RECURSIVE SPLINE INTERPOLATION METHOD FOR REAL TIME ENGINE CONTROL APPLICATIONS

A. Stotsky
Volvo Car Corporation
Engine Design and Development
Dept. 97542, HA1N,
SE- 405 31 Gothenburg
Sweden.
Email: `astotsky@volvocars.com`

A. Forgo
Volvo Car Corporation
Engine Design and Development
Dept. 97568, HA1S
SE- 405 31 Gothenburg
Sweden
Email: `aforgo@volvocars.com`

Keywords: Spline interpolation method, Observers, Estimation, Combustion quality monitoring, Spark ignition engines.

Abstract In this paper new computationally efficient algorithms for spline interpolation method are explained. Theoretical comparative analysis of the spline interpolation method with combined high-gain observer and spline interpolation method is presented. New spline interpolation algorithms are implemented for estimation of the engine angular acceleration from the crankshaft angle measurements.

1 Introduction

Numerical calculation of the derivatives of a signal is an old problem in numerical analysis and digital signal processing. The backward difference method gives one of the simplest numerical differentiators. Despite the fact that it is quite common in engineering applications the behavior of the derivative is very often accompanied with peaking phenomena. Spline interpolation method proposed in [2] is based on on-line least-squares polynomial fitting over the moving in time window of a size w. The advantage of this method over the backward difference method is its good transient behavior. The idea for the spline interpolation method is to fit a polynomial of a certain order as a function of time in least squares sense and take the derivatives analytically. Properties of this method are described in [2],[1]. However, several practical problems remain. Relatively large window size w requires more on-line computations and makes practical implementation of the method difficult. This necessitates the development of computationally efficient recursive algorithms. Moreover, in [2],[1] only constant intersampling time is considered, which in many practical applications is not constant. For example, crankshaft angle measurements in automotive engines are based on the measurements of the crankwheel tooth number and therefore the elapsed time between two teeth passing a fixed point varies. This necessitates the development of recursive computationally efficient algorithms for variable discretization step.

The contributions of this paper are the following : new computationally efficient recursive spline interpolation algorithms for variable discretization step ; theoretical comparative analysis of the spline interpolation method and combined spline method with high gain observer; real-time implementation of the spline interpolation method for the crankshaft acceleration estimation.

2 Estimation of the derivatives of signal

The first step is to choose the interpolating polynomial as

$$\hat{\alpha} = c_0 + c_1 t + ... + c_n t^n \qquad (1)$$

where $\hat{\alpha}$ is an estimate of the measured signal α, t is continuous time, c_i,$i = 0, ..., n$ are coefficients to be found. The estimates of the derivatives are obtained by differentiating (1) analytically.

Suppose that there is a measured window of data $\{\alpha_{k-(w-1)},...,\alpha_k\}$ of a size w, measured discretely with variable discretization step. It is convenient to place the origin at the point $k-(w-1)$, where k is the step number, then there is a window of data $\{\alpha_0,...,\alpha_{w-1}\}$ measured at $\{0,...,t_{w-1}\}$ respectively. Variable discretization step is presented as $\Delta t_i = t_i - t_{i-1}$, $i = 1,...,w-1$. The sum to be minimized at every step is

$$S = \sum_{i=0}^{w-1} (\alpha_i - (c_0 + c_1 t_i + ... + c_n t_i^n))^2 \qquad (2)$$

where t_i, $i = 0,...,w-1$ is a discrete time which corresponds to the signal measurements α_i, $t_0 = 0$.

Minimum of S is achieved when equating to zero partial derivatives of S with respect to c_i, $i = 0,...,n$, i.e.,

$$\frac{\partial S}{\partial c_i} = 0 \qquad (3)$$

Equations (3) can be written as follows

$$c_0 \quad w + c_1 \sum_{i=0}^{w-1} t_i + ... + c_n \sum_{i=0}^{w-1} t_i^n = \sum_{i=0}^{w-1} \alpha_i \qquad (4)$$

$$\begin{aligned} c_0 & \sum_{i=0}^{w-1} t_i + c_1 \sum_{i=0}^{w-1} t_i^2 + ... + c_n \sum_{i=0}^{w-1} t_i^{n+1} \\ = & \sum_{i=0}^{w-1} \alpha_i t_i \qquad (5) \\ ... & ...... \\ c_0 & \sum_{i=0}^{w-1} t_i^n + c_1 \sum_{i=0}^{w-1} t_i^{n+1} + ... + c_n \sum_{i=0}^{w-1} t_i^{2n} \\ = & \sum_{i=0}^{w-1} \alpha_i t_i^n \qquad (6) \end{aligned}$$

where equation (4) represents $\frac{\partial S}{\partial c_0} = 0$, equation (5) represents $\frac{\partial S}{\partial c_1} = 0$ and finally equation (6) represents $\frac{\partial S}{\partial c_n} = 0$.

The system (4) - (6) should be resolved at every step in order to find the coefficients c_i, $i = 0,...,n$. It is clear that for a sufficiently large window size w, the calculation of sums in the system (4) - (6) requires a lot of computational power and our next step is to present recursive computationally efficient algorithms to compute these sums.

Let us consider one step of the moving window of a size w. Suppose that at step $k-1$ there is the following data $\{\alpha_0,...,\alpha_{w-1}\}$ measured at $\{0,...,t_{w-1}\}$, and at step k there is $\{\alpha_1,...,\alpha_w\}$ measured at $\{0,...,t_w\}$. The new value α_w measured at the time t_w enters the window (buffer) and the value α_0 leaves the window. Our problem statement is to find computationally efficient recursive algorithms to compute the sums $S_{m_k} = \sum_{i=1}^{w} t_i^m$ via the sums on the previous step $S_{m_{k-1}} = \sum_{i=0}^{w-1} t_i^m$, where $m = 1,...2n$.

Define the sum of order m at step k as follows

$$\begin{aligned} S_{m_k} &= \sum_{i=1}^{w} t_i^m = \Delta t_2^m + (\Delta t_2 + \Delta t_3)^m + ... \\ &+ (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^m \qquad (7) \end{aligned}$$

where $t_1 = 0$.
The sum (7) should be computed using the same sum on a previous step $k-1$ which can written as

$$\begin{aligned} S_{m_{k-1}} &= \sum_{i=0}^{w-1} t_i^m = \Delta t_1^m + (\Delta t_1 + \Delta t_2)^m + ... \\ &+ (\Delta t_1 + \Delta t_2 + ... + \Delta t_{w-1})^m \qquad (8) \end{aligned}$$

and the sums of lower order at step k which are defined as

$$\begin{aligned} S_{(m-j)k} &= \sum_{i=1}^{w} t_i^{m-j} = \Delta t_2^{m-j} + (\Delta t_2 + \Delta t_3)^{m-j} + ... \\ &+ (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^{m-j} \qquad (9) \end{aligned}$$

where $1 \le j \le (m-1)$, $m > 1$.

Starting with (8) and using the following identity

$$(x+y)^m = \sum_{j=0}^{m} C_j^m x^j y^{m-j}, \qquad C_j^m = \frac{m!}{j!(m-j)!} \quad (10)$$

where $m = 0,1,2,...$, one gets

$$S_{m_{k-1}} = (w-1)\Delta t_1^m + \Delta t_2^m + (\Delta t_2 + \Delta t_3)^m + ...$$
$$+(\Delta t_2 + \Delta t_3 + ... + \Delta t_{w-1})^m$$
$$+\sum_{j=1}^{m-1} C_j^m (\Delta t_1^j \Delta t_2^{m-j} + ...$$
$$+\Delta t_1^j (\Delta t_2 + ... + \Delta t_{w-1})^{m-j}) \quad (11)$$

Notice that

$$\Delta t_2^m + (\Delta t_2 + \Delta t_3)^m + ...$$
$$+(\Delta t_2 + \Delta t_3 + ... + \Delta t_{w-1})^m$$
$$= S_{mk} - (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^m \quad (12)$$

and

$$\sum_{j=1}^{m-1} C_j^m (\Delta t_1^j \Delta t_2^{m-j} + ...$$
$$+\Delta t_1^j (\Delta t_2 + ... + \Delta t_{w-1})^{m-j})$$
$$= \sum_{j=1}^{m-1} C_j^m \Delta t^j (S_{(m-j)k}$$
$$-(\Delta t_2 + ... + \Delta t_w)^{m-j}) \quad (13)$$

Substituting (12) and (13) in (11) one gets

$$S_{m_k} = S_{m_{k-1}} - (w-1)\Delta t_1^m$$
$$-\sum_{j=1}^{m-1} C_j^m \Delta t^j (S_{(m-j)k} - (\Delta t_2 + ... + \Delta t_w)^{m-j})$$
$$+(\Delta t_2 + \Delta t_3 + ... + \Delta t_{w-1})^m \quad (14)$$

where $C_j^m = \frac{m!}{j!(m-j)!}$, $1 \leq j \leq (m-1)$, $m > 1$.

Our next step is to calculate the sums on the right hand side of the equations (4) - (6). The sums $S_{\alpha m_k} = \sum_{i=1}^{w} \alpha_i t_i^m$ should be calculated via the sums on the previous step $S_{\alpha m_{k-1}} = \sum_{i=0}^{w-1} \alpha_i t_i^m$, where $m = 2, ...n$.

Using similar arguments one can show that

$$S_{\alpha m_k} = S_{\alpha m_{k-1}} - (\alpha_1 + \alpha_2 + ... + \alpha_{w-1})\Delta t_1^m -$$
$$\sum_{j=1}^{m-1} C_j^m \Delta t_1^j (S_{\alpha(m-j)k} - \alpha_w (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^{m-j})$$
$$+\alpha_w (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^m \quad (15)$$

where

$$S_{\alpha m_{k-1}} = \sum_{i=0}^{w-1} \alpha_i t_i^m = \alpha_1 \Delta t_1^m + \alpha_2 (\Delta t_1 + \Delta t_2)^m + ...$$
$$+\alpha_{w-1} (\Delta t_1 + \Delta t_2 + ... + \Delta t_{w-1})^m \quad (16)$$

is the sum on the step $(k-1)$, and

$$S_{\alpha(m-j)k} = \sum_{i=1}^{w} \alpha_i t_i^{m-j} = \alpha_2 \Delta t_2^{m-j}$$
$$+\alpha_3 (\Delta t_2 + \Delta t_3)^{m-j} +$$
$$... + \alpha_w (\Delta t_2 + \Delta t_3 + ... + \Delta t_w)^{m-j} \quad (17)$$

where $1 \leq j \leq (m-1)$, $m > 1$.

The order of the interpolating polynomial should be so selected to be as low as possible in order to reduce the computational burden and to filter out measurement noise.

In the next Section the detailed solution of the interpolation problem for the second order polynomial is presented. This example is used in Section 4 for the crankshaft acceleration estimation.

3 Second Order Example

3.1 Spline Interpolation Method

For the second order polynomial (1), where $n = 2$, equations (4) - (6) can be written as follows

$$c_0 \quad w + c_1 \sum_{i=0}^{w-1} t_i + c_2 \sum_{i=0}^{w-1} t_i^2 = \sum_{i=0}^{w-1} \alpha_i \quad (18)$$
$$c_0 \quad \sum_{i=0}^{w-1} t_i + c_1 \sum_{i=0}^{w-1} t_i^2 + c_2 \sum_{i=0}^{w-1} t_i^3 = \sum_{i=0}^{w-1} \alpha_i t_i \quad (19)$$

$$c_0 \sum_{i=0}^{w-1} t_i^2 + c_1 \sum_{i=0}^{w-1} t_i^3 + c_2 \sum_{i=0}^{w-1} t_i^4 = \sum_{i=0}^{w-1} \alpha_i t_i^2 \quad (20)$$

where the sums are computed using recursive formulas (14) and (17).
Presenting equations (18) - (20) in matrix form yields

$$Ac = b \quad (21)$$

where

$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$, $c^T = (c_0, c_1, c_2)$, $b^T = (b_1, b_2, b_3)$, where

$$\begin{aligned} a_{11} &= w, \quad a_{12} = S_{1k}, \quad a_{13} = S_{2k} \\ a_{21} &= S_{1k}, \quad a_{22} = S_{2k}, \quad a_{23} = S_{3k} \\ a_{31} &= S_{2k}, \quad a_{32} = S_{3k}, \quad a_{33} = S_{4k} \\ b_1 &= S_{\alpha k}, \quad b_2 = S_{\alpha 1k}, \quad b_3 = S_{\alpha 2k} \end{aligned}$$

Notice that, $a_{12} = a_{21}$ and $a_{22} = a_{31} = a_{13}$, $a_{23} = a_{32}$.

In order to find spline coefficients c_i, $i = 0, 1, 2$ the matrix equation (21) should be solved with respect to c_i at every step ($c = A^{-1}b$). To this end matrix A is inverted analytically.

Remark 1. This algorithm has only one parameter to be optimized, this being the size of the moving window w. If the derivative of the measured signal changes slowly it is advisable to have a relatively large window size to filter out measurement noise. If the derivative changes quickly, the window size should be sufficiently small to capture corresponding fast changes in the derivative. The disadvantage of a small window size is the noise in the estimated signal. Ideally, the window size should be adjustable so that it is small enough during transients to capture fast changes in the derivative of the signal, and large enough under steady-state conditions to filter out measurement and space-discretization noise.

Remark 2. For constant discretization step the sums S_m do not change with time, matrix A is constant and computational burden is minimal.

Remark 3. The recursive computations presented above accumulate an approximation error, which increases with time. To avoid this error accumulation problem, repeatable initialization of the algorithms is required.

3.2 Combination of high gain observer and spline interpolation method and their comparative analysis

In [2], [1] it is mentioned that the spline interpolation method can be combined with high-gain observers. The combined scheme shows improved transient behavior. In this Section the comparative analysis of the spline interpolation method and combined method is presented.

The signal is estimated via the second order polynomial described above

$$\hat{\alpha}(t) = c_0 + c_1 t + c_2 t^2 \quad (22)$$

where $\hat{\alpha}(t)$ is the estimate of the signal α.

A combination of the spline interpolation algorithm with a simple high gain observer is presented as follows

$$\begin{aligned} y(t) &= \frac{1}{\tau}\alpha + y_1(t) & (23) \\ \dot{y}_1(t) &= -\frac{1}{\tau}y(t) + \hat{\ddot{\alpha}}, y_1(t_0) = \hat{\dot{\alpha}}(t_0) - \frac{1}{\tau}\alpha(t_0) & (24) \end{aligned}$$

where $y(t)$ is the estimate of the $\dot{\alpha}$, $\tau > 0$ is the algorithm parameter, $\hat{\dot{\alpha}}(t_0)$ is the estimate of the derivative from spline interpolation method evaluated at time t_0, $\hat{\ddot{\alpha}}$ is spline estimate of the second derivative. The condition $y_1(t_0) = \hat{\dot{\alpha}}(t_0) - \frac{1}{\tau}\alpha(t_0)$ corresponds to the following initial value assignment $y(t_0) = \hat{\dot{\alpha}}(t_0)$. This means that high gain observer (23), (24) is initialized to the spline estimate. Moreover, the second order derivative $\hat{\ddot{\alpha}}$ is used directly as an input to the high gain observer. Notice that, $\hat{\dot{\alpha}}(t_0) = c_1 + 2c_2 t_0$ and $\hat{\ddot{\alpha}} = 2c_2$.

Remark 4. Equations (23) and (24) represent the following estimator of the derivative

$$y(p) = \frac{p}{\tau p + 1}\alpha(p) \quad (25)$$

if $\hat{\ddot{\alpha}} = 0$, where $p = jw$ is a Laplace variable. Estimates from the spline interpolation method are used as a feedforward part in the estimator (25).

Assuming a spline approximation error, then this error is expressed in terms of the error in spline coefficients and the true signal can be expressed as follows

$$\alpha(t) = c_{0*} + c_{1*}t + c_{2*}t^2 \tag{26}$$

where $\alpha(t)$ is the signal, c_{i*}, $i = 0, 1, 2$ are true coefficients.

Assuming a constant error in the coefficients w.r.t. time one gets

$$c_i = c_{i*} + \Delta c_i \tag{27}$$

where Δc_i, $i = 0, 1, 2$ are constant errors in spline coefficients.

Our task is to compare the following estimation errors

$$e_1(t_1) = \hat{\dot{\alpha}}(t_1) - \dot{\alpha}(t_1) \tag{28}$$
$$e_2(t_1) = y(t_1) - \dot{\alpha}(t_1) \tag{29}$$

where t_1 is a fixed time.

First we evaluate $e_1(t_1)$

$$e_1(t_1) = \Delta c_1 + 2\Delta c_2 t_1 \tag{30}$$

The next step is to evaluate $e_2(t_1)$. Differentiating (23) and taking into account that $\hat{\ddot{\alpha}} = \ddot{\alpha} + 2\Delta c_2$ one gets

$$\dot{y} - \ddot{\alpha} = -\frac{1}{\tau}(y - \dot{\alpha}) + 2\Delta c_2 \tag{31}$$

The solution of (31) is the following

$$y(t) - \dot{\alpha}(t) = (y(t_0) - \dot{\alpha}(t_0) - 2\Delta c_2\tau)e^{-\frac{t - t_0}{\tau}} + 2\Delta c_2\tau \tag{32}$$

Initialization of high gain observer gives the following: $y(t_0) = \hat{\dot{\alpha}}(t_0)$. Taking into account that $\hat{\dot{\alpha}}(t_0) - \dot{\alpha}(t_0) = \Delta c_1 + 2\Delta c_2 t_0$ one obtains

$$y(t) - \dot{\alpha}(t) = (\Delta c_1 + 2\Delta c_2 t_0 - 2\Delta c_2\tau)e^{-\frac{t - t_0}{\tau}} + 2\Delta c_2\tau \tag{33}$$

Evaluating $e_2(t_1)$ one gets

$$e_2(t_1) = (\Delta c_1 + 2\Delta c_2 t_0 - 2\Delta c_2\tau)e^{-\frac{t_1 - t_0}{\tau}} + 2\Delta c_2\tau \tag{34}$$

Comparing (34) and (30), it can be seen that $|e_2(t_1)|$ can always be made smaller than $|e_1(t_1)|$ by reducing the design parameter τ. However, in the presence of noise the reduction of τ leads to a deterioration of the signal quality. The advantages of this combined scheme can be shown if the parameter τ can be reduced to a sufficiently small value without any signal quality deterioration.

As the simulation results did not show any significant superiority of the combined method over the spline interpolation method and the latter is chosen for implementation.

4 Implementation Results

In this Section the implementation results are presented for a second order spline fitting method, as described in Section 3.1. A Volvo S80 passenger car equipped with a crankshaft angle sensor is used for experiments. The method described in Section 3.1 is used to estimate the crankshaft acceleration,which gives vital information about the quality of the combustion in the engine, from the crankshaft angle measurements.

The car is equipped with a special Engine Control Unit, which is called Volvo Rapid Prototyping System. This system consists of the PM (Power Module) and the AM (Application Module) connected together via a 1 Mbit/s CAN channel.

The PM provides information based on a number of engine sensor signals. The crankshaft wheel has 58 teeth spaced every 6 degrees and a gap corresponding to two " missing teeth ". Engine speed and cylinder position can be calculated from the resulting crankshaft sensor signal. The PM calculates the engine speed based on the measured time between the tooth events via the backward difference method. This engine speed is transmitted along with other measurements every 4th millisecond to the AM.

The 4 millisecond sampled crankshaft angle calculated from the tooth number signal is the input to the crankshaft acceleration estimation, which runs in the

AM. The accuracy of the estimation is verified by comparing the engine speed, calculated by the spline interpolation method, with the engine speed calculated in the PM.

Implementation results are presented in Fig.1 which shows engine speeds as a function of time. The accuracy of the estimation is verified by comparing fast engine speed measured in the PM module (dotted line) and engine speed calculated via spline interpolation method (solid line).

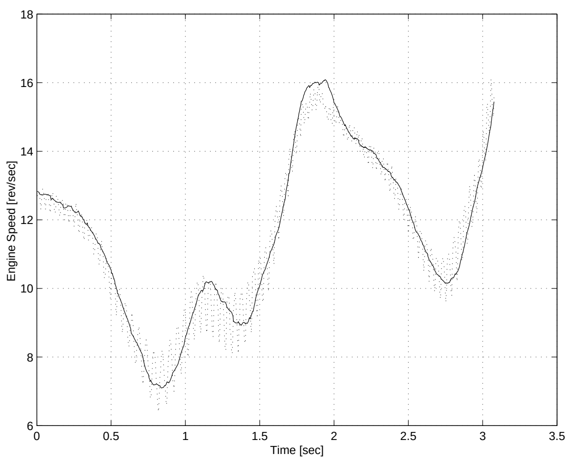

Figure 1
Experimental results. The comparison of two engine speeds as functions of time. Engine speed computed in PM is plotted with dotted line. Engine speed calculated with spline interpolation method is plotted with solid line.

Figure 2 shows an application of the proposed recursive spline interpolation method to the misfire diagnostics. In this Figure two engine cycles of a 5 cylinder engine are shown, and the misfire is generated on the second cycle of cylinder N 4. The tooth number signal is plotted with a dashdot line. Crankshaft acceleration estimated by the spline interpolation method is plotted with a solid line. In the event of a misfire, the crankshaft acceleration changes dramatically, which permits the cylinder individual misfire detection.

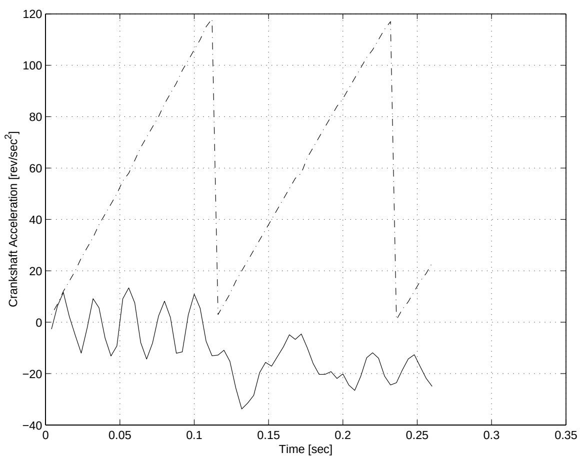

Figure 2
Experimental results. Two engine cycles of 5 cylinder engine. The misfire is generated on the second cycle in the cylinder 4. Tooth number signal is plotted with dashdot line. Crankshaft acceleration estimated by the spline interpolation method is plotted as a function of time with a solid line.

5 Conclusion

Computationally efficient algorithms for spline interpolation method allowing a reduction in computational burden as well as the implementation of the spline interpolation method are presented. A further challenge is to improve the performance of these algorithms by adjusting the size of the moving window.

References

[1] Dabroom A., Khalil H. (1999). Discrete-Time Implementation of High-Gain Observers for Numerical Differentiation, *International Journal of Control* **72** (17), p.p. 1523-1527.

[2] Diop S., Grizzle J., Moraal P., Stefanopoulou A. (1994). Interpolation and Numerical Differentiation for Observer Design, *Proc. American Control Conference*, Baltimore, Maryland, p.p. 1329 - 1333.