DESIGN OF DIGITAL IIR INTEGRATOR USING B-SPLINE INTERPOLATION AND GAUSS-LEGENDRE INTEGRATION RULE

Chien-Cheng Tseng[1] *and Su-Ling Lee*[2]

[1]Depart. of Computer and Communication Engineering
National Kaohsiung First University of Sci. and Tech.
Kaohsiung, Taiwan
tcc@ccms.nkfust.edu.tw

[2]Depart. of Computer Sci. and Information Engi.
Chung-Jung Christian University
Tainan, Taiwan
lilee@mail.cjcu.edu.tw

ABSTRACT

In this paper, the design of digital IIR integrator is investigated. First, the B-spline interpolation method is described. Then, non-integer delay sample estimation of discrete-time sequence is derived by using B-spline interpolation approach. Next, the Gauss-Legendre integration rule and non-integer delay sample estimation are applied to obtain the transfer function of digital integrator. Finally, some numerical comparisons with conventional digital integrators are made to demonstrate the effectiveness of this new design approach.

1. INTRODUCTION

Digital integrators are useful devices in the application areas of control, radar and biomedical engineering [1]-[4]. The ideal frequency response of digital integrator is given by

$$D(\omega) = \frac{1}{j\omega} e^{-j\omega I} \tag{1}$$

where I is a prescribed integer delay. The problem is how to design a digital filter such that its frequency response fits $D(\omega)$ as well as possible. So far, the methods of digital integrator design can generally be classified into two categories. One is the linear phase FIR filter approach in which the filter coefficients are obtained by using maximal flatness constraints [1][2], the other is the IIR filter method in which the filter coefficients are determined directly from well-known numerical integration rule [3][4]. In [3], Ngo presented a third-order digital integrator whose transfer function is given by

$$F_1(z) = \frac{z^{-I}}{24} \frac{9 + 19\, z^{-1} - 5 z^{-2} + z^{-3}}{1 - z^{-1}} \tag{2}$$

In [4], Tseng and Lee have used Richardson extrapolation and polyphase decomposition to design digital integrators. From Eq.(62) in [4], the transfer function of a typical forth-order integrator is given by

$$F_2(z) = \frac{z^{-I+1}}{139968} \frac{\begin{pmatrix} -3693 + 67260\, z^{-1} + 88650\, z^{-2} \\ -14388\, z^{-3} + 2139\, z^{-4} \end{pmatrix}}{1 - z^{-1}} \tag{3}$$

From Eq.(1), it is clear that the gain of integrator at zero frequency $\omega = 0$ is infinity, so the above transfer functions have one pole at $z = 1$. On the other hand, B-spline interpolation has been successfully used in numerical interpolation, image processing, digital filter design, digital filter bank, computer graphics, and analog-to-digital conversion [5]-[7]. The early work on the B-spline theory and implementation is surveyed in the tutorial paper [8]. Thus, it is an interesting topic that uses B-spline interpolation method to design digital IIR integrators. The purpose of this paper is to study this topic. As a result, the design error can be reduced by suitably choosing the degree of B-spline function. The numerical comparisons with conventional digital integrators also show the effectiveness of this new approach.

2. B-SPLINE INTERPOLATION

In this section, B-spline function is first reviewed briefly. Then, the B-spline interpolation method is described. Splines are piecewise polynomials with pieces which are connected smoothly. The jointing points of the polynomials are called knots. The normalized symmetrical, bell-shaped B-spline functions of degree p with $p+2$ equally spaced knots are defined by

$$\beta^p(t) = \underbrace{\beta^0(t) * \beta^0(t) * \cdots * \beta^0(t)}_{(p+1)\ times} \tag{4}$$

where rectangular pulse

$$\beta^0(t) = \begin{cases} 1 & |t| < \frac{1}{2} \\ \frac{1}{2} & t = \frac{1}{2} \\ 0 & |t| > \frac{1}{2} \end{cases} \tag{5}$$

and * denotes the convolution operator. After some manipulation, the function $\beta^p(t)$ can be written as

$$\beta^p(t) = \frac{1}{p!} \sum_{i=0}^{p+1} (-1)^i \binom{p+1}{i} (t + \tfrac{p+1}{2} - i)^p u(t + \tfrac{p+1}{2} - i) \tag{6}$$

where $u(t)$ is the unit step function given by

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{7}$$

The result in Eq.(6) clearly shows that $\beta^p(t)$ is a piecewise polynomial of degree p. For example, when $p = 3$, the closed-form representation of the cubic B-spline is given by

$$\beta^3(t) = \begin{cases} \frac{2}{3} - |t|^2 + \frac{|t|^3}{2} & |t| < 1 \\ \frac{(2-|t|)^3}{6} & 1 \le |t| < 2 \\ 0 & |t| \ge 2 \end{cases} \quad (8)$$

in which both pieces are third-degree polynomials. Moreover, Fig.1 shows the normalized B-splines of first four degree. Clearly, $\beta^p(t)$ only has non-zero value in the interval $(-\frac{p+1}{2}, \frac{p+1}{2})$. After describing the B-spline function, let us study B-spline interpolation method below: Given a set of $N+1$ sampled points (t_k, s_k) $(k = 0,1,\cdots,N)$, the interpolation problem is to find a function $s(t)$ that satisfies the interpolation condition

$$s(t_k) = s_k \qquad k = 0,1,2,\cdots,N \quad (9)$$

For B-spline interpolation method, the function $s(t)$ is characterized in terms of the following B-spline model:

$$s(t) = \sum_{k=0}^{N} w_k \beta^p(t - t_k) \quad (10)$$

That is, the function $s(t)$ is represented as a linear combination of the shifted B-spline functions. Substituting the interpolation condition of Eq.(9) into Eq.(10), we get the following simultaneous linear equation

$$\begin{bmatrix} \phi_{00} & \phi_{01} & \phi_{02} & \cdots & \phi_{0N} \\ \phi_{10} & \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{N0} & \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_N \end{bmatrix} \quad (11)$$

where $\phi_{mk} = \beta^p(t_m - t_k)$. Let vectors S and W be

$$S = [s_0 \quad s_1 \quad \cdots \quad s_N]^T \quad (12a)$$

$$W = [w_0 \quad w_1 \quad \cdots \quad w_N]^T \quad (12b)$$

and Φ denotes the $(N+1)$-by-$(N+1)$ matrix in the left side of Eq.(11), then Eq.(11) can be rewritten as

$$\Phi W = S \quad (13)$$

Thus, the unknown vector W is given by

$$W = \Phi^{-1} S \quad (14)$$

Once W has been obtained, the function $s(t)$ in Eq.(10) is known. So, $s(t)$ is computable for the given t. Finally, an example is use to illustrate the B-spline interpolation method. The data s_k is obtained by uniformly sampling the sinusoidal function $\cos(0.04\pi t)$, that is, $t_k = k$ and $s_k = \cos(0.04\pi k)$. The B-spline function is chosen as $\beta^3(t)$. The number of points are $N+1 = 101$. Fig.2(a)(b) shows the interpolated function $s(t)$ in Eq.(10) and the sinusoidal function $\cos(0.04\pi t)$. It is clear that both functions look almost the same. To observe where the errors occur, Fig.2(c) shows the absolute errors $|s(t) - \cos(0.04\pi t)|$. Clearly, the errors are very small except at the edge points $t = 0$ and $t = 100$.

3. NON-INTEGER DELAY SAMPLE ESTIMATION

In this section, we will use B-spline interpolation method to solve non-integer delay sample estimation problem because the proposed IIR integrator design method is based on this estimation method. The problem to be studied is how to estimate non-integer delay sample $s(n-I-d)$ from the given integer delay samples $s(n)$, $s(n-1)$, $s(n-2)$,..., $s(n-N)$, where I and N are integers and d is a real number in the interval $[0,1]$. And, I is usually chosen in the range $[0, N-1]$. In this paper, we use the weighted average approach to achieve the purpose, that is, non-integer delay sample is estimated by

$$s(n-I-d) = \sum_{m=0}^{N} h(m,d) s(n-m) \quad (15)$$

Thus, the remaining problem is how to use the B-spline interpolation method in the preceding section to determine the weights $h(m,d)$. To solve this problem, let us use the following substitution:

$$t_k \mapsto n-k \quad (16a)$$

$$s_k \mapsto s(n-k). \quad (16b)$$

Using the above substitution, the B-spline interpolation formula in Eq.(10) becomes

$$s(t) = \sum_{k=0}^{N} w_k \beta^p(t - (n-k)) \quad (17)$$

and the simultaneous equation in Eq.(11) reduces to

$$\begin{bmatrix} \beta^p(0) & \beta^p(1) & \beta^p(2) & \cdots & \beta^p(N) \\ \beta^p(-1) & \beta^p(0) & \beta^p(1) & \cdots & \beta^p(N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^p(-N) & \beta^p(-N+1) & \beta^p(-N+2) & \cdots & \beta^p(0) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-N) \end{bmatrix} \quad (18)$$

This equation can be shorten as the form $\Phi W = S$, as described in Eq.(13). Clearly, Φ is a Toeplitz matrix. Let the inverse of matrix Φ be denoted by

$$\Phi^{-1} = \begin{bmatrix} \alpha_{00} & \alpha_{01} & \alpha_{02} & \cdots & \alpha_{0N} \\ \alpha_{10} & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{N0} & \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} \end{bmatrix} \quad (19)$$

then the solution of simultaneous equation in Eq.(18) is given by

$$\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} = \Phi^{-1} \begin{bmatrix} s(n) \\ s(n-1) \\ \vdots \\ s(n-N) \end{bmatrix} = \begin{bmatrix} \sum_{m=0}^{N} \alpha_{0m} s(n-m) \\ \sum_{m=0}^{N} \alpha_{1m} s(n-m) \\ \vdots \\ \sum_{m=0}^{N} \alpha_{Nm} s(n-m) \end{bmatrix} \quad (20)$$

This result implies that

$$w_k = \sum_{m=0}^{N} \alpha_{km} s(n-m) \qquad k = 0,1,2,\cdots,N \quad (21)$$

Substituting Eq.(21) into Eq.(17), we get

$$s(t) = \sum_{k=0}^{N} w_k \beta^p\big(t-(n-k)\big)$$
$$= \sum_{k=0}^{N}\left(\sum_{m=0}^{N} \alpha_{km} s(n-m)\right)\beta^p\big(t-(n-k)\big) \quad (22)$$
$$= \sum_{m=0}^{N}\left(\sum_{k=0}^{N} \alpha_{km} \beta^p\big(t-(n-k)\big)\right)s(n-m)$$

Taking $t = n - I - d$, the above equation reduces to

$$s(n-I-d) = \sum_{m=0}^{N}\left(\sum_{k=0}^{N}\alpha_{km}\beta^p(k-I-d)\right)s(n-m) \quad (23)$$

Compared Eq.(23) with Eq.(15), the weights $h(m,d)$ are given by

$$h(m,d) = \sum_{k=0}^{N} \alpha_{km}\beta^p(k-I-d) \quad (24)$$

Now, let us summarized the estimation procedure below: Given the B-spline function $\beta^p(t)$ with degree p, integer N, and delay $I+d$, the procedure to estimate non-integer delay sample $s(n-I-d)$ from the given integer delay samples $s(n)$, $s(n-1)$, $s(n-2)$,..., $s(n-N)$ is summarized below:

Step 1: Compute the matrix Φ whose elements are given by $\phi_{mk} = \beta^p(k-m)$.

Step2: Calculate the inverse matrix Φ^{-1} with element α_{km}.

Step 3: Use Eq.(24) to compute the weights $h(m,d)$.

Step 4: The non-integer delay sample is estimated by $s(n-I-d) = \sum_{m=0}^{N} h(m,d)s(n-m)$.

4. DESIGN OF DIGITAL INTEGRATOR USING GAUSS-LEGENDRE INTEGRATION RULE

In this section, the Gauss-Legendre integration rule and B-spline-based non-integer delay estimation method are used to design digital IIR integrator. First, the Gauss-Legendre integration rule is described briefly. Then, this rule is applied to design digital integrator. The Gauss-Legendre rule is used to estimate the following definite integral numerically:

$$Q = \int_{-1}^{1} s(t)dt \quad (25)$$

The *M*-point Gauss-Legendre approximation formula for this definite integral is given by

$$Q_M = \sum_{k=1}^{M} u_{M,k}\, s(t_{M,k}) \quad (26)$$

where the abscissas $t_{M,k}$ and weights $u_{M,k}$ must satisfy the following constraints

$$\int_{-1}^{1} t^m dt = \sum_{k=1}^{M} u_{M,k}(t_{M,k})^m \qquad m = 0,1,2,\cdots,2M-1 \quad (27)$$

By solving the above nonlinear equations, the abscissas $t_{M,k}$ and weights $u_{M,k}$ can be easily obtained [9]. Now, two typical cases are listed below. For two-point Gauss-Legendre rule, i.e., *M*=2, the abscissas $t_{M,k}$ and weights $u_{M,k}$ are

$$u_{2,1} = u_{2,2} = 1 \quad (28a)$$
$$-t_{2,1} = t_{2,2} = \tfrac{1}{\sqrt{3}} \quad (28b)$$

For three-point Gauss-Legendre rule, i.e., *M*=3, the abscissas $t_{M,k}$ and weights $u_{M,k}$ are

$$u_{3,1} = u_{3,3} = \tfrac{5}{9} \quad u_{3,2} = \tfrac{8}{9} \quad (29a)$$
$$-t_{3,1} = t_{3,3} = \sqrt{\tfrac{3}{5}} \quad t_{3,2} = 0 \quad (29b)$$

So far, the Gauss-Legendre rule has been reviewed briefly. Now, let us use this rule to design digital integrator. When a signal $s(t)$ passes through the ideal integrator with integer delay I, its output $y(t)$ is given by

$$y(t) = \int_{-\infty}^{t-I} s(\tau)d\tau \quad (30)$$

Setting $t = n-1$ and $t = n$, we have

$$y(n-1) = \int_{-\infty}^{n-I-1} s(\tau)d\tau \quad (31a)$$
$$y(n) = \int_{-\infty}^{n-I} s(\tau)d\tau \quad (31b)$$

Using the following equality:

$$\int_{-\infty}^{n-I} s(\tau)d\tau = \int_{-\infty}^{n-I-1} s(\tau)d\tau + \int_{n-I-1}^{n-I} s(\tau)d\tau \quad (32)$$

we get

$$y(n) = y(n-1) + \int_{n-I-1}^{n-I} s(\tau)d\tau \quad (33)$$

Thus, the design problem reduces to how to evaluate the definite integral of the second term in Eq.(33). This problem can be solved by using various numerical integration rules in textbook [9]. Because the Gauss-Legendre integration rule is a more accurate method, we use it to design digital integrator in this paper. The integral interval in Eq.(25) is [-1,1] which is not consistent with the integral interval $[n-I-1, n-I]$ in the second term of Eq.(33). So, the following variable substitution is used to make translation:

$$\tau = \frac{x + (2n-2I-1)}{2} \quad (34)$$

After using this substitution, the second term in Eq.(33) is given by

$$\int_{n-I-1}^{n-I} s(\tau)d\tau = \tfrac{1}{2}\int_{-1}^{1} s\left(\tfrac{x+(2n-2I-1)}{2}\right)dx \quad (35)$$

If two-point Gauss-Legendre rule is used, Eq.(35) can be approximated by

$$\int_{n-I-1}^{n-I} s(\tau)d\tau \approx \frac{1}{2}\left[s\left(\frac{\frac{1}{\sqrt{3}}+(2n-2I-1)}{2}\right)+s\left(\frac{\frac{-1}{\sqrt{3}}+(2n-2I-1)}{2}\right)\right] \quad (36)$$
$$\approx \frac{1}{2}\left[s(n-I-0.2113)+s(n-I-0.7887)\right]$$

Using Eq.(15), the Eq.(36) can be further reduced to

$$\int_{n-I-1}^{n-I} s(\tau)d\tau \approx \frac{1}{2}\sum_{m=0}^{N} g_1(m)s(n-m) \tag{37}$$

where

$$g_1(m) = h(m,0.2113) + h(m,0.7887) \tag{38}$$

Substituting Eq.(37) into Eq.(33), we have

$$y(n) = y(n-1) + \frac{1}{2}\sum_{m=0}^{N} g_1(m)s(n-m) \tag{39}$$

Taking the z transform at both sides, we obtain

$$H_1(z) = \frac{Y(z)}{S(z)} = \frac{1}{2}\frac{\sum_{m=0}^{N} g_1(m)z^{-m}}{1-z^{-1}} \tag{40}$$

The above $H_1(z)$ is the designed two-point Gauss-Legendre integrator using B-spline interpolation method.

Next, if three-point Gauss-Legendre rule is used, Eq.(35) can be approximated by

$$\begin{aligned}
&\int_{n-I-1}^{n-I} s(\tau)d\tau \\
&\approx \frac{1}{18}\left[\begin{array}{l} 5s\left(\dfrac{\sqrt{\frac{3}{5}}+(2n-2I-1)}{2}\right)+8s\left(\dfrac{2n-2I-1}{2}\right) \\ +5s\left(\dfrac{-\sqrt{\frac{3}{5}}+(2n-2I-1)}{2}\right)\end{array}\right] \\
&\approx \frac{1}{18}\left[\begin{array}{l} 5s(n-I-0.1127)+8s(n-I-0.5) \\ +5s(n-I-0.8873)\end{array}\right]
\end{aligned} \tag{41}$$

Using Eq.(15), the Eq.(41) can be further reduced to

$$\int_{n-I-1}^{n-I} s(\tau)d\tau \approx \frac{1}{18}\sum_{m=0}^{N} g_2(m)s(n-m) \tag{42}$$

where

$$\begin{aligned} g_2(m) = 5h(m,0.1127) + 8h(m,0.5) \\ + 5h(m,0.8873) \end{aligned} \tag{43}$$

Substituting Eq.(42) into Eq.(33), we have

$$y(n) = y(n-1) + \frac{1}{18}\sum_{m=0}^{N} g_2(m)s(n-m) \tag{44}$$

Taking the z transform at both sides, we obtain

$$H_2(z) = \frac{Y(z)}{S(z)} = \frac{1}{18}\frac{\sum_{m=0}^{N} g_2(m)z^{-m}}{1-z^{-1}} \tag{45}$$

The above $H_2(z)$ is the designed three-point Gauss-Legendre integrator using B-spline interpolation method.

5. DESIGN EXAMPLES AND COMPARISON

In this section, we will study the design error of the proposed B-spline-based integrator and compare it with conventional integrators. To evaluate the performance, the integral squares error of frequency response is defined by

$$E_k = \sqrt{\int_0^{\lambda\pi} \left|H_k(e^{j\omega}) - D(\omega)\right|^2 d\omega} \tag{46}$$

Obviously, the smaller the error E_k is, the better the performance of the design method is.

Example 1: In this example, we first study the relation between design error E_k and degree p for B-spline function $\beta^p(t)$ in Eq.(6). The parameters are chosen as *N*=20, *I*=10, and $\lambda = 0.95$. Fig.3(a)(b) shows the error curve E_k of the proposed B-spline integrator $H_k(z)$ for $p \in [1,\ 20]$. From these results, it is clear that both errors reach the minimum value when degree $p = 18$ is chosen. The minimum value of E_1 is 0.0296 and the minimum value of E_2 is 0.0276. So, the integrator $H_2(z)$ is slightly better than $H_1(z)$ in this design. Moreover, Fig.4 depicts the magnitude responses (solid line) of the integrator $H_1(z)$ for B-spline with $p = 18$. The dashed line is the ideal magnitude response $\frac{1}{\omega}$. Obviously, the specification is fitted well.

Example 2: In this example, we compare B-spline integrator with the Ngo integrator in Eq.(2) under the same implementation complexity. The design parameters are chosen as *N*=3, *I*=1 and $\lambda = 0.95$. Fig.5(a) shows the error curve E_1 of the integrator $H_1(z)$. From this result, it is clear that the error E_1 reach the minimum value 0.1973 when $p = 40$ is chosen. Fig.5(b) shows the frequency response error $20\log_{10}(|D(\omega) - F_1(e^{j\omega})|)$. The dashed line is the error $20\log_{10}(|D(\omega) - H_1(e^{j\omega})|)$ for B-spline integrator with $p = 40$. Obviously, $H_1(z)$ has smaller error than Ngo integrator in the frequency band $[0.23\pi, \pi]$. However, Ngo integrator is better than B-spline integrator in the low frequency band $[0, 0.23\pi]$

Example 3: In this example, we compare B-spline integrator with the Tseng integrator in Eq.(3) under the same implementation complexity. The design parameters are chosen as *N*=4, *I*=1 and $\lambda = 0.95$. Fig.6(a) shows the error curve E_1 of the integrator $H_1(z)$. From this result, it is clear that the error E_1 reach the minimum value 0.1568 when $p = 16$ is chosen. Fig.6(b) shows the frequency response error $20\log_{10}(|D(\omega) - F_2(e^{j\omega})|)$. The dashed line is the error $20\log_{10}(|D(\omega) - H_1(e^{j\omega})|)$ for B-spline integrator with $p = 16$. Obviously, $H_1(z)$ has smaller error than Tseng integrator in the frequency band $[0.4\pi, \pi]$. However, Tseng integrator is better than B-spline integrator in the low frequency band $[0, 0.23\pi]$

6. CONCLUSIONS

In this paper, the design of digital IIR integrator using B-spline interpolation method and Gauss-Legendre integration rule has been presented. The numerical comparisons with conventional digital integrators are also made. However, only digital integrator design is studied here. Thus, it is interesting to extend the B-spline interpolation method to design various digital filters in the future.

REFERENCES

[1] B. Kumar, D. Roy Choudhury and A. Kumar, "On the design of linear phase FIR integrators for midband frequencies," *IEEE Trans. on Signal Processing*, vol.44, pp.345-353, Oct. 1996.

[2] B. Kumar and A. Kumar, "FIR linear-phase approximations of frequency response $1/(j\omega)$ for maximal flatness at an arbitrary frequency ω_0, $(0 < \omega_0 < \pi)$," *IEEE Trans. on Signal Processing*, vol.47, pp.1772-1775, June 1999.

[3] N.Q. Ngo, "A new approach for the design of wideband digital integrator and differentiator," *IEEE Trans. on Circuits and Systems-II*, vol.53, pp.936-940, Sept. 2006.

[4] C.C. Tseng and S.L. Lee, "Digital IIR integrator design using Richardson extrapolation and fractional delay," *IEEE Trans. on Circuits and Systems-I*, vol.55, pp.2300-2309, Sept. 2008.

[5] T.M. Lehmann, C. Gonner and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Trans. on Medical Imaging*, vol.18, pp.1049-1075, Nov. 1999.

[6] S. Samadi, M.O. Ahmad and M.N.S. Swamy, "Characterization of B-spline digital filters," *IEEE Trans. on Circuits and Systems-I*, vol.51, pp.808-816, Apr. 2004.

[7] D. Petrinovic, "Causal cubic splines: formulations, interpolation properties and implementations," *IEEE Trans. on Signal Processing*, vol.56, pp.5442-5453, Nov. 2008.

[8] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp.22-38, Nov. 1999.

[9] J.H. Mathews and K.D. Fink, *Numerical Methods Using MATLAB*, Third Edition, Prentice-Hall, 1999.

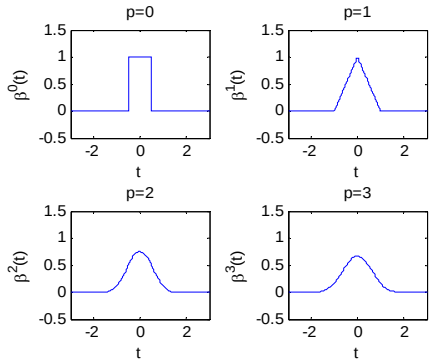

Fig.1 The normalized B-splines $\beta^p(t)$ for p=0,1,2,3.

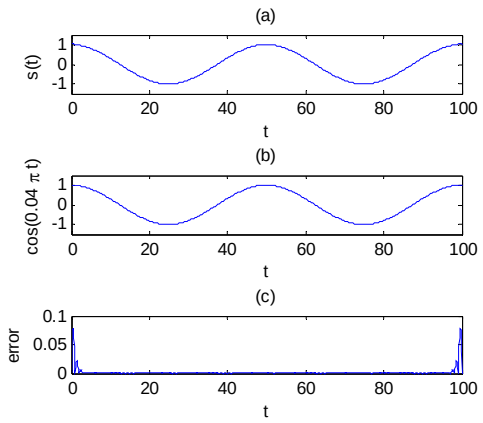

Fig.2 The B-spline interpolation. (a) The interpolated function $s(t)$. (b) Sinusoidal function $\cos(0.04\pi t)$. (c) The absolute errors $|s(t) - \cos(0.04\pi t)|$.

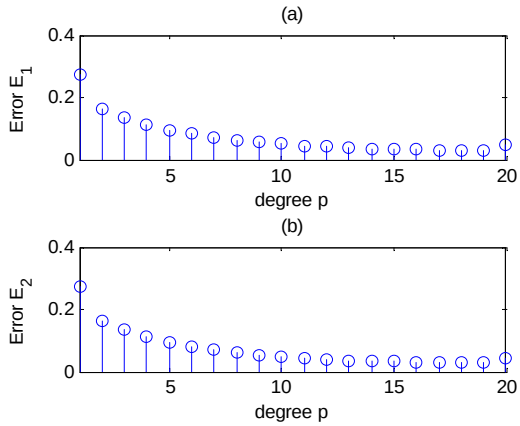

Fig.3 The error curve of proposed B-spline design method for degree $p \in [1,\ 20]$. (a) E_1 of integrator $H_1(z)$. (b) E_2 of integrator $H_2(z)$.

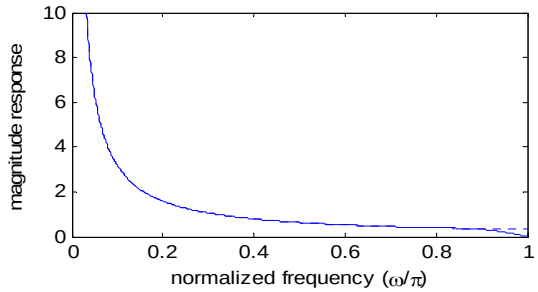

Fig.4 The magnitude response of integrator $H_1(z)$ designed by B-spline interpolation method with p=18. The dashed line is the ideal magnitude response.

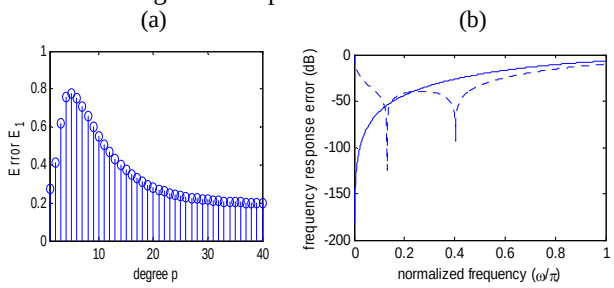

Fig.5 (a) Error curve E_1. (b) Error $20\log_{10}(|D(\omega) - F_1(e^{j\omega})|)$. The dashed line is the error $20\log_{10}(|D(\omega) - H_1(e^{j\omega})|)$.

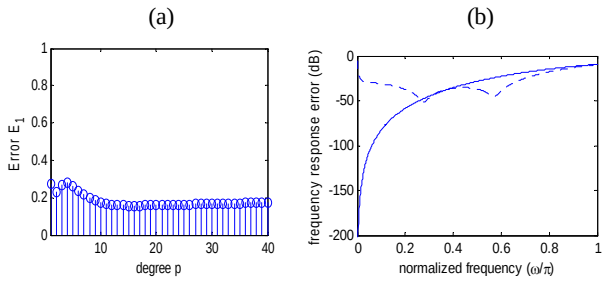

Fig.6 (a) Error curve E_1. (b) Error $20\log_{10}(|D(\omega) - F_2(e^{j\omega})|)$. The dashed line is the error $20\log_{10}(|D(\omega) - H_1(e^{j\omega})|)$.