Strain gage balance signal filtering with piecewise representations

Mihai Victor PRICOP*[,1], Sarif OMAR[1], Ionut BUNESCU[1],
Alexandru Marius PANAIT[1]

*Corresponding author
[1]INCAS – National Institute for Aerospace Research "Elie Carafoli",
B-dul Iuliu Maniu 220, Bucharest 061126, Romania,
pricop.victor@incas.ro*

DOI: 10.13111/2066-8201.2019.11.2.10

Received: 28 April 2019/ Accepted: 19 May 2019/ Published: June 2019

7th International Workshop on Numerical Modelling in Aerospace Sciences "NMAS 2019"
15-16 May 2019, Bucharest, Romania, (held at INCAS, B-dul Iuliu Maniu 220, sector 6)
Section 2 – Flight dynamics simulation

Abstract: *Strain gage balances provide the most important output of the supersonic wind tunnels. Typically, the time histories of the six forces/moments components must be filtered out, to provide useful results for performance assessment of the models. Various filtering models do exist, from basic regressions, to frequency domain and combinations of piecewise fit polynomials. Results with decoupled piecewise linear fit (moving window) showed the need of more suitable methods, namely piecewise, that must properly connect the neighboring fit curves, to provide continuity up to the first derivative. A polygonal line fit is derived and modified for increased smoothness. A third order cubic spline is derived, offering connectivity up to the first derivative. A fifth order spline is derived with continuity up to the second order derivative. Filtering results are shown for real and synthetic sets of data with conclusions.*

Key Words: *strain gage balance, filtering, polygonal line, spline fit*

1. INTRODUCTION

Strain gauge balances for wind tunnels in quasi-static conditions require dedicated signal acquisition and processing. The first stage considers setting the optimal intrinsic filtering and sampling rate according to the physics. Another step is numerical filtering and data reduction, according to the aerodynamics practice and this constitutes the topic of the current work. Single or standard function polynomial regression is of limited use because data may be sometimes too complex. The simple connection of more polynomials, under various classes of continuity is a very strong instrument, enabling the reconstruction of virtually all types of data in experimental physics.

There is a long standing tradition in scientific data interpolation in the form of smooth curves, if only to present a general visual representation of experimental data in a more natural format whenever only discrete data points are available, and perhaps the most used method is the least squares fit using curves chosen so that they are somehow conform to local data trends

[1]. Complementary data smoothing methods include Fourier transformations, polynomial fitting and carious convolution operations that have been found to ease data fitting using splines by reducing data noise [2]. A special consideration is given to the avoidance of data fitting to noise not to actual data trend whenever very noisy signals appear [1]. The data points used as well as the first and second derivatives of the approximation function are optimizable parameters. The fitness of the approximation can be probed in a number of ways, for example evaluating the weighted mean-square residual error of the spline approximation, treated as an objective function to be minimized. There is a method of constraining this optimization problem, using various criteria such as having a term proportional to the mean of the curvature of the spline in the objective function [1], and various methods to shape the spline using control points in a robust and not overly resource consuming way for large data sets using a progressive and iterative approximation method for the least squares fit (LSPIA) [3]. There are other methods for spline fitting, using for example the relatively recent and innovative firefly algorithm to trace an optimum B-spline at minimum square distance from existing data points [4]. However, none of these works used a direct computational approach in the sense of explicitly solving an equation system to join the existing data segments with functions that are continuous and derivable (at least once, for third order approximation and at least twice for the fifth order splines). This is the approach used in this work, and for simplicity and briefness we will initially assume the knots (control points for the spline) to be uniformly distributed. Optimizing the knots distribution is a complex problem in itself, that can be tackled in various ways, from observing data trends and gradients to the more exotic ones as the iterative evolution of the several fitness functions used in the firefly algorithm [4].

The third order spline fit methods are well reviewed in [5] where the 3rd order spline fit method is commended for obtaining first and second order continuous derivatives, an important aspect since many parameters evaluated in experimental aerodynamics are themselves or use these derivatives. The smoothness of the resulted curves is therefore of prime importance to the evaluation of performance of the tested objects on the entire operational envelope. The algorithms discussed are also flexible by using parametric placement of knots for the spline fit, allowing for local optimization.

Other fitting strategies such as the Zernicke method were investigated and found inferior to B-splines for complex wave fronts [6] such as is the usual case in experimental aerodynamics, observed from datasets collected during the years of operations at the Trisonic Wind Tunnel.

For the strategies using Bspline fit, the problem of choosing the number and location of knots is of utmost importance, as the number and position of the knots greatly influence the fit quality for the B-spline approximators. However, an arbitrarily large number of knots complicates computation and consumes resources unnecesarily without any improvement of the fit. Therefore adaptive methods appeared [7] to help with automatic knot selection, [8] resulting in a sparse spline regression with a reduced number of points but keeping a good fit, comparable to penalized spline regression results.

2. CONTINUOUS PIECEWISE LINEAR FIT – POLYGONAL FIT

The classical moving window method based on regression lines has as greatest limitation the lack of continuity, generating artificial noise. The idea of trying to connect the piecewise elements is straightforward, although at the expense of solving a linear system. Given the high price of the data obtained in an industrial supersonic wind tunnel facility, it makes sense to use relatively complex methods for the upper level of data filtering. The derivation of the polygonal line reconstruction is straight, using the standard least squares procedure. The resulted linear system can adequately be solved with the method of Thomas.

The signal is constituted by $(x_k, y_k), k = \overline{1, n}$. The purpose is to produce *nf* filtered values $\overline{y}_k$ in the knots $\overline{x}_k$ provided by the user, but typically equally distributed. Sums in equations (2) - (4) are applied for k index, upon each filtering subinterval $[x_{i-1}, x_i]$. Equation (2) corresponds to the left knot, equation (3) is for the inner knots and equation (4) is for the right knot.

$$s_k = \frac{x_k - \overline{x}_i}{\overline{x}_{i+1} - \overline{x}_i} \tag{1}$$

$$\overline{y}_1 \sum\nolimits_1 (1 - s_k)^2 + \overline{y}_2 \sum\nolimits_1 (1 - s_k) s_k = \sum\nolimits_1 (1 - s_k) y_k \tag{2}$$

$$\begin{aligned} \overline{y}_{i-1} \sum\nolimits_{i-1} (1 - s_k) s_k + \overline{y}_i \left(\sum\nolimits_{i-1} s_k^2 + \sum\nolimits_i (1 - s_k)^2 \right) + \overline{y}_{i+1} \sum\nolimits_i (1 - s_k) s_k \\ = \sum\nolimits_{i-1} s_k y_k + \sum\nolimits_i (1 - s_k) y_k \end{aligned} \tag{3}$$

$$\overline{y}_{nf-1} \sum\nolimits_{nf-1} (1 - s_k) s_k + \overline{y}_{nf} \sum\nolimits_{nf-1} s_k^2 = \sum\nolimits_{nf-1} s_k y_k \tag{4}$$

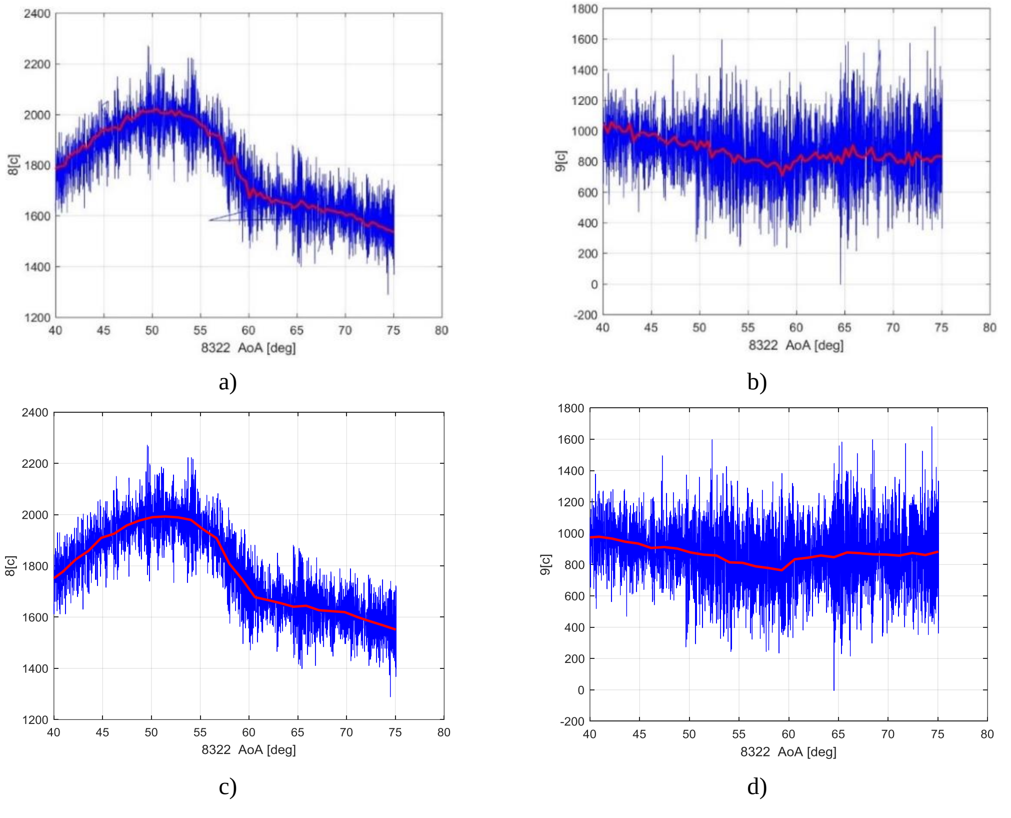


Fig. 1 Moving window regression lines with 100 knots a) & b), polygonal fit with 30 knots c) & d) for identical datasets

A comparison between moving window regression lines and polygonal fit is in Fig. 1. A typical requirement is to extract about 100 filtered values from a time series, while decades ago, 20-25 values were considered satisfactorily. We may conclude that the polygonal fit requires less knots for better results and slope discontinuities are handled well. Convergence with respect to the signal size (from 5000 to 5 million data points) is shown in Fig. 2. This artificial signal is generated by perturbing sine function with random numbers. It is obvious that the noise related to the peaks is reducing when increasing the size of the signal. The procedure was tested for up to 50 million signal points, requiring a couple of seconds to run.

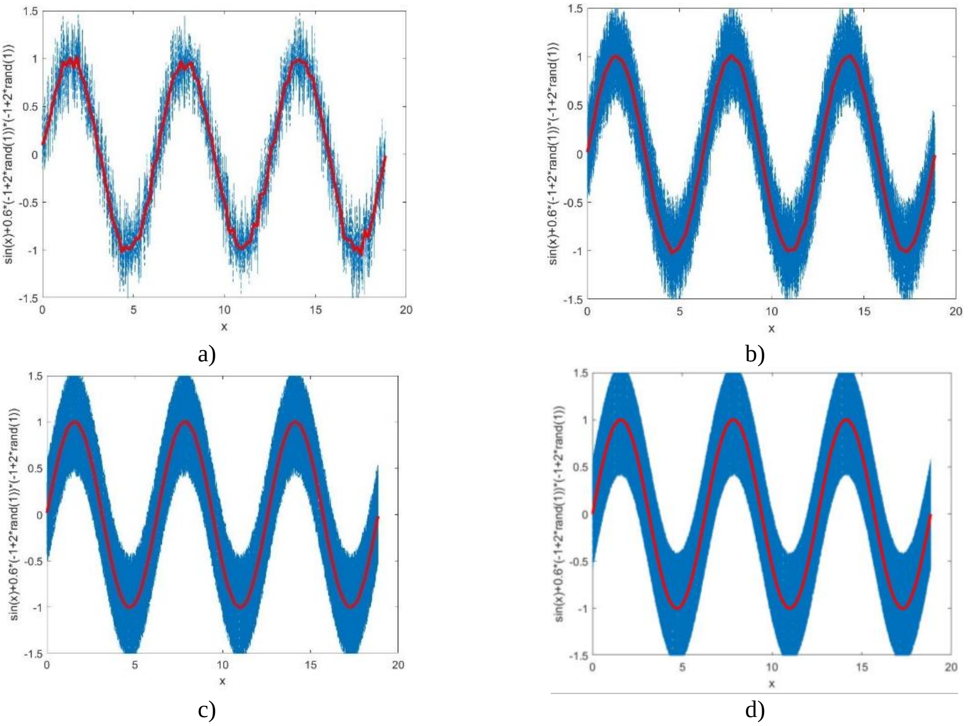


Fig. 2 Polygonal line filtering with 100 points for perturbed sinus function, signal size 5000 a), 50000 b), 500000 c), 5000000 d)

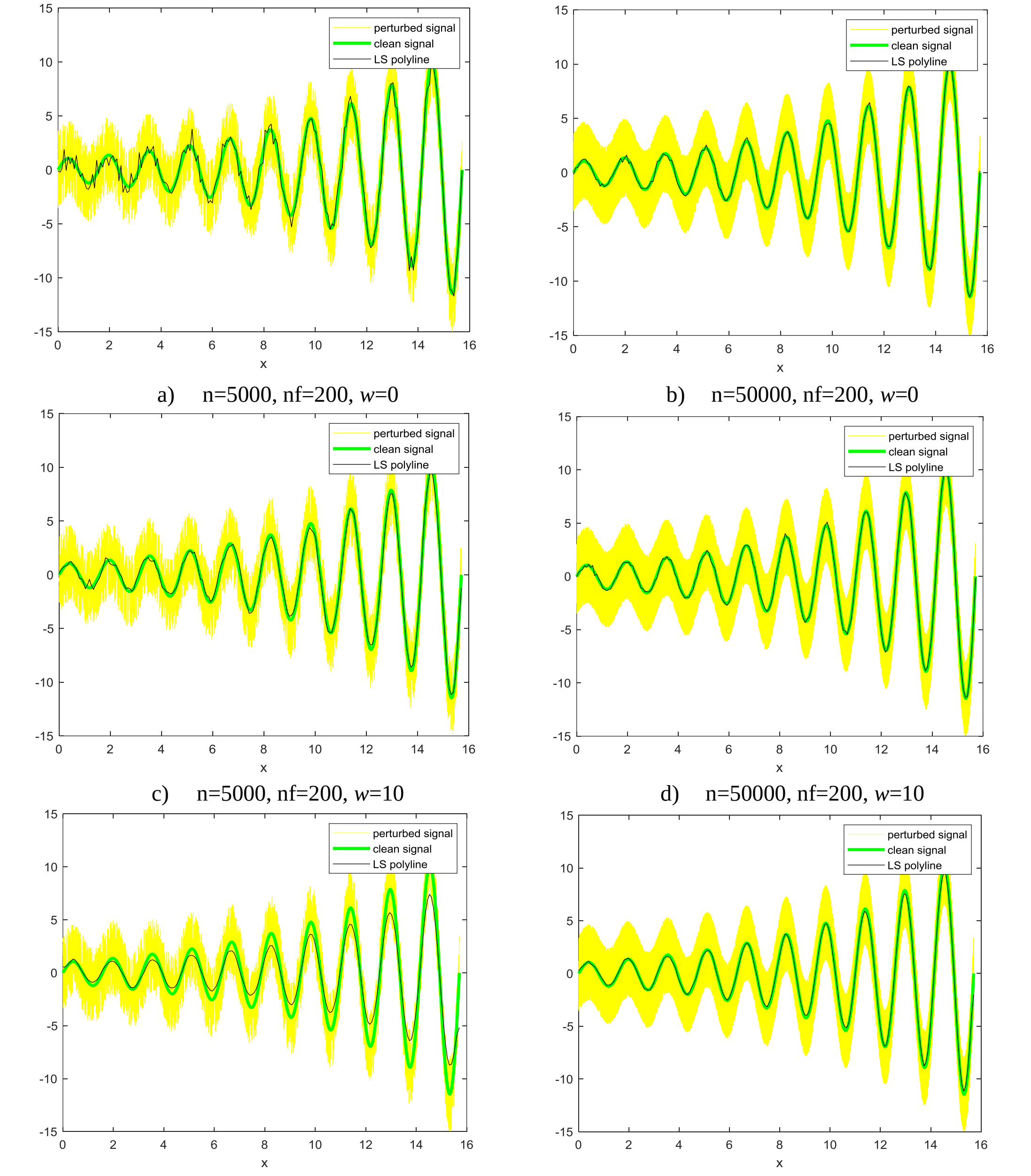
Smoother polygonal lines may be built by modifying the least square formulation in such a way that the sum of squared lengths of the line segments is added to the objective function. When multiplying the total length with the weight w, the tridiagonal system takes the new form as in equations (5), (6) and (7).

$$\overline{y}_1\left(w+\sum\nolimits_1(1-s_k)^2\right)+\overline{y}_2\left(w+\sum\nolimits_1(1-s_k)s_k\right)=\sum\nolimits_1(1-s_k)y_k \tag{5}$$

$$\begin{aligned}\overline{y}_{i-1}\left(\sum\nolimits_{i-1}((1-s_k)s_k)-w\right)+\overline{y}_i\left(2w+\sum\nolimits_{i-1}s_k^2+\sum\nolimits_i(1-s_k)^2\right)\\+\overline{y}_{i+1}\left(\sum\nolimits_i(1-s_k)s_k-w\right)=\sum\nolimits_{i-1}s_ky_k+\sum\nolimits_i(1-s_k)y_k\end{aligned} \tag{6}$$

$$\overline{y}_{nf-1}\left(\sum\nolimits_{nf-1}((1-s_k)s_k)-w\right)+\overline{y}_{nf}\left(w+\sum\nolimits_{nf-1}s_k^2\right)=\sum\nolimits_{nf-1}s_ky_k \tag{7}$$

In the modified polyline fit formulation, the inner points equations contain an additional left-hand side term suggesting numerical diffusion: $-w(\overline{y}_{i+1}-2\overline{y}_i+\overline{y}_{i-1})$. When applying the method to cvasiperiodical signals with multiple time averged peaks, the filtered values are dampen out proportional with the value of the weight w, see Fig. 3.



a) n=5000, nf=200, *w*=0 b) n=50000, nf=200, *w*=0

c) n=5000, nf=200, *w*=10 d) n=50000, nf=200, *w*=10

e) n=5000, nf=200, *w*=100 f) n=50000, nf=200, *w*=100

Fig. 3 Effect of smoothing weigh and signal size upon the polygonal line

3. THIRD ORDER SPLINE FIT

The foundation of these spline approximation methods is placed by Carl de Boor and John R. Rice in their seminal paper [9], dealing with a fixed knots variant of the least squares cubic spline approximation method. Here the accent is placed on driving a best-fit spline with a prior fixed number of knots through a data point scatter using the least squares metric optimization. This simple algorithm does not impose the number or placement of knots but lets the user find the appropriate values to these parameters, but the work clearly states the crucial importance of these two parameters on approximation fit quality and computational ease. Historically the first attempts to standardize the procedure of knot finding and control by a computational method are described in [10] where simultaneous approximation of points and normal vectors associated to them followed by successive iterations adjusting the field of normal vectors leads to a simple and computationally efficient method for planar curve reconstruction from noisy data represented by a scattered cloud of data points. The method is generalized for 3D space use, but a trivial 2D method is derived and exemplified. Both the parametric and the implicit (algebraic) curve fitting problems can be thus reduced to solving sequences of systems of linear equations.

The current version of cubic splines considers the knot values and slopes as optimization variables. From the standard least squares formulation, we provide the resulting linear system of equations. Thus, equation (8) shows the relation between the polynomial coefficients and the knot filtered m_1, m_3 and slopes m_3, m_4, that all have to be optimized. Polynomials α, β, γ and δ are introduced in equation (9). They are used to compute the components of vector f, introduced in equations (10) and (11), that are combined to become elements of linear system' matrix. First index of f refers to the subinterval upon that all sums are computed.

$$\begin{bmatrix} 1 & x_i & x_i^2 & x_i^3 \\ 0 & 1 & 2x_{i+1} & 3x_i^2 \\ 1 & x_{i+1} & x_{i+1}^2 & x_{i+1}^3 \\ 0 & 1 & 2x_{i+1} & 3x_{i+1}^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = Mc = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} = M^{-1^T} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \tag{9}$$

$\sum_j \alpha_k^2$	$\sum_j \alpha_k\beta_k$	$\sum_j \alpha_k\gamma_k$	$\sum_j \alpha_k\delta_k$	$\sum_j \beta_k^2$	$\sum_j \beta_k\gamma_k$	$\sum_j \beta_k\delta_k$
$f_{j,1}$	$f_{j,2}$	$f_{j,3}$	$f_{j,4}$	$f_{j,5}$	$f_{j,6}$	$f_{j,7}$

(10)

$\sum_j \gamma_k^2$	$\sum_j \gamma_k\delta_k$	$\sum_j \delta_k^2$	$\sum_j \alpha_k y_k$	$\sum_j \beta_k y_k$	$\sum_j \gamma_k y_k$	$\sum_j \delta_k y_k$
$f_{j,8}$	$f_{j,9}$	$f_{j,10}$	$f_{j,11}$	$f_{j,12}$	$f_{j,13}$	$f_{j,14}$

(11)

First knot equations (12) are particular, as well as for the last knot (15), that closes the linear system. Inner knots equations stand for $i = \overline{3: 2nf - 2}$ as in equations (13) and (14). From the implementation point of view, the matrix is stored in a sparse format with bandwith equal to six and the solver is chosen accordingly. The unknown vector is stored in the order: filtered value, slope value, for all knots and therefore equation (8) is used to compute the local polynomial coefficients, using the stored inverted matrices.

$$\begin{aligned} m_1 f_{1,1} + m_2 f_{1,2} + m_3 f_{1,3} + m_4 f_{1,4} &= f_{1,11} \\ m_1 f_{1,2} + m_2 f_{1,5} + m_3 f_{1,6} + m_4 f_{1,7} &= f_{1,12} \end{aligned} \tag{12}$$

$$j = (i-1)/2\,, i\ odd$$

$$\begin{aligned} m_{i-2} f_{j,3} + m_{i-1} f_{j,6} + m_i (f_{j,8} + f_{j+1,1}) + m_{i+1} (f_{j,9} + f_{j+1,2}) + m_{i+2} f_{j+1,3} \\ + m_{i+3} f_{j+1,4} = f_{j+1,3} + f_{j+1,11} \end{aligned} \tag{13}$$

$$j = (i-1)/2\,, i\ even$$

$$m_{i-3}f_{j,4} + m_{i-2}f_{j,7} + m_{i-1}(f_{j,9} + f_{j+1,2}) + m_i(f_{j,10} + f_{j+1,5}) + m_{i+1}f_{j+1,6} + m_{i+2}f_{j+1,7} = f_{j+1,4} + f_{j+1,12} \tag{14}$$

$$\begin{gathered} m_{2nf-3,}f_{nf-1,3} + m_{2nf-2}f_{nf-1,6} + m_{2nf-1}f_{nf-1,8} + m_{2nf}f_{nf-1,9} = f_{nf-1,13} \\ m_{2nf-3}f_{nf-1,4} + m_{2nf-2}f_{nf-1,7} + m_{2nf-1}f_{nf-1,9} + m_{2nf}f_{nf-1,10} = f_{nf-1,14} \end{gathered} \tag{15}$$

4. FIFTH ORDER SPLINE FIT

This method comes as an attempt to increase the accuracy of the third order, after implementing it with a reasonable effort. The same formalism as in the previous case stands, this time adding the second order knot derivatives as optimization variables: m_3 and m_6 for each subinterval, equation (16). The number of subinterval polynomials goes from 5 to 7, as in equation (17). Also the number of polynomial combination sums f is growing from 14 to 27 for each subinterval, equations (18) - (21). The effort to derive the linear system plus the implementation was reasonable, given the experience gathered with the third order spline.

$$\begin{bmatrix} 1 & x_i & x_i^2 & x_i^3 & x_i^4 & x_i^5 \\ 0 & 1 & 2x_i & 3x_i^2 & 4x_i^3 & 5x_i^4 \\ 0 & 0 & 2 & 6x_i & 12x_i^2 & 20x_i^3 \\ 1 & x_{i+1} & x_{i+1}^2 & x_{i+1}^3 & x_{i+1}^4 & x_{i+1}^5 \\ 0 & 1 & 2x_{i+1} & 3x_{i+1}^2 & 4x_{i+1}^3 & 5x_{i+1}^4 \\ 0 & 0 & 2 & 6x_{i+1} & 12x_{i+1}^2 & 20x_{i+1}^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = Mc = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \zeta \end{bmatrix} = M^{-1^T} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \\ x^5 \end{bmatrix} \tag{17}$$

(18)

$\sum_j \alpha_k^2$	$\sum_j \alpha_k\beta_k$	$\sum_j \alpha_k\gamma_k$	$\sum_j \alpha_k\delta_k$	$\sum_j \alpha_k\epsilon_k$	$\sum_j \alpha_k\zeta_k$	$\sum_j \beta_k^2$
$f_{j,1}$	$f_{j,2}$	$f_{j,3}$	$f_{j,4}$	$f_{j,5}$	$f_{j,6}$	$f_{j,7}$

(19)

$\sum_j \beta_k\gamma_k$	$\sum_j \beta_k\delta_k$	$\sum_j \beta_k\epsilon_k$	$\sum_j \beta_k\zeta_k$	$\sum_j \gamma_k^2$	$\sum_j \gamma_k\delta_k$	$\sum_j \delta_k\epsilon_k$
$f_{j,8}$	$f_{j,9}$	$f_{j,10}$	$f_{j,11}$	$f_{j,12}$	$f_{j,13}$	$f_{j,14}$

(20)

$\sum_j \gamma_k\zeta_k$	$\sum_j \delta_k^2$	$\sum_j \delta_k\epsilon_k$	$\sum_j \delta_k\zeta_k$	$\sum_j \epsilon_k^2$	$\sum_j \epsilon_k\zeta_k$	$\sum_j \zeta_k^2$
$f_{j,15}$	$f_{j,16}$	$f_{j,17}$	$f_{j,18}$	$f_{j,19}$	$f_{j,20}$	$f_{j,21}$

(21)

$\sum_j \alpha_k y_k$	$\sum_j \beta_k y_k$	$\sum_j \gamma_k y_k$	$\sum_j \delta_k y_k$	$\sum_j \epsilon_k y_k$	$\sum_j \zeta_k y_k$
$f_{j,22}$	$f_{j,23}$	$f_{j,24}$	$f_{j,25}$	$f_{j,26}$	$f_{j,27}$

First knot equations (22) are particular, as well as for the last knot (26), that closes the linear system. Inner knots equations stand for $i = \overline{4\colon 2nf-3}$ as in equations (23), (24) and (25). The unknown vector is stored in the order: filtered value, slope value, second derivative, for all knots and therefore equation (16) is used to compute the local polynomial coefficients, using the stored inverted matrices. Bandwith is increased from 6 in the case of cubic spline, to 9 in the case of fith order spline.

$$
\begin{aligned}
m_1 f_{1,1} + m_2 f_{1,2} + m_3 f_{1,3} + m_4 f_{1,4} + m_5 f_{1,5} + m_6 f_{1,6} &= f_{1,22} \\
m_1 f_{1,2} + m_2 f_{1,7} + m_3 f_{1,8} + m_4 f_{1,9} + m_5 f_{1,10} + m_6 f_{1,11} &= f_{1,23} \\
m_1 f_{1,3} + m_2 f_{1,8} + m_3 f_{1,12} + m_4 f_{1,13} + m_5 f_{1,14} + m_6 f_{1,15} &= f_{1,24}
\end{aligned}
\tag{22}
$$

Inner subinterval equations stand for $j = 2\colon nf - 1$.

$$
\begin{aligned}
&i = 3(j-1)+1 \\
&m_{i-3} f_{j-1,4} + m_{i-2} f_{j-1,9} + m_{i-1} f_{j-1,13} + m_i\left(f_{j-1,16} + f_{j,1}\right) \\
&\quad + m_{i+1}\left(f_{j-1,17} + f_{j,2}\right) + m_{i+2}\left(f_{j-1,18} + f_{j,3}\right) + m_{i+3} f_{j,4} \\
&\quad + m_{i+4} f_{j,5} + m_{i+5} f_{j,6} = f_{j-1,25} + f_{j,22}
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
&i = 3(j-1)+2 \\
&m_{i-4} f_{j-1,5} + m_{i-3} f_{j-1,10} + m_{i-2} f_{j-1,14} + m_{i-1}\left(f_{j-1,17} + f_{j,2}\right) \\
&\quad + m_i\left(f_{j-1,19} + f_{j,7}\right) + m_{i+1}\left(f_{j-1,20} + f_{j,8}\right) + m_{i+2} f_{j,9} \\
&\quad + m_{i+3} f_{j,10} + m_{i+4} f_{j,11} = f_{j-1,26} + f_{j,23}
\end{aligned}
\tag{24}
$$

$$
\begin{aligned}
&i = 3(j-1)+3 \\
&m_{i-5} f_{j-1,6} + m_{i-4} f_{j-1,11} + m_{i-3} f_{j-1,15} + m_{i-2}\left(f_{j-1,18} + f_{j,3}\right) \\
&\quad + m_{i-1}\left(f_{j-1,20} + f_{j,8}\right) + m_i\left(f_{j-1,21} + f_{j,12}\right) + m_{i+1} f_{j,13} \\
&\quad + m_{i+2} f_{j,14} + m_{i+3} f_{j,15} = f_{j-1,27} + f_{j,24}
\end{aligned}
\tag{25}
$$

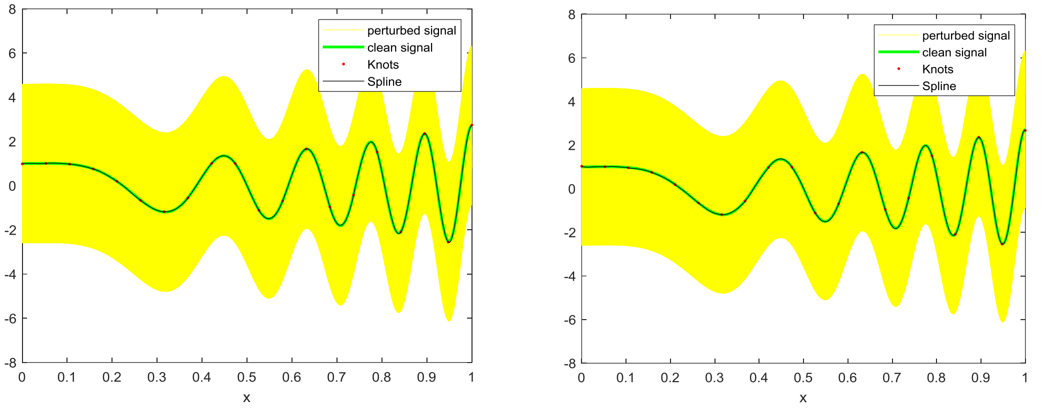
$$
\begin{aligned}
&j = nf - 1;\ i = 3nf - 2 \\
&m_{i-3} f_{j,4} + m_{i-2} f_{j,9} + m_{i-1} f_{j,13} + m_i f_{j,16} + m_{i+1} f_{1,17} + m_{i+2} f_{1,18} = f_{1,25} \\
&i = 3nf - 1 \\
&m_{i-4} f_{j,5} + m_{i-3} f_{j,10} + m_{i-2} f_{j,14} + m_{i-1} f_{j,17} + m_i f_{1,19} + m_{i+1} f_{1,20} = f_{1,26} \\
&i = 3nf \\
&m_{i-5} f_{j,6} + m_{i-4} f_{j,11} + m_{i-3} f_{j,15} + m_{i-2} f_{j,18} + m_{i-1} f_{1,20} + m_i f_{1,21} = f_{1,27}
\end{aligned}
\tag{26}
$$

5. CONCLUSIONS

The polyline fit in its modified form is probably the most effective method and the number of knots and weight w are easiest to set. Convergence with signal size is the fastest. Third order spline is more complex in picking up the number of knots, althouth it is the only parameter to set up. Oscilations may occur if number of knots is either too large or too small. Fith order spline is the most sensitive of all methods. Splines require one order of magnitude less knots than the polygonal fit Fig. 4. Convergence of data reconstructions with splines shows slower rate. The absolute accuracy level of splines can reach superior levels in comparison with the polygonal fit, but some efort is required to properly set the number of knots.

Given the fact that the typical signals aquired with the strain gauge balances in high speed wind tunnels are presenting much simpler shapes Fig. 1 than the benchmarking cases from

Fig. 2 - Fig. 3, we consider that the developed tools are more than adequate, in the following hierarchy: modified polygonal fit, third order spline fit and fifth order spline fit. Future work will focus on optimal knot positioning.



a) b)

Fig. 4 Third order spline a), Fifth order spline b), 20 knots, 2.5 million signal points

REFERENCES

[1] B. J. Thijsse, M. A. Hollanders and J. Hendrickse, A Practical Algorithm for least-squares spline approximation of data containing noise, *Computers In Physics*, vol. **12**, nr. 4, 1998.

[2] M. H. Zhu, L. Liu, Z. You and A. Xu, *Least Squares Fitting of Low-Level Gamma-Ray Spectra with B-Spline Basis Functions*, Space Exploration Technology Laboratory, Macao University of Science and Technology, Macao.

[3] Deng and H. Lin, Progressive and iterative approximation for least squares B-Spline curve and surface fitting, *Computer Aided Design*, nr. 47, pp. 32-44, 2014.

[4] A. Galvez and A. Iglesias, Firefly Algorithm for Explicit B-Spline Curve Fitting to Data Points, *Mathematical Problems in Engineering*, 2013.

[5] R. E. Smith Jr; J. M. Price, L. M. Houser, *A Smoothing Algorithm Using Cubic Spline Functions*, Langley Research Center, NASA Technical Note TN D-7397, Washington D.C., 1974.

[6] M. Ares, S. Royo, J. Caum, C. Pizarro, *Comparison of B-Spline and Zernike fitting techniques in complex wavefront surfaces*, International Society For Optics and Photonics, 2005.

[7] V. Goepp, O. Bouaziz and G. Nuel, *Spline Regression with Automatic Knot Selection*, HAL archives-ouvertes.fr, 2018.

[8] H. Kang, F. Chen, Y. Li, J. Deng and Z. Yang, Knot Calculation for spline fitting via sparse optimization, *Computer Aided Design*, nr. 58, pp. 179-188, 2015.

[9] C. de Boor and J. R. Rice, *Least Squares Cubic Spline Approximation I - Fixed Knots*, Department of Computer Science Technical Reports, 1968.

[10] B. Juttler, *Least-squares fitting of algebraic spline curves via normal vector estimation*, Darmstadt University of Technololgy, Dept. of Mathematics, Schlossgartenstrasse 7, 64289 Darmstadt,Germany.