer function and frequency response. The verification of the transfer function and frequency response task took around 150 lines of the HOL Light code and a couple of man-hours each. We believe that reported formalization demonstrates the maturity of interactive theorem provers.

6 Conclusion and Future Directions

In this paper, we reported the formalization of Z-transform which is one of the most widely used transform methods in signal processing and communication theory. We presented the formal definitions of unilateral Z-transform and its associated region of convergence along with the formal verification of some important properties such as linearity, time shifting and difference equations. Finally, in order to demonstrate the effectiveness of the developed formalization, we presented the formal analysis of a generalized infinite impulse repones filter. Consequently, we verified the transfer function and frequency response of a second order low-pass IIR filter.

The utilization of higher-order logic theorem proving in industrial settings (particularly, physical systems) is always questionable due to the huge amount of time required to formalize the underlying theories. Another, important factor is the gap between the theorem proving and engineering communities which limits its usage in industry. For example, it is hard to find engineers with theorem proving background and vice-versa. Our reported work can be considered as a one 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).

Our immediate future work is the formalization of the uniqueness theorem of Z-transform [12], which is required to reliably deduce some important properties of difference equations and discrete-time linear systems. The proof of this theorem entails some additional properties of complex differentiation and infinite

summations. Another future direction is the formalization of most commonly used inverse transform techniques like power series method, partial fractions and the Cauchy's integral method.

References

1. Alfa, A.S.: Queueing Theory for Telecommunications - Discrete Time Modelling of a Single Node System. Springer (2010)
2. Binh, L.N.: Photonic Signal Processing: Techniques and Applications. Optical Science and Engineering. Taylor & Francis (2010)
3. Elaydi, S.: An Introduction to Difference Equations. Springer (2005)
4. Fadali, S., Visioli, A.: Digital Control Engineering: Analysis and Design. Academic Press (2012)
5. Harrison, J.: Formalizing Basic Complex Analysis. In: From Insight to Proof: Festschrift in Honour of Andrzej Trybulec. Studies in Logic, Grammar and Rhetoric, vol. 10(23), pp. 151–165 (2007)
6. Harrison, J.: The HOL Light Theory of Euclidean Space. Journal of Automated Reasoning 50(2) (2013)
7. Jury, E.I.: Theory and Application of the Z-Transform Method. Wiley (1964)
8. Lathi, B.P.: Linear Systems and Signals. Oxford University Press (2005)
9. Mandal, S., Dasgupta, K., Basak, T.K., Ghosh, S.K.: A Generalized Approach for Modeling and Analysis of Ring-Resonator Performance as Optical Filter. Optics Communications 264(1), 97–104 (2006)
10. Mathematica Guide: Signal Processing Related Functions (2014), `http://reference.wolfram.com/mathematica/guide/SignalProcessing.html`
11. MathWorks: Signal Processing Toolbox (2014), `http://www.mathworks.com/products/signal/`
12. Oppenheim, A.V., Schafer, R.W., Buck, J.R.: Discrete-Time Signal Processing. Prentice Hall (1999)
13. Siddique, U., Aravantinos, V., Tahar, S.: Formal Stability Analysis of Optical Resonators. In: Brat, G., Rungta, N., Venet, A. (eds.) NFM 2013. LNCS, vol. 7871, pp. 368–382. Springer, Heidelberg (2013)
14. Siddique, U., Mahmoud, M.Y.: On the Formalization of Z-Transform - HOL Light Script (2014), `http://hvg.ece.concordia.ca/projects/signal-processing/z-transform.html`
15. Siddique, U., Tahar, S.: Towards the Formal Analysis of Microresonators Based Photonic Systems. In: IEEE/ACM Design Automation and Test in Europe, pp. 1–6 (2014)
16. Sundararajan, D.: A Practical Approach to Signals and Systems. Wiley (2009)
17. Taqdees, S.H., Hasan, O.: Formalization of Laplace Transform Using the Multivariable Calculus Theory of HOL-Light. In: McMillan, K., Middeldorp, A., Voronkov, A. (eds.) LPAR-19. LNCS, vol. 8312, pp. 744–758. Springer, Heidelberg (2013)
18. Yang, X.S.: Mathematical Modeling with Multidisciplinary Applications. John Wiley & Sons (2013)