

Peak Points Detection Using Spline Interpolation Based on FPGA Implementation

Alperen Mustafa COLAK¹, Taito MANABE¹, Ryo KAMASAKA¹, Yuichiro SHIBATA¹, Fujio KUROKAWA²

¹Graduate School of Engineering, Nagasaki University, Nagasaki, Japan

²Nagasaki Institute of Applied Science, Nagasaki, Japan

colak@pca.cis.nagasaki-u.ac.jp

Abstract—Peak detection of any time serial signals can be done using different methods. However, exact positions of the peaks cannot be achieved, if the method does not have a high sensitivity detection capability. It is known that various errors appear between the real peak value and the measured value; therefore, a true peak value can be obtained from the measured value by using a highly sensitive detection method. For this reason, spline interpolation is used to estimate the peak points from measured values. In this paper, highly sensitive peak detection of high-frequency signals based on FPGA is proposed using spline algorithm, which is capable of calculating the middle points using the first and last points on the signal. To evaluate the system, the histograms, the mean, and the variance of the peak values before and after spline interpolation are compared. Simulation results show that calculated peak points using splines and real peak values of a signal are very close to each other since the spline algorithm has very high calculation sensitivity. In addition, spline results show us how much error the existing peak values have.

Index Terms—spline, architecture, field programmable gate arrays, peak, detection.

I. INTRODUCTION

Interpolation is often used as one of the operations of signal processing, similar to sampling and conversion. Therefore, many researchers have been using interpolation methods to analyze data of different applications. Some of the interpolation methods are as following: B-spline interpolation using linear approximations for setting up viewing and projection matrices and describing complex objects were studied in reference [1], implementing a cubic spline interpolation algorithm on DSP was studied in reference [2], an iterative linear interpolation based on fuzzy gradient model for low-cost VLSI implementation was introduced by reference [3], a polynomial interpolation for space-efficient verifiable secret sharing was given in reference [4], an interpolation method to investigate the improvement in image quality for ground penetrating radar (GPR) acquisition was highlighted in reference [5], gauss interpolation algorithms for nonlinear rational parameters were presented in reference [6], four image interpolation methods for 2-D AR modeling were given in reference [7], a polynomial approach to evaluate the fragmented function approach for two secure two-party computation (STPC) was introduced in reference [8], a bivariate splines in piecewise constant tension as the solution for a functional minimization problem was given in reference [9], a cubic spline interpolation algorithm for smoothness interpolation model of non-circular curve mechanical mold was studied in

reference [10], a spline interpolation functions for solution of a non-linear equation was given in reference [11], a view interpolation method from defocused stereo images using linear filtering was introduced in reference [12], a linear interpolation effects on signal transferring was highlighted in reference [13], a novel and fast cubic B-splines algorithm for cancellation of random valued impulse noise was given in reference [14], a real time implementation of cubic B-spline algorithm for electro optical tracking system was studied in reference [15], a cubic B-spline curves based research of the approximate algorithm was given in reference [16], and a fast algorithm for quadratic and cubic spline wavelets was provided in reference [17]. Among these, the spline curve makes it easy to build an interface that will allow designing and controlling the shape of complex curves and surfaces by using low-degree polynomials in each of the intervals and by choosing the polynomial pieces such that they fit smoothly together.

Spline was also used in reference [18] for control and optimization of UAV trajectory for aerial coverage in photogrammetry applications, in reference [19] for modeling of a switched reluctance generator using cubic spline coefficients on the phase flux linkage, inductance and torque equations, and in reference [20] for modeling shapes for pattern recognition by a simple low-cost spline-based approach.

In addition to the papers given above, there are other studies including hardware implementations of spline. In reference [21], the streamed implementation of a higher order interpolation filter, with a weighted median classifier was explored. In reference [22], a bilinear interpolation algorithm to demosaic images with bayer color pattern was presented; the algorithm was implemented in a single Field Programmable Gate Array (FPGA) device using a pipelined architecture. In reference [23], FPGA implementation of bilinear interpolation algorithm for color filter array demosaicing was introduced, as real-time FPGA processing for high-speed optical frequency domain, imaging was studied in reference [24].

On the other hand, in our previous work, we showed that the automatic multiscale-based peak detection (AMPD) [25] method has been proven to be effective for periodic and quasi-periodic noisy signals. However, one of the problems about the AMPD peak detection is that processing time of a signal to be analyzed becomes longer due to the sampling point of AC/DC converter. Basically, it happens when the peak point is detected using variance that is extensive data on the time axis due to the deviation. Consequently, we can say that the problem is to have a big data enlarging time

axis. To solve this problem from the detected peak data, it is necessary to interpolate the assumed function and estimate the maximum value (correct peak position). Representation of interpolation method includes Lagrange polynomial interpolation, Newton polynomial interpolation and spline interpolation, etc.

In this paper, we proposed a highly sensitive real peak point estimation system using the spline interpolation based on AMPD. For this reason, the data was obtained from spline interpolation by means of AMPD peak values. To solve the spline interpolation, it is necessary to solve simultaneous equations. As the amount of calculation time increases, in general real-time in hardware that has been deemed not suitable for processing, also increases. However, in this paper, the calculation process is simplified using the fact that the data sampling interval is constant. To design a simple spline algorithm hardware, a circuit architecture is proposed for complementary in-line processing implemented on an FPGA. The reason of introducing spline interpolation is to obtain precise peak point values, even if the sampling frequency is relatively low. FPGA implementation of a high-accuracy peak value estimation system from high frequency signals has been proposed in this paper. It is the FPGA implementation of a spline interpolation system that estimates the peak value. In addition, the histogram of the detected peak value with respect to the sine wave input was evaluated and the effect of the system was confirmed. Consequently, this study will allow us to estimate the real peak value of data at both low and high sampling frequencies.

II. MATERIALS AND METHODS

A. Algorithm

Estimation of intermediate values between precise data points is necessary for the signal processing. Polynomial interpolation algorithms such as Lagrange interpolation or Newton interpolation are the most common methods used for this purpose. However, in some cases these functions can cause incorrect results due to round-off error and overshoot. Therefore, lower-order polynomials can be applied to subsets of data points as an alternative approach. These polynomials connected with each other are named as spline functions. In addition, spline interpolations [26] have less vibration and more resistance against noise compared to Lagrange interpolation and Newton interpolation. The general idea of a spline is represented in Fig. 1 [27] that is drawn by connecting all points and then it is formulated as a function as in Equation 1.

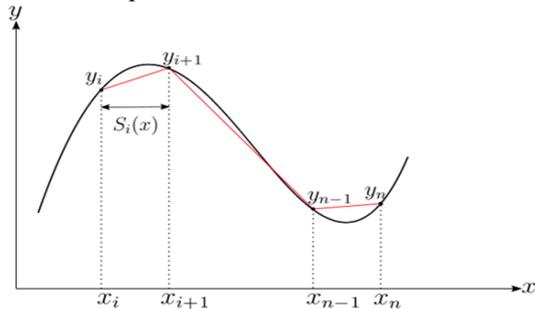


Figure 1. Analysis of spline

Linear, quadratic, and cubic splines are well-known spline interpolations. However, among these, linear spline and

quadratic spline are not used that much because of low accuracy. Moreover, if the degree becomes too large, the calculation time will increase. In general, cubic spline functions are often used from the point of balance of interpolation accuracy and computational complexity. Additionally, spline interpolation encompasses a range of interpolation techniques to reduce the effects of overfitting. Furthermore, it applies only in one dimension but is useful for modeling yield curves, forward curves, and other term structures.

The objective in cubic spline is to derive a third-order polynomial for each interval between knots, as in Equation (1).

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (1)$$

The piecewise polynomial is S_i ($i = 0, 1, 2, \dots, n - 1$) in the interval $[x_i, x_{i+1}]$. Thus, for $n + 1$ data points ($i = 0, 1, 2, \dots, n - 1$), there are n intervals. Consequently, to determine the cubic spline S_i , we must determine the coefficients a_i , b_i , c_i and d_i . For this reason, we follow some conditions as given below;

Condition 1:

S_i is on (x_i, y_i)

$$S_i(x_i) = y_i, \quad (i = 0, \dots, n - 1) \quad (2)$$

Condition 2:

Each polynomial passes through its respective end points:

S_i is on (x_{i+1}, y_{i+1})

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}) = y_{i+1}, \quad (i = 0, \dots, n - 1) \quad (3)$$

Condition 3:

First derivatives match at interior points:

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \quad (i = 0, \dots, n - 2) \quad (4)$$

Condition 4:

Second derivatives match at interior points:

$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), \quad (i = 0, \dots, n - 2) \quad (5)$$

Condition 5:

Second derivatives are vanished at the end points:

$$S_0''(0) = S_{n-1}''(0) = 0 \quad (6)$$

Here, if the value of the second derivative is u_i when $x = x_i$ and the size of the interval $(x, x_i + 1)$ is h_i , it is expressed by the following equation:

$$u_i = S''(x_i) = 2b_i, \quad (i = 0, 1, 2, \dots, n) \quad (7)$$

Then, h_i can be calculated as in Equation 8.

$$h_i = x_{i+1} - x_i, \quad (i = 0, 1, 2, \dots, n) \quad (8)$$

Thus, all unknowns a_i , b_i , c_i and d_i can be represented by u_i and known h_i and y_i expressions.

$$a_i = \frac{u_{i+1} - u_i}{6h_i} \quad (9)$$

$$b_i = \frac{u_i}{2} \quad (10)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2u_i + u_{i+1}) \quad (11)$$

$$d_i = y_i \quad (12)$$

If u_i is found, all unknown coefficients are determined. u_i can be obtained from condition 3 and Equation (1). The

final simultaneous equation of u_i is as follows.

$$(x_{i+1} - x_i)u_i + 2(x_{i+2} - x_i)u_{i+1} + 1 + (x_{i+2} - x_{i+1})u_{i+2} = 6 \left(\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right) \quad (13)$$

Here, substituting v_i for the constant term on the right side, the Equation (13) can be expressed by the following Equations (14) and (15).

$$v_i = 6 \left(\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right) \quad (i=1, \dots, n-1) \quad (14)$$

$$v_i = h_{i-1}u_{i-1} + 2(h_{i-1} + h_i)u_i + h_i u_{i+1} \quad (i=1, \dots, n-1) \quad (15)$$

By solving Equation (15), u_i can be determined. When replacing the constant term with v_i , the simultaneous linear equation obtained by substituting $u_0 = u_n = 0$ is shown as follows.

$$\begin{bmatrix} 2(h_0 + h_1) & h_1 & & & & & & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & & & & & \\ & h_2 & 2(h_2 + h_3) & h_3 & & & & \\ & & & \ddots & & & & \\ & & & & h_{i-1} & 2(h_{i-1} + h_i) & h_i & \\ & & & & & & \ddots & \\ 0 & & & & & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_i \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_i \\ \vdots \\ v_{n-1} \end{bmatrix} \quad (16)$$

The matrix on the left-hand side of Equation (16) is a tridiagonal matrix, and all the elements on the variable side are characterized by a linear equation of the difference of x . Finally, when all variables are calculated such as a_i , b_i , c_i and d_i , we can plot the spline figure by using the peak points.

B. The Gauss-Jordan method

In the process of spline interpolation, the solution is necessary for simultaneous equations. For this reason, we prefer the Gauss-Jordan method to solve the tridiagonal matrix in Equation (16). The Gauss-Jordan method [28] is an improvement of the sweeping method. Although it is more than the sweeping out method in the calculation amount, there is the advantage that a direct solution is obtained. Generally, when a simultaneous equation such as Equation (17) is given, it is a method of obtaining a solution by transforming it as shown in Equation (18).

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & \cdots & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & \cdots & \cdots & x_{2,n} \\ x_{3,1} & x_{3,1} & x_{3,3} & x_{3,4} & \cdots & \cdots & x_{3,n} \\ x_{4,1} & x_{4,1} & x_{4,3} & x_{4,4} & \cdots & \cdots & x_{4,n} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \cdots & \vdots \\ x_{n,1} & x_{n,2} & x_{n,3} & x_{n,4} & \cdots & \cdots & x_{n,n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ \vdots \\ z_n \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_n \end{bmatrix} \quad (18)$$

By deforming like this, $y_n = z_n$ is obtained, then y_n can easily be obtained.

III. SPLINE ALGORITHM ARCHITECTURE DESIGN

Spline algorithm architecture design can be explained by Fig. 2. It includes a selector module, a differentiation module and interpolation modules. If the spline system is explained simply, it is assumed that 8 pieces of data are sent simultaneously for each clock. So, there are 8 input ports d_i ($i = 0, \dots, 7$) to receive the 8 data as well as flags for determining whether or not each data point is a peak at same clock cycle. The value input to d_{in0} is the oldest data among the 8 points, and d_{in7} is the newest data.

The input d_i is sent to the selector module, and some part of d_i is stored in the register between d_i ($i = 0, \dots, 7$) at the same clock. Thereafter, data adjacent to the registers y_i ($i = 0, \dots, 7$) is stored. The data stored in the register d is sent to the selector, and at the same time, a part of it is stored in the register p_i ($i = 0, \dots, 3$). The data stored in the register p is sent to the selector.

In the selector module, 8 points necessary for interpolation are determined from 15 pieces of data. The 8 points to be interpolated are 8 points adjacent to the center of the position of the peak data stored in register d . Thereafter, data adjacent to the register y_i ($i = 0, \dots, 7$) is stored so that the peak data is stored in y_4 . The data of register y is sent to the differentiation module to calculate the tridiagonal matrix part of the spline function. The calculated value sp_i ($i = 3, \dots, 5$) in the differentiation module and the received value y_i ($i = 3, \dots, 5$) from the selector module are sent to the interpolation module. In the interpolation module, an interpolation value for a certain time x is obtained by solving a spline equation. Note that the time x is interval between peak points. The real peak value is selected from the obtained interpolation value spl_i ($i = 0, \dots, 9$) and the peak data sent from delay before interpolation. The selected real peak value is output as interpolated peak data.

A. Selector module

Fig. 3(a) and Fig. 3(b) are diagrams showing the correspondence between input data and registers in which data is stored. The input data is stored in the register in this way. Fig. 3(b) illustrates the figure where 8 points to be interpolated are determined by defining the peak position from the peak flag of register d . Register y that stores 8 points to be interpolated contains any data from p_0 to d_{in7} that are data from y_0 to y_3 at the four points before the peak point, data at the peak at y_4 , and data after the peak at y_5 to y_7 . In the figure, the peak position is d_5 and its value is stored in y_4 .

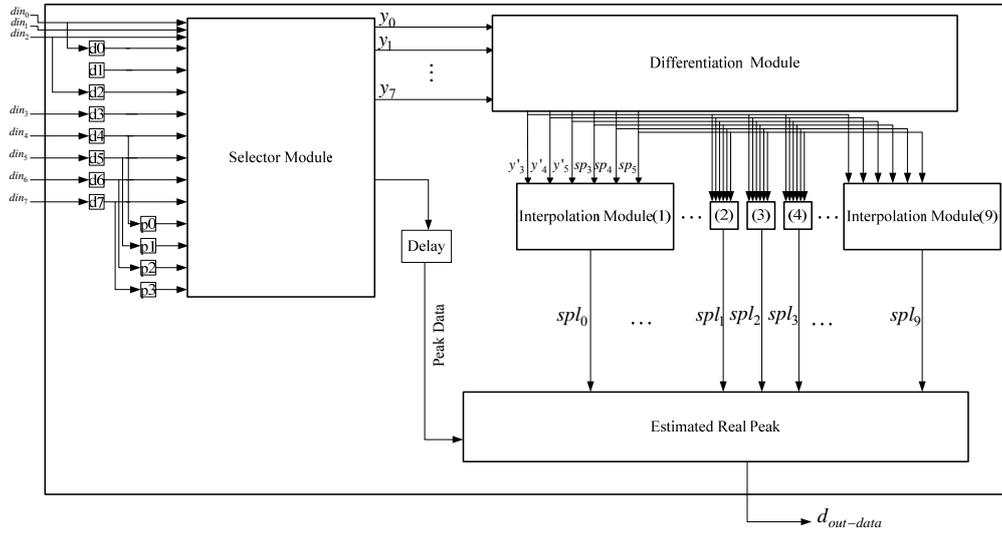


Figure 2. Spline architecture design

$$(x_{i+1} - x_i)u_{i-1} + 2(x_{i+1} - x_{i-1})u_i + (x_{i+1} - x_i)u_{i+1} = v_i, (i = 1, \dots, 6) \quad (19)$$

Dividing the whole of Equation (19) by $(x_i - x_{i-1})$ yields Equation (23). $h_i p_i$, h_i , sp_i are expressed by Equations (20) to (22). The definition of h_i is different from that of the previous section II.

$$h_i p_i = x_{i+1} - x_i, (i = 1, \dots, 6) \quad (20)$$

$$h_i = x_i - x_{i-1}, (i = 1, \dots, 6) \quad (21)$$

$$sp_i = \frac{v_i}{h_i}, (i = 1, \dots, 6) \quad (22)$$

$$sp_i = u_{i-1} + 2 \left(1 + \frac{h_i p_i}{h_i} \right) u_i + \frac{h_i p_i}{h_i} u_{i+1} \quad (23)$$

Here, when the coefficient of u_i in Equation (23) is dc_i and the coefficient of u_{i+1} is du_i , Equation (23) is expressed by Equation (26).

$$dc_i = 2(1 + du_i) \quad (24)$$

$$du_i = \frac{h_i p_i}{h_i} \quad (25)$$

$$u_{i-1} + dc_i u_i + du_i u_{i+1} = sp_i \quad (26)$$

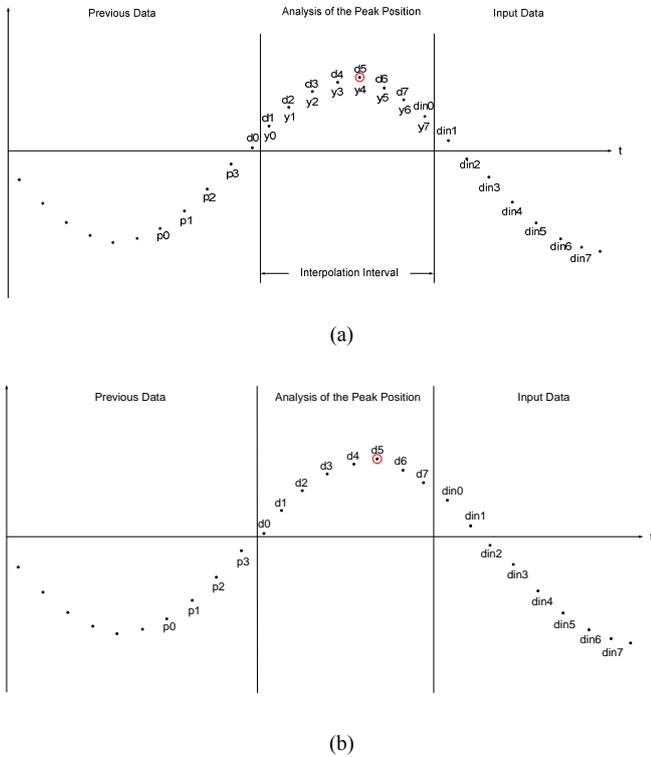


Figure 3. Selector module design

B. Differentiation module

In the differentiation module, the second derivative u_i ($i = 0, \dots, 6$) of each segment $S_i(x)$ ($i = 0, \dots, 6$) in spline interpolation is calculated. The calculation method of u_i is represented by Equation (16). In this implementation, since there are 8 data points input at the same time, there are 7 segments. Therefore, the tridiagonal matrix part is a 7×7 matrix. Also, since the data points are sampled at regular intervals, the calculation of Equation (8) is replaced as a constant, and the spline function can be simplified and solved. For computing tridiagonal matrix, the Gauss-Jordan method described in section 2 is used.

Here, the equation is modified to simplify the calculation. From the Equation (15), the tridiagonal matrix can be expressed as given in Equation (19). v_i is expressed by Equation (14).

When Equation (26) is written using a tridiagonal matrix, it becomes Equation (27), and this solution is obtained by the Gauss-Jordan method.

$$\begin{bmatrix} dc_1 & du_1 & & & & & \\ 1 & dc_2 & du_2 & & & & \\ & & \ddots & & & & \\ & & & 1 & dc_i & du_i & \\ & & & & \ddots & \ddots & \\ 0 & & & & & 1 & dc_6 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_4 \\ \vdots \\ u_6 \end{bmatrix} = \begin{bmatrix} sp_1 \\ sp_2 \\ \vdots \\ sp_4 \\ \vdots \\ sp_6 \end{bmatrix} \quad (27)$$

Fig. 4 depicts the data flow of the differentiation module. The fs_0 , fs_1 , and fs_2 portions of Fig. 4 calculate the sp_i value of Expression (26). Because, the data is sampled at regular intervals, the space between the data points is equal and the calculation can be simplified.

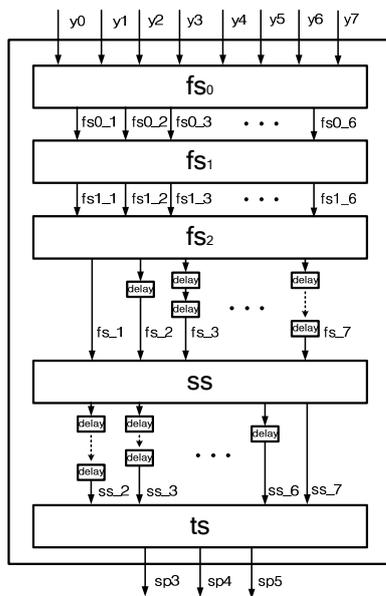


Figure 4. Differentiation module

In this implementation, the interval between data is set to 10 and put it in Equation (28).

$$x_1 - x_0 = x_2 - x_1 = \dots = x_7 - x_6 = 10 \quad (28)$$

Therefore, expressions (29) and (30) are obtained from Equations (20), (21) and (28).

$$h_i p_i = h_i = 10 \quad (29)$$

$$sp_i = \frac{v_i}{10} \quad (30)$$

Also, Equations (31) and (32) are obtained from Equations (24) and (25).

$$du_i = 1 \quad (31)$$

$$dc_i = 4 \quad (32)$$

Therefore, Equation (26) can be expressed by Equation (33).

$$u_{i-1} + 4u_i + u_{i+1} = \frac{v_i}{10} = \frac{1}{10} \left[\frac{6}{10} (y_{i+1} - y_i) - (y_i - y_{i-1}) \right] \quad (33)$$

In Fig. 5, Equation (33) is calculated separately from Equations (34) to (36).

$$fs_2 = \frac{fs_1}{100} \quad (34)$$

$$fs_1 = 6fs_0 \quad (35)$$

$$fs_0 = (y_{i+1} - y_i) - (y_i - y_{i-1}) \quad (36)$$

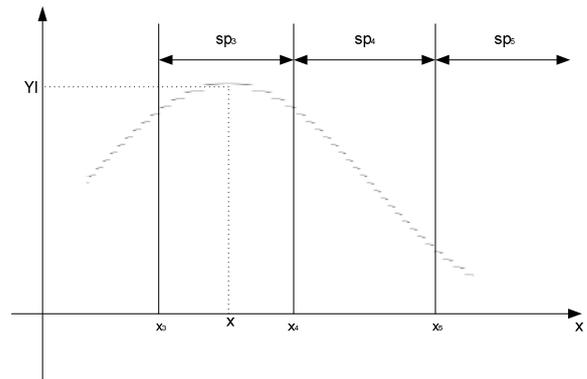
The *ss* part and *ts* part in Fig. 4 are calculated by the Gauss-Jordan method. Since the peak value is always stored in $y[4]$ and sent to differentiation module, u_i , which is necessary for subsequent calculations u_3 , u_4 , and u_5 . Therefore, only 6 data including y_3 , y_4 , and y_5 are sent to interpolation module and the algorithm can be simplified.

C. Interpolation module

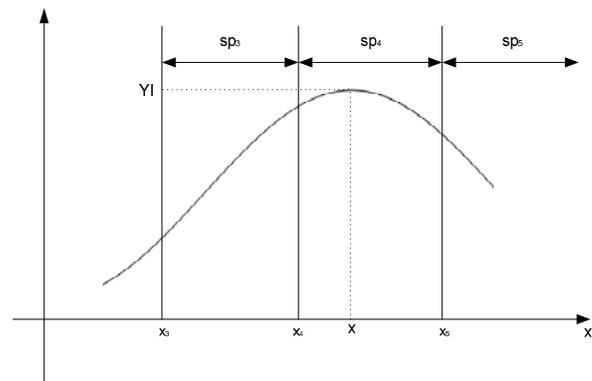
From the interpolation module given, based on sp_3 , sp_4 , sp_5 sent from differentiation module, interpolated value for a certain time x is output.

The interpolated value is represented by (x, YI) . x falls into the peak value and the time before and after the peak value. First, it is divided into cases where x is between x_3 and x_4 , and between x_4 and x_5 . This can be determined by

comparing the magnitudes of y_3 and y_5 . Figs. 5(a) and 5(b) show the image of the interpolated waveform and the peak value.



(a)



(b)

Figure 5. Interpolation waveform

When x is between x_3 and x_4 , and between x_4 and x_5 , the basic calculation method is the same through the numerical value is different. Here, the former will be described as an example.

Fig. 6 illustrates the data flow of the interpolation part. As a calculation procedure, coefficients of the piecewise polynomial are calculated using sp_3 , sp_4 , y_3 , and y_4 sent from differentiation module at the beginning. Calculation formulas of coefficients can be obtained by applying numerical values to formulas (9) to (12), respectively.

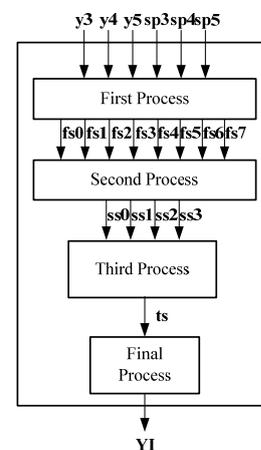


Figure 6. Block diagram of interpolation module

This calculation is performed in the *fs* part and the *ss* part in Fig. 6. The coefficients calculated at *fs* and *ss* are

corrected to the form of Equation (1) at ts and finally detect YI at x . In addition, this process is parallelized as depicted in Fig. 2, and it is possible to detect the target value at high speed.

Finally, the interpolated value spl_i sent from interpolation module and the maximum value of peak data before interpolation coming from delay is detected. The output value is output as interpolated peak data.

IV. RESULTS AND DISCUSSION

In this section, spline algorithm is evaluated in detail by comparing peak points of the same data series used in AMPD algorithm in the previous study of authors. Basic definition of AMPD method is to detect the real position of any time series data. It calculates all input data by using matrix equation to find the peak points.

The results obtained from both AMPD algorithm (raw-peak data), and spline algorithm (fitted-peak data) have been compared in order to estimate the right positions of the peak data on X and Y coordinate system. It is not only the AMPD method; any peak detection method can also be analyzed with spline algorithm to estimate the right positions.

A. Simulation of the spline algorithm hardware

In this section, a simulation has been performed with input data of the phase-to-phase effective voltage values of a medium-voltage transformer located in the Organized Industrial Zone used for the period from October 1 to October 31, 2015. The corresponding data set contains 4470 data points recorded at 10-min intervals for each L3-L2, L2-L1 and L1-L3 phase-to-phase effective voltages. L1, L2 and L3 denote the power line in order in 3 phase power system. The important point here is that the input peak data used for spline simulation were obtained after implementing the AMPD algorithm. That means inputs of spline algorithm were obtained from previous AMPD algorithm studies (raw-peak data). Fig. 7 illustrates the details of the raw-peak data and fitted-peak data.

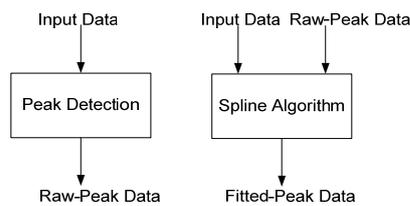


Figure 7. Raw-peak data and fitted-peak data

Daily fitted peak points on hardware design by the spline algorithm method are illustrated in Figs. 8, 9 and 10, for lines L2-L1 (V_{L2-L1}), L3-L2 (V_{L3-L2}) and L1-L3 (V_{L1-L3}) voltage values, respectively. In Figs. 8, 9 and 10, upper traces are showing peak points (raw-peak data) obtained in red color and real fitted data (fitted-peak data) obtained in green color. It can be clearly seen that the difference between two methods are small. Maximum raw peak data, maximum fit peak data and maximum differences between them are given in Table I. Nevertheless, the reason of using spline algorithm is not only to detect the peak points, but also to identify the real position of the peak points obtained. So that in all these graphics given in Fig. 8, Fig. 9 and Fig.

10, lower traces visualize the differences between raw-peak data and fitted-peak data.

TABLE I. MAXIMUM DIFFERENCES BETWEEN RAW-PEAK DATA AND FITTED DATA

Phase-To-Phase Effective Peak Data	Maximum Raw-Peak Data (kV)	Maximum Fitted-Peak Data (kV)	Maximum Differences (kV)
L2-L1 line voltage peak data	35.062500	34.984375	0.078125
L3-L2 line voltage peak data	33.890625	33.828125	0.062500
L1-L3 line voltage peak data	34.187500	34.156250	0.031250

The maximum difference of positive edge is found as 0.078125 kV for L2-L1 phase.

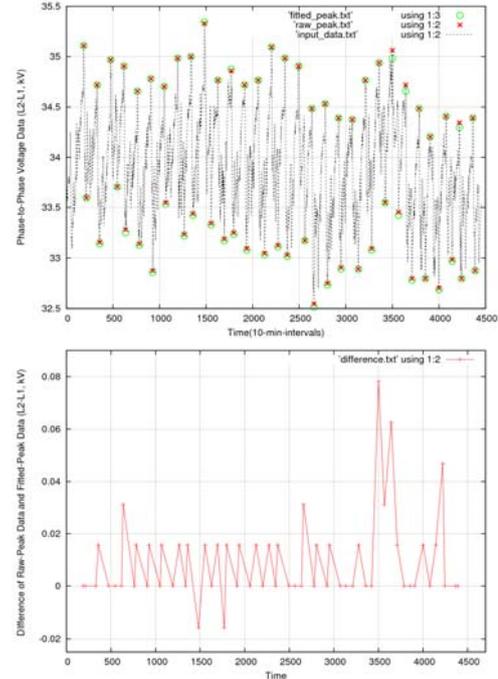


Figure 8. Differences of peak data obtained from raw-peak data and fitted-peak data for L2-L1 line voltage values

Next, this study analyzes the data of MSE and RMSE in detail in order to show the difference between raw-peak data and fitted-peak data. First of all, Mean Absolute Error (MAE) can be defined with the equation (37) given below.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x_i^{\wedge}| \quad (37)$$

Here, n = the number of errors, $|x_i - x_i^{\wedge}|$ = absolute errors.

Secondly, Root Mean Squared Error (RMSE) is analyzed and, then, error ratios are calculated with the equation given below, where, n = the number of errors.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x_i^{\wedge})^2} \quad (38)$$

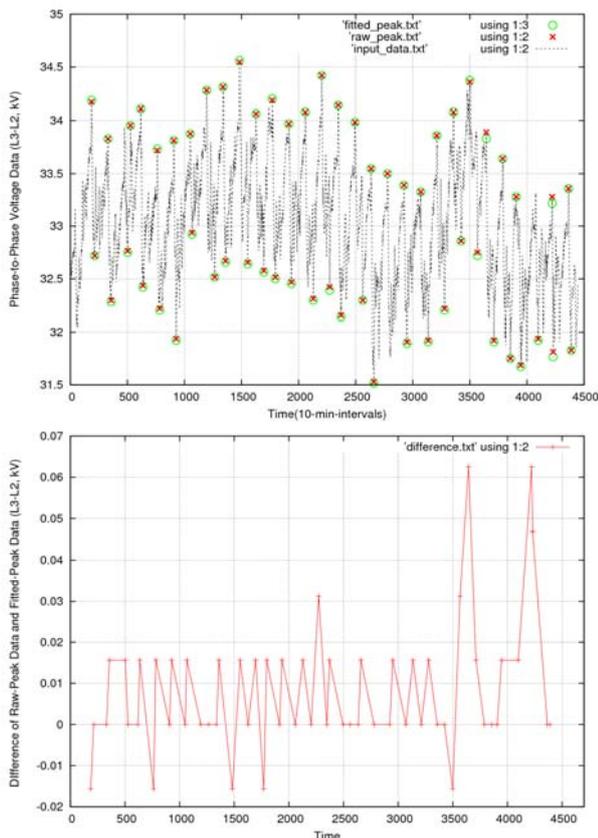


Figure 9. Differences of peak data obtained from raw-peak data and fitted-peak data for L3–L2 line voltage values

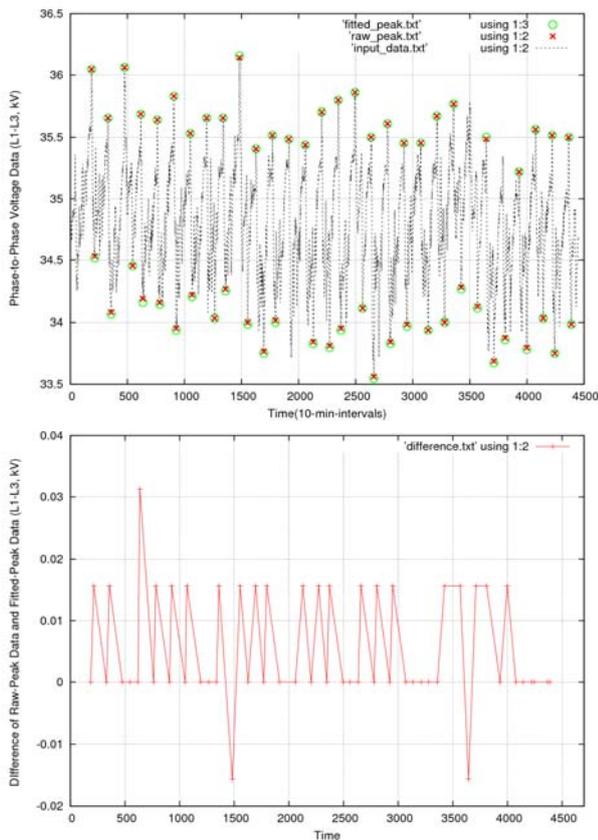


Figure 10. Differences of peak data obtained from raw-peak data and fitted-peak data for L1–L3 line voltage values

Tables II and Table III illustrate results of the MAE and RMSE as 0.00548246 kV and 0.0105528 kV in order for phase voltages of L1_L3.

TABLE II. MEAN ABSOLUTE ERROR FOR ALL PHASE-TO-PHASE VOLTAGE RAW PEAK DATA AND FIT-PEAK DATA

Phase-To-Phase Voltage Data	L1–L2 Phase	L1–L3 Phase	L3–L2 Phase
Mean Absolute Error (MAE)	0.00911458	0.00548246	0.0080819

TABLE III. ROOT MEAN ERROR FOR ALL PHASE-TO-PHASE VOLTAGE RAW PEAK DATA AND FIT-PEAK DATA

Phase-To-Phase Voltage Data	L1–L2 Phase	L1–L3 Phase	L3–L2 Phase
Root Mean Squared Error (RMSE)	0.0183774	0.0105528	0.0176491

B. Evaluation of the spline method with an FPGA board

In this section, the device utilization and performance of the modified algorithm on the Kintex-7 XC7K325T are evaluated. As a mapping tool, a Vivado 2016.3 tool was used. In this study, the maximum system frequency was preferred as 100MHz, and input data width is 12 bit. The system was implemented on the FPGA board. Then we analyzed the latency, memory usage and performance of the implemented system. Table IV shows the resource usage (area). Maximum frequency is 140.865 MHz, and latency is 78.088 ns.

TABLE IV. RESOURCE USAGE

Resource	Utilization	Available	Utilization %
LUT	7168	203800	3.52
FF	3135	407600	0.77
DSP	24	840	2.86
IO	10	500	2.00
Max. Frequency (MHz)	140.865	-	-
Latency (ns)	78.088	-	-

Fig. 11 depicts FPGA card design that includes an FPGA, an AC/DC converter and a function generator. Moreover, Fig. 12 highlights the flow chart about the hardware design.

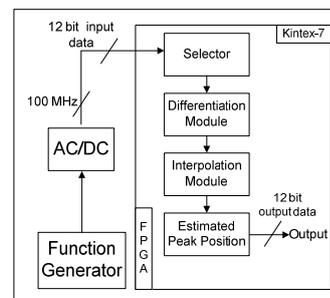


Figure 11. FPGA card design

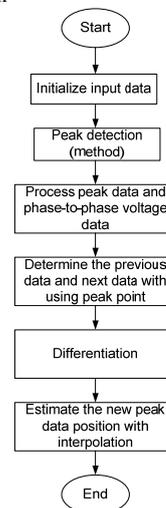


Figure 12. Hardware design flowchart

In this study any specific hardware architecture was not used in order to optimize the use of LUT, BRAM and DSP resource, because, the pipeline was designed to implement on an FPGA to show the high-speed design. Thus, it is not necessary to use any BRAM. Furthermore, in this research, the high frequency signal is considered for real time applications on an FPGA, and it offers short latency time compared to the GPU and CPU. In here, the 100 MHz frequency is preferred because most AC/DC converters are compatible adaptable with 100 MHz.

V. CONCLUSION

In this study, a spline interpolation algorithm was used to detect the peak point detection of a time series signal. As the data, peak points detected previously based on automatic multiscale-based peak detection (AMPD) were used. The study was carried out using hardware implementation and evaluation of the spline method on an FPGA board. Later on, results obtained from spline interpolation and previously obtained peak points were compared for each phase data of the power system. It was observed that peak points calculated from fitted-peak data obtained using spline algorithm and raw-peak data obtained using AMPD algorithm were very close to each other. Differences between AMPD and spline algorithms were then analyzed in detail in order to show sensitivity of spline algorithm. Finally, it can be said that the spline algorithm helped us to identify the real position of the time serial data.

REFERENCES

- [1] H. B. Carithers, "Computer graphics with Open GL", Fourth Edition, My Pearson Store, pp. 336-337, 2011
- [2] L. Mățiu-lovan, "Some aspects of implementing a cubic spline interpolation algorithm on a DSP," 2012 10th International Symposium on Electronics and Telecommunications, Timisoara, 2012, pp. 291-294. doi: 10.1109/ISETC.2012.6408096
- [3] C. Chen and C. Lai, "Iterative linear interpolation based on Fuzzy gradient model for low-cost VLSI implementation," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 7, pp. 1526-1538, July 2014. doi: 10.1109/TVLSI.2013.2276410
- [4] M. Cafaro and P. Pellè, "Space-efficient verifiable secret sharing using polynomial interpolation," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 453-463, 1 April-June 2018. doi: 10.1109/TCC.2015.2396072
- [5] I. Intyas, A. B. Suksmono and A. Munir, "Image quality improvement for GPR acquisition using interpolation method," 2016 22nd Asia-Pacific Conference on Communications (APCC), Yogyakarta, 2016, pp. 371-374. doi: 10.1109/APCC.2016.7581486
- [6] T. Zhan, G. Wei and M. Li, "Two interpolation approximation algorithms and its application in ceramic price forecasting," 2017 International Conference on Smart Grid and Electrical Automation (ICSGEA), Changsha, 2017, pp. 312-314. doi: 10.1109/ICSGEA.2017.132
- [7] M. Lu, L. Huang and Y. Xia, "Two dimensional autoregressive modeling-based interpolation algorithms for image super-resolution: A comparison study," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, 2017, pp. 1-6. doi: 10.1109/CISP-BMEI.2017.8302020
- [8] R. Lazzarotti, T. Pignata and M. Barni, "Piecewise function approximation with private data," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 642-657, March 2016. doi: 10.1109/TIFS.2015.2503268
- [9] M. Kamada and K. Takahashi, "Locally supported bivariate splines in piecewise constant tension," 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEEES), Putrajaya, 2016, pp. 427-432. doi: 10.1109/ICAEEES.2016.7888082
- [10] Z. Feng, H. Liu and R. Hao, "Research on spline interpolation of smoothness of machine tool," 2017 International Conference on Robots & Intelligent System (ICRIS), Huai'an, 2017, pp. 266-269. doi: 10.1109/ICRIS.2017.74
- [11] N. Rokhman, "Solving non-linear equations containing spline interpolation function by relaxing the Newton method," 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, 2017, pp. 1-5. doi: 10.1109/IAC.2017.8280641
- [12] Y. Narukiyo and A. Kubota, "Linear view interpolation from defocused stereo images for multiple-depths scenes," 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, 2018, pp. 1-3. doi: 10.1109/IWAIT.2018.8369740
- [13] Y. Yuan, C. Zhang, Y. Wang, C. Liu, J. Ji and C. Feng, "Linear interpolation process and its influence on the secondary equipment in substations," 2017 China International Electrical and Energy Conference (CIEEC), Beijing, 2017, pp. 205-209. doi: 10.1109/CIEEC.2017.8388447
- [14] M. Saxena and M. K. Gupta, "A novel and fast algorithm for random valued impulse image noise cancellation using cubic B-splines," 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), Shimla, 2013, pp. 421-426. doi: 10.1109/ICIIP.2013.6707627
- [15] D. Sahu, P. Roy and B. K. Das, "Real time implementation of cubic B-spline algorithm for electro optical tracking system," 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), Bangalore, 2013, pp. 1-5. doi: 10.1109/C2SPCA.2013.6749395
- [16] Y. Jiang and Y. Li, "The research of the approximate algorithm based on cubic B-spline curves," 2012 5th ICICS, Liverpool, 2012, pp. 23-26. doi: 10.1109/ICIC.2012.63
- [17] N. F. Law and W. C. Siu, "Fast algorithm for quadratic and cubic spline wavelets," Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489), Hong Kong, China, 2001, pp. 247-250. doi: 10.1109/ISIMP.2001.925380
- [18] D. Popescu, F. Stoican, L. Ichim, "Control and Optimization of UAV Trajectory for Aerial Coverage in Photogrammetry Applications", Advances in Electrical and Computer Engineering, vol.16, no. 3, pp. 99 – 106, 2016. doi:10.4316/AECE.2016.03014
- [19] P. Kerdtuad, S. Kittiratsatcha, "Modeling of a Switched Reluctance Generator Using Cubic Spline Coefficients on the Phase Flux Linkage, Inductance and Torque Equations", Advances in Electrical and Computer Engineering, vol. 15, no. 1, pp. 41 – 48, 2015. doi:10.4316/AECE.2015.01006
- [20] R. D. Vatavu, S. G. Pentiuc, L. Grisoni, C. Chaillou, "Modeling Shapes for Pattern Recognition: A Simple Low-Cost Spline-based Approach", Advances in Electrical and Computer Engineering, vol.8, no. 1, pp. 67 – 71, 2008. doi:10.4316/AECE.2008.01012
- [21] D. Bailey, S. Randhawa and J. S. J. Li, "Advanced bayer demosaicing on FPGAs," 2015 International Conference on Field Programmable Technology (FPT), Queenstown, 2015, pp. 216-220. doi: 10.1109/FPT.2015.7393154
- [22] I. O. H. Fuentes, M. E. B. Zanoguera and G. G. Yanez, "FPGA implementation of the bilinear interpolation algorithm for image demosaicking," 2009 International Conference on Electrical, Communications, and Computers, Cholula, Puebla, 2009, pp. 25-28. doi: 10.1109/CONIELECOMP.2009.47
- [23] K. S. Rani and W. J. Hans, "FPGA implementation of bilinear interpolation algorithm for CFA demosaicing," 2013 International Conference on Communication and Signal Processing, Melmaruvathur, 2013, pp. 857-863. doi: 10.1109/iccsp.2013.6577178
- [24] A. E. Desjardins, B. J. Vakoc, M. J. Suter, S. Yun, G. J. Tearney and B. E. Bouma, "Real-time FPGA processing for high-speed optical frequency domain imaging," in IEEE Transactions on Medical Imaging, vol. 28, no. 9, pp. 1468-1472, Sept. 2009. doi: 10.1109/TMI.2009.2017740
- [25] A. M. Colak, T. Manabe, Y. Shibata and F. Kurokawa, "Peak detection implementation for real-time signal analysis based on FPGA," Circuit and Systems, vol. 9, pp. 148-167, 2018. doi.org/10.4236/cs.2018.910016
- [26] J. Betten, "Comparison between Lagrange and spline interpolation," RWTH Aachen University RWTH Aachen University, Mathematical Models in Materials Science and Continuum Mechanics, 2014
- [27] S. C. Chapra and R. P. Canale, "Numerical methods for engineers", Seventh Edition, MacGraw-Hill Education, ISBN: 978-0-07-339792-4, 2015
- [28] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical recipes. The Art of Scientific Computing", Third Edition, Cambridge University Press, ISBN: 978-0-521-88068-8, 2007