Recurrent Formula and Property of Generalized Pascal Matrix

Tian-Bo Deng†, Sorawat Chivapreecha‡, and Kobchai Dejhan‡

†Department of Information Science, Toho University, Japan
E-mail: deng@is.sci.toho-u.ac.jp

‡Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

Abstract: This paper derives a recurrence formula for recursively computing the inner elements of the generalized Pascal matrix from its boundary ones so that all the elements of the whole generalized Pascal matrix can be easily generated through utilizing their neighbourhood. We also reveal and prove an interesting property of the generalized Pascal matrix. Numerical examples are given to verify the recurrence formula and property.

1. Introduction

The so-called s-z transformations provide a simple way for analyzing and designing discrete-time (DT) linear systems in the z-domain through utilizing continuous-time (CT) linear systems in the s-domain, and vice versa. For designing infinite-impulse-response (IIR) DT linear filters, it is the most common practice and successful way to convert a CT filter to a DT filter. This is because CT filter design techniques have been highly advanced, thus the existing closed-form solutions and well-documented tables can be adopted directly for designing DT filters. One can design a DT filter with specified design requirements by performing s-z transformations, and the resulting DT filters can preserve the desired characteristics of the original CT filters [1], [2]. For s-z transformations, a transformation matrix called *generalized Pascal matrix* is required. The main objectives of this paper are as follows.

1) We derive a recurrence formula for computing the generalized Pascal matrices, which uses the general first-order s-z transformation model [3] and thus embeds the bilinear (BL), backward-difference (BD), forward-difference (FD), and parametric BD-BL transformations. As a result, the FFT-based numerical method proposed in [3] can be substituted by the new recurrence formula because the latter is analytical and more computationally efficient. The recurrence formula starts from the boundary elements (first row and first column) of the generalized Pascal matrix, and then computes the inner elements recursively, where only the neighbouring elements are used.
2) We also reveal and prove an interesting property of the generalized Pascal matrix. Numerical examples are given to verify the recurrence formula and interesting property.

2. s-z Transformation

We assume that the s-domain transfer function

$$H_{\mathrm{CT}}(s) = \frac{\sum_{i=0}^{N} A_i s^i}{\sum_{i=0}^{N} B_i s^i} = \frac{N_{\mathrm{CT}}(s)}{D_{\mathrm{CT}}(s)} \tag{1}$$

with

$$N_{\mathrm{CT}}(s) = \sum_{i=0}^{N} A_i s^i, \quad D_{\mathrm{CT}}(s) = \sum_{i=0}^{N} B_i s^i$$

and z-domain transfer function

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{\sum_{k=0}^{N} b_k z^{-k}} = \frac{N(z)}{D(z)} \tag{2}$$

with

$$N(z) = \sum_{k=0}^{N} a_k z^{N-k}, \quad D(z) = \sum_{k=0}^{N} b_k z^{N-k} \tag{3}$$

are related to each other through the first-order s-z transformation

$$s = (-u) \cdot \frac{z-1}{wz-v} \Leftrightarrow z = \frac{u+vs}{u+ws} \tag{4}$$

where u, $v \neq w$ are real constants [3]. Substituting (4) into (1) yields

$$\widehat{H}(z) = \frac{\sum_{i=0}^{N} A_i (-u)^i \left(\frac{z-1}{wz-v}\right)^i}{\sum_{i=0}^{N} B_i (-u)^i \left(\frac{z-1}{wz-v}\right)^i} = \frac{\widehat{N}(z)}{\widehat{D}(z)} \tag{5}$$

with

$$\begin{aligned} \widehat{N}(z) &= \sum_{i=0}^{N} u^i A_i \cdot (-1)^{N-i} P_i(z) \\ \widehat{D}(z) &= \sum_{i=0}^{N} u^i B_i \cdot (-1)^{N-i} P_i(z) \\ P_i(z) &= (z-1)^i (wz-v)^{N-i}. \end{aligned} \tag{6}$$

The Nth degree polynomial $P_i(z)$ can be expressed as

$$P_i(z) = \sum_{k=0}^{N} p_{i,k} z^{N-k}. \tag{7}$$

By using the binomial theorem, we can expand $(z-1)^i$ and $(wz-v)^{N-i}$ as

$$(z-1)^i = \sum_{n=0}^{i} \binom{i}{n} z^{i-n} \cdot (-1)^n$$

$$(wz-v)^{N-i} = \sum_{n=0}^{N-i} \binom{N-i}{n} (wz)^{(N-i)-n} (-v)^n.$$

Thus,

$$\begin{aligned} p_{i,k} &= \sum_{n=0}^{k} \binom{N-i}{n} \binom{i}{k-n} (-1)^{k-n} (-v)^n w^{(N-i)-n} \\ &= \sum_{n=0}^{k} \binom{N-i}{n} \binom{i}{k-n} (-1)^k v^n w^{(N-i)-n} \end{aligned} \tag{8}$$

and

$$(-1)^{N-i} p_{i,k} = \sum_{n=0}^{k} \binom{N-i}{n} \binom{i}{k-n} v^n w^{k-n} \cdot (-w)^{N-i-k}.$$

Consequently, the numerator $\widehat{N}(z)$ in (5) can be expressed as

$$\widehat{N}(z) = \sum_{k=0}^{N} \widehat{a}_k z^{N-k} \tag{9}$$

with

$$\widehat{a}_k = \sum_{i=0}^{N} M(k,i) \widetilde{A}_i \cdot (-w)^{N-i-k} \tag{10}$$

and

$$\begin{aligned} M(k,i) &= \sum_{n=0}^{k} \binom{N-i}{n} \binom{i}{k-n} v^n w^{k-n} \\ \widetilde{A}_i &= u^i A_i. \end{aligned} \tag{11}$$

Using (11), we can prove that

$$M(k,i) = \begin{cases} 1 & \text{for } k=0 \quad \text{(first row)} \\ v^{N-i} w^i & \text{for } k=N \quad \text{(last row)} \\ \binom{N}{k} v^k & \text{for } i=0 \quad \text{(first column)} \\ \binom{N}{k} w^k & \text{for } i=N \quad \text{(last column).} \end{cases} \tag{12}$$

Therefore, the matrix $\mathbf{M}$ takes the form

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \binom{N}{1} v^1 & \cdots & \cdots & \cdots & \binom{N}{1} w^1 \\ \binom{N}{2} v^2 & \cdots & \cdots & \cdots & \binom{N}{2} w^2 \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \binom{N}{N} v^N & v^{N-1} w^1 & \cdots & v^1 w^{N-1} & \binom{N}{N} w^N \end{bmatrix}. \tag{13}$$

In [3], the matrix $\mathbf{M}$ is called the *generalized Pascal matrix*. For the BL, BD, and parametric BD-BL transformations ($w = -1$), (10) can be rewritten as

$$\widehat{a}_k = \sum_{i=0}^{N} M(k,i) \widetilde{A}_i$$

or we can express the above equation in matrix form as

$$\widehat{\mathbf{a}} = \mathbf{M} \widetilde{\mathbf{A}} \tag{14}$$

with

$$\widehat{\mathbf{a}} = \begin{bmatrix} \widehat{a}_0 \\ \widehat{a}_1 \\ \vdots \\ \widehat{a}_N \end{bmatrix}, \quad \widetilde{\mathbf{A}} = \begin{bmatrix} \widetilde{A}_0 \\ \widetilde{A}_1 \\ \vdots \\ \widetilde{A}_N \end{bmatrix} = \begin{bmatrix} u^0 A_0 \\ u^1 A_1 \\ \vdots \\ u^N A_N \end{bmatrix}. \tag{15}$$

3. Recurrence Formula

In [3], it is shown that the elements $M(k,i)$ of the generalized Pascal matrix $\mathbf{M}$ in (13) can be computed through using the FFT and IFFT. Although this idea is interesting, the final results are numerical solutions and not in a closed-form. In this section, we derive a recurrence formula for computing the elements $M(k,i)$, where the parameters v and w can take any real values, but $v \neq w$. Consequently, the BL, BD, FD, and parametric BD-BL transformations can be regarded as special cases. As compared with the DFT-based numerical algorithm [3], the recurrence formula can recursively compute the elements $M(k,i)$ in a closed-form, and the computational complexity can also be reduced significantly. We start from the element expression

$$M(k,i) = \sum_{n=0}^{k} \binom{N-i}{n} \binom{i}{k-n} v^n w^{k-n}. \tag{16}$$

Utilizing the first row and first column in (12), we can recursively compute all the elements $M(k,i)$ using the closed-form recurrence formula

$$M(k,i) = M(k,i-1) + wM(k-1,i-1) - vM(k-1,i) \tag{17}$$

which can be proved as follows. Since

$$\binom{N-(i-1)}{n} = \binom{N-i+1}{n} = \binom{N-i}{n-1} + \binom{N-i}{n}$$

then we have

$$\binom{N-i}{n} = \binom{N-(i-1)}{n} - \binom{N-i}{n-1}$$

and

$$\binom{N-i}{n}\binom{i}{k-n} = \binom{N-(i-1)}{n}\binom{i}{k-n} - \binom{N-i}{n-1}\binom{i}{k-n}. \tag{18}$$

Substituting the identity

$$\binom{i}{k-n} = \binom{i-1}{k-n-1} + \binom{i-1}{k-n}$$

into (18) leads to

$$\binom{N-i}{n}\binom{i}{k-n}$$
$$=\binom{N-(i-1)}{n}\binom{i-1}{k-n}+\binom{N-(i-1)}{n}\binom{i-1}{(k-1)-n}$$
$$-\binom{N-i}{n-1}\binom{i}{k-n}. \tag{19}$$

Thus, we obtain

$$M(k,i)=\sum_{n=0}^{k}\binom{N-i}{n}\binom{i}{k-n}v^n w^{k-n} = \alpha_1+\alpha_2-\alpha_3 \tag{20}$$

with

$$\alpha_1=\sum_{n=0}^{k}\binom{N-(i-1)}{n}\binom{i-1}{k-n}v^n w^{k-n}=M(k,i-1)$$

$$\alpha_2=\sum_{n=0}^{k}\binom{N-(i-1)}{n}\binom{i-1}{(k-1)-n}v^n w^{k-n} = \sum_{n=0}^{k-1}\binom{N-(i-1)}{n}\binom{i-1}{(k-1)-n}v^n w^{(k-1)-n}\cdot w = wM(k-1,i-1)$$

and

$$\alpha_3=\sum_{n=0}^{k}\binom{N-i}{n-1}\binom{i}{k-n}v^n w^{k-n} = \sum_{n=1}^{k}\binom{N-i}{n-1}\binom{i}{k-n}v^n w^{k-n}.$$

If we let $n'=n-1$, then $n=n'+1$, and α_3 becomes

$$\alpha_3=\sum_{n'=0}^{k-1}\binom{N-i}{n'}\binom{i}{(k-1)-n'}v^{n'+1}w^{k-(n'+1)} = \sum_{n'=0}^{k-1}\binom{N-i}{n'}\binom{i}{(k-1)-n'}v^{n'}w^{(k-1)-n'}\cdot v = vM(k-1,i).$$

Substituting α_1, α_2, and α_3 into (20) yields (17). This recurrence formula indicates that once the constants v and w are given, all the elements $M(k,i)$ can be recursively computed by using their left and top neighbourhood. For computing each $M(k,i)$, at most 2 multiplications and 2 additions are required. Using the recurrence formula (17), the BL, BD, FD, and parametric BD-BL transformations can be simplified as follows.

♦ For BL transformation, $v=1$, $w=-1$, then

$$M_{\mathrm{BL}}(k,i)=M_{\mathrm{BL}}(k,i-1)-M_{\mathrm{BL}}(k-1,i-1)-M_{\mathrm{BL}}(k-1,i). \tag{21}$$

♦ For BD transformation, $v=0$, $w=-1$, then we have

$$M_{\mathrm{BD}}(k,i)=M_{\mathrm{BD}}(k,i-1)-M_{\mathrm{BD}}(k-1,i-1). \tag{22}$$

♦ For FD transformation, $v=1$, $w=0$, then

$$M_{\mathrm{FD}}(k,i)=M_{\mathrm{FD}}(k,i-1)-M_{\mathrm{FD}}(k-1,i). \tag{23}$$

♦ For parametric BD-BL transformation, $v=r$, $w=-1$, then

$$M_{\mathrm{P}}(k,i)=M_{\mathrm{P}}(k,i-1)-M_{\mathrm{P}}(k-1,i-1)-rM_{\mathrm{P}}(k-1,i). \tag{24}$$

As an example, let us consider $N=3$ for the parametric BD-BL case. Using the recurrence formula (24), we can verify

$$\mathbf{M}_{\mathrm{P}}=\begin{bmatrix}1 & 1 & 1 & 1\\ 3r & 2r-1 & r-2 & -3\\ 3r^2 & r^2-2r & -2r+1 & 3\\ r^3 & -r^2 & r & -1\end{bmatrix}. \tag{25}$$

Finally, some remarks should be given on the computational complexity (computation time). If the FFT-based algorithm [3] is used, while $(N+1)$ is not equal to an integer power of two, then zero-padding is necessary before performing the FFT. Therefore, the computational complexity also depends on how far the number $(N+1)$ from the smallest integer power of two towards infinity. The most efficient case for the algorithm [3] is that $(N+1)$ is exactly equal to the integer power of two. Assume that $(N+1)=2^p$, p is an integer. First, the vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ in [3] must be computed before the FFT. After generating the vectors $\mathbf{a}_i$ and $\mathbf{b}_i$, the FFT-IFFT process requires

$$3(N+1)\log_2(N+1)+(N+1)=(N+1)(3p+1)$$

complex multiplications and

$$3(N+1)\log_2(N+1)+N\approx(N+1)(3p+1)$$

complex additions. Since one complex multiplication requires 4 real multiplications, and one complex addition requires 2 real additions, thus the FFT-based algorithm [3] requires $4(N+1)(3p+1)$ real multiplications and $2(N+1)(3p+1)$ real additions. In contrast, the recurrence formula (17) only requires at most $2N$ real multiplications and $2N$ real additions. As a result, the ratio of real multiplications is

$$\frac{2N}{4(N+1)(3p+1)}\approx\frac{1}{2(3p+1)} \tag{26}$$

and the ratio of real additions

$$\frac{2N}{2(N+1)(3p+1)}\approx\frac{1}{3p+1}. \tag{27}$$

Therefore, the recurrence formula requires lower computational complexity than the FFT-based algorithm [3]. For example, if $N=16$, and $v=0.5$, $w=-1$, then 15 zeros must be padded before the FFT-IFFT process, and the FFT-based algorithm [3] requires 137002 Flops including all the multiplications and additions to complete the computation of the generalized Pascal matrix $\mathbf{M}$, while the recurrence formula only requires 3355 Flops, the ratio of the multiplications is about 0.0245, i.e., only about 2.45% of the multiplications are required by using the recurrence formula. Taking the $\mathbf{M}$ computed directly from (16) as the true result, then the FFT-based algorithm [3] and the recurrence formula (17) lead to normalized root-mean-squared (RMS) approximation errors $1.97 \times 10^{-13}\%$, and 0, respectively, i.e., no errors are caused.

4. Interesting Property

In this section, we prove an interesting property that the sum of each column (except the first column) elements of the generalized Pascal matrix $\mathbf{M}$ (for $w = -1$) is zero. Assume that S_i denotes the sum of the $(i+1)$th column elements, i.e.,

$$S_i = \sum_{k=0}^{N} M(k,i), \qquad i = 0, 1, \cdots, N$$

then we can prove the following interesting property.

Property: If $w = -1$, $v \neq w$, then

$$S_i = \begin{cases} (1+v)^N, & \text{for } i = 0 \\ 0, & \text{for } i = 1, 2, \cdots, N. \end{cases} \tag{28}$$

Proof: For $w = -1$, the recurrence formula (17) can be modified to

$$M(k,i) + vM(k-1,i) = M(k,i-1) - M(k-1,i-1).$$

For $i \geq 1$, summing up the two sides separately leads to

$$\sum_{k=0}^{N} [M(k,i) + vM(k-1,i)] = \sum_{k=0}^{N} [M(k,i-1) - M(k-1,i-1)].$$

The left-hand side can be rewritten as

$$\sum_{k=0}^{N} [M(k,i) + vM(k-1,i)] = (1+v)S_i - vM(N,i)$$

and the right-hand side becomes

$$\sum_{k=0}^{N} [M(k,i-1) - M(k-1,i-1)] = M(N,i-1)$$

where we assume that the boundary elements $M(-1,i)$ and $M(-1,i-1)$ are zero. Thus, we have

$$(1+v)S_i - vM(N,i) = M(N,i-1)$$

i.e.,

$$S_i = \frac{vM(N,i) + M(N,i-1)}{1+v}. \tag{29}$$

Moreover, as mentioned in (12), since

$$M(N,i) = v^{N-i}w^i = v^{N-i}(-1)^i$$

and

$$\begin{aligned} M(N,i-1) &= v^{N-(i-1)}(-1)^{i-1} \\ &= -v \cdot v^{N-i}(-1)^i \\ &= -vM(N,i) \end{aligned}$$

thus (29) becomes

$$S_i = \frac{vM(N,i) - vM(N,i)}{1+v} = 0, \qquad i = 1, 2, \cdots, N. \tag{30}$$

If $i = 0$, then

$$M(k,i) = M(k,0) = \binom{N}{k} v^k$$

thus we have

$$S_0 = \sum_{k=0}^{N} M(k,0) = \sum_{k=0}^{N} \binom{N}{k} v^k = (1+v)^N. \tag{31}$$

Combining (30) with (31) together completes the proof of the *property* (28). Obviously, the BL, BD, and parametric BD-BL transformations ($w = -1$) have the property (28). As an example, let us check the generalized Pascal matrix $\mathbf{M}_\mathrm{P}$ in (25) for the parametric BD-BL case, where $N = 3$ and $v = r = 0.5$. Clearly,

$$S_i = \begin{cases} (1+r)^N = 1.5^N = 3.375, & \text{for } i = 0 \\ 0, & \text{for } i = 1, 2, 3, 4. \end{cases}$$

Therefore, the generalized Pascal matrix (25) has the property (28).

5. Conclusion

In this paper, we have derived a recurrence formula for recursively computing the generalized Pascal matrix from its boundary elements, which simplifies the generations of the generalized Pascal matrix. We have also proved an interesting property of the generalized Pascal matrix for $w = -1$, and numerical examples have been included to verify the recurrence formula and interesting property.

References

[1] B. Psenicka, F. Garcia-Ugalde, and A. Herrera-Camacho, “The bilinear z transform by Pascal matrix and its application in the design of digital filters,” *IEEE Trans. Signal Process. Lett.*, vol. 9, no. 11, pp. 368-370, Nov. 2002.

[2] S. Chivapreecha and K. Dejhan, “Pascal matrix operation for bilinear s-z with frequency transformation,” *Proc. ITC-CSCC’06*, vo. III, pp. 129-132, Chiang Mai, Thailand, July 2006.

[3] D. Biolek and V. Biolkova, “Algorithmic s-z transformations for continuous-time to discrete-time filter conversion,” *Proc. IEEE ISCAS’01*, vol. I, pp. 588-590, Sydney, Australia, May 2001.