Distributed approach for aerodynamic model identification of the ice aircraft using the alternating direction method of multipliers in combination with simplotope b-splines

van den Aarssen, Marc; Visser, Tim; de Visser, Coen

DOI
10.2514/6.2019-1320

Publication date
2019

Document Version
Final published version

Published in
AIAA Scitech 2019 Forum

AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-1320

Distributed approach for Aerodynamic Model Identification of the ICE aircraft using the Alternating Direction Method of Multipliers in combination with Simplotope B-Splines

M.S.T. van den Aarssen*, T. Visser†, and C. C. de Visser‡
Delft University of Technology, Delft, 2600GB, The Netherlands

High performance control allocation methods for the Innovative Control Effectors (ICE) aircraft require accurate onboard aerodynamic models, with preferably first order continuity. Simplotope B-Splines, an extension on Simplex B-Splines, have a high approximation power by using local cost functions. However, enforcing global continuity produces computationally expensive optimization problems. This paper presents a distributed approach, using the Alternating Direction Method of Multipliers (ADMM), to reduce the complexity of the B-Coefficients' estimation. ADMM decouples the simplotopes, and introduces coupling coefficients to enforce global continuity, resulting in a parallel estimation algorithm whose complexity is depending solely on the partition size, being independent of refinement of the model tessellation. Results show that for a 3D model, the distributed algorithm converges steadily to the global solution with a good approximation after a couple hundred iterations. Validation results of the distributed approach were similar to those of the global optimal solution for various noise intensities, and the continuity constraints were satisfied with maximum mismatches below 10^{-4}. The distributed approach has been used to construct a first order continuous aerodynamic model for the ICE aircraft, which has been implemented in Simulink, and proven to perform well compared to the original model.

Nomenclature

$\boldsymbol{b}$	=	Simplex barycentric coordinates
B	=	Simplex basis polynomials
C	=	Aerodynamic coefficient

*MSc Student, Control and Simulation Group, Faculty of Aerospace Engineering, Kluyverweg 1, 2629HS Delft, The Netherlands, marcv-daarssen@gmail.com
†PhD candidate, Astrodynamics and Space Missions Group, Faculty of Aerospace Engineering, Kluyverweg 1, 2629HS Delft, The Netherlands, t.visser-1@tudelft.nl
‡Assistant Professor, Control and Simulation Group, Faculty of Aerospace Engineering, Kluyverweg 1, 2629HS Delft, The Netherlands, c.c.devisser@tudelft.nl. AIAA Member

$\boldsymbol{c}$	=	B-Coefficients of a Simplex or Simplotope B-Polynomial
H	=	Continuity Matrix
$\mathcal{H}$	=	Indices of constraints
J	=	Number of Simplotopes in tessellation
$\mathcal{J}$	=	Cost Function
$\mathcal{L}$	=	(Augmented) Lagrangian
M	=	Mach number
n	=	Number of Dimensions
$\mathcal{N}$	=	Null space
P	=	Number of Partitions
p, q, r	=	Roll, pitch, and yaw rate in body x, y, z directions
r	=	Continuity order
R	=	Number of Continuity Constraints
$\boldsymbol{r}, \boldsymbol{s}$	=	Primal- and Dual residual
t	=	Simplex
X	=	Regression Matrix
Y	=	Measurements
z	=	Coupling coefficient
α, β	=	Angle of Attack and Angle of Sideslip
$\boldsymbol{\beta}$	=	Simplotope barycentric coordinates
Γ	=	Simplotope
δ	=	Control effector deflection
ϵ	=	Tolerance
κ	=	Simplex multi-index for the B-Coefficients
λ	=	Dual vector
$\boldsymbol{\lambda}$	=	Simplotope multi-index for the B-Coefficients
π	=	Simplotope basis polynomial

ρ = Penalty factor

Subscripts

l, m, n = Aerodynamic moment in body x, y, z directions

p = Partitions index

X, Y, Z = Aerodynamic Force in body x, y, z directions

I. Introduction

Aerodynamic Model Identification (AMI) plays an important role in safe- and high-performance control of aircraft, as various control techniques use an onboard aerodynamic model [1–3]. Inversion methods require e.g. an estimation of the aircraft's aerodynamic coefficients or control Jacobian. Especially a high performance fighter, such as the Innovative Control Effectors (ICE) aircraft, depends upon proper control allocation, as without that, the redundant set of control effectors makes it nearly impossible to be flown by a human pilot.

Numerous techniques for AMI exist. Ordinary polynomials are most common because of their relative simplicity. These are generally used in small models and work well in those cases, but become inaccurate when the aerodynamic model is large or highly non-linear, as is the case with the ICE aircraft [4]. Artificial Neural Networks (ANN) are used in many machine learning cases, because of their high approximation power. However, its nature is considered black box as the internal workings are hard to identify. In order to estimate all the parameters, neural networks require a high amount of training data, while optimizing global cost functions. In addition, ANNs are generally tempting to overfit the identification data [5]. Tensor Product splines combine multiple univariate functions to obtain higher dimensional models, but are not able to fit scattered data [6, 7]. As scattered measurements are most common in the context of AMI, this method is not well suited in general.

Multivariate Simplex B-Splines combine the approximation power of artificial neural networks and the simplicity of the polynomials [8, 9]. Because it divides the full domain in smaller patches with local cost functions, it has a high approximation power. By using the barycentric formulation of the polynomials, the basis is stable and shows easy formulations to enforce any order of continuity between the subdomains [10]. This model structure has been successfully applied to AMI, but also to other fields, such as adaptive optics or partial differential equations [8, 11, 12].

Instead of simplices, other geometric basis have been proposed for splines in the past. Most relevant for this work is the simplotope, defined as the product of simplices. Although this polytope has been mostly used in the field of triangulations (e.g. [13]) and game theory [14, 15], it has also made its appearance in the context of vertex splines [16**?**]. These studies have however focused on the two- and three-dimensional case only, or were limited to the simplex and

the hypercube. Applications of the simplotope spline have been limited to a single study into quadrotor modeling (in which the basis is called a multiplex) [17].

A beneficial aspect of the B-Spline structure is that the regression problem is linear-in-the-parameters, which means that the B-Coefficients can be estimated using linear methods as Equality Constrained Ordinary Least Squares (ECOLS). This well-known method has multiple solvers, such as directly inverting the Karush-Kuhn-Tucker matrix, null space projection, or a highly efficient iterative solver that has been developed especially considering the B-Spline structure [9, 18, 19]. However, all these methods still solve a global problem, not fully exploiting the local properties of the simplotope splines. Distributed methods have been used as well with B-Splines, using Dual Ascent or the Alternating Direction Method of Multipliers (ADMM) [12, 20]. However, this has only been used for the 2-dimensional wave front reconstruction problem and has not been applied to the higher dimensional AMI case yet.

The current aerodynamic model of the ICE aircraft consists of a multivariate Simplex B-Spline model, linked together with zeroth order continuity [21]. Although it was accurately identifying the data, it showed some wiggles between the datapoints and discontinuities in the control derivatives. Moreover, the coefficients were estimated using the iterative solver described in [19], which was computationally extremely expensive, because some models required over 200,000 coefficients. This was not only inconvenient, but also limiting model's degree and thus approximation power.

In order to reduce the computational cost of the estimation, distributed approaches for AMI in combination with multivariate Simplotope B-Splines have been investigated. Especially a desire to have infinitely scalable methods encouraged research towards fully parallel optimization schemes able to maintain global continuity between the subdomains. This paper presents such a distributed approach, using ADMM, splitting the B-Coefficients per simplotope, and introducing coupling coefficients to ensure continuity. This method's complexity is dependent on the size of the individual units, but is indifferent to refinement of the tessellation because of its parallel nature. The method has been evaluated for several noise intensity levels and compared to its global solution in terms of convergence, validation errors, and global constraint satisfaction.

The distributed approach will be used to identify a model for the ICE-aircraft, which will be compared to the Simulink model that has been developed for research purposes [22].

This paper is structured as follows. First a brief description of the history and the aerodynamic model of the ICE aircraft will be given in Section II, together with the challenges faced with the current aerodynamic model. Next the concept of multivariate B-Splines with Simplices and Simplotopes is introduced in Section III. How this model structure can be used to identify the aerodynamic model, and the introduction of ADMM is presented in Section IV. Results are given in Section V and the findings are concluded in Section VI.

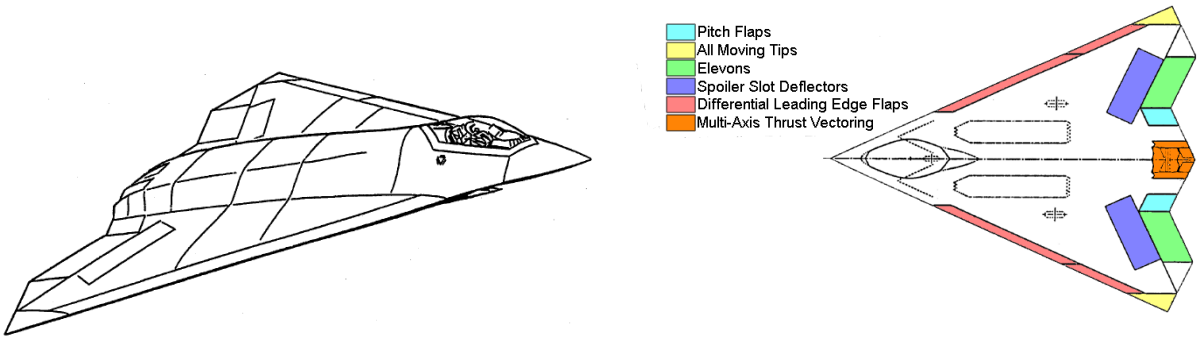
II. Description of the Innovative Control Effectors aircraft

The distributed approach that will be presented in Section IV, will be applied to the aerodynamic model of Lockheed Martin's Innovative Control Effectors (ICE) aircraft. The ICE-aircraft has been designed in the mid 1990s as a high performance fighter jet and includes a redundant set of unconventional control effectors. These control effectors ensure controllability and stability in all regions of the flight envelope. A general description will be given in Section II.A. The aerodynamic model is described in Section II.B.

A. History and Description

The ICE project has been initiated by the United States' Government's desire to have a high performance fighter in all parts of the flight envelope, while reducing the aircraft's radar cross section. This restricted the fighter's design, resulting in a tailless aircraft and integrated controls as shown in Fig. 1a. Stability and controllability issues initiated the research into innovative control effectors. This has been done for two baseline configurations of the fighter. One is land based, which has a 65 degree sweep delta wing; the other is a carrier based version, and has a 45 degree sweep angle, and a canard configuration.

During two phases of researching various novel control concepts, a selection of control effectors had been made. The land based version that will be studied in this paper consists of a total of 13 control effectors [23, 24].



(a) The land based version. **(b) Top view planform**

Fig. 1 The configuration of the ICE aircraft studied in this paper [23, 24]

Except for the more conventional Elevons (2), Pitch Flaps, and Multi-Axis Thrust Vectoring, some novel control surfaces were introduced. The All Moving Tips (AMT) are located at the wing tips and are able to fully rotate down. These were considered overall, the best performing control effectors for directional control in all regions of the envelope.

The Spoiler Slot Deflectors (SSD), located on the wings, combine a conventional spoiler with a deflector on the lower side of the wing, and a hole that enables air flowing through. Although providing good directional control, they were less effective than the AMTs. Moreover, deflecting the SSDs highly affected the trailing edge effectors.

The Differential Leading Edge Flaps (DLEF) are able to provide lateral control. There are inboard- and outboard flaps placed on each side of the wing. This control surface was deemed most useful in power approaches on the one hand, and low speed, high Angle of Attack conditions on the other hand. Moreove, they retained its control effectiveness at high Mach numbers.

A full overview of all the control effectors, their conventions, and the maximum deflections and rates, are shown in Table 2. The convention can either be Trailing Edge (TE) or Leading Edge (LE) Up (U) or Down (D). The exact positions are colored in the planform in Fig. 1b.

Table 2 An overview of all the control effectors of the ICE aircraft, and their conventions, limits, and notations [22].

Control Surface	Convention Positive	Limits	Rate limit	Notation
Spoiler Slot Deflectors (2)	TEU	[0,60]	150	δ_{LSSD}, δ_{RSSD}
Differential Leading Edge Flaps (4)	LED	[0,40]	40	δ_{LIBLEF}, δ_{RIBLEF} (Inboard)
		[-40,40]	40	δ_{LOBLEF}, δ_{ROBLEF} (Outboard)
All Moving Tips (2)	TED	[0,60]	150	δ_{LAMT}, δ_{RAMT}
Elevons (2)	TED	[-30,30]	150	δ_{LEL}, δ_{REL}
Pitch Flaps (1)	TED	[-30,30]	150	δ_{PF}
Multi-Axis Thrust Vectoring	$\dot{p} > 0, \dot{q} > 0$	[-15,15]	150	δ_{TV}

The redundant set of effectors gives both opportunities and challenges, considering control and stability. With particular effectors more effective in different areas of the flight envelope, it rises the opportunity to fly at a large range of aerodynamic angles; e.g. the Angle of Attack is able to grow up to 90 degrees.

Since it is over-effected, it requires control allocation methods in order to have it properly steered by a human pilot. Only recently an effective Incremental Non-Linear Control Allocation method has been designed and proven to work well in the simulation environment [3]. This method requires an onboard aerodynamic model for the estimation of the control effectiveness Jacobian. This calls for an accurate aerodynamic model whose derivatives can be evaluated during flight time.

B. Aerodynamic Model Description

The aerodynamic model of the ICE aircraft that is currently being used includes non-linear effects of the control effectors, individually as well as coupling effects, dynamic derivatives, aeroelastic effects, and hinge moment derivatives [22].

The data has been collected using subsonic-, transonic-, and supersonic wind tunnel tests on a small scale model of the aircraft and numerical methods, e.g. the DYNAMIC panel method to estimate dynamic derivatives at high mach numbers.

The full aerodynamic model is composed of a total of 108 submodels. Each of the three force coefficients (C_X, C_Y,

C_Z) and moment coefficients (C_l, C_m, C_n) consist of 17-19 submodels. Each submodel represents a contribution of either the bare airframe, an individual control effector, a combination of control effectors, or a dynamic derivative. The main coefficients are being estimated by combining the individual submodels. An example of how a main coefficient is built up is shown in Eq. (1) for the yawing moment coefficient C_n. All the main coefficients have the same subcoefficients in terms of dependency on states, only the dynamic derivatives $C_{n_{18}}$ and $C_{n_{19}}$ are not present for all coefficients. There are submodels that are added to the total, whereas some are subtracted. This combination is different for each main coefficient. The influence of the multi-axis thrust vectoring has not been included in the submodels, but combined with the propulsion model instead.

$$
\begin{aligned}
C_n \;= & \; C_{n_1}(\alpha, M) + C_{n_2}(\alpha, \beta, M) - C_{n_3}(\alpha, \beta, \delta_{LIBLEF}) - C_{n_4}(\alpha, \beta, \delta_{LIBLEF}, \delta_{LOBLEF}, M) \\
& + C_{n_5}(\alpha, \delta_{LSSD}, \delta_{LEL}, M) + C_{n_6}(\alpha, \delta_{LSSD}, \delta_{RSSD}, \delta_{PF}, M) + C_{n_7}(\alpha, \beta, \delta_{LAMT}) \\
& + C_{n_8}(\alpha, \delta_{LEL}, \delta_{LAMT}) + C_{n_9}(\alpha, \delta_{LOBLEF}, \delta_{LAMT}) - C_{n_{10}}(\alpha, \delta_{REL}, \delta_{RAMT}) \\
& - C_{n_{11}}(\alpha, \delta_{ROBLEF}, \delta_{RAMT}) + C_{n_{12}}(\alpha, \beta, \delta_{LSSD}) + C_{n_{13}}(\alpha, \beta, \delta_{RIBLEF}) \\
& + C_{n_{14}}(\alpha, \beta, \delta_{RIBLEF}, \delta_{ROBLEF}, M) - C_{n_{15}}(\alpha, \delta_{RSSD}, \delta_{REL}, M) - C_{n_{16}}(\alpha, \beta, \delta_{RAMT}) \\
& - C_{n_{17}}(\alpha, \beta, \delta_{RSSD}) + \frac{bp}{2V} C_{n_{18}}(\alpha, M) + \frac{br}{2V} C_{n_{19}}(\alpha, M)
\end{aligned}
\tag{1}
$$

The data has been combined into a Simulink model that interpolates, linearly or by cubic spline, the measurement data in order to find each submodel's contribution. The actuator dynamics are implemented separately as linear actuation models. There is a separation between the leading edge devices, which have a low bandwidth actuator, and all others, which have a high bandwidth [3, 22]. These actuator implementations have been used for the results in Section V.

The currently used aerodynamic model, constructed by [21], and uses multivariate Simplex B-Splines with zeroth order continuity between the simplices. For all submodels, the simplex splines resulted in relative validation RMSs below 1.6%, and relative maximum validation residuals below 21.6%. Because it only contained zeroth order continuity, discontinuities occurred in the first derivative, causing visible ripples in the model responses. These discontinuities also harmed the control allocation method from [3].

Another challenge was evaluating the model in real time. Since the model output is determined by the simplex in which the state falls, this simplex needs to be determined at every time step. This is a costly search, but was implemented with high performance in C++, allowing sufficient possible evaluations per second [21].

III. Preliminaries on Multivariate B-Splines

This section will provide preliminary information of the used model structure: Simplotope B-Splines. First, the theory of Simplex B-Splines is discussed in Section III.A, and how the local patches can be linked together using continuity conditions (Section III.B). Then the theory is extended into Simplotope B-Splines in Section III.C, which

divides the dimensions into multiple layers, containing lower dimensional simplices.

A. Spline Functions and Barycentric Coordinates

Bernstein basis polynomials are formed on simplices, where an n-dimensional simplex is defined as the convex hull of $n+1$ different vertices $\boldsymbol{v}$ [8]. A simplex spline can be created on a Triangulation $\mathcal{T}$, which is the collection of J non-overlapping simplices that together form the domain of the model Ω [25]:

$$\mathcal{T} = \{t_1, ..., t_J\}, \quad \Omega = \bigcup_{j=1}^{J} t_j, \quad t_i \cap t_j \in \{\emptyset, \tilde{t}\}, \quad \forall\, t_i, t_j \in \mathcal{T} \tag{2}$$

The edge facets consist of lower dimensional simplices $\tilde{t}$. Each edge between two neighboring simplices t_i and t_j can be specified by the out of edge vertices v_* of each simplex. This vertices are the ones of the simplices that are not part of the edge facet. The barycentric coordinates $\boldsymbol{b}$ of a data point are defined locally in the simplex. In this coordinate system all points x get a set of weights $\boldsymbol{b}$ which are multiplied with the vertices $\boldsymbol{v}$ of the particular simplex, as can be seen in Eq. (3) [9].

$$x = \sum_{i=0}^{n} b_i v_i \qquad\qquad \sum_{i=0}^{n} b_i = 1 \tag{3}$$

Every $n-$dimensional data point has $n+1$ barycentric coordinates, corresponding to the amount of vertices. When the chosen point is within the convex hull of the vertices $\boldsymbol{v}$, all barycentric coordinates will be non-negative. Any point outside this hull can be transformed into barycentric coordinates as well, but will have at least one negative coordinate.

The polynomial basis function in Eq. (4) can be used to write any given polynomial $p(\boldsymbol{b})$ of degree d as the weighted sum of these basis functions. This is shown in Eq. (5). By defining the basis function of any point outside the convex hull of the simplex 0, the basis becomes local. The property that all barycentric coordinates add up to one, makes that the basis is also stable.

$$B_\kappa^d(\boldsymbol{b}) = \frac{d!}{\kappa!}\boldsymbol{b}^\kappa \tag{4}$$

$$p(\boldsymbol{b}) = \sum_{|\kappa|=d} c_\kappa B_\kappa^d(\boldsymbol{b}) \tag{5}$$

The weights c_κ in Eq. (5) are the B-Coefficients for this particular simplex and define the shape of the polynomial. Each coefficient corresponds to a single permutation of the multi-index $\kappa = (\kappa_0, \kappa_1, \ldots, \kappa_n)$, provided that $0 \leq \kappa_i \leq d$ and $|\kappa| = d$. B-coefficients have a spatial location in the simplex, defined by the barycentric coordinate κ/d. The collective of B-coefficients at their locations is called the B-net. The total number of B-Coefficients that is used within a single simplex, $\hat{d}$, can be computed by Eq. (6).

$$\hat{d} = \binom{d+n}{d} = \frac{(d+n)!}{d!n!} \tag{6}$$

B. Continuity

As the Bernstein polynomials are defined locally on the simplices, continuity constraints have to be introduced to ensure that the model has a single surface that is smooth up to a chosen order. For orders of continuity r between two simplices t_i and t_j, the continuity equations can be formed by using Eq. (7) [10].

$$c^{t_i}_{\kappa_0,m,\kappa_1,\ldots,\kappa_{n-1}} = \sum_{|\gamma|=m} c^{t_j}_{(\kappa_0,0,\kappa_1,\ldots,\kappa_{n-1})+\gamma} B^m_\gamma(v^{t_i}_*), \qquad 0 \leq m \leq r \tag{7}$$

For every permutation of the continuity multi-index $\gamma = [\gamma_0, \gamma_1, , \ldots, \gamma_n]$, $0 \leq \gamma_i \leq m, |\gamma| = m$, a continuity equation is formed based on the out of edge vertex $v^{t_i}_*$ of simplex t_i. The barycentric coordinates have to be computed w.r.t. the vertices of the simplex t_j. Additionally, the condition $|\kappa| + |\gamma| = d$ has to hold for every constraint. With the chosen continuity order r, every edge has R continuity conditions, which can be computed with Eq. (8)

$$R = \sum_{m=0}^{r} \frac{(d-m+n-1)!}{(d-m)!(n-1)!} \tag{8}$$

Because all these constraints are linear, they can be combined for all simplices into a single, global smoothness matrix $\mathbf{H}$. This makes that $\mathbf{H} \cdot \boldsymbol{c} = 0$ can be used as the linear equality constraints of the optimization problem.

C. Simplotope Splines

As the dimension of a simplex increases, its corners become sharper and its content becomes more distributed (see [17, Fig. 5]). This implies that it is increasingly difficult to properly distribute data over the content of a high-dimensional simplex. To mitigate this problem we can choose to span the full-dimensional domain by taking the tensor-product of lower-dimensional simplex splines. The result is a B-spline defined on the product of simplices, a so called simplotope (or: multiplex in [17]). In this section we discuss the implications of using this basis for the definition of the geometric basis and tesselation, barycentric coordinates, basis polynomials, and continuity conditions.

A simplotope Γ is defined as the product of ℓ lower-dimensional, mutually perpendicular simplices Δ [14], here referred to as layers. The type of a simplotope is defined by the dimension of the layers, collected in a multi-index $\nu = (\nu_1, ..., \nu_\ell)$. Because the layers are mutually perpendicular the total dimension of the simplotope n is equal to the sum of the dimensions of the layers. In Fig. 2 all types of simplotopes are shown for dimensions 1 up to 3. Both the n-simplex and the n-cube are special cases of the simplotope. The former is formed by a single n-dimensional layer, the latter by n one-dimensional layers.

Instead of tessellating a domain using simplotopes directly, we can subdivide the domain in layers first and then triangulate the layers. The simplotopes are then formed by taking each possible product of a simplex from each layer.

A point within the simplotope can be uniquely described using the barycentric coordinates $\boldsymbol{b}_i$ for the projection of that point onto every layer i, collected in a vector $\boldsymbol{\beta} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell)$. Where a point in a simplex has $n + 1$ barycentric

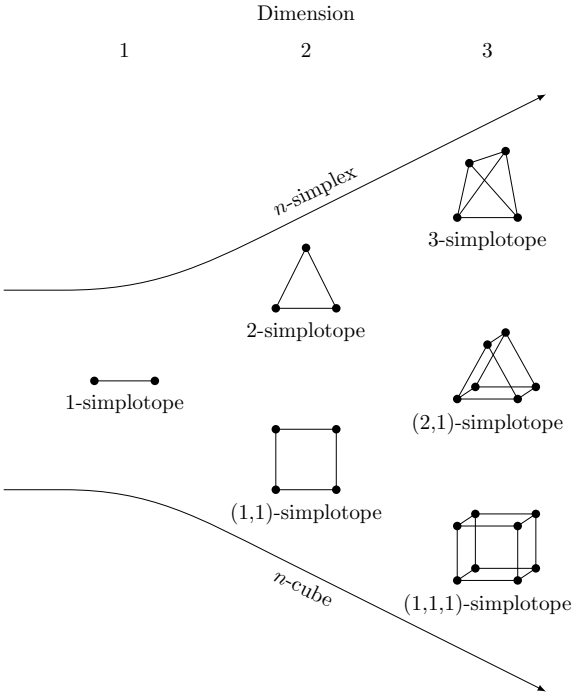


Fig. 2 All simplotopes for dimensions 1 up to 3. Note that the n-simplex and the n-cube are special cases of the simplotope.

coordinates that sum to 1, a point within a simplotope thus has $n + \ell$ coordinates that sum to ℓ.

The polynomial degree can be chosen freely in each layer, to form a total degree vector $\boldsymbol{d} = (d_1, \ldots, d_\ell)$. The basis polynomials defined on the simplotope are constructed by taking the tensor-product of the sets of basis polynomials in the layers. For every combination $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell)$ of multi-indices from each layer, we find a basis polynomial

$$\boldsymbol{B}_{\boldsymbol{\lambda}}(\boldsymbol{\beta}) = \prod_{i=1}^{\ell} B_{\boldsymbol{\lambda}_i}^{d_i}(\boldsymbol{b}_i)\,. \tag{9}$$

The complete polynomial is formed by taking the linear combination of these basis polynomials, as

$$\pi(\boldsymbol{\beta}) = \sum_{|\boldsymbol{\lambda}_i|=d_i, \forall i \in [1,\ell]} \boldsymbol{c}_{\boldsymbol{\lambda}} \boldsymbol{B}_{\boldsymbol{\lambda}}(\boldsymbol{\beta}), \tag{10}$$

with the B-coefficients $\boldsymbol{c}_{\boldsymbol{\lambda}}$ as parameters. The total number of B-coefficients is equal to the product of the number of B-coefficients in each layer.

The definition of the multi-index $\boldsymbol{\lambda}$ reveals the structure of the B-net of simplotope splines. The B-net in the first layer, defined by the full set of permutations of $\boldsymbol{\lambda}_1$, is repeated for every valid permutation of the remaining multi-index $(\boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell)$.

Two simplotopes share an edge if they share all but one layer completely, and a $(\nu_i - 1)$-dimensional edge in the i^{th} layer. In this case the i^{th} layer is called the out-of-edge layer. Between two simplotopes that share an edge, continuity

can be achieved by copying the continuity conditions defined in the out-of-edge layer for each copy of the B-net in that layer [26]. The result is a set of $\prod_{j\neq i} \hat{d}_j$ linear relations that can again be collected in a smoothness matrix $\mathbf{H}$.

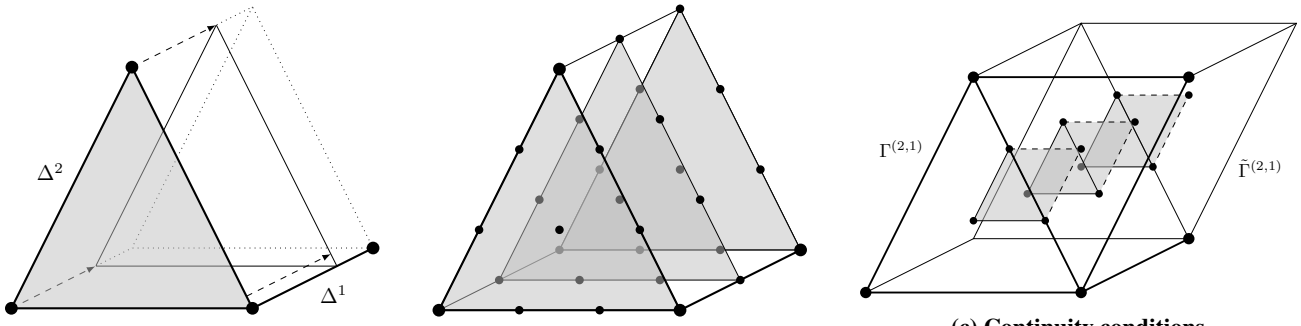
Example 1. Consider a three-dimensional domain, divided into a two-dimensional and a one-dimensional layer. Taking the product between a simplex from each layer, we find a (2,1)-simplotope $\Gamma^{(2,1)} = \Delta^2 \times \Delta^1$ as the extrusion of Δ^2 along Δ^1 (see Fig. 3a). The result is a triangular prism. We can freely choose the polynomial degree in each layer. Choosing $\boldsymbol{d} = (3, 2)$ we find multi-index permutations

$$\begin{aligned}\boldsymbol{\lambda}_1 =&(3,0,0),(2,1,0),(2,0,1),(1,2,0),(1,1,1),(1,0,2),(0,3,0),(0,2,1),(0,1,2),(0,0,3)\\ \boldsymbol{\lambda}_2 =&(2,0),(1,1),(0,2)\end{aligned}$$

in the first and second layer respectively. Every combination of a permutation of $\boldsymbol{\lambda}_1$ and a permutation of $\boldsymbol{\lambda}_2$ is a valid permutation of the simplotopic multi-index $\boldsymbol{\lambda}$. Taking all possible combinations of the multi-indices $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ amounts to copying the B-net in the first layer for each B-coefficient in the second layer or, equivalently, the other way around (see Fig. 3b). The basis polynomials are defined as the product of basis polynomials form the two layers. As an example consider the basis polynomial defined by the multi-index $\boldsymbol{\lambda} = ((1, 2, 0), (0, 2))$

$$\begin{aligned}\boldsymbol{B}^{(3,2)}_{((1,2,0),(0,2))}(\boldsymbol{\beta}) =& B^3_{(1,2,0)}(\boldsymbol{b}_1)\, B^2_{(0,2)}(\boldsymbol{b}_2)\\ =& \frac{3!}{1!2!0!} b^1_{10} b^2_{11} b^0_{12} \cdot \frac{2!}{0!2!} b^0_{20} b^2_{21} = 3 b_{10} b^2_{11} b^2_{21}\end{aligned}$$

Continuity conditions defined in the first layer can be copied for every copy of that B-net at each coefficient in the second layer. This is illustrated for a single, first order continuity condition in the first (two-dimensional) layer in Fig. 3c.



(a) (2,1)-Simplotope **(b) B-net** **(c) Continuity conditions**

Fig. 3 Definition of the (2,1)-simplotope as an extrusion of simplex Δ^2 along simplex Δ^1, the construction of its B-net for degrees (3,2), and the replication of a first order continuity condition between $\Gamma^{(2,1)}$ and $\tilde{\Gamma}^{(2,1)}$ in the two-dimensional layer into the full-dimensional simplotopes. (Adapted from [17].)

Advantages of using simplotopes over simplices are the flexibility it offers to decouple dimensions, such that different polynomial degrees can be used to describe different variables. This is beneficial in case highly unequal amounts of data are available in particular dimensions, or when an analytical model indicates coupling between some variables. Moreover, simplotopes are generally better shaped, easier to be filled with data, and often require less coefficients and less continuity constraints [17]. The tensor product between the layers will however result in high cross-coupling terms up to degree $|\boldsymbol{d}| = d_1 + \cdots + d_\ell$, introducing the risk of Runge's phenomenon near the corners of the domain. Although the total amount of coefficients is reduced, the simplotope structure will express every data point in a higher number of polynomial terms and consequently more coefficients. This increases the memory requirements of e.g. the regression matrix.

Example 2. In order to demonstrate the differences between the model structures, a small model was identified for the second subcoefficient of the yaw moment of the ICE aircraft $C_{n_2}(\alpha, \beta, M)$ using both Simplex and $(1, 1, 1)-$Simplotope structures. Both models are constructed with first order continuity and use the same hypercube grid on which Kuhn's triangulation is applied.

Table 3 shows the differences between the two model structures, and indicate that because the $(1, 1, 1)-$Simplotope divides the domain into a smaller tessellation, it globally uses fewer B-Coefficients and continuity conditions. Moreover, this smaller model results in a better fit to the identification data, as the coefficient of determination is higher for the simplotope model.

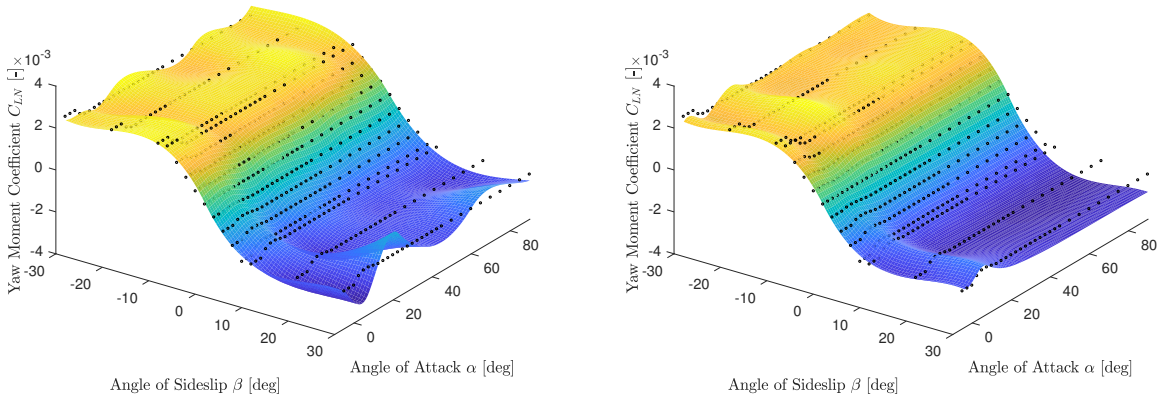
Table 3 Comparison of the Simplex structure with the Simplotope Structure.

Metric	*Simplex*	*Simplotope*
Degree	(3)	(3,3,3)
Number of Subdomains	96	16
Number of B-Coefficients	1920	1024
Number of Continuity Conditions	1788	664
Coefficient of Determination	0.75	0.89

In Fig. 4 a slice of the model at Mach = 2.16 is given for both model structures. The better fitness as indicated numerically in Table 3 is visible in these two slices, as Fig. 4a clearly shows wiggles and inaccuracies, whereas Fig. 4b does not show these. The simplotope model is better able to follow the trend of the data and does not introduce unrealistic sign changes in the first derivative.

IV. Aerodynamic Model Identification

Because the Bernstein polynomials are linear-in-the-parameters, the estimation can be done by using linear methods such as Ordinary Least Squares (OLS) [9]. The OLS cost function is the sum of the squared residuals, and because of



(a) Simplex **(b) Simplotope**

Fig. 4 B-Spline model of subcoefficient $C_{n_2}(\alpha, \beta, M)$ at $M = 2.16$

the locality of the polynomials, this results in a summation of the simplotopes' individual cost functions as described in Eq. (11).

$$\mathcal{J}(\boldsymbol{c}) = \sum_{i=1}^{J} ||\mathbf{X}_i \boldsymbol{c}_i - \boldsymbol{Y}_i||_2 \tag{11}$$

Using this cost function, in combination with the continuity constraints as defined earlier, the problem can be set up as in Eq. (12), which is a Equality Constrained Ordinary Least Squares (ECOLS) problem.

$$\begin{aligned} &\text{argmin}_{\boldsymbol{c}} \mathcal{J}(\boldsymbol{c}) \\ &\text{subject to } \mathbf{H}\boldsymbol{c} = 0 \end{aligned} \tag{12}$$

Various solvers have been used for solving this regression problem [9, 18, 19], but all of these solve a global problem, and do not completely profit from the local cost function of the Simplotope B-Spline structure. The Alternating Direction Method of Multipliers (ADMM) is introduced in Section IV.A, with the Variable Splitting ADMM (VSADMM) version explained in Section IV.B. VSADMM is fully parallel and able to estimate the coefficients of the B-Splines at a lower computational complexity, as described in Section IV.C. Additional options that are used can be found in Section IV.D.

A. Alternating Direction Method of Multipliers

ADMM is able to solve convex optimization problems in a distributed fashion, and has been used in various scientific fields [27]. It has also been applied in combination with multivariate Simplex B-Splines, solving a 2-dimensional wavefront reconstruction problem for adaptive optics purposes [20]. In general, ADMM solves equations of the form in

Eq. (13); minimizing the sum of two separable convex functions, subject to linear (equality) constraints.

$$\begin{aligned} &\text{minimize} \;\; f(x) + g(z) \\ &\text{subject to} \;\; \mathbf{A}x + \mathbf{B}z = c \end{aligned} \tag{13}$$

It solves the system by forming the augmented Lagrangian in Eq. (14). The penalty factor $\rho > 0$ is chosen to penalize not satisfying the constraints, and thus reaching a feasible solution faster. There are analytical solutions for an optimal penalty factor for quadratic problems [28, 29], but these are not fully applicable to the B-Splines case. For robustness against badly chosen penalty factors, an adaptive approach as explained in Section IV.D.2 has been taken.

$$\mathcal{L}(x, z, \lambda) = f(x) + g(z) + \lambda^T (\mathbf{A}x + \mathbf{B}z - c) + \frac{\rho}{2}||\mathbf{A}x + \mathbf{B}z - c||_2^2 \tag{14}$$

The Lagrangian is minimized by updating the parameters x and z in an alternating fashion, as in Eqs. (15) and (16), combined with the dual update step in Eq. (17). Because of the augmentation term in the Lagrangian, the algorithm has been proven to converge under mild assumptions. On top of that, if there is an optimal, unique solution, ADMM will converge to that value [27].

$$x^{k+1} = \text{argmin}_x \mathcal{L}(x, z^k, \lambda^k) \tag{15}$$

$$z^{k+1} = \text{argmin}_z \mathcal{L}(x^{k+1}, z, \lambda^k) \tag{16}$$

$$\lambda^{k+1} = \lambda^k + \rho \left(\mathbf{A}x^{k+1} + \mathbf{B}z^{k+1} - c \right) \tag{17}$$

The z-update in Eq. (16) uses the updated value of the parameters x^{k+1}, which means that the steps can only be performed sequentially. There are schemes known using the values of the previous iteration, i.e. x^k in Eq. (16), but these do not necessarily have the same convergence properties, and are not even always convergent in general [30].

This algorithm is applicable to Simplotope B-Splines, as there are multiple, separable cost functions to be optimized, with global equality constraints. In that case there would be more than two alternating directions, unless the domain is subdivided into only two partitions. However, the general extension of this algorithm, with three of more alternating directions, is not necessarily convergent [31] and thus may lead to an unstable algorithm. Nevertheless, a general extension has been written and resulting in convergence towards the global optimal solution. However, this still leads to a sequential algorithm, as updates from the current iteration are required for convergence. However, not only a distributed algorithm, but a fully parallel algorithm is preferred, as this can run on multiple cores and thus have better scaling properties. Various options have been presented in literature to parallelize the algorithm, such as adding a proximal term, using a relaxation step, or a combination of both [30, 32, 33]. The Variable Splitting method as described

in Section IV.B is used for the distributed aerodynamic model identification approach.

B. Variable Splitting ADMM

By making the algorithm in a distributed manner, the full tessellation is divided into $P \leq J$ partitions. This can either be partitions of a single simplotope, or with multiple simplotopes. The distributed algorithm iteratively optimizes the coefficients, and enforces the constraints as composed in the continuity matrix $\mathbf{H}$. Using VSADMM, a vector of coupling coefficients z_p is introduced for each partition p, such that $\mathbf{H}_p \boldsymbol{c}_p = z_p$ [33, 34]. Here $\mathbf{H}_p$ are the columns of the continuity matrix $\mathbf{H}$ that are depending on the coefficients $\boldsymbol{c}_p$, i.e. $\mathbf{H} = \begin{bmatrix}\mathbf{H}_1 \dots \mathbf{H}_p \dots \mathbf{H}_P\end{bmatrix}$. To have the continuity conditions satisfied, the additional constraint $\sum_{p=1}^{P} z_p = 0$ is posed on these coefficients. Using this coupling coefficient, the optimization problem can be stated as in Eq. (18).

$$\begin{aligned} &\text{argmin}_{\boldsymbol{c}}\ \mathcal{J}(\boldsymbol{c}) = \textstyle\sum_{p=1}^{P} \frac{1}{2}||\mathbf{X}_p \boldsymbol{c}_p - \boldsymbol{Y}_p||_2^2 \\ &\text{subject to} \quad \mathbf{H}_p \boldsymbol{c}_p = z_p \\ &\qquad\qquad \textstyle\sum_{p=1}^{P} z_p = 0 \end{aligned} \tag{18}$$

Setting up the augmented Lagrangian in Eq. (19) for this optimization problem, the constraints put on the coupling coefficients are exchanged for an indicator function $\mathcal{I}_{\mathcal{Z}}$, where $\mathcal{Z}$ is the convex set $\left\{[z_1, \dots, z_P] : \sum_{p=1}^{P} z_p = 0\right\}$ [33]. This function will indicate 0 when the solution is part of the set $\mathcal{Z}$, and ∞ otherwise.

$$\mathcal{L}(\boldsymbol{c}, z, \lambda) = \sum_{p=1}^{P} \frac{1}{2}||\mathbf{X}_p \boldsymbol{c}_p - \boldsymbol{Y}_p||_2^2 + \mathcal{I}_{\mathcal{Z}} + \lambda^T \left(\mathbf{H}_p \boldsymbol{c}_p - z_p\right) + \frac{\rho}{2}||\mathbf{H}_p \boldsymbol{c}_p - z_p||_2^2 \tag{19}$$

The decoupling of the partitions is clear, as the Lagrangian in Eq. (19) is purely a sum of P individual Lagrangians of the partitions. By taking the derivative of this Lagrangian w.r.t. the coefficients $\boldsymbol{c}_p$, no influences from other partitions are present. Therefore, the optimal update as in Eq. (23) is found by setting the derivative to zero and solving for the coefficients $\boldsymbol{c}_p$.

The coupling coefficients are still linked to other partitions, as the sum of all has to be zero. Therefore this update is found in two steps, by first finding the unconstrained optimal update in Eq. (20).

$$z_p = \mathbf{H}_p \boldsymbol{c}_p + \frac{\lambda_p}{\rho} \tag{20}$$

The next step is to correct for the global constraints put on the coupling coefficients. For all constraints, this compensation can be done to subtract the sum of all partitions' unconstrained contributions to constraint j, divided by the number of partitions m_j that appear in that constraint [34]. Knowing that the $\mathbf{H}$ matrix is sparse, the rows are reduced in further computations. Let $\mathcal{H} = 1, 2, \dots, R$ be the indices of all continuity conditions, of which two subsets can be formed: $\mathcal{H}_p$

includes the indices of the internal constraints of partition p, and $\mathcal{H}_{p,n}$ those of the constraints between partition p and its direct neighbors. The rows of $\mathbf{H}_p$ are limited to $\mathcal{H}_p \cup \mathcal{H}_{p,n}$, which means that $\boldsymbol{z}_p \in \mathbb{R}^{R_p}$, with $R_p = |\mathcal{H}_p \cup \mathcal{H}_{p,n}|$. This also makes that $\lambda_p \in \mathbb{R}^{R_p}$. For every partition, only the coupling coefficients of constraints in which it appears should be updated.

Using these reduced sets, the update rule for the coupling coefficients can be found in Eq. (21). For all constraints, the satisfaction of the global constraints as in $\mathbf{H}$ can be confirmed by summing all the contributions $\boldsymbol{z}_{p,j}$.

$$\boldsymbol{z}_{p,j} = \left(\mathbf{H}_{p,j}\boldsymbol{c}_p + \frac{\lambda_{p,j}}{\rho}\right) - \frac{1}{m_j}\left(\sum_{i=1}^{P}\mathbf{H}_{i,j}\boldsymbol{c}_i + \frac{\lambda_{i,j}}{\rho}\right), \quad \forall j \in \left(\mathcal{H}_p \cup \mathcal{H}_{p,n}\right) \tag{21}$$

In the case of Simplotope B-Splines, there are only two possibilities for m_j: either a constraint is internal, part of $\mathcal{H}_p$, which means that $m_j = 1$; or a constraint is part of $\mathcal{H}_{p,n}$, and then $m_j = 2$, because a continuity constraint is always between exactly two simplotopes. If $m_j = 1$, the coupling update will simply reduce to 0. The update steps for iteration $k+1$, that have to be carried out per partition, are shown in Eqs. (22) to (24).

$$\boldsymbol{z}_{p,j}^{k+1} = \begin{cases} \left(\mathbf{H}_{p,j}\boldsymbol{c}_p^k + \frac{\lambda_{p,j}^k}{\rho}\right) - \frac{1}{2}\left(\sum_{i=1}^{P}\mathbf{H}_{i,j}c_i^k + \frac{\lambda_{i,j}^k}{\rho}\right), & \forall j \in \mathcal{H}_{p,n} \\ 0, & \forall j \in \mathcal{H}_p \end{cases} \tag{22}$$

$$\boldsymbol{c}_p^{k+1} = \left(\mathbf{X}_p^T\mathbf{X}_p + \rho\mathbf{H}_p^T\mathbf{H}_p\right)^{-1}\left(\mathbf{X}_p^T\boldsymbol{Y}_p + \rho\mathbf{H}_p^T\boldsymbol{z}_p^{k+1} - \mathbf{H}_p^T\lambda_p^k\right) \tag{23}$$

$$\lambda_p^{k+1} = \lambda_p^k + \rho\left(\mathbf{H}_p\boldsymbol{c}_p^{k+1} - \boldsymbol{z}_p^{k+1}\right) \tag{24}$$

The iterative updates of the coefficients and the duals in Eqs. (23) and (24) respectively do not depend in any case on updated iterations of other partitions. This means that each iteration can run fully in parallel, limited by either the number of partitions, or the amount of parallel pools that are available.

C. Complexity Analysis

The complexity of the VSADMM is analyzed and compared to a global estimation. A completely parallel implementation is assumed, with the necessary amount of parallel pools available and perfect communication between the partitions.

The coupling coefficient step in Eq. (22) can be updated with $O\left(R_p\right)$, with R_p the number of continuity conditions affecting partition p, as the matrix $\mathbf{H}_{p,j}$ is sparse. The same holds for the dual update step in Eq. (24). The amount of continuity conditions in a single partition is $O\left(\hat{d}\right)$ for a simplotope, and thus scales linearly with increasing partition size, but does not depend on the total number of partitions. Combining all the operations in the two update steps results in a $O\left(ck\hat{d}\right)$ complexity, where the constant c depends on the number of non-zeroes in the local constraint matrix and the multiplication and additions for Eqs. (22) and (24). Generally, $c < \hat{d}$, so the worst case is assumed as $c = \hat{d}$.

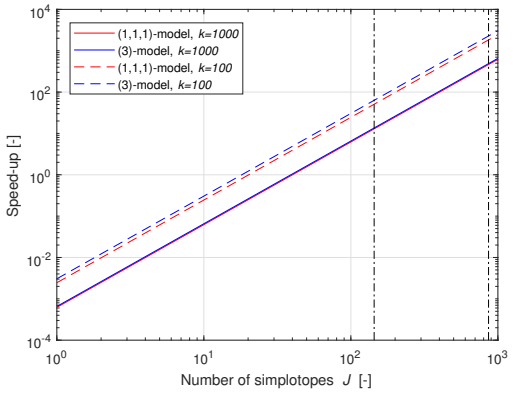


Fig. 5 Speed-up by using VSADMM, with vertical lines at 144 and 864 Simplotopes

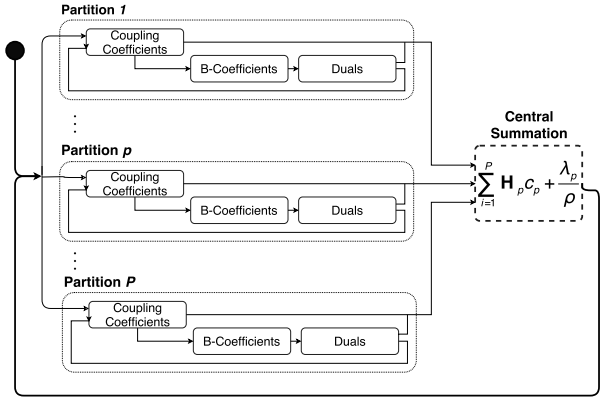


Fig. 6 Parallel flow of VSADMM

The most expensive operations are the inversion and vector-matrix multiplication in Eq. (23). Finding this results in a single $O(\hat{d}^3)$ operation, followed by k iterations with $O(2\hat{d}^2)$ operations, making the total complexity $O\left(\hat{d}^3 + 2k\hat{d}^2 + ck\hat{d}\right)$.

The algorithm gives the possibility of infinite scalability in terms of refinement of the tessellations, as this only increases the number of partitions, but not the individual size. Therefore, if a tessellation is refined from J to $2J$ simplices, adding J parallel pools will result in no increment in the computational cost of estimating the parameters.

The complexity is compared to the commonly used global method, the iterative solver in [19]. The solver typically converges in 2-3 iterations, requiring $O\left(J^2\hat{d}^2\right)$ FLOPs per iteration because of the sparse, global matrix that is solved not needing a $J^3\hat{d}^3$ operation first because of the sparseness of the matrix. For the distributed approach, assuming two iterations for the global solver, this will result in the speed-up in Eq. (25) when VSADMM works fully in parallel.

$$O\left(\frac{2J^2\hat{d}^2}{\hat{d}^3 + 2k\hat{d}^2 + ck\hat{d}}\right) = O\left(\frac{2J^2}{\hat{d} + 3k}\right) \tag{25}$$

Equation (25) breaks even at $J = \sqrt{\frac{1}{2}(\hat{d} + 3k)}$. The speed-up is shown in Fig. 5 for two models that will be introduced in Table 4 in Section V: a $(1, 1, 1)$-model with 216 coefficients per simplotope and a (3)-model with 56 coefficients. Both have $k = 1,000$ iterations, which is generally sufficient to obtain a good approximation of the aerodynamic model. Note that in these cases the $\hat{d}$ is small compared to the amount of iterations. For reference, two lines for 100 iterations are added as well.

For both models the distributed method is beneficial from approximately 45 simplotopes. The two vertical lines indicate the 144 partitions that are being used in the $(1, 1, 1)$-model, which has a speed-up of approximately 10, and

the 864 simplotopes in the (3)-model, that can achieve a speed-up as high as 500. This only improves when e.g. 100 iterations are used. It clearly indicates that the distributed approach achieves increasingly beneficial speed-ups with increasing number of partitions. Reducing the number of iterations will have a major effect on this speed-up, with some suggestions provided in Section IV.D.

This parallel workflow is depicted in Fig. 6. The communication overhead between different partitions is limited by the unconstrained, optimal coupling update in Eq. (20). These are collected and summed at a central point, after which the sum $\frac{1}{2}\sum_{i=1}^{P}\mathbf{H}_i\boldsymbol{c}_i + \frac{\lambda_i}{\rho}$, as used in Eq. (21), needs to be communicated back to the partitions. This centralized operation will receive $\sum_{p=1}^{P}|\mathcal{H}_{p,n}|$ scalars from the partitions, and will have to communicate as many back after they have been summed. The complexity analysis, and thus the speed-up, does not take any transport latency into account that may occur when these values are communicated back and forth.

D. ADMM Variations and Improvements

As a lot of research has been done into ADMM algorithms, numerous improvements and speed-ups are found [27]. First, a stopping criterion has been defined Section IV.D.1, which sets feasibility tolerances on the dual- and the primal function for accepting suboptimal, but sufficiently optimal, solutions. While exploring, it occurred that the choice of the penalty factor ρ had a major effect on the convergence properties and the quality of the solution. In order to make the algorithm more robust against this initial choice of ρ, it is made adaptive as explained in Section IV.D.2. More options and variations are available for ADMM, such as over-relaxation and fast-ADMM [27, 35], but these are not implemented.

1. Stopping Criterion

In order to assess the VSADMM's solution during the iterations, two residuals are used [27]: the primal residual $\boldsymbol{r}^{k+1}$, indicating the feasibility of the primal function, and the dual residual $\boldsymbol{s}^{k+1}$, indicating the feasibility of the dual function and thus the optimality of the primal function. These two residuals for iteration $k+1$ are given by Eqs. (26) and (27). The constraints on the coupling coefficients, $\sum_{p=1}^{P}\boldsymbol{z}_p = 0$, have not been included here, as the update step for $\boldsymbol{z}_p$, in Eq. (22), per definition enforces these constraints.

$$\boldsymbol{r}_p^{k+1} = \mathbf{H}_p\boldsymbol{c}_p^{k+1} - \boldsymbol{z}_p^{k+1} \tag{26}$$

$$\boldsymbol{s}_p^{k+1} = -\rho\mathbf{H}_p\left(\boldsymbol{c}_p^{k+1} - \boldsymbol{c}_p^{k}\right) \tag{27}$$

The stopping criteria can be defined as $||\boldsymbol{r}_p^{k+1}||_2 \leq \epsilon^{pri}$ and $||\boldsymbol{s}_p^{k+1}||_2 \leq \epsilon^{dual}$, with the tolerances ϵ as defined in Eqs. (28) and (29). Here $\epsilon^{abs} \geq 0$ and $\epsilon^{rel} \geq 0$ are the absolute and relative tolerance respectively, and can be chosen to set the

desired accuracy. Here, the accuracies are set to be $\epsilon^{abs} = 10^{-8}$ and $\epsilon^{rel} = 10^{-6}$ for the absolute and relative accuracy respectively. The stopping criterion is also defined per partition, such that each partition will have a primal- and dual feasible solution.

$$\epsilon^{pri} = \sqrt{\hat{d}_p}\epsilon^{abs} + \epsilon^{rel} \max\left\{||\mathbf{H}_p \boldsymbol{c}_p^{k+1}||_2, || - \boldsymbol{z}_p^{k+1}||_2\right\} \tag{28}$$

$$\epsilon^{dual} = \sqrt{R_p}\epsilon^{abs} + \epsilon^{rel}||\mathbf{H}_p^T \lambda_p^{k+1}||_2 \tag{29}$$

Additionally to the stopping criterion, a maximum amount of iterations should be set in case the criterion will not be reached. A sufficiently high number, such as 1,000, can be chosen to obtain a suboptimal solution. In this case the solution should be treated and analyzed carefully, as it might not be close to an optimal or feasible solution yet.

2. Adaptive Penalty Factor

To make the algorithm more robust against the choice for penalty factor ρ, a method that changes this parameter per iteration has proven to be effective in practice [36, 37]. Since it is best to let the dual and the primal converge to feasibility at the same pace, the two residuals are compared as in Eq. (30).

$$\rho_{k+1} = \begin{cases} \rho_k \cdot (1 + \tau_k) & \text{if } ||\boldsymbol{r}^k||_2 > \mu \cdot ||\boldsymbol{s}^k||_2 \\ \rho_k \cdot \frac{1}{1+\tau_k} & \text{if } \mu \cdot ||\boldsymbol{r}^k||_2 < ||\boldsymbol{s}^k||_2 \\ \rho_k & \text{otherwise} \end{cases} \tag{30}$$

This adaptive scheme increases ρ when the primal residual is more than a factor μ higher than the dual residual, and decreases ρ when it is μ times lower. In this way it changes the focus from primal to dual feasibility, if the two are converging at a different pace. In this case, τ_k has been chosen as 1, and $\mu = 5$. At every update of the penalty factor, the new iteration is initiated with the coefficients and duals from the previous iteration.

To make the estimation stable, $\sum_{k=1}^{\infty} \tau_k$ should be finite [36, 37]. For this reason, the penalty factor is only allowed to change 5 times before it has to remain constant and $\tau = 0$.

V. Simulations and Results

The results are divided into two main parts. First, in Section V.A the performance of the distributed approach is analyzed by comparing the distributed solutions with the global solution. This has been done with the adaptive penalty factor, and two different starting values for ρ: 1 and 0.01. The tests are performed on a 3-dimensional model, to which noise has been added of various noise intensities. The models with first order continuity, that have been used in the estimation of the ICE model, are implemented in Section V.B and compared to the ICE's Simulink model. For all results,

the dataset and hypercube partitioning from [21] have been used. These are triangulated using Kuhn's triangulation. This triangulation method is chosen over others, such as Delaunay, because it gives conforming simplices, required for good edges, and simplices of a single congruence class, providing practical benefits [38]. The same, extended datasets have been used as well, using single imputed datapoints, allowing an increase of model degree.

A. Evaluation of the Distributed Approach

This section assesses the performance of the VSADMM algorithm compared to the global solution. The datasets are artificially contaminated with noise by using a method described in [8], in order to show the performance in different noisy environments. For a noise intensity of $k\%$, the noise can be generated by using a uniform distribution $\nu = U(-0.5, 0.5)$ and Eq. (31).

$$\nu(k) = 0.01 \cdot k \cdot (\max \boldsymbol{Y} - \min \boldsymbol{Y}) \cdot \nu \tag{31}$$

Before the noise is added to the data, a distinction is made between the identification data and the validation data. Noise has only been added to the identification data, and the methods are validated by using clean data. The validation set consists of 20% of the full set and is chosen randomly.

Four different tessellations of the model have been compared, as described in Table 4. The combinations are chosen in order to show performance at several layer divisions, from fully decoupled to fully coupled dimensions. The $(2, 1)$-simplotope contains α and β together in one layer and M in its own layer. The (3)-simplotope model has all three dimensions in a single layer and is essentially a 3-dimensional simplex spline. In order to assess the adaptability of the distributed method in a bigger partition size, the simplex model has also been partitioned per hypercube, each consisting of 6 simplices.

Table 4 Layer compositions of $C_{X_2}(\alpha, \beta, M)$ used for noise comparisons.

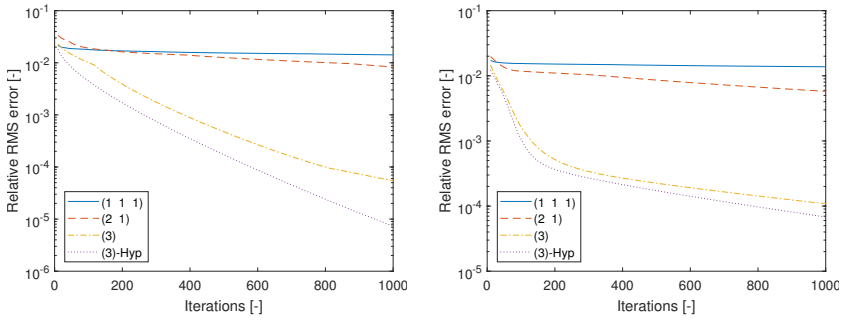
Layer Division	Dimensions	Continuity	Simplotopes	Partitions
$(1, 1, 1)$	$(5, 5, 5)$	$(1, 1, 1)$	144	144
$(2, 1)$	$(5, 5)$	$(1, 1)$	296	296
(3)	(5)	(1)	864	864
(3)	(5)	(1)	864	144

Because of the tensor product between the layers, the $(1, 1, 1)$ model consists of terms with a total degree of 15. These terms might result in unrealistic model terms that help fitting a good model through the identification data, but result in high validation errors.

The maximum amount of iterations that has been set is 1,000 for the distributed approach. The global solution is found using the iterative solver described in [19], which has been limited to a maximum of 250 iterations. The distributed solver does generally not reach the stopping criterion and is thus limited the maximum amount of iterations,

while the global iterative solver typically reaches convergence before that.

Figure 7 shows convergence properties of the distributed algorithm for the several tessellations. It computes the relative RMS between the intermediate solutions of the VSADMM and the global solutions. The $(1, 1, 1)$- and $(2, 1)$-simplotopes advance to the the global solution slowly after approximately 200 iterations, but reaches within a few percent. On the other hand, the (3)-models approximate the global solution better and keep on improving until 1,000 iterations. The hypercube partitioned model performs slightly better, but is also computationally more complex. The simplotope models benefit slightly from reducing the initial penalty factor ρ to 0.01, while this has a worsening effect on the simplex models. Possible explanations for the difference between simplotope and simplices are that degrees of freedom it contains. Simplices have considerably more constraints than simplotopes, giving the simplotope spline more coefficients to be optimized. To conclude, even though the adaptive penalty factor reduces the effect of a bad initial choice, it still has an influence and thus should be considered at initialization of the algorithm.

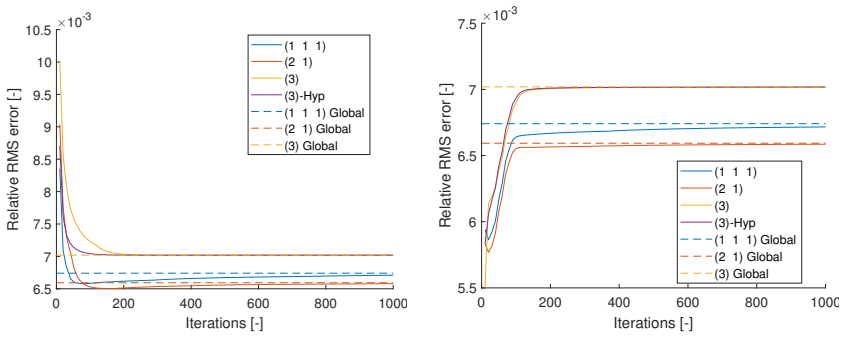


(a) $\rho = 1.0$ **(b)** $\rho = 0.01$

Fig. 7 Convergence of the distributed algorithm towards the global optimum, at 0% noise intensity.

Similarly, at 0% noise, Fig. 8 shows that the validation performance of the distributed algorithm, in terms of relative RMS validation error, converges to the global results as well. Comparing Figs. 8a and 8b shows a difference between how the choice of penalty factor affects the approach to the global solution. A high ρ (Fig. 8a) puts more weight on satisfying the constraints and is less optimal at first than starting with a low ρ in Fig. 8b. When iterating and thus enforcing the constraints more, will make the results advance to the global solution. Because these figures are made at 0% noise, local estimations are likely to be better than the ones with continuity more strictly enforced.

To assess constraint satisfaction, two metrics are used: the mean constraint mismatch, $||\mathbf{H}\boldsymbol{c}||_1/R$, and the maximum constraint mismatch, $||\mathbf{H}\boldsymbol{c}||_\infty$. Figure 9 shows that in 1,000 iterations, in all cases the constraints are approximately satisfied for both settings of the penalty factor. It is not completely (numerically) zero yet, as would be the case with global methods, but is getting close with a maximum mismatch at 100% noise in the order of 10^{-4}. The common trend

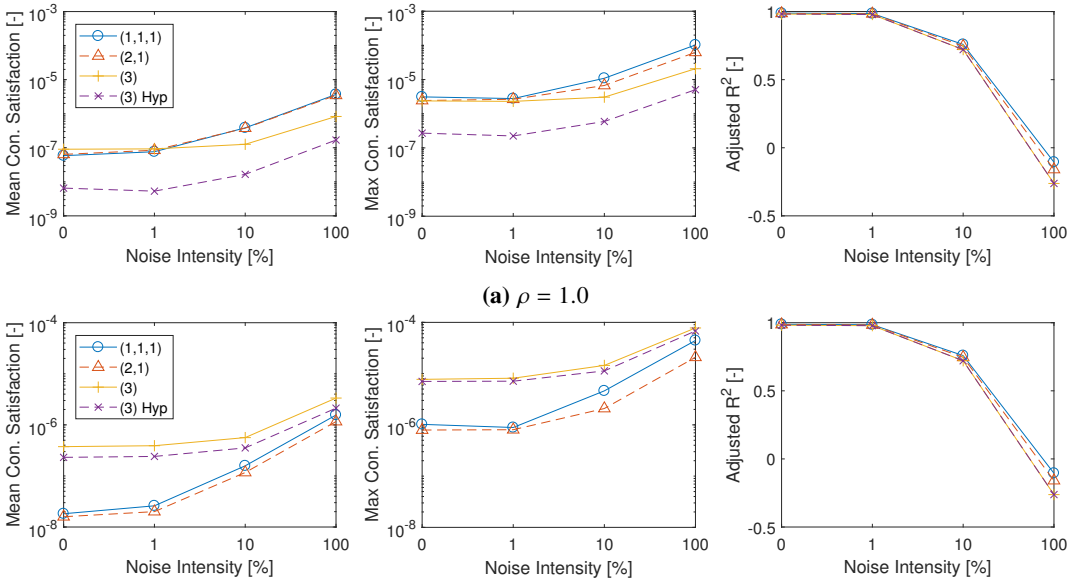


(a) $\rho = 1.0$ **(b)** $\rho = 0.01$

Fig. 8 Convergence of the validation relative RMS error of the distributed algorithm towards the global optimum, at 0% noise intensity.

is that the constraint satisfaction worsens with increasing noise levels, indicating that noisy environments would require more iterations in case the same quality of continuity is desired.

In Fig. 9a the simplex models show better continuity than the simplotopes, while Fig. 9b show the opposite. In both cases the coefficients of determination show equal values, with simplotopes that have more decoupled dimensions, performing better in fitting to the identification data. The fact that the simplex models show the same coefficient of determination in both partition sizes, verifies these have the same solutions.



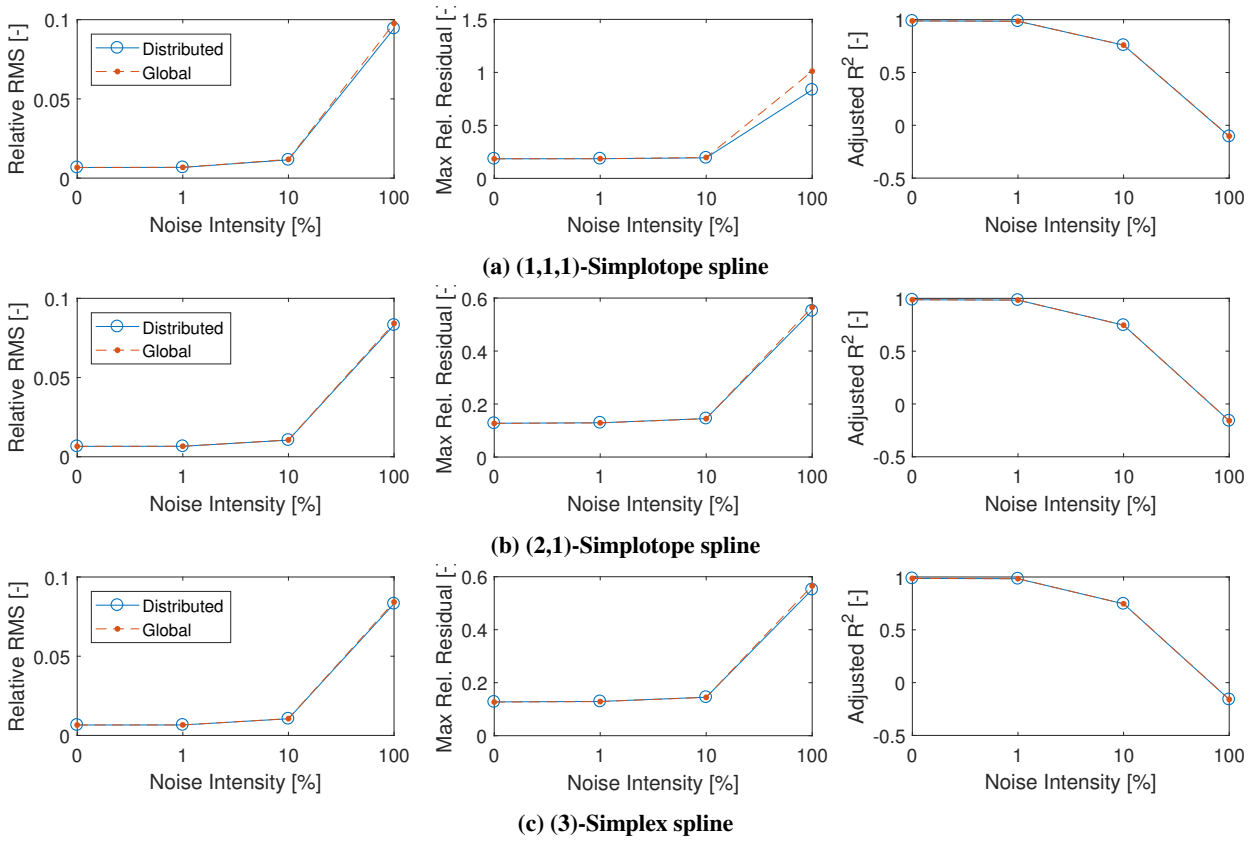
(a) $\rho = 1.0$

(b) $\rho = 0.01$

Fig. 9 Mean and Maximum constraint satisfaction and the coefficient of determination for the 3-dimensional test cases.

Comparisons in Fig. 10 indicate that for all layer combinations the fitness to the identification data and the validation

results are close or equal to that of the globally estimated solution. A slight difference in the maximum relative residual can be found in Fig. 10a for the $(1, 1, 1)$-simplotope spline. However, it is remarkable that in this case the distributed, suboptimal solution performs better than the global solution.



(a) (1,1,1)-Simplotope spline

(b) (2,1)-Simplotope spline

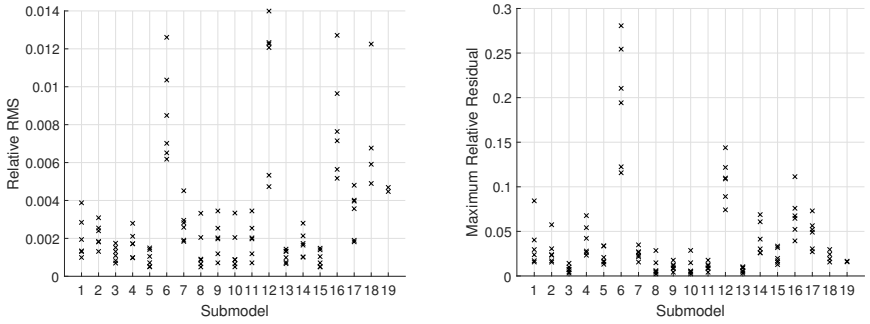
(c) (3)-Simplex spline

Fig. 10 Comparison between the results of the distributed- and global approaches in terms of: relative RMS validation error, relative maximum validation residual, and adjusted coefficient of determination. ($\rho = 1.0$.)

When comparing the results between the different model structures in Fig. 10, the RMS and the coefficients of determination are comparable. The $(1, 1, 1)$-simplotope structure shows slightly better performance in the case of the fitness, as can be seen in Fig. 9 as well. On the other hand, the RMS shows slightly worse performance when more dimensions are decoupled at higher noise intensities. The maximum residual of the $(1, 1, 1)$-simplotope, is considerably worse than those of the other two model structures, with residuals up to 100%. The Simplotope B-Spline theory in Section III already indicated that high cross-coupling terms occur in simplotope polynomials. In the $(1, 1, 1)$-simplotope model with degrees $(5, 5, 5)$ consists of polynomial terms with a combined degree of 15. Combining these high degree terms with the higher coefficient of determination and bad maximum residual, this indicates an overfitting regression. This is especially harmful at high noise intensities, because the unnatural terms might have a statistical significant effect in fitting the model, while in reality there is none.

B. Implementation in the Innovative Control Effectors model

Various model combinations for all the submodels, in terms of layers' divisions and degrees, are tested in order to find the best performancing models. The combinations that were found to perform best were implemented in the final ICE Simplotope model. The details of the models can be found in Table 6 in the Appendix. The validation results of the chosen models can be seen in Fig. 11, showing the relative RMS in Fig. 11a and the maximum relative residual in Fig. 11b. The maximum RMS of any submodel is 1.40%, while the maximum relative error that is found is 28.1%. Of all 108 submodels, the median of the RMS is at 0.192% and that of the maximum residual at 2.33%. These results are comparable to the validation results of the zeroth order continuity model developed in [21].



(a) RMS error **(b) Maximum residual**

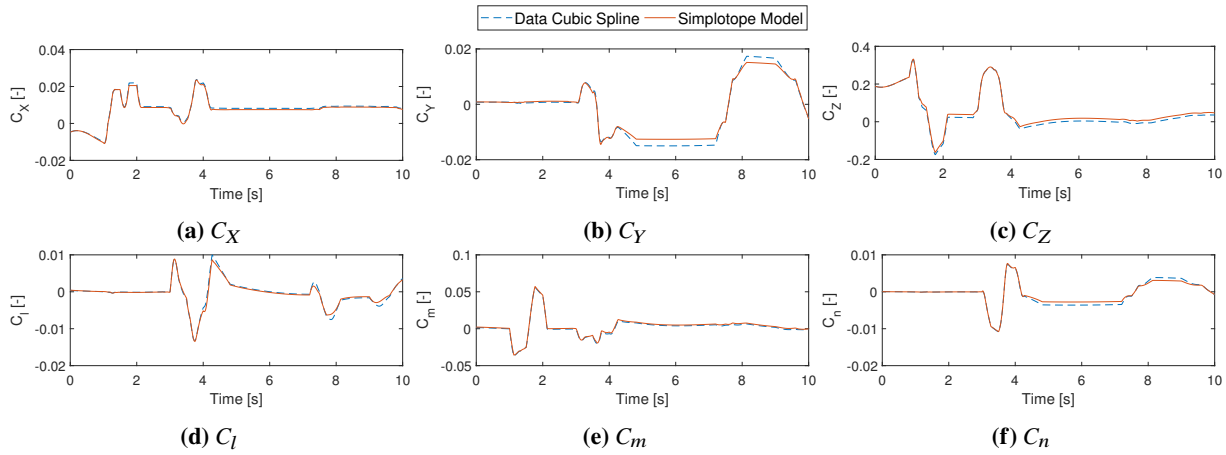
Fig. 11 Relative validation results of all the implemented simplotope models

The performance of the Simplotope model was compared with the model as implemented in Simulink. In order to make the Simplotope model able to be evaluated in real time, an optimized evaluation function had to be written. For ICE's Simplex model a `MEX` function was written in C++ [21], which has been extended to work with Simplotopes as well and is able to perform over 2,000 evaluations every second*. The `MEX` function was placed in Simulink as an embedded `Matlab` block. The version of ICE's Simulink model used for comparison is the one that uses a cubic spline to interpolate the measurements and clips all values as extrapolation mode. Moreover, the inputs are combined with the linear models for the actuators, in order to provide a more realistic input. [3, 21, 22].

The first input is a doublet on both elevons symmetrically, followed by a block input on the left AMT and a block input on the right AMT. The resulting model responses are given in Fig. 12.

It can be seen that the model generally does not have any problems with following the coefficients as determined by the Simulink model. However, after approximately 4 seconds, the estimations start to differ slightly. When comparing

*With Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz

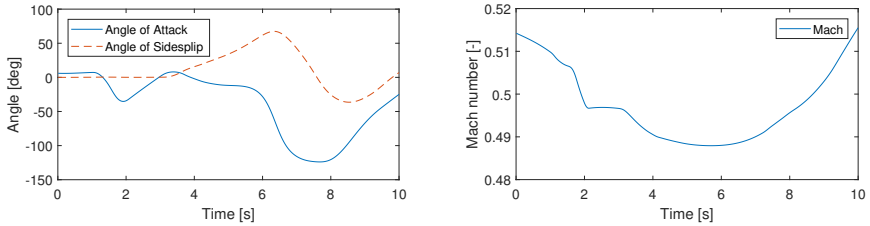


(a) C_X **(b)** C_Y **(c)** C_Z

(d) C_l **(e)** C_m **(f)** C_n

Fig. 12 Simplotope Models' Results compared with ICE's Simulink Model after elevons and AMT inputs

this point in time with the states' responses in Fig. 13, it corresponds to the moment the Angle of Attack reaches out of bounds. Although both the models have clipping extrapolation modes, yet a difference occurs.



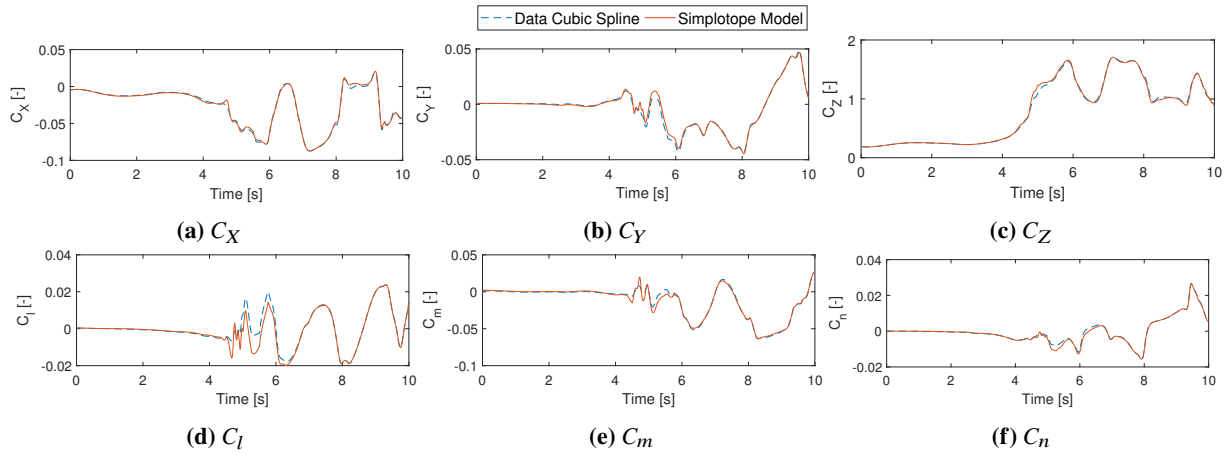
(a) Angle of Attack and Angle of Sideslip **(b) Mach number**

Fig. 13 Aerodynamic states' responses to the elevons and AMT inputs.

The second input is a combined input of the Differential Leading Edge Flaps and the Spoiler Slot Deflectors, with the responses shown in Fig. 14. Although the difference between the two responses are bigger than above, the simplotopes still provide a good approximation of the Simulink model.

The aerodynamic states resulting from the inputs are shown in Fig. 15. Here the aerodynamic angles stay within the range of the simplotope models, but the Mach number in Fig. 15b reduces below 0.3, which is outside the boundaries. However, the inaccuracies in Fig. 14 are not due to this behavior, as this occurs before the mach number reaches out of bounds. The inaccuracies at 5-6 seconds occur becuase that is the moment the Spoiler Slot Deflector are extended, with several models reacting badly at high inputs of those.

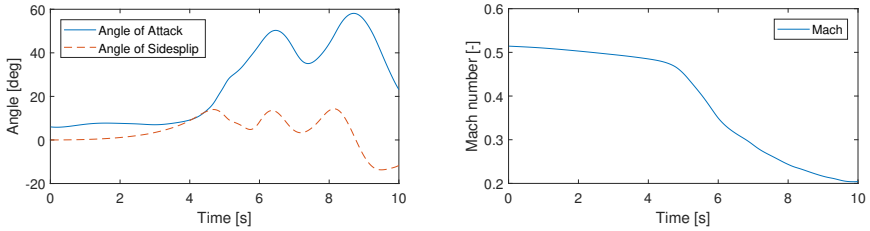
An overview of these and two other inputs is given in Table 5. Two additional scenarios are: no inputs at all during 10 seconds and sequentially deflecting all 11 control effectors. It can be seen that the maximum RMS that is seen in this case is 6.54% between the original baseline model and the estimation of all 108 simplotope methods. There is no coefficient that performs generally better or worse than the others. The difference between the total results and



(a) C_X **(b)** C_Y **(c)** C_Z

(d) C_l **(e)** C_m **(f)** C_n

Fig. 14 Simplotope Models' Results compared with ICE's Simulink Model after DLEF and SSD inputs



(a) Angle of Attack and Angle of Sideslip **(b) Mach number**

Fig. 15 Aerodynamic states' responses to the elevons and AMT inputs.

the individual submodel's results in Fig. 11 can be explained by the fact that many models get out of bounds during operation, while the error is cumulated through all the models.

Table 5 Scaled RMS between the Simulink data and the Simplotope Model during 10 seconds.

Input	C_X	C_Y	C_Z	C_l	C_m	C_n
None	2.14%	1.90%	2.82%	3.36%	2.75%	2.18%
Elevons / AMTs	1.75%	4.39%	2.50%	1.99%	1.23%	2.78%
DLEFs / SSDs	1.54%	2.10%	1.71%	6.54%	2.76%	1.68%
All	1.20%	1.56%	1.55%	6.41%	2.49%	3.03%

Another observation from the simplotope model responses in Figs. 12 and 14 is that it contains no visible ripples or discontinuities, even though the distributed approach has been taken. This justifies the reason why first order continuity has been enforced, as the line is smoothly following the original model. Moreover, it is a confirmation of the performance of the distributed approach in terms of constraint satisfaction.

VI. Conclusion

A novel distributed approach for Aerodynamic Model Identification has been presented, using the Alternating Direction Method of Multipliers (ADMM) in combination with multivariate Simplotope B-Splines. The simplotope splines have the possibility to decouple dimensions into separate layers, containing lower dimensional simplex splines; each of these have their own individual properties as triangulation, model degree, and continuity order. These layers are combined by taking the tensor product, resulting in simplotopes with an own set of barycentric coordinates and B-Coefficients. Using simplotopes gives more flexibility in choosing a model structure, gives better shaped domains, and reduces the number of coefficients and constraints. However, it may also introduce high degree cross coupling terms that may result in unwanted overfitting.

The distributed approach divides the full tessellation into a number of partitions. The Variable Splitting ADMM (VSADMM) method fully decouples the estimation of the coefficients of the individual partitions. The global continuity matrix is split up by introducing coupling coefficients for each constraint. Global continuity is enforced, by requiring the sum of the coupling coefficients, over all partitions, to be zero. Because VSADMM is able to work fully in parallel, the complexity is determined solely by the partition size, and refining the grid will not increase complexity as parallel pools can be added. The only centralized operation is gathering the coupling coefficients, summing them, and return the sum to the partitions. This limits the communication between the partitions to twice as many scalars as there are continuity equations.

The performance of the distributed approach has been evaluated by adding noise at various intensities to a 3-dimensional submodel of the Innovative Control Effectors (ICE) aircraft. This model has been evaluated for several combinations of layers of the dimensions and different partition sizes, for initial penalty factors ρ of 1.0 and 0.01.

The simplotope models did converge to a value within a few percents of the global optimal solution, but the pure simplex models kept on advancing beyond an RMS of 10^{-4} compared to the global solution. Bigger sized partition performed better and are beneficial for the quality, but is also computationally more expensive.

In all cases the global continuity constraints were enforced well, with maximum mismatches in the order of 10^{-4} after 1,000 iterations, only for the highest noise levels. In general, higher noise intensities resulted in worsened enforcement of the constraints, and thus require more iterations to get equal continuity quality. A higher penalty factor was beneficial for the pure simplex model, while the simplotopes performed better with lower penalty factor. Even though an adaptive penalty factor has been used, the initial choice still affects the final solution of the ADMM algorithm.

For all combinations, the validation performance, in terms of relative RMS error, and maximum relative error, were similar to the global results, as well as the fitness to the identification data in terms of adjusted coefficient of determination. In case of the $(1, 1, 1)$-simplotope model, the global solution was a little worse than the distributed one. This model showed considerably worse maximum residuals, when the noise intensity reached 100%. Because the coefficient of determination is still high, this is a result of overfitting, caused by the high degree polynomial terms

formed by the tensor product between the layers.

The distributed approach is used to estimate the aerodynamic model of the ICE-aircraft with first order continuity. All 108 submodels were estimated with several layer combinations, and the best in terms of validation RMS error and maximum residual, were used in the final model. All models had a relative RMS below 1.4%, and the maximum relative error was 28.1%. These values are similar to the results of the zeroth order continuous model.

For real time simulation in Simulink, all the models were implemented in an optimized `MEX` evaluation function, able to perform over 2,000 evaluations every second. The model followed the original model with RMSs of 7% maximum, which is mainly because of the aerodynamic states getting out of bounds of the model domain. Contrary to the zeroth order continuity model, it showed no ripples due to discontinuities in the first derivative. This justifies the higher order continuity, and confirms the functionality of the distributed estimator when it comes to constraint satisfaction.

Appendix

Table 6 Overview of the simplotope models of the ICE aircraft and their properties.

Coefficient	Parameters	Data Range	Datapoints	Cubes	Layers	Degrees	Simplotopes	Coefficients
C_{*_1}	α	[-5.0, 90.0]	48	12	(1, 1)	(7, 6)	120	6,720
	Mach	[0.3, 2.16]	10	10				
C_{*_2}	α	[-5, 90]	30	6	(1, 1, 1)	(7, 6, 6)	144	56,448
	β	[-30, 30]	15	6				
	Mach	[0.3, 2.16]	8	4				
C_{*_3}	α	[-2.5, 45]	20	10	(1, 1, 1)	(3, 3, 3)	10	640
	β	[-10, 10]	3	1				
	δ_{LIBLEF}	[0, 40]	2	1				
C_{*_4}	α	[-2.5, 45]	20	5	(1, 1, 1, 1)	(5, 5, 5, 5)	10	45,360
	β	[-10, 10]	3	1				
	δ_{LIBLEF}	[0, 40]	2	1				
	δ_{LOBLEF}	[-40, 40]	5	1				
	Mach	[0.3, 1.2]	4	1				
C_{*_5}	α	[-2.5, 90]	28	10	(1, 1, 1, 1)	(5, 4, 4, 5)	160	144,000
	δ_{LSSD}	[0, 60]	4	2				
	δ_{LEL}	[-30, 30]	5	4				
	Mach	[0.3, 2.16]	7	2				
C_{*_6}	α	[-2.5, 90]	29	10	(2, 1, 1, 1)	(3, 1, 1, 3)	80	12,800
	δ_{LSSD}	[0, 60]	2	1				
	δ_{RSSD}	[0, 60]	2	1				
	δ_{PF}	[-30, 30]	5	2				
	Mach	[0.3, 2.16]	7	2				
C_{*_7}	α	[-2.5, 90]	29	10	(1, 1, 1)	(5, 5, 4)	40	7,200
	β	[-30, 30]	7	2				
	δ_{LAMT}	[0, 60]	5	2				
C_{*_8}	α	[-2.5, 42.5]	19	9	(1, 1, 1)	(4, 3, 3)	9	720
	δ_{LEL}	[-30, 30]	3	1				
	δ_{LAMT}	[0, 60]	3	1				
C_{*_9}	α	[-2.5, 42.5]	19	9	(1, 1, 1)	(4, 3, 3)	9	720
	δ_{LOBLEF}	[0, 40]	3	1				
	δ_{LAMT}	[0, 60]	3	1				
$C_{*_{10}}$	α	[-2.5, 42.5]	19	9	(1, 1, 1)	(4, 3, 3)	9	720
	δ_{REL}	[-30, 30]	3	1				
	δ_{RAMT}	[0, 60]	3	1				

$C_{*_{11}}$	α	[-2.5, 42.5]	19	9	(1, 1, 1)	(4, 3, 3)	9	720
	δ_{ROBLEF}	[0, 40]	3	1				
	δ_{RAMT}	[0, 60]	3	1				
$C_{*_{12}}$	α	[-2.5, 90]	29	15	(1, 1, 1)	(4, 3, 3)	90	7,200
	β	[-30, 30]	7	3				
	δ_{LSSD}	[0, 60]	4	2				
$C_{*_{13}}$	α	[-2.5, 45]	20	10	(1, 1, 1)	(3, 3, 3)	10	640
	β	[-10, 10]	3	1				
	δ_{RIBLEF}	[0, 40]	2	1				
$C_{*_{14}}$	α	[-2.5, 45]	20	5	(1, 1, 1, 1)	(5, 5, 5, 5)	10	45,360
	β	[-10, 10]	3	1				
	δ_{RIBLEF}	[0, 40]	2	1				
	δ_{ROBLEF}	[-40, 40]	5	1				
	Mach	[0.3, 1.2]	4	1				
$C_{*_{15}}$	α	[-2.5, 90]	28	10	(1, 1, 1, 1)	(5, 4, 4, 5)	160	144,000
	δ_{RSSD}	[0, 60]	4	4				
	δ_{REL}	[-30, 30]	5	2				
	Mach	[0.3, 2.16]	7	2				
$C_{*_{16}}$	α	[-2.5, 90]	29	10	(1, 1, 1)	(4, 4, 3)	40	4,000
	β	[-30, 30]	7	2				
	δ_{RAMT}	[0, 60]	5	2				
$C_{*_{17}}$	α	[-2.5, 90]	29	15	(1, 1, 1)	(4, 4, 3)	90	9,000
	β	[-30, 30]	7	3				
	δ_{RSSD}	[0, 60]	4	2				
$C_{*_{18,19}}$	α	[0, 30]	13	3	(1, 1)	(5, 5)	12	432
	Mach	[0.6, 2.2]	6	4				

References

[1] Tol, H. J., De Visser, C. C., van Kampen, E., and Chu, Q. P., "Nonlinear Multivariate Spline-Based Control Allocation for High-Performance Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1840–1862. doi:10.2514/1.G000065, URL http://arc.aiaa.org/doi/10.2514/1.G000065.

[2] Tol, H. J., De Visser, C. C., Sun, L. G., van Kampen, E., and Chu, Q. P., "Multivariate Spline-Based Adaptive Control of High-Performance Aircraft with Aerodynamic Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2016, pp. 781–800. doi:10.2514/1.G001079, URL http://arc.aiaa.org/doi/10.2514/1.G001079.

[3] Matamoros, I., and De Visser, C. C., "Incremental Nonlinear Control Allocation for a Tailless Aircraft with Innovative Control Effectors," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018, p. 220. doi:doi:10.2514/6.2018-1116, URL https://doi.org/10.2514/6.2018-1116.

[4] Klein, V., and Morelli, E. A., *Aircraft System Identification: Theory and Practice*, American Institute of Aeronautics and Astronautics, Inc., 2006.

[5] Hong, X., Mitchell, R. J., Chen, S., Harris, C. J., Li, K., and Irwin, G. W., "Model selection approaches for non-linear system identification: A review," *International Journal of Systems Science*, Vol. 39, No. 10, 2008, pp. 925–946. doi: 10.1080/00207720802083018.

[6] Farin, G., "A History of Curves and Surfaces in CAGD," *Handbook of computer aided geometric design*, Elsevier, 1984, Chap. 1, pp. 1–23.

[7] Anderson, I. J., Cox, M. G., and Mason, J. C., "Tensor-product spline interpolation to data on or near a family of lines," *Numerical Algorithms*, Vol. 5, No. 4, 1993, pp. 193–204. doi:10.1007/BF02210502, URL https://doi.org/10.1007/BF02210502http://link.springer.com/10.1007/BF02210502.

[8] De Visser, C. C., "Global Nonlinear Model Identification with Multivariate Splines," Ph.D. thesis, Delft University of Technology, 2011.

[9] De Visser, C. C., Chu, Q. P., and Mulder, J. A., "A new approach to linear regression with multivariate splines," *Automatica*, Vol. 45, No. 12, 2009, pp. 2903–2909. doi:10.1016/j.automatica.2009.09.017.

[10] Lai, M.-J., and Schumaker, L. L., *Spline Functions on Triangulations*, 1st ed., Cambridge University Press, Cambridge, 2007. URL http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780511889479.

[11] Govindarajan, N., De Visser, C. C., and Krishnakumar, K., "A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines," *Automatica*, Vol. 50, No. 9, 2014, pp. 2234–2244. doi:10.1016/j.automatica.2014.07.012, URL http://dx.doi.org/10.1016/j.automatica.2014.07.012.

[12] De Visser, C. C., Brunner, E., and Verhaegen, M., "On distributed wavefront reconstruction for large-scale adaptive optics systems," *Journal of the Optical Society of America A*, Vol. 33, No. 5, 2016, pp. 817–831. doi:10.1364/JOSAA.33.000817.

[13] Loera, J. A. D., Rambau, J., and Santos, F., *Triangulations; structures for algorithms and applications*, No. 25 in Algorithms and computation in mathematics, Springer-Verlag, Berlin Heidelberg, 2010.

[14] Freund, R. M., "Combinatorial theorems on the simplotope that generalize results on the simplex and cube," *Mathematics of Operations Research*, Vol. 11, No. 1, 1986, pp. 169–179.

[15] Talman, A. J. J., "Intersection theorems on the unit simplex and the simplotope," *Imperfections and Behavior in Economic Organizations*, Theory and Decision Library, Vol. 11, edited by R. P. Gilles and P. H. M. Ruys, Springer Netherlands, 1991, Chap. 11, pp. 257–278.

[16] Lai, M. J., "On Construction of Bivariate and Trivariate Vertex Splines on Arbitrary Mixed Grid Partitions," Ph.D. thesis, Hangzhou University, Hangzhou, China, August 1989.

[17] Visser, T., De Visser, C. C., and Van Kampen, E.-J., "Quadrotor System Identification Using the Multivariate Multiplex B-Spline," *AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Kissimmee, Florida, 2015, pp. 1–13. doi:doi:10.2514/6.2015-0747, URL https://doi.org/10.2514/6.2015-0747.

[18] Sun, L. G., De Visser, C. C., and Chu, Q. P., "A New Substitution Based Recursive B-Splines Method for Aerodynamic Model Identification," *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control*, edited by Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 233–245.

[19] Awanou, G., Lai, M.-J., and Wenston, P., "The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differential Equations," *Wavelets and Splines*, Nashboro Press, Brentwood, Tennessee, 2005, pp. 24–75.

[20] Silva, J., Keviczky, T., and Verhaegen, M., "A Distributed Approach for Solving Wavefront Reconstruction Problems," *American Control Conference (ACC)*, Boston, 2016, pp. 6513–6518. URL ieeexplore.ieee.org/document/7526695/.

[21] van der Peijl, I. V., "Physical Splines for Aerodynamic Modelling of Innovative Control Effectors," Master's thesis, Delft University of Technology, 2017.

[22] Niestroy, M. A., Dorsett, K. M., and Markstein, K., "A Tailless Fighter Aircraft Model for Control-Related Research and Development," *AIAA Modeling and Simulation Technologies Conference*, 2017, pp. 1–18. doi:10.2514/6.2017-1757, URL http://arc.aiaa.org/doi/10.2514/6.2017-1757.

[23] Dorsett, K. M., and Mehl, D. R., "Innovative Control Effectors (ICE) Phase I," Tech. rep., Lockheed Martin Tactical Aircraft Systems, 1996. doi:WL-TR-96-3043, URL http://oai.dtic.mil/oai/oai?verb=getRecord{&}metadataPrefix=html{&}identifier=ADB041508.

[24] Dorsett, K. M., Fears, S. P., and Houlden, H. P., "Innovative Control Effectors (ICE) Phase II," Tech. rep., Lockheed Martin Tactical Aircraft Systems and Bihrle Applied Research, Inc., 1997. doi:WL-TR-96-3043.

[25] Lai, M.-J., and Schumaker, L. L., *Spline functions on triangulations*, 1st ed., Cambridge University Press, 2007. URL http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780511889479.

[26] Visser, T., De Visser, C. C., and van Kampen, E.-J., "Towards the multivariate simplotope spline : continuity conditions in a class of mixed simplotopic grids," , 2016.

[27] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, Vol. 3, No. 1, 2010, pp. 1–122. doi:10.1561/2200000016, URL http://www.nowpublishers.com/article/Details/MAL-016.

[28] Ghadimi, E., Teixeira, A., Shames, I., and Johansson, M., "Optimal Parameter Selection for the Alternating Direction Method of Multipliers (ADMM): Quadratic Problems," *IEEE Transactions on Automatic Control*, Vol. 60, No. 3, 2015, pp. 644–658. doi:10.1109/TAC.2014.2354892, URL http://arxiv.org/abs/1306.2454{%}0Ahttp://dx.doi.org/10.1109/TAC.2014.2354892.

[29] Wohlberg, B., "ADMM Penalty Parameter Selection by Residual Balancing," , 2017. URL http://arxiv.org/abs/1704.06209.

[30] He, B., Hou, L., and Yuan, X., "On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming," *SIAM Journal on Optimization*, Vol. 25, No. 4, 2015, pp. 2274–2312. doi:10.1137/130922793, URL http://epubs.siam.org/doi/abs/10.1137/130922793.

[31] Chen, C., He, B., Ye, Y., and Yuan, X., "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, Vol. 155, No. 1-2, 2016, pp. 57–79. doi:10.1007/s10107-014-0826-5, URL http://dx.doi.org/10.1007/s10107-014-0826-5.

[32] Wang, J. J., and Song, W., "An algorithm twisted from generalized ADMM for multi-block separable convex minimization models," *Journal of Computational and Applied Mathematics*, Vol. 309, 2017, pp. 342–358. doi:10.1016/j.cam.2016.02.001, URL http://dx.doi.org/10.1016/j.cam.2016.02.001.

[33] Deng, W., Lai, M. J., Peng, Z., and Yin, W., "Parallel Multi-Block ADMM with o(1 / k) Convergence," *Journal of Scientific Computing*, Vol. 71, No. 2, 2017, pp. 712–736. doi:10.1007/s10915-016-0318-2.

[34] Bertsekas, D. P., and Tsitsiklis, J. N. J., *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, Massachusetts, 1989. URL http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:parallel+and+distributed+computation:+numerical+methods{#}2.

[35] Goldstein, T., O'Donoghue, B., Setzer, S., and Baraniuk, R., "Fast Alternating Direction Optimization Methods," *SIAM Journal on Imaging Sciences*, Vol. 7, No. 3, 2014, pp. 1588–1623. doi:10.1137/120896219, URL http://epubs.siam.org/doi/10.1137/120896219.

[36] He, B. S., Yang, H., and Wang, S. L., "Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities," *Journal of Optimization Theory and Applications*, Vol. 106, No. 2, 2000, pp. 337–356. doi: 10.1023/A:1004603514434, URL http://link.springer.com/10.1023/A:1004603514434.

[37] Wang, S. L., and Liao, L. Z., "Decomposition method with a variable parameter for a class of monotone variational inequality problems," *Journal of Optimization Theory and Applications*, Vol. 109, No. 2, 2001, pp. 415–429. doi:10.1023/A:1017522623963.

[38] Bey, J., "Simplicial grid refinement: on Freudenthal's algorithm and the optimal number of congruence classes," *Numerische Mathematik*, Vol. 85, No. 1, 2000, pp. 1–29. doi:0.1007/s002110050475.