Quadrotor System Identification using the Multivariate Multiplex B-Spline

T. Visser, * C. C. de Visser † and E. van Kampen ‡

Delft University of Technology, P.O. Box 5058, 2600GB Delft, The Netherlands

A novel method for aircraft system identification is presented that is based on a new multivariate spline type; the multivariate multiplex B-spline. The multivariate multiplex B-spline is a generalization of the recently introduced tensor-simplex B-spline. Multivariate multiplex splines obtain similar or better approximation accuracy using less parameters (B-coefficients) than standard multivariate simplex B-splines which are currently used for aircraft system identification. The multiplex spline allows the user to incorporate a-priori knowledge of the modelled system in the definition of the model structure. In particular, while the standard simplex B-splines use a multi-dimensional triangulation in which all dimensions are coupled, the multiplex spline enables the user to decouple specific model dimensions based on expert knowledge of the system. The new method is used to approximate a 4-dimensional nonlinear quadrotor inflow dataset. The results show that the multiplex B-spline obtains a relative root mean square error of 0.672% using 1440 B-coefficients. This compares favorably to results obtained with the standard 4-dimensional simplex B-spline on the same dataset, which resulted in an relative root mean square error of 1.608% using 1540 B-coefficients.

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Matrix form of basis polynomials of a spline</td>
</tr>
<tr>
<td>$B^d$</td>
<td>Vector of basis polynomials of a simplex spline</td>
</tr>
<tr>
<td>$B^\kappa$</td>
<td>Vector of barycentric coordinates in a simplex</td>
</tr>
<tr>
<td>$B^\lambda$</td>
<td>Vector of basis polynomials of a multiplex spline</td>
</tr>
<tr>
<td>c</td>
<td>Vector of B-coefficients</td>
</tr>
<tr>
<td>d</td>
<td>Polynomial degree</td>
</tr>
<tr>
<td>H</td>
<td>Smoothness matrix</td>
</tr>
<tr>
<td>$K_{emf}$</td>
<td>Constant related to back EMF in engine</td>
</tr>
<tr>
<td>ℓ</td>
<td>Number of layers in a multiplex spline</td>
</tr>
<tr>
<td>$M$</td>
<td>Tessellation consisting of multiplices</td>
</tr>
<tr>
<td>m</td>
<td>Continuity order or dimension smaller than $n$</td>
</tr>
<tr>
<td>N</td>
<td>Number of simplices or multiplices in a tessellation</td>
</tr>
<tr>
<td>n</td>
<td>Dimension, number of variables</td>
</tr>
<tr>
<td>P</td>
<td>Engine power, W</td>
</tr>
<tr>
<td>$p$</td>
<td>Simplex polynomial</td>
</tr>
<tr>
<td>$R_a$</td>
<td>Motor armament resistance, Ω</td>
</tr>
<tr>
<td>r</td>
<td>Total continuity order</td>
</tr>
<tr>
<td>T</td>
<td>Thrust, N</td>
</tr>
<tr>
<td>$T$</td>
<td>Triangulation consisting of simplices</td>
</tr>
<tr>
<td>U</td>
<td>Input voltage, V</td>
</tr>
<tr>
<td>v</td>
<td>Vertex of a simplex</td>
</tr>
<tr>
<td>v</td>
<td>Velocity, m/s</td>
</tr>
</tbody>
</table>

---

* Master Student, Control and Simulation Department, t.visser-1@student.tudelft.nl.
† Assistant Professor, Control and Simulation Department, c.c.devisser@tudelft.nl, AIAA Member.
‡ Assistant Professor, Control and Simulation Department, e.vankampen@tudelft.nl, AIAA Member.
I. Introduction

Due to the increased popularity of model-based control techniques such as nonlinear dynamic inversion, the need for accurate non-linear models has grown. In many industrial applications data tables are used, whereas the research community has embraced neural networks as a function approximator. Recently the multivariate simplex B-spline was presented as an alternative. The main benefit of this technique over neural networks is the fact that it is linear in the parameters. This allows for the use of efficient least squares solvers.

The major downside of the simplex spline is its triangulation. Although it provides the user with a lot of flexibility in spanning the domain of interest, it is unclear how this flexibility can be exploited. Most triangulation optimization algorithms are developed for two-dimensional domains. In higher-dimensional domains these algorithms cannot be used. Current practice is to split up the domain in rectangular shapes first. These boxes are then split up using a standard triangulation. Another option that is currently being investigated is the simultaneous optimization of the triangulation and the parameters. Although a global optimum can be found with both the parameters and the triangulation as arguments, its high computational complexity currently forbids the use of this technique on large triangulations.

On top of the excess flexibility in defining a triangulation, the shape of the high-dimensional simplex also gives reason for concern. As the dimension increases, the content of the simplex moves towards the vertices, away from the center. In practical terms, it becomes increasingly hard to fill the entire simplex with data.

In this paper a new spline is proposed that can immediately be defined on orthotopes (or boxes), without triangulating them. This has significant benefits over standard simplex B-spline methods in terms of the distribution of data over a domain element. The flexibility in defining the triangulation is reduced, but the technique employed by De Visser can still be used, with the difference that it is no longer required to triangulate the orthotopes.

The definition of the orthotope spline is based in the tensor-product simplex spline, which was recently introduced by Govindarajan et al.to solve the Hamilton-Jacobi-Bellman equation, central in reinforcement learning and optimal control. In this framework the B-coefficients of a multivariate simplex spline are defined using a univariate spline function of a different variable. This framework was used by Govindarajan et al.to estimate autopilot safety margins, and by Sun et al.in an aircraft system identification framework.

The multivariate multiplex spline used in the current research is a direct generalization of the tensor-product simplex spline. It allows for the use of multiple multivariate layers. Each layer represents a multivariate simplex spline that describes the coefficients in the previous layer. Fitting data on non-rectangular
domains is facilitated by studying the combination of a single simplex from each layer: a multiplex.

The goal of this paper is to present and test an algorithm for aircraft system identification based on this multivariate multiplex spline. As a test case, the thrust and torque of a quadrotor MAV are modeled, taking engine dynamics and nonuniform inflow into account. The advantage of the multiplex spline over the standard simplex B-spline is demonstrated; a similar or better approximation accuracy is obtained with a lower total number of parameters (B-coefficients) than the standard simplex B-splines.

Because this is the first introduction of the multiplex spline in an engineering application, a short introduction is presented here as section II. A more extensive discussion, as well as many of the proofs of presented theory, can be found in Ref. 9. In section III the main contribution of this article is presented. An algorithm is proposed for the use of the multiplex spline in system identification. The application of the algorithm to a quadrotor modeling task is described in section IV. Finally in section V the performance of the algorithm is discussed and conclusions are drawn regarding its applicability.

II. The multiplex spline framework

The origin of the multiplex spline lies in the tensor-product simplex spline of Govindarajan.\(^6\) That is, B-coefficients of one spline are represented by a spline in other variables. The multiplex spline results when this process is repeated and B-coefficients are reordered. The process and the effect on the defining elements of a simplex spline are discussed in this section.

To properly introduce the multiplex spline, first a short explanation of the simplex spline framework is presented. A more elaborate discussion of the bivariate case is given by Lai and Schumaker.\(^10\)

A. The multivariate simplex spline

A multivariate simplex spline is a piecewise polynomial defined on a geometric basis of simplices. Continuity of arbitrary order can be defined between simplices if they share an edge. The polynomials on individual simplices are weighted sums of Bernstein basis polynomials defined in barycentric coordinates. The most important facets of such a spline are discussed in this subsection.

1. Simplices, barycentric coordinates and triangulations

An \(n\)-simplex is defined as the convex hull of \(n+1\) non-degenerate vertices \(\mathcal{V} = \{v_0, ..., v_n\}, v_i \in \mathbb{R}^n\). Any point in \(\mathbb{R}^n\) can be uniquely defined using a vector of \(n+1\) barycentric coordinates \(b := (b_0, ..., b_n) \in \mathbb{R}^{n+1}\) relative to this set of vertices, such that \(\sum_{i=0}^{n} b_i = 1\). For a general point \(x \in \mathbb{R}^n\) we have

\[
x = \sum_{i=0}^{n} b_i v_i
\]

If \(b_i \geq 0, \forall i \in [0,n]\), then \(x\) lies in the simplex described by \(\mathcal{V}\).

Multiple \(n\)-simplices \(\Delta\) can be combined to form a triangulation \(\mathcal{T}\) of a domain. We define a triangulation as the collective of simplices, such that

\[
\mathcal{T} = \bigcup_{i=1}^{N} \Delta_i, \quad \Delta_i \cap \Delta_j \in \{\emptyset, \bar{\Delta}\}, \forall i \neq j
\]

with \(N\) the total number of simplices and \(\bar{\Delta}\) an \(m\)-simplex, \(0 \leq m < n\). That is, in a triangulation simplices do not overlap but may share complete edges of dimension lower than \(n\).

2. B-form polynomial

Polynomials are defined on a simplex by taking the weighted sum of Bernstein basis polynomials, defined in barycentric coordinates. The B-form of such a polynomial is\(^11\)

\[
p(b) = \sum_{|\kappa| = d} c_{\kappa} B_{\kappa}^d(b), \quad B_{\kappa}^d(b) := \frac{d!}{\kappa!} b^\kappa
\]

3 of 13

American Institute of Aeronautics and Astronautics
where \( c_\kappa \) are B-coefficients, discussed in the next subsubsection, and \( \kappa = (\kappa_0, \ldots, \kappa_n) \in \mathbb{N}^n \) is a multi-index with \( |\kappa| := \sum \kappa_i \) its one-norm. Also, \( \kappa! := \prod \kappa_i \) and \( b^\kappa := \prod b_i^{\kappa_i} \). Every possible permutation of \( \kappa \) results in a separate Bernstein basis polynomial. The complete set forms a basis for the space of all polynomials of degree \( d \) in \( n \) variables.

To define a complete set of basis polynomials on each simplex in a triangulation, the vector form of the basis polynomials is defined as

\[
B^d_\kappa(b(x)) := \begin{cases} 
B^d_\kappa(b^\Delta_i(x)) & \text{if } x \in \Delta_i \\
0 & \text{otherwise}
\end{cases}
\]

where \( b^\Delta_i(x) \) signifies that \( x \) is described in barycentric coordinates with respect to the vertices of simplex \( \Delta_i \). The vector form is thus a collection of the basis polynomials in one simplex, padded with zeros for all other simplices.

3. \textit{B-coefficients and the B-net}

One important feature of the simplex spline is that the B-coefficients \( c_\kappa \), introduced in Eq. (3) as the weights of the basis polynomials, have a spatial location in the simplex. They coincide with control points, such that

\[
x_\kappa = \frac{1}{d} \sum_{i=0}^n \kappa_i v_i 
\]

The collective of these locations is called the B-net. It plays a central role in the definition of continuity conditions. An example B-net is shown in figure 1.

Much like the basis polynomials, the B-coefficients are collected in a vector. We write \( c^\Delta_i = (c^\Delta_i)_|\kappa|=d \), where the coefficients are ordered according to the multi-index in lexicographical order, and \( c = (c^\Delta_1, \ldots, c^\Delta_N) \). The superscript \( \Delta_i \) indicates that the coefficients lie in that simplex.

4. \textit{Continuity conditions}

Two polynomials \( p \) and \( \tilde{p} \) defined on \( n \)-simplices \( \Delta \) and \( \tilde{\Delta} \) that share an \( n-1 \)-edge can be joined with arbitrary continuity order \( r < d \). This is done by imposing the following constraints\(^{1,10}\)

\[
\tilde{c}_{m,\kappa_1,\ldots,\kappa_n} = \sum_{|\gamma| = m} c_{0,\kappa_1,\ldots,\kappa_n + \gamma} B^m_\gamma(b^\Delta(v_0)) \quad \forall m \in [0, r] 
\]

where \( \gamma \) is again a multi-index of the same size as \( \kappa \) and the sum between \( \kappa \) and \( \gamma \) is element-wise. It is assumed that \( v_0 \) and \( \tilde{v}_0 \) are the out-of-edge vertices. An example of a first order (that is \( m = 1 \)) continuity condition is given for the bivariate case in figure 1.

Because the conditions in Eq. (6) are linear in \( c_\kappa \), they can be collected in a smoothness matrix \( H \), such that we obtain\(^{12}\)

\[
Hc = 0 
\]

\( H \) is a very sparse matrix, because each row describes a single constraint between two simplices. That is, in each row a maximum of \( 1 + \frac{(d+n-1)\!}{(d-1)\!n!} \) elements are nonzero.
5. Fitting scattered data

The above can be combined into a single equality constrained least squares problem.

\[ \text{min } ||y - B(b(X))c|| \]

subject to: \( Hc = 0 \) \hspace{1cm} (8)

In the above, \( || \cdot || \) is the 2-norm, \( X \) is a matrix containing the input data, and \( B \) is the matrix of basis polynomials. The output vector is given as \( y \).

The solution of Eq. (8) can be found using Lagrangian multipliers. This is discussed in detail by De Visser et al.\textsuperscript{12} Because the basis polynomial matrix \( B \) is block diagonal and \( H \) is sparse, distributed solvers can be employed.

B. The multivariate multiplex spline

The multiplex spline is a generalization of the tensor-product simplex spline presented by Govindarajan et al.\textsuperscript{6} In the latter framework the \( B \)-coefficients \( c_\kappa \) are represented by a univariate simplex spline. In the current discussion \( c_\kappa \) is instead described using a multivariate spline. On top of that, the coefficients in this second layer can again be described using a spline. The result is a spline defined on nested triangulations. In the multiplex spline framework one simplex from each triangulation is considered at a time. The combination of these simplices is called a multiplex.

1. Multiplices, barycentric coordinates and tessellations

A \( \nu \)-multiplex \( \Gamma \), with \( \nu \) a multi-index with \( \ell \) entries and \( |\nu| = n \), is an \( n \)-dimensional polytope that is constructed by, starting from the \( \nu_1 \)-simplex, iteratively combining the \( (\nu_1, ..., \nu_j) \)-multiplex with a \( \nu_{j+1} \)-simplex. Combining layers in this case means to append the \( \nu_{j+1} \)-simplex to every point in the \( (\nu_1, ..., \nu_j) \)-multiplex (see figure 2). The \( \nu_{j+1} \)-simplex is called the \( (j+1)\)th layer of the multiplex. In order for the multiplex to be \( n \)-dimensional, each \( \nu_i \)-dimensional layer should introduce \( \nu_i \) new variables. In our current discussion this means that the layers are mutually perpendicular. Note that the order in which layers are combined does not affect the end result, as illustrated in figure 2.

Any point in the multiplex can be uniquely described using barycentric coordinates \( b_i \) in each layer. By collecting the vectors \( b_i \) we obtain \( \beta = (b_1, ..., b_\ell) \in \mathbb{R}^{n+\ell} \), where \( b_i = (b_{i0}, ..., b_{i\nu_i}) \in \mathbb{R}^{\nu_i+1} \) with constraints \( \sum_j b_{ij} = 1, \forall i \in [1, \ell] \). Instead of solving Eq. (1) for \( b_i \) in each layer, a direct solution can be found by considering the vertices of layers. For this we select a copy of each layer such that they share a vertex \( w_0 \). That is, we select vertex sets \( W_i = \{ w_0, w_{i1}, ..., w_{i\nu_i} \} \), as indicated in figure 2 for \( i \in \{1, 2\} \). Then we have for any point \( x \in \mathbb{R}^n \) in the multiplex
\[ \mathbf{x} = (1 - \ell) \mathbf{w}_0 + \sum_{i=1}^{\ell} \sum_{j=0}^{\nu_i} \beta_{ij} \mathbf{w}_{ij} \]  

(9)

where \( \mathbf{w}_{i0} := \mathbf{w}_0, \forall i \in [1, \ell] \). Equation (9) is a direct result of adding the origin location \( \mathbf{w}_0 \) to the locations of the projections of \( \mathbf{x} \) in the layers with respect to the same origin point \( \mathbf{w}_0 \).

A tessellation of multiplices is easily constructed from triangulations in each layer of the spline. For example, a tessellation of (2,1)-multiplices can be constructed by making a triangulation of 2-simplices and one of 1-simplices, then constructing the multiplices from any combination of simplices from both triangulations. The resulting tessellation \( \mathcal{M} \) consisting of multiplices \( \Gamma_i \) is defined as

\[ \mathcal{M} = \bigcup_{i=1}^{N} \Gamma_i, \quad \Gamma_i \cap \Gamma_j \in \{\emptyset, \bar{\Gamma}\}, \forall i \neq j \]  

(10)

where \( N \) is the number of multiplices. The multiplex \( \bar{\Gamma} \) is a \((\nu - \mu)\)-multiplex, where \( \mu \) is a multi-index like \( \nu \) with \( 0 \leq |\mu| < n \) and \( \nu - \mu > 0 \).

2. B-form polynomial

If \( c_\kappa \) in Eq. (3) is replaced by a polynomial in the B-form, a tensor-product between the basis polynomials results. The multi-indices \( \kappa \) from the layers are renamed and combined to form one multi-index \( \lambda = (\lambda_1, \ldots, \lambda_\ell) \), where \( \lambda_i = (\lambda_{i0}, \ldots, \lambda_{i\nu_i}) \) the multi-index in the \( i \)th layer with \( |\lambda_i| = d_i \). The degree is defined per layer, leading to a degree vector \( \mathbf{d} = (d_1, \ldots, d_\ell) \).

Applying these changes to Eq. (3) we obtain

\[ \pi(\beta) = \sum_{|\lambda_i|=d_i, \forall i \in [1,\ell]} c_\lambda B^d_{\lambda}(\beta), \quad B^d_{\lambda}(\beta) := \prod_{i=1}^{\ell} B^d_{\lambda_i}(b_i) = \prod_{i=1}^{\ell} \frac{d_i!}{\lambda_i!} \beta^{\lambda_i} \]  

(11)

Again, the barycentric coordinate vector \( \beta = (b_1, \ldots, b_\ell) \) is a collection of barycentric coordinates in the layers. The resulting basis polynomials have a degree of \( \prod_i d_i \), but not all basis polynomials of this degree are present.

In the vector form this procedure results in a tensor-product per multiplex. The vector form of \( B^d_{\lambda}(\beta) \) can be obtained by taking the tensor-product of the subsets \( B^d_{\lambda_i} \) and then reordering, but obtaining them directly is preferable.

\[ B^d_{\lambda}(\beta(x)) := \begin{cases} B^d_{\lambda}(\beta^{\Gamma_i}(x)) & \text{if } x \in \Gamma_i \\ 0 & \text{otherwise} \end{cases} \]  

(12)

Note the similarity with Eq. (4).

3. B-coefficients and the B-net

The B-coefficients \( c_\lambda \) are generated by collecting the multi-indices in the layers. Their location is again determined by the control points \( \mathbf{x}_\lambda \in \mathbb{R}^n \)

\[ \mathbf{x}_\lambda = (1 - \ell) \mathbf{w}_0 + \sum_{i=1}^{\ell} \frac{1}{d_i} \sum_{j=0}^{\nu_i} \lambda_{ij} \mathbf{w}_{ij} \]  

(13)

Note the similarity between Eq. (9) and Eq. (13). The B-net of the (2,1)-multiplex is given in figure 3.

The coefficients can again be collected in a vector \( \mathbf{c} = (c_{\Gamma_1}, \ldots, c_{\Gamma_N}) \) with \( c_{\Gamma_i} = (c_{\lambda})_{|\lambda_i|=d_i, \forall i \in [1,\ell]} \). The coefficients are in lexicographical order based on the multi-index \( \lambda \).
Figure 4: The shared edge and related continuity conditions between (2,1)-multiplices $\Gamma = \{w_{ij}\}$ and $\tilde{\Gamma} = \{\tilde{w}_{ij}\}$, $i \in \{1, 2\}$. Since the out-of-edge vertex lies in the two-dimensional layer, the shared edge is a (1,1)-multiplex (shaded). Much like the B-net (in figure 3), the continuity conditions are formed by making parallel copies of 2-simplex continuity for each B-coefficient of the second layer. Note that for clarity much of the B-net is left out and each example first order continuity condition is given a different color.

4. Continuity conditions

Continuity between two polynomials $\pi$ and $\tilde{\pi}$ on $\nu$-multiplices $\Gamma$ and $\tilde{\Gamma}$ is defined per layer. The conditions in Eq. (6) are applied to the triangulations in each layer first. Then they are copied for each parallel copy of the B-net. In the end the conditions for $r^{th}$ order continuity can be written as

$$
\tilde{c}(m, \lambda_1, ..., \lambda_n) = \sum_{|\varepsilon| = m} c(0, \lambda_1, ..., \lambda_n) + \varepsilon B^m(\beta^{\tilde{\Gamma}}(\tilde{w}_{10})), \quad \forall m \in [0, r]
$$

where it is assumed that $\tilde{w}_{10}$ is the out-of-edge vertex. That is, the out-of-edge vertex lies in the first layer. Note that $\varepsilon = (\varepsilon_1, ..., \varepsilon_\ell)$ contains many zeros by default, because any adjustment to $\lambda$ that violates $|\lambda_i + \varepsilon_i| = d_i$ results in a zero B-coefficient. In other words, the degree cannot be changed in other layers than the one containing the out-of-edge vertex. In the case that the out-of-edge vertex lies in the first layer, we therefore have $\varepsilon_i = (0, ..., 0), \forall i \in [2, \ell]$.

The conditions are linear in the B-coefficients, and can therefore again be written as

$$
Hc = 0
$$

The matrix $H$ contains the same conditions as the one in Eq. (6). Each condition (or row) is duplicated multiple times, such that the conditions hold in each parallel copy of the B-net in a layer. This is illustrated in figure 4b.

5. Fitting scattered data

If Eq. (12) and (15) are compared to Eq. (4) and (7) respectively, the conclusion can be drawn that the multiplex spline fitting problem is the same as that of the simplex spline. That means that Eq. (8) can be reused if $b$, $B$, and $c$ are replaced with $\beta$, $B$, and $c$.

C. Comparison

The simplex and multiplex spline can be compared on the basis of their defining elements. First the geometric implications of using a multiplex are discussed, followed by the polynomial consequences of choosing layers.

1. The $n$-simplex versus the $\nu$-multiplex

A major downside of the use of simplices is the distribution of the content (or high-dimensional volume). As the dimension $n$ increases, the content of the $n$-simplex moves towards the vertices, yielding a very thin polytope. This can be illustrated by plotting the ratio of the content of the $n$-simplex and its inscribed $n$-sphere. The same can be done for the $n$-cube, which is the $(1, ..., 1)$-multiplex. Both are plotted in figure...
5. As the ratio decreases, content is moving towards the vertices, meaning the polytope becomes more stretched. In practical terms, it becomes increasingly hard to properly fill each polytope with data. The simplex is the polytope that suffers most from this effect. Using multiplices will thus always improve the content distribution.

A second advantage of the multiplex spline is that a high-dimensional tessellation can be defined by multiple low-dimensional triangulations. If groups can be made of less than four variables, all triangulations can be plotted for visual inspection and optimal triangulation algorithms can be employed.

2. The use of layers

From the polynomial point of view, the multiplex spline presents the opportunity to choose the form of coupling between variables. When two variables lie in the same layer, they become part of a complete multivariate polynomial, resulting in nonlinear ($d^{th}$ degree) cross-coupling between the variables. Separating variables results in a multilinear, tensor product coupling. In a simplex spline all variables will always be coupled.

In many cases different variables require different polynomial degrees for appropriate modeling. In the multiplex spline it is possible to make a subdivision of the variables in layers based on this required degree. This may prevent overfitting or divergence near the edges of the domain of variables for which few data are available.

### III. System identification algorithm

The great promise of the multiplex spline lies in the fact that variables can be decoupled within a single spline model. In practice this means that we have a lot more freedom to incorporate our knowledge of the system in the model structure. In this section an approach to choosing the structure is presented as an algorithm.

In short, the algorithm contains the following steps.

1. Group variables and project the data set;
2. Triangulate each projection;
3. Combine triangulations into a tessellation;
4. Fit a multiplex spline;
5. Iterate on step 1 or 2.
The complete process is illustrated in figure 6. The individual steps are discussed in more detail in dedicated subsections.

A. Group variables

Groups of variables define layers in the multiplex spline. If an analytic model is available, the coupling between variables should form the basis for this grouping. Variables that require similar degrees and are strongly coupled should together form a layer. If no analytic model is available, the shape of the data set can play a leading role. The goal would then be to make sure that the projections can be triangulated efficiently in the next step.

B. Design triangulations

A triangulation is made in each layer based on the shape of the projection of the data set and the desired degree. With the dimension of the layers the complexity of making triangulations can be scaled. One could even decide to use only one- and two-dimensional layers to allow for the use of simple triangulation optimization algorithms.

C. Generate a tessellation

Combining the triangulations in each layer into a single tessellation is mostly a formal procedure. In many applications it is preferable to consider separate triangulations, e.g. converting input data to barycentric coordinates. If a multiplex contains less data points than B-coefficients, it should be removed from the tessellation.

D. Fit a spline model

The knowledge of the system is used to choose an appropriate degree and continuity order. This is combined with the tessellation to form a spline. The B-coefficients can be found using the method described in Ref. 12.

E. Iterate

Based on the results obtained in the previous step it is decided if and how the model can be improved. If locally there is a lack of approximation power, simplices can be added in the layers that contain the relevant variables. If there are signs of unmodeled dynamics one can choose to either raise the degree in certain layers or move variables from one layer to the other.

IV. Quadrotor modeling

To proof the applicability of the multiplex spline, it was used to model the thrust of a quadrotor as a function of free-stream velocity, angle of attack, rotor rotational rate, and input voltage to the engine. The analytic model presented in subsection A was used to generate a data set.
A. Analytic model

The analytic model used to generate a data set is mostly based on helicopter theory that has been applied to quadrotors before.\textsuperscript{13,14} The core of the model is the ideal thrust

\[ T = \frac{\eta P}{v_\infty \sin \alpha + v_i} \]  

(16)

\( P \) signifies the electric engine power input, which is converted to aerodynamic power using a single efficiency \( \eta \). For the electric power we use\textsuperscript{13}

\[ P = \frac{U - K_{em} f \omega}{R_a} U \]  

(17)

The induced velocity \( v_i \) is given by the well-known equation

\[ v_i = v_h \frac{v_z^2}{\sqrt{(v_\infty \cos \alpha)^2 + (v_\infty \sin \alpha + v_i)^2}} \]  

(18)

where \( v_h \) is the induced velocity in hover that is required to support one quarter of the assumed weight of the quadrotor. In the range of vertical velocity \(-2v_h \leq v_z < 0\), where \( v_z = v_\infty \sin \alpha \), it is assumed that the rotor is in the vortex ring state. Its induced velocity is then approximated by\textsuperscript{15}

\[ v_i = v_h \left( k_0 + k_1 \frac{v_z}{v_h} + k_2 \left( \frac{v_z}{v_h} \right)^2 + k_3 \left( \frac{v_z}{v_h} \right)^3 + k_4 \left( \frac{v_z}{v_h} \right)^4 \right) \]  

(19)

with \( k_0 = 1, k_1 = -1.125, k_2 = -1.372, k_3 = -1.718, \) and \( k_4 = -0.655 \).

The data set is generated using a set of 5000 random inputs within the operating range of the motor. Pseudo-random noise bounded to 1\% of the maximum thrust is added to the simulated output.

B. Simplex spline model

To fit the data set generated using the analytic model of the previous subsection, a simplex spline is employed first. The construction of the model is inspired by De Visser’s approach.\textsuperscript{1}

The first step in the system identification process is deciding on the model structure. By filling in Eq. (17), (18), and (19) into Eq. (16) we find a system of equations with four variables: \( v_\infty, \alpha, \omega, \) and \( u \). The spline model will thus be four-dimensional. Because Eq. (18) is a quartic equation, a fourth degree polynomial is required. To limit the freedom of the separate polynomials, first order continuity constraints are imposed. To preserve the simplicity of the model, a standard type-I triangulation of the rectangular domain is used. This results in a total of 22 simplices. The model structure is summarized in table 1.
The second step is to use the model structure in the defining equations of subsection A to arrive at the
convex problem of Eq. (8). The most time-consuming part of this process is defining the smoothness matrix.
Finally the problem can be solved using the iterative solver described before. A total of 1540 B-coefficients
are estimated using the data set of 5000 points.
Several cuts of the spline are shown in figure 7. When a value for rotor rpm and input voltage is chosen,
the well-known dependence of thrust on velocity and angle of attack is clearly visible. In the other cuts the
quadratic relation between the thrust and the input voltage and rotor rpm can be observed. Especially at
higher input voltages the parabolic shape is adhered to. Close inspection however shows that in some cases
small non-quadratic variations can be observed.

C. Multiplex spline model

The multiplex spline model is constructed along the lines of section III. An attempt is made to arrive at a
model with a comparable number of B-coefficients as in the simplex spline model.

The model should again include the four variables $v_\infty$, $\alpha$, $\omega$, and $u$, so it will be four-dimensional. From
Eq. (17) and Eq. (18) it is clear that although the induced velocity requires a polynomial model of degree
4, this is not the case for the power. In fact, a quadratic model should suffice to perfectly model the power
as a function of rotor rpm and input voltage. In the multiplex spline framework this difference in degrees
can be adhered to by choosing appropriate layers.

In the first layer the freestream velocity $v_\infty$ and the angle of attack $\alpha$ are combined. From early iteration
of the algorithm it was found that in angle of attack extra approximation power was required. Therefore a
triangulation of four simplices was chosen in this layer, generated by splitting the range of angle of attack in
half before making a minimal type-I triangulation. In the end this procedure also results in a similar amount
of B-coefficients as in the simplex spline model. First order continuity is used to limit the spline freedom, as
before.

In the second layer the remaining two variables $\omega$ and $u$ are taken together to model the engine dynamics.
The simple model allows for a quadratic spline on a minimal triangulation of two simplices. First order
continuity is again imposed, even though this is rather high for a quadratic spline. In practice this means
that the two piecewise polynomials will almost form a single parabola. The resulting structure of the
multiplex spline is compared with the simplex spline structure in table 1.

In combining the triangulations in a tessellation, no extra operations are required. A rectangular data set
is used, meaning no multiplices will be (too) empty. The tessellation thus consists of 8 multiplices, together
containing 1440 B-coefficients.

The resulting spline model is shown in figure 8. The relation between thrust, freestream velocity and
angle of attack is again properly modeled. The main difference is the fact that the engine dynamics are now
quadratic by definition. This removes the unwanted higher degree effects in this part of the model without
affecting the approximation power in the other layer.
Table 1: Comparison between the simplex and multiplex spline structure used to model the quadrotor thrust as a function of air inflow and engine dynamics. The term 'inflow layer' refers to the layer of variables $v_\infty$ and $\alpha$.

<table>
<thead>
<tr>
<th>Model design parameters</th>
<th>Model characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$ $d$ $r$ $N$ B-coefficients Constraints</td>
</tr>
<tr>
<td>Simplex spline</td>
<td>4 4 1 22 1540 1760</td>
</tr>
<tr>
<td>Multiplex spline</td>
<td>4 6* 1 8 1440 1548</td>
</tr>
<tr>
<td>Inflow layer</td>
<td>2 4 1 4 60 27</td>
</tr>
<tr>
<td>Engine layer</td>
<td>2 2 1 2 24 15</td>
</tr>
</tbody>
</table>

* No complete basis (see section II).

Table 2: The relative root mean square error (RRMS), mean square error (MSE, normalized with maximum thrust), and maximum error (ME, in percent of maximum thrust) for both the modeling data set (with noise) and the verification data set. The differences between the two data sets show that no significant overfitting to noise takes place. The overall performance of the multiplex spline model is better than that of the simplex spline model.

<table>
<thead>
<tr>
<th>Modeling</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRMS, % MSE, $10^{-5}$ ME, %</td>
<td>RRMS, % MSE, $10^{-5}$ ME, %</td>
</tr>
<tr>
<td>Simplex spline</td>
<td>2.397 5.012 4.494</td>
</tr>
<tr>
<td>Multiplex spline</td>
<td>1.934 3.269 1.998</td>
</tr>
</tbody>
</table>

D. Model quality assessment and comparison

The comparison between the simplex and multiplex spline models focuses on efficiency. The central question is what approximation power is achieved at what cost.

It was noted before that both models were designed to use similar amounts of B-coefficients. In the end, the simplex spline uses 1540 B-coefficients, whereas the multiplex spline contains 1440. These are constrained using a total of 1760 and 1548 continuity conditions respectively, of which 403 and 405 are redundant. The simplex spline thus requires more continuity constraints per B-coefficient, resulting in a larger smoothness matrix.

The smoothness matrix is not just smaller in the multiplex spline framework, but also less demanding to compute. Whereas the complete smoothness matrix for the simplex spline must be generated based on relations between simplices, the matrix is constructed of copies of smaller matrices in the multiplex case. This leads to a significant reduction in computation time.

To measure the quality of the fit, the relative root mean square error (RRMS, RMS of the error divided by RMS of the data), the mean square error (MSE), and maximum error (ME) are used. The results are presented in table 2. Note that due to the nature of the model, it is possible to generate a verification data set that does not contain noise. The RRMS and MSE values are given for both data sets. In the first place these figures show that no significant overfitting of the noise in the modeling data takes place, since the errors are smaller for the crisp verification data set. Most important is that in every metric the multiplex spline model performs better than the simplex spline.

The differences between the two splines become clear when looking at the error metrics. The relative root mean square error shows that both splines have a small error relative to the system dynamics and noise. The multiplex spline however is almost three times as accurate in predicting system outcome as the simplex spline. The mean square error for the simplex spline is more than a factor 8 higher, indicating that this spline is less capable of modeling the complete dynamics in the way the multiplex spline does. The same can be observed from the difference in maximum error and when comparing figure 7a and 8a. One explanation for this may be that the continuity constraints in the simplex spline are not in line with the system dynamics. That is, they relate B-coefficients that lie at control points that differ in all variables, whereas the constraints in the multiplex spline lie in the chosen layers.
V. Conclusion

In this paper a new multivariate spline was introduced and an algorithm for system identification using this spline was proposed. The thrust of a quadrotor was modeled to show the applicability of the spline and the algorithm.

The major benefit of the multiplex spline over the simplex spline is the fact that it allows for including knowledge of the system in the model. Any subdivision of the variables in layers can be made, depending on the expected type of coupling between them. Due to the nature of the data set used in this paper, the choice for layers was rather straightforward. If no analytic model of the system is present, many iterations of the proposed algorithm are required.

In modeling the quadrotor thrust, the multiplex is able to take full advantage of the decoupling between engine dynamics and aerodynamics. This results in a better fit to the data set than in the case of the simplex spline. Although the differences are small in this specific case, it is expected that especially the option of choosing multiple degrees will introduce significant improvements in the overall model fit. One part of the model can be given high approximation power by using a high degree while simultaneously preventing overfitting in other layers.

An extra benefit of the multiplex spline is the reduced computation time for the smoothness matrix. Because continuity constraints are defined in layers and then copied, the total amount of operations is significantly smaller than in the case of the simplex spline. The current research implies that in many cases the multiplex spline will also require less continuity conditions in general, resulting in faster computations overall.

References