Data Smoothing and Interpolation Using Eighth Order Algebraic Splines

Dan Simon *
Cleveland State University
Department of Electrical and Computer Engineering
1960 East 24th Street
Cleveland, OH 44115

June 13, 2003

Abstract

A new type of algebraic spline is used to derive a filter for smoothing or interpolating discrete data points. The spline is dependent on control parameters that specify the relative importance of data fitting and the derivatives of the spline. A general spline of arbitrary order is first formulated using matrix equations. We then focus on eighth order splines because of the continuity of their first three derivatives (desirable for motor and robotics applications). The spline's matrix equations are rewritten to give a recursive filter that can be implemented in real time for lengthy data sequences. The filter is low pass with a bandwidth that is dependent on the spline's control parameters. Numerical results, including a simple image processing application, show the tradeoffs that can be achieved using the algebraic splines.

Key Words – Splines, algebraic splines, optimization, recursive filters, data smoothing, interpolation, image processing.

*Email address d.j.simon@csuohio.edu

1 Introduction

Algebraic splines are useful tools for smoothing noisy data or interpolating between data points [1]. Splines have many practical applications, including image processing and robot path planning [4, 9]. An Mth order algebraic spline consists of piecewise continuous Mth order algebraic polynomials that have continuous derivatives up to order $(M-2)$ or less (depending on the details of the formulation). Higher order splines result in the continuity of higher order derivatives, which is desirable for many applications (such as robot path planning), but this may result in large oscillations in the resultant spline.

Cubic splines are popular choices for algebraic splines because they offer a good balance between simplicity and smoothness [3]. The problem solved by algebraic cubic splines can be stated as follows. Given the data set $f_i = f(t_i), i \in [0, N]$, with $t_0 < t_1 < \cdots < t_N$, find a piecewise continuous cubic algebraic function $y(t)$ that satisfies the following criteria.

- The function $y(t)$ has the form

$$\begin{aligned} y(t) &= y_i(t),\ t \in [t_{i-1}, t_i],\ (i = 1, \ldots, N) \\ y_i(t) &= a_i t^3 + b_i t^2 + c_i t + d_i \end{aligned} \tag{1}$$

  The spline $y(t)$ consists of N piecewise continuous spline segements $y_i(t)$ $(i = 1, \ldots, N)$. The parameters a_i, b_i, c_i, and d_i are the coefficients of the ith spline segment.

- The second derivatives at the endpoints are constrained to be zero.

$$y''(t_0) = y''(t_N) = 0 \tag{2}$$

- $y(t)$, $y'(t)$ and $y''(t)$ are continuous functions between t_0 and t_N.

- Among all piecewise continuous cubic algebraic polynomials that satisfy the above criteria, find the one that minimizes the function

$$J = \sum_{i=0}^{N} p_i (y_i - f_i)^2 + \int_{t_0}^{t_N} [y''(t)]^2 dt \quad (3)$$

where $y_i \equiv y(t_i)$, and p_i allows for individual control of the relative importance of data fitting at each knot.

In practice the control parameters p_i are often constant for all data points, and the data spacing is often constant so that $t_i - t_{i-1} = h$ $(i = 1, \ldots, N)$. In fact it can be shown that the minimization of

$$\sum_{i=0}^{N} p_i (y_i - f_i)^2 + \int_{t_0}^{t_N} [y^{(m)}(t)]^2 dt \quad (4)$$

over *all* functions $y(t)$ with continuous derivatives up to at least the mth order is solved by an algebraic spline of order $2m$ [6, 8, 2, pp. 235ff]. Splines with larger orders or nonalgebraic bases cannot reduce the cost function below that achieved by the optimal spline of order $2m$.

As an alternative to the standard algebraic spline formulation, we propose the use of eighth order splines. Our eighth order splines are formulated to have continuous derivatives up the third order and can be optimized with respect to a user-specified combination of their first three derivatives. Continuous third derivatives are desirable in applications such as robot path planning and motion trajectories for motors.

Section 2 derives the new type of algebraic spline. The spline formulation depends on the control parameters p_i that specify the importance of data fitting at each knot, and α_i parameters that specify the importance of the spline's derivatives. The algebraic spline problem can be posed as a set of matrix equations. Section 3 shows that if all of the $p_i = p =$ constant, the spline can be formulated as a recursive filter, a format that is well suited for large data sets. The bandwidth of the filter depends on the spline control parameter p and the α_i parameters. This recursive formulation is identical to the matrix formulation except at the endpoints of the data set. Section 4 presents some numerical results, including a simple image processing example. Section 5 offers some concluding remarks.

2 General Algebraic Splines

This section introduces a new way of formulating algebraic splines as solutions to a particular optimization problem related to that posed in the previous section. For now we will assume that the knots are normalized and are equally spaced so that $t_i = i$ $(i = 0, \ldots, N)$. We will time-scale the resulting spline later.

Definition: Let M be a positive even integer. An Mth order algebraic spline $y(t)$ is a function that satisfies the following criteria:

- The function $y(t)$ has the form

$$\begin{aligned} y(t) &= y_i(t),\ t \in [t_{i-1}, t_i],\ (i = 1, \ldots, N) \\ y_i(t) &= x_i(t - t_{i-1}) \\ x_i(t) &= \sum_{k=0}^{M-1} a_{ik} t^k,\ t \in [0, T] \end{aligned} \tag{5}$$

  where T is the normalized length of each spline segment. The spline $y(t)$ consists of N piece-wise continuous spline segements $y_i(t)$ $(i = 1, \ldots, N)$. The parameters a_{ik} $(k = 0, \ldots, M-1)$ are the coefficients of the ith spline segment.

- $y(t)$, $y'(t)$, $\cdots$, $y^{(R)}(t)$ are continuous functions between t_0 and t_N, where $R = M/2 - 1$.

- Among all piecewise continuous polynomials that satisfy the above criteria, find the one that minimizes the function

$$J = \sum_{i=0}^{N} p_i (y_i - f_i)^2 + \int_{t_0}^{t_N} \left\{ \alpha_1 [y'(t)]^2 + \cdots + \alpha_R [y^{(R)}(t)]^2 \right\} dt \tag{6}$$

where $y_i \equiv y(t_i)$ and p_i allows for individual control of the importance of data fitting at each knot. The parameters α_j allow the user to control the magnitude of each of the derivatives of the spline.

It can be seen each $x_i(t)$ is a normalized spline segment from $t = 0$ to $t = T$. These normalized segments are then shifted in time to form the $y_i(t)$ spline segments. The $y_i(t)$ segments are then pieced together to form the complete spline function $y(t)$. The function $y(t)$ has length NT, but it will be scaled in time later in this section to its desired length. In practice the control parameters $p_i = p$ are often constant for all data points, and the data spacing is often constant so that $t_i - t_{i-1} = h$ $(i = 1, \ldots, N)$.

We require continuity of $y(t)$ and its first R derivatives at the $(N-1)$ interior knots. We will use the notation $y_i \equiv y(t_i)$, $y_i' \equiv y'(t_i)$, $\cdots$, $y_i^{(R)} \equiv y^{(R)}(t_i)$. We therefore obtain the following M constraints from (5).

$$\begin{aligned} y_{i-1}^{(r)} &= x_i^{(r)}(0) \qquad (r = 0, \cdots, R) \\ y_i^{(r)} &= x_i^{(r)}(T) \qquad (r = 0, \cdots, R) \end{aligned} \tag{7}$$

These M equations can be put together into a single matrix equation as follows.

$$\begin{bmatrix} y_{i-1} \\ y_i \\ y'_{i-1} \\ y'_i \\ \vdots \\ y_{i-1}^{(R)} \\ y_i^{(R)} \end{bmatrix} = A \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ \vdots \\ a_{i,M-2} \\ a_{i,M-1} \end{bmatrix} \quad (i = 1, \ldots, N) \tag{8}$$

The normalized spline segment length T must be chosen so that the $M \times M$ matrix A is invertible. The matrix A defines the relationship between the ith spline segment coefficients, and the spline segment and derivative values at the endpoints of the ith spline segment.

Each normalized spline segment has M coefficients. Since there are N spline segments in the spline function, we have a total of MN independent variables. The optimization of (6) can be viewed as an optimization with respect to these MN spline coefficients, subject to the continuity constraints of $y(t)$, $y'(t)$, $\cdots$, $y^{(R)}(t)$, at the knots. But since the MN spline coeffients are functions of y_i, y_i', $\cdots$, $y_i^{(R)}$ $(i = 0, \ldots, N)$ as shown in (8), we can reformulate the problem as an *unconstrained* optimization problem with respect to these $(R+1)(N+1)$ parameters. The minimization of (6) can therefore be solved as

$$\frac{\partial J}{\partial y_i^{(r)}} = 0,\ (i = 0, \ldots, N),\ (r = 0, 1, \cdots, R) \tag{9}$$

This gives us $(R+1)(N+1)$ equations to solve for the $(R+1)(N+1)$ independent parameters. Now we note that the integral in the cost (6) can be divided into a sum of smaller integrals. Further note that if the desired knot spacing is h then, when the spline $y(t)$ is time-scaled from its normalized length NT to the desired length Nh, the rth derivative is scaled by a factor $(T/h)^r$. Also, integration of a function from 0 to h is larger than integration from 0 to T by a factor of h/T. We can therefore rewrite the cost (6) as follows.

$$\begin{aligned} J &= \sum_{i=0}^{N} p_i (y_i - f_i)^2 + \sum_{k=1}^{N} \int_0^T \left\{ \left(\frac{T}{h}\right) \alpha_1 [y_k'(t)]^2 + \cdots + \left(\frac{T}{h}\right)^{2R-1} \alpha_R [y_k^{(R)}(t)]^2 \right\} dt \qquad (10) \\ &= \sum_{i=0}^{N} p_i (y_i - f_i)^2 + \sum_{k=1}^{N} J_k \end{aligned}$$

where the definition of J_k is apparent in the above equation. Realizing that $y_k(t)$ is a function of $y_i^{(r)}$ only for $k = i$ and $k = i + 1$ we can combine (6), (9), and (10) to obtain the following minimization criteria.

$$\begin{aligned} 2(p_0 - f_0)\delta_r + \frac{\partial J_1}{\partial y_0^{(r)}} &= 0,\ (r = 0, 1, \cdots, R) \qquad (11) \\ 2(p_i - f_i)\delta_r + \frac{\partial (J_i + J_{i+1})}{\partial y_i^{(r)}} &= 0,\ (r = 0, 1, \cdots, R),\ (i = 1, \ldots, N-1) \\ 2(p_N - f_N)\delta_r + \frac{\partial J_N}{\partial y_N^{(r)}} &= 0,\ (r = 0, 1, \cdots, R) \end{aligned}$$

where $\delta_r = 1$ if $r = 0$ and $\delta_r = 0$ if $r \neq 0$. This gives us $(R+1)(N+1)$ equations to solve for the $(R+1)(N+1)$ unknowns $y_i,\, y_i',\, \cdots,\, y_i^{(R)}$ $(i = 0, \ldots, N)$. After we obtain $y_i,\, y_i',\, \cdots,\, y_i^{(R)}$, we can use (8) to obtain the optimal spline coefficients.

2.1 Eighth Order Splines

For eighth order splines we have $M = 8$ and $R = 3$. We therefore require continuity of $y(t)$ and its first three derivatives at the $(N-1)$ interior knots. If we use $T = 1$ as the normalized spline segment length, we obtain the following eight constraints from (5).

$$
\begin{aligned}
y_{i-1} &= x_i(0) = a_{i0} \\
y_i &= x_i(1) = \sum_{k=0}^{7} a_{ik} \\
y_{i-1}' &= x_i'(0) = a_{i1} \\
y_i' &= x_i'(1) = \sum_{k=1}^{7} k a_{ik} \\
y_{i-1}'' &= x_i''(0) = 2a_{i2} \\
y_i'' &= x_i''(1) = \sum_{k=2}^{7} k(k-1) a_{ik} \\
y_{i-1}''' &= x_i'''(0) = 6a_{i3} \\
y_i''' &= x_i'''(1) = \sum_{k=3}^{7} k(k-1)(k-2) a_{ik}
\end{aligned}
\tag{12}
$$

These equations can be put together into a single matrix equation as follows.

$$
\begin{bmatrix} y_{i-1} \\ y_i \\ y_{i-1}' \\ y_i' \\ y_{i-1}'' \\ y_i'' \\ y_{i-1}''' \\ y_i''' \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 6 & 24 & 60 & 120 & 210
\end{bmatrix}
\begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{i6} \\ a_{i7} \end{bmatrix}
\quad (i = 1, \ldots, N)
\tag{13}
$$

The 8×8 invertible matrix above defines the relationship between the ith spline segment coefficients, and the spline segment and derivative values at the endpoints of the ith spline segment.

The partial derivatives shown above can be computed analytically with the help of symbolic math software. When we use symbolic math software to compute (11) with $r = 1$, 2, and 3, we can combine the resulting $3(N+1)$ equations to obtain the matrix equation

$$DY^{(r)} + CY = 0 \tag{14}$$

where $Y^{(r)}$ is the $3(N+1) \times 1$ vector

$$\begin{aligned} Y^{(r)} &= \begin{bmatrix} Y_0^{(r)T} & \cdots & Y_N^{(r)T} \end{bmatrix}^T \\ &= \begin{bmatrix} y_0' & y_0'' & y_0''' & \cdots & y_N' & y_N'' & y_N''' \end{bmatrix}^T \end{aligned} \tag{15}$$

and Y is the $(N+1) \times 1$ vector

$$Y = \begin{bmatrix} y_0 & \cdots & y_N \end{bmatrix}^T \tag{16}$$

and the D and C matrices are given in the appendix. When we compute (11) with $r = 0$ we can combine the resulting $N+1$ equations to obtain the matrix equation

$$EY^{(r)} + LY + PY + PF = 0 \tag{17}$$

where the F is the $(N+1) \times 1$ vector

$$F = \begin{bmatrix} f_0 & \cdots & f_N \end{bmatrix}^T \tag{18}$$

with f_i being the original data points. The E, L, and P matrices are given in the appendix. Combining (14) and (17) we obtain

$$Y^{(r)} = D^{-1}C(L + P - ED^{-1}C)^{-1}PF \tag{19}$$

which gives us the optimal values of y_i, y_i', y_i'', and y_i''' ($i = 0, \ldots, N$) for the spline. Once these values are obtained, they can be used in (8) to obtain the optimal spline coefficients.

2.2 Tradeoffs Between Spline Orders

The procedure of this section can be used to obtain splines of any order M that is positive and even. In order to solve for the spline, a linear equation with a size proportional to M must be solved. The computational effort of this process is (in general) proportional to M^3. So an increase in M requires more computation but gives the user more flexibility in optimization and continuity of higher derivatives. An Mth order spline allows continuous derivatives up to order $M/2-1$ and also allows the user to optimize any derivative up to this order. For $M = 8$ the spline has continuous and optimizable derivatives up to the third order. This may be desirable for jerk sensitive applications such as motor control or robotics. But the benefit of increasing M beyond 8 is questionable. It may be difficult to find an application where derivatives of order greater than 3 are of any practical interest.

In this paper we have focused on 8th order splines because of their continuous and optimizable derivatives up to order 3. We have presented a general approach for spline formulation but have not implemented higher order splines because of their questionable utility. We have not explored lower order splines because of their lack of continuity and optimizability of higher order derivatives.

3 Recursive Formulation

Now consider the special case for eighth order splines that $p_i = p =$ constant, which is satisfied in many practical applications of splines. Then, from (14) and (17), we can write

$$D_2 Y_{k-1}^{(r)} + D_3 Y_k^{(r)} + D_4 Y_{k+1}^{(r)} + C_2 \begin{bmatrix} y_{k-1} \\ y_k \\ y_{k+1} \end{bmatrix} = 0, \ (k = 1, \ldots, N-1) \quad (20)$$

$$E_2 Y_{k-1}^{(r)} + E_3 Y_k^{(r)} + E_4 Y_{k+1}^{(r)} + \begin{bmatrix} 2L_1 & L_1 & 2L_1 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ y_k \\ y_{k+1} \end{bmatrix} +$$
$$2py_k - 2pf_k = 0, \ (k = 1, \ldots, N-1) \quad (21)$$

where the D_i, C_2, E_i, and L_1 matrices are given in the appendix. These equations can be rearranged to obtain the equation

$$U\psi_{k-1} + V\psi_k + W\psi_{k+1} = 2p\eta_k \tag{22}$$

where the 4×4 U, V, and W matrices are given in the appendix, and ψ_k and η_k are given as

$$\begin{aligned} \psi_k &= \begin{bmatrix} y_k & y'_k & y''_k & y'''_k \end{bmatrix}^T \\ \eta_k &= \begin{bmatrix} 0 & 0 & 0 & f_k \end{bmatrix}^T \end{aligned} \tag{23}$$

Taking the z-transform of (22) results in

$$\begin{aligned} \psi(z) &= 2p(Uz^{-1} + V + Wz)^{-1}\eta(z) \\ &= H(z)\eta(z) \end{aligned} \tag{24}$$

So the recursive algebraic spline can be viewed as a linear filter with the transfer function matrix given above. The transfer function from f_k to y_k is the element in the first row and fourth column of $H(z)$, denoted $H_{14}(z)$, and is a function of the parameters h, p, α_1, α_2, and α_3. The transfer function, given in the appendix, has a seventh order numerator and an eighth order denominator.

Figure 1 shows some transfer function magnitudes for $h = 1$ and various values of p for the minimum velocity spline $\{\alpha_1, \alpha_2, \alpha_3\} = \{1, 0, 0\}$, the minimum acceleration spline $\{\alpha_1, \alpha_2, \alpha_3\} = \{0, 1, 0\}$, and the minimum jerk spline where $\{\alpha_1, \alpha_2, \alpha_3\} = \{0, 0, 1\}$. As we expect, the algebraic spline is a low pass filter from the original knot values $\{f_k\}$ to the smoothed knot values $\{y_k\}$. A larger value of p means that the difference between the original knot values and the spline knot values becomes smaller, which means that the bandwidth of the filter increases. If we define the bandwidth ω_p of the filter by setting $|H(e^{j\omega_p})|^2 = |H(1)|^2/2$, then we can derive a plot of bandwidth as a function of p. This is shown in Figure 2 when $h = 1$. The relationship between bandwidth and p is nearly linear on a log-log scale, with the slope dependent on the values of α_1, α_2, and α_3. This plot can give a useful guide to the choice of the parameter p for data smoothing problems. For example, *a priori* knowledge of the spectral content of the original signal or the noise signal can be used in order to determine a suitable bandwidth for the smoothing filter, and thus determine an appropriate value for p.

4 Numerical Results

4.1 Smoothing and Interpolation

This section presents some simulation results comparing algebraic splines with various control parameters. As an example, consider the problem of generating a curve through the arbitrarily chosen and equally spaced knots $\{1, 5, 10, 8, 1, 6\}$ when $p = 1$ and $h = 1$. Figure 3 shows the minimum velocity spline $\{\alpha_1, \alpha_2, \alpha_3\} = \{1, 0, 0\}$, the minimum acceleration spline $\{\alpha_1, \alpha_2, \alpha_3\} = \{0, 1, 0\}$, and the minimum jerk spline $\{\alpha_1, \alpha_2, \alpha_3\} = \{0, 0, 1\}$. The sum part of the cost function is 26.2 for the minimum velocity spline, 31.8 for the minimum acceleration spline, and 29.6 for the minimum jerk spline. Figure 4 shows the first derivatives of the three splines for this example. The velocity integral is 15.0 for the minimum velocity spline, 16.1 for the minimum acceleration spline, and 31.0 for the minimum jerk spline. Figure 5 shows the second derivatives of the three splines for this example. The acceleration integral is 162 for the minimum velocity spline, 10.0 for the minimum acceleration spline, and 21.4 for the minimum jerk spline. Figure 6 shows the third derivatives of the three splines for this example. The jerk integral is 28900 for the minimum velocity spline, 14.9 for the minimum acceleration spline, and 5.47 for the minimum jerk spline.

Now consider the problem of generating an interpolating curve through the knots given in the previous example. In this case we have to use a larger value of p than before. Figure 7 shows the algebraic splines when $p = 500$ and $h = 1$. The first half of the cost function, $\sum_{i=0}^{N} p_i(y_i - f_i)^2$, is equal to 0.001 for the minimum velocity spline, 0.017 for the minimum acceleration spline, and 0.071 for the minimum jerk spline. The velocity integral is 123 for the minimum velocity spline, 136 for the minimum acceleration spline, and 148 for the minimum jerk spline. The acceleration integral is 1500 for the minimum velocity spline, 348 for the minimum acceleration spline, and 445 for the minimum jerk spline. The jerk integral is 234000 for the minimum velocity spline, 1380 for the minimum acceleration spline, and 581 for the minimum jerk spline.

To get a more general idea of the merits of algebraic splines, we generated 50 sets of random knots, with six knots in each set. The knot values were generated from a uniform distribution between 1 and 100. A low value of p was chosen to more clearly illustrate the differences in performance between the different splines. Table 1 shows the average performance of the splines for these sets of random knots when $p = 0.01$.

Note that the commonly used cubic spline is virtually identical in performance to the minimum acceleration spline. This is to be expected since cubic splines are minimum acceleration splines, as discussed in Section 1. However, cubic splines are not exactly identical to minimum acceleration eighth order splines because of a couple of differences. First, eighth order splines have continuous third derivatives while cubic splines do not have this constraint. Second, cubic splines constrain the second derivative at the endpoints to zero while eighth order splines do not have this constraint.

4.2 Image Processing

To further investigate the behavior of algebraic splines we applied them to a simple image processing example. We used a 256×256 pixel gray-scale version of the Lenna benchmark image. Each pixel's gray-scale was represented with an integer between 0 and 255. We compressed the image with various compression ratios using a simple decimation approach, and then restored the image using algebraic splines to interpolate between the data points of the compressed image. Table 2 shows the peak signal-to-noise ratio (PSNR) that was achieved with the splines for various compression ratios, where PSNR is defined as

$$\mathrm{PSNR} = 20 \ln \left(\frac{N}{\sqrt{\sum e_i^2}} \right) \tag{25}$$

where e_i is the error between the value of the ith original pixel and the ith restored pixel, and N is the total number of pixels (256^2 in this case). We also show in Table 2 the PSNR that was achieved with cubic splines. As discussed in the previous subsection, cubic splines perform almost identically to minimum acceleration splines. It appears from Table 2 that minimum velocity splines perform best for image processing applications. Note that the MPEG committee uses 0.5 dB PSNR as an informal threshold for improvement that is visible to the human eye [5]. Figure 8 shows close-ups of Lenna's right eye in the original image, and in three spline reconstructions after 50% compression ($p = 10$). It can be seen that the minimum acceleration spline (and hence the cubic spline) smudges the image noticably more than the minimum velocity spline, but less than the minimum jerk spline.

5 Conclusion

A general approach to the optimization of algebraic splines has been proposed. The spline can be optimized with respect to knot errors, velocity, acceleration, jerk, or any combination thereof. This offers a more general alternative to standard splines for data smoothing and interpolation. The splines can be formulated in a recursive manner to minimize computation time for long data sets. Initial results indicate that minimum velocity splines provide better image processing performance than minimum acceleration or minimum jerk splines. Matlab code that can be used to generate general algebraic splines can be found on the internet at http://academic.csuohio.edu/simond/splines.

For future work, orthogonal functions such as trigonometric polynomials [7], Legendre polynomials, and Laguerre polynomials may also provide a good basis for splines in certain problem-specific circumstances. The minimax property of Chebyshev polynomials has already been used for optimal control [10]. This might make Chebyshev polynomials an especially attractive alternative for spline functions.

Acknowledgements

The comments of the anonymous reviewers were instrumental in correcting and strengthening this paper from its original version.

Appendix

This appendix specifies some of the matrices that are used in this paper. The $3(N+1)\times 3(N+1)$ matrix D that is used in (14) is a block tridiagonal matrix. The $(N+1)$ blocks along the diagonal are computed as

$$\tilde{D}_3, D_3, \cdots, D_3, \tilde{\tilde{D}}_3 \tag{26}$$

where the above matrices are given as

$$\begin{aligned}
\tilde{D}_3 &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{1200}{77} & \frac{379}{231} & \frac{16}{231} \\ \frac{379}{231} & \frac{100}{231} & \frac{2}{99} \\ \frac{16}{231} & \frac{2}{99} & \frac{4}{3465} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{600}{1001} & \frac{123}{2002} & \frac{25}{9009} \\ \frac{123}{2002} & \frac{73}{9009} & \frac{37}{90090} \\ \frac{25}{9009} & \frac{37}{90090} & \frac{1}{45045} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} \frac{4160}{7} & \frac{820}{7} & \frac{88}{21} \\ \frac{820}{7} & \frac{200}{7} & \frac{23}{21} \\ \frac{88}{21} & \frac{23}{21} & \frac{8}{63} \end{bmatrix} \\
D_3 &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{1200}{1001} & 0 & \frac{50}{9009} \\ 0 & \frac{146}{9009} & 0 \\ \frac{50}{9009} & 0 & \frac{2}{45045} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{2400}{77} & 0 & \frac{32}{231} \\ 0 & \frac{200}{231} & 0 \\ \frac{32}{231} & 0 & \frac{8}{3465} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} \frac{8320}{7} & 0 & \frac{176}{21} \\ 0 & \frac{400}{7} & 0 \\ \frac{176}{21} & 0 & \frac{16}{63} \end{bmatrix} \\
\tilde{\tilde{D}}_3 &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{600}{1001} & -\frac{123}{2002} & \frac{25}{9009} \\ -\frac{123}{2002} & \frac{73}{9009} & -\frac{37}{90090} \\ \frac{25}{9009} & -\frac{37}{90090} & \frac{1}{45045} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{1200}{77} & -\frac{379}{231} & \frac{16}{231} \\ -\frac{379}{231} & \frac{100}{231} & -\frac{2}{99} \\ \frac{16}{231} & -\frac{2}{99} & \frac{4}{3465} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} \frac{4160}{7} & -\frac{820}{7} & \frac{88}{21} \\ -\frac{820}{7} & \frac{200}{7} & -\frac{23}{21} \\ \frac{88}{21} & -\frac{23}{21} & \frac{8}{63} \end{bmatrix}
\end{aligned} \tag{27}$$

The $(N-1)$ blocks along the superdiagonal are computed as

$$D_4 = \frac{\alpha_1}{h}\begin{bmatrix} \frac{97}{3003} & \frac{47}{6006} & -\frac{5}{6006} \\ -\frac{47}{6006} & \frac{7}{2574} & -\frac{73}{360360} \\ -\frac{5}{6006} & \frac{73}{360360} & -\frac{1}{72072} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{760}{77} & -\frac{181}{231} & \frac{5}{231} \\ \frac{181}{231} & -\frac{1}{231} & -\frac{5}{1386} \\ \frac{5}{231} & \frac{5}{1386} & -\frac{1}{2310} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} \frac{3680}{7} & -\frac{580}{7} & \frac{52}{21} \\ \frac{580}{7} & -\frac{80}{7} & \frac{5}{21} \\ \frac{52}{21} & -\frac{5}{21} & -\frac{1}{63} \end{bmatrix} \tag{28}$$

The $(N-1)$ blocks along the subdiagonal are computed as

$$D_2 = \frac{\alpha_1}{h}\begin{bmatrix} \frac{97}{3003} & -\frac{47}{6006} & -\frac{5}{6006} \\ \frac{47}{6006} & \frac{7}{2574} & \frac{73}{360360} \\ -\frac{5}{6006} & -\frac{73}{360360} & -\frac{1}{72072} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{760}{77} & -\frac{181}{231} & \frac{5}{231} \\ \frac{181}{231} & -\frac{1}{231} & -\frac{5}{1386} \\ \frac{5}{231} & \frac{5}{1386} & -\frac{1}{2310} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} \frac{3680}{7} & \frac{580}{7} & \frac{52}{21} \\ -\frac{580}{7} & -\frac{80}{7} & -\frac{5}{21} \\ \frac{52}{21} & \frac{5}{21} & -\frac{1}{63} \end{bmatrix} \tag{29}$$

The $3(N+1)\times(N+1)$ matrix C that is used in (14) is a type of block diagonal matrix.

$$C = \text{diag}(C_1, \cdots, C_{N+1}) \tag{30}$$

Each of the C_i matrices is a 3×3 matrix. The upper left element of C_1 is in the first row and first column. The upper left element of C_i is in the $(3i-2)$nd row and $(i-1)$st column for $(i = 2, \cdots, N)$. The upper left element of C_{N+1} is in the $(3N+1)$st row and $(N-1)$st column. The C_i matrices are computed as

$$
\begin{aligned}
C_1 &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{271}{429} & -\frac{271}{429} & 0 \\ \frac{23}{429} & -\frac{23}{429} & 0 \\ \frac{5}{2574} & -\frac{5}{2574} & 0 \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{280}{11} & -\frac{280}{11} & 0 \\ \frac{80}{33} & -\frac{80}{33} & 0 \\ \frac{1}{11} & -\frac{1}{11} & 0 \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 1120 & -1120 & 0 \\ 200 & -200 & 0 \\ \frac{20}{3} & -\frac{20}{3} & 0 \end{bmatrix} \qquad (31) \\
C_i &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{271}{429} & 0 & -\frac{271}{429} \\ -\frac{23}{429} & \frac{46}{429} & -\frac{23}{429} \\ \frac{5}{2574} & 0 & -\frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{280}{11} & 0 & -\frac{280}{11} \\ -\frac{80}{33} & \frac{160}{33} & -\frac{80}{33} \\ \frac{1}{11} & 0 & -\frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 1120 & 0 & -1120 \\ -200 & 400 & -200 \\ \frac{20}{3} & 0 & -\frac{20}{3} \end{bmatrix} \\
& \quad (i = 2, \ldots, N) \\
C_{N+1} &= \frac{\alpha_1}{h}\begin{bmatrix} 0 & \frac{271}{429} & -\frac{271}{429} \\ 0 & -\frac{23}{429} & \frac{23}{429} \\ 0 & \frac{5}{2574} & -\frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} 0 & \frac{280}{11} & -\frac{280}{11} \\ 0 & -\frac{80}{33} & \frac{80}{33} \\ 0 & \frac{1}{11} & -\frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 0 & 1120 & -1120 \\ 0 & -200 & 200 \\ 0 & \frac{20}{3} & -\frac{20}{3} \end{bmatrix}
\end{aligned}
$$

The $(N+1) \times 3(N+1)$ matrix E that is used in (17) is a block tridiagonal matrix. The $(N+1)$ blocks along the diagonal are computed as

$$\tilde{E}_3, E_3, \cdots, E_3, \tilde{\tilde{E}}_3 \qquad (32)$$

where the above matrices are given as

$$
\begin{aligned}
\tilde{E}_3 &= \frac{\alpha_1}{h}\begin{bmatrix} \frac{271}{429} & \frac{23}{429} & \frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{280}{11} & \frac{80}{33} & \frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 1120 & 200 & \frac{20}{3} \end{bmatrix} \qquad (33) \\
E_3 &= \frac{\alpha_1}{h}\begin{bmatrix} 0 & \frac{46}{429} & 0 \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} 0 & \frac{160}{33} & 0 \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 0 & 400 & 0 \end{bmatrix} \\
\tilde{\tilde{E}}_3 &= \frac{\alpha_1}{h}\begin{bmatrix} -\frac{271}{429} & \frac{23}{429} & -\frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} -\frac{280}{11} & \frac{80}{33} & -\frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} -1120 & 200 & -\frac{20}{3} \end{bmatrix}
\end{aligned}
$$

The $(N-1)$ blocks along the superdiagonal are computed as

$$E_4 = \frac{\alpha_1}{h}\begin{bmatrix} \frac{271}{429} & -\frac{23}{429} & \frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{280}{11} & -\frac{80}{33} & \frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} 1120 & -200 & \frac{20}{3} \end{bmatrix} \qquad (34)$$

The $(N-1)$ blocks along the subdiagonal are computed as

$$E_2 = \frac{\alpha_1}{h}\begin{bmatrix} -\frac{271}{429} & -\frac{23}{429} & -\frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} -\frac{280}{11} & -\frac{80}{33} & -\frac{1}{11} \end{bmatrix} + \frac{\alpha_3}{h^5}\begin{bmatrix} -1120 & -200 & -\frac{20}{3} \end{bmatrix} \qquad (35)$$

The $(N+1)\times(N+1)$ matrix L that is used in (17) is a tridiagonal matrix. The $(N+1)$ elements along the diagonal are computed as

$$L_1, 2L_1, \cdots, 2L_1, L_1 \tag{36}$$

where L_1 is given as

$$L_1 = \frac{\alpha_1}{h}\frac{1400}{429} + \frac{\alpha_2}{h^3}\frac{560}{11} + \frac{\alpha_3}{h^5}2240 \tag{37}$$

The $(N-1)$ elements along the superdiagonal and the $(N-1)$ elements along the subdiagonal are all equal to $-L_1$.

The $(N+1)\times(N+1)$ matrix P that is used in (17) is a diagonal matrix. The $(N+1)$ elements along the diagonal are computed as

$$2p_0, \cdots, 2p_N \tag{38}$$

where p_i is the weight of the spline fit at knot i relative to the smoothness criteria as shown in (6).

The U, V, and W matrices used in (22) are given as follows

$$\begin{aligned} U \quad = \quad & \frac{\alpha_1}{h}\begin{bmatrix} \frac{271}{429} & \frac{97}{3003} & -\frac{47}{6006} & -\frac{5}{6006} \\ -\frac{23}{429} & \frac{47}{6006} & \frac{7}{2574} & \frac{73}{360360} \\ \frac{5}{2574} & -\frac{5}{6006} & -\frac{73}{360360} & -\frac{1}{72072} \\ -\frac{1400}{429} & -\frac{271}{429} & -\frac{23}{429} & -\frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} \frac{280}{11} & \frac{760}{77} & \frac{181}{231} & \frac{5}{231} \\ -\frac{80}{33} & -\frac{181}{231} & -\frac{1}{231} & \frac{5}{1386} \\ \frac{1}{11} & \frac{5}{231} & -\frac{5}{1386} & -\frac{1}{2310} \\ -\frac{560}{11} & -\frac{280}{11} & -\frac{80}{33} & -\frac{1}{11} \end{bmatrix} + \\ & \frac{\alpha_3}{h^5}\begin{bmatrix} 1120 & \frac{3680}{7} & \frac{580}{7} & \frac{52}{21} \\ -200 & -\frac{580}{7} & -\frac{80}{7} & -\frac{5}{21} \\ \frac{20}{3} & \frac{52}{21} & \frac{5}{21} & -\frac{1}{63} \\ -2240 & -1120 & -200 & -\frac{20}{3} \end{bmatrix} \end{aligned} \tag{39}$$

$$V \quad = \quad \frac{\alpha_1}{h}\begin{bmatrix} 0 & \frac{1200}{1001} & 0 & \frac{50}{9009} \\ \frac{46}{429} & 0 & \frac{146}{9009} & 0 \\ 0 & \frac{50}{9009} & 0 & \frac{2}{45045} \\ \frac{2800}{429}+2\,p & 0 & \frac{46}{429} & 0 \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} 0 & \frac{2400}{77} & 0 & \frac{32}{231} \\ \frac{160}{33} & 0 & \frac{200}{231} & 0 \\ 0 & \frac{32}{231} & 0 & \frac{8}{3465} \\ \frac{1120}{11}+2\,p & 0 & \frac{160}{33} & 0 \end{bmatrix} +$$

$$\frac{\alpha_3}{h^5}\begin{bmatrix} 0 & \frac{8320}{7} & 0 & \frac{176}{21} \\ 400 & 0 & \frac{400}{7} & 0 \\ 0 & \frac{176}{21} & 0 & \frac{16}{63} \\ 4480+2\,p & 0 & 400 & 0 \end{bmatrix}$$

$$W = \frac{\alpha_1}{h}\begin{bmatrix} -\frac{271}{429} & \frac{97}{3003} & \frac{47}{6006} & -\frac{5}{6006} \\ -\frac{23}{429} & -\frac{47}{6006} & \frac{7}{2574} & -\frac{73}{360360} \\ -\frac{5}{2574} & -\frac{5}{6006} & \frac{73}{360360} & -\frac{1}{72072} \\ -\frac{1400}{429} & \frac{271}{429} & -\frac{23}{429} & \frac{5}{2574} \end{bmatrix} + \frac{\alpha_2}{h^3}\begin{bmatrix} -\frac{280}{11} & \frac{760}{77} & -\frac{181}{231} & \frac{5}{231} \\ -\frac{80}{33} & \frac{181}{231} & -\frac{1}{231} & -\frac{5}{1386} \\ -1/11 & \frac{5}{231} & \frac{5}{1386} & -\frac{1}{2310} \\ -\frac{560}{11} & \frac{280}{11} & -\frac{80}{33} & 1/11 \end{bmatrix} +$$

$$\frac{\alpha_3}{h^5}\begin{bmatrix} -1120 & \frac{3680}{7} & -\frac{580}{7} & \frac{52}{21} \\ -200 & \frac{580}{7} & -\frac{80}{7} & \frac{5}{21} \\ -\frac{20}{3} & \frac{52}{21} & -\frac{5}{21} & -\frac{1}{63} \\ -2240 & 1120 & -200 & \frac{20}{3} \end{bmatrix}$$

The element in the first row and fourth column of $H(z)$ in (24) is the transfer function from the original knot sequence $\{f_k\}$ to the spline values $\{y_k\}$ at the knot times. This transfer function can be written as

$$H_{14}(z) = \alpha_1 N_{1\alpha_1}(z)/D_{\alpha_1}(z) + \alpha_2 N_{1\alpha_2}(z)/D_{\alpha_2}(z) + \alpha_3 N_{1\alpha_3}(z)/D_{\alpha_3}(z) \tag{40}$$

where the numerator and denominator terms are given as

$$\begin{aligned} N_{1j}(z) &= n_{1j7}z^7 + n_{1j6}z^6 + n_{1j5}z^5 + n_{1j4}z^4 + n_{1j3}z^3 + n_{1j2}z^2 + n_{1j1}z + n_{1j0} \qquad (41) \\ D_j(z) &= d_{j8}z^8 + d_{j7}z^7 + d_{j6}z^6 + d_{j5}z^5 + d_{j4}z^4 + d_{j3}z^3 + d_{j2}z^2 + d_{j1}z + d_{j0},\ (j = \alpha_1, \alpha_2, \alpha_3) \end{aligned}$$

The numerator polynomial coefficients are given as

$$\begin{aligned} n_{1\alpha_1 0} &= 0 \qquad (42) \\ n_{1\alpha_1 1} = n_{1\alpha_1 7} &= 486ph \\ n_{1\alpha_1 2} = n_{1\alpha_1 6} &= -15840ph \\ n_{1\alpha_1 3} = n_{1\alpha_1 5} &= 1993926ph \\ n_{1\alpha_1 4} &= 9360960ph \end{aligned}$$

$$
\begin{aligned}
n_{1\alpha_2 0} &= 0 \\
n_{1\alpha_2 1} = n_{1\alpha_2 7} &= 17ph^3 \\
n_{1\alpha_2 2} = n_{1\alpha_2 6} &= -4624ph^3 \\
n_{1\alpha_2 3} = n_{1\alpha_2 5} &= 8263ph^3 \\
n_{1\alpha_2 4} &= 98528ph^3
\end{aligned} \tag{43}
$$

$$
\begin{aligned}
n_{1\alpha_3 0} = n_{1\alpha_3 6} = n_{1\alpha_3 7} &= 0 \\
n_{1\alpha_3 1} = n_{1\alpha_3 5} &= -ph^5 \\
n_{1\alpha_3 2} =_{1\alpha_3 4} &= -26ph^5 \\
n_{1\alpha_3 3} &= -66ph^5
\end{aligned} \tag{44}
$$

The demonimator polynomial coefficients are given as

$$
\begin{aligned}
d_{\alpha_1 0} = d_{\alpha_1 8} &= 7 \\
d_{\alpha_1 1} = d_{\alpha_1 7} &= -25200 + 486ph \\
d_{\alpha_1 2} = d_{\alpha_1 6} &= -15840ph + 2369332 \\
d_{\alpha_1 3} = d_{\alpha_1 5} &= -14593040 - 1993926ph \\
d_{\alpha_1 4} &= 24497802 + 9360960ph
\end{aligned} \tag{45}
$$

$$
\begin{aligned}
d_{\alpha_2 0} = d_{\alpha_2 8} &= 105 \\
d_{\alpha_2 1} = d_{\alpha_2 7} &= 17ph^3 - 28560 \\
d_{\alpha_2 2} = d_{\alpha_2 6} &= -4624ph^3 + 275100 \\
d_{\alpha_2 3} = d_{\alpha_2 5} &= 8263ph^3 - 845040 \\
d_{\alpha_2 4} &= 1196790 + 98528ph^3
\end{aligned} \tag{46}
$$

$$
\begin{aligned}
d_{\alpha_3 0} = d_{\alpha_3 6} &= 120 \\
d_{\alpha_3 1} = d_{\alpha_3 5} &= -720 - ph^5 \\
d_{\alpha_3 2} = d_{\alpha_3 4} &= 1800 - 26ph^5 \\
d_{\alpha_3 3} &= -2400 - 66ph^5 \\
d_{\alpha_3 7} = d_{\alpha_3 8} &= 0
\end{aligned} \tag{47}
$$

The element in the second row and fourth column of $H(z)$ in (24) is the transfer function from the original knot sequence $\{f_k\}$ to the first spline derivatives $\{y'_k\}$ at the knot times. This transfer function can be written as

$$
H_{24}(z) = \alpha_1 N_{2\alpha_1}(z)/D_{\alpha_1}(z) + \alpha_2 N_{2\alpha_2}(z)/D_{\alpha_2}(z) + \alpha_3 N_{2\alpha_3}(z)/D_{\alpha_3}(z) \tag{48}
$$

where the denominator polynomials are given above and the numerator terms are given as

$$
N_{2j}(z) = n_{2j7}z^7 + n_{2j6}z^6 + n_{2j5}z^5 + n_{2j4}z^4 + n_{2j3}z^3 + n_{2j2}z^2 + n_{2j1}z + n_{2j0}, \; (j = \alpha_1, \alpha_2, \alpha_3) \tag{49}
$$

The numerator polynomial coefficients are given as

$$
\begin{aligned}
n_{2\alpha_1 0} = n_{2\alpha_1 4} &= 0 \\
n_{2\alpha_1 1} = -n_{2\alpha_1 7} &= -7686ph \\
n_{2\alpha_1 2} = -n_{2\alpha_1 6} &= 863520ph \\
n_{2\alpha_1 3} = -n_{2\alpha_1 5} &= -4375182ph
\end{aligned} \tag{50}
$$

$$
\begin{aligned}
n_{2\alpha_2 0} = n_{2\alpha_2 4} &= 0 \\
n_{2\alpha_2 1} = -n_{2\alpha_2 7} &= -35ph^3 \\
n_{2\alpha_2 2} = -n_{2\alpha_2 6} &= 14000ph^3 \\
n_{2\alpha_2 3} = -n_{2\alpha_2 5} &= -80815ph^3
\end{aligned} \tag{51}
$$

$$n_{2\alpha_3 0} = n_{2\alpha_3 6} = n_{2\alpha_3 7} = n_{2\alpha_3 3} = 0 \tag{52}$$
$$n_{2\alpha_3 1} = -n_{2\alpha_3 5} = 5ph^5$$
$$n_{2\alpha_3 2} = -n_{2\alpha_3 4} = 50ph^5$$

The element in the third row and fourth column of $H(z)$ in (24) is the transfer function from the original knot sequence $\{f_k\}$ to the second spline derivatives $\{y''_k\}$ at the knot times. This transfer function can be written as

$$H_{34}(z) = \alpha_1 N_{3\alpha_1}(z)/D_{\alpha_1}(z) + \alpha_2 N_{3\alpha_2}(z)/D_{\alpha_2}(z) + \alpha_3 N_{3\alpha_3}(z)/D_{\alpha_3}(z) \tag{53}$$

where the denominator polynomials are given above and the numerator terms are given as

$$N_{3j}(z) = n_{3j7}z^7 + n_{3j6}z^6 + n_{3j5}z^5 + n_{3j4}z^4 + n_{3j3}z^3 + n_{3j2}z^2 + n_{3j1}z + n_{3j0},\ (j = \alpha_1, \alpha_2, \alpha_3) \tag{54}$$

The numerator polynomial coefficients are given as

$$n_{3\alpha_1 0} = 0 \tag{55}$$
$$n_{3\alpha_1 1} = n_{3\alpha_1 7} = 88200ph$$
$$n_{3\alpha_1 2} = n_{3\alpha_1 6} = -12479040ph$$
$$n_{3\alpha_1 3} = n_{3\alpha_1 5} = 66218040ph$$
$$n_{3\alpha_1 4} = -107654400ph$$

$$n_{3\alpha_2 0} = 0 \tag{56}$$
$$n_{3\alpha_2 1} = n_{3\alpha_2 7} = -210ph^3$$
$$n_{3\alpha_2 2} = n_{3\alpha_2 6} = -23520ph^3$$
$$n_{3\alpha_2 3} = n_{3\alpha_2 5} = 201810ph^3$$
$$n_{3\alpha_2 4} = -356160ph^3$$

$$
\begin{aligned}
n_{3\alpha_3 0} = n_{3\alpha_3 6} = n_{3\alpha_3 7} &= 0 \\
n_{3\alpha_3 1} = n_{3\alpha_3 5} &= -20ph^5 \\
n_{3\alpha_3 2} = n_{3\alpha_3 4} &= -40ph^5 \\
n_{3\alpha_3 3} &= 120ph^5
\end{aligned} \tag{57}
$$

The element in the fourth row and fourth column of $H(z)$ in (24) is the transfer function from the original knot sequence $\{f_k\}$ to the third spline derivatives $\{y_k'''\}$ at the knot times. This transfer function can be written as

$$
H_{44}(z) = \alpha_1 N_{4\alpha_1}(z)/D_{\alpha_1}(z) + \alpha_2 N_{4\alpha_2}(z)/D_{\alpha_2}(z) + \alpha_3 N_{4\alpha_3}(z)/D_{\alpha_3}(z) \tag{58}
$$

where the denominator polynomials are given above and the numerator terms are given as

$$
N_{4j}(z) = n_{4j7}z^7 + n_{4j6}z^6 + n_{4j5}z^5 + n_{4j4}z^4 + n_{4j3}z^3 + n_{4j2}z^2 + n_{4j1}z + n_{4j0},\ (j = \alpha_1, \alpha_2, \alpha_3) \tag{59}
$$

The numerator polynomial coefficients are given as

$$
\begin{aligned}
n_{4\alpha_1 0} = n_{4\alpha_1 4} &= 0 \\
n_{4\alpha_1 1} = -n_{4\alpha_1 7} &= -758520ph \\
n_{4\alpha_1 2} = -n_{4\alpha_1 6} &= 122663520ph \\
n_{4\alpha_1 3} = -n_{4\alpha_1 5} &= -243051480ph
\end{aligned} \tag{60}
$$

$$
\begin{aligned}
n_{4\alpha_2 0} = n_{4\alpha_2 4} &= 0 \\
n_{4\alpha_2 1} = -n_{4\alpha_2 7} &= 3570ph^3 \\
n_{4\alpha_2 2} = -n_{4\alpha_2 6} &= -67200ph^3 \\
n_{4\alpha_2 3} = -n_{4\alpha_2 5} &= 123690ph^3
\end{aligned} \tag{61}
$$

$$
n_{4\alpha_3 0} = n_{4\alpha_3 3} = n_{4\alpha_3 6} = n_{4\alpha_3 7} = 0 \tag{62}
$$

$$
\begin{aligned}
n_{4\alpha_3 1} = -n_{4\alpha_3 5} &= 60ph^5 \\
n_{4\alpha_3 2} = -n_{4\alpha_3 4} &= -120ph^5
\end{aligned}
$$

References

[1] K. Atkinson, An Introduction to Numerical Analysis (Wiley, New York, 1989).

[2] C. de Boor, A Practical Guide to Splines (Springer-Verlag, New York, 1978).

[3] G. Feng, Data smoothing by cubic spline filters, IEEE Transactions on Signal Processing 46(1) pp. 2790-2796, October 1998.

[4] C. Lin, P. Chang, and J. Luh, Formulation and optimization of cubic polynomial joint trajectories for industrial robots, IEEE Transactions on Automatic Control AC-28, pp. 1066-1073, 1983.

[5] A. Netravali and B. Haskell, Digital Pictures: Representation, Compression, and Standards (2nd ed.) (Plenum Press, New York, 1995).

[6] C. Reinsch, Smoothing by spline functions, Numerische Mathematik 10, pp. 177-183, 1967.

[7] I. Schoenberg, On trigonometric spline interpolation, Journal of Mathematics and Mechanics 13 pp. 795-825, 1964.

[8] I. Schoenberg, Spline functions and the problem of graduation, Proceedings of the National Academy of Sciences 52 pp. 947-950, 1964.

[9] D. Simon and C. Isik, A trigonometric trajectory generator for robotic arms, International Journal of Control 57 pp. 505-517, 1993.

[10] J. Vlassenbroeck and R. Van Dooren, A Chebyshev technique for solving nonlinear optimal control problems, IEEE Transactions on Automatic Control AC-33 pp. 333-340, 1988.

	Min. Vel. Spline	Min. Accel. Spline	Min. Jerk Spline	Cubic Spline
Velocity Integral	**0.381**	183	1160	183
Accel. Integral	1.48	**0.228**	473	**0.228**
Jerk Integral	203	0.166	**0.040**	0.166
Knot Error Sum	3720	3120	2270	3120

Table 1: Performance of algebraic splines when $h = 1$ and $p = 0.01$, averaged over 50 sets of knots with six knots in each set.

Compression	Min. Vel. Spline	Min. Accel. Spline	Min. Jerk Spline	Cubic Spline
1:2	28.8	27.6	26.6	27.6
1:4	24.3	23.6	22.9	23.6
1:6	22.2	21.7	21.0	21.7
1:8	20.9	20.3	19.7	20.3

Table 2: Image restoration quality (dB of PSNR) of algebraic splines ($p = 10$).

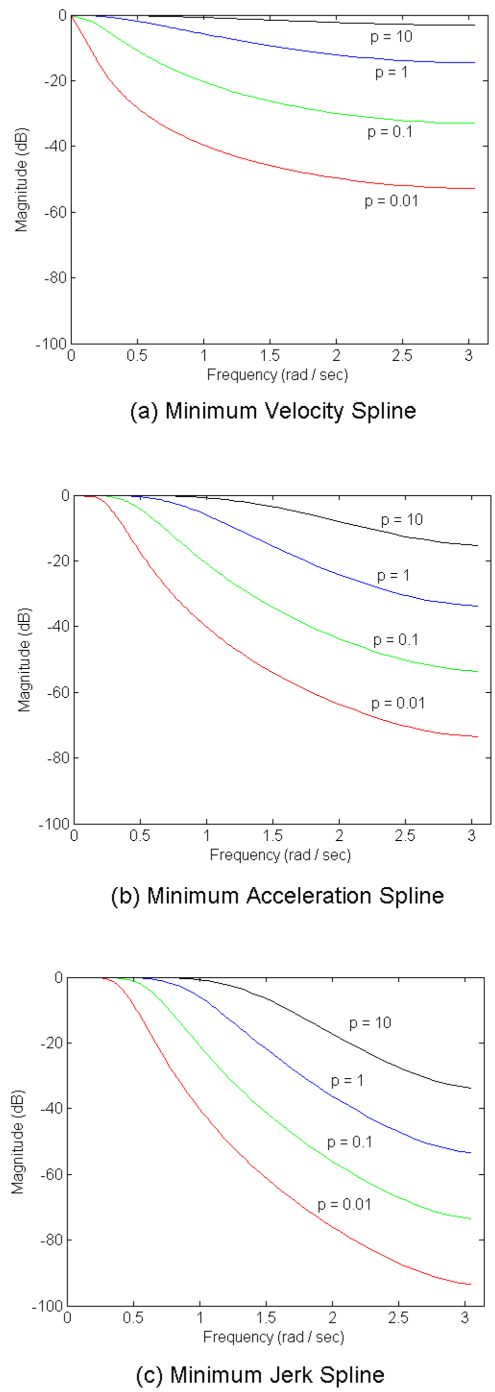

Figure 1: Transfer functions of the algebraic spline filters for various values of p when $h = 1$.

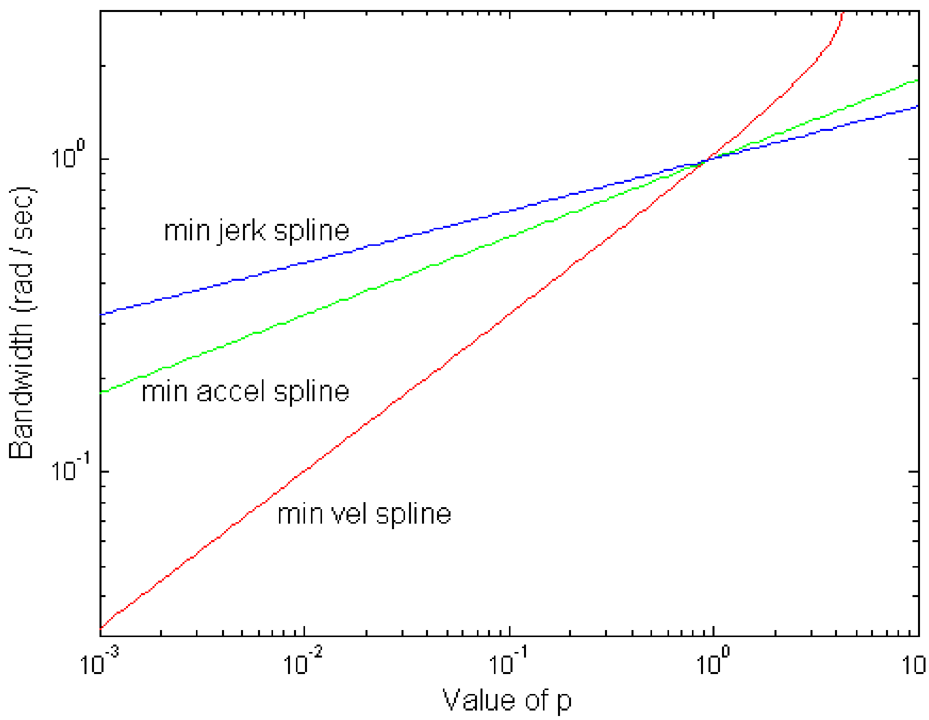

Figure 2: Bandwidths of the algebraic spline filters as a function of p when $h = 1$.

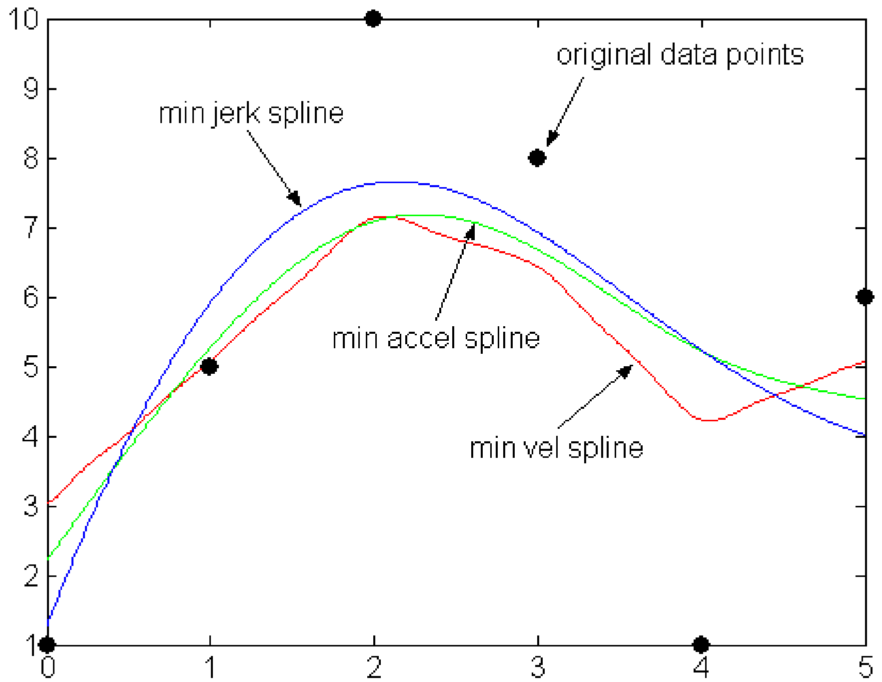

Figure 3: Algebraic spline examples ($p = 1$, $h = 1$).

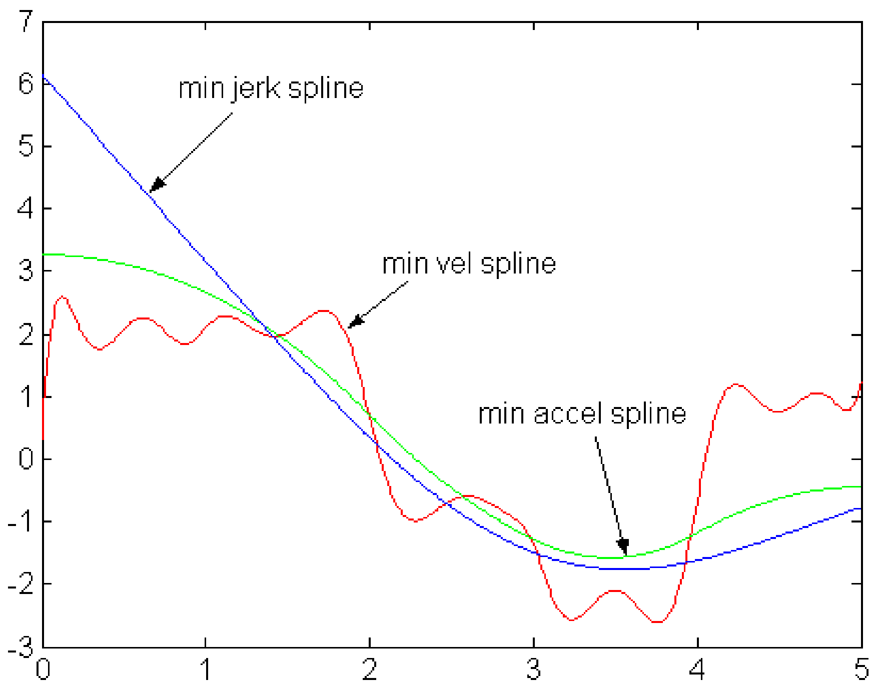

Figure 4: First derivatives of algebraic spline examples ($p = 1$, $h = 1$).

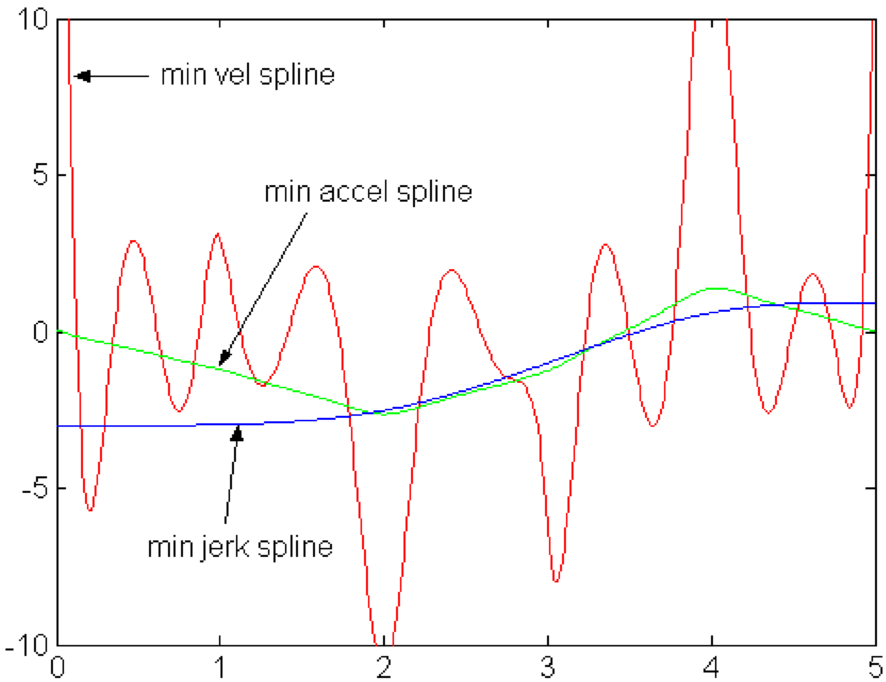

Figure 5: Second derivatives of algebraic spline examples ($p = 1$, $h = 1$).

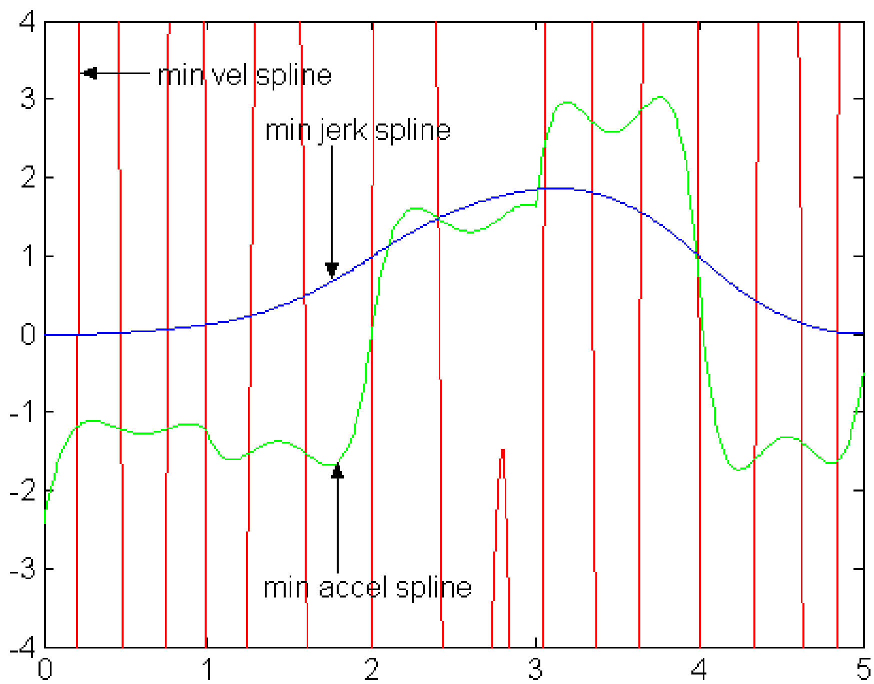

Figure 6: Third derivatives of algebraic spline examples ($p = 1$, $h = 1$).

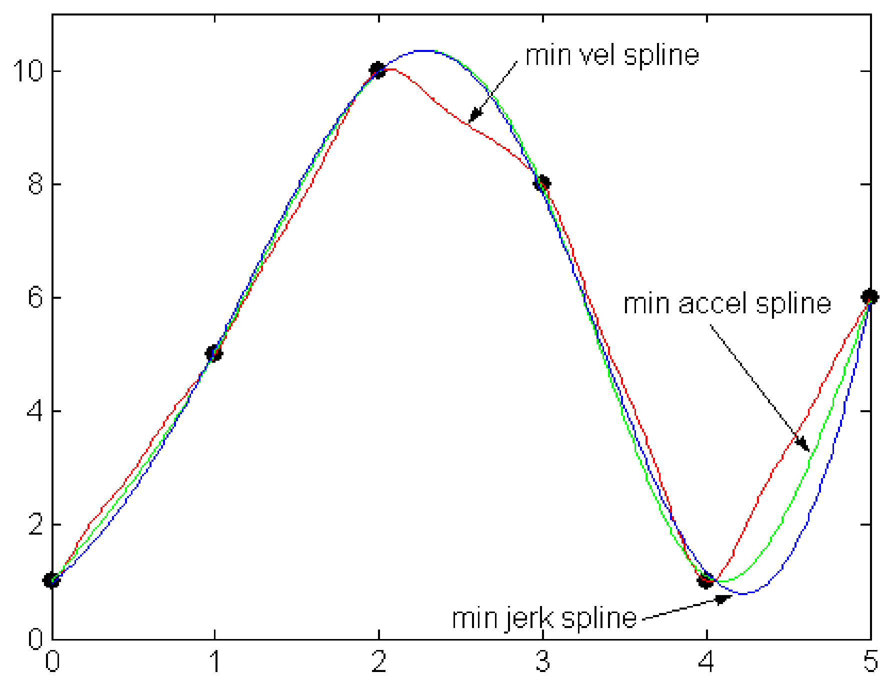

Figure 7: Algebraic spline examples ($p = 500$, $h = 1$).

Figure 8: Close-ups of Lenna's right eye – (a) Original image; (b) Minimum velocity spline; (c) Minimum acceleration spline (cubic spline); (d) Minimum jerk spline.