A New Substitution Based Recursive B-Splines Method for Aerodynamic Model Identification

L. G. Sun and C. C. de Visser and Q. P. Chu

Abstract A new substitution based (SB) recursive identification method, using multivariate simplex B-splines (MVSBs), has been developed for the purpose of reducing the computational time in updating the spline B-coefficients. Once the structure selected, the recursive identification problem using the MVSBs turns to be a constrained recursive identification problem. In the proposed approach, the constrained identification problem is converted into an unconstrained problem through a transformation using the orthonormal bases of the kernel space associated with the constraint equations. The main advantage of this algorithm is that the required computational time is greatly reduced due to the fact that the scale of the identification problem, as well as the scale of the global covariance matrix, is reduced by the transformation. For validation purpose, the SB-RMVSBs algorithm has been applied to approximate a wind tunnel data set of the F-16 fighter aircraft. Compared with the batch MVSBs method and the equality constrained recursive least squares (ECRLS) MVSBs method, the computational load of the proposed SB-RMVSBs method is much lower than that of the batch type method while it is comparable to that of the ECRLS-MVSBs method. Moreover, the higher the continuity order is, the less computational time the SB-RMVSBs method requires compared with the ECRLS-MVSBs method.

L. G. Sun
Delft University of Technology, Delft, The Netherlands, 2600GB, e-mail: L.sun@tudelft.nl

C. C. de Visser
Delft University of Technology, Delft, The Netherlands, 2600GB e-mail: c.c.devisser@tudelft.nl

Q.P. Chu
Delft University of Technology, Delft, The Netherlands, 2600GB e-mail: q.p.chu@tudelft.nl

1 Introduction

The control performance of a model-based automatic control system, like for example the adaptive nonlinear dynamic inversion (ANDI) flight control system [4, 7] and the module based adaptive backstepping flight control system [7], heavily relies on the accuracy of the object model that is identified in real-time. Recently, de Visser et al. [11] proposed a novel batch type identification method using multivariate simplex B-splines. Comparing with the ordinary polynomial basis (OPB) based method, this simplex spline basis (SSB) based method can provide a relatively more stable basis and enjoys a higher approximation power owing to the fact that multiple local modules are identified instead of identifying a single overall model[10]. Another main merit of the multivariate simplex B-splines (MVSBs) is that they are capable of using the scattered dataset as training data. This is a property that the multivariate sensor product splines method does not have [11].

Later, de Visser and Chu et al. [12] developed an equality constrained recursive least squares (ECRLS) based MVSBs identification method after combining the linear regression formulation of the spline bases from [11] with the recursive least squares identification method from [15]. The recursive identification method presented in [15] can convert a constrained identification problem into a free-of-constraint identification problem . In this recursive identification method, the constrained recursive identification process is circumvented by merely injecting the equality constraint information into the general least square solution calculated using an initial training data collection.

However, in order to enable the real-time aerodynamic model identification, it is still necessary to reduce the computational load of the recursive MVSBs method. This paper is aimed at providing a more effective recursive identification method than the ECRLS-MVSBs method developed in [12]. The new method should enjoy a much lower computational load than the batch MVSBs, and have a lower computational load than the ECRLS-MVSBs method. In this paper, a new substitution based multivariate simplex B-splines (SB-MVSBs) method is developed. The kernel-space bases based transformation can greatly cut down the computational time required by the SB-MVSBs method.

This paper is structured as follows. The preliminaries on the multivariate simplex B-splines are introduced in section 2. The SB-MVSBs method is developed in section 3. In section 4, the proposed SB-RMVSBs method is applied to a wind tunnel data set of the F-16 fighter aircraft, and the selection of the spline function structure is investigated. Subsequently, the proposed method is compared with both the batch method and the ECRLS-MVSBs method in section 5. Finally, this paper is concluded by section 6.

2 Preliminaries on Multivariate Simplex B-splines

The basic principles for simplex splines are briefly introduced in this section. Without this introduction, the formulation of the SB-MVSBs method will be incomplete.

2.1 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

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

which follows from the following implicit relation:

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

2.2 Triangulations of Simplices

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

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

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 [8].

2.3 Basis Functions of the Simplex B-splines

According to [3] and [11], the Bernstein basis polynomial $B^d_\kappa(b(\mathbf{x}))$ 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 [2], it was proved that any polynomial $p(b)$ 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:

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

with c_κ^t 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.

2.4 Vector Formulations of the B-form

As introduced in [12], 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[12].

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:

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

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

For a single observation on y we have:

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

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

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

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 $\xi = [\varepsilon_1, \varepsilon_2, ...\varepsilon_N]^T$ the residue vector. For writing convenience, $\mathbf{X}(b(\mathbf{x}))$ will be written as $\mathbf{X}$ in the remainder of this paper.

2.5 Global continuity constraints

The formulation for the continuity conditions from [1] and [3] is used:

$$\mathbf{c}^{t_i}_{(\kappa_0,\dots,\kappa_{n-1},m)} = \sum_{|\gamma|=m} \mathbf{c}^{t_j}_{(\kappa_0,\dots,\kappa_{n-1},0)+\gamma} B^m_\gamma(\mathbf{v}), \quad 0 \le m \le r \tag{13}$$

with $\mathbf{v}$ the Bernstein coordinates of the vertex which only belongs to the i^{th} simplex, $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_n)$ a multi-index independent of κ, $|(\kappa_0, \dots, \kappa_{n-1}, m) + \gamma| = d$. t_i, t_j denote the i-th and j-th simplices separately.

Eventually, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H} \cdot \mathbf{c} = 0 \tag{14}$$

with $\mathbf{H} \in \mathbb{R}^{(E\cdot R)\times(J\cdot\hat{d})}$ the smoothness matrix [11], 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) and Eq. (14).

2.6 Spline Function Space and a Polynomial Function Space

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

$$P_d(n) := \{p_k(\mathbf{x}) : p_k|_{\mathbf{x}} \in \mathbb{P}_k, \ \forall \mathbf{x} \in \mathbb{R}^n \ and \ \forall k \le d\} \tag{15}$$

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 [3]:

$$S_d^m(n) := \{s_d^m(\mathbf{x}) \in C^m : s_d^m|_{\mathbf{x}} \in \mathbb{P}_d,\ \forall \mathbf{x} \in \mathbb{R}^n\} \tag{16}$$

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 $P_2(2) := c_1 + c_2 x + c_4 y + c_3 x^2 + c_6 xy + c_5 y^2$ with x and y two elements of $\mathbf{x}$.

3 Transformation based recursive identification method

The kernel space information of the equality constraint matrix $\mathbf{H}$, formulated in Eq. (14), has been utilized to transform the constrained recursive identification problem into a free-of-constraint recursive identification problem.

3.1 Transformation of constraints

Once the triangulation and the spline function structure are chosen, the equality constraints have the property that they are time invariant and known a priori. In this case, a straightforward substitution method can be applied to remove the constraints in each recursion step.

Following from Eq. (8), the original constrained recursive identification problem has the following expression:

$$f = \mathbf{B} \cdot \mathbf{c} + \varepsilon \tag{17}$$

$$\text{s.t. } \mathbf{H} \cdot \mathbf{c} = 0 \tag{18}$$

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

$$\mathbf{H}_{n\times m} = \mathbf{V}_{n\times n} \begin{bmatrix} \sum_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)} \end{bmatrix} \mathbf{U}_{m\times m}^T \tag{19}$$

where $\sum = \text{diag}\left(\sigma_1\ ,\ \cdots\ \sigma_r\right)$ is the diagonal vector of all singular values , $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and r is the rank of $\mathbf{H}$. $\mathbf{V} = \left[\mathbf{V}_1\ \mathbf{V}_2\right]$ is an n_{th} order orthogonal matrix, $\mathbf{V}_1$ is an n by r matrix. $\mathbf{U} = \left[\mathbf{U}_1\ \mathbf{U}_2\right]$ is a m_{th}-order orthogonal matrix, $\mathbf{U}_1$ is an m by r matrix. Because $\mathbf{c} \in null(\mathbf{H})$, one feasible general solution for the homogeneous equation Eq. (18) is:

$$\mathbf{c} = \mathbf{U}_2 \mathbf{y} \tag{20}$$

where the column vectors of $\mathbf{U}_2$ form an orthonormal basis of $null\,(\mathbf{H})$ [14, 5]. $\mathbf{y}$ is a column vector which needs to be calculated (identified) later, and its length is $m-r$. The feasibility of the above mentioned conversion will be proved later in theorem 1.

By introducing this general solution into Eq. (17), we get the following formation:

$$f = \mathbf{B}\mathbf{U}_2\mathbf{y} + \varepsilon \tag{21}$$

with $\mathbf{U}_2$ a basis for $null\,(\mathbf{H})$. Since Eq. (21) only represents an unconstrained identification problem, a regular recursive least squares identification method becomes capable to solve it. In order to obtain the final unknown parameters (B-coefficients), we only need to substitute the identified vector $\mathbf{y}$ into Eq. (20). The computational flow chart is concluded as follows.

Algorithm 1:

step.1 determine the triangulation $\mathscr{T}$, calculate the smoothness matrix $\mathbf{H}$, and carry out the SVD according to Eq. (19) to get $\mathbf{U}_2$.

step.2 calculate the spline basis vector according to Eq. (9).

step.3 identify the unknown vector $\mathbf{y}$ contained by Eq. (21) using regular recursive least squares method.

step.4 reconstruct the B-coefficient vector $\mathbf{c}$ from the vector $\mathbf{y}$ using Eq. (20). Return to step.2 if a new data is available.

Theorem 1: Optimal approximation

Given $\mathbf{y}$ the unique and optimal least square estimation vector of problem Eq. (21), $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ is the optimal least squares solution of the constrained problem Eq. (17).

Proof:

Given $\mathbf{U}_2$ derived from Eq. (19), columns of matrix $\mathbf{U}_2$ constitute orthonormal bases for the kernel space of $\mathbf{H}$. Therefore, we have $\mathbf{H}\mathbf{U}_2 = 0$. Hence, we can get $\mathbf{H}\mathbf{U}_2 \cdot \mathbf{y} = 0$. Because $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ as shown in Eq. (20), we can get $\mathbf{H} \cdot \mathbf{c} = 0$. The equality constraints $\mathbf{H} \cdot \mathbf{c} = 0$ are satisfied during parameter estimation.

Because Eq. (17) and Eq. (21) hold, we have

$$\mathbf{f} - \mathbf{X} \cdot \mathbf{c} = \xi = \mathbf{f} - \mathbf{X}\mathbf{U}_2\mathbf{y} \tag{22}$$

We define the cost function of the least square problem as $C\,(\mathbf{c}) = \min\limits_{\mathbf{c}} \xi^T \xi$, where $\mathbf{c}$ is the vector to estimate. As $\mathbf{y}$ is the optimal and unique least square solution of problem 21, we assume that it leads to a minimum residual vector ξ_d, so the minimum cost function value can be written as $C\,(\mathbf{y}) = \xi_d^T \xi_d$. Because the two problems described by Eq. (17) Eq. (18) and Eq. (21) are identical systems in view of the output approximation, we can get the following result: $C\,(\mathbf{c}) = C\,(\mathbf{y}) = \xi_d^T \xi_d$ from Eq. (22). □

3.2 Remarks

Note that, according to Eq. (21), the proposed recursive identification method has cut down the scale of the original identification problem by multiplying the regression data matrix by $\mathbf{U}_2$ from the right hand side.

There exist some similarities between the SB-MVSBs method and the orthogonal least squares based identification method presented in [9]. In theory, the singular value decomposition allows to reduce the structure of the aerodynamic model. By keeping all (non-zero) singular values, the SB-MVSBs method has removed the dependent columns in the data matrix. However, it is not reasonable to cut out the smallest singular values and further reduce the scale of the model because the constraints are originally added to the unknown parameters rather than to the regression data matrix.

4 Validation using wind tunnel data of the F-16 fighter aircraft

4.1 F-16 Aerodynamic Model Structure

According to the F-16 aerodynamic wind tunnel data presented in [6], the following structure is a good option for X-direction aerodynamic force (moment) coefficient:

$$\begin{aligned} F_x\left(\alpha,\beta,\delta_e,\delta_{lef},\frac{q\bar{c}}{V}\right)=&f_1(\alpha,\beta,\delta_e)+f_2(\alpha,\beta)\cdot\delta_{lef} \\ &+f_3(\alpha)\cdot\frac{q\bar{c}}{V}+f_4(\alpha)\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef} \end{aligned} \tag{23}$$

Note that the engine thrust is assumed to be constant and its related term is removed from Eq. (23). According to Eq. (23), once the q, V and δ_{lef} are fixed, we can derive the following linear regression formulation for a three dimensional MVSBs function.

$$S(x)=\mathbf{B}\cdot\mathbf{c} \tag{24}$$

where $\mathbf{B}$ is the B-form spline vector calculated using Eq. (9).

According to de Visser [13], the global continuity matrix $\mathbf{H}$ for the three dimensional MVSBs function should be calculated using Eq. (13).

In the simulation, an aerodynamic model of the F-16 aircraft was identified using simulated flight test data generated with a nonlinear F-16 simulator based on a NASA wind tunnel dataset [6]. The training inputs of the simulated flight test dataset were obtained by generating 20,000 uniformly distributed inputs within their own valid regions. The inputs of the test dataset containing 4331 points are produced by the grids determined by α and β. The system output were calculated through the high resolution interpolation from the wind tunnel data provided by [6] with $\delta_{lef}=1^\circ, V=600\,ft/s$, $q=0.1\,rad/s$, $\bar{c}=11.32\,m$. Moreover, the model outputs of the aerodynamic model is contaminated artificially by adding a white noise with a magnitude of 1% (relative to its maximum and minimum value).

4.2 Cross validation results in determining the structure

In the numerical simulation, we have chosen the MVSBs function to have only one three dimensional sub-function. The notation $S_d^m(n)$ from Sec. 2 has been used, and the overall spline function becomes the following expression:
$S(x) = S_d^m(n)$, where $n = 3$, while d, m are kept undetermined. The partitioning vector of α is $[-20\ 10\ 40]$. The partitioning vector of β is $[-25\ 25]$. The partitioning vector of δ_e is $[-20\ 20]$. In order to enhance the approximation ability of this algorithm, all the inputs are normalized into the closed range of $[0\ 1]$. In order to select a suitable structure for the spline model of C_m (i.e. the nondimensional pitch moment coefficient), the effects of the structural parameters (i.e. d and m) will be investigated. To demonstrate the approximation power of the SB-MVSBs method, we compared it with the batch MVSBs method.

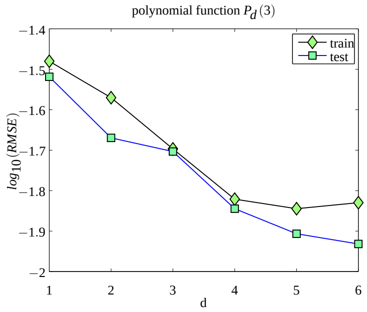
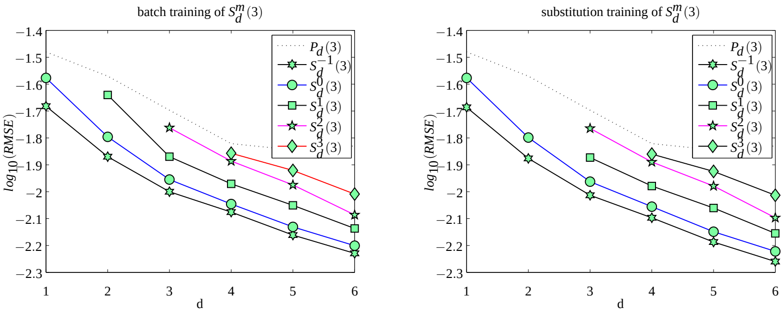


Fig. 1 Different selection of d for $P_d(3)$, C_m.

Fig. 1 shows the root mean squared errors (RMSE) of the fitting outputs (C_m) using the ordinary polynomial basis (OPB) based recursive least squares identification method.

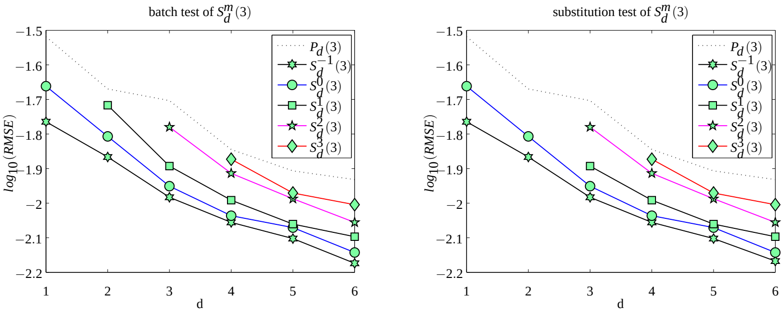
Fig. 2(a) and Fig. 2(b) show the RMSE of the training data set using the batch MVSBs method and the proposed SB-MVSBs method respectively. Comparing these two figures, it has been found that the SB-MVSBs method enjoys the same level of approximation power as that of the batch MVSBs.

Fig. 3(a) and Fig. 3(b) show the RMSE of the testing data set based on the B-coefficients identified using the batch MVSBs method and the SB-MVSBs method respectively. As can be seen from these two figures, the approximation power of the batch MVSBs method and the SB-MVSBs method are very close. Moreover, compared with the results shown in Fig. 1, Fig. 3 indicate that both the batch MVSBs



(a) batch MVSBs training. (b) SB-MVSBs training.

Fig. 2 Different combination of m and d for $S_d^m(3)$, $\mathscr{T}_{12}$, C_m.



(a) batch MVSBs test. (b) SB-MVSBs test.

Fig. 3 Different combination of m and d for $S_d^m(3)$, $\mathscr{T}_{12}$, C_m.

method and the SB-MVSBs method enjoy a higher approximation power than the OPB based recursive identification method.

5 Comparison with the ECRLS-MVSBs and the batch MVSBs

5.1 *Computational Complexity*

The computational complexity of the substitution based MVSBs (SB-MVSBs) method is split into two parts. Firstly, according to Eq. (21), the multiplication between the **B** vector and the $\mathbf{U}_2$ matrix needs $m \cdot (m-r)$ with r the rank of the continuity matrix, and m the length of the B-coefficient vector **c**. Similar to the ECRLS method, the computational complexity for the pure regression process using the recursive least squares is $\mathscr{O}\left(3(m-r)^2\right)$. By summing them up, we can get the total computational complexity of the SB-MVSBs method: $C(m,r) =$

$(m-r)\cdot(4m-3r) = 3r^2 - 7mr + 4m^2$. The computational complexity in time of the batch MVSBs method, the ECRLS-MVSBs method and the SB-MVSBs method are tabulated in Table 1.

Given m, function $C(m,r)$ monotonously increases as $r < m$. Therefore the minimum computational complexity of the SB-MVSBs method is $4m^2$ when $r = 0$, while its highest limit is 0. In addition, $C(m,r) = 3n^2$ holds when $r = \frac{(7-\sqrt{37})}{6}m$.

Table 1 Computational Complexity (CC) in time

Methods	batch MVSBs	ECRLS-MVSBs	SB-MVSBs
CC	$\mathscr{O}(m^3)$	$\mathscr{O}(3m^2)$	$(m-r)\cdot(4m-3r)$

5.2 Computational time comparison with the ECRLS-MVSBs

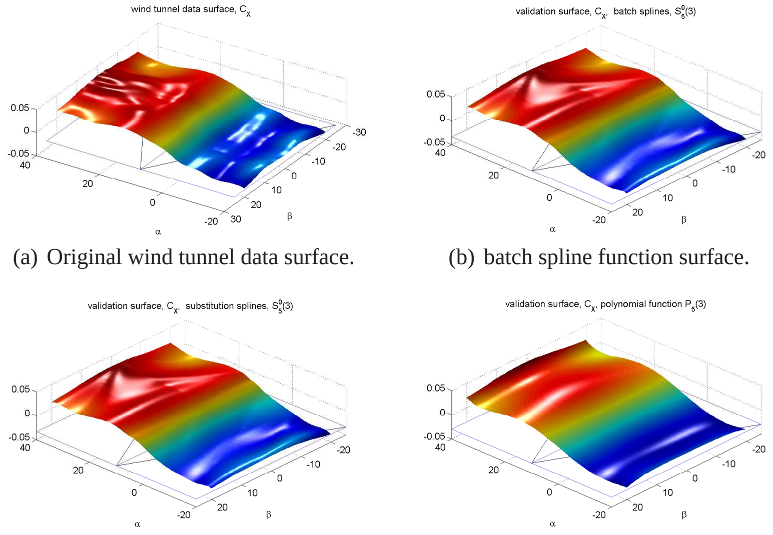
Table 2 Computational time for 20k data of C_m, $\mathscr{T}_{12}$, B-coefficient number 1008, $S_6^m(3)$

condition	$S_6^{-1}(3)$	$S_6^0(3)$	$S_6^1(3)$	$S_6^2(3)$	$S_6^3(3)$	$S_6^4(3)$
ECRLS	104.5092	105.5291	105.0324	106.2780	106.2854	106.6970
SB-MVSBs(operated)	101.7709	33.2808	13.4270	5.3797	4.2410	3.7263
SB-MVSBs(normal)	139.4835	67.7009	24.0644	7.9068	6.0565	5.6464

In order to reveal the influence of the continuity order m on the computational complexity in time, a numerical experiment is performed with different selection for the continuity order m. In the remainder of this paper, we will always choose the MVSBs function to have only one three dimensional sub-function in all of the numerical experiments. The simulation results are listed in Table 2. In Table 2, '*operated*' means that the $\mathbf{BU}_2$ multiplication shown in Eq. (21) is executed in advance in a batch manner. According to Table 2, the SB-MVSBs method require less computational time than the ECRLS-MVSBs method, and this advantage will become more apparent with the increase of the continuity order m.

5.3 Evaluation results on the approximation power

The OPB based recursive identification method, the batch MVSBs method and the SB-MVSBs recursive identification method are utilized to fit the same training data set of C_x respectively. The models identified using these three different methods respectively are validated using the testing data that are located on the mesh grids. The



(a) Original wind tunnel data surface.

(b) batch spline function surface.

(c) substitution spline function surface.

(d) Polynomial fitting surface.

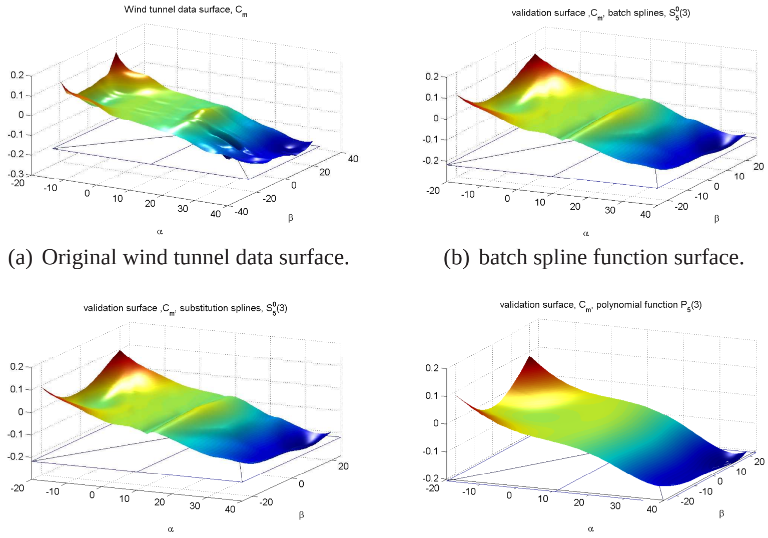
Fig. 4 Validation surface of C_x ($\delta_e = 2^o$), $\mathscr{T}_{12}$.

validation surfaces of C_x are shown in Fig. 4. Apparently, the SB-MVSBs method enjoys an equal fitting accuracy to that of the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.

The OPB based recursive identification method, the batch MVSBs and the SB-MVSBs recursive identification methods are utilized to fit the same training data set of C_m. The models identified using three different methods are validated using the same testing data set as that mentioned previously. The validation surfaces of C_m are plotted in Fig. 5. We can get a similar conclusion as that drawn from last experiment that the SB-MVSBs method has the same fitting power as the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.

6 Conclusions

A new substitution based recursive MVSBs method is proposed for the online aerodynamic model identification. In view of the equality constraints contained by the MVSBs, a SVD based transformation is empoyed to convert an originally constrained recursive identification problem into a free-of-constraint identification problem. The proposed recursive model identification method namely SB-MVSBs method was applied to approximate a series of two wind tunnel data sets of F-16 aircraft, and were compared with the batch MVSBs method and the ECRLS-



(a) Original wind tunnel data surface.

(b) batch spline function surface.

(c) substitution spline function surface.

(d) Polynomial fitting surface.

Fig. 5 Validation surface of C_m ($\delta_e = 2^o$), $\mathscr{T}_{12}$.

MVSBs method. The numerical simulation results show that the proposed SB-MVSBs method requires less computational time than the batch MVSBs method and the ECRLS-MVSBs method. In addition, the computational time required by the SB-MVSBs decreases with the increase of the continuity order m. The reduction of the computational time is caused by the fact that the kernel space bases based transformation has cut down the scale of the original spline basis based model.

References

1. Awanou, G., Lai, M.J., Wenston, P.: The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differencial Equations. Wavelets and Splines (2005)
2. de Boor, C.: B-form basics. In:G. Farin, editor, Geometric modeling: algorithms and new trends. Society for Industrial and Applied Mathematics (1987)
3. Lai, M.J., Schumaker, L.L.: Spline Functions on Triangulations. Cambridge (2007)
4. Lombaerts, T.J., Smaili, M.H., Stroosma, O.: Piloted simulator evaluation results of new fault-tolerant flight control algorithm. Journal of Guidance, Control and Dynamics **32**(6), 1747–1765 (2009)
5. Meyer, C.D.: Matrix Analysis and Applied Linear Algebra. SIAM:Society for Industrial and Applied Mathematics (2001)
6. Nguyen, L.T., Ogburn, M.E., 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 (1979)
7. van Oort, E.R., Sonneveldt, L., Chu, Q.P., Mulder, J.A.: Full-envelope modular adaptive control of a fighter aircraft using orthogonal least squares. Journal of Guidance, Control and Dynamics **33**(5), 1461–1472 (2010)

8. Shewchuk, J.R.: Delauney refinement algorithms for triangular mesh generation. Computational Geometry **22**, 21–74 (2001)
9. Stark, J.: Adaptive model selection using orthogonal least squares methods. In: Mathematical, Physical and Engineering Sciences, pp. 21–42. The Royal Society, Britain (1997)
10. de Visser, C.: Global nonlinear model identification with multivariate splines. Ph.D. thesis, Delft University of Technology, The Netherlands (2011)
11. de Visser, C.C., Chu, Q.P., Mulder, J.A.: A new approach to linear regression with multivariate splines. Automatica Journal **45**, 2903–2909 (2009)
12. de Visser, C.C., Chu, Q.P., Mulder, J.A.: Differential constraints for bounded recursive identification with multivariate splines. Automatica Journal **47**, 2059–2066 (2011)
13. de Visser, C.C., Mulder, J.A., Chu, Q.: Global nonlinear aerodynamic model identification with multivariate splines. In: AIAA Atomospheric Flight Mechanics Conference. AIAA, Chicago, Illinois (2009)
14. Zhang, X.D.: Matrix Analysis and Applications. Tsinghua University Press (2004)
15. Zhu, Y., Li, X.R.: Recursive least squares with linear constraints. Communications in Information and Systems **7**(3), 287–312 (2007)