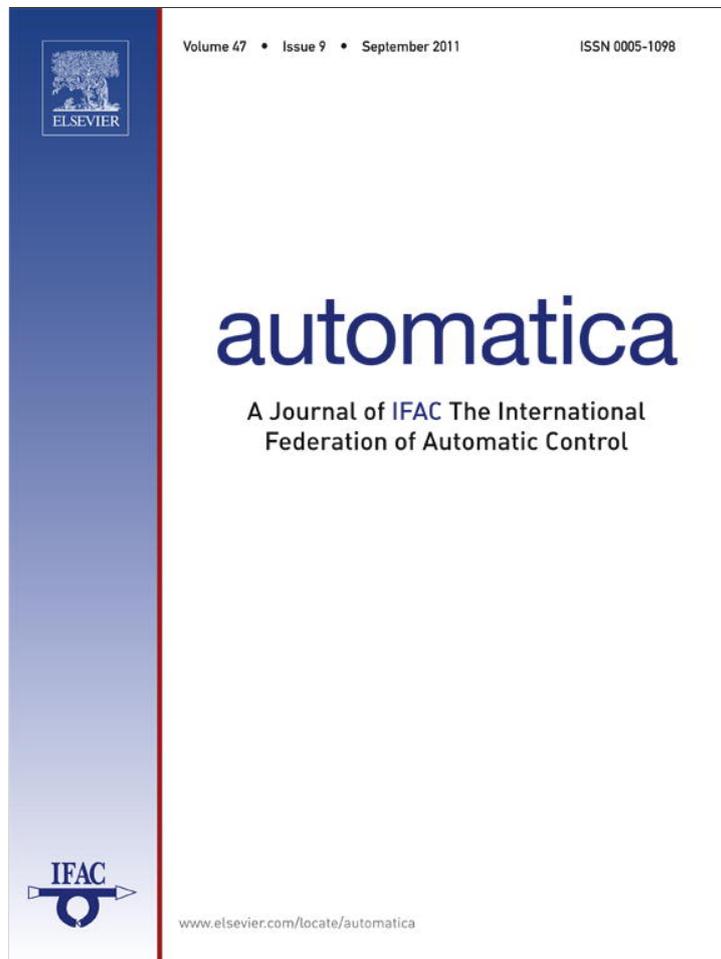


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Differential constraints for bounded recursive identification with multivariate splines[☆]

C.C. de Visser^{*}, Q.P. Chu, J.A. Mulder

Control and Simulation Division, Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600GB Delft, The Netherlands

ARTICLE INFO

Article history:

Received 11 April 2010

Received in revised form

21 February 2011

Accepted 8 March 2011

Available online 2 July 2011

Keywords:

Multivariate splines

Parameter estimation

Scattered data

Function approximation

Estimators

ABSTRACT

The ability to perform online model identification for nonlinear systems with unknown dynamics is essential to any adaptive model-based control system. In this paper, a new differential equality constrained recursive least squares estimator for multivariate simplex splines is presented that is able to perform online model identification and bounded model extrapolation in the framework of a model-based control system. A new type of linear constraints, the differential constraints, are used as differential boundary conditions within the recursive estimator which limit polynomial divergence when extrapolating data. The differential constraints are derived with a new, one-step matrix form of the de Casteljau algorithm, which reduces their formulation into a single matrix multiplication. The recursive estimator is demonstrated on a bivariate dataset, where it is shown to provide a speedup of two orders of magnitude over an ordinary least squares batch method. Additionally, it is demonstrated that inclusion of differential constraints in the least squares optimization scheme can prevent polynomial divergence close to edges of the model domain where local data coverage may be insufficient, a situation often encountered with global recursive data approximation.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Accurate models of dynamic systems are essential to any model-based automatic control system, like for example nonlinear dynamic inversion (NDI) control (Reiner, Balas, & Garrard, 1996). While robust control techniques can be used to compensate for some model inaccuracies, they also reduce the reference signal tracking performance. When a system is subject to changing dynamics, however, robust control techniques no longer suffice. In that case a recursive system identification method must be used to adapt the system model based on incoming observations.

Recursive system identification and its algorithmic implementations have been well studied in the past, see e.g. Ljung (2002), Ljung and Söderström (1983) and Söderström, Ljung, and Gustavsson (1978). Current trends in recursive system identification point towards nonlinear recursive identification methods like recursive neural networks (Mirikitani & Nikolaev, 2010) and direct weighting methods (Roll, Nazin, & Ljung, 2005). A promising new direction in recursive identification would be a method based on multivariate simplex splines. Multivariate simplex splines are a

recent type of multivariate polynomial spline which are capable of fitting scattered multidimensional datasets on non-rectangular domains (Awanou, Lai, & Wenston, 2005; Lai & Schumaker, 2007). Multivariate simplex splines have a number of advantages over other methods. Firstly, the simplex splines have an arbitrarily high approximation power on a global model scale. Secondly, simplex spline models are parametric models, which allows for efficient approximation of very large datasets. Thirdly, the simplex splines are linear in the parameters, which means that linear regression methods can be used for their estimation (de Visser, Chu, & Mulder, 2009). Finally, the simplex splines have a local polynomial basis, which implies that only small subsets of parameters and basis polynomials need to be considered during estimation and evaluation, resulting in efficient (sparse) computational schemes. Recently, these capabilities were demonstrated in large scale image processing (Kalbe, Koch, & Goesele, 2009) and aerodynamic model identification (de Visser, Mulder, & Chu, 2010) applications.

The objective of this paper is the presentation of a differentially constrained recursive least squares (RLS) estimator for the B -coefficients of multivariate simplex splines. This estimator can be used for online nonlinear system identification in an adaptive model-based automatic control system, see Fig. 1. In this paper a new type of linear constraint is introduced in the form of the differential constraints. The differential constraints are based on a new, one-step matrix form of the well-known de Casteljau algorithm. The differential constraints are used to constrain the directional derivatives of the simplex splines at selected

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonio Vicino under the direction of Editor Torsten Söderström.

^{*} Corresponding author. Tel.: +31 527 87147; fax: +31 527 86480.

E-mail addresses: c.c.devisser@tudelft.nl (C.C. de Visser), q.p.chu@tudelft.nl (Q.P. Chu), j.a.mulder@tudelft.nl (J.A. Mulder).

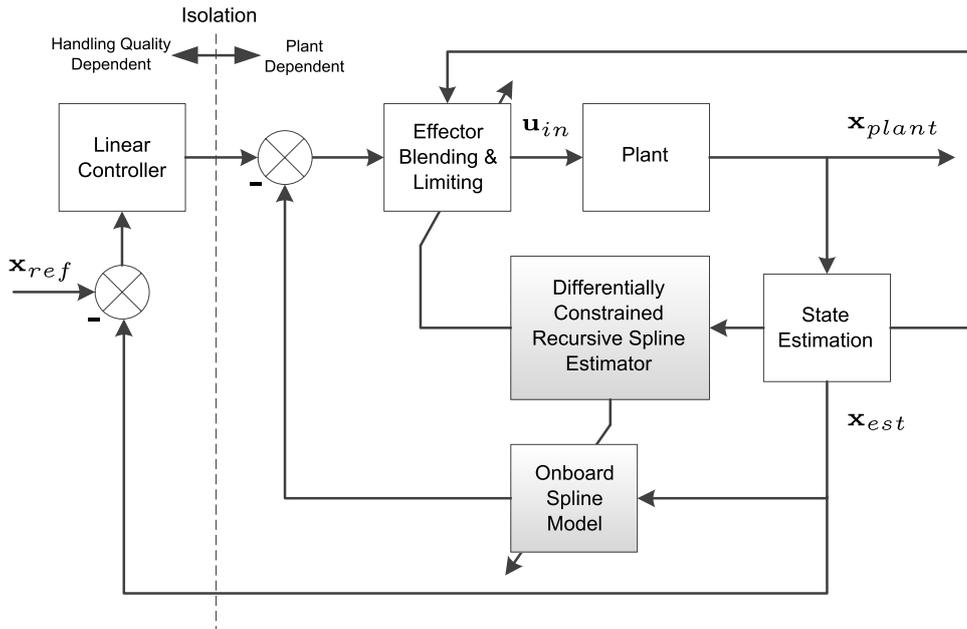


Fig. 1. Control loop for an adaptive multivariate spline model-based controller.

locations within the spline domain. The differential constraints can be applied to impose Dirichlet, Neumann, Robin or Cauchy boundary conditions on the simplex spline functions. This is essential if simplex splines are used to approximate solutions to (partial) differential equations. From an engineering perspective, the differential constraints can be applied to facilitate bounded model extrapolation by limiting polynomial divergence beyond the bounds of the data domain. This is achieved through the constraining of the directional derivatives of the simplex spline polynomials based on expert knowledge of the modeled system. This is very useful during real-life recursive identification with simplex splines because new observations may be too scarce to provide a sufficient local data coverage, requiring local data extrapolation to maintain the global model.

The differentially constrained RLS estimator for simplex splines was demonstrated on a nonlinear bivariate dataset which required local data extrapolation. Cauchy boundary conditions were imposed on the boundaries of the spline domain using the differential constraints. It was found that the differential constraints not only virtually eliminate polynomial divergence in areas where the data is extrapolated, but also significantly reduce the parameter variance of B -coefficients located near these areas. The differential constraints effectively increase the performance of the recursive estimator, which is an application of linear constraints that has been recognized in the literature (Mahata & Söderström, 2004).

2. Preliminaries on multivariate simplex splines

In this section a brief introduction on the mathematical theory of the multivariate simplex splines is given. A more complete introduction can be found in de Visser et al. (2009).

2.1. The simplex and barycentric coordinates

Let t be an n -simplex formed by the convex hull of its $n + 1$ non-degenerate vertices $(v_0, v_1, \dots, v_n) \in \mathbb{R}^n$ as follows. The normalized barycentric coordinate $b(x) = (b_0, b_1, \dots, b_n)$ of some point $x \in \mathbb{R}^n$ with respect to t follows from the implicit relation:

$$x = \sum_{i=0}^n b_i v_i, \quad \sum_{i=0}^n b_i = 1. \quad (1)$$

2.2. Triangulations of simplices

The high approximation power of the multivariate simplex spline comes from the combining of many simplices into a structure called a triangulation. A triangulation \mathcal{T} is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathcal{T} := \bigcup_{i=1}^J t_i, \quad t_i \cap t_j \in \{\emptyset, \tilde{t}\}, \quad \forall t_i, t_j \in \mathcal{T}, \quad (2)$$

with the edge simplex \tilde{t} a k -simplex with $0 \leq k \leq n - 1$.

Creating high quality triangulations is not a trivial task, especially in higher dimensions. High quality triangulations can be provided by constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented by Shewchuk (2001). More recently, Shewchuk presented a tentative method for n -dimensional CDT (Shewchuk, 2008) which would be very useful in n -dimensional scattered data modeling with simplex splines.

2.3. Basis functions of the simplex spline

The Bernstein basis polynomial $B_\kappa^d(b)$ of degree d in terms of the barycentric coordinate $b = (b_0, b_1, \dots, b_n)$ from (1) is:

$$B_\kappa^d(b) := \frac{d!}{\kappa!} b^\kappa, \quad (3)$$

with $\kappa = (\kappa_0, \kappa_1, \dots, \kappa_n) \in \mathbb{N}^{n+1}$ is a multi-index with the following properties: $\kappa! = \kappa_0! \kappa_1! \dots \kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + \dots + \kappa_n$. In (3) we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} \dots b_n^{\kappa_n}$. When it is given that $|\kappa| = d$, the total number of valid permutations of the multi-index κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!}. \quad (4)$$

De Boor proved that the Bernstein basis polynomials (3) form a stable local basis for the space of polynomials of degree d (de Boor, 1987). 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:

$$p(b) = \sum_{|\kappa|=d} c_\kappa^t B_\kappa^d(b), \quad (5)$$

with c_κ^t the B -coefficients which uniquely determine $p(b)$. The total number of basis function terms is equal to \hat{d} , which is the total number of valid permutations of κ from (4).

2.4. Vector formulations of the B -form

In de Visser et al. (2009) a useful vector formulation of the B -form on a single simplex was introduced:

$$p(b) = \mathbf{B}^d(b) \cdot \mathbf{c}^t \in \mathbb{R}, \quad (6)$$

with $\mathbf{B}^d(b)$ a vector of Bernstein basis polynomials which are sorted lexicographically according to Hu, Han, and Lai (2007):

$$\mathbf{B}^d(b) = [B_\kappa^d(b)]_{|\kappa|=d} \in \mathbb{R}^{1 \times \hat{d}}. \quad (7)$$

and with \mathbf{c}^t the vector of lexicographically sorted B -coefficients on the simplex t :

$$\mathbf{c}^t = [c_\kappa^t]_{|\kappa|=d} \in \mathbb{R}^{\hat{d} \times 1}. \quad (8)$$

For example, for a trivariate B -form of degree $d = |\kappa| = 1$ we have $\kappa \in \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. In this case the B -form from (5) can be expressed in the vector form (6) as follows:

$$\begin{aligned} p(b) &= \sum_{|\kappa|=1} c_\kappa^t B_\kappa^1(b) = \mathbf{B}^1(b) \cdot \mathbf{c}^t \\ &= [B_{1,0,0}^1(b) \quad B_{0,1,0}^1(b) \quad B_{0,0,1}^1(b)] [c_{1,0,0}^t \quad c_{0,1,0}^t \quad c_{0,0,1}^t]^\top. \end{aligned}$$

3. A multi-degree matrix form of the de Casteljau algorithm

The de Casteljau algorithm was invented in 1959 by Paul de Casteljau for the efficient computation of Bézier curves. In this section we introduce a one-step form of the de Casteljau algorithm which does not require the recursion of the original algorithm. The one-step form can be translated into a matrix form which will prove to be essential in the formulation of the differential constraints for the B -coefficients.

3.1. A multi-degree formulation of de Casteljau recursion

The de Casteljau algorithm is a well-known recursive method for calculating the value of a polynomial in the B -form, see e.g. Farin (2002). The de Casteljau algorithm relates the value of the B -coefficient of iteration m to the B -coefficient of iteration $m - 1$ as follows:

$$c_\kappa^{(m)}(b) := \sum_{|\gamma|=1} b_\gamma c_{\kappa+\gamma}^{(m-1)}(b), \quad m \leq d, \quad (9)$$

with $c_\kappa^{(0)}(b) = c_\kappa$ and with $c_0^{(d)}(b)$ the value of the B -form polynomial at b . A multi-degree form of the de Casteljau algorithm will now be derived that relates $c^{(m)}$ with $c^{(m-q)}$ with $q \geq 1$. First, notice that the polynomial basis function $B_\kappa^d(b)$ from (3) can be expanded as follows:

$$\begin{aligned} B_\kappa^d(b) &= b_0 B_{\kappa_0-1, \kappa_1, \dots, \kappa_n}^{d-1}(b) + b_1 B_{\kappa_0, \kappa_1-1, \dots, \kappa_n}^{d-1}(b) + \dots \\ &\quad + b_n B_{\kappa_0, \kappa_1, \dots, \kappa_{n-1}}^{d-1}(b), \quad |\kappa| = d \\ &= \sum_{|\gamma|=1} b_\gamma B_{\kappa-\gamma}^{d-1}(b), \end{aligned} \quad (10)$$

with $\gamma = (\gamma_0, \gamma_1, \dots, \gamma_n)$ a multi-index dependent on κ as follows:

$$|\gamma| = d - |\kappa|. \quad (11)$$

In general, for any degree $m \leq d$ we have for a single basis function term:

$$\begin{aligned} B_\kappa^d(b) &= P_\gamma^m(b) B_{\kappa_0-m, \kappa_1, \dots, \kappa_n}^{d-m}(b) + P_\gamma^m(b) B_{\kappa_0, \kappa_1-m, \dots, \kappa_n}^{d-m}(b) \\ &\quad + \dots + P_\gamma^m(b) B_{\kappa_0, \kappa_1, \dots, \kappa_{n-m}}^{d-m}(b) \\ &= \sum_{|\gamma|=m} P_\gamma^m(b) B_{\kappa-\gamma}^{d-m}(b), \end{aligned} \quad (12)$$

with the basis function $P_\gamma^m(b)$ of degree m defined as follows:

$$P_\gamma^m(b) = \frac{m!}{\gamma!} b_\gamma^m. \quad (13)$$

It is important to note here that polynomial terms with negative multi-indices are not defined and equal to zero. Using (12) and (13) we can restate the B -form polynomial as follows:

$$p(b) = \sum_{|\kappa|=d} \left[c_\kappa \sum_{|\gamma|=m} P_\gamma^m(b) B_{\kappa-\gamma}^{d-m}(b) \right]. \quad (14)$$

When we let $|\kappa| = d - m$ this expression can be reformulated such that no negative multi-indices are produced:

$$p(b) = \sum_{|\kappa|=d-m} \sum_{|\gamma|=m} c_{\kappa+\gamma} P_\gamma^m(b) B_\kappa^{d-m}(b). \quad (15)$$

We will now introduce a theorem for the multi-degree de Casteljau algorithm, which is necessary for the definition of the one-step de Casteljau matrix.

Theorem 1. *The multi-degree de Casteljau algorithm, which relates the B -coefficients of iteration q with those of iteration $m + q$ is*

$$c_\kappa^{(q+m)}(b) = \sum_{|\gamma|=m} P_\gamma^m(b) c_{\kappa+\gamma}^{(q)}(b), \quad q \geq 0, \quad (16)$$

with $c_\kappa^{(0)}(b) = c_\kappa$ as in (9).

Proof. Let δ^i , $i = 1, 2, \dots, m$ be a set of q independent multi-indices. The ordinary de Casteljau algorithm then is:

$$c_\kappa^{(q+1)}(b) = \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\delta^1}^{(q)}(b), \quad q \geq 0. \quad (17)$$

The next iteration then is

$$\begin{aligned} c_\kappa^{(q+2)}(b) &= \sum_{|\delta^2|=1} b_{\delta^2} c_{\kappa+\delta^2}^{(q+1)}(b) \\ &= \sum_{|\delta^2|=1} b_{\delta^2} \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\delta^1+\delta^2}^{(q)}(b). \end{aligned} \quad (18)$$

Repeating this process and letting $\gamma = \sum_{i=1}^m \delta^i$ we get for the m th iteration:

$$c_\kappa^{(q+m)}(b) = \sum_{|\delta^m|=1} b_{\delta^m} \sum_{|\delta^{m-1}|=1} b_{\delta^{m-1}} \dots \sum_{|\delta^1|=1} b_{\delta^1} c_{\kappa+\gamma}^{(q)}(b). \quad (19)$$

Using the multinomial theorem the product of sums can be rewritten as follows:

$$c_\kappa^{(q+m)}(b) = \sum_{|\gamma|=m} \frac{d!}{\gamma!} b_\gamma^m c_{\kappa+\gamma}^{(q)}(b) \quad (20)$$

$$= \sum_{|\gamma|=m} P_\gamma^m(b) c_{\kappa+\gamma}^{(q)}(b). \quad (21)$$

The last equality follows from the definition of $P_\gamma^m(b)$ from (13) and proves the theorem. \square

3.2. The matrix form of the de Casteljau algorithm

We define the de Casteljau matrix function of degree m which reduces a set of B -coefficients of degree d into a set of B -coefficients of degree $d - m$ as follows:

$$\mathbf{P}^{d,d-m}(b) \in \mathbb{R}^{\hat{d}^* \times \hat{d}}, \quad (22)$$

with \hat{d}^* the total number of basis function terms for degree $d - m$ and dimension n :

$$\hat{d}^* = \frac{(d - m + n)!}{(d - m)!n!}. \quad (23)$$

As an example of the use of (22), consider the case of the de Casteljau matrix $\mathbf{P}^{3,1}(b)$ for $n = 1$. In this case, the de Casteljau matrix reduces the degree of an original set of B -coefficients from degree 3 to degree 1 as follows:

$$[c_{1,0}(b) \ c_{0,1}(b)]^\top = \mathbf{P}^{3,1}(b) \cdot [c_{3,0} \ c_{2,1} \ c_{1,2} \ c_{0,3}]^\top.$$

The structure of the de Casteljau matrix is defined as follows:

$$[\mathbf{P}^{d,d-m}(b)]_{i(\kappa),i(\theta)} = P_{\theta-\kappa}^m(b), \quad |\theta| = d, |\kappa| = d - m, \quad (24)$$

with $i(\kappa)$ and $i(\theta)$ index functions for the rows and columns of $\mathbf{P}^{d,d-m}(b)$, respectively. The index function $i(\kappa)$ for the rows of the de Casteljau matrix is defined as follows:

$$i(\kappa) = \sum_{|\gamma|=d-m} 1, \quad \gamma \leq \kappa, |\kappa| = d - m, \quad (25)$$

while the index function $i(\theta)$ for the columns of the de Casteljau matrix has the following definition:

$$i(\theta) = \sum_{|\gamma|=d} 1, \quad \gamma \leq \theta, |\theta| = d. \quad (26)$$

Before a vector form of the B -form based on the de Casteljau matrix can be derived, a proof is required for the following theorem.

Theorem 2. *Based on the results from Theorem 1, the following statement must hold for the de Casteljau matrix from (24):*

$$\mathbf{P}^{d,d-m}(b) \cdot \mathbf{c}^j = \left[\sum_{|\gamma|=m} P_{\gamma}^m(b) c_{\kappa+\gamma} \right]_{|\kappa+\gamma|=d}, \quad (27)$$

where the right hand term of this statement follows from (16) using $q = 0$.

Proof. The proof starts by reformulating (27) in a form that permits negative multi-indices. For this, we let $\theta = \kappa + \gamma$ which can be substituted in the right hand side of (27) resulting in:

$$\mathbf{P}^{d,d-m}(b) \cdot \mathbf{c}^j = \left[\sum_{|\theta|=d} P_{\theta-\kappa}^m(b) c_{\theta} \right]_{|\kappa|=d-m}. \quad (28)$$

Using the definition of the structure of $\mathbf{P}^{d,d-m}(b)$ from (24) and the definitions of the index functions from (25) and (26), the following must hold for every single element of the de Casteljau matrix, and thus for every value of κ and θ :

$$[\mathbf{P}^{d,d-m}(b)]_{i(\kappa),i(\theta)} \cdot \mathbf{c}_{i(\theta)}^j = P_{\theta-\kappa}^m(b) \cdot \mathbf{c}_{i(\theta)}^j \quad (29)$$

$$= P_{\theta-\kappa}^m(b) \cdot c_{\theta}, \quad (30)$$

as long as $|\theta| = d$ and $|\kappa| = d - m$. Therefore, we get for the multiplication of a single row of the de Casteljau matrix with the vector of B -coefficients:

$$[\mathbf{P}^{d,d-m}(b)]_{i(\kappa),\bullet} \cdot \mathbf{c}^j = \sum_{|\theta|=d} P_{\theta-\kappa}^m(b) c_{\theta}. \quad (31)$$

Finally, the multiplication of all rows of the de Casteljau matrix with the vector of B -coefficients immediately results in (28), thus proving the theorem. \square

With Theorem 2 a special vector form of the B -form using the de Casteljau matrix can be derived. This vector form will be used to define the directional derivatives of B -form polynomials in terms of the original vector of B -coefficients, as will be shown in Section 4. Substitution of the B -coefficient vector from (8), the vector form of the basis function from (7), and the de Casteljau matrix from (24) in (15) results in the vector form of the de Casteljau B -form of degree d on a single simplex t_j :

$$p(b) = \mathbf{B}^{d-m}(b) \mathbf{P}^{m,d-m}(b) \cdot \mathbf{c}^j. \quad (32)$$

It is easy to check that this equation reduces to the ordinary vector form of the B -form from (6) by letting $m = 0$.

As an example of the above, we will proceed with the derivation of de Casteljau matrix $\mathbf{P}^{3,1}(b)$ for $n = 1, d = 3$ and $d - m = 1$. In this case we have $\kappa \in \{(1, 0), (0, 1)\}$ and $\theta \in \{(3, 0), (2, 1), (1, 2), (0, 3)\}$. Using (24) the de Casteljau matrix then is:

$$\mathbf{P}^{3,1}(b) = \begin{bmatrix} P_{2,0}^2(b) & P_{1,1}^2(b) & P_{0,2}^2(b) & 0 \\ 0 & P_{2,0}^2(b) & P_{1,1}^2(b) & P_{0,2}^2(b) \end{bmatrix}.$$

The de Casteljau matrix for a complete triangulation is constructed by placing per-simplex de Casteljau matrix blocks on the main diagonal of \mathbf{P} :

$$\mathbf{P} = \text{diag}([\mathbf{P}^{d,d-m}(b)]_j, j = 1, 2, \dots, J). \quad (33)$$

4. Differential constraints for B -coefficients

The differential constraints are the main new contribution of this paper. Differential constraints locally constrain the directional derivatives of the spline polynomials. The differential constraints can be used, for example, to limit polynomial divergence in areas of the spline domain with insufficient data coverage or to enforce differential boundary conditions on the edges of the spline domain. In order to define differential constraints on the B -coefficients of a spline function, an expression is required for the directional derivatives of the simplex splines in terms of the original vector of B -coefficients. As we will show in this section, the derivation of the directional derivatives in terms of the original B -coefficients is based on the one-step matrix form of the de Casteljau algorithm introduced in Section 3.

4.1. The directional derivative of B -form polynomials

The directional derivatives of B -form polynomials are well known in the literature, see e.g. Lai and Schumaker (2007). The m th order derivative of the B -form polynomial $p(b)$ on a single simplex in the direction $u \in \mathbb{R}^n$ is:

$$D_u^m p(b) = \frac{d!}{(d - m)!} \sum_{|\kappa|=d-m} c_{\kappa}^{t,(m)}(a) B_{\kappa}^{d-m}(b), \quad (34)$$

with $c_{\kappa}^{t,(m)}(a)$ the m th de Casteljau iteration of the B -coefficient c_{κ}^t on the n -simplex t . In (34), a is a directional coordinate which is the barycentric representation of u with respect to the simplex t . That is, if $u = v - w$, with v and w vectors in \mathbb{R}^n , then a is defined as follows:

$$a := b(v) - b(w) \in \mathbb{R}^{n+1}, \quad (35)$$

with $b(v)$ and $b(w)$ the barycentric coordinates of v and w with respect to t , respectively.

4.2. The matrix form of the directional derivative

In Section 3 a one-step matrix form of the de Casteljau algorithm was presented. It will be proven here that this matrix

form allows the directional derivatives of a B -form polynomial of order m to be expressed in terms of the original B -coefficient vector rather than in the m th de Casteljau iteration of the original B -coefficients.

Theorem 3. *The matrix form of the directional derivative of order m of the B -form polynomial $p(b)$ in the direction u in terms of the original B -coefficients is:*

$$D_u^m p(b) = \frac{d!}{(d-m)!} \mathbf{B}^{d-m}(b) \mathbf{P}^{d,d-m}(a) \cdot \mathbf{c}^t, \quad (36)$$

with $\mathbf{P}^{d,d-m}(a)$ the de Casteljau matrix from (24) of degree d to $d-m$ expressed in terms of the directional coordinate a of u , with $\mathbf{B}^{d-m}(b)$ the vector form of the basis polynomials, and with \mathbf{c}^t the vector of B -coefficients for a single n -simplex t .

Proof. Starting with the m th order directional derivative of the B -form from (34), and substituting the multi-degree de Casteljau form from (16) with $q = 0$ and the directional coordinate a we get:

$$D_u^m p(b) = \frac{d!}{(d-m)!} \sum_{|\kappa|=d-m} \left(\sum_{|\gamma|=m} P_\gamma^m(a) c_{\kappa+\gamma}^t \right) B_\kappa^{d-m}(b). \quad (37)$$

The theorem is proved by using Theorem 2 to replace $\sum_{|\gamma|=m} P_\gamma^m(a) c_{\kappa+\gamma}^t$ with its matrix form and by substituting the vector form of the polynomial basis function of degree $d-m$ from (7). \square

4.3. Differential constraints

The general matrix form of the directional derivative allows for the definition of differential constraints using the original set of B -coefficients, as we will show in this section. These differential constraints can be included in a Karush–Kuhn–Tucker (KKT) solution system (de Visser et al., 2009) for the B -coefficients or the equivalent null-space solution system from (47). For this, the differential constraints are formulated as follows:

$$\mathbf{D} \cdot \mathbf{c} = \mathbf{d}, \quad (38)$$

with \mathbf{D} a block diagonal matrix holding M differential constraints for the optimization problem. The vector $\mathbf{c} = [\mathbf{c}^t]_{j=1}^J \in \mathbb{R}^{J \cdot \hat{d} \times 1}$ is the global vector of B -coefficients, while $\mathbf{d} \in \mathbb{R}^{M \times 1}$ is the vector of differential constraint values.

The blocks of \mathbf{D} are formed by evaluating a matrix function $\mathbf{D}_m^{t_j}(a, b)$ for any simplex t_j :

$$\mathbf{D} = \text{diag}(\mathbf{D}_m^{t_j}(a, b), 1 \leq j \leq J) \in \mathbb{R}^{M \times J \cdot \hat{d}}, \quad (39)$$

with each $\mathbf{D}_m^{t_j}(a, b)$ a matrix function, defined using (36):

$$\mathbf{D}_m^{t_j}(a, b) := \frac{d!}{(d-m)!} \mathbf{B}^{d-m}(b) \mathbf{P}^{d,m}(a). \quad (40)$$

For every differential constraint, $\mathbf{D}_m^{t_j}(a, b)$ is evaluated for a given differential order m , directional coordinate a and evaluation location b such that

$$\mathbf{D}_m^{t_j}(a, b) \cdot \mathbf{c}^t = D_u^m p(b) \Big|_{t_j} = \mathbf{d}_i \quad (41)$$

with \mathbf{d}_i a single constraint value.

4.4. Differentially constraining polynomials on simplex edge facets

In this section we provide a proof that simplex polynomials can be differentially constrained along and across simplex edge facets

by formulating the constraints for a specific number of points located in the respective facet.

Theorem 4. *The directional derivative of a B -form polynomial $D_u^m p(b)$ along and across a simplex edge facet $\tilde{t}_j \in \mathbb{R}^{n-1}$ is fully constrained at a value \mathbf{d} by constraining a total of N_δ unique points in \tilde{t}_j with N_δ defined as follows:*

$$N_\delta := \frac{(d-m+n-1)!}{(n-1)!(d-m)!}. \quad (42)$$

Proof. From (36) it is clear that the directional derivative $D_u^m p(b)$ is a polynomial of degree $d-m$ in b . Now let $q(\tilde{b})$ be a polynomial of degree $d-m$ defined on \tilde{t}_j which is of dimension $n-1$.

$$q(\tilde{b}) = \sum_{|\kappa|=d-m} c_\kappa^{\tilde{t}_j} B_\kappa^{d-m}(\tilde{b}), \quad (43)$$

with \tilde{b} barycentric coordinates on \tilde{t}_j . Using (4), it is easy to check that $q(\tilde{b})$ consists of a linear combination of N_δ basis polynomials and B -coefficients. This system is fully determined when N_δ equations are present, which proves the theorem. \square

5. A recursive least squares estimator for simplex splines

In this section an equality constrained Recursive Least Squares (RLS) estimator for the B -coefficients of multivariate simplex splines is presented. This recursive estimator is based on a linear regression method for simplex splines presented in de Visser et al. (2009). The RLS estimator significantly reduces the computational efforts for real-time model modification with simplex splines, because no matrix inversions are required beyond the initial covariance matrix estimation.

5.1. RLS with linear equality constraints

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

$$J(\mathbf{c}) = (\mathbf{y} - \mathbf{X}\mathbf{c})^\top (\mathbf{y} - \mathbf{X}\mathbf{c}), \quad (44)$$

with $\mathbf{X} \in \mathbb{R}^{N \times J \cdot \hat{d}}$ the matrix of B -form regressors for N observations as derived in de Visser et al. (2009), and with \mathbf{y} the vector containing all N observations. The equality constrained least squares (ECLS) optimization problem then is:

$$\min_{\mathbf{c}} J(\mathbf{c}), \quad \text{subject to } \mathbf{G}\mathbf{c} = \mathbf{g}, \quad (45)$$

with \mathbf{G} the matrix with linear constraints and with \mathbf{g} a vector of constraint values:

$$\mathbf{G} = [\mathbf{H} \quad \mathbf{D}]^\top, \quad \mathbf{g} = [\mathbf{0} \quad \mathbf{d}]^\top. \quad (46)$$

In (46), \mathbf{H} is the matrix holding the smoothness constraints as derived in de Visser et al. (2009). In this case, \mathbf{D} and \mathbf{d} are the differential constraint matrix and the vector of differential constraint values from (38), respectively. In the literature the ECLS problem (45) has been solved with Lagrange multipliers, see Awanou et al. (2005) and de Visser et al. (2009). The differential equality constrained recursive least squares estimator (DECRLS) presented here is derived from a null-space solver for (45):

$$\hat{\mathbf{c}} = \mathbf{G}^+ \mathbf{g} + \mathbf{R}\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{G}^+ \mathbf{g}), \quad (47)$$

with \mathbf{R} the parameter covariance matrix estimate:

$$\mathbf{R} = (\mathbf{Z}\mathbf{X}^\top \mathbf{X}\mathbf{Z})^+, \quad (48)$$

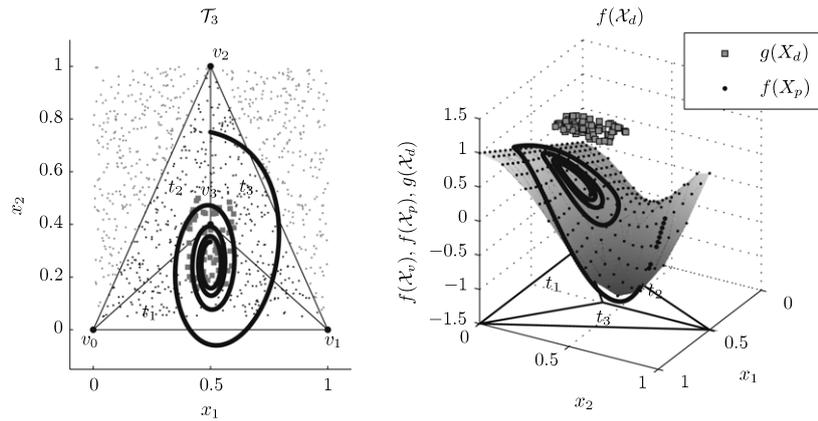


Fig. 2. Triangulation \mathcal{T}_3 and identification/validation/correction datasets (left) and validation function $f(\mathcal{X}_v)$ together with corruption function $g(\mathcal{X}_d)$ and correction function $f(\mathcal{X}_p)$ (right).

and with \mathbf{Z} an orthogonal projector onto the null-space of \mathbf{G} :

$$\mathbf{Z} = \mathbf{I} - \mathbf{G}^+ \mathbf{G}. \quad (49)$$

Zhu and Li (2007) proved that any equality constrained RLS estimator differs from an unconstrained RLS estimator only during initialization. The DECRLS estimator $\hat{\mathbf{c}}(t + 1)$ therefore is:

$$\begin{aligned} \mathbf{L}(t + 1) &= \mathbf{R}(t) \mathbf{x}^\top(t + 1) [\mathbf{I} + \mathbf{x}(t + 1) \mathbf{R}(t) \mathbf{x}^\top(t + 1)]^{-1}, \\ \mathbf{R}(t + 1) &= \mathbf{R}(t) - \mathbf{L}(t + 1) \mathbf{x}(t + 1) \mathbf{R}(t), \\ \hat{\mathbf{c}}(t + 1) &= \hat{\mathbf{c}}(t) + \mathbf{L}(t + 1) [\mathbf{y}(t + 1) - \mathbf{x}(t + 1) \hat{\mathbf{c}}(t)], \end{aligned} \quad (50)$$

with \mathbf{x} the regression model for a new observation and with \mathbf{R} the parameter covariance matrix estimate from (48). The initial values $\mathbf{R}(0)$ and $\hat{\mathbf{c}}(0)$ are found by using (48) and (47) respectively, with the initial regressor matrix $\mathbf{X}(0)$.

6. Demonstration of the equality constrained recursive least squares estimator

In this section, the equality constrained recursive least squares estimator for the B -coefficients of the simplex spline functions is demonstrated. This demonstration should be seen as the inner recursive identification loop within an adaptive model-based automatic control system, such as that shown in Fig. 1. Two types of equality constrained RLS estimators are demonstrated; an equality constrained RLS (ECRLS) estimator which only uses the smoothness conditions as equality constraints, and a differential equality constrained RLS (DECRLS) estimator which uses the smoothness conditions as well as a set of differential equality constraints in the form of Cauchy boundary conditions. It will be demonstrated that adding differential constraints in the solution system increases the quality of estimated spline models.

6.1. Demonstration setup

The ECRLS and DECRLS estimators are initialized with (47) using their respective constraint matrices \mathbf{G} . Three distinct sets of data are used in the experiment. The first of these datasets is \mathcal{X}_d which consists of 1000 scattered data points in \mathbb{R}^2 , generated with a uniform random number generator in the interval $\{[0, 1], [0.05, 1]\}$:

$$\mathcal{X}_d = \{x_1, x_2\} \in \{U(0, 1), U(0.05, 1)\}. \quad (51)$$

Note that the \mathcal{X}_d dataset does not fully cover the triangulation \mathcal{T}_3 thereby simulating an incomplete simplex data covering. The second dataset is \mathcal{X}_p which consists of 1257 data points and simulates a real-time data stream. It is used to correct the initial

measurement corruption in the RLS process. \mathcal{X}_p is a parametric curve in \mathbb{R}^2 defined as follows:

$$\begin{aligned} \mathcal{X}_p &= \left\{ \frac{1}{2} \left(1 + \frac{\sin(t)}{0.5t + 1} \right), \frac{1}{2} \left(\frac{1}{2} + \frac{\cos(t)}{0.2t + 1} \right) \right\}, \\ 0 &\leq t \leq 8\pi. \end{aligned} \quad (52)$$

The third dataset is the validation dataset \mathcal{X}_v which consists of 1301 data points in \mathbb{R}^2 laid out on a uniform rectangular grid on the interval $\{[0, 1], [0, 1]\}$:

$$\mathcal{X}_v = \{x_1, x_2\} \in \{[0, 1], [0, 1]\}. \quad (53)$$

The validation dataset is used to validate the spline models after the RLS process has been completed. The measurement values for the RLS process as well as the model validation are generated with the bivariate function $f(x_1, x_2)$:

$$f(x_1, x_2) = x_2^2 \sin(10x_1 + 10) + x_1 \cos(5x_2). \quad (54)$$

The (corrupted) measurement values for the initial identification process are generated with the bivariate function $g(x_1, x_2)$:

$$g(x_1, x_2) = \begin{cases} 1, & 0.4 \leq x_1 \leq 0.6, 0.15 \leq x_2 \leq 0.5 \\ f(x_1, x_2), & \text{otherwise,} \end{cases} \quad (55)$$

with $f(x_1, x_2)$ as in (54). Summarizing, three measurement sets are produced: the initial (corrupted) identification set $\{\mathcal{X}_d, g(\mathcal{X}_d)\}$, the correction set $\{\mathcal{X}_p, f(\mathcal{X}_p)\}$, and the validation set $\{\mathcal{X}_v, f(\mathcal{X}_v)\}$, see Fig. 2. Finally, the spline model residual is $\epsilon(\mathcal{X}_v) = s(\mathcal{X}_v) - f(\mathcal{X}_v)$.

6.2. Simplex spline model structure

The purpose of the numerical experiment is the demonstration of the properties and capabilities of the ECRLS and DECRLS estimators. As such, the model structure selection process for the bivariate spline function $s(x_1, x_2)$ was reduced to assuming that $s(x_1, x_2)$ is a fifth degree polynomial on a triangulation \mathcal{T}_3 with C^1 continuity between the spline pieces. The triangulation \mathcal{T}_3 was a non-rectangular triangulation consisting of 3 simplices.

Second order differential constraints were imposed in the form of Cauchy boundary conditions on the simplex edge facets:

$$\left. \frac{d^2 s(x_1, x_2)}{d\nu_j^2} \right|_{\tilde{t}_j} = 0, \quad 1 \leq j \leq 3, \quad (56)$$

with \tilde{t}_j the external edge facet of the simplex t_j and with ν_j the outward pointing normal vector of the edge facet. It was found in our experience that these particular constraints significantly

Table 1
Results of the demonstration experiment.

	ECRLS	DECRLS	Batch DECLS
$RMS(\epsilon)$	0.0681	0.0467	0.0467
$\max(\epsilon)$	0.4142	0.1348	0.1348
Mean $Var(\hat{c})$	0.3341	0.0440	0.0440
Time (s)	0.35	0.35	98.83
Relative time (-)	1	1	282.4

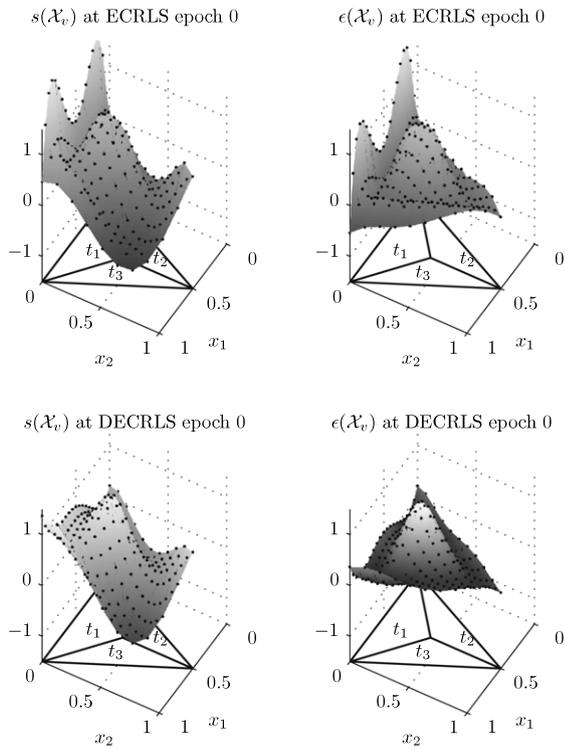


Fig. 3. Initial ECRLS (top) and DECRLS (bottom) spline models together with their model residuals.

reduce polynomial divergence while having only minor influence on the approximation power of the spline function. When modeling a real life system, however, differential constraints should be imposed based on expert knowledge of the derivatives of the system.

From (42) it follows that three sets of four equations of the form (41) are required to constrain the second order derivative of the spline function on the external edge facets of \mathcal{T}_3 :

$$D_{v_j}^2 s(b_i)|_{t_j} = \mathbf{D}_2^j(a_j, b_i) \cdot \mathbf{c}^{t_j} = 0, \quad 1 \leq j \leq 3, \quad 1 \leq i \leq N_\delta, \quad (57)$$

with a_j the directional coordinates of the normal vector v_j relative to simplex t_j . The points b_i in (57) are the barycentric coordinates relative to t_j of six unique points on \tilde{t}_j . The resulting global differential constraint matrix is then formed from the blocks $\mathbf{D}_2^j(a_j, b_i)$ according to (39) such that $\mathbf{D} \in \mathbb{R}^{12 \times 3 \cdot d}$.

6.3. Demonstration results: residual analysis

The numerical experiment was implemented in Matlab and run on a single core of an Intel Q6600 processor. In Table 1 the performance data are presented for the final epoch of the ECRLS, DECRLS, and the batch DECLS estimators. From the table it is clear that the recursive estimators are more than 280 times faster than the batch estimator. Table 1 also shows that adding differential constraints to the solution system significantly reduces the validation error RMS and the maximum error.

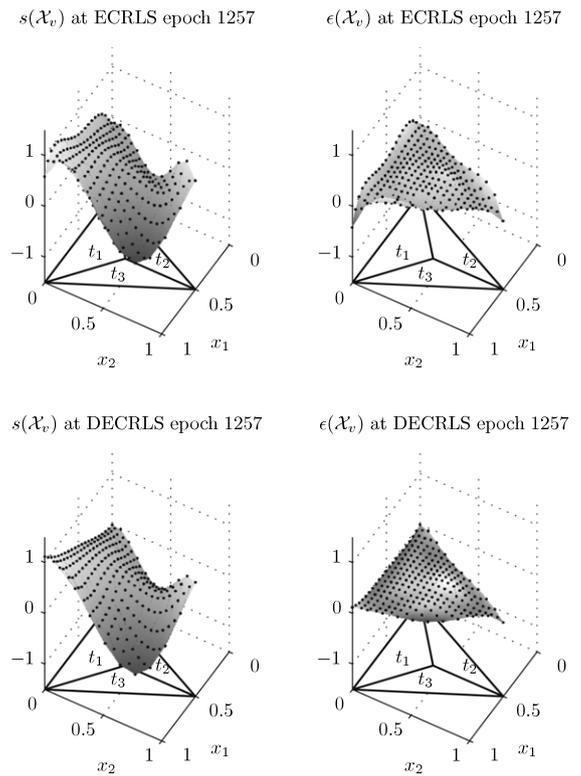


Fig. 4. Final ECRLS (top) and DECRLS (bottom) spline models together with their model residuals.

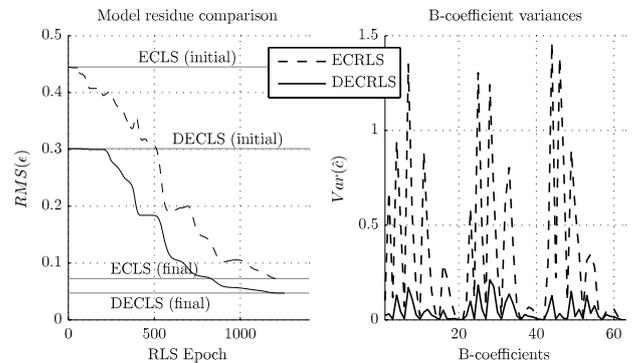


Fig. 5. Comparison between ECRLS and DECRLS residual RMS values during the recursion (left) and final B -coefficient variances (right).

In the top plot of Fig. 3 the initial spline models estimated with the ECLS and DECLS estimators are shown together with their respective model residuals. The spline function estimated with the ECLS estimator clearly diverges near the $x_2 = 0$ axis. This is caused by the fact that the initial dataset \mathcal{X}_d only partially covers simplex t_1 , see Fig. 2. The ECLS estimator therefore extrapolates the data near the boundary of t_1 . The ECRLS and DECRLS spline models and model residuals for the final RLS epoch are shown in Fig. 4. While the correction data has improved the ECRLS spline model, it still shows significant divergent behavior near the vertices v_0 , v_1 and v_2 . In contrast, the DECRLS spline model shows virtually no divergence near these vertices.

In the left hand plot of Fig. 5 the RMS of the model residual of the ECRLS and DECRLS models is shown as a function of the RLS epoch. It is clear from this figure that the RMS of the model residual of the DECRLS model is lower for every single RLS epoch.

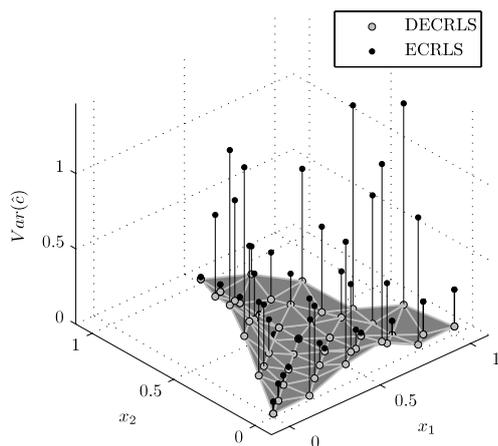


Fig. 6. Comparison between ECRLS and DECRLS B -coefficient variance surfaces.

6.4. Demonstration results: Statistical analysis

Next to an analysis of the model residual, an analysis of the B -coefficient variances was performed. The differential constraints significantly reduce the overall variance in the estimated B -coefficients, see Table 1 and the right hand plot of Fig. 5. In Fig. 6 the B -coefficient variance surfaces are plotted. This plot clearly shows that the differential constraints reduce B -coefficient variances, especially near the edges of the triangulation.

7. Conclusions

In this paper, a new differential equality constrained recursive least squares (DECRLS) estimator for multivariate simplex splines is presented. The estimator provides an online model identification and bounded extrapolation functionality for multivariate simplex splines. The new estimator is efficient computationally, providing a speedup of at least two orders of magnitude over a differentially constrained batch estimator on the same dataset. This makes the (DECRLS) for simplex splines highly desirable inside an online nonlinear system identification loop.

The differential equality constraints are a new type of constraint for the simplex splines and are based on a new one-step matrix formulation of the de Casteljau algorithm, which is presented in this paper. The differential constraints can be used to impose Dirichlet, Neumann, Robin or Cauchy boundary conditions on the simplex spline functions. The differential constraints are used to reduce divergent behavior of simplex polynomials near the boundaries of the triangulation by locally constraining the directional derivatives of the spline polynomials.

The RLS estimator for simplex splines was demonstrated with a numerical experiment implemented in Matlab. The differential constraints were used to impose second order Cauchy boundary conditions on the external edges of the spline domain. It was shown that the differential constraints significantly improved the quality of the simplex spline models, especially near these edges. Additionally, it was shown that the application of differential constraints on simplex edges greatly reduce B -coefficient variances close to these same edges.

References

- Awanou, G., Lai, M. J., & Wenston, P. (2005). The multivariate spline method for scattered data fitting and numerical solutions of partial differential equations. In G. Chen, & M. J. Lai (Eds.), *Wavelets and splines* (pp. 24–75).
- de Boor, C. (1987). B -form basics. In G. Farin (Ed.), *Geometric modeling: algorithms and new trends*. SIAM.
- de Visser, C. C., Chu, Q. P., & Mulder, J. A. (2009). A new approach to linear regression with multivariate splines. *Automatica*, 45(12), 2903–2909.

- de Visser, C.C., Mulder, J.A., & Chu, Q.P. (2010). A multidimensional spline based global nonlinear aerodynamic model for the Cessna Citation II. In *AIAA atmospheric flight mechanics conference*.
- Farin, G. (2002). A history of curves and surfaces in CAGD. In *Handbook of computer aided geometric design* (pp. 1–21). Elsevier Science B.V.
- Hu, X. L., Han, D. F., & Lai, M. J. (2007). Bivariate splines of various degrees for numerical solution of partial differential equations. *SIAM Journal on Scientific Computing*, 29, 1338–1354.
- Kalbe, T., Koch, T., & Goesele, M. (2009). High-quality rendering of varying isosurfaces with cubic trivariate C^1 -continuous splines. In *ISVC 2009 Vol. 5875*. (pp. 596–607).
- Lai, M. J., & Schumaker, L. L. (2007). *Spline functions on triangulations*. Cambridge University Press.
- Ljung, L. (2002). Recursive identification algorithms. *Circuits Systems Signal Processing*, 21, 57–68.
- Ljung, L., & Söderström, T. (1983). *Theory and practice of recursive identification*. MIT Press.
- Mahata, K., & Söderström, T. (2004). Improved estimation performance using known linear constraints. *Automatica*, 40, 1307–1318.
- Mirikitani, D. T., & Nikolaev, N. (2010). Recursive Bayesian recurrent neural networks for time-series modeling. *IEEE Transactions on Neural Networks*, 21, 262–274.
- Reiner, J., Balas, G. J., & Garrard, W. L. (1996). Flight control using robust dynamic inversion and time-scale separation. *Automatica*, 32, 1493–1504.
- Roll, J., Nazin, A., & Ljung, L. (2005). Nonlinear system identification via direct weight optimization. *Automatica*, 41, 475–490.
- Shewchuk, J. R. (2001). Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, 22, 21–74.
- Shewchuk, J. R. (2008). General-dimensional constrained Delaunay and constrained regular triangulations i: combinatorial properties. *Discrete and Computational Geometry*, 39(1), 580–637.
- Söderström, T., Ljung, L., & Gustavsson, I. (1978). A theoretical analysis of recursive identification methods. *Automatica*, 14, 231–244.
- Zhu, Y., & Li, X. R. (2007). Recursive least squares with linear constraints. *Communications in Information and Systems*, 7, 287–312.



C. C. de Visser received his B.Sc. and M.Sc. degrees from the Delft University of Technology in 2006 and 2007. From 2007 he has been a Ph.D. candidate at the Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands. His research interests are nonlinear system identification, multivariate spline theory, flight dynamics, aerodynamics and fault tolerant control.



Q. P. Chu received his Ph.D. degree from the Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands, in 1987. Currently, he is an Associate Professor at the Faculty of Aerospace Engineering, Delft University of Technology, responsible for education and research activities in aerospace guidance, navigation and control. He has (co)authored more than 150 journal and conference papers ranged from adaptive control, nonlinear control, robust control and intelligent control to nonlinear state estimation, system identification and nonlinear optimization for aerospace vehicles. Dr. Chu was the designer of the attitude control system for the third Dutch satellite 'Sloshsat' launched in Feb. 2005 and is a member of the American Institute of Aeronautics and Astronautics (AIAA).



J. A. Mulder graduated as 'engineer-pilot' at the faculty of Aerospace Engineering of the Delft University of Technology, The Netherlands, in 1968. He received his Ph.D. degree on aerodynamic model identification in 1986 and was next appointed Full Professor in 1989 as chair of the Control and Simulation Division in the Faculty of Aerospace Engineering. He is founding Scientific Director of the Institute for research in Simulation, Motion and Navigation Technologies (SIMONA) of Delft University of Technology. He (co)authored more than 200 conference and journal papers on subjects ranging from human-machine interface design and cybernetics to identification, estimation and advanced flight control. Prof. Mulder has served in numerous ministerial advisory committees, as adviser to the board of the National Aerospace Laboratory (NLR) and the Radiobiological Institute of the Netherlands Organization for Applied Scientific Research (TNO). He has been a member of the AGARD Flight Mechanics panel from 1982 to 1997. Presently he is member of the Aerospace Control & Guidance Systems Committee of the Society of Automotive Engineers (SAE), of the Society of Flight Test Engineers (SFTE) and senior member of the American Institute of Aeronautics and Astronautics (AIAA), of its Avionics Committee and Associate Editor of the *AIAA Journal of Aerospace Computing, Information, and Communication* and of the *Journal of Micro Aerial Vehicles*.