

# 小 모듈러스들로 構成된 RNS 를 사용한 디지털 필터의 實現

## (An Implementation of Digital Filters Using the Residue Number System of Small Modulus

李 廷 紋\*, 裴 正 二\*\*, 崔 桂 根\*\*

(Jeong Moon Lee, Jeong Ei Bae and Kye Keun Choi)

### 要 約

本 論文에서는 레지듀 演算(residue arithmetic)을 사용하여 디지털 필터를 實現하는 한 가지 方法을 제시하였다.

既存의 레지듀 디지털 필터가 비교적 적은 비트 수를 갖는 信號만을 처리할 수 있었던 것에 비해, 여기서는 비트 슬라이스 알고리즘<sup>1)</sup>의 原理를 적용함으로써 보다 많은 비트 수를 갖는 信號를 처리할 수 있도록 했으며, 또한 레지듀 加減算 및 乘算은 演算表에 의해서 高速으로 수행될 수 있으므로 이를 利用하여 비트 슬라이스 알고리즘의 短點인 융통성(flexibility)의 문제를 해결하였다.

실제로 중속 집속형 4차 버터워즈 低域通過 디지털 필터를 構成하여 實驗함으로써 이와 같은 사실들을 입증할 수 있었다.

### Abstract

In this paper, an implementation method for digital filters using the residue arithmetic is proposed.

This method can be used for processing digital signals with larger number of bits by applying the idea of the bit-slice algorithm, while previous residue digital filters can process digital signals with only a small number of bits. Furthermore, high-speed residue addition, subtraction, and multiplication using look-up tables make it possible to get more flexible filters.

Everything that is mentioned above is proved by implementing a cascade fourth-order Butterworth lowpass digital filter using this method.

### I. 序 論

高速 汎用 컴퓨터의 設計를 위하여 RNS(residue number system)가 도입<sup>2)</sup>된 以來, 高速의 레지듀 演算을 디지털 필터의 實現에 응용하려는 研究가 계속되

어 왔다.<sup>3), 4), 5)</sup> 그런데 이제까지의 레지듀 디지털 필터에 있어서는 小 모듈러스(modulus)들로 構成된 RNS를 사용하면 dynamic range가 작아서 비트 수가 많은 디지털 信號를 처리하기가 곤란하였고, dynamic range를 충분히 크게 하기 위해서 大 모듈러스들로 構成된 RNS를 사용하면 演算表를 만드는 데 쓰이는 ROM의 용량이 상당히 커야만 했다.

本 論文에서는 비트 슬라이스 알고리즘의 原理를 적용하여 하나의 디지털 信號를 여러 개의 비트 群

\*準會員, \*\*正會員, 서울大學校 電子工學科  
(Dept. of Electronic Eng., Seoul National Univ.)  
接受日字: 1983年 3月 14日

(group)으로 나누어 順次的으로 처리되 各各의 비트 群을 처리하는 과정에서만 레지듀 演算을 함으로써 小 모듈러스들로 構成된 RNS를 사용하는 경우에도 비트 수가 많은 디지털 信號를 처리할 수 있는 實現 方式을 제시하였다.

한편 비트 슬라이스 알고리즘<sup>11)</sup>에서는 디지털 信號의 비트를 順次的으로 처리하는 과정에서 나올 수 있는 모든 計算 結果를 미리 ROM에 저장해 두어야 하므로 필터의 係數를 쉽게 바꿀 수 없었으나, 이 方式에서는 이를 레지듀 演算에 의해서 직접 計算하므로 필터의 係數를 바꾸어 特性을 쉽게 변경할 수 있도록 했다.

먼저 제 II 장에서는 RNS에 관한 기본적인 사항들을 간단하게 소개하였고, 제 III 장에서는 제시하고자 하는 디지털 필터의 實現 方式을 既存의 레지듀 디지털 필터와 비교하여 서술하였다. 그리고 제 IV 장에서는 중속 접속형 4次 버터워즈 低域通過 디지털 필터에 대한 시뮬레이션을 통해 이 方式을 사용한 경우와 既存의 레지듀 實現 方式을 사용한 경우를 비교한 후, 前者의 경우를 實現하여 이 方式이 실제로 적용 가능함을 보였다.

II. 레지듀 數

RNS는 N개의 整數,  $m_1, m_2, \dots, m_N$ 을 밑(base)으로 하여 構成되는데, 이 各各의 整數를 모듈러스라고 하며 이들은 서로 素(relatively prime)가 되도록 정하는 것이 보통이다. 주어진 밑에 대하여 整數 x의 레지듀 表現도 다른 N개의 整數,  $x_1, x_2, \dots, x_N$ 이 되는데,  $X_i$ 는 다음과 같은 N개의 式에 의해서 정의된다.

$$x = q_i m_i + x_i \quad i = 1, 2, \dots, N$$

이때 整數  $q_i$ 는  $0 \leq X_i < m_i$ 가 되도록 정한다. 즉  $X_i$ 는 x를  $m_i$ 로 나누었을 경우에 생기는 나머지로서  $|X| m_i$ 로도 表示되며 'x mod  $m_i$ '라고 읽는다. 또  $X_i$ 를 x의 i번째 레지듀 자리수라고 한다.

$m_i$ 가 서로 素일 때 有效 區間(legitimate range)  $[0, M-1]$ 内的 모든 整數는 各各 唯一한 表現을 갖는 다<sup>12)</sup>는 사실이 알려져 있다. 단,

$$MM = \prod_{i=1}^N m_i \quad (2)$$

RNS에서는 有效 區間을 보통 正領域(positive region)과 負領域(negative region)으로 나누어 陰의 整數를 M의 補數(M's complement) 형태로 表示한다.

記號 \*가 加算, 減算, 혹은 乘算을 나타낸다고 할 때,  $z = x * y$ 라면 z의 레지듀 表現은

$$z_i = |X_i * Y_i| m_i \quad i = 1, 2, \dots, N \quad (3)$$

으로 된다.<sup>12)</sup> 이와 같이 各 자리수마다 獨立的으로 演

算이 이루어지기 때문에 高速 演算이 가능하다.

레지듀 數에서 보통 數로의 변환은 Chinese Remainder Theorem<sup>12)13)</sup>을 통해서 이루어질 수 있다. 그러나 이 方法에서는 mod  $m_i$ 의 演算 뿐 아니라 mod M의 演算도 要求되며 더우기 補數 表現을 사용할 경우 부호 판정을 위한 별도의 알고리즘이 필요하므로 매우 불편하다. 반면에 혼합 래디스 변환(mixed radix conversion)을 이용하면 레지듀 數를 쉽게 보통 數로 바꿀 수 있다. 整數 x는 다음 式과 같이 혼합 래디스 數(mixed radix number)로 表示된다.

$$x = a_N \prod_{i=1}^{N-1} R_i + \dots + a_3 R_1 R_2 + a_2 R_1 + a_1 \quad (4)$$

여기에서  $R_i$ 는 래디스이고  $a_i$ 는 혼합 래디스 자리수이며  $0 \leq a_i < R_i$ 이다. 만약 어떤 RNS의 모듈러스  $m_1, m_2, \dots, m_N$ 과 어떤 MRNS(mixed radix number system)의 래디스  $R_1, R_2, \dots, R_N$  사이에  $R_1 = m_1, R_2 = m_2, \dots, R_N = m_N$ 의 관계가 성립한다면 이들은 聯關(associate)되어 있다고 한다. 聯關 MRNS에서는 다음 式에 따라 대응되는 레지듀 數로부터 혼합 래디스 數가 용이하게 얻어진다.<sup>14)15)</sup>

$$a_i = T_{ii} \quad i = 1, 2, \dots, N \quad (5)$$

$$\text{단, } T_{ii} = |x| m_i = X_i \quad (6)$$

$T_{i(k+1)i} = |T_{ki} - a_k| m_i \cdot |m_k^{-1}| m_i |m_i| \quad k = 1, 2, \dots, i-1$ 이고  $|m_k^{-1}| m_i$ 는  $|m_k| m_i$ 의 乘算 逆元(multiplicative inverse)이다.  $R_N$ 을 2의 冪(power)으로 정하면  $a_N$ 의 부호 비트가 x의 부호 비트의 역할을 하므로 부호 판정이 간편하다.<sup>12)14)</sup>

III. 필터의 設計

디지털 필터는 일반적으로 다음과 같은 入出力 관계 式으로 表示된다.<sup>16)</sup>

$$y(n) = \sum_{k=1}^K c_k x_k \quad (7)$$

단,  $c_k$ 는 필터의 係數이고,  $x_k$ 는 入出力 標準值를 나타낸다.

여기에 小 모듈러스들로 構成되는 RNS를 도입하면 一般의인 레지듀 乘算을 쉽게 實現할 수 있다. 이는 式 (7)을 직접 레지듀 演算으로 遂行하는 것으로서 필요한 演算은 모두 演算表에 의해서 이루어진다. 즉,

$$|y(n)| m_i = \sum_{k=1}^K |c_k| m_i \cdot |x_k| m_i |m_i| \quad i = 1, 2, \dots, N \quad (8)$$

그런데 이와 같이 小 모듈러스를 갖는 레지듀 디지털 필터에서는 서로 素인 모듈러스의 갯수가 적기 때문에 dynamic range가 작아서 信號 및 係數의 비트 수를 크게 할 수 없다. 一例로서 4비트이하의 서로 素인 모듈러스는 16, 15, 13, 11, 7 등이며 이들로 構成

되는 RNS에서 表現할 수 있는 數의 범위는 18비트를 넘지 못한다. 더욱이 dynamic range를 늘이기 위해서 모듈러스의 갯수를 증가시키면 필터의 構造가 훨씬 複雜해진다.

따라서 비트 슬라이스 알고리즘의 原理를 적용하여 하나의 디지털 信號를 여러 개의 비트 群으로 나누어 順次的으로 처리하되, 每 비트 群을 처리하는 과정에서만 레지듀 演算을 함으로써 既存의 레지듀 디지털 필터에 비해 信號 및 係數의 비트 數를 크게 할 수 있는 다음 方式을 제시하고자 한다.

먼저  $x_k$ 를 다음 式과 같이 各各이 m 비트로 構成된 몇 개의 비트 群으로 表示한다.

$$x_k = \sum_{t=0}^s 2^{tm} g_{kt} \quad k=1, 2, \dots, K \quad (9)$$

단,

$$g_{kt} = \sum_{j=tm}^{tm+m-1} 2^{j-tm} x_{kj}, \quad t=0, 1, \dots, s-1$$

$$g_{ks} = \sum_{j=sm}^{B-1} 2^{j-sm} x_{kj} - 2^{B-sm} x_{kB} \quad (10)$$

이고,  $B+1=sm+m, m>1$ 이며,  $x_{kj}$ 는  $x_k$ 의 한 비트를 나타낸다. 式(9)를 이용하여 式(7)을 다시 써 보면

$$y(n) = \sum_{t=0}^s 2^{tm} G(g_{1t}, g_{2t}, \dots, g_{kt}) \quad (11)$$

이때

$$G(g_{1t}, g_{2t}, \dots, g_{kt}) = \sum_{k=1}^K c_k g_{kt} \quad t=0, 1, \dots, s \quad (12)$$

이 方式이 비트 슬라이스 알고리즘과 다른 점은 함수 값  $G(\cdot)$ 를 미리 저장시켜 두지 않고, 레지듀 演算에 의해서 직접 計算한다는 것이다. 그러므로 필터의 係數를 바꾸어 特性을 쉽게 변화시킬 수 있다. 또 레지듀 演算은 함수 값  $G(\cdot)$ 를 計算하는 데에만 쓰이므로 동일한 RNS를 사용할 경우에 既存의 레지듀 디지털 필터<sup>[8][9]</sup>에 비해서 信號 및 係數의 비트 수를 크게 할 수 있다. 물론 이러한 잇점이 있는 반면,  $G(\cdot)$ 의 計算 時間으로 인하여 필터의 速度가 減少되기는 하지만 式(9)에 보인 바와 같이 한 번에 m 비트씩 처리함으로써 個個의 비트를 順次的으로 처리하는 것보다는 속도를 m배 정도 향상시킬 수 있다.

IV. 시뮬레이션 및 實驗

먼저 다음과 같은 特性을 갖는 4次 버터워즈 低域 通過 디지털 필터를 設計하였다.

- 통과대역 감쇠: 1dB 이하
- 저지대역 감쇠: 20dB 이상
- 통과대역:  $0 < f/f_s < 0.1$
- 저지대역:  $0.2 < f/f_s < 0.5$

이 필터의 계차 방정식은 다음과 같다.<sup>[9]</sup>

$$y_1(n) = 0.102603x_1(n) + 0.205206x_1(n-1) + 0.102603x_1(n-2) + 1.181333y_1(n-1) - 0.591745y_1(n-2)$$

$$x_2(n) = y_1(n)$$

$$y_2(n) = 0.079619x_2(n) + 0.159238x_2(n-1) + 0.079619x_2(n-2) + 0.9167y_2(n-1) - 0.235176y_2(n-2)$$

$$y(n) = y_2(n) \quad (13)$$

이와 같은 디지털 필터를 實現할 때, 既存의 레지듀 디지털 필터는 式(8)을 이용하므로, 모듈러스가 16, 15, 13, 11인 RNS ( $M \approx 2^{16}$ )를 사용할 경우 演算 結果가 항상 15비트 이내로 유지되어야 한다. 한편 비트 슬라이스 알고리즘의 原理를 적용한 레지듀 디지털 필터는 式(12)를 이용하여 함수 값  $G(\cdot)$ 를 구할 때에만 레지듀 演算을 행한다. 따라서 信號 및 係數의 비트 수를 앞의 경우보다 크게 할 수 있으므로 양자화 誤差(quantization error)를 줄임으로써 필터의 精度를 높일 수 있게 된다.

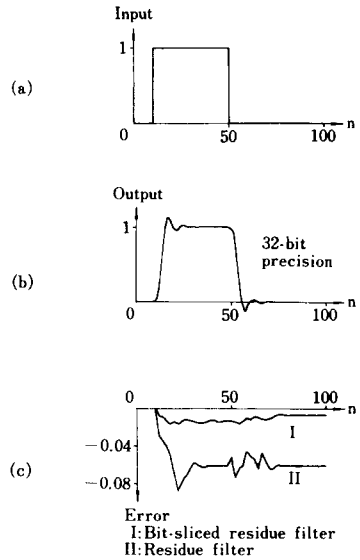


그림 1. 구형과 입력에 대한 시뮬레이션  
Fig. 1. Simulation for a square wave input.

시뮬레이션의 結果는 그림 1에 보인 바와 같다. 그림 1의 (a)는 필터의 入力 信號이고 (b)는 式(13)을 32비트의 컴퓨터로 計算하여 얻은 필터의 出力 信號이다. 그리고 (c)는 式(8)로 表現되는 既存의 레지듀 디지털 필터와 式(11)로 表現되는 비트 슬라이스 알고리즘의 原理를 적용한 레지듀 디지털 필터의 出力을

에 사용된 係數들의 레지듀 表現이다. RNS에서는 整數 演算만 가능하므로 係數에 적당한 상수를 곱해서 整數化한 다음 演算 結果를 다시 이 상수로 나누어 주어야 한다. 그런데 레지듀 演算에서 어떤 數를 모듈러스들의 곱으로 나누는 것은 R/B 변환 과정에서 수행될 수 있으므로 이 상수를 모듈러스들의 곱으로 정하는 것이 좋다.<sup>1)6)11)12)</sup> 既存의 레지듀 디지털 필터인 경우에는 式 (8)의 結果가 15비트를 넘지 않도록 하기 위해서 信號 및 係數를 7비트 정도로 하는 것이 좋으므로 係數에  $13 \times 11 = 143$ 을 곱하여 整數化하였다. 한편 비트 슬라이스 알고리즘의 原理를 적용한 레지듀 디지털 필터인 경우에는 10비트의 信號를 2비트로 된 5개의 비트 群으로 나누어 順次的으로 처리한다면, 式 (11)의 演算 과정에서 4회의 2비트 行 시프트가 일어나며 式 (12)의 結果를 R/B 변환하는 과정에서  $G(\cdot)$ 가 10비트 정도로 되도록 하기 위해서 11로 나누게 되므로 전체적으로는  $2^8 \times 11 = 2816$ 으로 나누는 셈이 된다. 따라서 係數에 이 상수를 곱하여 整數化해야 한다.

實驗에서는 이미 앞에서 제시한 방식에 따라 중속 접속형 4次 디지털 필터를 構成하였다. 여기에서는 시뮬레이션에서와 꼭 같이 式 (13)으로 表示되는 버터워즈 低域通過 필터를 實現하되 모듈러스가 16, 15, 13, 11인 RNS를 사용하였고 10비트의 入力 信號를 2비트의 비트 群으로 나누어 처리하였다.

표 1. 레지듀 數로 表現한 필터의 係數

Table 1. The filter coefficient with representation of residue number.

式 (13)의 係數		既存의 레지듀 디지털 필터		비트 슬라이스 알고리즘의 原理를 적용한 레지듀 디지털 필터	
		整數化된 값 ( $\times 143$ )	레지듀 表現 ( $X_{13}, X_{11}, X_{13}, X_{11}$ )	整數化된 값 ( $\times 2816$ )	레지듀 表現 ( $X_{13}, X_{11}, X_{13}, X_{11}$ )
Section 1	0.102603	15	15, 0, 2, 4	289	1, 4, 3, 3
	0.205206	29	13, 14, 3, 7	578	2, 8, 6, 6
	0.102603	15	15, 0, 2, 4	289	1, 4, 3, 3
	1.181333	169	9, 4, 0, 4	3327	15, 12, 12, 5
	-0.591745	-85	11, 5, 6, 3	-1666	14, 14, 11, 6
Section 2	0.079619	11	11, 11, 11, 0	224	0, 14, 3, 4
	0.159238	23	7, 8, 10, 1	448	0, 13, 6, 8
	0.079619	11	11, 11, 11, 0	224	0, 14, 3, 4
	0.9167	131	3, 11, 1, 10	2581	5, 1, 7, 7
	-0.235176	-34	14, 11, 5, 10	-662	10, 13, 1, 9

필터의 構造는 그림 2와 같이 된다. 레지듀 演算裝置 (residue arithmetic unit)는 각 모듈러스에 대한 副演算裝置들로 되어 있고 이들은 그림 3에 보인 바와 같이 각각 레지듀 加算表  $A_i$ 와 乘算表  $M_i$ 로 構成되어 있는데 여기에는  $256 \times 4$ 비트 PROM (82S129)이 사용되었다. 그리고 레지듀 數로 表現된 함수 값  $G(\cdot)$ 는 R/B 변환기 (residue-to-binary converter)<sup>1)6)11)12)</sup>에서 2進數로 변환되어 出力  $y(n)$ 이 計算된다.

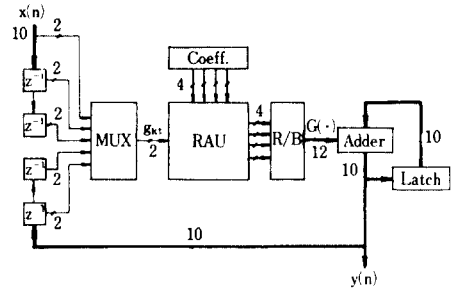


그림 2. 레지듀 디지털 필터의 構成

Fig. 2. Schematic diagram of a residue digital filter.

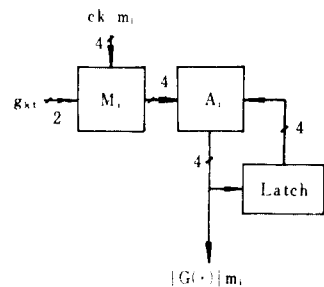


그림 3. 모듈러스  $m_i$ 에 대한 레지듀 副演算裝置

Fig. 3. Residue arithmetic sub-unit for modulus  $m_i$ .

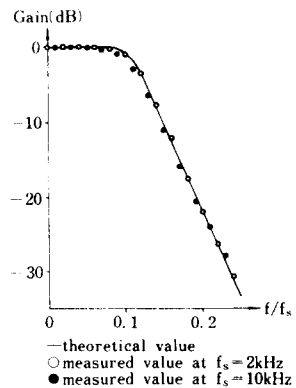


그림 4. 필터의 周波數 應答

Fig. 4. Frequency responses of the filter.

實驗 結果로서 2 KHz 및 10 KHz의 標本化 周波數에 대한 필터의 特性을 측정하여 그림 4에 表示하였는데 通過帶域에서 實驗值가 理論值에 1 dB 이내로 일치함을 보여 주고 있다.

V. 結 論

디지털 필터를 實現할 때 레지듀 演算을 사용하면 各各의 레지듀 자리수에 대해 독립적인 演算이 가능하므로 긴 wordlength의 演算을 여러 개의 짧은 word-length의 演算으로써 수행할 수 있다. 따라서 논리 회로 대신 ROM을 사용한 演算表로써 이를 實現할 수 있으며 演算 速度가 빠르다. 또한 構造도 간단하여 비용이 적게 드는 長點이 있다.

本 論文에서는 이와 같은 長點을 살리기 위해 4비트 이하의 小 모듈러스들로 構成된 RNS를 도입한 레지듀 디지털 필터를 設計하되, 비트 슬라이스 알고리즘의 原理를 적용하여 信號를 여러 개의 비트 群으로 나누어 順次的으로 처리함으로써 既存의 레지듀 디지털 필터에 비해 信號 및 係數의 비트 수를 크게 늘일 수 있었다. 이와 동시에 함수 값  $G(\cdot)$ 를 미리 ROM에 저장해 두는 대신 이를 레지듀 演算에 의해서 직접 計算함으로써 필터의 特性을 쉽게 변경할 수 있다

표 2. 종속접속형 4次 디지털 필터의 實現 방식의 비교

Table 2. Comparison of the realization method of cascaded 4th order digital filter.

	既存의 레지듀 實現 방식	비트 슬라이스 알고리즘의 原理를 적용한 레지듀 實現 방식
係數의 비트數	약 8비트	약 12비트
信號의 비트數	7비트	10비트
最大 標本化 周波數	수 MHz	수십 KHz
誤差의 원인	<ul style="list-style-type: none"> <li>• 주로 양자화 誤差</li> <li>• Scaling時的 truncation 誤差(演算時的 誤差는 없음)</li> </ul>	<ul style="list-style-type: none"> <li>• 주로 binary 演算(add-shift)時的 truncation 誤差</li> <li>• Scaling時的 truncation 誤差(양자화 誤差는 비교적 작음)</li> </ul>
Hardware 상의 특 징	<ul style="list-style-type: none"> <li>• RNS 加算 및 乘算</li> <li>• Binary 演算 不必要</li> <li>• B/R 변환 및 R/B 변환</li> </ul>	<ul style="list-style-type: none"> <li>• RNS 加算 및 乘算</li> <li>• binary 演算(add-shift) 必要</li> <li>• R/B 변환(B/R 변환은 不必要)</li> </ul>

록 했다.

그러나 함수 값  $G(\cdot)$ 의 計算 및 R/B 변환으로 인해 필터의 信號 처리 速度가 느려지는 短點이 있다. 여기서는 信號를 2비트씩 順次的으로 처리함으로써 標本化 周波數를 20 KHz까지 높일 수 있었다. 표 2는 모듈러스가 16, 15, 13, 11인 RNS를 사용했을 경우에, 既存의 레지듀 實現 방식과 비트 슬라이스 알고리즘의 原理를 적용한 레지듀 實現 방식을 종속 접속형 4次 디지털 필터에 대해 비교한 것이다.

參 考 文 獻

- [1] A. Peled, B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456-462, Dec. 1974.
- [2] N.S. Szabo, R.I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*. McGraw-Hill, New York, pp. 1-51, 1967.
- [3] W.K. Jenkins, B.J. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Trans. Circuit Syst.*, vol. CAS-24, pp. 191-201, Apr. 1977.
- [4] M.A. Soderstrand, "A high-speed low-cost recursive digital filter using residue number arithmetic," *Proc. IEEE*, vol. 65, no. 7, July 1977.
- [5] W.K. Jenkins, "Recent advances in residue number techniques for recursive digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 19-30, Feb. 1979.
- [6] G.A. Jullien, "Residue number scaling and other operations using ROM arrays," *IEEE Trans. Comput.*, vol. C-27, pp. 325-336, Apr. 1978.
- [7] A. Baraniecka, G.A. Jullien, "On decoding techniques for residue number system realizations of digital signal processing hardware," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 935-936, Nov. 1978.
- [8] W.K. Jenkins, "Techniques for residue-to-analog conversion for residue-encoded digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 555-562, July 1978.
- [9] A. Antoniou, *Digital Filters: Analysis and Design*. McGraw-Hill, New York, pp. 196-217, 1979. \*