Online Aerodynamic Model Identification using a Recursive Sequential Method for Multivariate Splines

L. G. Sun[1] and C. C. de Visser[2], Q. P. Chu[3], and J. A. Mulder[4]
Delft University of Technology, Delft, 2600 GB, The Netherlands

Avoiding high computational loads is essential to online aerodynamic model identification algorithms, which are at the heart of any model-based adaptive flight control system. Multivariate simplex B-spline (MVSB) methods are excellent function approximation tools for modeling the nonlinear aerodynamics of high performance aircraft. However, the computational efficiency of the MVSB method must be improved in order to enable real-time onboard applications, for example in adaptive nonlinear flight control systems. In this paper, a new recursive sequential identification strategy is proposed for the MVSB method aimed at increasing its computational efficiency, thereby allowing its use in onboard system identification applications. The main contribution of this new method is a significant reduction of computational load for large scale online identification problems as compared to the existing MVSB methods. The proposed method consists of two sequential steps for each time interval, and makes use of a decomposition of the global problem domain into a number of subdomains, called modules. In the first step the B-coefficients for each module are estimated using a least squares estimator. In the second step the local B-coefficients for each module are then smoothened into a single global B-coefficient vector using a linear minimum mean

---

[1] Ph.D. Student, Control and Simulation Devision, Faculty of Aerospace Engineering, P.O.Box 5058; l.sun@tudelft.nl, Student Member AIAA.
[2] Assistant Professor, Control and Simulation Devision, Faculty of Aerospace Engineering, P.O.Box 5058; C.C.deVisser@tudelft.nl, Member AIAA.
[3] Associate Professor, Control and Simulation Devision, Faculty of Aerospace Engineering, P.O.Box 5058; q.p.chu@tudelft.nl, Member AIAA.
[4] Professor, Control and Simulation Devision, Faculty of Aerospace Engineering, P.O.Box 5058; j.a.mulder@tudelft.nl, Member AIAA.

square errors (LMMSE) estimation. The new method is compared to existing batch and recursive MVSB methods in a numerical experiment in which an aerodynamic model is recursively identified based on data from an NASA F-16 wind-tunnel model.

Nomenclature

$b(\mathbf{x})$	=	barycentric coordinate vector of point $\mathbf{x}$
C	=	covariance matrix
C_m	=	dimensionless pitch moment coefficient
C_X, C_Z	=	dimensionless force coefficient in the X or Z direction
$\bar{c}$	=	mean aerodynamic chord, m
$\mathbf{c}$	=	B-coefficient vector
$\tilde{\mathbf{c}}$	=	B-coefficient vector derived using the recursive sequential MVSB method
d	=	polynomial order
$\hat{d}$	=	spline basis number within each simplex
E	=	the number of edges in a triangulation
$\mathbf{H}$	=	smoothness constraint matrix
J	=	simplex number in a triangulation
m	=	smoothness order of spline function
n	=	dimension of function input space
$P(\cdot)$	=	polynomial function
q	=	pitch rate around the body Y axis, rad/s
R	=	the number of continuity conditions per edge
$S(\cdot)$	=	spline function
V	=	airspeed, m/s
$\mathbf{X}(\cdot)$	=	regression matrix
α, β	=	angle of attack, angle of sideslip, rad
δ_e, δ_{lef}	=	elevator deflection, deflection of leading edge flap, degree
$\mathcal{J}$	=	cost function
$\mathcal{T}$	=	triangulation
κ	=	a multi-index on barycentric coordinates
γ	=	a multi-index independent of κ
ϵ	=	residual vector
ε	=	random variable vector
χ_{2d}, χ_{3d}	=	a dataset in 2 dimensions and 3 dimensions separately

Subscripts

t_i	=	the i^{th} simplex

Superscripts

$\tilde{}$	=	smooth vector derived using the new recursive sequential spline method
$\hat{}$	=	smooth vector derived using the batch spline method

I. Introduction

In the implementation of any model-based adaptive flight control system, like for example nonlinear dynamic inversion control [1] and modular adaptive backstepping control [2][3], it is critical to at all times maintain an accurate onboard model of the aircraft under control. Central to such a control system is a recursive system identification loop, which constantly updates the onboard aerodynamic model as new measurements become available [4][5]. It is essential that this loop is computationally efficient, as onboard computational capabilities in most applications are severely limited. In the past, the field of recursive system identification has been well studied, like for example variants of recursive least square methods and maximum likelihood method, see e.g. [6] [7]and [8]. More recently, it has been suggested that recursive identification methods are developing towards the direction of nonlinear recursive identification methods [9] [10] [11] [12] [13] [14] [15].

The field of recursive identification can be split into parametric methods and nonparametric methods. However, this paper will only focus on the parametric methods. A classical online parametric identification method is the ordinary polynomial basis based (OPBB) method. This method has been intensively studied by Klein, Morelli et al. [16] [17][18] [19][20]. The model structure selection problem is solved by ranking the polynomial terms according to their effect factors using the orthogonal least squares methods[19][18].

A recent promising parametric identification method is the multivariate simplex B-splines (MVSB) method [21] [22] [23] [13]. In [23] a new identification method was introduced which used the B-form basis polynomials of multivariate simplex B-splines in a linear regression framework. A batch estimation method is then adopted to identify the coefficients of the splines, also known as B-coefficients. This linear regression based MVSB scheme is a full-domain verifiable method due to the fact that its functional output are bounded by the maximum and the minimum B-coefficients[22]. In comparison with the OPBB identification method, the B-spline basis used in the MVSB method enjoys higher numerical stability since it is defined in terms of normalized Barycentric Coordinates[21][22]. In addition, the approximation power of the MVSB method can be increased far beyond the capabilities of any OPBB method by dividing the flight envelope into any number of subdomains called simplices [22][23], which are organized in a structure called a triangulation. Continuity constraints are used to enforce a predefined continuity order between neighboring simplices in the triangulation, thereby ensuring that the resulting spline function is globally continuous[22][21]. Compared with the well-known multivariate tensor product splines, the advantage of MVSB method is that it can fit scattered multidimensional datasets on non-rectangular domains[21][22].

Given the structure of the triangulation and the polynomial order of the basis functions, the MVSB identification problem can be formulated as a constrained linear system which can be solved using for example Lagrange multipliers [21]. The recursive identification scheme from [24] has been introduced into MVSB theory by de Visser et al. [13] in the form of the equality constrained recursive least squares (ECRLS) method. In the ECRLS method, the B-coefficients are recursively updated using the newly available data. This recursive identification algorithm has shown to be significantly more efficient in terms of computational load than the above mentioned batch method. However, real-time application of the MVSB is still hard to achieve due to the relatively high computational complexity when the B-coefficient vector contains more than one thousand elements. When applied to the problem of aerodynamic model identification of aircraft with highly nonlinear aerodynamics and extensive flight envelopes, the total number of elements in the B-coefficient vector can easily exceed this number. This may be either caused by a high resolution of the triangulation, a high polynomial order within each simplex, or a combination of both.

The objective of this paper is to present an improved recursive identification method using MVSB that has a lower computational load than the ECRLS method, while having a comparable approximation power. It is found that the computational efficiency of the ECRLS method can be further improved by avoiding the updating of the global covariance matrix, which becomes highly time-consuming for large scale problems. The merit of the new method is that it avoids the computation of a global covariance matrix altogether, and therefore its computational efficiency can be higher than recursive MVSB methods like ECRLS that do require the updating of a global covariance matrix.

Simulated flight data, generated with a subsonic F-16 flight simulation based on a NASA wind-tunnel dataset [25], is used to validate the proposed recursive sequential MVSB (RS-MVSB) method.

The proposed new method is more efficient in computational terms than the batch MVSB and ECRLS methods at the cost of a minor loss in approximation power. As a baseline comparison, the OPBB method is also implemented. Our final goal is to provide an efficient MVSB method that is able to provide a global nonlinear aerodynamic model for the advanced model based flight control systems of the next generation fighter aircraft.

This paper is organized as follows. In section 2, preliminaries are given on multivariate simplex B-splines. In section 3, the recursive sequential identification scheme for the multivariate simplex B-splines is proposed. The analysis of computationally complexity is provided in section 4. In section 5, the method is compared with both the batch method and the ECRLS method for computational aspects. The new method is applied in the identification of an aerodynamic model for the F-16 in section 6. Finally, the conclusions and remarks are given in section 7.

II. Preliminaries on Multivariate Simplex B-splines

The basic principles for simplex splines are briefly introduced in this section. Firstly, subsection II A introduces how to compute barycentric coordinates for a single simplex. Secondly, the triangulation techniques are presented in subsection II B. Subsection II C describes the calculation of the simplex B-splines basis within a single simplex. Then, subsection II D presents the methodology of constructing the global basis vector (i.e. B-form vector). Thereafter, the equality constraints on the global B-coefficient vector are provided in subsection II E. Finally, the spline function space is defined.

A. Simplex and Barycentric Coordinates

Let t be an n-simplex formed by the convex hull of its $n+1$ non-degenerate vertices $(v_0, v_1, ..., v_n) \subset \mathbb{R}^n$. The normalized barycentric coordinates of some evaluation point $\mathbf{x} \in \mathbb{R}^n$ with respect to simplex t are defined as

$$b(\mathbf{x}) \;:=\; (b_0, b_1, ..., b_n) \in \mathbb{R}^{n+1}, \quad \mathbf{x} \in \mathbb{R}^n \tag{1}$$

which follows from the following implicit relation:

$$\mathbf{x} = \sum_{i=0}^{n} b_i v_i,\ \sum_{i=0}^{n} b_i = 1 \tag{2}$$

B. Triangulations of Simplices

The approximation power of the multivariate simplex spline is partly determined by the structure of the triangulation. A triangulation $\mathcal{T}$ is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathcal{T} := \bigcup_{i=1}^{J} t_i,\ t_i \cap t_j \in \left\{\varnothing, \tilde{t}\right\}, \forall\, t_i, t_j \in \mathcal{T} \tag{3}$$

with the edge simplex $\tilde{t}$ a k-simplex with $0 \leq k \leq n-1$. High quality triangulations can be obtained using constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented by Shewchuk [26].

C. Basis Functions of the Simplex B-splines

According to [22] and [23], the Bernstein basis polynomial $B^d_\kappa\left(b(\mathbf{x})\right)$ of degree d in terms of the barycentric coordinates $b(\mathbf{x}) = (b_0, b_1, ..., b_n)$ from Eq. 2 is defined as:

$$B^d_\kappa(b(\mathbf{x})) := \begin{cases} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} & , \mathbf{x} \in t \\ 0 & , \mathbf{x} \notin t \end{cases} \tag{4}$$

where $\kappa = (\kappa_0, \kappa_1, ..., \kappa_n) \in N^{n+1}$ is a *multi-index* with the following properties: $\kappa! = \kappa_0!\kappa_1!...\kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + ... + \kappa_n$. In Eq. 4 we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} ... b_n^{\kappa_n}$. Given that $|\kappa| = d$, the total number of valid permutations of the *multi-index* κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!} \tag{5}$$

In [27], it was proved that any polynomial $p\left(b\right)$ of degree d on a simplex t can therefore be written as a linear combination of $\hat{d}$ basis polynomials in what is known as the B-form as follows:

$$p^t(b(\mathbf{x})) := \begin{cases} \sum_{|\kappa|=d} c^t_\kappa B^d_\kappa(b(\mathbf{x})) & , \mathbf{x} \in t \\ 0 & , \mathbf{x} \notin t \end{cases} \tag{6}$$

with c^t_κ the B-coefficients which uniquely determines $p^t(b(\mathbf{x}))$, where the superscript 't' indicates that p is defined on the simplex 't'. The total number of basis function terms is equal to $\hat{d}$, which is the total number of valid permutations of κ.

D. Vector Formulations of the B-form

As introduced in [13], the vector formulation, according to Eq. 6, for a B-form polynomial $p(b(\mathbf{x}))$ in barycentric $\mathbb{R}^{n+1}$ has the following expression:

$$p^t(\mathbf{x}) := \begin{cases} \mathbf{B}_t^d(b(\mathbf{x})) \cdot \mathbf{c}^t & , \mathbf{x} \in t \\ 0 & , \mathbf{x} \notin t \end{cases}, \tag{7}$$

with $b(\mathbf{x})$ the barycentric coordinates of the Cartesian $\mathbf{x}$. The row vector $\mathbf{B}_t^d(b(\mathbf{x}))$ in Eq. 7 is constructed from individual basis polynomials which are sorted lexicographically[13].

The simplex B-spline function $s_d^m(b(\mathbf{x}))$ of degree d and continuity order m, defined on a triangulation consisting of J simplices, is defined as follows:

$$s_d^m(\mathbf{x}) := \mathbf{B}^d(b(\mathbf{x})) \cdot \mathbf{c} \quad \in \mathbb{R}, \tag{8}$$

with $\mathbf{B}^d(b(\mathbf{x}))$ the global vector of basis polynomials which has the following full expression:

$$\mathbf{B}^d(b(\mathbf{x})) := [\ \mathbf{B}_{t_1}^d(b(\mathbf{x}))\ \ \mathbf{B}_{t_2}^d(b(\mathbf{x}))\ \ \cdots\ \ \mathbf{B}_{t_J}^d(b(\mathbf{x}))\ ] \ \in \mathbb{R}^{1 \times J \cdot \hat{d}} \tag{9}$$

Note that according to Eq. 7 we have $\mathbf{B}_{t_j}^d(b(\mathbf{x})) = 0$ for all evaluation locations $\mathbf{x}$ that are located outside of the triangle t_j. This results in that $\mathbf{B}^d$ is a sparse row vector.

The global vector of B-coefficients $\mathbf{c}$ in Eq. 8 has the following formulation:

$$\mathbf{c} := \left[\mathbf{c}^{t_1\top} \ \ \mathbf{c}^{t_2\top} \ \ \ldots \ \ \mathbf{c}^{t_J\top} \right]^\top \ \in \mathbb{R}^{J \cdot \hat{d} \times 1} \tag{10}$$

with each $\mathbf{c}^{t_j}$ a per-simplex vector of lexicographically sorted B-coefficients.

For a single observation on y we have:

$$y = \mathbf{B}^d(b(\mathbf{x}))\mathbf{c} + r \tag{11}$$

with r the residue. Then, for all the N observations, we have the following well-known formulation:

$$\mathbf{y} = \mathbf{X}(b(\mathbf{x}))\mathbf{c} + \epsilon \in \mathbb{R}^{N \times 1} \tag{12}$$

with $\mathbf{X}(b(\mathbf{x})) \in \mathbb{R}^{N \times J \cdot \hat{d}}$ a collection matrix of the row vector $\mathbf{B}^d$ from Eq. 9, and $\epsilon = [r_1, r_2, ...r_N]^T$ the residue vector. For writing convenience, $\mathbf{X}(b(\mathbf{x}))$ will be written as $\mathbf{X}$ in the remainder of this paper.

E. Global Continuity Constraints

To keep the smoothness among all subdomains, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H} \cdot \mathbf{c} = 0 \tag{13}$$

with $\mathbf{H} \in \mathbb{R}^{(E \cdot R) \times \left(J \cdot \hat{d}\right)}$ the smoothness matrix [23][21], R is the number of continuity conditions per edge. E is the number of edges in the specified triangulation. If all the simplices' surfaces connect smoothly on the edges within the whole triangulation, we call the simplex splines globally continuous. Global continuity is determined by Eq. 13.

F. Spline Function Space and a Polynomial Function Space

In this paper, we use a new type of definition of polynomial function space:

$$P_d\left(n\right) := \left\{p_k\left(\mathbf{x}\right) : p_k|_{\mathbf{x}} \in \mathbb{P}_k,\ \forall \mathbf{x} \in \mathbb{R}^n \ and\ \forall k \leq d\right\} \tag{14}$$

with $\mathbf{x}$ the input vector, $\mathbb{P}_k$ the space of polynomials of degree k.

We use the following definition of the spline space, which is a modified form of the definition given by Lai et al. in [22]:

$$S_d^m\left(n\right) := \left\{s_d^m\left(\mathbf{x}\right) \in C^m : s_d^m|_{\mathbf{x}} \in \mathbb{P}_d,\ \forall \mathbf{x} \in \mathbb{R}^n\right\} \tag{15}$$

with $\mathbb{P}_d$ the space of polynomials of degree d, and n the dimension of function inputs.

Note that, the former represents the ordinary polynomial function bases with the order up to d. For example, if we select $\mathbf{x} = [x,\ y]^T$, then $P_2\left(2\right) := c_1 + c_2 x + c_4 y + c_3 x^2 + c_6 xy + c_5 y^2$ with x and y two elements of $\mathbf{x}$.

III. Recursive Sequential Identification Method with Multivariate Simplex Spline

The computational load of the batch [23] and ECRLS [13] methods increases to the point of being impractical for use in online identification when the complexity and required update rate of a system increases beyond a certain point. The new recursive sequential least squares (RS-LS) method circumvents this problem by negating the need to update an increasingly large covariance matrix, thereby providing an effective online identification method for large scale systems of high complexity. In this section, the theory of the RS-LS identification method is presented.

A. Theoretical Development

Combining Eq. 12 with Eq. 13, the linear regression formulation of the multivariate simplex B-splines is derived as follows:

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{c} + \epsilon \tag{16}$$

$$\text{s.t. } \mathbf{H} \cdot \mathbf{c} = 0 \tag{17}$$

Assume that the singular value decomposition of $\mathbf{H}$ is as follows:

$$\mathbf{H}_{n\times m} = \mathbf{V}_{n\times n} \begin{bmatrix} \sum_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)} \end{bmatrix} \mathbf{U}^{\top}_{m\times m} \tag{18}$$

where $\sum = \text{diag}\left(\sigma_1 \ , \ \cdots \ \sigma_r\right)$ is the diagonal vector of eigenvalues , $\sigma_1 \geq \cdots \geq \sigma_r > 0$, and r is the rank of $\mathbf{H}$. $\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix}$ is an n_{th} order orthogonal matrix, $\mathbf{V}_1$ is a n by r matrix. $\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}$ is a m_{th}-order orthogonal matrix, $\mathbf{U}_1$ is a m by r matrix. It can be seen from Eq. 17 that $\mathbf{c} \in \mathit{null}\left(\mathbf{H}\right)$.

Let $\mathcal{J}\left(\mathbf{c}\right)$ be a least squares cost function of the global B-coefficient vector $\mathbf{c} = [\mathbf{c}^{t_j}]_{j=1}^{J} \in \mathbb{R}^{J\cdot\hat{d}\times 1}$ as follows:

$$\mathcal{J}\left(\mathbf{c}\right) = \left(\mathbf{y} - \mathbf{X}\mathbf{c}\right)^{\top}\left(\mathbf{y} - \mathbf{X}\mathbf{c}\right) \tag{19}$$

where $\mathbf{X} \in \mathbb{R}^{N\times J\cdot\hat{d}}$ is a matrix of B-form regressors for N observations as derived in [13], and vector $\mathbf{y}$ contains all N observations. J is the number of simplices, and $\hat{d}$ is the length of the coefficients vector located on each simplex.

A definition of the optimal linear minimum mean square error(LMMSE) estimation from [28] is stated as follows:

Given a set of linearly independent vector bases $\mathbf{M} = \{\eta_1, \eta_2, ..., \eta_{n-1}, \eta_n\}$, we use a linear combination $\hat{\varepsilon} = \sum\limits_{i=1}^{n} a_i\eta_i$ to approximate the unknown random variable vector ε. The fitting error is defined as:

$$\xi = \varepsilon - \hat{\varepsilon} = \varepsilon - \sum_{i=1}^{n} a_i \eta_i \tag{20}$$

The mean square fitting error is defined as:

$$\mathbf{P} = \left\| \varepsilon - \sum_{i=1}^{n} a_i \eta_i \right\|^2 \tag{21}$$

Our mission is to derive n constant parameters $a_1, a_2, ...a_n$ that make the mean square error have the minimum value.

One theorem from [29] and [30] also needs to be introduced: If the set of vector bases $\mathbf{M} = \{\eta_1, \eta_2, ...\eta_{n-1}, \eta_n\}$ are orthonormal bases, then the linear minimum mean square error (LMMSE) estimation of the random variable vector ε is given by

$$\hat{\varepsilon} = \sum_{i=1}^{n} \langle \varepsilon, \eta_i \rangle \eta_i = \sum_{i=1}^{n} a_i \eta_i \tag{22}$$

where $\langle \varepsilon, \eta_i \rangle$ is the inner product of ε and η_i.

B. Recursive Sequential Multivariate Simplex B-splines (RS-MVSB)

This method will be illustrated using a training dataset generated by the following function:

$$f = f(\mathbf{x}_1, \mathbf{x}_2) \tag{23}$$

where the column vectors $\mathbf{x}_1$, $\mathbf{x}_2$ are the inputs of function f. A spline function $s_d^m(b(\mathbf{x}_1, \mathbf{x}_2))$ is used to approximate function f.

A triangulation $\mathcal{T}$ is obtained using the Delaunay method according to the span in each dimension of $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]^T$. In the recursive sequential MVSB scheme, a modular subsystem is defined for each simplex (or set of simplices) of $\mathcal{T}$. Each modular subsystem defined on a small number of subdomains will be refered to as a module in the remainder of this paper. An example of constructing modules among the whole triangulation for two dimensional training datasets is illustrated in Fig. 1(a) and Fig. 1(b). $s_1, s_2, ...s_8$ are eight simplices in $\mathcal{T}$ as shown in Fig. 1(a). Two modules are constructed as shown in Fig. 1(b), where each module contains four simplices. In practice, the number of simplices in each module is selected through a cross validation process.

(a) triangulation in two dimensions.

(b) module construction from triangulation.

Fig. 1 Illustration of module construction.

Let $\mathbf{c}_i$ B-coefficient vector of the i^{th} module. In the recursive sequential method, we construct an independent estimation problem for each module. During every time step, we update the identification vector $\mathbf{c}_i$ which belongs completely to the specified module. Although updating the covariance matrix for each module is inevitable, the computation of the global covariance matrix is avoided. Because the global covariance matrix is many times larger than the local covariance matrix which belongs to one single module, this greatly reduces the computational time.

Let J be the number of simplices included in $\mathcal{T}$. The discontinuous global B-coefficient vector $\mathbf{c}$ has the following structure:

$$\mathbf{c} = \left[\mathbf{c}_1{}^\top \mathbf{c}_2{}^\top \ldots \mathbf{c}_i{}^\top \mathbf{c}_J{}^\top\right]^\top \tag{24}$$

The RS-MVSB method is comprised of two consecutive steps for each new data point: 1)updating the B-coefficient vector $\mathbf{c}_i$ of one single module and constructing vector $\mathbf{c}$; 2) deriving the smooth global B-coefficient vector $\widetilde{\mathbf{c}}$ from the discontinuous vector $\mathbf{c}$ by some methodology under the constraints Eq. 17.

Given the splines' triangulation $\mathcal{T}$, the kernel space matrix $\mathbf{U}_2$ for the linear mapping transformation is fixed and thus the orthonormal basis vectors which are columns of $\mathbf{U}_2$ are fixed and time invariant. According to [31], the smooth vector $\widetilde{\mathbf{c}}$ which satisfies Eq. 17 should have the same degree of freedom as the kernel space determined by the columns of matrix $\mathbf{U}_2$. Inspired by this conclusion, a hard curtailing method is proposed to enforce the continuity constraints Eq. 17 at each recursive time step.

According to Eq. 17, vector $\widetilde{\mathbf{c}}$ should be located in the kernel space of H which is determined by $\mathbf{U}_2$ as shown in Eq. 18. In addition, the column vectors of matrix $\mathbf{U}_2$ are orthonormal bases of space $null\left(\mathbf{H}\right)$ according to [31]. Let

$$\mathbf{U}_2 = \{\eta_1, \eta_2, ...\eta_{n-1}, \eta_n\} \tag{25}$$

By means of linear minimum mean square error (LMMSE) estimation from Eq. 22, the final smooth global B-coefficient vector $\widetilde{\mathbf{c}}$ is derived from the discontinuous global B-coefficient vector $\mathbf{c}$ as follows:

$$\widetilde{\mathbf{c}} = \sum_{i=1}^{n} \langle \mathbf{c}, \eta_i \rangle \, \eta_i \tag{26}$$

Fig. 2 provides an overview of the overall adaptive control system. The function of the 'Onboard Spline Model' block is to reconstruct an aircraft aerodynamic model, and the 'Inner-loop Controller' block is aimed at producing part of control input signals using the identified aerodynamic coefficients. As can be seen from Fig. 2, the recursive sequential setup can be summarized into three steps:

1) As a preparation, the $\mathbf{U}_2$ matrix is derived from Eq. 18, and a local module for each simplex of $\mathcal{T}$ triangulation is constructed. Let the number of simplices be J, then J estimation sub-problems are constructed.

2) When a new data point is available, the simplex index is used to decide which module it belongs to. Suppose that the current data point belongs to the i^{th} module, then updating the local parameter vector segment $\mathbf{c}_i$ of the i^{th} module using a least squares algorithm. Then we can get the corrected $\mathbf{c}$ according to Eq. 24. Simultaneously, the regressors' covariance matrix $\mathbf{P}_i$ of the i^{th} module is also updated.

3) The smooth global B-coefficient vector $\widetilde{\mathbf{c}}$ is derived from $\mathbf{c}$ using LMMSE estimation according to Eq. 26. Then, go to step 2 and wait for new data points.

It should be noted here that the first step of the recursive sequential method is independent of the second step. Therefore, we are allowed to skip the smoothing step (LMMSE estimation) occasionally as long as the smoothed vector $\widetilde{\mathbf{c}}$ is not readily needed in order to reduce the computational time.

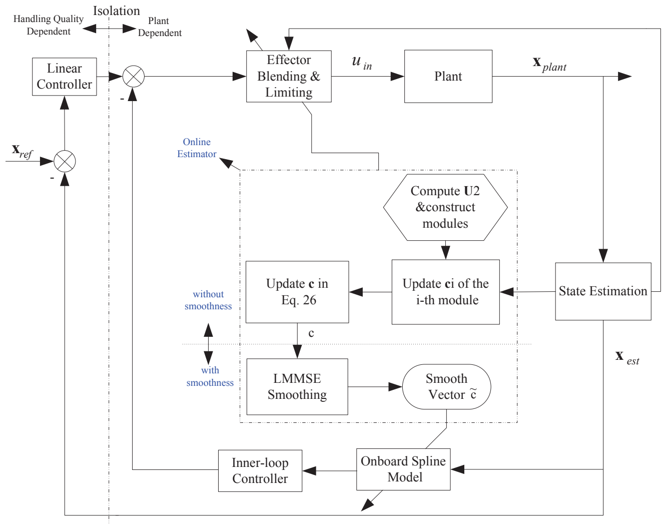


Fig. 2 Structure of an adaptive control system based on multivariate spline model.(Computational flow chart of the RS-LS method is described in the central block.)

The main contribution of this algorithm is that updating the global covariance matrix is avoided in using recursive least square method, and therefore computational load is greatly reduced.

IV. Computational Complexity

The batch method, presented in [23], contains a matrix inversion. Therefore, the batch method has a computational complexity of $\mathcal{O}\left(m^3\right)$ with m the dimensionality of the covariance matrix if it uses the Gaussian elimination to calculate the matrix inversion. The computational complexity of the ECRLS method, presented in [24] and [13], is $\mathcal{O}\left(3m^2\right)$ [24] which is the result of the recursive updating of the global covariance matrix. Therefore, the ECRLS method is more suitable for real-time applications because the number of algebraic operations and the required memory volumes are reduced in each cycle [24].

The computational complexity of the RS-MVSB method presented in Sec. III is the result of two operations. The first operation is the updating of the global covariance matrix at each cycle. Since the computational complexity of the ECRLS method is $\mathcal{O}\left(3m^2\right)$, we can calculate the computational time which is consumed in updating the covariance matrix of the RS-MVSB method by substituting the scale of the identification problem. Assume the number of modules in

Table 1 Computational Complexity (CC) in time

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
CC	$\mathcal{O}\left(m^3\right)$	$\mathcal{O}\left(3m^2\right)$	$\mathcal{O}\left(3\frac{m^2}{k^2}\right)$	$\mathcal{O}\left(2m^2-2mr+3\frac{m^2}{k^2}\right)$

the $\mathcal{T}$ triangulation is k, then the computational complexity of updating the covariance matrix is $\mathcal{O}\left(3\left(\frac{m}{k}\right)^2\right)=\mathcal{O}\left(3\frac{m^2}{k^2}\right)$. The second operation is the smoothing of the global B-coefficient vector by Eq. 22:

$$\mathcal{O}\left(m\left(m-r\right)+m\left(m-r\right)\right)\approx\mathcal{O}\left(2m\left(m-r\right)\right) \tag{27}$$

where r is the rank of the smoothness matrix $\mathbf{H}$, and only the multiplication operation is considered.

As mentioned in Sec. III B, the smoothing step in the recursive sequential method is not necessary to be executed if the smooth global covariance matrix is not readily needed. In this case, the computational complexity of the overall method reduces to $\mathcal{O}\left(3\frac{m^2}{k^2}\right)$ in each cycle. The computational complexity (CC) analysis is concluded in Table 1.

In Table 3, the letter 'R' is an acronym of 'Reduced', which means the LMMSE smoothing process is carried out every 1000 time steps. Accordingly, the letter 'N' is an abbreviation of 'Not reduced'.

V. Computational Aspects

In this section, the recursive sequential method is compared with the batch method presented in [23] and the ordinary recursive least squares (ECRLS) method presented in [13]. Firstly, the recursive sequential least squares based multivariate spline method is applied to fit a bivariate dataset, and compared with the batch method and ECRLS method. Secondly, it was validated by being applied to approximate a trivariate dataset in a hybrid manner using multi-function structure. Thirdly, the proposed method was applied to fit the same three dimensional dataset but with a different triangulation structure, where total number of B-coefficients is larger than 2000 as a result of the increased number of simplices. This experiment is designed for demonstrating the advantage in terms of computational efficiency of the recursive sequential MVSB (RS-MVSB) method when it is applied to large scale problems.

A. Demonstration Setup

In this section, two datasets are used. The first one is a trivariate dataset χ_{3d} which consists of 22000 scattered data points in $\mathbb{R}^3$, generated using a uniform random number generator in the interval $\{[0,1],[0,1],[0,1]\}$:

$$\chi_{3d} = \{x_1, x_2, x_3\} \in \{U[0,1], U[0,1], U[0,1]\} \tag{28}$$

The output measurements of the data points were generated using a three dimensional compound function:

$$f_t(x_1, x_2, x_3) = 0.33\{f_1(x_1, x_2, x_3) + f_2(x_1, x_2)\} + k_1 \cdot v \tag{29}$$

where $k_1 \cdot v$ is a uniformly distributed white noise sequence of magnitude k_1, the subfunctions take the following formulation:

$$f_1(x_1, x_2, x_3) = x_1 x_2 + x_3 \sin(1.5x_2 + x_1) \tag{30a}$$

$$f_2(x_1, x_2) = {x_2}^2 \sin(2x_1 + 10) + x_1 \cos(1.5x_2) \tag{30b}$$

20000 data points are used as training data and the remaining 2000 data are used as validation data.

The second dataset χ_{2d}, which contains 20000 data points in $\mathbb{R}^2$, is a bivariate dataset with its outputs generated using Eq. 30b together with $k_1 \cdot v$. In all of the following numerical experiments, we selected $k_1 = 0.02$.

B. Comparison with the Batch Method and ECRLS Method

In the first numerical experiment, the bivariate training dataset χ_{2d} consisting of 20000 scattered data points is used. For the multivariate spline function $S(x) = S_d^m(n)$, we select n=2, d=5, m=1, where n,d, and m are defined in Eq. 15. The structure of the triangulation is determined by vertices located on the grid $\{[0, 0.5, 1], [0, 0.5, 1]\}$.

The recursive sequential least squares (RS-LS) method developed in Sec. III was compared with the batch method presented in [23] and the ordinary recursive least squares (ECRLS) method

Table 2 Comparison in 2-D, 168 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)
normalized time	514.1	1.0	0.6110
time per cycle(ms)	53.97	0.105	0.0641
max ($\|\epsilon\|$)	0.0792	0.0797	0.0839
RMSE (ϵ)	0.0201	0.0201	0.0201

presented in [13]. The comparison results are shown in Table 2. The length of the B-coefficient vector $\tilde{\mathbf{c}}$ is 168. The total computational time, which is shown in the first row, is normalized using the computational time of ECRLS method. Apparently, the RS-LS method enjoys a significantly lower computational time than the batch method and ECRLS method while maintaining equivalent fitting accuracy (implied by max ($|\epsilon|$) and root mean squared error (RMSE)). The RMSEs are very close to the magnitude of the white noise add into the training data.

In the second numerical experiment, the RS-LS method is used to approximate the three dimensional dataset χ_{3d} consisting of 20000 data points, and the results are elaborated in Table 3. In this case, a compound spline function $S\left(x\right) = S_{d_1}^{m_1}\left(n_1\right) + S_{d_2}^{m_2}\left(n_2\right)$ consisting of a trivariate and a bivariate simplex B-spline function is used to approximate the function in Eq. 29. We selected $n_1 = 3$, $d_1 = 5$, $m_1 = 0$ and $n_2 = 2$, $d_2 = 4$, $m_2 = 1$. In the implementation of the recursive sequential method, a virtual covariance matrix with block diagonal form is constructed in Eq. 31.

$$\mathbf{C}_g = \begin{bmatrix} \mathbf{C}_{11} & & & & & & \\ & \mathbf{C}_{12} & & & & & \\ & & \cdot & & & & \\ & & & \mathbf{C}_{1i} & & & \\ & & & & \cdot & & \\ & & & & & \mathbf{C}_{1m} & \\ & & & & & & \mathbf{C}_{2t} \end{bmatrix} \tag{31}$$

where $\mathbf{C}_{1i}$ is covariance matrix of the i^{th} simplex in triangulation $\mathcal{T}_1$, $\mathbf{C}_{2t}$ is the global covariance matrix of the triangulation $\mathcal{T}_2$. Let n_1 and n_2 be the total number of B-coefficients which belong to $\mathcal{T}_1$ and $\mathcal{T}_2$ separately, then the size of the square matrix $\mathbf{C}_{1i}$ is $\frac{n_1}{m}$ with m the number of modules

Table 3 Comparison in 3-D, 366 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
normalized time	709.0420	1.0	0.3150	0.4057
per cycle(ms)	348.9557	0.4922	0.1552	0.1997
max ($\lvert\epsilon\rvert$)	0.0795	0.0750	0.0745	0.0736
RMSE (ϵ)	0.0200	0.0199	0.0205	0.0201

in $S_{d_1}^{m_1}(n_1)$ and the size of the square matrix $\mathbf{C}_{2t}$ is n_2. Assume the current data point is located in the i^{th} module of $S_{d_1}^{m_1}(n_1)$, then the local covariance matrix for the current time step is:

$$\mathbf{C}_l = \begin{bmatrix} \mathbf{C}_{1i} & \\ & \mathbf{C}_2 \end{bmatrix} \tag{32}$$

To get the results in Table 3, the support vertices $\{[0,1],[0,1],[0,1]\}$ were selected to construct the $\mathcal{T}$ triangulation. The length of the B-coefficient vector $\tilde{\mathbf{c}}$ is 366.

It can be seen from Table 3 that both the ECRLS method and the RS-LS method are significantly faster than the batch method. The best performing RS-LS method (i.e. RS-LS(R-s)) is more than 2000 times faster than the batch method. According to column 2 and 3, RS-LS method is 3 times faster than the ECRLS method while maintaining nearly the same approximation ability (implied by the RMSE). Note that, the RMSEs are very close to the magnitude of the white noise added to the training data.

Thirdly, the advantage of the RS-MVSB method for large scale and multi-function identification problems is demonstrated using the three dimensional dataset χ_{3d}, which contains 20000 data points as described in Sec. V A. In this numerical experiment, a spline function $S(x) = S_{d_1}^{m_1}(n_1) + S_{d_2}^{m_2}(n_2)$ is used to approximate the function $f_t(x_1, x_2, x_3)$ shown in Eq. 29. A different triangulation $\mathcal{T}$ was constructed using the support vertices $\{[0,0.5,1],[0,0.5,1],[0,0.5,1]\}$. The B-spline function $S_{d_1}^{m_1}(n_1)$ was constructed using $n_1 = 3$, $d_1 = 5$, $m_1 = 0$, and $S_{d_2}^{m_2}(n_2)$ was constructed using $n_2 = 2$, $d_2 = 4$, $m_2 = 1$. The length of the global B-coefficient vector $\tilde{\mathbf{c}}$ is 2808.

In this numerical experiment, 20000 data points from dataset χ_{3d} from Eq. 28 were used as training data for the batch method, the ECRLS method and the RS-LS method. The results in Table 4 clearly show that the ECRLS method is more than 200 times faster than the batch method,

Table 4 Comparison in 3-D, 2808 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
normalized time	243.42	1.0	0.0154	0.1317
per cycle(ms)	10094.4	41.4682	0.6390	5.4641
max ($\|\epsilon\|$)	0.0824	0.0792	0.0860	0.0915
RMSE (ϵ)	0.0200	0.0201	0.0253	0.0242

while the RS-LS method is 60 times faster than the ECRLS method while the approximation power (implied by RMSE) is close to that of the batch method. The RMSE approaches the magnitude of the white noise added to the training data. Again, this speedup is the direct result of not having to update a global covariance matrix.

VI. Application on Aerodynamic Model Identification of the F-16

In this section, the recursive sequential algorithm was applied to identify an aerodynamic model of the F-16 fighter aircraft based on data from a NASA wind-tunnel database of the F-16 [25]. The next generation of fighter aircraft will have highly nonlinear aerodynamics. In order to obtain accurate aerodynamic models for such aircraft, their full flight envelopes need to be partitioned using high resolution triangulations. The result of this is a longer B-coefficient vector for the simplex B-splines. In case the aerodynamic model needs to be updated online, for example after a damage event, the batch and ECRLS methods may become infeasible due to their low computational efficiency. This section is aimed at demonstrating the approximation power of the new RS-MVSB method when dealing with a large scale, complicated nonlinear modeling real-world problem. The RS-MVSB method is compared with both the polynomial basis based identification method and the batch MVSB method. In the remainder of this paper, we assume that all aircraft states have been obtained from the 'State Estimation' block (e.g. an extended Kalman filter) shown in Fig. 2. The model identification method proposed in this paper forms the second step of the so-called two step method [32].

A. F-16 Aerodynamic Model Structure

$$F_x\left(\alpha,\beta,\delta_e,\delta_{lef},\frac{q\bar{c}}{V}\right)=f_1\left(\alpha,\beta,\delta_e\right)+f_2\left(\alpha,\beta\right)\cdot\delta_{lef} + f_3\left(\alpha\right)\cdot\frac{q\bar{c}}{V}+f_4\left(\alpha\right)\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef} \tag{33}$$

According to Eq. 33, we can derive the following linear regression formulation for compound multivariate B-splines.

$$F_x = F_{x1}F_{x2} \tag{34}$$

where

- $F_{x1}=\left[\mathbf{B}_1,\mathbf{B}_2\cdot\delta_{lef},\mathbf{B}_3\cdot\frac{q\bar{c}}{V},\mathbf{B}_4\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef}\right]$
- $F_{x2}=\left[\mathbf{c}_1^\top,\mathbf{c}_2^\top,\mathbf{c}_3^\top,\mathbf{c}_4^\top\right]^\top$

For simplicity, let $\mathbf{B}_3=\mathbf{B}_4$, then the third term and the fourth term in Eq. 34 can be merged:

$$F_x = F_{x3}F_{x4} \tag{35}$$

where

- $F_{x3}=\left[\mathbf{B}_1,\mathbf{B}_2\cdot\delta_{lef},\mathbf{B}_3\cdot\frac{q\bar{c}}{V}\cdot(1+\delta_{lef})\right]$
- $F_{x4}=\left[\mathbf{c}_1^\top,\mathbf{c}_2^\top,\mathbf{c}_3^\top\right]^\top$

According to de Visser [33], the global continuity matrix for the combined spline function has the following expression:

$$\mathbf{H}=\begin{bmatrix}\mathbf{H}_1 & & \\ & \mathbf{H}_2 & \\ & & \mathbf{H}_3\end{bmatrix} \tag{36}$$

with $\mathbf{H}_1,\mathbf{H}_2,\mathbf{H}_3$ the continuity matrices of the three independent sub functions.

In the implementation of the RS-MVSB method, a virtual covariance matrix in the block diagonal form is constructed in Eq. 37.

$$\mathbf{C}_g = \begin{bmatrix} \mathbf{C}_{tn_1} & & & & & & & \\ & \mathbf{C}_{tn_2} & & & & & & \\ & & \cdot & & & & & \\ & & & \mathbf{C}_{tn_i} & & & & \\ & & & & \cdot & & & \\ & & & & & \mathbf{C}_{tn_m} & & \\ & & & & & & \mathbf{C}_{t_{n2}} & \\ & & & & & & & \mathbf{C}_{t_{n3}} \end{bmatrix} \tag{37}$$

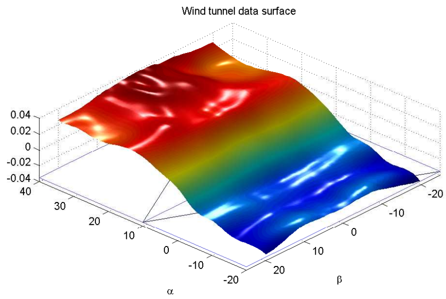
where $\mathbf{C}_{tn_i}$ is the covariance matrix of the i^{th} simplex in triangulation $\mathcal{T}_1$, $\mathbf{C}_{n2}$ and $\mathbf{C}_{n3}$ are covariance matrices of triangulation $\mathcal{T}_2$ and $\mathcal{T}_3$ separately. In this equation, $tn_1+tn_2+...+tn_i+...+tn_m = n1$ where $n1$, $n2$ and $n3$ are the total number of B-coefficients which belong to $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$ separately. The local covariance matrix at the current time step is:

$$\mathbf{C}_l = \begin{bmatrix} \mathbf{C}_{tn_i} & & \\ & \mathbf{C}_{t_{n2}} & \\ & & \mathbf{C}_{t_{n3}} \end{bmatrix} \tag{38}$$

B. Original Wind Tunnel Data for F-16 Aircraft

The inputs of the training dataset were obtained by randomly generating 20000 input states inside the domain of the NASA wind-tunnel model. The inputs of the validation dataset containing 4331 points are produced by the grids determined by α and β. The outputs of both datasets were calculated using Eq. 33 based on the wind-tunnel data provided by [25] with $\delta_{lef} = 1\,degree, V = 600\,ft/s$, $q = 0.1\,rad/s$, $\bar{c} = 11.32\,m$. To investigate the effects from the measurement noise, the white noise with a magnitude of 1% (relative value to the corresponding upper and lower extreme) has been added to the model outputs (i.e. C_X, C_m) in generating the training datasets. This observation noise together with Eq. 33 is playing the same role as Eq. 29.

In this paper, we are only concerned with the identification of the longitudinal force and moment coefficients. In addition, only the results for C_X and C_m are provided since they show much more



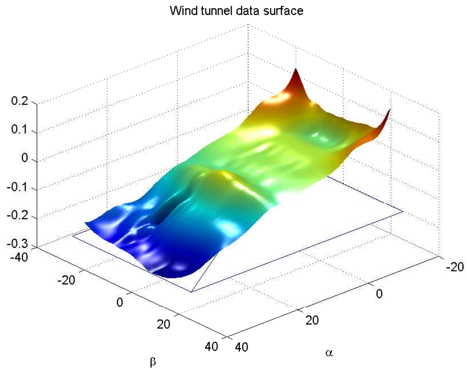


Fig. 3 C_X **wind-tunnel data surface (**$\delta_e = 0^o$**,** $\delta_{lef} = 1^o$**,** $\frac{q\bar{c}}{V} = 0.0018866$**).**

Fig. 4 C_m **wind-tunnel data surface (**$\delta_e = 0^o$**,** $\delta_{lef} = 1^o$**,** $\frac{q\bar{c}}{V} = 0.0018866$**).**

significant nonlinearities than C_Z. Three-dimensional surfaces of longitudinal force coefficient C_X and moment coefficient C_m when $\delta_e = 0$ were plotted in Fig. 3 and Fig. 4 separately. Highly nonlinear behavior of the aerodynamic coefficients can be observed in these figures.

C. Model Structure Selection and Recursive Sequential Identification

In the simulation, the MVSB function approximator contains three subfunctions: $S(x) = S_{d_1}^{m_1}(n_1) + S_{d_2}^{m_2}(n_2) + S_{d_3}^{m_3}(n_3)$, where we selected $n_1 = 3$, $n_2 = 2$, $d_2 = 4$, $m_2 = 1$; $n_3 = 1$, $d_3 = 3$, $m_3 = 1$ with d_1, m_1 undetermined. That is, the compound spline function is the following: $S(x) = S_d^m(3) + S_4^1(2) + S_3^1(1)$. The partitioning vector of α is $\begin{bmatrix} -20 & 10 & 40 \end{bmatrix}$. The partitioning vector of β is $\begin{bmatrix} -25 & 25 \end{bmatrix}$. The partitioning vector of δ_e is $\begin{bmatrix} -20 & 20 \end{bmatrix}$. The inputs were normalized into the range of [0 1] in order to further improve the approximation accuracy. In order to select a suitable model structure for the spline model, the effects of the structural parameters (i.e. d and m) will be investigated.

The results of the identification experiment with the OPBB method are shown in Fig. 5. The approximation results on the training dataset using batch MVSB method and RS-MVSB method are given separately in Fig. 6. Fig. 6(a) and Fig. 6(b) clearly shows that the approximation accuracy of both methods will increase with the increase of d, while decrease with the increase of continuity order m. Apparently, the approximation accuracy of the RS-MVSB method is very close to that of the batch MVSB method when continuity order is less than two, although the performance of the

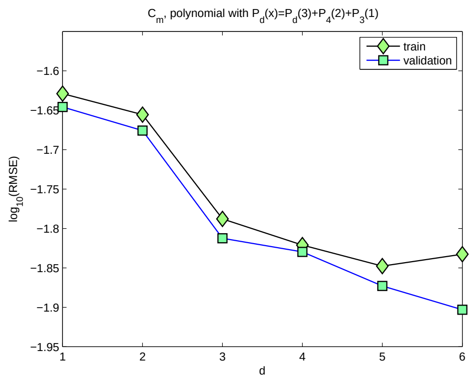
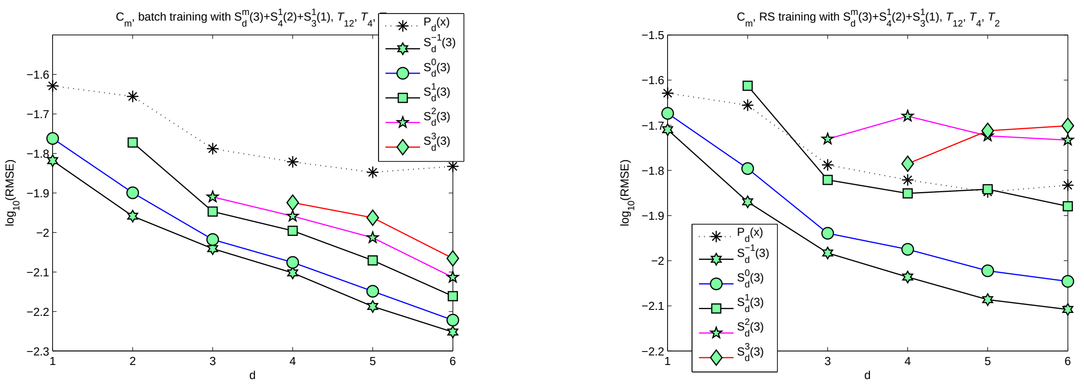


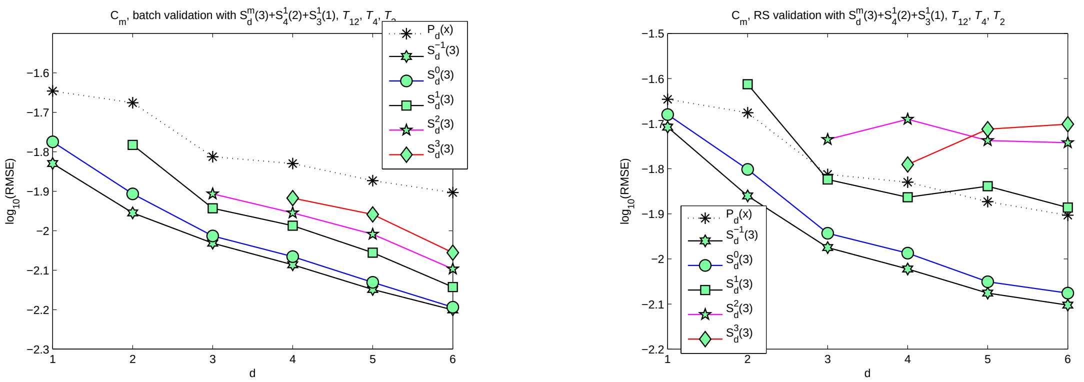
Fig. 5 Different combination of d for $P_d(3)$ together with $P_4(2)$ and $P_3(1)$.



(a) Batch training.

(b) RS-MVSB training.

Fig. 6 Different combination of m and d for $S_d^m(3)$ together with $S_4^1(2)$ and $S_3^1(1)$.



(a) Batch test.

(b) RS-MVSB test.

Fig. 7 Different combination of m and d for $S(3)_d^m$ together with $S_4^1(2)$ and $S_3^1(1)$.

new method is lower than the batch MVSB method when continuity order is high. Compared with

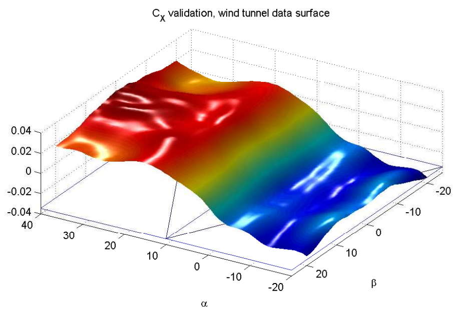
Fig. 5, Fig. 6 also shows that the MVSB method has a higher approximation power than the OPBB method, especially when the polynomial order is larger than 3.

The approximation accuracy of the aforementioned methods are validated using 4331 validation data points; the RMSEs are plotted in Fig. 7(a) and Fig. 7(b) separately. According to Fig. 7(a), the fitting accuracy of the Batch MVSB will increase with the increase of d and it will decrease with the increase of m. According to Fig. 7(b), when the continuity order m is lower than 3, the fitting accuracy of the RS-MVSB will also increase when the polynomial order increases or the continuity order decreases. On the other hand, when the continuity order is equal to 3, the quality of fit will decrease with the increase of d. Comparing Fig. 7(b) with Fig. 7(a), it can also be seen that the RS-MVSB method has a slightly lower fitting accuracy than batch MVSB, which is the price paid for the increased computational efficiency of updating the B-coefficients.

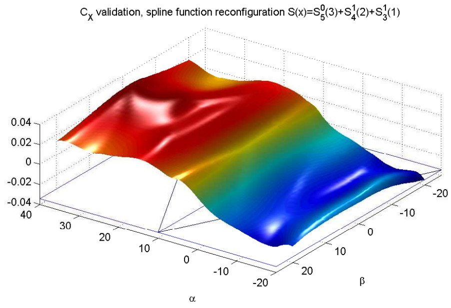
Finally, the performance of the RS-MVSB method is illustrated by comparing the predicted output on the 4331 knots with wind-tunnel data output. For the plotting of the 3 dimensional surface, we selected $\delta_e = 2^o$ in order to fix the third dimension δ_e. The spline function has the following structure: $S(x) = S_5^0(3) + S_4^1(2) + S_3^1(1)$. The structure of the OPBB approximator has the following form:$P(x) = P_5(3) + P_4(2) + P_3(1)$.

The wind-tunnel model for C_X for $\delta_e = 2^o$ is shown in Fig. 8(a) as a comparison baseline. The spline function surface reconstructed from the estimated global B-coefficient vector $\tilde{\mathbf{c}}$ is given by Fig. 8(b). Comparing Fig. 8(b) and Fig. 8(c) with Fig. 8(a), it can be seen that the MVSB model closer resembles the wind-tunnel model than the OPBB model. The surfaces of C_m plotted in Fig. 9(a), Fig. 9(b), and Fig. 9(c) show a similar result.

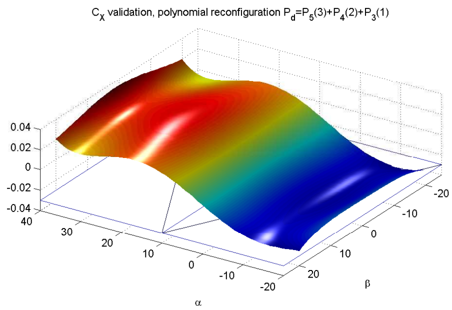
Finally, the most important advantage of the RS-MVSB method over the ECRLS MVSB method is that it requires a lower computational time at each recursive step when updating the B-coefficients. In the above mentioned simulation experiment, the length of the B-coefficient vector is 740 and the computational time of the RS-MVSB method is 0.9766 ms per data point. Whereas, the computational time of the ECRLS MVSB method is 3.6582 ms per data point. The low computational time requirement makes the RS-MVSB method more suitable for application in real-time adaptive flight control systems than the ECRLS MVSB method and the batch MVSB method.



(a) Original wind-tunnel data surface.

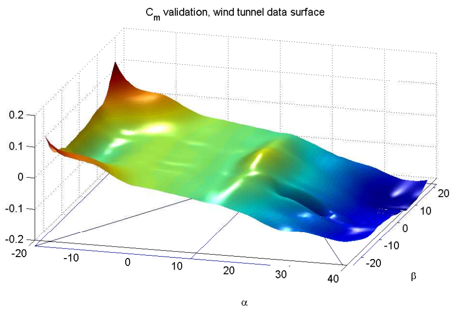


(b) Spline function surface, $\mathcal{T}_{12}, \mathcal{T}_4, \mathcal{T}_2$.

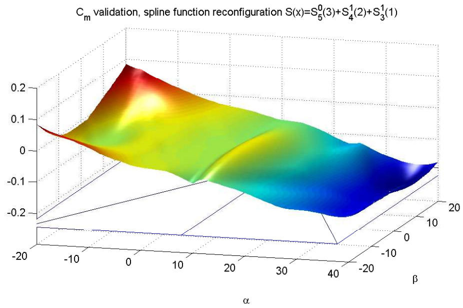


(c) Polynomial fitting surface.

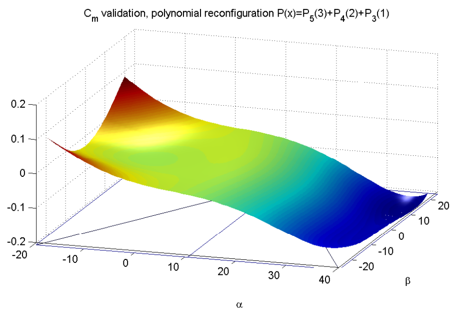
Fig. 8 Validation surface of $C_X\,(\alpha, \beta, \delta_e = 2^o)$ with $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$, 4331 data.



(a) Original wind-tunnel data surface.



(b) Spline function surface, $\mathcal{T}_{12}, \mathcal{T}_4, \mathcal{T}_2$.



(c) Polynomial fitting surface.

Fig. 9 Validation surface of $C_m\,(\alpha, \beta, \delta_e = 2^o)$ with $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$, 4331 data.

VII. Conclusions

A new recursive sequential strategy is developed and combined with the multivariate simplex B-splines with the aim of increasing the computational efficiency over existing multivariate spline based identification methods. This new recursive sequential multivariate B-splines method, called the RS-MVSB method, enables the online identification of large scale nonlinear aerodynamic models which are at the heart of modern adaptive model-based flight control systems. The RS-MVSB method consists of two sequential procedures at each time step. The first is the estimation of the B-coefficients for the local spline functions defined on the local simplex modules. The second procedure is the LMMSE-based smoothing step which results in a smooth global spline function.

The performance of the RS-MVSB method was compared to existing batch and recursive methods for simplex B-spline identification in a set of three numerical experiments. It was shown that the RS-MVSB method enjoys a significant advantage in terms of computational efficiency over the existing ECRLS method for large scale problems at a cost of a slight decrease in approximation power. In particular, the approximation power of the RS-MVSB method is comparable to the batch MVSB method and the ECRLS MVSB methods when the continuity order of the spline functions is lower than 2. For continuity orders larger than or equal to 2, the approximation power of the RS-MVSB method is lower than existing MVSB methods. In addition, all MVSB methods have a higher approximation power than the OPBB approximation method.

The RS-MVSB method is applied in the recursive identification of an aerodynamic model using data from a NASA nonlinear wind-tunnel data set of the F-16 fighter aircraft. A comparison with the ordinary polynomial method and the batch MVSB method shows that the new RS-MVSB method has all the advantages of the MVSB method (i.e. high approximation power) while at the same time greatly reducing the required computational time in updating the spline model parameters.

Acknowledgments

The authors would like to acknowledge Professor Jian-guo Sun, Nanjing University of Aeronautics and Astronautics, China for his constructive suggestions.

References

[1] Lombaerts, T. J., Stroosma, Q., Chu, Q. P., Mulder, J. A., and Joosten, D. A., "Piloted Simulator Evaluation Results of New Fault-Tolerant Flight Control Algorithm," *Journal of Guidance, Control and Dynamics*, Vol. 32, No. 6, 2009, pp. 1747–1765.

[2] Jay, F., "Backstepping-Based flight control with adaptive function approximation," *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 6, 2005, pp. 1089–1102.

[3] van Oort, E. R., Sonneveldt, L., Chu, Q. P., and Mulder, J. A., "Full-Envelope Modular Adaptive control of a Fighter Aircraft Using Orthogonal Least Squares," *Journal of Guidance, Control and Dynamics*, Vol. 33, No. 5, 2010, pp. 1461–1472.

[4] Morelli, E. A., "Real-Time Parameter Estimation in the Frequency Domain," *Journal of Guidance, Control and Dynamics*, Vol. 33, No. 3, 2000, pp. 707–723.

[5] Morelli, E. A., "Real-Time Aerodynamic Parameter Estimation without Air Flow Angle Measurements," *AIAA Atomospheric Flight Mechanics Conference*, AIAA, Toronto, Ontario Canada, 2010.

[6] Ljung, L., "Recursive identification algorithms," *Circuits Systems Signal Processing*, Vol. 21, 2002, pp. 57–68.

[7] Soderstrom, T., Ljung, L., and Gustavsson, I., "A theoretical analysis of recursive identification methods," *Automatica Journal*, Vol. 14, 1987, pp. 231–244.

[8] Ljung, L. and Soderstrom, T., *Theory and practice of recursive identification*, MIT Press, 1983, pp.67-135.

[9] Vincent, A. A. and Hassapis, G. D., "Nonlinear model identification and adaptive model predictive control using neural networks," *ISA Transactions*, Vol. 50, No. 1, 2011, pp. 177–194.

[10] Pillonetto, G., Quang, M. H., and Chiuso, A., "A New Kernel-Based Approach for Nonlinear System Identification," *IEEE Trans. on Automatic Control*, Vol. 56, No. 12, 2011, pp. 2825–2840.

[11] Zhao, Y. P. and Sun, J. G., "Recursive reduced least squares support vector regression," *Pattern Recognition*, Vol. 42, No. 5, 2009, pp. 837–842.

[12] Zhao, Y. P., Sun, J. G., Du, Z. H., and Zhang, Z. A., "An Improved Recursive reduced least squares support vector regression," *Neurocomputing*, Vol. 87, No. 1, 2012, pp. 1–9.

[13] de Visser, C. C., Chu, Q. P., and Mulder, J. A., "Differential Constraints for Bounded Recursive Identification with Multivariate Splines," *Automatica Journal*, Vol. 47, 2011, pp. 2059–2066.

[14] Ning, H. W., Jing, X. J., and Cheng, L., "Online identification of Nonlinear Spatiotemporal Systems using kernel learning approach," *IEEE Transactions on neural networks*, Vol. 22, No. 9, 2011, pp. 1381–1394.

[15] Girich, C. and Ravindra, J., "Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter," *Journal of Guidance, Control and Dynamics*, Vol. 14, No. 1, 2010, pp. 106–1117.

[16] Klein, V., Batterson, J. G., and Murphy, P. C., "Determination of airplane model structure from flight data by using modified stepwise regression," Nasa technical paper tp-1637, Langley Research Center, Washington, D.C., Oct. 1981.

[17] Morelli, E. A., "Global nonlinear aerodynamic modeling using multivariate orthogonal functions," *Journal of Aircraft*, Vol. 32, No. 2, 1995, pp. 270–277.

[18] Klein, V. and Morelli, E. A., *Aircraft System Identification, Theory and Practice*, AIAA Education Series, 2006, pp.92-221.

[19] Stark, J., "Adaptive Model Selection Using Orthogonal Least Squares Methods," *Mathematical, Physical and Engineering Sciences*, The Royal Society, Britain, 1997, pp. 21–42.

[20] Lombaerts, T. J., Van Oort, E. R., and Chu, Q. P., "Online Aerodynamic Model Structure Selection and Parameter Estimation for Fault-Tolerant Control," *Journal of Guidance, Control and Dynamics*, Vol. 33, No. 3, 2010, pp. 707–723.

[21] Awanou, G., Lai, M. J., and Wenston, P., *The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differencial Equations*, Wavelets and Splines, 2005, pp.24-75.

[22] Lai, M. J. and Schumaker, L. L., *Spline Functions on Triangulations*, Cambridge, 2007, pp.18-27.

[23] de Visser, C. C., Chu, Q. P., and Mulder, J. A., "A new Approach to Linear Regression with multivariate Splines," *Automatica Journal*, Vol. 45, 2009, pp. 2903–2909.

[24] Zhu, Y. and Li, X. R., "Recursive Least Squares With Linear Constraints," *Communications in Information and Systems*, Vol. 7, No. 3, 2007, pp. 287–312.

[25] Nguyen, L. T., Ogburn, M. E., and Gilbert, W. P., "Simulator Study of Stall/Post-Stall Characteristics of a Fighter airplane With Relaxed Longitudinal Static Stability," Nasa technical paper 1538, Langley Research Center, Hampion, Virginia, Dec. 1979.

[26] Shewchuk, J. R., "Delauney refinement algorithms for triangular mesh generation," *Computational Geometry*, Vol. 22, 2001, pp. 21–74.

[27] de Boor, C., *B-form basics. In:G. Farin, editor, Geometric modeling: algorithms and new trends*, Society for Industrial and Applied Mathematics, 1987, pp.131-148.

[28] Mohanty, N., *Random Signals Estimation and Identification:Analysis and Applications*, Van Nostrand Reinhold, 1986, pp.266-291.

[29] Zhang, X. D., *Matrix Analysis and Applications*, Tsinghua University Press, 2004, pp.659-662.

[30] Lawson, C. L. and Hanson, R. J., *Solving Least Squares Problems*, SIAM:Society for Industrial and Applied Mathematics, 1995, pp.148-168.

[31] Meyer, C. D., *Matrix Analysis and Applied Linear Algebra*, SIAM:Society for Industrial and Applied Mathematics, 2001, pp.403-450.

[32] Mulder, J. A., *Designed and evaluation of dynamic flight test manoeuvers*, Ph.D. thesis, Delft University of Technology, The Netherlands, Oct. 1986.

[33] de Visser, C. C., Mulder, J. A., and Chu, Q., "Global NonLinear Aerodynamic Model Identification with Multivariate Splines," *AIAA Atomospheric Flight Mechanics Conference*, AIAA, Chicago, Illinois, 2009.