MODEL PREDICTIVE CONTROL FOR MANEUVER LOAD ALLEVIATION IN FLEXIBLE AIRLINERS

Mateus de F. V. Pereira¹, Ilya Kolmanovsky¹, Carlos E. S. Cesnik¹, Fabio Vetrano²

¹University of Michigan
Ann Arbor, Michigan USA
mfvp@umich.edu
ilya@umich.edu
cesnik@umich.edu

²Airbus Operations SAS
Toulouse, France
fabio.vetrano@airbus.com

Keywords: flexible aircraft, load alleviation, model predictive control

Abstract: This paper addresses the development of model predictive control (MPC)-based MLA governors for maneuver load alleviation (MLA) in airliners with increased levels of flexibility. An MLA governor augments the existing linear controller in order to enforce MLA constraints. To enforce a large number of MLA constraints a constraint aggregation approach which exploits Kreisselmeier-Steinhauser function is used. The performance of the resulting control strategy is evaluated and compared with a traditional approach used for MLA in commercial aircraft.

1 INTRODUCTION

Commercial aircraft designs are evolving towards being more lightweight, having higher aspect ratios and becoming increasingly flexible in order to improve aerodynamic performance and meet demanding flight mission specifications which call for reduced fuel consumption, heavier payload and greater range and endurance [1]. Airworthiness certification mandated by regulatory agencies requires demonstrating that critical loads in these aircraft do not exceed specified limits that ensure safety and structural integrity. These requirements also apply to loads generated during different maneuver conditions. The objective of limiting wing loads imposes restrictions on the V-n envelope and thus invariably conflicts with the goal of maximizing maneuver performance. However, with the knowledge of how each control surface affects maneuvers and loads, the control system design can exploit the strengths of the various control effectors to minimize conflicts [2].

Maneuver load alleviation (MLA) can be approached as a constrained control problem [3] in which loads at critical stations must be kept within safety limits by deflecting the control surfaces. Model predictive control (MPC) [4] and reference governors (RG) [5] are of significant interest for MLA due to their ability to enforce pointwise-in-time constraints on the loads and actuation of control effectors. In addition, these methods can be integrated with other control design techniques, such as nonlinear, adaptive and robust control, in order to exploit a combination of several methods. Previous studies on MLA systems for flexible aircraft [6–8] and
very flexible aircraft [9–12] showed that both control strategies are successful in achieving command tracking and enforcing constraints in order to achieve load alleviation when subjected to maneuver or gust loads.

One of the challenges for implementing MPC to perform MLA is the large computational footprint of solving a dynamic optimization problem in real-time, in which a sequence of control inputs is computed that minimizes a performance index subject to the specified constraints within a defined prediction horizon. This large computational footprint is a consequence of flexible aircraft having numerous constraints at various critical stations along the fuselage, wing and tail and their models and dynamics being high order. In particular, as the aircraft flexibility increases, the computation footprint of the MPC controller also increases [11]. At the same time, flight control systems for large transport aircraft have typically only a cycle time of 10 ms to 30 ms within which measurements, state estimation and control computation have to be performed [3].

This paper presents an MPC-based MLA governor for augmenting the nominal control system in flexible aircraft with different levels of flexibility to perform MLA. The MLA governor adjusts the references to the inner-loop LQ-I controller, similarly to the extended command governor [5], and, in addition, directly manipulates the control effectors that are used to perform MLA. Constraints are imposed on the out-of-plane curvature at critical stations to limit the bending moment. Differently from the work presented in [11], the strain states are obtained by reconstruction of the state vector using an observer and measurements from inertial and accelerometer sensors. Furthermore, this work presents various techniques to reduce the computational footprint of implementing the MPC controller. In particular, multiple constraints are handled by using the Kreisselmeier-Steinhauser function to perform constraint aggregation over the prediction horizon and for different output channels [13]. Model order reduction and move blocking techniques are also applied to obtain further reduction in the computation cost. The performance of the proposed MLA governor is compared against a traditional approach used for MLA in commercial aircraft to demonstrate the strengths of the MPC solution.

The remaining sections of this paper are organized as follows. Section 2 describes the MLA problem and control requirements. Section 3 presents the control system design for MLA. Section 4 describes the flexible aircraft models. Section 5 presents closed-loop simulation results based on linear models. Finally, Section 6 presents the conclusions from the numerical studies and comments on future research opportunities.

2 MANEUVER LOAD ALLEVIATION

Conventional approaches to maneuver load alleviation call for automatically deflecting control surfaces, such as elevons, to concentrate lift inboard and reduce the wing bending moment at critical stations. These control surfaces are deflected proportionally to some monitored parameters (e.g., load factor or wing curvature), which are obtained from sensor measurements. Sensors used in MLA applications include wing deflection sensors, accelerometers, strain gauges and wing angle of attack sensors [14].

Typical MLA designs, such as the one described in [15], define a threshold to trigger the MLA system. The threshold-based activation is implemented to avoid repeated and superfluous operation of the control surfaces, which should be deflected only when the monitored parameters at the selected stations reach critical values.

However, when MPC is employed to perform MLA, no minimum load threshold specification
is necessary. Instead MPC exploits prediction of imminent constraint violations within a pre-defined prediction horizon, and computes the control action to prevent such violations.

While the load factor is a usual choice of a monitored parameter in MLA systems, this parameter only captures the transient behavior of loads during maneuvers. For instance, pitch-up maneuvers performed at a slow rate may develop small to moderate accelerations that do not trigger the MLA system. However, the aerodynamic forces that build up as the angle of attack gradually increases can result in significant structural loads.

An alternative approach to perform MLA is by monitoring the curvature at critical stations, since this parameter is directly connected to the magnitude of loads at those stations. Indeed, considering the wing modeled as an Euler-Bernoulli beam, the axial strain at a spanwise station located at distance $y$ and deflected by $z$ from its neutral position is given by

$$\epsilon_{yy} = -\kappa_x z,$$

where $\kappa_x$ is the curvature about the $x$ axis. The resultant bending moment $M_x$ is giving by:

$$M_x = EI_{xx} \kappa_x,$$

where $EI_{xx}$ is the out-of-plane bending stiffness at wing station $y$.

In this paper, as in [11], the following control requirements for an automatic flight control system with MLA capability are considered:

a) Track defined command signals and meet response specifications while ensuring stability during operation;

b) Keep loads at critical stations within pre-defined limits;

c) Minimize the effect of the MLA system on the nominal trajectory of the aircraft.

The MPC-based MLA governor proposed in this paper performs MLA by enforcing curvature constraints at critical stations while meeting the above requirements. Note that the MLA governor can handle multiple constraints at various stations, including wing, fuselage and tail.

3 CONTROL SYSTEM DESIGN FOR MLA

In this paper, the control objective is to perform a prescribed maneuver by tracking command signals while enforcing input and output constraints to perform MLA.

The continuous-time linearized model around a defined flight condition is given by

$$\begin{bmatrix} \dot{x}_F \\ \dot{x}_R \end{bmatrix} = \begin{bmatrix} A_{FF} & A_{FR} \\ A_{RF} & A_{RR} \end{bmatrix} \begin{bmatrix} x_F \\ x_R \end{bmatrix} + \begin{bmatrix} B_F \\ B_R \end{bmatrix} u = Ax + Bu,$$

$$y = Cx + Du,$$

where $x = [x_F^T \ x_R^T]^T \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input vector, $y \in \mathbb{R}^{n_y}$ is the output vector, and subscripts $F$ and $R$ stand for flexible and rigid, respectively. The control design in the next sections will be based on discrete-time models with the subscript $d$ designating such models (e.g.: $B_d$ is the discrete-time input matrix). For design purposes, the input vector is divided into control effectors, $u_a$, that manipulate the aircraft attitude, control effectors, $u_T$, that manipulate the total thrust, and control effectors, $u_{MLA}$, used to perform MLA:

$$u = [u_a^T \ u_T^T \ u_{MLA}^T]^T.$$
A subset of the outputs, \( y^t = C^t x + D^t u, \) \( C^t \in \mathbb{R}^{n_y \times n_x}, \) \( D^t \in \mathbb{R}^{n_y \times n_u}, \) \( n_r \leq n_y, \) is to be controlled to set-points. The vector of set-points is denoted by \( r \in \mathbb{R}^{n_r}. \) In this work, the components of \( r \) are target values (in terms of deviations from the trim values) for aircraft longitudinal velocity, flight path angle, side-slip angle, and heading angle, i.e.,

\[
r = \begin{bmatrix} r_V & r_\gamma & r_\beta & r_\psi \end{bmatrix}^T.
\]

The tracking error is defined as

\[
e = y^t - r.
\]

### 3.1 Nominal controller

A nominal discrete-time controller is designed for output tracking. This controller is divided into two LQ-I controllers that independently manipulate control effectors \( u_a \) and \( u_T \) for attitude and velocity tracking, respectively. The control signals, \( u_{MLA} \), are assumed to be zero at this design step, i.e., the corresponding control effectors are assumed to be at the trim values. Figure 1 shows a schematic of the implementation, in which the output \( u_I \) is the concatenation of vectors \( u_a \) and \( u_T \), i.e.,

\[
u_I = \begin{bmatrix} u_a^T & u_T^T \end{bmatrix}^T.
\]

The rigid body part of the linear model in Eq. 3 is used to design the nominal controller. To obtain the residualized linear model for the rigid-body dynamics with states \( x_R \in \mathbb{R}^{12} \), the flexible states are assumed to be in steady-state, yielding:

\[
\dot{x}_R = (A_{RR} - A_{RF}A_{FF}^{-1}A_{FR}) x_R + (B_R - A_{RF}A_{FF}^{-1}B_F) u = \bar{A}_R x_R + \bar{B}_R u.
\]

The model in Eq. 9 is then discretized using zero-order hold with the sampling period \( T_{inner} \) to obtain the discrete-time model with matrices \( \bar{A}_Rd \) and \( \bar{B}_Rd \). The control law design for tracking \( r_\gamma, r_\beta \) and \( r_\psi \) is shown next. Similar derivations are performed to obtain the gains for the LQ-I controller responsible for tracking velocity set-point, \( r_V \), by adjusting \( u_T \) (the details are skipped). Let \( (u_a)_{k} \) be the control effectors manipulated by the controller at instant \( k \). The LQ-I control design for \( (u_a)_{k} \) minimizes the following cost function,

\[
J = \sum_{k=0}^{\infty}(e_{ak}^T Q e_{ak} + (\Delta u_a)_k^T R_z (\Delta u_a)_k) = \sum_{k=0}^{\infty}(z_k^T Q z_k + (\Delta u_a)_k^T R_z (\Delta u_a)_k),
\]

where \( (\Delta u_a)_k = (u_a)_{k+1} - (u_a)_k \), \( Q_z \) and \( R_z \) are symmetric positive definite weighting matrices, \( e_{ak} \) is the attitude tracking error, and

\[
Q_z \triangleq \begin{bmatrix} 0 & 0 \\ 0 & Q_e \end{bmatrix}.
\]
The vector \( z_k \) is the augmented state vector defined as
\[
    z_k \triangleq \begin{bmatrix} \Delta x_{R_k}^T & e_{a_k}^T \end{bmatrix}^T,
\]
where \( \Delta x_{R_k} = x_{R_k+1} - x_{R_k} \). Hence, the augmented model is given by
\[
    z_{k+1} = \begin{bmatrix} \hat{A}_{R_k} & 0 \\ \hat{C}_{a_k} & I \end{bmatrix} z_k + \begin{bmatrix} [\hat{B}_{R_k}]_a \\ [D^i]_a \end{bmatrix} (\Delta u_a)_k = A_z z_k + B_z (\Delta u_a)_k.
\] (13)

The notation \([\cdot]_a\) in Eq. 13 means the selection of the matrix columns and/or rows that are connected to the tracked attitude variables and/or the control effectors \( u_a \). The solution of this optimal control problem has the form:
\[
    (\Delta u_a)_k = - (R_z + B_z^T P_z B_z)^{-1} B_z^T P_z A_z z_k = K_1 \Delta x_k + K_2 e_k,
\] (14)
where \( P_z = P_z^T \) is the solution of the algebraic Riccati equation,
\[
    P_z = A_z^T P_z A_z - (A_z^T P_z B_z)(R_z + B_z^T P_z B_z)^{-1}(B_z^T P_z A_z) + Q_z,
\] (15)
and \( K_1, K_2 \) are the resulting control gains.

From Eq. 14, the controller update equation has the form,
\[
    (u_a)_{k+1} = (u_a)_k + K_1 (x_{k+1} - x_k) + K_2 ((y_a^1)_k - r).
\] (16)

Eq. 16 shows that the control law includes an integral action, which allows the tracking of commands with zero steady-state offset. In the final implementation, the states and outputs, \( x_k \) and \( (y_a^1)_k \), respectively, in the control law in Eq. 16 will be replaced by their estimated values \( \hat{x}_k \) and \( \hat{y}_k \), respectively, provided by a state observer (see Section 3.4).

### 3.2 Baseline for MLA

The classical MLA logic based on acceleration signals is as follows [15]. Let \( \ddot{z}(t) \) be the vertical acceleration at the time instant \( t \), \( t \in \{0, 1, 2, \ldots\} \), at which new acceleration measurements are available. The corresponding vertical load factor is given by
\[
    n_z(t) = 1 + \frac{\ddot{z}(t)}{g},
\] (17)
where \( g \) is the acceleration due to gravity. Assume, for the purpose of this discussion, that \( n_z(t) > 0 \). Let \( n_z^* \) be a predetermined threshold (e.g., 2\( g \)), and \( n_z^{\text{max}} > n_z^* \) denotes the maximum admissible vertical load factor (e.g., 2.5\( g \)). Let \( u_{\text{MLA}}(t) \) be the deflection of the control effector assigned to perform MLA, such that \( u_{\text{MLA}}(t) = u_{\text{MLA}}^0 - k(t) u_{\text{MLA}}^\text{max} \), where \( u_{\text{MLA}}^0 \) is the nominal deflection (e.g., the trim value or the deflection needed to perform a maneuver), \( u_{\text{MLA}}^\text{max} \) is the maximum control surface deflection allowed for MLA purposes, and \( 0 \leq k(t) \leq 1 \) is a gain.

Define
\[
    \sigma(t) \triangleq \max \left\{ 0, \min \left\{ 1, \frac{n_z(t) - n_z^*}{n_z^{\text{max}} - n_z^*} \right\} \right\}.
\] (18)

Then, the following rules are used to determine \( \kappa(t) \) and the resulting MLA action:
1. If \( n_z(t) \leq n_z^* \), then \( k(t) = 0 \);
2. If \( n_z(t) \geq n_z^{\text{max}} \), then \( k(t) = 1 \);
3. If \( n_z(t) > n_z^* \), and \( n_z(t) \geq n_z(t-1) \), then \( k(t) = \sigma(t) \);
4. If \( n_z(t) > n_z^* \), and \( n_z(t) < n_z(t-1) \), and \( \sigma(t) \leq 0.1 \), then \( k(t) = \sigma(t) \);
5. If \( n_z(t) > n_z^* \), and \( n_z(t) < n_z(t-1) \), and \( \sigma(t) > 0.1 \), and \( |\sigma(t) - k(t-1)| \leq 0.2 \), then \( k(t) = k(t-1) \);
6. If \( n_z(t) > n_z^* \), and \( n_z(t) < n_z(t-1) \), and \( \sigma(t) > 0.1 \), and \( |\sigma(t) - k(t-1)| > 0.2 \), then \( k(t) = \sigma(t) \).

In rules 4 and 5 the trend of the load factor to decrease is confirmed and a hysteresis with incremental dead-zone is implemented, so as to not adversely affect the stability of the aircraft. The MLA system is only activated when one of antecedents of the rules 2-6 holds true and the aircraft is in flight, in a cruise configuration (leading edges and wing flaps retracted), and the control stick is deflected beyond a certain angular threshold for confirming the maneuver (e.g. 8 deg in a nose-up direction). Based on the MLA system described in [15], the vertical acceleration measurements are obtained from accelerometers placed at the the front part of the fuselage.

In this work, this traditional MLA system will serve as a baseline to evaluate the performance of the MLA governor that will be presented in Section 3.3. Since the control design objective is to track a command signal and perform load alleviation, the baseline MLA system will be implemented along with the nominal controller presented in Section 3.1. Figure 2 shows the resulting control system with a state observer.

3.3 MLA governor

The nominal controller in Section 3.1 is augmented with the MLA governor in order to enforce state, output and control constraints. In the architecture shown in Fig. 3, the MLA governor plays the role of a reference governor for the inner loop, in addition to manipulating the extra degrees of freedom provided by \( u_{MLA} \) to perform MLA. Appealing features of this architecture include the preservation of the nominal controller in the inner loop, which can be advantageous from a design perspective, since the conventional flight control system is augmented rather than replaced. Moreover, the performance and stability of the vehicle are preserved even if the outer loop has to be deactivated [11].

The MLA governor modifies the references \( r^c = [r^c_{\dot{V}} \ r^c_{\dot{\gamma}} \ r^c_{\dot{\beta}} \ r^c_{\dot{\psi}}]^T \) to be fed into the inner loop and \( u_{MLA} \) to avoid constraint violations. The modifications are kept to a minimum, so that \( r^c = r \) and \( u_{MLA} = 0 \) whenever there is no danger of constraint violation.

To design the MLA governor, a discrete-time linear system model representative of the inner-
loop system consisting of LQ-I controllers and the plant is used which has the following form,

\[
\begin{align*}
    x_{k+1} &= A_r x_k + B_r r^c_k, \\
    y_k &= C_r x_k.
\end{align*}
\] (19)

This continuous-time model is re-sampled assuming an update period \(T_{MPC}\), which is typically larger than the update period of the inner loop controller, i.e., \(T_{MPC} > T_{inner}\). To include the control effectors \(u_{MLA}\) as control inputs, columns are added to the control matrix of the inner-loop system so that the discrete-time model becomes:

\[
\begin{align*}
    x_{k+1} &= \begin{bmatrix} A_r & 0 \\ B_r & B_{dMLA} \end{bmatrix} \begin{bmatrix} r^c_k \\ u_{MLA} \end{bmatrix} = A_r x_k + B_r u_{r,k}.
\end{align*}
\] (20)

where \(B_{dMLA}\) are the columns of \(B_d\) in Eq. 3 corresponding to \(u_{MLA}\). Note that since the MLA governor is applied to the inner-loop system, control constraints can be reformulated as equivalent constraints on the states of the plant, states of the inner loop controller, \(u_{MLA}\) and references.

For the augmented state vector, \(x_{k}^{aug} = [x_k^T \quad (u_{r,k})^T \quad r^T]^T\), the discrete-time linear system model is given by

\[
\begin{align*}
    x_{k+1}^{aug} &= \begin{bmatrix} A_r & \bar{B}_r \\ 0 & I \end{bmatrix} x_k^{aug} + \begin{bmatrix} 0 \\ I \end{bmatrix} \Delta u_{r,k} = A_r x_k^{aug} + B_r \Delta u_{r,k},
\end{align*}
\] (21)

where \(\Delta u_{r,k} = u_{r,k+1} - u_{r,k}\). The MLA governor is now defined based on the solution to the following constrained optimization problem,

\[
\begin{align*}
    \text{minimize} \quad J_N &= \sum_{k=0}^{N-1} ||r^c_k - r||_Q^2 + ||\Delta u_{r,k}||_R^2 + ||(u_{MLA})_k||_{R_{MLA}}^2 + \mu \varepsilon_{k+1}^2, \\
    \text{subject to:} \\
    x_{k+1}^{aug} &= A_r x_k^{aug} + B_r \Delta u_{r,k}, \\
    x_0^{aug} &= \text{current augmented state}, \\
    h(x_{k+1}^{aug}, (u_{MLA})_k) &\leq \varepsilon, \quad \text{for } k = 0, \ldots, N - 1 \\
    \varepsilon &\geq 0,
\end{align*}
\] (22)
where $Q_r, R_r$ and $R_{MLA}$ are positive definite weighting matrices, $\mu > 0$ is a scalar weight, and $h(\cdot)$ represents $n_c$ inequality constraints on the states, inputs and outputs. The term $\varepsilon$ is a slack variable used to relax the constraints on states and outputs if necessary so that it is always possible to guarantee feasibility of the optimization problem. As feasibility is trivially enforced on input constraints, they are treated as hard constraints. If not all plant states in the vector $x^a_{0}$ are available from sensor measurements, then these are replaced by their estimated values provided by a state observer (see Section 3.4).

Note that the cost function in Eq. 22 has a special form which promotes MLA system deactivation ($r = r^*_k, u_{MLA} = 0$) whenever feasible under the imposed constraints. In particular, the penalty on $u_{MLA}$ ensures that the control effectors used for MLA, usually ailerons and/or elevators, return back to the trim settings after performing load alleviation. This is desired to minimize drag and deviation from the nominal trajectory when the MLA system is deactivated.

If the constraints in Eq. 23 are linear, then the optimization problem is a quadratic program (QP). This QP can be reformulated in the condensed form [16] to eliminate the states $\{x^a_{k+1}\}_{k=0}^{N-1}$ from the set of decision variables since they are linked to the input sequence $\{\Delta u_{r}^*\}_{k=0}^{N-1}$ by the linear system in Eq. 21. The reformulated QP can be represented in the standard form:

$$\min_{U} \quad J = \frac{1}{2} U^T H U + q^T U$$
$$\text{s.t.} \quad G U \leq W$$

where $U$ is the vector of decision variables:

$$U = \begin{bmatrix} (\Delta u^*_{0})^T & \ldots & (\Delta u^*_{N-1})^T & \varepsilon_1 & \ldots & \varepsilon_N \end{bmatrix}^T \in \mathbb{R}^{(n_u+1)N},$$

and $H \in \mathbb{R}^{(n_u+1)N \times (n_u+1)N}, q \in \mathbb{R}^{(n_u+1)N}, G \in \mathbb{R}^{n_cN \times (n_u+1)N}$ and $W \in \mathbb{R}^{n_cN}$ are matrices and vectors (see [16] for definitions).

The MLA governor output is defined by the first element, $\Delta u^*_{0}$, i.e., the reference command provided to the inner-loop and the deflection of the control effectors $u_{MLA}$ have the form:

$$\begin{bmatrix} r^c \\ u_{MLA} \end{bmatrix}_{\text{MPC}} = u^*_0 + \Delta u^*_{0} (x^a_{0}, u^*_0),$$

where $u^*_0$ denotes the previous value of $u^*$.

Any standard QP solver can, in principle, be used to solve the MPC problem at each sampling time. However, due to the large number of states, inputs and constraints in large scale systems such as flexible and very flexible aircraft, the computational cost can be prohibitive. The next subsections present some strategies to reduce the computational footprint of the MPC implementation.

### 3.3.1 Model order reduction

To reduce the number of states, model order reduction applied to system in Eq. 3 is considered. When performing model order reduction one seeks to represent a system by a lower dimensional model that can still capture the dominant characteristics of the system response. Towards this end, balanced truncation is used [17]. The reduced order model is then used for prediction by the MLA governor.
Furthermore, the use of ROMs obtained through balanced truncation is also a work-around for controllability and observability issues that may exist in the original linear model. By performing balanced truncation, one obtains a model that is observable and controllable, thus being suitable for observer and controller design.

### 3.3.2 Input blocking

To reduce the number of decision variables in the optimization problem, a common practice in MPC design is to use move blocking. With move blocking, groups of adjacent-in-time predicted inputs are constrained to the the same value, so that the number of degrees of freedom of the predicted input actions is reduced. Move blocking results in a modified QP problem of the form,

\[
\min_{\bar{U}} \quad \bar{J} = \frac{1}{2} \bar{U}^T \bar{H} \bar{U} + \bar{q}^T \bar{U} \\
\text{s.t.} \quad \bar{G} \bar{U} \leq W
\]

where \( \bar{H} \triangleq V^T HV, \quad \bar{q} \triangleq V^T q, \quad \bar{G} \triangleq GV, \quad V = \begin{bmatrix} T_k \otimes I_u & 0 \\ 0 & I_N \end{bmatrix}, \)

\( U = V \bar{U} \) and the matrix \( T_k \in \mathbb{R}^{N \times M} \) is the blocking matrix, with \( M, 1 \leq M < N \) designating the number of degrees of freedom given to the predicted inputs.

The blocking matrix can be designed in a number of different ways [18]. Here the input blocking method is used, in which inputs are fixed to be constant over \( t \geq k + M \) time-steps within the prediction horizon. Hence, the blocking matrix is given by:

\[
T_k = \begin{bmatrix} I_M \\ [0_{(N-M) \times (M-1)} & 1_{N-M}] \end{bmatrix}.
\]

### 3.3.3 Constraint aggregation

To reduce the number of constraints and consequently the computational cost, the Kreisselmeier-Steinhauser (KS) function is used for constraint aggregation into a smaller number of constraints. Such constraint aggregation using KS functions has been previously used in multidisciplinary design optimization [19–21]. The KS function is given by:

\[
KS_j(g_i(\bar{U}), \rho_j) = g_{\max}(\bar{U}) + \frac{1}{\rho_j} \ln \left( \sum_{i \in N_{g_j}} e^{\rho_j(g_i(\bar{U}) - g_{\max}(\bar{U}))} \right),
\]

where \( \rho_j \) is the aggregation parameter,

\[
g_i(\bar{U}) = [\bar{G}]_i \bar{U} - [W]_i, \quad g_{\max}(\bar{U}) = \max_{i \in N_{g_j}} \{ g_i(\bar{U}) \},
\]

and where \([\bar{G}]_i\) and \([W]_i\) represent the \( i \)th row of \( \bar{G} \) and \( \bar{W} \), respectively. The set \( N_{g_j} \) contains the indices of constraints that are aggregated into a single KS function. Some properties of the KS function are as follows [22]:

1. \( KS_j(g_i(\bar{U}), \rho_j) \geq g_{\max}(\bar{U}) \) for \( \rho_j > 0 \)
2. \( \lim_{\rho_j \to \infty} KS_j(g_i(\bar{U}), \rho_j) = g_{\max}(\bar{U}) \)
3. $KS_2(g_i(U), \rho_2) \geq KS_1(g_i(U), \rho_1)$ for $\rho_1 > \rho_2 > 0$
4. $KS_j(g_i(U), \rho_j)$ is convex if and only if all constraints $g_i(U)$ are convex
5. $g_{\text{max}}(U) \leq KS_j(g_i(U), \rho_j) \leq g_{\text{max}}(U) + \ln|N_{gj}|/\rho$.

In particular, note that Property 1 shows that the approximation of the feasible set is conservative; Property 4 shows that the aggregation preserves convexity; and Property 5 shows that the approximation error is bounded. These properties make the application of KS functions appealing for linear MPC problems.

The aggregation is performed over the prediction horizon, from $k = 0$ (or $k = 1$) to $N$ steps into the future, for each constrained input and output channel. This allows the selection of the appropriate $\rho$ for each aggregation according to the order of magnitude of each constrained variable. Further reduction in the number of constraints can be achieved by performing a second aggregation over constrained channels of similar nature (e.g.: constraints on the right and left ailerons can be aggregated together), which leads to

$$KS_l(KS_j(g_i(U), \rho_j), \rho_l) = KS_{\text{max}}(U) + 1/\rho_l \ln \left[ \sum_{j \in N_{KS}} e^{\rho_l(KS_j(g_i(U), \rho_j) - KS_{\text{max}}(U))} \right],$$

where

$$KS_{\text{max}}(U) = \max_{j \in N_{KS}} \{ KS_j(g_i(U), \rho_j) \},$$

and $N_{KS}$ is the set of indices for the KS functions to be aggregated together.

The resulting optimization problem with aggregated constraints has the following form:

$$\min_{\tilde{U}} \quad \tilde{J} = \frac{1}{2} \tilde{U}^T \tilde{H} \tilde{U} + \tilde{q}^T \tilde{U}$$

s.t. $$\begin{bmatrix} [KS_j(g_i(U), \rho_j)]_{n_s}^{n_s} & [KS_l(KS_j(g_i(U), \rho_j), \rho_l)]_{n_d}^{n_d} \end{bmatrix} \leq 0,$$

where $n_s$ and $n_d$ are the number of single and double aggregation functions, respectively. Note that the optimization problem in Eq. (36) is not a QP anymore. Sequential quadratic programming (SQP) is used to solve it. Multiple strategies can be exploited to implement SQP, which will lead to significant reduction in the computational cost when the constraint aggregation is used [13].

### 3.4 Observer design

In an actual application, the states $x_F$ and some components of the states $x_R$ in Eq. 3 may not be directly measured by sensors. In this work, a discrete-time Luenberber observer is used to estimate them. Let $y_k$ be a vector of measured outputs at discrete time instant $k$. Assuming that the pair $(A, C)$ is observable, the observer equations are:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k),$$

$$\hat{y}_k = C\hat{x}_k + Du_k,$$

where $L$ is chosen so that the matrix $A - LC$ is Schur. The sampling time for the observer updates is chosen to be the same as for the inner loop controller, i.e., $T_{\text{inner}}$. Since the controller is designed based on a ROM, the observer can also be designed using the same ROM.
Inertial navigation systems (INSs) and inertial measurement units (IMUs) are chosen to provide measurements at various stations along the fuselage, wing and tail. IMUs provide measurements of accelerations, angular rates and Euler angles. INSs provide the same measurements in addition to velocity and inertial position. Pitot probes and alpha vanes are also sensors commonly used to obtain measurements of velocity, angle of attack and side-slip angle. The use of strain gauges are not recommended due to lack of reliability in large industrial applications. In this work, the following vector of measurements is considered to be available at each discrete-time $k$:

$$y_k = \begin{bmatrix} V & \alpha & \beta & \gamma & (y_{\text{INS}}^{n_{\text{ins}}})^T & (y_{\text{IMU}}^{n_{\text{imu}}})^T \end{bmatrix}^T,$$

where $y_{\text{INS}}^{n_{\text{ins}}}$ represents INS measurements at $n_{\text{ins}}$ stations, and $y_{\text{IMU}}^{n_{\text{imu}}}$ represents IMU measurements at $n_{\text{imu}}$ stations.

4 AIRCRAFT MODELS

To illustrate the MLA systems presented in Section 3, linear models for flexible aircraft with different levels of flexibility are used which represent the XRF1 aircraft. The XRF1 is an Airbus provided industrial standard multi-disciplinary research testcase representing a typical configuration for a long range wide body aircraft. The XRF1 research testcase is used by Airbus to engage with external partners on development and demonstration of relevant capabilities/technologies. A front view of the aircraft is shown in Fig. 4.

The fuselage, wing and tail structures of XRF1 are flexible, but the level of flexibility falls within a range where the flexible and rigid body dynamics can still be decoupled for control design. Hence this baseline XRF1 model is representative of a flexible aircraft and is used for the numerical simulation in Section 5. To create a model that represents a flexible aircraft with increased flexibility, the Airbus-Michigan Center for Aero-Servo-Elasticity of Very Flexible Aircraft (CASE-VFA) modified the baseline XRF1 to create XRF1-HARW, a model for a future high-aspect-ratio-wing commercial transport vehicle. XRF1-HARW has the same geometry and properties as XRF1, but with a wing 20% longer, as shown in Fig. 4. Therefore, the nonlinear effects become more pronounced, and the flexible and rigid body responses can have frequencies of similar magnitude. In fact, the first out-of-plane bending moment of XRF1-HARW is 40.26% smaller than the same frequency of the baseline XRF1.

Both XRF1 and XRF1-HARW were modeled using the University of Michigan’s Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) [23]. It employs geometrically nonlinear strain based finite elements, different options for steady and unsteady aerodynamics, and nonlinear
6-degree of freedom (DOF) rigid body equations of motion to numerically simulate the dynamics of the aircraft. The modeling, trimming and linearization of the aircraft dynamics were performed within the UM/NAST environment. Nonlinear closed-loop dynamic simulations can also be performed in UM/NAST, which is left as future work.

Both models have nine control effectors available for control design, namely, outer \((\delta_{oal}, \delta_{oar})\) and inner \((\delta_{ial}, \delta_{iar})\) ailerons at each semi-wing, elevators \((\delta_{el}, \delta_{er})\) at the left and right horizontal tail planes, a rudder \((\delta_r)\) at the vertical tail plane, and left and right \((\delta_{Tl}, \delta_{Tr})\) point forces acting as thrust. For control design purposes, the degrees of freedom of the control effectors are reduced to six by imposing the elevators to deflect symmetrically \((\delta_{el} = \delta_{er} = \delta_e)\), the inner ailerons to deflect anti-symmetrically \((\delta_{ial} = -\delta_{iar} = \delta_{ia})\), and symmetric thrust \((\delta_{Tl} = \delta_{Tr} = \delta_T)\). The outer ailerons work independently as elevons, and they are the main surfaces assigned to perform MLA since they have the highest control authority over the wing’s geometry. To be consistent with the notation introduced in Section 3, the following input vectors are defined:

\[
\begin{align*}
  u_{MLA} &= \begin{bmatrix} \delta_{oal} & \delta_{oar} \end{bmatrix}^T, \\
  u_T &= \delta_T, \\
  u_a &= \begin{bmatrix} \delta_{ia} & \delta_e & \delta_r \end{bmatrix}^T.
\end{align*}
\] (41)

To design the MLA systems described in Section 3, the stations monitored during the aircraft operation have to be selected. Among the stations along the XRF1 wing, tail and fuselage, some of them develop higher loads during maneuvers. To identify the critical stations with respect to the out-of-plane bending moment, a series of open-loop simulations was performed. In each simulation, a doublet signal was input in one of the control channels and the loads, moments and load factor at each station were monitored. The stations shown in Fig. 5a were identified as the critical station with respect to out-of-plane bending moment.

To design the state observer, one INS and twenty-four IMUs for XRF1 (twenty-six IMUs for the XRF1-HARW) were placed on the aircraft fuselage, wing, tail and pylons, as shown in Fig. 5. The sensors were evenly distributed on these structures and the number of IMU sensors was selected as the smallest one that made the ROM fully observable. The nonlinear kinematics of sensor attached to a flexible structure were implemented in UM/NAST [24] and linearized around a trim condition to obtain the \(C\) and \(D\) matrices in Eq. 39.

5 SIMULATION RESULTS

In this section, the results of the numerical simulations of the baseline MLA system and the MLA governor architecture proposed in Section 3 are presented. Both XRF1 and XRF1-HARW aircraft models were trimmed and linearized at their typical cruise condition. The full-order models of XRF1 and XRF1-HARW had 1020 and 1084 states, respectively. After performing balanced truncation, the linear models were reduced to 310 and 356 states, respectively. These were the ROM’s with the fewest number of states that still provided a good match between the output of the original model and of the ROM’s.

For the baseline MLA system described in Section 3.2, various combinations of parameters \(n_z^s\) (ranging from 1.5 to 2.5 in increments of 0.3), \(n_{z\text{max}}\) (ranging from 2.0 to 4.5 in increments of 0.2), and \(u_{MLA}\) (ranging from 5 to 30 deg in increments of 5 deg) were tested. The results shown in this section correspond to the combination that provided the best performance in terms of load alleviation at critical stations: the threshold for the activation of the system is set to \(n_z^s = 1.5\), and the maximum load factor and deflection of the outer ailerons are set to \(n_{z\text{max}} = 4.0\) and \(u_{MLA} = 20\) deg for XRF1, respectively. The vertical load factor signal used by the baseline
MLA system is filtered by a second-order low-pass filter. This is performed to remove the high frequency response of the plant, probably originated from the flexible dynamics of the structure.

The sampling rate used by the MLA governor is 50 Hz and the prediction horizon is 100 steps. This means that the MLA governor looks 2 seconds ahead of the current time to predict constraint violations and then determine the optimal control action. The LQ-I controller in the baseline system and in the MLA governor architecture, and the Luenberger observer, are all updated at sampling rate of 200 Hz.

The constraints imposed on the control problem for the MLA governor architecture are shown in Table 1. The constraints on the out-of-plane curvature on the wing and horizontal tail plane are defined according to the MLA objective discussed in Section 2. In addition to these constraints, limits on the angle of attack, and on control effector values and rates are imposed.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa_x$, XRF1 wing</td>
<td>Out-of-plane curvature at critical station on wing root</td>
<td>$-3.0 \times 10^{-4}$</td>
<td>$3.0 \times 10^{-4}$</td>
<td>[1/m]</td>
</tr>
<tr>
<td>$\kappa_x$, XRF1-HARW wing</td>
<td>Out-of-plane curvature at critical station on wing root</td>
<td>$-1.8 \times 10^{-3}$</td>
<td>$1.8 \times 10^{-3}$</td>
<td>[1/m]</td>
</tr>
<tr>
<td>$\kappa_x$, horizontal tail</td>
<td>Out-of-plane curvature at critical station on wing</td>
<td>$-5.0 \times 10^{-4}$</td>
<td>$5.0 \times 10^{-4}$</td>
<td>[1/m]</td>
</tr>
<tr>
<td>$\delta_{oal}, \delta_{oar}, \delta_{ia}, \delta_e, \delta_r$</td>
<td>Control surfaces deflection</td>
<td>$-30$</td>
<td>$30$</td>
<td>[deg]</td>
</tr>
<tr>
<td>$\delta_T$</td>
<td>Thrust</td>
<td>0</td>
<td>60000</td>
<td>[N]</td>
</tr>
<tr>
<td>$\Delta \delta_{oal}, \Delta \delta_{oar}, \Delta \delta_{ia}, \Delta \delta_e, \Delta \delta_r$</td>
<td>Rate of deflection of control surfaces in 1 s</td>
<td>$-45$</td>
<td>$45$</td>
<td>[deg]</td>
</tr>
<tr>
<td>$\Delta \delta_T$</td>
<td>Rate of variation of thrust in 1 s</td>
<td>$-1000$</td>
<td>$1000$</td>
<td>[N]</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Angle of attack</td>
<td>$-10$</td>
<td>10</td>
<td>[deg]</td>
</tr>
</tbody>
</table>
The final QP had 700 decision variables and 3900 constraints. Following the input blocking (Section 3.3.2) with $M = 50$ and the constraint aggregation procedures (Section 3.3.3), the problem was reformulated with 400 decision variables and 39 constraints. The SQP was then implemented to solve the nonlinear program. The solver QPKWIK [25], which is an active set optimization algorithm, was used to solve each QP sub-problem. In all simulations, the solver was warm-started.

In the linear simulations presented next, the aircraft starts at the trim condition and then a pitch up maneuver is commanded. The objective is to reach a flight path angle of $r_\gamma = 4$ deg while keeping the heading angle, side-slip angle and velocity at the trim values.

Figure 6 shows the out-of-plane curvature at the right wing and horizontal tail plane roots for the XRF1 and XRF1-HARW aircraft. Figure 7 shows the out-of-plane curvature at the most outboard critical station on the right wing. The results shown correspond to simulations without the MLA system (i.e.: with only the nominal controller to track signals), simulations with the baseline MLA system, and with the MLA governor. Similar results were obtained at the other critical stations on the right and left wing, and are therefore omitted for sake of space. Note that the baseline MLA system provided a small percentage load alleviation on the wing, especially for XRF1-HARW, and no load alleviation on the tail. In fact, the baseline system has no means of reducing the loads at the critical station on the horizontal tail plane, since it does not manipulate control effectors at the tail. On the other hand, the MLA governor was successful in enforcing constraints on the out-of-plane curvature in both models. Note also that even higher percentages of load alleviation could be achieved by tightening the constraints, but that could degrade the tracking performance. Figure 7 shows that the baseline MLA system is more effective in reducing loads at outboard stations, while the MLA governor had good performance in reducing loads at both inboard and outboard stations.

Figure 8 shows the flight path angle tracking performance. The baseline system preserved the flight path angle response, while the response provided by the system with the MLA governor was delayed, in terms of rise time, by approximately 1.0 s for XRF1, and 1.5 s for XRF1-HARW. Figure 8 also shows the reference $r_\gamma^*$ generated by the MLA governor. Note that this reference signal was modified in order to enforce constraints on both wing and tail critical stations.

Figure 6: Out-of-plane curvature and bending moment at wing and horizontal tail plane root.

(a) XRF1 flexible aircraft.  
(b) XRF1-HARW flexible aircraft with increased flexibility.

Figure 8 shows the flight path angle tracking performance. The baseline system preserved the flight path angle response, while the response provided by the system with the MLA governor was delayed, in terms of rise time, by approximately 1.0 s for XRF1, and 1.5 s for XRF1-HARW. Figure 8 also shows the reference $r_\gamma^*$ generated by the MLA governor. Note that this reference signal was modified in order to enforce constraints on both wing and tail critical stations.
Figure 7: Out-of-plane curvature and bending moment at the most outboard critical station on the wing.

Figure 8: Flight path angle tracking.

Figure 9 shows the deviation from the nominal trajectory at each inertial position axis. The nominal trajectory is defined as the trajectory generated by the nominal controller alone, with no MLA system. The baseline MLA system had a better performance in maintaining the trajectory close to the nominal one. However, both controllers generated deviations that are very small when compared to the aircraft size and flight altitude.

The control effectors time histories generated to follow the commanded reference signal and perform MLA are shown in Fig. 10. The control actuation was similar for the three control architectures tested, except for the deflection of the outer elevons used for MLA. The nominal controller does not manipulate these actuators, and therefore they remain at the trim condition when the MLA system is not used. The elevons were only deflected when either the baseline MLA system or MLA governor were triggered to perform MLA. After performing the load alleviation, these control surfaces returned to the trim condition. Note that even though the
baseline system resulted in higher deflections of the elevons, especially in the XRF1-HARW model, this did not reflect as an increased percentage of load alleviation.

6 CONCLUSIONS

A predictive control strategy, referred to as an MLA governor, for Maneuver Load Alleviation (MLA) applicable to flexible commercial aircraft with different levels of flexibility was presented. The MLA governor augments rather than replaces the nominal controller and achieves load alleviation by enforcing constraints imposed on the out-of-plane curvature at various stations on the wing and horizontal tail plane. Different strategies were implemented to reduce the computational footprint of the MPC controller including a technique that reduces the number of constraints by performing aggregation.

The proposed MLA governor was evaluated via linear simulations with models representing flexible airliners. The MLA governor performance was compared against a typical design used in industry. The MPC-based MLA governor presented better performance in reducing maneuver loads at various stations simultaneously, while keeping the deviation from the nominal trajectory small.

Future research opportunities include the application of the proposed architecture on a nonlinear model for flexible and very flexible aircraft.
Figure 10: Control effectors time history.

7 ACKNOWLEDGEMENTS

The authors would like to thank Cristina Riso and Divya Sanghi (University of Michigan) for creating the XRF1 and XRF1-HARW aircraft models in UM/NAST. This material is based upon work supported by Airbus in the frame of the Airbus-Michigan Center for Aero-Servo-Elasticity of Very Flexible Aircraft (CASE-VFA).

8 REFERENCES


**COPYRIGHT STATEMENT**

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the IFASD-2019 proceedings or as individual off-prints from the proceedings.