Nonlinear Control Allocation Using a Piecewise Multilinear Representation

JAHANZEB RAJPUT  
HAFIZ ZEESHAN IQBAL KHAN

Centers of Excellence in Science and Applied Technologies, Islamabad, Pakistan

Nonlinear control allocation is an important part of modern nonlinear dynamic inversion based flight control systems, which require a highly accurate model of aircraft aerodynamics. Generally, an accurately implemented onboard model determines how well the system nonlinearities can be canceled. Thus, a more accurate model results in better cancelation, leading to higher performance of the controller. In this article, a control system is presented that combines nonlinear dynamic inversion with a piecewise multilinear representation based control allocation. The piecewise multilinear representation is developed through a generalization of the Kronecker product for block matrices, combined with the canonical piecewise linear representation of nonlinear functions. Analytical expressions for the Jacobian of the piecewise multilinear model are also presented. The proposed formulation gives an equivalent analytical representation of piecewise multilinear aerodynamic data and, thus, is capable of accurately modeling nonlinear aerodynamics over the entire flight envelope of an aircraft. The resulting nonlinear controller is applied to control a tailless flying wing aircraft with ten independently operating control surfaces. The simulation results for two innovative control surface configurations indicate that an accurate control allocation performance can be achieved, leading to better tracking performance compared with the control allocation methods based on multivariate polynomials and splines.

I. INTRODUCTION

Nonlinear dynamic inversion (NDI) is increasingly becoming a popular control technique for modern high-performance aerospace systems, primarily due to inherent automatic gain scheduling [1], [2]. Part of the NDI, which inverts the control effector model, is otherwise known as control allocation. Modern aircraft are often designed to be overactuated with each control effector influencing multiple axes. The modular nature of NDI allows to reassign/reconfigure the role of each control effector, according to the requirements of the operating flight regime, using only control allocation and without modification of baseline control law [3], [4]. Using a control allocation method, the desired commands are distributed over the available control effector suite, in such a way that the desired control effect is produced, along with fulfilling some additional objectives like minimization of deflections, drag and/or radar cross-section, etc. In modern tailless configurations, such as the innovative control effector concept, sometimes it becomes necessary to exploit the secondary axis yaw-power of elevons to provide artificial directional stability [5], [6]. Usually, the yaw-power of an elevon is a highly nonlinear and asymmetric function of deflection. Other control devices, like spoiler-slot-deflectors and clamshell surfaces, also show nonlinear characteristics as well as control interactions [7]. Thus, a suitable nonlinear control allocation method is necessarily required to compute surface deflections that accurately produce the demanded control effect [8].

A popular and computationally efficient framework for solving the nonlinear control allocation problem is known as the affine control allocation, in which the locally affine approximation of a nonlinear model of control surface effectiveness is computed at every sampling instant, and used as an input to a linear control allocation method [9], [10]. A discrete-time form of the affine control allocation that is suitable for implementation on digital computers is known as the incremental (or Frame-wise) control allocation [11], [12], [13]. The advantage of the incremental control allocation is that it can also account for the actuator rate-limits, actuator dynamics, and interactions between control effectors. Other approaches include sequential quadratic programming [14], mixed-integer linear programming [15], rule-based linear programming [16], second-order cone programming [17], neural networks [18], [19], etc.

A nonlinear control allocation method requires an accurate model of control surface effectiveness. Traditionally, this model is obtained by approximating the aerodynamic data of control surfaces using ordinary polynomials [11]. The aerodynamic data normally depend on multiple variables, for which the use of multidimensional polynomials can provide limited modeling accuracy. Moreover, high-degree polynomials exhibit Runge’s phenomenon leading to false nonmonotonicity, which may render affine control allocation inapplicable [20]. Recently, efforts have been
made to improve the accuracy of polynomial-based on board models using simplex spline functions [21], [22]. Simplex splines show higher approximation power than ordinary polynomials. Some disadvantages of the spline-based effector model are that it requires a large number of coefficients and, the local effectiveness matrix is obtained through a complicated transformation from Barycentric-coordinates to Cartesian-coordinates. These disadvantages increase the algorithmic complexity and add to the computational cost. In [15], the control allocation problem was formulated as a piecewise linear program to be solved with mixed-integer linear programming (MILP). The downsides of this method are that it is only applicable for the separable functions, and, the MILP is computationally very slow. Thus, neither can it incorporate the control surface interactions, nor it is suitable for the real-time digital flight control systems. In [16], a modified simplex algorithm was proposed to avoid the computational issues of MILP, but with an additional issue of premature termination of the simplex algorithm even for an otherwise feasible solution.

In this article, a new nonlinear control allocation method is presented, which incorporates the control surface model based on a piecewise multilinear representation (PMLR). The proposed PMLR is developed by combining a new generalization of the Kronecker product for block matrices with the canonical piecewise linear representation of nonlinear functions [23], [24]. Since aerodynamic data of an aircraft’s control surfaces, obtained through either wind-tunnel testing or CFD simulations, are available at grid points and usually linearly interpolated in between, for the nonlinear 6-DOF simulation of an aircraft, it is inherently a piecewise multilinear function of states and control surface deflections. The proposed formulation is capable of accurately modeling such data, thus an accurate onboard implementation of the model of nonlinear control surface effectiveness over the entire flight envelope of an aircraft is possible. Moreover, the nonseparable functions can also be modeled, thus, the control surface interactions can be incorporated in the control allocation method. The PMLR-based model is embedded within incremental control allocation scheme to provide computationally efficient control allocation solution. The nonlinear 6-DOF simulation model of a miniature tailless flying wing aircraft is used in this work to evaluate the performance of the proposed PMLR-based nonlinear control allocation. This aircraft has ten independently operating aerodynamic control surfaces with nonlinear moment versus deflection characteristics and control interactions. A modular control law, comprising the control allocation method and a baseline NDI controller, is designed for this aircraft to evaluate the control allocation performance.

The rest of this article is organized as follows. Section II recapitulates the incremental nonlinear control allocation. In Section III, the proposed piecewise multilinear representation is presented. In Section IV, a PMLR-based nonlinear control law design is presented, which incorporates the proposed nonlinear control allocation method. In Section V, the proposed, polynomial-based and spline-based nonlinear control allocation methods are evaluated and compared using closed-loop 6-DOF simulation results. Finally, Section VI concludes the article.

II. INCREMENTAL NONLINEAR CONTROL ALLOCATION

Generally, a nonlinear control allocation method solves an underdetermined, typically constrained, system of equations representing a nonlinear mapping between control surface deflections $\delta \in \mathbb{R}^n$ and corresponding control effect, $T_b \in \mathbb{R}^d$, where $k > d$. This nonlinear mapping is given as

$$T_b = g(x, \delta)$$ (1)

where $x \in \mathbb{R}^n$ is a vector of aerodynamic state-variables. Given the required control effect (the control demand), $T_{dem} \in \mathbb{R}^d$, $\delta$ is sought such that

$$g(x, \delta) = T_{dem}$$ (2)

subject to

$$\delta_{min} \leq \delta \leq \delta_{max}$$

$$|\delta| \leq \delta_{max}$$ (3)

where $\delta_{min} \in \mathbb{R}^n$ and $\delta_{max} \in \mathbb{R}^n$ are the vectors of upper and lower limits of deflections, respectively; $\delta_{max} \in \mathbb{R}^n$ are control rate-limits.

With incremental reformulation of the above nonlinear control allocation problem [11], [13], instead of solving for the whole magnitude of deflection, a solution for increment in the previous deflection is sought at every sampling instant. Thus, one requires to find incremental deflection, $\Delta \delta$, such that

$$G\Delta \delta = \Delta T_{dem}$$ (4)

$$\Delta \delta \leq \Delta \delta \leq \Delta \delta$$ (5)

where, $G \in \mathbb{R}^{d \times n}$ is the control effectiveness matrix

$$G = \left( \frac{\partial g}{\partial \delta} \right)_{(x_0, \delta_0)}$$ (6)

and

$$\Delta T_{dem} = T_{dem} - g(x_0, \delta_0).$$ (7)

Here $g(x_0, \delta_0)$ is the onboard estimation of control-effect produced by the previous deflection, $\delta_0$. The incremental limits, $\Delta \delta \in \mathbb{R}^n$ and $\Delta \delta \in \mathbb{R}^n$, are specified as

$$\Delta \delta = \min(\delta_{max} - \delta_0, \delta_{max} \Delta t)$$

$$\Delta \delta = \max(\delta_{min} - \delta_0, -\delta_{max} \Delta t).$$ (8)

The vector of total actuator commands, $\delta$, is then computed as

$$\delta = \delta_0 + \Delta \delta.$$ (9)

The resulting affine control allocation problem of (4)–(5) is then solved by any of the several available efficient and well-tested linear control allocation solvers [10], [25].

The affine/incremental framework of nonlinear control allocation requires an updated local control-effectiveness
matrix, $G$, at every sampling instant, to compute control surface deflections. Thus, the accuracy of nonlinear control allocation largely depends upon how well the matrix, $G$, locally approximates the nonlinear effectiveness of a control surface suit, given as $g(x, \delta)$. Thus, the estimation of the matrix, $G$, requires an onboard implementation of the control surface model, $g(x, \delta)$. This model is also required for the computation of incremental control demand, $\Delta T_{dem}$, at every sampling instant.

III. PIECEWISE MULTILINEAR REPRESENTATION

In this section, we develop a new formulation to model multivariate piecewise multilinear functions, which is used subsequently to develop control surface models for use with incremental control allocation. First, a new generalization of the Kronecker product for block matrices is presented. Such generalizations are often termed as the block Kronecker products (see, e.g., [26], [27], [28]). This block Kronecker product is then combined with the canonical piecewise linear representation [23], [24] to form a piecewise multilinear representation.

**Definition 1 (Block Kronecker Product)** Let $A$ be a $m \times n$ matrix and $B$ be a $p \times q$ matrix, where $n = kp$ for some positive integer $k$. Then, the Block Kronecker Product is defined in two steps:

1) Partition $A$ into $k$ blocks $A_i$, each of size $m \times p$, i.e.,

$$
A = \begin{bmatrix} A_1 & \cdots & A_k \end{bmatrix}.
$$

(10)

2) Perform the product operation as follows:

$$
A \otimes B = \begin{bmatrix} A_1B & \cdots & A_kB \end{bmatrix}
$$

(11)

where “$\otimes$” is defined as the Block Kronecker product operator.

**Remark** It must be noted that the condition for existence of the Block Kronecker Product ($A \otimes B$) is that the number of columns of $A$ must be multiple integer of $A$ must be multiple of the number of rows of $B$. However, if both are equal, then this product reduces to the standard matrix product.

Now let us present some important properties of the Block Kronecker Product.

1) **Homogeneity**: For any scalar $\lambda$, and matrices $A$ and $B$ of appropriate sizes, we have

$$
\lambda(A \otimes B) = (\lambda A) \otimes B = A \otimes (\lambda B).
$$

(12)

2) **Distributivity**: For any matrices $A, B, A_1, A_2, B_1, B_2$ of appropriate sizes, we have

$$
A \otimes (B_1 + B_2) = (A \otimes B_1) + (A \otimes B_2),
$$

$$
(A_1 + A_2) \otimes B = (A_1 \otimes B) + (A_2 \otimes B).
$$

(13)

3) **Associativity**: For any matrices $A, B$, and $C$ of appropriate sizes, we have

$$
A \otimes (B \otimes C) = (A \otimes B) \otimes C.
$$

(14)

4) **Inverse**: For any invertible matrix $A$ with inverse $A^{-1}$

$$
A \otimes A^{-1} = A^{-1} \otimes A = I.
$$

(15)

The proof of these properties directly follows from definition (11). Note that the linearity of the operator is an immediate consequence of homogeneity and distributivity.

Using Definition 1, now we present the piecewise multilinear representation in the theorem below.

**Theorem 1 (Piecewise Multilinear Representation)** Consider a vector-valued piecewise multilinear function $g : \mathbb{R}^k \mapsto \mathbb{R}^m$, whose values are given along a rectilinear grid $\{z_1, z_2, \ldots, z_k\}$ at $[\mu_1^{(1)}, \ldots, \mu_k^{(1)}] \times [\mu_2^{(2)}, \ldots, \mu_k^{(2)}] \times \cdots \times [\mu_k^{(L)}, \mu_k^{(L)}]$. Then, the $g(z_1, z_2, \ldots, z_k)$ can be compactly expressed using the piecewise multilinear representation as

$$
g = \left( (\Gamma_1 \otimes \hat{z}_k) \otimes \hat{z}_{k-1} \cdots \otimes \hat{z}_2 \right) \hat{z}_1
$$

(16)

where $\Gamma \in \mathbb{R}^{m \times \prod_{i=1}^{L} L_i}$, and

| $z_j$ |
|-----|
| $\Gamma_j = \begin{bmatrix} 1 \\
\hat{z}_j - \gamma_j^{(2)} \\
\vdots \\
\hat{z}_j - \gamma_j^{(L)} \end{bmatrix}$, \quad \forall j \in [1, k].
| (17)

**Proof** Since any single variable piecewise linear function can be written in canonical form [23], [24] as

$$
f(z) = \gamma \hat{z}
$$

(18)

where $\gamma = [\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(L)}]$ is a matrix of coefficients, we can write the multivariate function $g(z_1, z_2, \ldots, z_k)$ as

$$
g = \Gamma_1(z_2, \ldots, z_k) \hat{z}_1
$$

(19)

where

$$
\Gamma_1 = \begin{bmatrix} \gamma_1^{(1)}(z_2, \ldots, z_k), & \ldots, & \gamma_1^{(L)}(z_2, \ldots, z_k) \\
\gamma_2^{(1)}(z_3, \ldots, z_k) \hat{z}_2, & \ldots, & \gamma_2^{(L)}(z_3, \ldots, z_k) \hat{z}_2 \end{bmatrix}
$$

(20)

Similarly, we can write

$$
\Gamma_j = \Gamma_{j+1} \otimes \hat{z}_{j+1}, \quad \forall j \in [1, k-1].
$$

(21)

Now, continuing the expansion of (19) as described by (21) for $k-1$ times (till $\hat{z}_k$) results in (16), where $\Gamma = \Gamma_k$, which completes the proof. ■

**Corollary 1 (Partial Differentiation)** Partial derivatives of a function expressed in PMLR form (16) can be computed as

$$
\frac{\partial g}{\partial z_j} = \left( \left( (\Gamma \otimes \hat{z}_k \cdots) \otimes \frac{\partial \hat{z}_j}{\partial z_j} \right) \otimes \hat{z}_{j-1} \cdots \right) \hat{z}_1
$$

(22)
where
\[
\frac{\partial \hat{z}_j}{\partial z_j} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ \text{sign} \left( z_j - \mu_j^{(1)} \right) \\ \vdots \\ \text{sign} \left( z_j - \mu_j^{(L_j - 1)} \right) \end{bmatrix}, \quad \forall \ j \in [1, k].
\]

**Proof** Proof follows directly from linearity of the Block Kronecker Product, which is evident from its properties (12) and (13). □

Next, a method is presented to compute $\Gamma$ from given values of the function $g(z)$ along the $k$-dimensional rectilinear grid points. Before proceeding further, let us define a matrix reshaping transformation.

**Definition 2 (Reshaping Transformation)** The reshaping transformation $T_{\hat{z}} : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{2 \times \kappa}$, such that $mn = \lambda \kappa$, is defined as
\[
T_{\hat{z}}(A) = (\text{vec}(x_i)^T \otimes I_k) \otimes \text{vec}(A)
\]
where vec(\cdot) denotes standard vectorization operation, $I$ denotes identity matrix, and "$\otimes$" represents standard Kronecker product.

Now consider a matrix $A \in \mathbb{R}^{m \times n}$, a set of vectors $x_i \in \mathbb{R}^L$ for all $i \in [1, L]$, and some positive integer $\lambda$ such that $mn = \lambda \kappa$ for some positive integer $\kappa$. Let $B_i = A \otimes x_i$. Then, if we define $\hat{X} = [x_1 \cdots x_L]$, and $B^T = [B_1^T \cdots B_L^T]$, then following identities holds:
\[
A = T_{\hat{z}}(T_{\hat{z}}(A))
\]
\[
B = T_{mL}(A \otimes \hat{X}).
\]

The proof of these relations directly follows from Definitions 1 and 2, and properties (12)–(15) and is omitted here for brevity. Next theorem presents the formulation to compute $\Gamma$ from a given dataset.

**Theorem 2 (PMLR Fitting)** Given a $k$-dimensional data $Y_i \in \mathbb{R}^{L_1 \times \cdots \times L_k}$ for each output $i \in [1, m]$, at a rectilinear grid defined as $[\mu^{(1)}_1, \ldots, \mu^{(L_1)}_1] \times [\mu^{(1)}_2, \ldots, \mu^{(L_2)}_2] \times \cdots \times [\mu^{(1)}_k, \ldots, \mu^{(L_k)}_k]$. Then, $\Gamma$ in (16) can be computed as follows:
\[
\Gamma = \begin{bmatrix} Q_1^T \\ \vdots \\ Q_k^T \end{bmatrix}
\]
\[
(26)
\]

where
\[
Q_i = \text{vec}(\tilde{Y}_i)
\]
\[
Q_j = T_{\hat{z}_j} \left( Q_{j-1}^T \otimes \hat{Z}_j^{-1} \right), \quad \forall \ j \in [1, k]
\]
where $\tilde{Y}_i$ is obtained by flipping dimensions of $k$-dimensional array $Y_i$ for each $i \in [1, m]$, and
\[
\hat{Z}_j = \begin{bmatrix} \tilde{z}_j^{(1)}_{z_j = \mu_j^{(L_j - 1)}} \\ \vdots \\ \tilde{z}_j^{(k)}_{z_j = \mu_j^{(L_j - 1)}} \end{bmatrix}, \quad \forall \ j \in [1, k].
\]

**Proof** Due to the structure of $\Gamma$ in (26), it is sufficient to show only the scalar-valued case ($m = 1$) since generalization to the vector-valued case is straightforward, so we will only show the scalar-valued case. Let $y_{(l_1, l_2, \ldots, l_k)}$ represent the output data at $[\mu^{(l_1)}_1, \mu^{(l_2)}_2, \ldots, \mu^{(l_k)}_k]$. Then, using (19) and (21), we can write
\[
y_{(l_1, l_2, \ldots, l_k)} = \Gamma_{(l_1, l_2, \ldots, l_k)} = \Gamma_{(l_1, l_2, \ldots, l_k)} y_{l_1, l_2, \ldots, l_k}, \quad \forall \ j \in [2, k - 1]
\]
\[
\Gamma_{(l_k)} = \Gamma \otimes \tilde{z}_j^{(k)}
\]
(27)

where $\tilde{z}_j^{(k)} = \tilde{z}_j^{(k)}_{z_j = \mu_j^{(L_j - 1)}}$. Now let us denote
\[
Q_0 = \begin{bmatrix} y_{(1,1,1,1)} \\ \vdots \\ y_{(1,1,\ldots,1)} \end{bmatrix}
\]
(28)

Since $y_{(l_1, l_2, \ldots, l_k)}$ are the elements of given dataset $Y$, here it must be noted that $Q_0 = \text{vec}(\tilde{Y})$. Also, let
\[
Q_j = \begin{bmatrix} \Gamma_{(1,1,\ldots,1)} \\ \vdots \\ \Gamma_{(1,1,\ldots,1)} \end{bmatrix}, \quad \forall \ j \in [1, k - 1].
\]
(29)

Also, since $m = 1$, so $\Gamma = Q_k$. Then, we can write (27), by using (25) as follows:
\[
Q_{(l_1, l_2, \ldots, l_k)} = T_{\hat{z}_j} \left( Q_j \otimes \tilde{Z}_j^{-1} \right), \quad \forall \ j \in [1, k].
\]
(30)

Now since $Q_j \in \mathbb{R}^{L_j \times \kappa}$, where $\lambda_j = \prod_{i=1}^{k} L_i$ and $\tilde{\lambda}_j = \prod_{i=1}^{k} L_i$, for each $j \in [0, k]$. Thus, by using (24), we get
\[
Q_j = T_{\hat{z}_j} \left( Q_{j-1} \otimes \tilde{Z}_j^{-1} \right), \quad \forall \ j \in [1, k]
\]
(31)

which concludes the proof. □

**Lemma 1** For any vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ and any matrix $A \in \mathbb{R}^{q \times mn}$, the following relation holds:
\[
(A \otimes y)x = A(x \otimes y).
\]
(32)
Using (11), we can write
\[(A \otimes y)x = \begin{bmatrix} A_1y & A_2y & \cdots & A_ny \end{bmatrix} x \]
\[= \begin{bmatrix} A_1 & A_2 & \cdots & A_n \end{bmatrix} \begin{bmatrix} x^{(1)}y \\ x^{(2)}y \\ \vdots \\ x^{(n)}y \end{bmatrix} = A(x \otimes y)\]
where \(x^{(i)}\) represents \(i\)th element of vector \(x\).

Next, an alternative formulation is presented, which uses the standard Kronecker product operation to represent the nested formulation of (16).

**Theorem 3 (Alternative Representation)** The piecewise multilinear function \(g\) [given by (16)] can be equivalently expressed as follows:
\[g = \Gamma \boxdot z_i\]
(33)
where \(\boxdot_{i=1}^k z_i\) is the compact representation of the successive standard Kronecker product operation \(z_1 \otimes z_2 \otimes \cdots \otimes z_k\).

**Proof** First consider the 2-D case \((k = 2)\) and using Lemma 1, we have
\[(\Gamma_2 \otimes \hat{z}_2)\hat{z}_1 = \Gamma_2(\hat{z}_1 \otimes \hat{z}_2)\]
(34)
Thus, (33) is true for \(k = 2\). Now, assume that it is also true for \(k = j\), so
\[\left((\Gamma_j \otimes \hat{z}_j) \otimes \hat{z}_{j-1} \cdots \hat{z}_1\right) = \Gamma_j \boxdot \hat{z}_i\]
(35)
Now, from (21)
\[\left((\Gamma_{j+1} \otimes \hat{z}_{j+1}) \otimes \hat{z}_{j} \cdots \hat{z}_1\right) = \Gamma_{j+1} \boxdot \hat{z}_i\]
(36)
applying Lemma 1 again yields
\[\left((\Gamma_{j+1} \otimes \hat{z}_{j+1}) \otimes \hat{z}_{j} \cdots \hat{z}_1\right) = \Gamma_{j+1} \boxdot \hat{z}_i\]
(37)
Hence, by the principal of mathematical induction (33) is true.

**Corollary 2** From (33), it follows that:
\[\frac{\partial g}{\partial z_j} = \Gamma \left[ \bigotimes_{i=1}^{j-1} \hat{z}_i \otimes \frac{\partial \hat{z}_j}{\partial z_j} \otimes \left( \bigotimes_{i=j+1}^k \hat{z}_i \right) \right]\]
(38)
**Proof** Its proof is a direct consequence of Theorems 1 and 3 and Corollary 1.

The advantage of alternative representation (33) is twofold. First, from an onboard implementation standpoint, alternative representation provides a straightforward formulation as compared to (16). Second, a linear regression scheme can also be formulated directly from (33) for the computation of coefficient matrix \(\Gamma\). Given the data at a grid of \(n = \prod_{i=1}^k L_j\) points, let \(x_1, x_2, \ldots, x_n \in \mathbb{R}^n\) represent values of \(\boxdot_{i=1}^k z_i\), evaluated using (17) at each grid point, and let \(y_1, y_2, \ldots, y_n \in \mathbb{R}^m\) represent corresponding output vectors. Now defining the observation matrix \(Y = [y_1, y_2, \ldots, y_n]\), and the regressor matrix \(X = [x_1, x_2, \ldots, x_n]\), we can write (33) collectively for all data points in the following compact form:
\[Y = \Gamma X\]
(39)
So \(\Gamma\) can be found by solving the linear system (39), as \(\Gamma = YX^{-1}\). It must be noted that a simple inverse is required to compute \(\Gamma\), which gives extra evidence of the fact that the multilinear representation of (16) is an equivalent analytical model of multivariate piecewise multilinear data.

For the application of nonlinear control allocation, where the number of independent variables is usually less than or equal to five, (39) can be used directly to compute \(\Gamma\). However, for applications having a large number of independent variables, the solution of (39) may outrun the computational resources of an average PC or laptop. Therefore, the iterative algorithm (Theorem 2) is recommended. However, for onboard implementation of PMLR, alternative representation (Theorem 3) is recommended due to its distinctive features.

**IV. PMLR-BASED NONLINEAR CONTROL LAW**

In this section, a PMLR-based nonlinear control law is presented for a miniature tailless flying wing aircraft. The detailed specification of the aircraft can be found in [29]. The aircraft has six flaps (control surfaces at the trailing edge of the wing) and, two (left/right) pairs of clamsheal surfaces (see Fig. 1). The numbering scheme of control surfaces is shown in Fig. 2. The control law consists of the baseline control law and the incremental control allocation, as shown in Fig. 3. The baseline control law is based on nonlinear dynamic inversion, which generates control demand in terms of a moment vector, \(T_{dem}\) [30], [31]. Control demand is then fed to the incremental control allocation algorithm to compute control commands for each control surface. The incremental control allocation algorithm is based on the redistributed pseudoinverse (RPI) method [10], which incorporates either PMLR, polynomial or spline-based control surface model. The control system
is purposefully made sensitive to control surface model inaccuracies by not including integral-action in the baseline control law so that the effects of control allocation errors could be made more noticeable.

A. Aircraft Model

The aerodynamic moment-model of the aircraft is given as follows:

\[ T = q_{\infty} S \ell_{ref} \left( C_A + \frac{1}{2V} L_{ref} C_{\omega} \omega + C_\delta \right) \tag{40} \]

where \( T \in \mathbb{R}^3 \) represents net moments; \( q_{\infty} \) is free-stream dynamic pressure; \( V \) is aircraft’s true velocity; \( S \) is reference area; \( L_{ref} = \text{diag}(b, \bar{c}, b) \) with \( \bar{c} \) and \( b \) being mean aerodynamic chord and wing span, respectively; \( \omega \in \mathbb{R}^3 \) is body-axes angular rates; \( C_A \in \mathbb{R}^3 \) represents aerodynamic moment coefficients at zero deflections; \( C_\omega \in \mathbb{R}^{3 \times 3} \) is the matrix of damping derivatives; \( C_\delta \in \mathbb{R}^3 \) is the vector of increment in aerodynamic moment coefficients due to control surface deflections, that is given as

\[ C_\delta = \sum_{i=1}^{6} \Delta C_i(\alpha, \delta_i) + \sum_{i=7}^{8} \Delta C_m(\alpha, \beta, \delta_i, \delta_U, \delta_{IL}) \tag{41} \]

where \( \Delta C_i, \Delta C_m, \) and \( \Delta C_a \) represent change in roll, pitch, and yaw moment coefficients, respectively, due to control surface deflections, which are also a function of the angle of attack \( \alpha \) and the side-slip angle \( \beta \).

All moment contributions due to the deflection of flaps \( \delta_i \) are 2-D functions. Flaps are known to have highly nonlinear and asymmetric yaw moment effectiveness about zero-deflection. Moreover, they show significant nonlinearities in pitch and roll moments for large deflection angles \([32], [33]\). Moment contributions of each clamshell pair are 4-D functions. The deflections of lower and upper clamshell surfaces are referred to as, \( \delta_{LU} \) and \( \delta_{IL} \), respectively. Traditionally, the clamshell pair is used as a pure yaw control device by deflected the lower and upper surfaces by an equal amount, thus, effectively nullifying the effectiveness in roll and pitch axes and maximizing the effectiveness in yaw axis. Apart from the traditional usage, it is also possible to use each surface in the clamshell pair as a separate control device affecting multiple axes of rotation. Such an innovative usage certainly increases the redundancy of the control surface suit. However, it requires modeling the clamshell pair data as an additively nonseparable function with respect to lower and upper surfaces, in order to account for possible control interaction effects between these surfaces \([7], [34]\). The clamshell surface pair shows fairly linear characteristics in the roll and pitch axes, but significantly nonlinear characteristics in the yaw axis \([29], [35]\).

B. PMLR-Based Incremental Control Allocation

From the aerodynamic model given in (40), the control surface effectiveness can be written as

\[ g(x, \delta) = T_\delta = q_{\infty} S \ell_{ref} C_\delta. \tag{42} \]

The PMLR-based control surface model of the aircraft is given as

\[ C_\delta = \sum_{i=1}^{6} \left( (\Gamma_i \otimes \hat{\alpha}) \otimes \hat{\delta}_i \right) + \sum_{i=7}^{8} \left( (\Gamma_i \otimes \hat{\alpha}) \otimes \hat{\beta} \otimes \delta_{IL} \right) \hat{\delta}_{LU}. \tag{43} \]

Partial derivatives of the above control surface model along all control variables are required in order to compute elements of the control effectiveness matrix, \( G := q_{\infty} S \ell_{ref} M \), where \( M \equiv \partial C_\delta / \partial \delta \) is given as

\[ M = \begin{bmatrix} \sigma_{\delta_1} & \cdots & \sigma_{\delta_6} & \sigma_{\delta_{LU}} & \sigma_{\delta_{IL}} & \sigma_{\delta_{LU}} & \sigma_{\delta_{IL}} \end{bmatrix}. \tag{44} \]

Elements of the control effectiveness matrix are computed, at every sampling instant, from the PMLR-based control surface model as follows:

\[ \sigma_{\delta_i} = (\Gamma_i \otimes \hat{\alpha}) \frac{\partial \hat{\delta}_i}{\partial \delta_i}, \quad i = 1, 2, \ldots, 6 \]

\[ \sigma_{\delta_{LU}} = \left( (\Gamma_i \otimes \hat{\alpha}) \otimes \hat{\beta} \otimes \delta_{IL} \right) \frac{\partial \hat{\delta}_{LU}}{\partial \delta_{LU}}, \quad i = 7, 8 \]

\[ \sigma_{\delta_{IL}} = \left( (\Gamma_i \otimes \hat{\alpha}) \otimes \hat{\beta} \otimes \delta_{IL} \right) \frac{\partial \hat{\delta}_{LU}}{\partial \delta_{IL}}, \quad i = 7, 8. \tag{45} \]

Now, it is straightforward to compute \( g(x_0, \delta_0) \) from (42)–(43). The computed control effectiveness matrix and \( g(x_0, \delta_0) \) are then used in (4)–(7) to form the incremental control allocation problem, which is then solved by using the RPI method as follows:

\[ \Delta \delta = \delta_p + G^T \left[ \Delta T_{dem} - G \delta_p \right]. \tag{46} \]

where

\[ G^T = W^{-1} G^T (GW^{-1}G^T)^{-1} \tag{47} \]

is the weighted pseudoinverse of matrix \( G; \delta_p \in \mathbb{R}^{10} \) is a preference vector; \( W \in \mathbb{R}^{10 \times 10} \) is a weighting matrix. The constraints of actuator position and rate are incorporated through the redistribution process as given in \([10]\).

Various components of control surface aerodynamic data of the aircraft, as given in (41), are modeled using traditional least square fittings of multivariate polynomials, splines, and also using the proposed PMLR formulation. Due to aircraft symmetry, it is sufficient to produce models of starboard control surfaces only. All three effector models are validated by passing 10 000 samples of randomly generated input vector containing surface deflections and
aerodynamic angles through the actual linearly interpolated lookup tables of aerodynamic data and the fitted control surface models. It was observed that, for all polynomial models, relative RMS errors are less than 9% and maximum errors are less than 0.0015. As for the spline models, the relative RMS errors and maximum errors are less than 8% and 0.001, respectively. Small values of errors indicate good accuracy of resulting polynomial and spline models. Since the PMLR-based effector model is an analytical but equivalent representation of the aerodynamic data, its fitting error is zero. Such accurate modeling, however, comes at the cost of a relatively larger number of coefficients than the polynomial counterparts. Table I gives a comparison of the number of coefficients for each case, which shows that the number of coefficients for PMLR model is moderately higher but acceptable for onboard implementation.

C. NDI Control Law

The rate-loop NDI control law is written in terms of a vector of demanded control moments, \( T_{dem} \), which is translated into control surface deflections using a control allocation method. Consider the aircraft’s moment equations in vector form

\[
\dot{\omega} = I^{-1}(T_a - \omega \times I\omega) + I^{-1}T_b
\]

where \( I \in \mathbb{R}^{3 \times 3} \) is the inertia matrix, \( T_a \in \mathbb{R}^3 \) is a vector of aerodynamic moments due to the airframe, which is given as follows:

\[
T_a = q_{\infty}S_{ref} \left( C_{\alpha} + \frac{1}{2V}L_{ref}C_{\alpha \omega} \right)
\]

where \( x \in \mathbb{R}^6 \) is a vector of aircraft state-variables. \( T_b \in \mathbb{R}^3 \) is a vector of moments due to control effector deflections. Now, let \( \dot{\omega}_{des} \in \mathbb{R}^3 \) be a vector of desired angular body accelerations, then to achieve the control objective \( \dot{\omega} = \dot{\omega}_{des} \), the inversion law is given as

\[
T_{dem} = T_b - I\dot{\omega}_{des} + \omega \times I\omega.
\]

In order to implement the above control law, an onboard model of multivariate function (\( T_b \)) is required. This model is also implemented using PMLR. Full-state feedback-linearization of the nonlinear system of (48) by the inversion law of (50) results in a system of three integrators, \( \dot{\omega} = \dot{\omega}_{des} \). The time-constant of each integrator channel can be set as desired by a linear controller. Thus, the complete rate-loop control law can be written as

\[
T_{dem} = IK_\omega(\dot{\omega} - \omega_{ref}) - T_a + \omega \times I\omega
\]

where \( K_\omega = \text{diag}(k_p, k_q, k_r) \). Tuned gains of the linear part of the inner-loop controller are listed in Table II.

For the angle-loop control law design, the aircraft dynamics with rate-loop closed is taken as the plant, whose input is a vector of reference angular rates, \( \omega_{ref} \). The angle-loop controlled variables are roll angle \( \phi \), pitch angle \( \theta \), and sideslip angle \( \beta \). The NDI control law of angle-loop, in terms of reference angular rates, is derived as follows.

Assuming small aerodynamic angles (\( \sin \alpha \approx \sin \beta \approx 0 \) and \( \cos \alpha \approx 1 \)), the kinematic equations can be written as

\[
\dot{\Phi} = \Lambda \omega + f_\Phi(x)
\]

where \( \Phi = [\phi, \theta, \beta]^T \) is attitude angles, \( \Lambda = \text{diag}(1, \cos \phi, -1) \), and

\[
f_\Phi(x) = \begin{bmatrix} q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ -r \sin \phi \\ \frac{1}{\beta} \cos \theta \sin \phi \end{bmatrix}.
\]

Let \( \Phi_{des} = [\dot{\phi}_{des}, \dot{\theta}_{des}, \dot{\beta}_{des}]^T \) be the desired attitude rate, then, to make \( \Phi = \Phi_{des} \), the NDI control law in terms of
reference body-axis roll-rate is obtained as

\[ \omega_{\text{ref}} = \omega = A^{-1} (\Phi_{\text{des}} - f_\Phi(x)) \]  

(53)

As a result of the above feedback-linearization, the system of nonlinear state-equations is reduced to a system of three integrators, \( \Phi = \Phi_{\text{des}} \). The outer-loop controlled variables are then controlled by using a linear proportional controller. Thus, the complete angle-loop control law can be written as

\[ \omega_{\text{ref}} = A^{-1} (K_\Phi (\Phi - \Phi_{\text{ref}}) - f_\Phi(x)) \]  

(54)

where the vector of reference input is \( \Phi_{\text{ref}} = [\phi_{\text{ref}}, \theta_{\text{ref}}, \beta_{\text{ref}}]^T \) and the matrix of proportional gains is \( K_\Phi = \text{diag}(k_\phi, k_\theta, k_\beta) \). The commands that are unachievable due to actuator constraints are avoided by passing the reference input through the first-order lag prefilters

\[ \dot{x}_{pf} = \frac{1}{\tau_{pf}} (x_{\text{ref}} - x_{pf}) \]  

(55)

where \( \tau_{pf} \in \{\tau_\phi, \tau_\theta, \tau_\beta\} \). The prefilter time-constants are chosen by a tradeoff between tracking performance and command saturation. Tuned gains of angle-loop and prefilter time-constants are listed in Table II.

V. SIMULATION RESULTS

In this section, the performance of the proposed PMLR-based nonlinear control allocation is evaluated. Simulation of the complete closed-loop control system is developed in the MATLAB and Simulink. Aircraft dynamics are based on the 6-DOF nonlinear model of the aircraft. The performance of nonlinear control allocation is compared for polynomial, spline, and PMLR-based control surface models using a U-turn maneuver. To generate such a maneuver, a square pulse of 50 deg roll angle command is tracked, whereas, the pitch angle is regulated at its initial trim value (3.73 deg) and, the sidleslip angle is regulated at zero. At the start of the maneuver, the aircraft is initially trimmed at a velocity of 25 m/s and an altitude of 500 m. Reason for the selection of such a maneuver is that sizable control demands can be produced in all three rotary axes of the aircraft body, which in turn force all control surfaces to sweep a large part of the deflection range.

Control allocation performance is evaluated for the following two different innovative effector configurations:

1) **split-aileron—Ruddervator** configuration;
2) **elevator—Rudder** configuration.

Let us express the local control effectiveness matrix in a partitioned form

\[ G = q_{\infty}L_{\text{ref}} \begin{bmatrix} M_1 & | & M_2 \end{bmatrix} \]  

(56)

where \( M_1 \in \mathbb{R}^{3 \times x_1} \) and \( M_2 \in \mathbb{R}^{3 \times x_2} \) are two partitions, which are defined according to the effector configuration under consideration.

In **Split-ailerons—Ruddervator** configuration, all four clamshell control surfaces are ganged together to work as split-aileron, whereas, all flaps work as ruddervators. This ganging scheme is expressed as follows:

\[ \delta_a = \frac{1}{2} [(\delta_{7L} - \delta_{8L}) + (\delta_{7U} - \delta_{8L})] \]  

(57)

where \( \delta_a \) is the aileron command. Any two diagonal clamshell surfaces are deflected at a time depending upon the polarity of the aileron command, which is defined such that a positive \( \delta_a \) produces a positive rolling moment and vice versa. This ganging scheme also reduces the total number of control-effectors to seven. Thus, partitions of the control effectiveness matrix become

\[ M_1 = \begin{bmatrix} \sigma_{\delta_1} & \sigma_{\delta_2} & \cdots & \sigma_{\delta_6} \end{bmatrix} \]  

(58)

\[ M_2 = \begin{bmatrix} -\sigma_{\delta_7L} + \sigma_{\delta_7U} & \delta_a < 0 \\ \sigma_{\delta_8L} - \sigma_{\delta_8U} & \delta_a \geq 0. \end{bmatrix} \]  

(59)

In this configuration, clamshell surfaces are restricted to produce only rolling moment, whereas no such restriction is applied to the flaps. Since the clamshell surfaces become the main contributors to the rolling moment, the flaps are forced to contribute in pitching and yawing moments by the control allocation method. The residual rolling moment of the flaps is also compensated by the clamshell surfaces through control allocation.

In **Elevator—Rudderon** configuration, all six flaps are ganged together to work as a single elevator. Moreover, all four clamshell control surfaces are operated independently to predominantly affect roll- and yaw-axes. The elevator ganging scheme is expressed as follows:

\[ \delta_e = \frac{1}{6} \sum_{i=1}^{6} \delta_i. \]  

(60)

This ganging scheme reduces the total number of control-effectors to five. Thus, partitions of the control effectiveness matrix become

\[ M_1 = \sum_{i=1}^{6} \sigma_{\delta_i} \]  

(61)

\[ M_2 = \begin{bmatrix} \sigma_{\delta_{7L}} & 0 & 0 & \sigma_{\delta_{8L}} & \sigma_{\delta_{8U}} \end{bmatrix}. \]  

(62)

In this configuration, the flaps are restricted to produce only pitching moment, whereas no such restriction is applied to the clamshell surfaces. Since the flaps become the main contributors to the pitching moment, the clamshell surfaces are forced to contribute in rolling and pitching moments by the control allocation method. The residual pitching moment of the clamshell surfaces is also compensated by the flaps through control allocation.

The performance is evaluated in terms of the control allocation error, \( E(t) \), which is a difference between the vector of demanded control moments, \( T_{\text{dem}}(t) \), and the vector of actual control moments delivered by the control surface
were obtained using the \texttt{tic/toc} commands, and are given for comparison purposes only.

In general, due to greater modeling accuracy, the PMLR-based control allocation produces much lower errors as compared to polynomial- and spline-based control allocation. For both configurations, the polynomial-based control allocation results in largest inaccuracies in all three control channels; thus, control performance degradation is observed. The spline-based control allocation produces relatively smaller errors and, thus, brings improvements in the control performance but with a large computational cost due to algorithmic complexity. On the other hand, the PMLR-based control allocation gives smallest errors with fastest execution times, which results in superior control performance.

VI. CONCLUSION

A nonlinear control allocation method was presented, which uses the control surface model based on a newly developed piecewise multilinear representation to compute control commands. Aerodynamic data of an aircraft’s control surfaces are usually a piecewise multilinear function of states and control surface deflections. Thus, it can be modeled accurately using the proposed piecewise multilinear representation. It was shown that the special structure of PMLR makes it possible to compute the locally affine approximation of nonlinear moment versus deflection relationships through explicit formulation. Consequently, a PMLR-based model directly fits into the computationally efficient affine control allocation framework for solving the nonlinear control allocation problem. From a simulation-based comparison, it was demonstrated that the PMLR-based control allocation is an accurate and efficient method as compared with polynomials and splines.

ACKNOWLEDGMENT

The authors are thankful to Dr. Zang Weiguo, Dr. Qu Xiaobo, and Dr. Shi Jingping (School of Automation, Northwestern Polytechnical University, Xi’an) for their technical and moral support.

REFERENCES

[1] E. Oland, R. Kristiansen, and J. T. Gravdahl, “A comparative study of different control structures for flight control with new results,” IEEE Trans. Control Syst. Technol., vol. 28, no. 2, pp. 291–305, Mar. 2020.
[2] E. Tal and S. Karaman, “Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness,” IEEE Trans. Control Syst. Technol., vol. 29, no. 3, pp. 1203–1218, May 2021.
[3] Y. Yang and Z. Gao, “A new method for control allocation of aircraft flight control system,” IEEE Trans. Autom. Control, vol. 65, no. 4, pp. 1413–1428, Apr. 2020.
[4] Q. Shen, D. Wang, S. Zhu, and E. K. Poh, “Robust control allocation for spacecraft attitude tracking under actuator faults,” IEEE Trans. Control Syst. Technol., vol. 25, no. 3, pp. 1068–1075, May 2017.
[5] M. A. Niestroy, K. M. Dorsett, and K. Markstein, “A tailless fighter aircraft model for control-related research and development,” in Proc. AIAA Model. Simul. Technol. Conf., 2017, pp. 1–18. [Online]. Available: https://doi.org/10.2514/6.2017-1757
[6] K. M. Dorsett and R. D. Mehl, “Innovative control effectors (ICE),” Lockheed Martin Tactical Aircraft Systems Fort Worth TX, Tech. Rep. WL-TR-96-3043, 1996.

[7] J. Rajput, Q. Xiaoabo, and H. Z. I. Khan, “Nonlinear control allocation for interacting control effectors using bivariate taylor expansion,” in Proc. 5th Int. Conf. Control Decis. Inf. Technol., 2018, pp. 352–357.

[8] M. Benosman, F. Liao, K. Lum, and J. L. Wang, “Nonlinear control allocation for non-minimum phase systems,” IEEE Trans. Control Syst. Technol., vol. 17, no. 2, pp. 394–404, Mar. 2009.

[9] D. Doman and M. Oppenheimer, “Improving control allocation accuracy for nonlinear aircraft dynamics,” in Proc. AIAA Guid. Navigation Control Conf. Exhibit., Aug. 2002, pp. 1–10. [Online]. Available: https://doi.org/10.2514/6.2002-4667

[10] M. Oppenheimer, D. Doman, and M. Bolender, The Control Handbook—Control System Applications, 2nd ed. Boca Raton, FL, USA: CRC Press, 2011.

[11] K. A. Bordignon, “Constrained control allocation for systems with redundant control effectors,” Ph.D. dissertation, Faculty of Virginia Polytechnic Institute State Univ., Blacksburg, Virginia, 1996.

[12] W. Durham, K. A. Bordignon, and R. Beck, Aircraft Control Allocation, Hoboken, NJ, USA: Wiley, 2017.

[13] I. Matamoros and C. C. de Visser, “Incremental nonlinear control allocation for a tailless aircraft with innovative control effectors,” in Proc. AIAA Guid. Navigation Control Conf., 2018, pp. 1–25. [Online]. Available: https://doi.org/10.2514/6.2018-1116

[14] T. A. Johansen, T. I. Fossen, and S. P. BERGE, “Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming,” IEEE Trans. Control Syst. Technol., vol. 12, no. 1, pp. 211–216, Jan. 2004.

[15] M. A. Bolender and D. B. Doman, “Nonlinear control allocation using piecewiselinear functions,” J. Guid. Control Dyn., vol. 27, no. 6, pp. 1017–1027, Nov. 2004. [Online]. Available: https://doi.org/10.2514/1.9546

[16] M. A. Bolender and D. B. Doman, “Nonlinear control allocation using piecewiselinear functions: A linear programming approach,” J. Guid. Control Dyn., vol. 28, no. 3, pp. 558–562, May 2005. [Online]. Available: https://doi.org/10.2514/1.12997

[17] Q. Hu, G. Niu, and C. Wang, “Spacecraft attitude fault-tolerant control based on iterative learning observer and control allocation,” Aerosp. Sci. Technol., vol. 75, pp. 245–253, 2018.

[18] Y. Yu, H. Wang, and N. Li, “Fault-tolerant control for over-actuated hypersonic reentry vehicle subject to multiple disturbances and actuator faults,” Aerosp. Sci. Technol., vol. 87, pp. 230–243, 2019.

[19] H. Z. I. Khan, S. Mboeen, J. Rajput, and J. Riaiz, “Nonlinear control allocation: A learning based approach,” 2022, arXiv:2201.06180.

[20] O. Härkegård, “Backstepping and control allocation with applications to flight control,” Ph.D. dissertation, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2003.

[21] H. J. Tol, C. C. de Visser, E. J. Van Kampen, and Q. P. Chu, “Multivariate simplex spline based nonlinear dynamic inversion control of high performance aircraft,” in Proc. AIAA Guid. Navigation Control Conf., Aug. 2013, pp. 1–23. [Online]. Available: https://doi.org/10.2514/6.2013-4925

[22] H. J. Tol, C. C. de Visser, E. J. Van Kampen, and Q. P. Chu, “Nonlinear multivariate spline-based control allocation for high-performance aircraft,” J. Guid. Control Dyn., vol. 37, no. 6, pp. 1840–1862, Apr. 2014. [Online]. Available: https://doi.org/10.2514/1.G000065

[23] L. O. Chua and A. Deng, “Canonical piecewise-linear representation,” IEEE Trans. Circuits Syst., vol. 35, no. 1, pp. 101–111, Jan. 1988.

[24] C. Kahlert and L. O. Chua, “A generalized canonical piecewise-linear representation,” IEEE Trans. Circuits Syst., vol. 37, no. 3, pp. 373–383, Mar. 1990.

[25] M. Bodson, “Evaluation of optimization methods for control allocation,” J. Guid. Control Dyn., vol. 25, no. 4, pp. 703–711, Jul. 2002. [Online]. Available: https://doi.org/10.2514/2.4937

[26] D. S. Tracy and R. P. Singh, “A new matrix product and its applications in partitioned matrix differentiation,” Statistica Neerlandica, vol. 26, no. 4, pp. 143–157, 1972.

[27] D. C. Hyland and E. G. Collins Jr., “Block kroncker products and block norm matrices in large-scale systems analysis,” SIAM J. Matrix Anal. Appl., vol. 10, no. 1, pp. 18–29, 1989. [Online]. Available: https://doi.org/10.1137/0610002

[28] R. H. Koning, H. Neudecker, and T. Wansbeek, “Block kroncker products and the vecb operator,” Linear Algebra Appl., vol. 149, pp. 165–184, 1991.

[29] X. Qu, W. Zhang, J. Shi, and Y. Lyu, “A novel yaw control method for flying-wing aircraft in low speed regime,” Aerosp. Sci. Technol., vol. 69, pp. 636–649, 2017.

[30] D. Ens, D. Bugajski, R. Hendrick, and G. Stein, “Dynamic inversion: An evolving methodology for flight control design,” Int. J. Control, vol. 59, no. 1, pp. 71–91, 1994.

[31] S. H. Lane and R. F. Stengel, “Flight control design using nonlinear inverse dynamics,” Automatica, vol. 24, no. 4, pp. 471–483, 1988.

[32] G. Stenfelt and U. Ringerz, “Lateral stability and control of a tailless aircraft configuration,” J. Aircr., vol. 46, no. 6, pp. 2161–2164, Nov. 2009. [Online]. Available: https://doi.org/10.2514/1.41092

[33] G. Stenfelt and U. Ringerz, “Yaw control of a tailless aircraft configuration,” J. Aircr., vol. 47, no. 5, pp. 1807–1811, Sep. 2010. [Online]. Available: https://doi.org/10.2514/1.C031017

[34] M. Oppenheimer and D. Doman, “A method for including control effector interactions in the control allocation problem,” in Proc. AIAA Guid. Navigation Control Conf. Exhibit, Hilton Head, South Carolina, 2007, pp. 1–10. [Online]. Available: https://doi.org/10.2514/6.2007-6418

[35] J. Rajput, W. G. Zhang, and X. B. Qu, “A differential configuration of split drag-rudders with variable bias for directional control of flying-wing,” Appl. Mechanics Mater., vol. 643, pp. 54–59, 2014.

Jahanzeb Rajput received the B.E. degree in electronics engineering from the Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2005, the M.S. degree in control engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2007, and the Ph.D. degree in navigation, guidance and control from Northwestern Polytechnical University, Xi’an, China, in 2016. His main research interests include nonlinear control, control allocation, fault-tolerant control, reconfigurable control, modeling, identification, and simulation.

Hafiz Zeeshan Iqbal Khan received the B.S. degree magna cum laude in aerospace engineering with honors and President of Pakistan gold medal and the M.S. degree magna cum laude in aerospace engineering specializing in guidance, navigation and control, with honors from the Institute of Space Technology, Islamabad, Pakistan, in 2015 and 2019, respectively. His main research interests include dynamics and control of aerospace vehicles, robust control, nonlinear control, geometric control, and machine learning.