A SLIDING-WINDOW MATRIX PENCIL METHOD FOR AEROELASTIC DESIGN OPTIMIZATION WITH LIMIT-CYCLE OSCILLATION CONSTRAINTS

A Thesis
Presented to
The Academic Faculty

By

Tarun Golla

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology

December 2023

A SLIDING-WINDOW MATRIX PENCIL METHOD FOR AEROELASTIC DESIGN OPTIMIZATION WITH LIMIT-CYCLE OSCILLATION CONSTRAINTS

Thesis committee:

Dr. Cristina Riso
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kai James
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Dr. Graeme J. Kennedy
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology

Date approved: 11/30/2023

I find that the harder I work, the more luck I seem to have.

Thomas Jefferson

ACKNOWLEDGMENTS

First and foremost, I would like to thank my research advisor, Dr. Cristina Riso, as without her tireless support and advice, this endeavor would not have been possible. I would additionally like to thank Dr. Graeme Kennedy and Dr. Kai James for their input in the direction and development of this project as it grew from a concept to an implementation.

I would like to make note of the impact of all my friends and family for inspiring me to pursue this path as it has been extremely rewarding academically and professionally. I would especially like to thank my parents for providing me with any and all resources that I have needed while always being in my corner.

Thank you to the Daniel Guggenheim School of Aerospace Engineering for awarding me a fellowship allowing me to conduct my research as a Graduate Research Assistant.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis presents a new approach to constraining limit-cycle oscillations (LCOs) in aeroelastic design optimization. LCOs are self-excited oscillations that can develop in nonlinear aeroelastic systems experiencing flutter, and they must be avoided during operation to keep safety and performance. One approach to addressing this problem is to design the system using an optimization process that includes an LCO constraint. Previous efforts have proposed various LCO constraints for aeroelastic design optimization but have not addressed realistic design applications. This gap persists because existing LCO constraints are not oriented toward scalable gradient-based optimization algorithms. The proposed approach builds on a recent LCO constraint that bounds the recovery rate to equilibrium and is suited to gradient-based optimization. The new contribution from this thesis consists of introducing a new matrix pencil method for accurately evaluating the recovery rate within the LCO constraint using output data from transient responses. The amplitude-varying behavior of the recovery rate in the presence of dynamic nonlinearities is captured using a sliding time window along the transient response for a chosen quantity of interest. This new approach differs from the conventional matrix pencil method, which considers an entire transient response at once under linearized assumptions. Sensitivity studies are conducted to identify the optimal singular-value decomposition tolerance, sliding window size, stride size, output data sampling step, and aggregation parameters for obtaining accurate results. The new sliding-window matrix pencil method is then used to optimize a typical aeroelastic section model with a subcritical LCO behavior, enforcing no flutter or LCOs at chosen operation conditions. Optimization results are compared with previous work that used the same LCO constraint formulation combined with an approximate, conservative method to evaluate the recovery rate. The LCO constraint evaluated using the new sliding-window matrix pencil method allows the optimizer to completely suppress subcritical LCOs within the specified operating conditions while minimizing design changes, achieving a less con-

servative optimized solution. This work is a step toward constraining LCOs in large-scale aeroelastic design optimization to enable higher-performance designs while avoiding undesirable dynamics, such as subcritical LCOs. Future work includes formulating adjoint derivatives of the LCO constraint and demonstrating the methodology for aeroelastic models of increasing physical and computational complexity.

CHAPTER 1
INTRODUCTION

1.1 Motivation

New wing designs favor increasingly lightweight structures and higher aspect ratios for enhanced aircraft fuel efficiency [1], requiring consideration of dynamic aeroelastic instabilities such as flutter earlier in the design process [2]. If flutter is found during testing, redesign steps are required to guarantee adequate flutter margins. This approach adds production costs and time delays and can potentially compromise performance through disadvantageous design tradeoffs. An alternative approach is to include an active flutter suppression system, which uses control inputs to prevent flutter onset or suppress diverging oscillations. This, however, can lead to major safety issues if the system fails, in addition to degraded aircraft maneuverability and consequent problems if the system does not function properly. With the potential for diverging oscillations to grow too fast for an active flutter suppression system to act as well, passive flutter prevention early in the design phase itself is critical. This problem has motivated research efforts to integrate flutter constraints into aircraft design optimization to produce fuel-efficient wing designs that are free from flutter [2].

To date, most flutter constraints for design optimization have been based on linear flutter analyses providing the aeroelastic eigenvalues of the wing in its undeformed shape at various flight conditions (e.g., Ref. [3]). In some cases, these analyses have included transonic effects relevant to commercial transport aircraft [4, 5, 6]. This approach provides feasible, flutter-free designs only if the wing experiences small deflections in flight [7]. To overcome this limitation, recent efforts have formulated geometrically nonlinear flutter constraints that consider the wing's statically deformed shape at each flight condition when computing the aeroelastic eigenvalues [8, 9, 10, 11, 12, 13]. However, capturing the impact of wing deflections in an eigenvalue-based flutter analysis process involves a local linearization of the governing equations about the nonlinear equilibrium solution at each flight condition, increasing the problem's complexity and computational cost.

Flutter constraints only capture dynamic behaviors for small disturbances about equilibrium states [2]. Nonlinear dynamics associated with large dynamic deflections, flow separation, moving shock waves, or other unsteady nonlinear phenomena may cause self-excited responses called limit-cycle oscillations (LCOs), which cannot be predicted by eigenvalue analyses [14]. Therefore, capturing dynamic nonlinear effects across a wide range of amplitudes is critical to properly characterize LCOs. These LCOs can manifest in two manners: supercritical and subcritical. Supercritical LCOs are "benign", as they only arise at linearly unstable conditions and their amplitude grows gradually with a control parameter (e.g., a parameter related to the flight condition, such as flow speed or dynamic pressure). These LCOs are avoidable by assessing the feasible flight envelope for a given design through an eigenvalue-based flutter analysis. For the same reason, supercritical LCOs are avoidable by adding a flutter constraint to a design optimization process.

Subcritical LCOs, instead, can develop even at linearly stable operating conditions if the system experiences sufficiently large disturbances. These dynamics have been identified through both experimental and numerical studies in aeroelastic systems [1]. For instance, flexible high-aspect-ratio wings may show subcritical behaviors at relatively low speeds due to the interplay between nonlinear effects related to the wing's structural response and the aerodynamic effects that come into play when the wing stalls (e.g., see Refs. [15, 16, 17]). Because subcritical LCOs can arise even at linearly stable conditions, they cannot be prevented by eigenvalue-based flutter analyses and constraints. Furthermore, when the system is optimized subject to a flutter constraint, the optimizer may produce a highly efficient, flutter-safe design with an unfeasible subcritical behavior (Figure 1.1). To avoid this issue, the optimization must include an LCO constraint [2]. This constraint is paramount

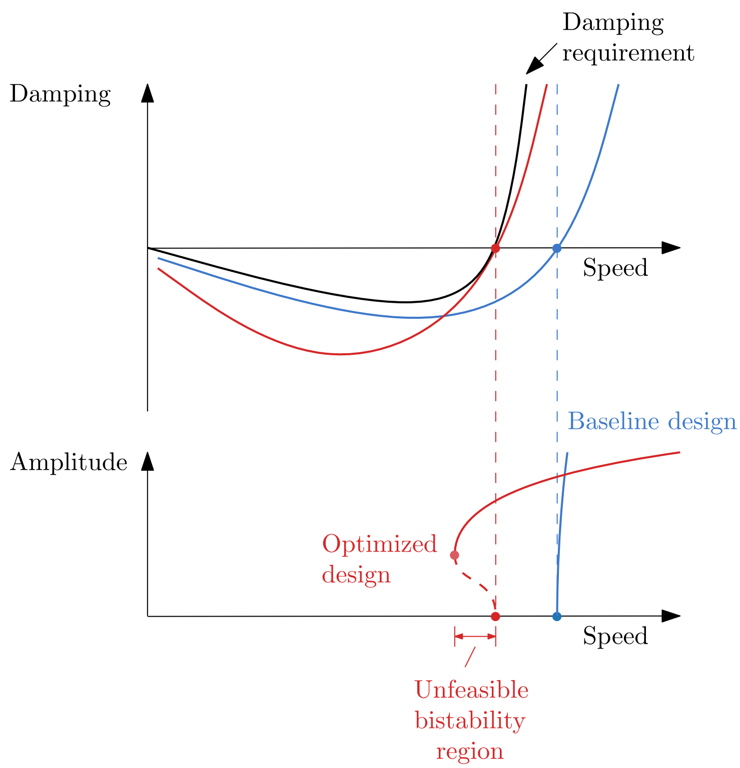


Figure 1.1: System optimized with a flutter constraint but no LCO constraint (Ref. [18]).

for aeroelastic systems showing nonlinear dynamics, such as future wings and aircraft.

1.2 Literature Review

LCO constraint formulation and implementation methodologies are being investigated using a variety of approaches [2]. Most of these approaches characterize LCOs through bifurcation diagrams, such as the one at the bottom of Figure 1.1. These diagrams are a general tool used to visualize the steady-state LCO amplitude for a given quantity of interest as a function of a certain control parameter. In the case of aeroelastic systems such as wings, the quantity of interest may be the linear or angular displacement at a point of the structure (e.g., wingtip vertical displacement) and the control parameter is typically a flight condition parameter, such as flow speed or dynamic pressure. There are many methods to obtain bifurcation diagrams, including [19] *i)* direct time-marching simulations [20, 21], *ii)* time-cyclic schemes [22, 23, 24, 25], *iii)* time-shooting [26], *iv)* harmonic balance methods [27, 28, 29, 30, 31], *v)* time-spectral methods [32], *vi)* nonlinear perturbation methods [33, 34, 35, 36], and *vii)* numerical continuation methods [37]. These methods may also be combined to improve their accuracy and performance [38]. However, the problem with these methods is that they are not sufficiently accurate and computationally efficient for complex aeroelastic systems such as those of interest for realistic design applications. Furthermore, several of these methods involve guesswork, intrusive solution processes, or other related computational challenges. A non-intrusive approach for computing bifurcation diagrams, known as bifurcation forecasting [39], involves using a limited number of transient responses, as few as two, to evaluate the recovery rate to equilibrium before the flutter onset for a range of amplitudes. This quantity is then extrapolated for each amplitude to predict LCO solutions, characterized by a zero recovery rate. This approach extends conventional output-based damping identification and extrapolation methods, which cannot predict LCOs [40, 41, 42]. Despite its advantages, it requires accurate recovery rate calculations based on output data from transient responses, posing challenges for large-dimensional systems due to issues such as the presence of multiple frequency components or numerical noise [43, 44].

Another problem in constraining LCOs based on bifurcation diagrams is that some of the current LCO constraint formulations are unsuitable for gradient-based optimization [18]. This optimization approach is necessary for cases where the design space is large-dimensional, as it can accurately converge to an optimized solution consistently and efficiently [2]. However, to ensure this is possible for the optimizer, several requirements on the objective and constraint functions must be met. Both of these functions must be continuous, smooth, and differentiable in the design variables of choice. These requirements make sure that there is no location within the design space for which a constraint or objective cannot be evaluated, or where a “sharp” response surface results in an inaccurate derivative, allowing the optimizer to progress along a descent direction. Direct constraints on bifurcation diagrams do not meet these requirements because aeroelastic systems can have multiple flutter mechanisms, potentially switching as the design variables change [18]. As such, these constraints are undesirable for gradient-based optimization because they are discontinuous if there is a switch in the critical flutter mechanism. Furthermore, to minimize the computational burden of the optimization, adjoint derivatives are desirable compared with derivatives from alternative methods such as finite differences, which are less accurate and not scalable. The computational cost of adjoint derivatives is independent of the number of design variables but only dependent on the number of objective and constraint functions, favoring scalar constraints. However, most existing LCO constraints do not come with adjoint derivatives or these derivatives are obtained under simplifying assumptions such as the absence of shifts in the critical flutter mechanism [18].

Due to the above-mentioned limitations in current LCO constraints, prior optimization research considering LCO behavior has been limited to structural or aerodynamic shape optimization of simple models. Examples of structural models optimized considering their LCO characteristics include typical aeroelastic sections or geometrically nonlinear beams and plates [45, 46, 47, 16, 48]. Aerodynamic shape optimizations focus on shape optimizations of the airfoil or wing design within the transonic regime, due to their applications to commercial flight as it exists today [49, 50, 5]. These types of optimizations have historically used the bifurcation diagram, constraining the amplitude, slope, or LCO onset for a single flutter mechanism. As previously mentioned, this approach poses significant challenges for gradient-based optimization, especially in cases with multiple instability mechanisms. These issues have so far prevented large-scale design optimization of realistic aeroelastic systems considering LCO constraints.

To meet the requirements for gradient-based optimizers, including handling shifts in the critical flutter mechanism, Ref. [18] developed an LCO constraint that bounds the recovery rate to equilibrium for chosen operating conditions, modes, and amplitudes. This approach does not require or use the bifurcation diagram, bypassing the need to compute it within the optimization process. Instead, it implicitly constrains LCOs by leveraging the condition of zero recovery rate satisfied by LCO solutions [39]. To overcome the problem of a potentially non-smooth design space and avoid the need for mode-tracking algorithms, the constraints on the recovery rates for different flight conditions, modes, and amplitudes are aggregated using the Kreisselmeier-Steinhauser (KS) function [51, 52, 53]. This function provides a continuous, smooth, and differentiable LCO constraint that conservatively estimates the most violated of the original recovery rate constraints and meets the requirements for gradient-based optimizers. The constraint also consists of a scalar metric, favoring adjoint derivatives. Reference [18] demonstrated a successful optimization with this methodology, eliminating subcritical LCOs from a typical aeroelastic section model within a range of flight conditions.

1.3 Thesis Objective, Contributions, and Outline

The objective of this thesis is to advance the methodology of Ref. [18] by addressing a key limitation. In the previous work, the LCO constraint was evaluated by approximating recovery rates using a projection-based approach [54]. This approach has only been demonstrated on systems with smooth structural nonlinearities described by polynomial functions [54] and was found to lose accuracy for larger LCO amplitudes, yielding conservative values of the LCO constraint [18]. This thesis introduces a new approach for evaluating recovery rates in the LCO constraint, which can be applied to general nonlinear behaviors and preserves accuracy at large LCO amplitudes while keeping numerical robustness. The recovery rates are evaluated using a new matrix pencil method that enhances the method of Ref. [55] through a sliding time window within the transient response of a chosen quantity of interest. This new sliding-window matrix pencil method captures amplitude-varying effects associated with dynamic nonlinearities, which must be retained to characterize LCO behaviors but are neglected in the conventional matrix pencil method due to its linearized assumptions. Sensitivity analyses are conducted to identify the optimal sliding-window matrix pencil method parameters for accurate recovery rate results.

The new developments are then demonstrated by revisiting the optimization problem of Ref. [18] to investigate their impact on the optimized solution and the computational efficiency and numerical robustness of the optimization process. The work is anticipated to pave the way for addressing larger-scale design optimization problems involving aeroelastic models of increasing physical and computational complexity, pending the formulation of adjoint derivatives. The developments presented in this thesis could also be applied to design optimization of systems exhibiting LCOs encountered in other engineering domains.

The remainder of the thesis is organized as follows: chapter 2 recalls the formulation of the LCO constraint adopted in this work and presents the new matrix pencil method based on the proposed sliding-window approach; chapter 3 describes the basic typical section aeroelastic model used for demonstration as well as the formulation for potential future models considering more sophisticated aerodynamic theories; chapter 4 investigates the ability of the proposed sliding-window matrix pencil method to capture recovery rates and presents sensitivity analyses aimed at identifying optimal method parameters; chapter 5 discusses the optimization results for a two-variable demonstration problem and compares them with prior results from Ref. [18]; chapter 6 references the next steps to be taken to further develop this work. Two appendices describe the implementation of an optimization framework that will serve as a starting point for future research on adjoint derivatives and a three-variable optimization demonstration as an additional proof of concept.

CHAPTER 2

CONSTRAINT FORMULATION

This section presents the constraint formulation used in this work. First, section 2.1 recalls the formulation of a state-of-the-art flutter constraint based on eigenvalues. The constraint formulation follows the derivation in Ref. [12] and is reported to provide the background necessary to understand the LCO constraint. The LCO constraint is presented in section 2.2 based on the formulation introduced in Ref. [18]. Finally, section 2.3 introduces the proposed approach to evaluating the LCO constraint based on recovery rates from the proposed sliding-window matrix pencil method, which is the primary contribution of this thesis.

2.1 Flutter Constraint

A nonlinear system can generally be defined by the following equation:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{c}, \mathbf{y}(\mathbf{c}, \mathbf{x}), \mathbf{x}) \tag{2.1}$$

In this equation, $\mathbf{c}$ is a vector of control parameters of size $N_c \times 1$, $\mathbf{x}$ is a vector of design variables of size $N_x \times 1$, $\mathbf{y}$ is the $N_y \times 1$ vector of the state variables, $\mathbf{f}$ is a $N_y \times 1$ vector field that provides the time rate of change of the state vector (termed state velocity). The control parameters $\mathbf{c}$ parametrize the system's boundary and flight conditions. Examples include the load factor and Mach number. The combination of such parameters characterizes a point within the flight envelope of the system of interest, such as a wing or aircraft.

Choosing N_s samples of the vector of control parameters, denoted by $\mathbf{c}_i$, for a fixed design variable set $\mathbf{x}$, each sample $\mathbf{c}_i$ $(i = 1, \ldots, N_s)$ has an associated equilibrium solution $\mathbf{y}_{e_i}(\mathbf{x}) = \mathbf{y}_e(\mathbf{c}_i, \mathbf{x})$ satisfying

$$\mathbf{f}(\mathbf{c}_i, \mathbf{y}_{e_i}(\mathbf{x}), \mathbf{x}) = \mathbf{0} \tag{2.2}$$

The dynamics can be linearized at small amplitudes about the equilibrium state $\mathbf{y}_{e_i}(\mathbf{x})$, resulting in a linearized equation defined by the local $N_y \times N_y$ Jacobian matrix of the system $\mathbf{A}_i(\mathbf{x})$:

$$\Delta\dot{\mathbf{y}} = \mathbf{A}_i(\mathbf{x})\,\Delta\mathbf{y}, \qquad \mathbf{A}_i(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{c}, \mathbf{y}(\mathbf{c}, \mathbf{x}), \mathbf{x})}{\partial \mathbf{y}}\bigg|_{\mathbf{c}=\mathbf{c}_i, \mathbf{y}(\mathbf{c},\mathbf{x})=\mathbf{y}_{e_i}(\mathbf{x})} \tag{2.3}$$

The quantity $\Delta\mathbf{y}$ is defined as $\Delta\mathbf{y} = \mathbf{y} - \mathbf{y}_{e_i}$ and represents a perturbation of the state vector $\mathbf{y}$ around the equilibrium state $\mathbf{y}_{e_i}$. The eigenvalues of $\mathbf{A}_i(\mathbf{x})$ are denoted by $\sigma_{ik}(\mathbf{x}) = g_{ik}(\mathbf{x}) + j\omega_{ik}(\mathbf{x})$ ($k = 1, \ldots, N_y$), where the real part $g_{ik}(\mathbf{x})$ is the damping associated with the kth mode at the ith equilibrium state, and the imaginary part $\omega_{ik}(\mathbf{x})$ is the corresponding angular frequency, and j the imaginary unit.

Stable dynamics require that all eigenvalues have negative real parts. As a result, the equilibrium state at control parameter $\mathbf{c}_i$ is stable if the following constraints are satisfied:

$$g'_{ik}(\mathbf{x}) = g_{ik}(\mathbf{x}) - G_i \leq 0 \qquad \begin{array}{l} \forall\, i = 1, \ldots, N_s \\ \forall\, k = 1, \ldots, N_m \end{array} \tag{2.4}$$

The quantity $N_m \leq N_y$ is the number of constrained aeroelastic modes, which, by nature, must be at or below the total number of states. This formulation includes a safety factor $G_i = G(\mathbf{c}_i)$, which is the value of a damping bounding curve at $\mathbf{c}_i$ that can be used to require residual damping while the flutter constraint is active for a more robust design [2].

Each damping constraint is a continuous, smooth, and differentiable function of $\mathbf{x}$, as required for gradient-based optimization [3]. However, the number of constraints increases with the number of modes and flight conditions. Thus, the constraints are aggregated through the KS function [51, 53, 52]:

$$\mathrm{KS}_{\text{flutter}}(\mathbf{x}) = g'_{\max}(\mathbf{x}) + \frac{1}{\rho_{\mathrm{KS}_{\text{flutter}}}} \ln\left\{\sum_{i=1}^{N_s}\sum_{k=1}^{N_m} \exp\left\{\rho_{\mathrm{KS}_{\text{flutter}}}\left[g'_{ik}(\mathbf{x}) - g'_{\max}(\mathbf{x})\right]\right\}\right\} \leq 0 \tag{2.5}$$

The KS aggregation facilitates the calculation of adjoint derivatives, whose computational cost is only dependent on the number of outputs of interest (objective and constraint functions). The above equation assumes a constant value of the aggregation parameter $\rho_{\mathrm{KS}_{\mathrm{flutter}}}$ for all the modes and equilibrium states, although Poon and Martins [53] offer a framework by which the aggregation parameter can be updated during the optimization. The scalar constraint denoted by $\mathrm{KS}_{\mathrm{flutter}}(x)$ is a function of the most positive bounded damping constraint, denoted by $g'_{\max}(\mathbf{x})$, and tends to its value as $\rho_{\mathrm{KS}_{\mathrm{flutter}}}$ tends to infinity. Essentially, as the $\rho_{\mathrm{KS}_{\mathrm{flutter}}}$ parameter decreases, the design space is reduced in favor of a more conservative estimation of the most violated damping constraint. On the other hand, as the KS parameter increases, the aggregated constraint more closely approximates the most violated constraint, opening up the feasible design space by reducing overestimation. If the parameter is too high, the design space can become non-smooth, leading to issues with gradient-based optimizers such as those considered in this work. As such, the KS parameter must be balanced within a range specific to the problem to ensure appropriate accuracy in capturing the true feasible design space while avoiding numerical issues for the optimizer. This problem-specific range depends on the order of magnitude of the aggregated quantities and the difference among their values.

2.2 Limit-Cycle Oscillation Constraint

If the system has a supercritical behavior, the above damping-based flutter constraint is sufficient to prevent LCOs during operation. This is not true for subcritical LCOs, which can also arise at linearly stable conditions. Additionally, flutter constraints cannot mitigate LCO amplitudes, as they do not account for dynamic nonlinear effects. These issues can be handled by the LCO constraint introduced in Ref. [18]. This constraint is mathematically similar to the above flutter constraint, but it is formulated in terms of a fully nonlinear metric called recovery rate, which accounts for amplitude-dependent effects that are otherwise neglected in damping values.

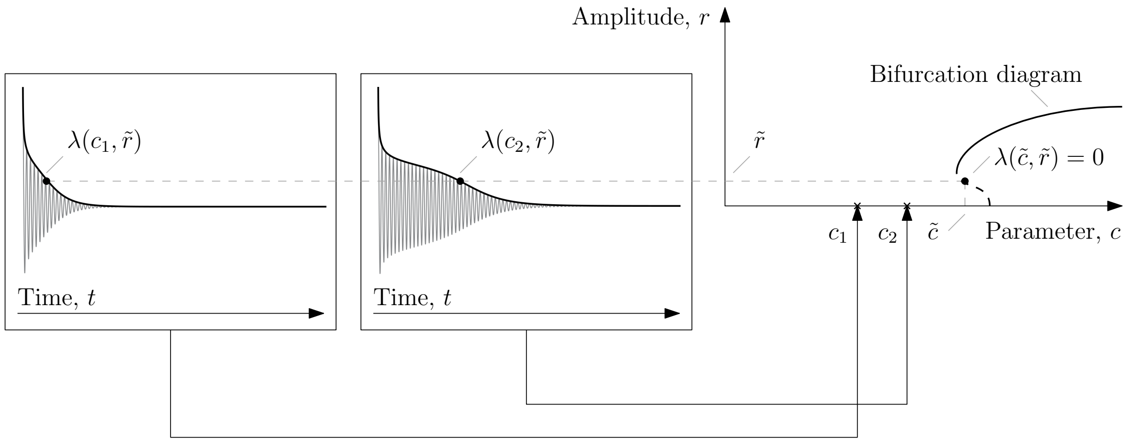


Figure 2.1: Recovery rate interpretation for a hypothetical system (Ref. [18]).

The concept of recovery rate can be illustrated for a one-dimensional nonlinear system

$$\dot{r} = f(c, r) \tag{2.6}$$

The quantity r in the above equation is the response amplitude measured from equilibrium and c is a control parameter. The recovery rate $\lambda = \lambda(c, r)$ is the time slope of the response amplitude logarithm [39]:

$$\lambda = \frac{d}{dt} \ln r = \frac{\dot{r}}{r} \tag{2.7}$$

This quantity depends on the amplitude r and the parameter c, and it describes the system's resilience in recovering to the original equilibrium state after a disturbance. Points $(\tilde{c}, \tilde{r})$ on the bifurcation diagram satisfy

$$\lambda(c, r) = 0 \tag{2.8}$$

This condition is verified when the system develops an LCO because it no longer recovers to its original equilibrium state but shows a bounded self-sustained oscillation. As a result, the recovery rate quantifies the distance of a dynamic state from its corresponding state on the bifurcation diagram (Figure 2.1). The LCO constraint of Ref. [18] uses this property to implicitly constrain LCO solutions without computing them directly. Mathematically, this is achieved by the following recovery rate constraints [18]:

$$\lambda'_{ikl}(\mathbf{x}) = \lambda_{ikl}(\mathbf{x}) - \Lambda_{il} \leq 0 \qquad \begin{array}{l} \forall\, i = 1, \dots, N_s \\ \forall\, k = 1, \dots, N_m \\ \forall\, r = 1, \dots, N_r \end{array} \tag{2.9}$$

The quantity $\lambda_{ikl}(\mathbf{x})$ is the recovery rate associated with the ith flight condition at the lth amplitude along the kth mode, whereas $\Lambda_{il} = \Lambda(\mathbf{c}_i, r_l)$ is a recovery rate safety factor similar to the safety factor used in the damping-based flutter constraint. The recovery rate constraints are then aggregated to produce the following scalar LCO constraint [18]:

$$\mathrm{KS}_{\mathrm{LCO}}(\mathbf{x}) = \lambda'_{\max}(\mathbf{x}) + \frac{1}{\rho_{\mathrm{KS}_{\mathrm{LCO}}}} \ln \left\{ \sum_{i=1}^{N_s} \sum_{k=1}^{N_m} \sum_{l=1}^{N_r} \exp \left\{ \rho_{\mathrm{KS}_{\mathrm{LCO}}} \left[\lambda'_{ikl}(\mathbf{x}) - \lambda'_{\max}(\mathbf{x}) \right] \right\} \right\} \leq 0 \tag{2.10}$$

The above KS aggregation yields a conservative estimate of the most violated recovery-rate constraint and tends to its true value as the aggregation parameter $\rho_{\mathrm{KS}_{\mathrm{LCO}}}$ (which is here the same for all aggregated constraints) goes to infinity. As for the flutter constraint, the choice of the aggregation parameter depends on the problem and requires a compromise between a less conservative constraint estimation and potential numerical issues in the presence of less smooth design spaces.

2.3 Sliding-Window Matrix Pencil Method for Recovery Rate Evaluation

Previously, the above LCO constraint has been evaluated using a projection-based approach introduced in Ref. [54]. This approach uses the local state matrix of the system at a given flight condition and its right eigenvectors to define a perturbed dynamic state along a particular mode and at a given phase. The resulting state velocity is then evaluated and projected onto the left eigenvectors to approximate the recovery rate, and the process is repeated across a set of amplitudes. This approach captures the amplitude trend of the recovery rate qualitatively, capturing the type of bifurcation (subcritical or supercritical). As a result, it allowed the LCO constraint to be successfully demonstrated by optimizing a typical aeroelastic section model while suppressing subcritical LCOs over a range of flight conditions [18]. However, the above-mentioned approach to evaluate the recovery rate has only been tested on systems with smooth structural nonlinearities of polynomial nature and its applicability to more general types of nonlinear behaviors has not yet been proved. The approach was also found to introduce approximations in the recovery rates for larger LCO amplitude, which may overestimate the LCO constraint and yield a conservative design solution as the constraint does not reflect the true recovery rate of the system. To address these limitations, this work introduces a new approach for evaluating the recovery rate using output data from transient responses. This approach can accurately capture the recovery rate over a larger amplitude range and is completely output-based for easier generalization to systems with various nonlinearities. The proposed approach is also anticipated to be robust in the presence of multi-component transient responses and numerical noise typical of large-dimensional models.

The approach consists of the following steps. The first step involves defining the flight conditions to simulate the transient responses of the system. In this work, the transient response at each flight condition is generated by perturbing the system from its local equilibrium state with an initial condition given by the real component of the right eigenvector of a mode of interest with an appropriate scaling factor. This approach is chosen to compare with the previous work in Ref. [18], where the recovery rates were approximated by evaluating the state velocity for perturbed dynamic states along various right eigenvectors. However, the proposed approach could be applied to other initial perturbations. Next, the system's transient response is integrated in time according to the system's governing equations. Next, the time history for a chosen quantity of interest (e.g., a state variable) is interpolated to a user-specified uniform sampling step and sliced into time windows with a specified window size and stride size between starting points. The signal within each window is used to estimate the local dominant damping using the matrix pencil method of Ref. [55]. The matrix pencil method is conventionally applied to a whole time history to obtain the dominant damping estimate under linearized assumptions, that is, by assuming damping to be constant in amplitude. The new contribution of this work consists of applying the conventional matrix pencil method *within each time window* to resolve amplitude effects associated with dynamic nonlinearities, which are essential for characterizing LCOs. The amplitude-varying behavior of the recovery rate is captured by sliding the time window along the time history until the local dominant damping no longer varies, either because the response amplitude around the original equilibrium state is sufficiently small for the system to show a locally linear behavior or because an LCO has fully developed.

An example of this sliding-window approach is shown in Figure 2.2 for a pitch angle time history. In the figure, the stride is the number of peaks between the starting points of two consecutive windows whereas the window size is the number of peaks between the start and end of an individual window. For simplicity, the window and stride size are assumed to be constant for each time history. The sample window size encompasses several oscillation periods across which the response amplitude varies. This can be used as a check to ensure that the window size is appropriately large for the time history of interest. Too small of a window size will not be sufficient for the method to accurately capture the local recovery rate due to an insufficient number of peaks, whereas too large of a window size will not resolve amplitude-dependent effects, missing information on the nonlinear dynamics of the system relevant to characterizing LCOs.

As mentioned, the time history within each time window is processed through the matrix pencil method of Ref. [55] to obtain the minimum damping estimate for that amplitude range. This local dominant damping estimate is hypothesized to coincide with the opposite of the recovery rate, as the recovery rate reduces to a conventional linearized damping for small amplitudes. This hypothesis is confirmed by numerical studies presented later in this thesis where the recovery rate estimated using the proposed method is compared with reference results based on an alternative method. The different sign of the recovery rate and

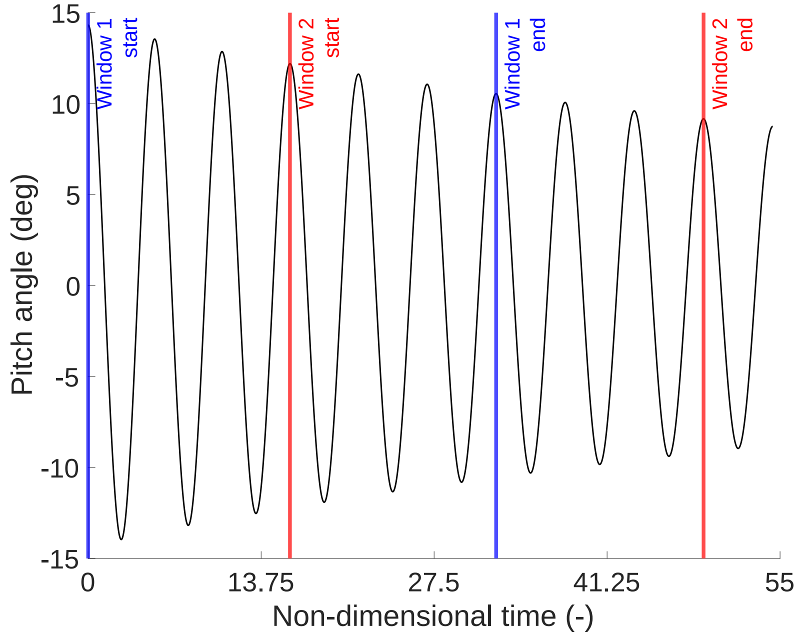


Figure 2.2: Schematic of proposed sliding-window matrix pencil method.

local minimum damping is due to the convention used in the matrix pencil method, with positive damping denoting a stable system. The recovery rate estimated for each time window is associated with an amplitude value equal to the mean value of the amplitude range spanned by the time window. Repeating the process by sliding the time window with the chosen stride along the time history allows one to obtain a set of recovery rates associated with different mean amplitudes, characterizing dynamic nonlinear effects in the system.

The matrix pencil method accurately estimates the coefficients in a Prony series (a series of damped sinusoids), which is suitable for describing the oscillatory responses of aeroelastic systems. A generalized form of a Prony series is [55]

$$y_n = \sum_{k=1}^{M} c_k e^{s_k n} + w_n \qquad n = 1, \ldots, N_t \tag{2.11}$$

In this equation, y_n denotes the nth output data sample for the quantity y, N_t is the total number of samples for the time history, M is the number of retained sinusoids, and the complex-valued amplitude and exponent of each sinusoid are given by c_k and s_k, respectively. The quantity w_n is an unknown signal noise term. The complex exponent s_k can be rewritten as follows:

$$s_k = (-\alpha_k + i\omega_k)\,\Delta t \tag{2.12}$$

In this formulation, the damping associated with the sinusoid is represented by α_k, the frequency by ω_k, and the output data are sampled with time step Δt. The matrix pencil method requires that the time step be uniform. The objective of the method is to estimate the complex amplitudes c_k and exponents α_k based on output data. As mentioned earlier, a positive damping value α_k corresponds to a converging oscillation due to the minus sign in the real part of the complex exponent s_k, whereas a negative damping value corresponds to a diverging oscillation. This convention is opposite to the one used for describing the flutter constraint, where damping is directly defined as the real part of the aeroelastic eigenvalue such that negative damping corresponds to a converging oscillation. Similarly, the convention is opposite to the one defining the recovery rate, as a negative recovery rate corresponds to the system recovering to the original equilibrium state (no LCOs).

A summary of the matrix pencil method's workflow is provided below, and further details can be found in Ref. [55]. The method first defines an iteration-dependent truncated Hankel matrix, containing the output data samples:

$$\boldsymbol{Y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_L & y_{L+1} \\ y_2 & y_3 & \cdots & y_{L+1} & y_{L+2} \\ \vdots & \vdots & \ddots & \vdots & \\ y_{N_t-L} & y_{N_t-L+1} & \cdots & y_{N_t-1} & y_{N_t} \end{bmatrix} \tag{2.13}$$

The matrix size depends on the matrix pencil parameter $L = N_t/2 - 1$. After performing a singular value decomposition (SVD) on $\boldsymbol{Y}$, singular values deemed to be purely noise (below a specified normalized SVD tolerance) are discarded. For instance, a tolerance of 0.1 requires that a normalized singular value be 10% of the maximum eigenvalue or larger to be retained. Next, the right singular vectors associated with the M retained singular values are stored in matrix $\hat{\boldsymbol{V}}$ whereas the remaining ones are discarded. From the filtered right singular vectors, one forms matrices $\hat{\boldsymbol{V}}_1$ and $\hat{\boldsymbol{V}}_2$, which consist of the first and last L rows, respectively. The matrices are used to build

$$\boldsymbol{H} = \hat{\boldsymbol{V}}_2^T \left[\hat{\boldsymbol{V}}_1^T \right]^+ \tag{2.14}$$

The superscript T indicates the transpose operation and $+$ indicates the right pseudoinverse operation. The eigenvalues of $\boldsymbol{H}$ approximate the complex exponents s_k ($k = 1, \ldots, M$), from which one obtains

$$\alpha_k = -\frac{\mathrm{Re}(s_k)}{\Delta t} \tag{2.15}$$

Finally, the minimum damping is estimated using the KS aggregation function [51]:

$$\alpha_{\mathrm{KS}} = \alpha_{\min} - \frac{1}{\rho_{\mathrm{KS}}} \ln \left[\sum_{k=1}^{M} e^{-\rho_{\mathrm{KS}}(\alpha_k - \alpha_{\min})} \right] \approx \alpha_{\min} \tag{2.16}$$

The quantity $\alpha_{\min}$ is the true minimum damping whereas α_{KS} is the estimate based on the aggregation parameters ρ_{KS}.

The matrix pencil method is advantageous over alternative output-based damping identification methods due to its accuracy for short and multi-component transient responses and inherent ability to filter noise within the method itself, avoiding the need for approaches based on alternative techniques such as frequency filtering [55, 42]. Furthermore, it solves an eigenvalue problem instead of using root-finding methods to compute the complex exponents and therefore the damping values of interest, minimizing the computational cost and making this method more computationally efficient than other damping identification methods. As mentioned previously, the main limitation of the method consists of assuming dynamically linear behavior, namely, of providing a single aggregated damping for a given set of processed output data samples. As a result, when conventionally applied to a complete time history, the matrix pencil method neglects amplitude-varying effects that must be captured to characterize LCOs. This limitation is overcome in this work using the proposed sliding window approach. In the sliding-window approach, the matrix pencil method is individually applied to a set of smaller time windows within the time histories, and the quantity α_{KS} is evaluated for each time window, resulting in a set of values corresponding to different amplitudes as the time window slides along the time history. The recovery rate for a given amplitude is assumed to coincide with the opposite of α_{KS} evaluated from the time window that spans a portion of the signal with that average amplitude. The process then repeats to estimate a complete recovery rate curve. Note that the proposed sliding-window approach can overshoot initial transients associated with cases where the system's response grows to a larger amplitude than the initial perturbation, resulting in a positive recovery rate. Therefore, if the root mean square values of the recovery rates are below an established normalized threshold of 0.001, the first three recovery rates are resampled with a smaller sliding window to ensure the initial transient is well captured.

The performance of the proposed sliding-window matrix pencil method depends on several parameters that must be appropriately set before evaluating an LCO constraint. The key parameters are the SVD tolerance, the KS aggregation parameter, the sliding window size, the stride size, and the output data sampling step. The SVD tolerance describes the relative contribution of each sinusoidal component to the signal, which must be above the set SVD threshold for that component to be retained. The SVD threshold acts as an in-built noise filtering mechanism within the matrix pencil method, which is one of the characteristic features of the method making it particularly appealing. The KS aggregation parameter is used for the KS aggregation across the damping values for different retained sinusoidal components in a given time window. This KS aggregation parameter corresponds to the one used in the nested KS aggregation over the modes involved in the flutter and LCO constraints. As the aggregation parameter increases, the aggregate more closely approximates the true aggregate damping value for that time window; as it decreases, the estimation is more conservative but smoother. The window size is defined by the number of peaks between the starting and ending peaks of the considered time interval of the time history. This is paired with the stride size, also specified as a certain number of peaks between the starting point of consecutive windows. Finally, the sampling step is the number of output data points between the output data samples selected from the original time history, whose initial sampling step is given by the integration time step (which may not be constant along the time history). As the output data sampling step in the matrix pencil method increases compared with the original integration time step, the time history processed using the method becomes more granular. This can be qualified by normalizing the sampling step by the oscillation period at the end of the transient response, which captures the least damped sinusoidal component typically associated with the bifurcating mode. Sensitivity analyses to investigate optimal values of these parameters are discussed in chapter 4.

CHAPTER 3

TYPICAL SECTION MODEL

The test case used to demonstrate the methodology consists of a basic typical aeroelastic section model. This model is an abstraction for a flexible wing undergoing out-of-plane bending and torsion and is a canonical test case for aeroelastic optimization subject to flutter and LCO constraints [2]. The typical section model being considered is illustrated in Figure 3.1. Points A, C, and E denote the aerodynamic center (assumed at the quarter-chord point), the center of gravity, and the elastic center, respectively. The equations assume zero body angle of attack and neglect gravity. The structural degrees of freedom are the downward-positive vertical (plunge) displacement of the elastic center, denoted by h, and the clockwise-positive pitch angle, denoted by α. This section first outlines the model formulation used in the previous work that introduced the LCO constraint [18], which uses a simplified potential flow aerodynamic model. This model is also used in the present studies for comparison purposes. Next, details are provided on how this formulation can be coupled with aerodynamic models of increasing complexity in the future. One potential candidate aerodynamic model is the theory of Peters et al. [56], which captures unsteady effects while keeping linear analytical equations under the assumption of potential flow and small-amplitude motions. Other models include two-dimensional unsteady Euler equations to be later extended to unsteady Navier-Stokes equations, which can capture nonlinear effects. Additional utility to couple with a computational fluid dynamics solver providing aerodynamic loads is under development.

3.1 Basic Quasi-Steady Potential Flow Formulation

In this basic formulation, the aerodynamics are modeled using quasi-steady, potential flow thin airfoil theory by considering the circulatory effects due to the effective angle of attack

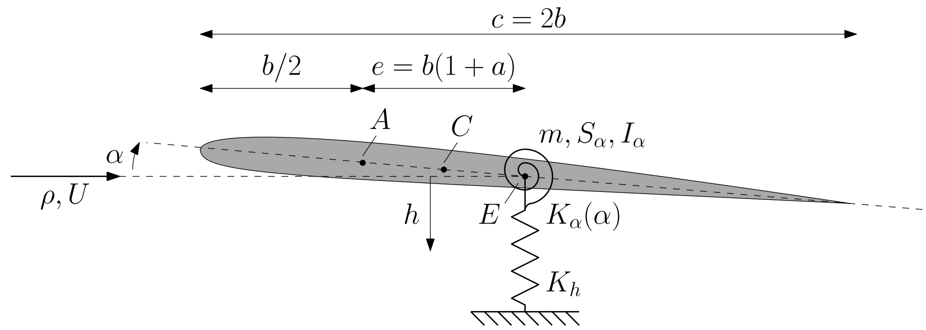


Figure 3.1: Schematic of a typical section model (from Ref. [18]).

and the apparent mass and inertia terms, but no wake effects. This aerodynamic model is chosen due to the availability of optimization results from previous work based on the same LCO constraint but a different method to evaluate the recovery rate [18]. It must be emphasized that a quasi-steady aerodynamic model will generally not capture the flutter onset accurately, although it can qualitatively capture basic instability mechanisms. These approximations are acceptable for an initial proof of concept because the scope of this work is to demonstrate the methodology and not to obtain a realistic design. Additionally, the "degenerate" quasi-steady steady aerodynamic model considered here, which includes aerodynamic mass, damping, and stiffness effects but no wake effects, may often yield more approximate flutter predictions than an even simpler quasi-steady aerodynamic model that, for instance, only retains the effect of the plunge velocity in the effective angle of attack but neglect other aerodynamic mass and damping effects. The reason for opting for such a model is to prepare for future extensions to fully unsteady aerodynamics, also considering that, as mentioned earlier, this work aims to demonstrate the proposed methodology and not to obtain accurate quantitative predictions.

The equations of motion for the typical section read

$$\begin{aligned} m\ddot{h} + S_\alpha \ddot{\alpha} + K_h\, h &= -L \\ S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + K_\alpha(\alpha)\alpha &= M \end{aligned} \tag{3.1}$$

Under the aforementioned assumptions, the lift L and pitching aerodynamic moment M

per unit span about the point E are given by

$$
\begin{aligned}
L &= \pi\rho b^2\left(\ddot{h} + U\dot{\alpha} - ba\ddot{\alpha}\right) + 2\pi\rho b U\left[\dot{h} + U\alpha + b\left(\frac{1}{2} - a\right)\dot{\alpha}\right] \\
M &= -\pi\rho b^3\left[\frac{\ddot{h}}{2} + U\dot{\alpha} + b\left(\frac{1}{8} - \frac{a}{2}\right)\ddot{\alpha}\right] + b\left(\frac{1}{2} + a\right)L
\end{aligned}
\tag{3.2}
$$

The quantity m is the typical section mass per unit length; S_α the static unbalance per unit length; I_α the pitch moment of inertia about E per unit length; ρ the air density; U the flow speed; b the half chord; $e = b(1+a)$ the offset of the elastic center E from the aerodynamic center A, taken positive downstream; and a the non-dimensional offset of E from the half-chord point, also take positive downstream. The elastic reactions by the translational and rotational springs are described by K_h, the spring constant per unit length associated with the translational spring, and $K_\alpha(\alpha)$, the spring coefficient per unit length associated with the rotational spring, which is assumed to be a polynomial function of the pitch angle:

$$
K_\alpha(\alpha) = K_\alpha^{(1)} + K_\alpha^{(3)}\alpha^2 + K_\alpha^{(5)}\alpha^4 \tag{3.3}
$$

The quantities $K_\alpha^{(i)}$ $(i = 1, 3, 5)$ are the coefficients of the linear, cubic, and fifth-order terms in the elastic reaction. Low-order polynomials are widely used to extend linear aeroelastic models with smooth structural nonlinearities [19]. In realistic systems, these nonlinearities arise from large displacements or nonlinear material behavior (or both), and they are softening if restoring elastic effects weaken compared with the linear case or hardening vice versa. Each nonlinear term in the expression for the rotational spring coefficient can be softening or hardening based on its sign. If both nonlinear terms are positive (including the case where one is zero), the model will exhibit supercritical LCOs associated with hardening nonlinear effects; if one of the two terms is negative, the model will exhibit subcritical LCOs associated with softening nonlinear effects. Typically, the case of subcritical LCOs is achieved by setting $K_\alpha^{(3)}$ to a negative value and $K_\alpha^{(5)}$ to a positive one.

The equations of motion can be normalized by introducing the quantities

$$\begin{aligned} \overline{m} &= \frac{m}{\rho\pi b^2} & x_\alpha &= \frac{S_\alpha}{mb} & \Omega^2 &= \frac{\omega_h^2}{\omega_\alpha^2} & r_\alpha^2 &= \frac{I_\alpha}{mb^2} \\ \overline{h} &= \frac{h}{b} & \bar{t} &= t\omega_\alpha & \overline{\omega} &= \frac{\omega}{\omega_\alpha} & \overline{U} &= \frac{U}{b\omega_\alpha} \end{aligned} \tag{3.4}$$

The quantities $\omega_h^2 = K_h/m$ and $\omega_\alpha^2 = K_\alpha^{(1)}/I_\alpha$ are the squared uncoupled structural frequencies. The nonlinear terms are described by the quantities

$$\kappa_\alpha^{(3)} = \frac{K_\alpha^{(3)}}{K_\alpha^{(1)}} \qquad \kappa_\alpha^{(5)} = \frac{K_\alpha^{(5)}}{K_\alpha^{(1)}} \tag{3.5}$$

Substituting Equation 3.2 to Equation 3.5 into the equation of motion and by dividing the first equation by $mb\omega_\alpha^2$ and the second equation by $mb^2\omega_\alpha^2$, the system is cast as

$$\mathbf{M}\begin{Bmatrix} \ddot{\overline{h}} \\ \ddot{\alpha} \end{Bmatrix} + \mathbf{D}\left(\overline{U}\right)\begin{Bmatrix} \dot{\overline{h}} \\ \dot{\alpha} \end{Bmatrix} + \mathbf{K}\left(\overline{U}\right)\begin{Bmatrix} \overline{h} \\ \alpha \end{Bmatrix} + \begin{Bmatrix} 0 \\ r_\alpha^2\left(\kappa_\alpha^{(3)}\,\alpha^3 + \kappa_\alpha^{(5)}\,\alpha^5\right) \end{Bmatrix} = \mathbf{0}_{2\times1} \tag{3.6}$$

The 2×2 aeroelastic mass matrix $\mathbf{M}$ is given by

$$\mathbf{M} = \mathbf{M}_s + \mathbf{M}_a \qquad \mathbf{M}_s = \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix} \qquad \mathbf{M}_a = \frac{1}{\overline{m}}\begin{bmatrix} 1 & -a \\ -a & \left(\frac{1}{8} + a^2\right) \end{bmatrix} \tag{3.7}$$

Matrices $\mathbf{M}_s$ and $\mathbf{M}_a$ are the 2×2 structural and aerodynamic mass matrices, respectively. The 2×2 aeroelastic damping matrix $\mathbf{D}$ consists only of an aerodynamic contribution

$$\mathbf{D}\left(\overline{U}\right) = \mathbf{D}_a\left(\overline{U}\right) = \frac{2\overline{U}}{\overline{m}}\begin{bmatrix} 1 & 1-a \\ -\left(\frac{1}{2}+a\right) & a\left(a-\frac{1}{2}\right) \end{bmatrix} \tag{3.8}$$

Structural damping has not been modeled because it only offsets the flutter point, achieving the same result as setting a non-zero value of the safety window in the flutter constraint.

Finally, the 2×2 aeroelastic stiffness matrix $\mathbf{K}$ is written as

$$\mathbf{K}\left(\overline{U}\right) = \mathbf{K}_s + \mathbf{K}_a\left(\overline{U}\right) \qquad \mathbf{K}_s = \begin{bmatrix} \Omega^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \qquad \mathbf{K}_a\left(\overline{U}\right) = \frac{2\overline{U}^2}{\overline{m}} \begin{bmatrix} 0 & 1 \\ 0 & -\left(\dfrac{1}{2} + a\right) \end{bmatrix} \tag{3.9}$$

Matrices $\mathbf{K}_s$ and $\mathbf{K}_a$ are the 2×2 structural and aerodynamic stiffness matrices.

The non-dimensional equations of motion can be rewritten as

$$\dot{\mathbf{y}} = \mathbf{f}\left(\overline{U}, \mathbf{y}\right) = \mathbf{A}\left(\overline{U}\right)\, \mathbf{y} + \mathbf{f}_{nl}(\mathbf{y}) \tag{3.10}$$

The 4×1 state vector is defined as follows:

$$\mathbf{y} = \left\{\overline{h}, \alpha, \dot{\overline{h}}, \dot{\alpha}\right\}^T \tag{3.11}$$

The corresponding state matrix describing the linear dynamics is given by

$$\mathbf{A}\left(\overline{U}\right) = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ -\mathbf{M}^{-1}\mathbf{K}\left(\overline{U}\right) & -\mathbf{M}^{-1}\mathbf{D}\left(\overline{U}\right) \end{bmatrix} \tag{3.12}$$

The nonlinear terms are given in the 4×1 vector

$$\mathbf{f}_{nl}(\mathbf{y}) = \begin{Bmatrix} \mathbf{0}_{2\times1} \\ \mathbf{f}_{nl}(\alpha) \end{Bmatrix} \tag{3.13}$$

with

$$\mathbf{f}_{nl}(\alpha) = -\mathbf{M}^{-1} \begin{Bmatrix} 0 \\ r_\alpha^2\left(\kappa_\alpha^{(3)}\,\alpha^3 + \kappa_\alpha^{(5)}\,\alpha^5\right) \end{Bmatrix} \tag{3.14}$$

3.2 Generalization to Other Aerodynamic Models

The structural and mass terms can be separated for the model implementation to be more modular and easier to couple with different aerodynamic solvers. In the case of the typical section, the equations of motion can be written as seen below:

$$\mathbf{M}_s \begin{Bmatrix} \ddot{\overline{h}} \\ \ddot{\alpha} \end{Bmatrix} + \mathbf{D}_s \begin{Bmatrix} \dot{\overline{h}} \\ \dot{\alpha} \end{Bmatrix} + \mathbf{K}_s \begin{Bmatrix} \overline{h} \\ \alpha \end{Bmatrix} = \mathbf{R}(\overline{U}, \mathbf{y}, \dot{\mathbf{y}}) \tag{3.15}$$

The term on the right-hand side is given by

$$\mathbf{R}(\overline{U}, \mathbf{y}, \dot{\mathbf{y}}) = -\mathbf{M}_a \begin{Bmatrix} \ddot{\overline{h}} \\ \ddot{\alpha} \end{Bmatrix} - \mathbf{D}_a(\overline{U}) \begin{Bmatrix} \dot{\overline{h}} \\ \dot{\alpha} \end{Bmatrix} - \mathbf{K}_a(\overline{U}) \begin{Bmatrix} \overline{h} \\ \alpha \end{Bmatrix} - \begin{Bmatrix} 0 \\ r_\alpha^2 \left(\kappa_\alpha^{(3)} \alpha^3 + \kappa_\alpha^{(5)} \alpha^5 \right) \end{Bmatrix} \tag{3.16}$$

In this generalized model, structural damping can be given by a damping matrix $\mathbf{D}_s$. As mentioned earlier, a structural damping contribution only shifts the aeroelastic damping values of the system without changing the qualitative behavior. Hence, it does not impact the system's overall behavior and is not considered in this work.

Through some mathematical reformulation, the state velocity can be rewritten as:

$$\dot{\mathbf{y}} = \mathbf{A}_s \, \mathbf{y} + \mathbf{f}_a(\overline{U}, \mathbf{y}, \dot{\mathbf{y}}) + \mathbf{f}_s(\mathbf{y}) \tag{3.17}$$

In this form, $\mathbf{A}_s$ is the state matrix associated with the structural terms:

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ -\mathbf{M}_s^{-1}\mathbf{K}_s & -\mathbf{M}_s^{-1}\mathbf{D}_s \end{bmatrix} \tag{3.18}$$

The quantities $\mathbf{f}_a(\overline{U}, \mathbf{y}, \dot{\mathbf{y}})$ and $\mathbf{f}_s(\mathbf{y})$ are the vectors of the aerodynamic terms, which depends on the specific aerodynamic model being employed, and of the nonlinear structural terms. Note that, because aerodynamics generally involve mass-type effects, the vector of the aerodynamic terms may also depend on the state velocity through the accelerations, requiring a staggered coupling approach or sub-iterations in the solution process. In the case of the typical section model considered previously, the vector of the aerodynamic terms is

$$\mathbf{f}_a(\overline{U},\mathbf{y},\dot{\mathbf{y}}) = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & -\mathbf{M}_s^{-1}\mathbf{M}_a \end{bmatrix} \dot{\mathbf{y}} + \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ -\mathbf{M}_s^{-1}\mathbf{K}_a(\overline{U}) & -\mathbf{M}_s^{-1}\mathbf{D}_a(\overline{U}) \end{bmatrix} \mathbf{y} \tag{3.19}$$

The vector of nonlinear structural terms is

$$\mathbf{f}_s(\mathbf{y}) = \begin{Bmatrix} \mathbf{0}_{2\times1} \\ \mathbf{f}_{nl}(\alpha) \end{Bmatrix} \tag{3.20}$$

with

$$\mathbf{f}_{nl}(\alpha) = -\mathbf{M_s}^{-1} \begin{Bmatrix} 0 \\ r_\alpha^2\left(\kappa_\alpha^{(3)}\,\alpha^3 + \kappa_\alpha^{(5)}\,\alpha^5\right) \end{Bmatrix} \tag{3.21}$$

With this form, a black-box aerodynamic model can supply the aerodynamic contribution to the state velocity. This more general formulation is not used in this work, but it is derived and reported for future implementation so that different types of aerodynamic models can be coupled with the structural typical section equations and used to investigate the LCO constraint considering, for instance, unsteady aerodynamic effects or aerodynamic nonlinearities.

3.3 Baseline Aeroelastic Behavior

The parameters used for the baseline model are listed in Table 3.1. The baseline typical section has a non-dimensional flutter speed slightly above $\overline{U}_F = 0.61$ (Figure 3.2). Figure 3.3 shows the pitch angle bifurcation diagram obtained through direct time marching (markers) and bifurcation forecasting (solid line), which was presented in Ref. [18]. The bifurcation diagram from time marching is constructed by extracting the amplitude of fully developed LCOs at the end of transient responses for 25 non-dimensional speeds between $\overline{U} = 0.57 \rightarrow 0.65$. The bifurcation diagram from bifurcation forecasting is obtained by linearly fitting and extrapolating recovery rates computed from the envelope of two pre-flutter transient responses at $\overline{U} = 0.58, 0.59$. These bifurcation diagrams are not used in the following optimizations but are shown to characterize the behavior of the typical baseline section. Note that the recovery rates used to obtain the bifurcation diagram are not computed using the new matrix pencil method introduced here, but by detecting the envelope of the pre-flutter transient responses using a fitting technique followed by finite differencing. Additional details are provided in Ref. [18]. All transient responses used to obtain recovery rates are generated by applying an initial condition along the real part of the right eigenvector associated with the mode that flutters with a suitable scaling factor to remain within the validity of the present linear aerodynamic model. In the optimization studies, the recovery rates are computed using the proposed sliding-window matrix pencil method and the results are compared with the previous optimization studies in Ref. [18]. In those results, an alternative projection-based method was used to approximate the qualitative amplitude trend of the recovery rate without conducting transient simulations. Additional details on the projection-based method are presented in Refs. [54, 18].

3.4 Model Significance and Limitations

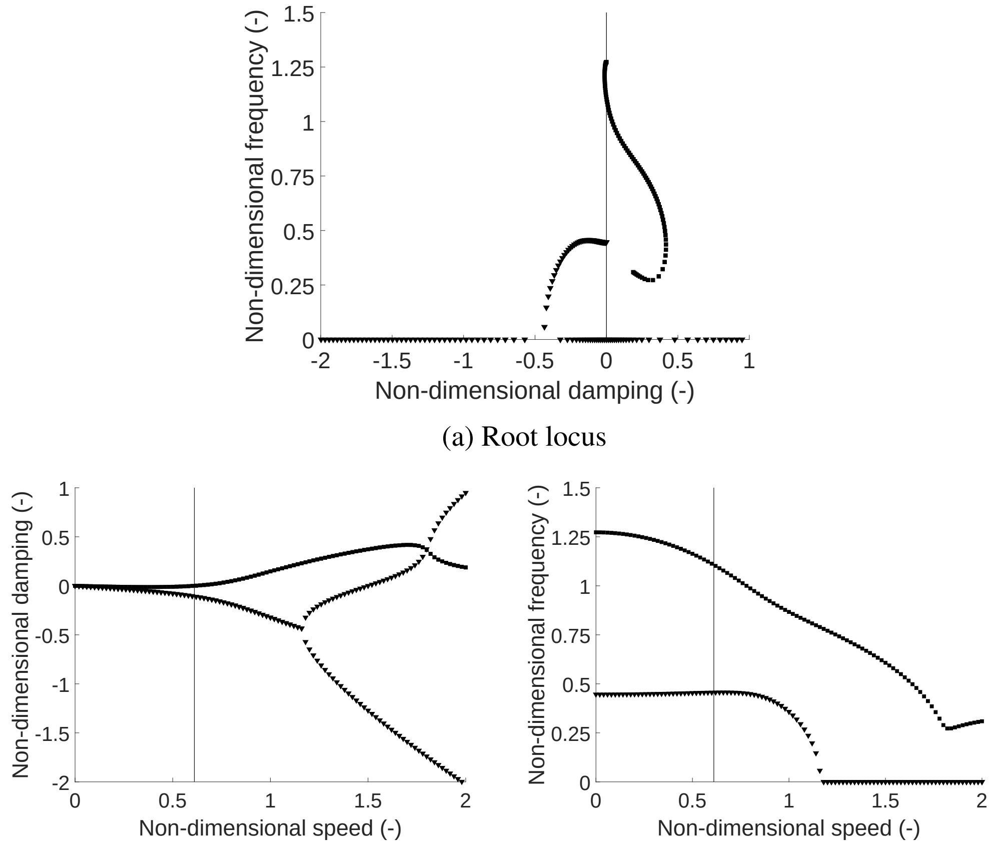
Before moving to the results, it is important to summarize the significance and limitations of the present demonstration model. In the context of aeroelastic optimization, the typical section represents a basic test case upon which the framework outlined in this work can be demonstrated before being applied to realistic design applications in future research. Typical section models have been extensively used for demonstrating aeroelastic design optimization methodologies, especially for dynamic constraints on flutter and LCOS [2]. The appeal of these models lies in their relatively low computational complexity combined with the ability to qualitatively capture key dynamic aeroelastic phenomena. As previously mentioned, the typical section model approximates a wing's out-of-plane bending and torsion, where the wing elasticity is represented by equivalent springs connected to the shear (elastic) center. The nonlinear rotational spring representing the torsional elasticity is modeled by a polynomial function leading to either stiffness hardening or softening depending on the signs of its coefficients. With the assumed baseline parameters, the system initially softens at lower pitch angles before hardening at larger pitch angles, which gives subcritical LCOs. By construction, the model assumes quasi-steady aerodynamics, implying that the aerodynamic forces and moments are only a function of the instantaneous body's movement described by the pitch angle, plunge and pitch velocities, and the corresponding accelerations. Fully unsteady aerodynamics is necessary to capture memory effects associated with wake shedding and other aerodynamic effects associated with flow separation or moving shocks, among other phenomena, which impact the flutter and LCO behaviors of realistic wings. However, quasi-steady aerodynamics are sufficient for demonstrating the framework. Note that this typical section model does not exhibit flutter mode switches. While the ultimate goal of the present formulation is to tackle cases with mode switches, they are not considered in this thesis as they require a more sophisticated, larger-dimensional aeroelastic model. The objective of this thesis is to introduce the new matrix pencil method for evaluating the recovery rate, demonstrate its use in a basic optimization problem, and compare the results with previous work on the same problem that considered an approximate recovery rate evaluation method. The chosen demonstration model is appropriate for this objective.

The optimization studies consider two design variables of interest: $\overline{m}$ and κ_α^3, which are the mass ratio and nondimensional cubic stiffness coefficient associated with the rotational spring. Physically, the mass ratio can be interpreted as the mass per unit length of the wing relative to the mass per unit length of the air contained in a circle having the wing half chord as the radius, which is displaced by the wing during its motion. In the context of design space exploration, changes to the mass ratio can be interpreted as either

Table 3.1: Baseline typical section non-dimensional parameters.

Parameter	Description	Value
$\overline{m}$	Mass ratio	10.00
x_α	Non-dimensional static unbalance	0.20
r_α	Non-dimensional radius of gyration	0.30
a	Non-dimensional offset of E from the half-chord point	-0.30
Ω	Frequency ratio	0.50
$\kappa_\alpha^{(3)}$	Non-dimensional cubic stiffness coefficient	-4.00
$\kappa_\alpha^{(5)}$	Non-dimensional fifth-order stiffness coefficient	100.00

a variation in the aircraft mass distribution at a given altitude or a change in altitude for a given aircraft (or a combination of both). Changes in the nondimensional cubic stiffness coefficient can be interpreted as changes in the elastic behavior of the wing. In a realistic wing model, the elastic behavior of the wing is not directly controlled by targeted design variables but depends on various quantities such as thickness distribution, wingbox dimensions, and material properties, some of which would also impact other quantities such as the mass properties and natural frequencies. Hence, varying the nondimensional cubic stiffness coefficient while keeping all other nondimensional parameters fixed is not a realistic design modification. However, it is acceptable for this thesis' research objective to test the ability of the LCO constraint combined with the new matrix pencil method to modify the system's dynamics. Overall, the typical section model and aforementioned design variables were chosen due to their simplicity and already proven ability to demonstrate the capability of an optimizer to suppress LCOs with a relatively trivial computational cost [18].



(a) Root locus

(b) Frequency and damping evolution

Figure 3.2: Flutter analysis of the baseline typical section (Ref. [18])

.

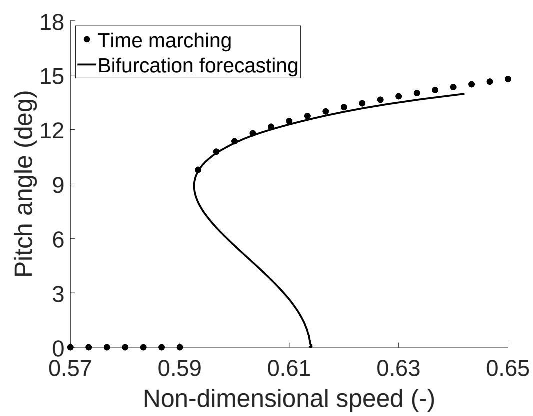


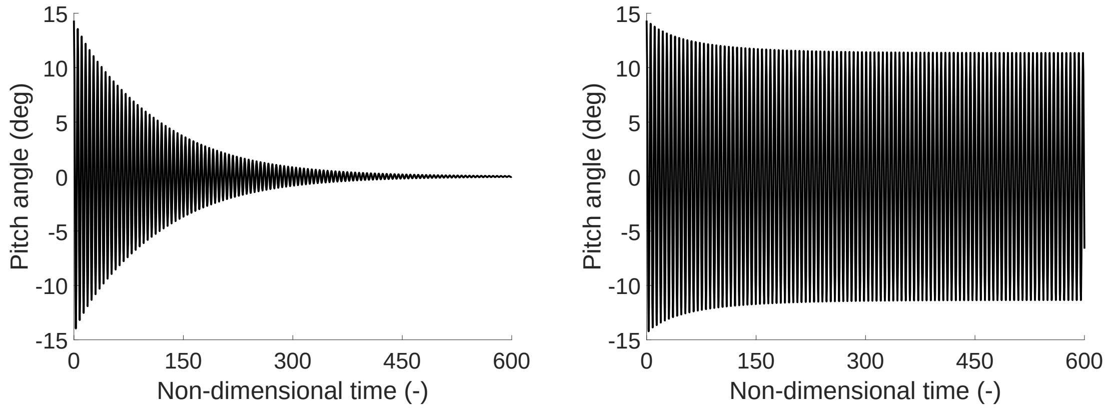
Figure 3.3: Bifurcation diagram of the baseline typical section (Ref. [18]).

CHAPTER 4

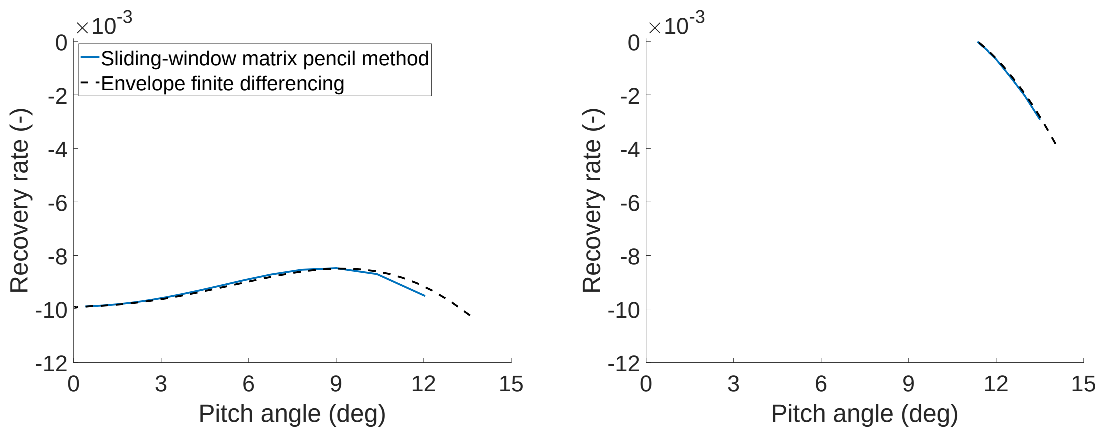
EVALUATION OF SLIDING-WINDOW MATRIX PENCIL METHOD

The sliding-window matrix pencil method is verified with reference recovery rates obtained by finite differencing the envelopes of transient responses obtained via peak fitting. The comparison considers pitch time histories of the baseline model shown in Figure 4.1. Each time history is obtained by generating a transient response applying an initial condition along the real part of the pitch-dominated eigenvector at a given non-dimensional speed, scaled by a factor of 0.25. This choice is based on the knowledge that the pitch-dominated eigenvector is the bifurcating one for this system, but other initial conditions will be investigated in future work. The verification considers the non-dimensional speeds $\overline{U} = 0.5, 0.6$ corresponding to a pre-flutter and an LCO case. The time marching uses a non-stiff differential equation solver with an initial non-dimensional integration time step $\Delta\bar{t} = 0.01$. The resulting time histories are downsampled to a non-dimensional time step of 0.08 before applying the sliding-window matrix pencil method. For an initial demonstration, the method considers an SVD tolerance of 0.1, a sliding window size of 6 peaks, and a stride of 3 peaks (half of the window size). The stride size is chosen to ensure that every output data point is considered in two matrix pencil computations aside from the first and last bounding points. Finally, the verification results use an aggregation parameter of 10^4 to aggregate across the modes within the matrix pencil method itself applied to a specific time window.

Figure 4.1 shows that the sliding-window matrix pencil method captures the recovery rate trend and values. The top plot shows the typical non-monotonic amplitude variation of recovery rates in the pre-flutter regime for a subcritical bifurcation. The plot on the bottom captures the presence of LCOs as seen from the concentrated points around a zero recovery rate value. The method also accurately estimates the amplitude range of the recovery rate. This amplitude was computed for each time window by averaging the amplitude of the first



(a) Pre-flutter case, $\overline{U} = 0.5$



(b) LCO case, $\overline{U} = 0.6$

Figure 4.1: Verification of recovery rates from sliding-window matrix pencil method.

and last peaks. There are slight discrepancies at larger amplitudes, which are associated with faster recoveries and thus a larger amplitude range within a given time window for a fixed window size. However, the obtained amplitudes are sufficiently accurate for this study, as the LCO constraint does not involve any mapping between the recovery rate values and amplitudes at which those values are obtained. However, this mapping would be crucial if the recovery rates were used to predict the bifurcation diagram, which is unnecessary for the LCO constraint.

With the sliding-window matrix pencil method verified, the following subsections present sensitivity studies aimed at identifying optimal sliding-window matrix pencil method parameters for the optimization results. The following parameters are investigated:

- SVD tolerance;
- KS aggregation parameter across modes;
- Window size;
- Stride size;
- Output data sampling step;
- KS aggregation parameter across amplitudes and flight conditions.

4.1 Constraint Sensitivity to Singular Value Decomposition Tolerance

Table 4.1: LCO constraint sensitivity to SVD tolerance (window size = 6 peaks, stride = 1/2 window size, $\Delta\bar{t} = 0.08$).

	$\overline{U} = 0.5$, Reference = -8.483×10^{-3}			$\overline{U} = 0.6$, Reference = 5.665×10^{-5}		
Tol. (-)	Value (-)	Error (%)	Time (s)	Value (-)	Error (%)	Time (s)
1.0×10^{-4}	3.715×10^{-2}	-537.97	0.05	2.336×10^{-3}	4023.63	0.16
1.0×10^{-3}	6.410×10^{-2}	-855.60	0.05	2.185×10^{-3}	3756.63	0.17
5.0×10^{-3}	-8.245×10^{-3}	-2.81	0.05	5.768×10^{-3}	10080.73	0.16
1.0×10^{-2}	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.16
5.0×10^{-2}	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
1.0×10^{-1}	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
2.5×10^{-1}	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.18
5.0×10^{-1}	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.16

As mentioned earlier, the recovery rates from the sliding-window matrix pencil method depend on the SVD tolerance. This quantity indicates the relative contribution of a given sinusoidal component compared with the dominant component associated with the highest singular value. High SVD tolerance values remove additional modes from consideration, potentially discarding key dynamics. Low SVD tolerance values may not filter noise out and thereby give inaccurate results. To understand the impact of this parameter, the KS-aggregated LCO constraint based on the sliding-window matrix pencil method is compared with a reference evaluation for different SVD tolerances. The reference evaluation is based on the recovery rates obtained by applying finite differences to envelope functions associated with the peaks of the time histories at the two non-dimensional speeds above. Table 4.1 shows the results for each case, which are obtained with a KS aggregation parameter of 10^4 for each aggregation step. Past a certain SVD threshold, the LCO constraint no longer changes, and its relative error compared with the reference value is below 0.0001. The high relative errors for the LCO case are due to the extremely low absolute value of the LCO constraint itself. This value is justified considering that the applied perturbation amplitude is higher than the LCO amplitude at the chosen non-dimensional speed, making the transient response converge to the LCO from above. This causes the recovery rate to initially be negative at larger amplitudes and then settle on a zero value once the system develops LCOs. As a result, the KS-aggregated constraint value is practically zero because it is dominated by the zero recovery rate associated with fully developed LCOs.

Note that the LCO constraint converges to a constant error compared with the reference value as the SVD tolerance increases for the LCO case. As mentioned, the LCO constraint achieves a constant value as the SVD tolerance increases due to having few dominant modes in the time histories. When the noise is filtered out, only the dominant modes remain regardless of the SVD tolerance. This behavior may change for higher-dimensional models with multiple active modes, and the SVD tolerance choice may need further investigation in those cases. The converged value of the LCO constraint retains a fixed error compared to the reference due to slight differences in the recovery rates, as the reference is based on recovery rates obtained by applying finite differences to an envelope function. Regardless, the agreement between the recovery rates from the two approaches and the resulting LCO constraints lends credibility to the sliding-window matrix pencil method. The SVD tolerance for the remainder of the analyses in this study is set to 0.1 based on the above sensitivity analyses, and this value is not indicated in the following tables.

The computational times listed are the times required for evaluating the results on a

Table 4.2: LCO constraint sensitivity to KS aggregation parameter across modes (window size = 6 peaks, stride = 1/2 window size, $\Delta\bar{t} = 0.08$).

	$\overline{U} = 0.5$, Reference = -8.483×10^{-3}			$\overline{U} = 0.6$, Reference = 5.665×10^{-5}		
ρ_{KS} (-)	Value (-)	Error (%)	Time (s)	Value (-)	Error (%)	Time (s)
10^1	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
10^2	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.18
10^3	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
10^4	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
10^5	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
10^6	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.16

standard laptop. For reference, the sliding-window matrix pencil method is about five times more computationally expensive than the envelope finite differencing method used to obtain the reference results. However, identifying envelope functions is a challenging task for large-dimensional models [42]. The proposed sliding-window matrix pencil method has the advantage of not involving envelope functions or finite differences. For this reason and despite its increased computational cost, the method is expected to be more suitable for practical optimization.

4.2 Constraint Sensitivity to Aggregation Across Modes

This KS aggregation parameter controls how conservative the matrix pencil method is in estimating the most positive recovery rate across the retained modes. At lower aggregation parameters, the method yields a smoother constraint but is more conservative. At higher aggregation parameters, the method more closely captures the most positive recovery rate but may result in a less smooth constraint. Table 4.2 shows the results for the pre-flutter case and the LCO case. There is no effect from the aggregation parameter across a range between 10 and 10^6, with no error compared with the reference value of the constraint for the pre-flutter case and a constant error for the LCO case (again, due to the very small constraint value). This behavior can be attributed to the large difference between the dominant recovery rate for a given non-dimensional speed and the other values. The aggregation pa-

Table 4.3: LCO constraint sensitivity to window size (stride = 1/2 window size, $\Delta\bar{t} = 0.08$).

	$\overline{U} = 0.5$, Reference = -8.483×10^{-3}			$\overline{U} = 0.6$, Reference = 5.665×10^{-5}		
Peaks	Value (-)	Error (%)	Time (s)	Value (-)	Error (%)	Time (s)
2	-8.445×10^{-3}	-0.46	0.08	2.416×10^{-5}	-57.36	0.13
4	-8.464×10^{-3}	-0.23	0.06	3.968×10^{-5}	-29.97	0.13
6	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
10	-8.498×10^{-3}	0.17	0.07	3.895×10^{-5}	-31.26	0.28
14	-8.635×10^{-3}	1.79	0.09	3.545×10^{-5}	-37.43	0.43
16	-8.740×10^{-3}	3.03	0.12	3.414×10^{-5}	-39.74	0.52
20	-8.951×10^{-3}	5.51	0.14	3.170×10^{-5}	-44.05	0.67
30	-9.347×10^{-3}	10.17	0.18	2.772×10^{-5}	-51.07	1.11

rameter also does not have a significant impact on the computational cost associated with the constraint evaluation for the pre-flutter case and only a slight one in the LCO case. A value of 10^4 is used in the remaining studies, in line with prior work [55]. This, therefore, informs the recovery rate constraint output when applying the matrix pencil method for a specific window.

4.3 Constraint Sensitivity to Window Size

As mentioned in section 2.3, the time window used in the proposed matrix pencil method must be sized appropriately to the transient responses to capture a sufficient number of oscillation cycles and therefore the local amplitude decay. To make the window size insensitive to the characteristic frequencies and decay rates of the problem, it is specified in terms of a user-defined number of peaks. The number of peaks varies from 2 to 30, with the stride size set to half the window size. This choice means that half of the signal in a given window is repeated from the previous window.

Table 4.3 shows that within a range of window sizes from 4 to 10 peaks, the constraint evaluation is relatively invariant and nearly equal to the reference. For a window size of 2 peaks or above 14 peaks, the error starts to markedly increase, with the results for a 30-peak window having a 10% error in the pre-flutter case and a 51% error in the LCO case compared with the reference. This can be attributed to the fact that too large of a window size leads to all recovery rate computations producing the same value, as would be the case if the system were dynamically linear. This is also the case when applying the conventional matrix pencil method that considers an entire time history at once, failing to capture dynamically nonlinear effects. On the other hand, too small of a window size may not contain sufficient peaks to characterize the amplitude decay, resulting in an approximate constraint evaluation.

The computational cost of the LCO constraint evaluation increases markedly with the window size due to the need to compute the SVD of a larger Hankel matrix. Although the implementation seeks to economize this evaluation, it remains the most computationally expensive step in the matrix pencil method, scaling directly with Hankel matrix size. Hence, even if the constraint evaluation involves more instances of the matrix pencil method for smaller window sizes to cover the entire time history, it is more computationally efficient. In the present implementation, peaks below an amplitude of 0.25 degrees are not considered because the recovery rate no longer changes with amplitude once the response oscillates in a small enough amplitude range that the system exhibits a linear behavior. For this reason, the constraint evaluation in the LCO case takes $2 - 6\times$ longer than the one for the pre-flutter case, as the oscillation amplitude recovers to zero in the pre-flutter regime, making the constraint evaluation discard a portion of the time history. In the pre-flutter case, the computational cost of a single evaluation doubles between a window size of 2 and 30 peaks. This scaling factor is even larger in the LCO case due to the need to consider additional peaks. The growth in computational cost with the Hankel matrix size is significant. For this reason, a 6-peak window is chosen for the optimization as it is the smallest window size that yields accurate results for both pre-flutter and LCO cases.

Table 4.4: LCO constraint sensitivity to stride size (window size = 6 peaks, $\Delta\bar{t} = 0.08$).

	$\overline{U} = 0.5$, Reference = -8.483×10^{-3}			$\overline{U} = 0.6$, Reference = 5.665×10^{-5}		
Peaks	Value (-)	Error (%)	Time (s)	Value (-)	Error (%)	Time (s)
2	-8.473×10^{-3}	−0.12	0.08	3.984×10^{-5}	−29.68	0.25
3	-8.475×10^{-3}	−0.11	0.05	4.238×10^{-5}	−25.19	0.17
4	-8.492×10^{-3}	0.10	0.04	4.129×10^{-5}	−27.12	0.12
5	-8.502×10^{-3}	0.22	0.03	3.887×10^{-5}	−31.39	0.10
6	-8.475×10^{-3}	−0.10	0.03	3.711×10^{-5}	−34.50	0.08

4.4 Constraint Sensitivity to Stride Size

The number of peaks between the start of adjacent windows is also an important parameter. Too large of a stride size can result in the method not accurately capturing a time interval associated with lower recovery rates. As such, a study was conducted on the stride size for a 6-peak window as selected from the prior study. As in the previous results, the SVD tolerance is set to 0.1 and the KS-aggregation parameter is set to 10^4.

Table 4.4 indicates the relatively negligible effects of the stride size on the constraint evaluation. All pre-flutter cases exhibited the same behavior even with no overlap between windows at a stride size of 6 peaks. However, the most accurate result seen for the LCO case was at a stride size of 3 peaks. This is also ideal as it allows for every point in the time history to be considered in two subsequent recovery rate computations except the first half of the first window and the last half of the last window. For this reason, a stride of 3 peaks was chosen for the overall optimization. This choice also reduces the computational cost of the constraint evaluation, which roughly halves as the stride size increases due to the reduced overlap between different time windows, resulting in fewer instances of the matrix pencil method over the time history.

Table 4.5: LCO constraint sensitivity to output data sampling step (window size = 6 peaks, stride = 1/2 window size).

	$\overline{U} = 0.5$, Reference = -8.483×10^{-3}			$\overline{U} = 0.6$, Reference = 5.665×10^{-5}		
$\Delta\bar{t}$	Value (-)	Error (%)	Time (s)	Value (-)	Error (%)	Time (s)
0.01	-8.475×10^{-3}	-0.11	2.28	4.239×10^{-5}	-25.18	13.31
0.04	-8.475×10^{-3}	-0.10	0.13	4.238×10^{-5}	-25.19	0.58
0.08	-8.475×10^{-3}	-0.11	0.05	4.238×10^{-5}	-25.19	0.17
0.10	-8.475×10^{-3}	-0.10	0.04	4.229×10^{-5}	-25.34	0.13
0.12	-8.475×10^{-3}	-0.10	0.04	4.237×10^{-5}	-25.20	0.11
0.20	-8.474×10^{-3}	-0.11	0.03	4.057×10^{-5}	-28.39	0.07
0.30	-8.473×10^{-3}	-0.12	0.03	4.177×10^{-5}	-26.27	0.05

4.5 Constraint Sensitivity to Output Data Sampling Step

All transient responses used to evaluate the LCO constraint are integrated using a non-dimensional time step of 0.01, which proved small enough to ensure a smooth time history that accurately represents the actual behavior of the system. However, this time step corresponds to too high of an output data sampling rate for the matrix pencil method. The matrix pencil method develops a Hankel matrix based on the output data and thus, the smaller the time step, the less the difference between adjacent output data samples and therefore the less descriptive of the system's change in state it is for a given window size. Thus, to test the limits of the downsampling possible, the LCO constraint evaluation is repeated for varying output data sampling steps. Note that, for all cases, the time step used to integrate the transient responses is still 0.01. The different output data sampling rates are achieved by resampling the integrated transient responses with a lower rate for the same integration time step. The results use a fixed window size of 6 peaks and a stride of 3 peaks, a combination that was previously shown to produce accurate results in the prior sections. The other parameters are fixed to the previously identified values. The time step between consecutive output data samples is varied from 0.01, the integration time step, to 0.3 to investigate the optimal compromise between LCO constraint accuracy and computational efficiency.

Table 4.5 shows that past a sampling step of 0.12, the error in the matrix pencil method increases. This is explained considering that, for a larger sampling step, there is too large of a gap between adjacent output data samples, making the data too granular for the method to accurately perform the SVD and thereby properly estimate the recovery rate. In this example, a single period of oscillation for the dominant sinusoid for this model is approximately 0.6. Time steps larger than 0.15, corresponding to four output data points between consecutive peaks, are too granular to accurately describe the dynamics. Looking at the effect of the output data sampling step on the computational cost, it is apparent that increasing the granularity of the data results in a computationally faster constraint evaluation. Between an output data sampling step of 0.01 and 0.08, the computational cost reduces by a factor of 50 for the pre-flutter case and by almost two orders of magnitude for the LCO case. As such, to avoid an excessively granular signal while drastically improving the computational efficiency of the constraint evaluation, a time step of 0.08 is chosen for the following optimization. This output data sampling step corresponds to 45 data points within the chosen 6-peak window size and about 8 points between consecutive peaks. Having about 8–10 points between consecutive peaks can be reasonably assumed as a representative guideline for other applications to accurately resolve a full oscillation cycle of the dominant sinusoid.

4.6 Constraint Sensitivity to Aggregation Across Amplitudes and Flight Conditions

The final sensitivity study considers a constraint evaluation for multiple non-dimensional speeds, representative of those used in the subsequent optimization. The aggregation parameter $\rho_{KS_{\mathrm{LCO}}}$ used to aggregate the results across amplitudes and non-dimensional speeds informs the final constraint value passed to the optimizer. The chosen non-dimensional speed range of interest is $\overline{U} \in (0, 0.5]$ based on the optimization study of Ref. [18], which is revisited in this work. This non-dimensional speed range is sampled with a non-dimensional speed step of 0.02. The non-dimensional speeds are all in the pre-flutter regime of the baseline model.

Table 4.6 shows a significant impact of the aggregation parameter. Low parameter val-

Table 4.6: LCO constraint sensitivity to KS aggregation parameter across amplitudes and non-dimensional speeds (window size = 6 peaks, stride = 1/2 window size, $\Delta\bar{t}$ = 0.08).

	Reference	Present		
$\rho_{\mathrm{KS}_{\mathrm{LCO}}}$	Value	Value	Diff. (%)	Time (s)
10^1	8.203×10^{-1}	6.553×10^{-1}	-20.11	16.38
10^2	7.591×10^{-2}	5.980×10^{-2}	-21.22	16.23
10^3	4.492×10^{-3}	3.050×10^{-3}	-32.10	16.42
10^4	-1.337×10^{-3}	-1.473×10^{-3}	10.12	16.50
10^5	-1.838×10^{-3}	-1.869×10^{-3}	1.67	16.52
10^6	-1.862×10^{-3}	-1.890×10^{-3}	1.47	16.53

ues give conservative results, leading to a constraint value significantly larger than zero. As the parameter increases by several orders of magnitude, the constraint approaches the reference value. Once again, there is a constant difference even at high aggregation parameters, which can be attributed to the differences in the methods used for the recovery rate computation. There is no significant impact on the computational cost as the aggregation parameter changes. For this reason, a large aggregation parameter of 10^5 was chosen to ensure appropriate constraint values in the optimization. Again, for reference, this aggregation parameter informs all aggregation done outside of the matrix pencil method itself (across different windows for a single time history and across time histories at different flight conditions). This value should not yield a design space that is excessively sharp while still allowing an order of magnitude of buffer from where the results converged.

4.7 Application Guidelines

This section summarizes the main trends for the sensitivity studies, providing guidelines for future applications of the proposed sliding-window matrix pencil method. The main trends observed are as follows:

- **SVD tolerance:** the constraint evaluation at a fixed non-dimensional speed no longer changes once the normalized SVD tolerance increases to a large enough value to discard noise while retaining the dominant dynamics. A value of 0.1 appears suitable for this study, in line with prior work on the matrix pencil method;

- **KS aggregation parameter across modes:** the parameter does not affect the LCO constraint evaluation at a fixed non-dimensional speed, likely due to the large difference between the most positive recovery rate at that flight condition and the other recovery rates. A value of 10^4 is chosen for this study, in line with prior work on the matrix pencil method;

- **Window size:** there is an optimal window size for obtaining accurate LCO constraint evaluations at a fixed non-dimensional speed, with smaller window sizes failing to capture the amplitude decay and larger window sizes averaging the recovery rates in addition to requiring much higher computational cost. The optimal window size chosen for this study includes 6 oscillation peaks;

- **Stride size:** the stride size does not affect the LCO constraint evaluation for pre-flutter non-dimensional speeds, but using half the window size gives the best results for LCO cases along with lower computational cost than smaller stride sizes;

- **Output data sampling step:** the output data sampling step trades off higher accuracy for smaller values and higher computational efficiency for larger values. A value of 0.08 corresponding to approximately 8 data points per oscillation cycle is suitable for this study;

- **KS aggregation parameter across amplitudes and flight conditions:** the parameter significantly affects the LCO constraint evaluation, which can be attributed to the continuous variation of the aggregate recovery rate over the modes with the amplitudes and flight conditions. A value of 10^5 is chosen for this study, aligned with Ref. [18] and leading to a converged constraint.

CHAPTER 5

OPTIMIZATION DEMONSTRATION

This section presents optimization results demonstrating the LCO constraint based on the new sliding-window matrix pencil method.

5.1 Optimization Statement

The optimization statement and parameters used for demonstration are shown in Table 5.1 and Table 5.2 and match prior work by Riso et al. [18]. The optimization problem varies the mass ratio $\overline{m}$ and the non-dimensional cubic stiffness coefficient $\kappa_\alpha^{(3)}$ subject to the flutter constraint, the LCO constraint, and side constraints. The objective function is constructed such that the primary focus of the optimizer is on driving the mass ratio down while it is free to vary the cubic stiffness coefficient in either direction, as the function considers the squared change relative to the original value. This composite function aims to mimic a mass minimization problem while also adding a penalty for changing the bifurcation diagram shape by varying the non-dimensional cubic stiffness coefficient, which is the primary design variable affecting the LCO constraint. Additional discussions on the significance of the optimization statement are given in Ref. [18]. This optimization statement should not be interpreted as a practical optimization problem aiming to produce a realistic design, but

Table 5.1: Optimization statement.

	Function/Variable	Description
minimize	$W_1 x_1 + W_2 \Delta x_2^2$	Objective function
with respect to	$x_1 = \overline{m}$	Mass ratio
	$x_2 = \kappa_\alpha^{(3)}$	Non-dimensional cubic stiffness coefficient
subject to	$\mathrm{KS}_{\text{flutter}}(x_1) \leq 0$	KS aggregate of damping
	$\mathrm{KS}_{\text{LCO}}(x_1, x_2) \leq 0$	KS aggregate of recovery rates

Table 5.2: Optimization problem parameters.

Variable/Parameter	Description	Range/Value
$\overline{m}$	Mass ratio range	[5, 15]
$\kappa_\alpha^{(3)}$	Non-dimensional cubic stiffness coefficient range	[−8, 0]
W_1	Weight applied to $\overline{m}$	1.0
W_2	Weight applied to $\Delta\kappa_\alpha^{(3)}$	0.001
$\overline{U}$	Flutter and LCO constraint speed range	$(0, 0.5]$
$\Delta\overline{U}$	Flutter and LCO constraint speed increment	0.02
$\rho_{\mathrm{KS}_{\mathrm{flutter}}}$	Flutter constraint aggregation parameter	1000
$\rho_{\mathrm{KS}_{\mathrm{LCO}}}$	LCO constraint aggregation parameter	100000

Table 5.3: Sliding-window matrix pencil method parameters.

Parameter	Value
SVD tolerance	0.1
Window size (peaks)	6
Stride size (peaks)	3
Output data sampling step	0.08
KS aggregation parameter across modes	10000

simply as a demonstration problem allowing one to test the new developments.

Because the typical section model and optimization statement are the same as in the prior work, the results obtained by the optimizations conducted for this work are directly comparable to Ref. [18]. This choice is on purpose, as the goal is to highlight the impact of using a different method to evaluate the recovery rates within the LCO constraint. The LCO constraint evaluation based on the sliding-window matrix pencil method uses the optimal parameters identified through the sensitivity studies in the previous section, shown in Table 5.2 and Table 5.3. The optimization results are obtained by differentiating the LCO constraint using finite differences, as done in Ref. [18]. This approach is sufficiently accurate and computationally efficient for the small-scale optimization problem considered for demonstration.

The optimizer uses the sequential quadratic programming (SQP) algorithm. This algorithm was chosen due to its robust ability to find minimizers for highly nonlinear prob-

Table 5.4: Optimizer parameters.

Parameter	Value
Algorithm	Sequential quadratic programming
Absolute optimality tolerance	10^{-9}
Absolute step size tolerance	10^{-8}
Absolute constraint tolerance	10^{-10}
Constraint functions gradients	Finite difference
Objective function gradient	Analytical
Finite difference type	Central
Finite difference relative step size	10^{-4}

lems. Within this method, at each iteration, a quadratic subproblem is defined based on the Hessian of the Lagrangian function updated with a quasi-Newton method. This quadratic subproblem is used to determine a search direction along which a line search is performed. The optimizer involves several parameters informing the convergence behavior, listed in Table 5.4. The first is the optimality tolerance, which is the maximum of two functions. The first function describes a metric of the magnitude of the Lagrangian function as defined considering the objective function and inequality constraints. The second computes the maximum product of any Lagrange multiplier with its corresponding constraint value. As such, at a feasible point, this optimality measure should be almost identically zero. The second tolerance value is the step size tolerance. This establishes a lower bound on the size of the step that can be taken to move in the design space and allows for the optimizer to exit its routine if the design variables no longer change. The step size is calculated as the norm of the step taken in each direction along the aforementioned line search defined through the SQP algorithm. The final tolerance is the constraint tolerance. This specifies how tightly the constraints must be satisfied. In this study, the constraint value can be, at most, 10^{-10} greater than zero for the constraint to be satisfied.

All gradient-based optimizations require computing the gradient of the constraint and objective functions with respect to the design variables. Due to its simple formulation, the objective function can be differentiated analytically with respect to the two design vari-

Table 5.5: Optimized results and comparison with previous work.

	Optimized solution			
LCO constraint method	$W_1\, x_1 + W_2\, \Delta x_2^2$	$x_1 = \overline{m}$	$x_2 = \kappa_\alpha^{(3)}$	Iterations
State velocity (Ref. [18])	6.975	6.970	-1.621	10
Sliding-window matrix pencil	6.971	6.970	-2.853	21

ables. However, analytical derivatives are not readily available for the flutter and LCO constraints. As such, derivatives of these constraints are computed using a finite-difference scheme to inform the SQP subproblem definition at each iteration. Specifically, the constraints are differentiated using a central-difference scheme, which is second-order accurate as opposed to a first-order accuracy for forward or backward differentiation. This does, however, come at the cost of additional function evaluations for each design iteration and variable added to the problem. It was deemed that for this two-variable demonstration, central differencing would not significantly impact the computational cost of the small-scale demonstration optimization problem, and thus is chosen for its higher accuracy. The relative step size of this finite-difference scheme is set to 10^{-4} based on sensitivity studies conducted in prior work [18].

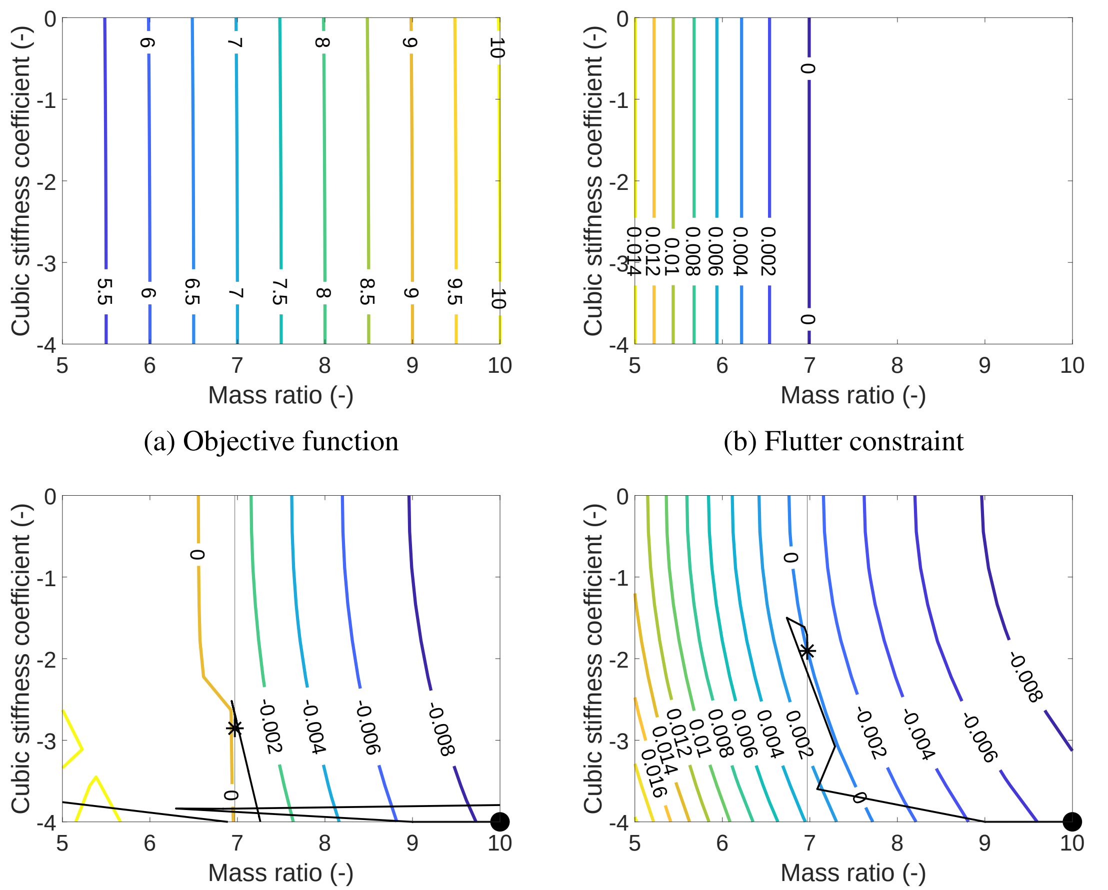
5.2 Optimization Results and Postoptimality Study

This section compares the optimization results obtained by evaluating the LCO constraint using the current sliding-window matrix pencil method with the results obtained in previous work [18], based on the previous approximate recovery rate evaluation method [54]. As mentioned when introducing the sliding-window approach, the previous method uses the local state matrix of the system at a given flight condition and its right eigenvectors to define a perturbed dynamic state along a given mode and phase. The resulting state velocity is then evaluated and projected onto the left eigenvectors to approximate the recovery rate, and the process is repeated for different operating conditions, modes, and amplitudes to evaluate the LCO constraint. This approach bypassed the need for transient simulations, resulting in a computationally efficient constraint evaluation. However, it was found to conservatively approximate the recovery rates for larger amplitudes, resulting in an improper LCO constraint estimate. While the sliding-window matrix pencil method requires transient simulations, it is expected to capture the true LCO constraint value more closely. Additionally, due to the advantageous features of the matrix pencil method, this approach is expected to more easily scale to more complex computational models with generic types of nonlinear behaviors.

Table 5.5 compares the results from the two approaches. The optimizations converge to the same x_1 or $\bar{m}$ value, and the optimized designs flutter at a non-dimensional speed $\overline{U} = 0.508$. This is expected because the objective function assigns a higher penalty to shifting the flutter point (by decreasing the mass ratio) than to reshaping the bifurcation diagram (by changing the cubic stiffness coefficient). Furthermore, the flutter constraint is only influenced by $\bar{m}$ and is independent of how the LCO constraint is evaluated. However, because the LCO constraint depends on both $\bar{m}$ and $\kappa_\alpha^{(3)}$, the optimized value of $\kappa_\alpha^{(3)}$ changes depending on the method used to evaluate the recovery rate. The reduced mass ratio of the optimized solution compared to the baseline can be understood to be the primary factor behind the reduced flutter point. The smaller magnitude of the $\kappa_\alpha^{(3)}$ value in magnitude indicates that the bistable range of the subcritical bifurcation diagram is reduced, as observed by comparing the optimized bifurcation diagram in Figure 5.2 with the baseline bifurcation diagram in Figure 3.3. The optimized value of this design variable is more negative for the current results compared with the previous results, which can be attributed to the previous recovery rate evaluation misrepresenting the true dynamic behavior of the system in the LCO constraint due to the conservative estimation of the recovery rates.

To further understand the differences in the optimization results, the design spaces associated with the two recovery rate evaluation methods are investigated. This is done using contour lines of the objective function, flutter constraint, and LCO constraint as functions of the two design variables (see Figure 5.1). For simplicity, the plots are based on the LCO constraint evaluated at $\overline{U} = 0.5$, which captures the most violated recovery rates. Note that the objective function and the flutter constraint are independent of the method used to evaluate the recovery rates. As a result, they are the same in the previous results of Ref. [18] and in the current results. The bottom plots highlight the contour plots for the LCO constraint evaluated with either method, with a vertical line indicating the optimized mass ratio. The optimizer's path towards the constrained solution is shown on the LCO constraint plot based on the sliding-window matrix pencil method. These optimizer's paths are also computed by limiting the LCO constraint evaluation to $\overline{U} = 0.5$, consistent with the contour lines, but the qualitative behavior is representative of the optimizations based on all non-dimensional speeds. Note that the black circles and asterisks in the bottom right corner of each LCO constraint plot represent the starting point and optimized point, respectively.

Figure 5.1 shows that while the optimizer does make several large steps away from the true minimum, it is still able to converge to the optimal solution. The difference in optimized results can be explained by looking at the contours of the objective function. The optimizer attempts to drive the mass ratio down since the function value is only slightly affected by the second design variable. There is, however, a limit indicated by the zero contour line in the flutter constraint, which is independent of the second design variable $\kappa_\alpha^{(3)}$. This boundary is set at approximately the 6.970 value for $\bar{m}$ in both optimized results, as both optimizations satisfy the flutter constraint by reaching the minimum feasible value of the first design variable. There are notable differences in the LCO constraint contour plot for the two evaluation methods. The result of these differing LCO constraint functions is that, at the optimal mass ratio, there is less curvature for the constraint function based on the sliding-window matrix pencil method, leading to a lower optimized $\kappa_\alpha^{(3)}$ value. The active constraint contour is seen to intersect the vertical line at approximately the optimized $\kappa_\alpha^{(3)}$ value of -2.853 for the new constraint evaluation and approximately -1.905 in the previous results, confirming the convergence to the true minimum of the design space. For



(a) Objective function

(b) Flutter constraint

(c) LCO constraint from matrix pencil method

(d) LCO constraint from state velocity method

Figure 5.1: Design space contour lines (black circle = baseline, black asterisk = optimized, black lines = iterations).

the new constraint evaluation, this value coincides with the one in Table 5.5 based on the optimization considering all non-dimensional speeds. For the previous constraint evaluation, limiting the constraint evaluation to $\overline{U} = 0.5$ when plotting the design space gives a slightly less conservative result, but the behavior remains qualitatively similar. The almost vertical behavior of the zero contour line for the LCO constraint based on the sliding-window matrix pencil method is due to the nature of the analysis. Within an approximate range of values from -2 to -5 for $\kappa_\alpha^{(3)}$ at around $\overline{m} = 6.970$, the system's transient responses that develop into an LCO converge to an LCO amplitude smaller than the applied initial condition. This causes the aggregated LCO constraint to always achieve a value of practically zero regardless of the value of $\kappa_\alpha^{(3)}$ because the recovery rate is initially negative at the beginning of the transient responses, hence the aggregate constraint is dominated by the zero recovery rate associated with LCOs. This is not the case for largely negative $\kappa_\alpha^{(3)}$ values at a constant $\overline{m}$. For these points of the design space, the bistable range is large enough that LCO amplitude grows past the initial condition, resulting in a positive constraint value. These behaviors are not observed in the contour plot based on the previous evaluation method, which did not involve transient simulations.

Both optimizations produce the same mass ratio. However, the LCO constraint in the prior work overestimated the change in $\kappa_\alpha^{(3)}$ required to satisfy the LCO constraint due to the conservative nature of the method to estimate recovery rates. The present method to evaluate the recovery rate provides a less conservative solution associated with a wider feasible bistable range in the bifurcation diagram and a reduced change in the value of $\kappa_\alpha^{(3)}$ compared with the baseline design.

Figure 5.2 shows the bifurcation diagrams for the optimized design obtained without the LCO and with the LCO constraint evaluated using the present sliding-window matrix pencil method and the previous state velocity method. The black vertical line indicates the non-dimensional speed below which flutter and LCOs are not allowed based on the specified constraints. The bifurcation diagrams are obtained using the bifurcation forecasting

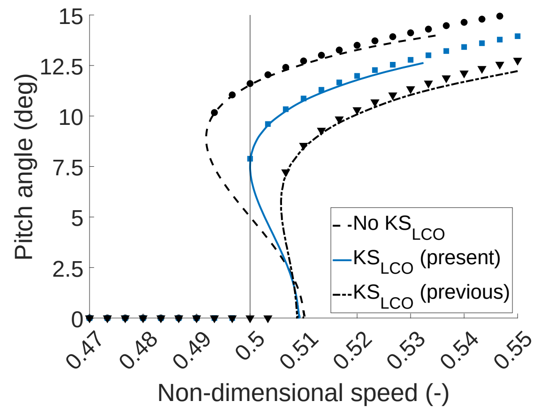


Figure 5.2: Optimized bifurcation diagrams.

method and verified with time-marching results. Note that the bifurcation diagrams are only presented for a post-optimality study but not used in the optimizations with the LCO constraint, as the formulation only uses the recovery rates. The bifurcation diagram obtained without using the LCO constraint in the optimization shows a bistable range that enters the non-dimensional speed range below 0.5. The bifurcation diagram obtained using the LCO constraint with the present method shows a slight subcritical behavior associated with a bistable range that starts at about $\overline{U} = 0.5$, as shown by the jump in the time-marching results within both plots. This is consistent with the LCO constraint requiring no LCOs above that non-dimensional speed, as the fold point in the bifurcation diagram exactly touches the vertical line, indicating the upper bound of the non-dimensional speed range where LCOs are not allowed. The bifurcation diagram obtained using the LCO constraint based on the previous method has a narrower bistable range starting at a non-dimensional speed past $\overline{U} = 0.5$, aligned with the conservative constraint evaluation.

The time required to compute the LCO constraint using the sliding-window matrix pencil constraint was approximately twice the one to evaluate the LCO constraint based on the state velocity method. This increased computational cost does come, however, with higher accuracy in characterizing the variation of the recovery rates with amplitude and therefore in capturing critical constraint values. The bifurcation diagrams in Figure 5.2 show that the state velocity method overshoots the minimum non-dimensional speed for

LCOs to be acceptable by approximately 1% in comparison to the sliding-window matrix pencil method, which perfectly matches the constraint. Overall, while the computational cost would likely scale and become notably different as the model's complexity increases, the novel approach promises to yield less conservative designs. Additionally, the sliding-window matrix pencil method is entirely based on time histories and therefore applicable to any computational model suitable for time-marching integration with no change in the methodology or implementation. This proof of concept demonstrates the ability of this new method to capture recovery rates and its viability for use in optimization to prevent undesirable LCOs in a desired range of operating conditions.

CHAPTER 6

CONCLUDING REMARKS AND OUTLOOK

This chapter ends the thesis by summarizing the key contributions and providing recommendations for future research.

6.1 Key Contributions

This thesis offered a new approach to constraining LCO in aeroelastic design optimization. The proposed approach builds on a previously developed LCO constraint that bounds the recovery rate to equilibrium at chosen flight conditions. In prior work, the recovery rates necessary to evaluate the LCO constraint were approximated using a projection-based method based on the system's state velocity. This method was observed to lose accuracy at larger amplitudes and has only been demonstrated on systems with polynomial nonlinearities. Toward larger-scale aeroelastic design optimization problems, this thesis advanced the prior work by proposing a new method to evaluate the recovery rates within the LCO constraint. The new method preserves accuracy at large amplitudes and is expected to better generalize to various types of nonlinear behaviors as well as to large-scale computational models of interest for practical design applications. The proposed method consists of a new matrix pencil formulation that considers a sliding time window along the transient response for a chosen quantity of interest. Using a sliding time window instead of considering the entire transient response overcomes the linearized assumptions of the conventional matrix pencil method, which cannot capture amplitude-varying effects necessary to characterize the recovery rate. Through a series of sensitivity analyses, the new sliding-window matrix pencil method was proved to sufficiently and accurately estimate recovery rate values. Additionally, guidelines were identified for future applications of the new method in either LCO prediction (not addressed in this work) or optimization. Finally, the LCO constraint based on the new sliding-window matrix pencil method with optimal parameters was demonstrated by revisiting a previous optimization problem for a typical section with an LCO constraint based on the same formulation but with approximate recovery rates. The new method allowed the optimizer to prevent subcritical LCOs at chosen operating conditions. Given the less conservative nature of the new recovery rate evaluation method relative to the method used in previous work, the optimizer minimized the changes to the design variables while satisfying all constraints, achieving a less conservative optimized design solution.

To summarize, the key new contributions from this thesis are listed below:

- A new sliding-window matrix pencil method able to resolve amplitude-varying effects associated with dynamic nonlinearities that are neglected in the conventional matrix pencil method, which uses linearized assumptions;
- A demonstration of the sliding-window matrix method for a typical section model, including extensive sensitivity analyses to identify application guidelines;
- An LCO-constrained optimization of the typical section model subject to the LCO constraint based on the sliding-window matrix pencil method, demonstrating its feasibility for aeroelastic design optimization and advantages compared with a prior work that used an approximate method to evaluate the LCO constraint.

These contributions can also be leveraged in the areas of LCO prediction or for optimization of systems experiencing LCOs arising in other areas of engineering beyond aeroelasticity.

6.2 Recommendations for Future Research

While the results achieved in this thesis demonstrate the effectiveness of the proposed sliding-window matrix pencil method for evaluating the LCO constraint, there are several areas for future research. First, the methodology must be demonstrated for larger-scale optimization problems involving more design variables. As an initial step to address this, the optimization framework used for the present results, implemented into MATLAB, has been transferred to a Python-based framework based on OpenMDAO [57]. The framework has already been developed with all necessary components defined and implemented. Additionally, the framework has been verified by reproducing the results from the original MATLAB implementation. The framework will facilitate defining more complex optimization problems and, eventually, using more sophisticated models. This will require the development of adjoint derivatives for the flutter constraint, available in the literature, and for the LCO constraint based on the new sliding-window matrix pencil method, which must be developed. Adjoint derivatives will enable scaling the methodology to large-scale aeroelastic design optimizations of practical interest. By implementing adjoint derivatives, the number of function evaluations currently necessary using finite-different derivatives will be reduced, and the overall computational cost of the optimization minimized to allow this to be as computationally efficient of a tool as possible. Finite-difference derivatives suffer from numerical truncation errors, which may mislead the optimizer. These errors are eliminated in the adjoint method, potentially allowing for faster convergence.

Additional efforts could focus on replacing the standard time-marching algorithm used in this work with capabilities provided by the Dymos [58] framework built upon OpenMDAO. The collocation method typically used by Dymos for solving a trajectory problem, such as that involved in generating a time history, assumes that both the state itself and the first derivative are continuous. With small enough time steps, this can be assumed to be true. The collocation method then uses the state and the state derivative information to fit a polynomial to the state information at a set of particular values. This is unreasonable for a highly oscillatory system such as the ones characteristic of aeroelastic time histories, especially for long simulations. The polynomial would be unreasonably high-order, requiring an unnecessarily large computational cost to fit appropriately. Dymos has preliminary support for explicit shooting methods, where the state value at a given time step and the resultant state velocity are used to develop the state at the subsequent time step. Further work on these methods is underway.

The longer-term goals of this effort are to demonstrate the methodology for aerodynamic models of increasing physical and computational complexity while keeping the current typical section structural model. Candidate aerodynamic models are the theory of Peters et al. [56] and two-dimensional Euler equations or Navier-Stokes equations, where the typical section model will be coupled with a computational fluid dynamics solver. An additional improvement is to move from a two-dimensional problem to a three-dimensional one, considering a finite wing, allowing for a more relevant and real-world use-case of the optimization framework developed. This extension requires a more sophisticated structural model, which could initially be a root-clamped, nonlinear beam model. The ultimate goal of the design optimization methodology investigated in this work is to optimize the detailed internal structural sizing of the wing, considering a built-up finite element model, along with its geometry and aerodynamic shape.

APPENDIX A: OPENMDAO FRAMEWORK STRUCTURE AND CONNECTIVITY

The original implementation of the optimization framework and demonstration problem discussed in this thesis was in MATLAB. The basic functionality of this framework, specifically the key components required for the optimization of the typical section presented in chapter 3, have been converted to a Python-based implementation in OpenMDAO [57] as part of this thesis. OpenMDAO is an open-source framework allowing for various optimization problems to be developed with analytic derivatives. This framework will be key for developing adjoint derivatives for future computational speedup. It therefore allows for quick and robust design optimization on a problem with defined dynamics. This, in turn, allows for developing high-fidelity optimization with modular components.

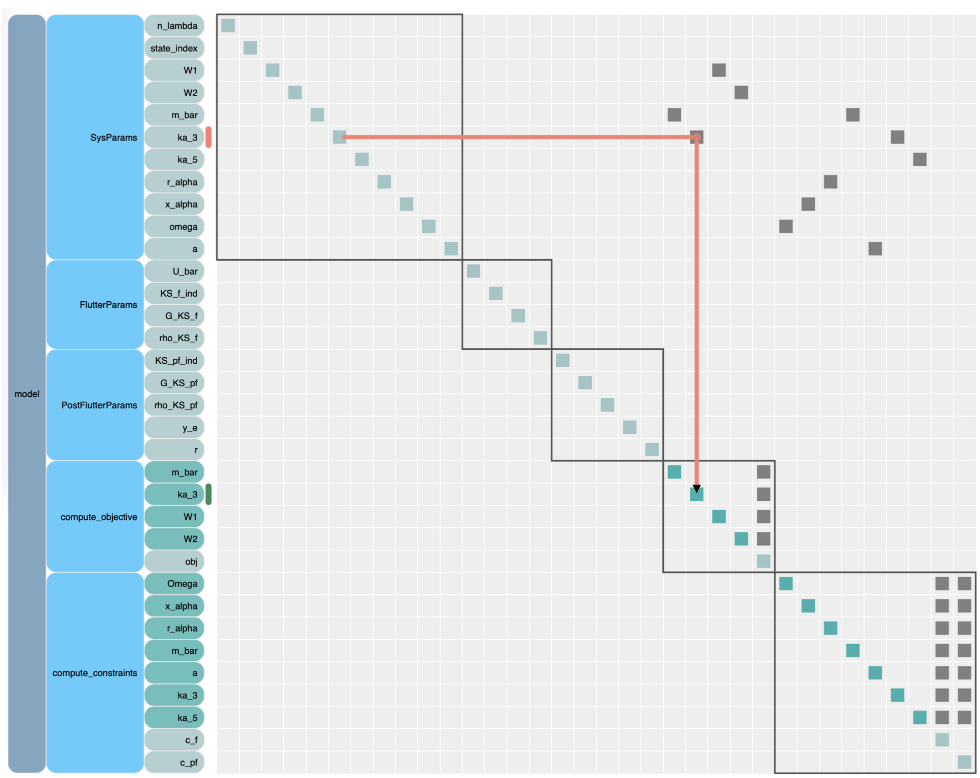


Figure 6.1: N2 diagram of the OpenMDAO implementation.

Figure 6.1 shows the connectivity of the OpenMDAO implementation. The overall structure can be described as a set of components. The first three components are defined as independent variable components (IVCs). These variables are defined externally to the model and thus are analogous to inputs to the design optimization problem. The three IVCs used in this implementation are as follows:

1. **SysParams** contains the number of states in the problem of interest, the weights on the objective function, and parameters describing the typical section model;

2. **FlutterParams** contains parameters specific to the computation of the flutter constraint, which include the flight condition range evaluated, the aggregation parameter, and any necessary bounding curve information;

3. **PostFlutterParams** contains parameters specific to the LCO constraint, such as the equilibrium state, the aggregation parameter, and the flight condition range over which the LCO must be eliminated.

The variables present in these components are passed into the compute_objective and compute_constraints modules. These modules are defined as explicit components in OpenMDAO. These components allow for a user-defined computation to be performed and outputs to be passed on. The two explicit components are described as follows:

1. **compute_objective** computes the objective function value based on the weights and design variable values from the above IVCs, which is passed into the optimizer and differentiated analytically;

2. **compute_constraints** contains both the flutter and post-flutter constraint evaluations, with the sliding-window matrix pencil implementation used for the LCO constraint. The output constraint values are passed directly to the optimizer and their derivatives are computed using central finite differencing with the same relative step size, 10^{-4}, as used in the original MATLAB implementation.

Since each of these is a standalone component and an individual Python class, either of the compute_objective or compute_constraints modules can be replaced quickly with the modification of several lines of code defining the problem and variable connectivity. Thus, for a more complex problem, once these individual components are established, they can be easily swapped in for robust, modular analysis and optimization.

The gray squares in the upper-right corner of the N2 diagram indicate points of connectivity. The example highlighted in Figure 6.1 shows the cubic stiffness coefficient, given by κ_α^3, connected from the SysParams component to the compute_objective component. The second gray square in this line indicates that this same value is connected to the compute_constraints component as well. Since the design variables of interest are κ_α^3 and $\overline{m}$, these are defined in a single location where they can be modified by the optimizer before being passed to the objective and constraint evaluations.

The constraint definition in OpenMDAO allows for specifying an upper and lower bound on the constraint value. However, if both an upper and lower constraint bound are specified, the optimizer does not converge. Note that the constraint sign convention used for MATLAB and OpenMDAO implementations is that a negative constraint value indicates that the constraint is satisfied. Therefore, a positive constraint value indicates that the constraint is violated. Thus, no lower bound was specified for the demonstration problem, whereas the upper bound was specified as positive 10^{-6} such that a design with an extremely small violation would still be accepted.

Additionally, the optimizer used in this case is the Sequential Least-Squares Quadratic Programming (SLSQP) optimizer within the SciPy Optimize Driver. This optimizer is similar to MATLAB's SQP optimizer, which generates and solvers a quadratic subproblem at each iteration. With all these components connected and the optimizer driver being run with the same initial conditions as the MATLAB implementation, the optimized κ_α^3 and $\overline{m}$ were -2.857 and 6.970 respectively. These values closely agree with the MATLAB implementation, thereby verifying the OpenMDAO implementation on the same design problem. The slight discrepancies can be attributed to differences in the time-marching implementations and optimizers between the two languages, contributing to slightly different computed recovery rates and resulting LCO constraints, which affect the derivatives and the final convergence to a slight extent.

APPENDIX B: THREE-VARIABLE OPTIMIZATION DEMONSTRATION

To further demonstrate the methodology, this work also solved a three-variable optimization problem. Of the variables of interest in this typical section model, those with contributions to the mass matrix are especially interesting because they affect both the flutter and LCO constraints. The three variables that contribute in this manner are r_α, x_α, and a. Out of these, the elastic axis offset, a, was chosen due to its relevance. Changes in this quantity can be caused by modifying the airfoil shape or the wingbox geometry, sizing, or materials, all of which are practical design modifications.

The optimization problem was formulated similarly to the two-variable case, with a added as a design variable. The bounds for the optimizer were set to -0.31 and -0.29 for the lower and upper bounds respectively, including the baseline value -0.3. The objective function was modified by adding the contribution for the new variable as done for the cubic stiffness coefficient. The added contribution is $W_3\,\Delta x_3^2$ with $W_3 = W_2 = 0.001$. The remainder of the bounds, initial conditions, analysis methods, and analysis parameters were left identical to the two-variable case.

The results in Table 6.1 show the difference between the two and three-variable optimizations. The two entries for the three-variable case represent the results obtained with the $W_3\,\Delta x_3^2$ component excluded and included in the objective function. Adding the elastic axis offset as a design variable allowed the optimizer to decrease the objective function

Table 6.1: Optimized results for the two- and three-variable cases.

	Optimized solution				
Case	Objective func.	$x_1 = \overline{m}$	$x_2 = \kappa_\alpha^{(3)}$	$x_3 = a$	Iterations
Two-variable	6.971	6.970	-2.853	-0.30	18
Three-variable, No obj.	5.953	5.950	-2.344	-0.31	25
Three-variable, Obj.	5.953	5.950	-2.344	-0.31	25

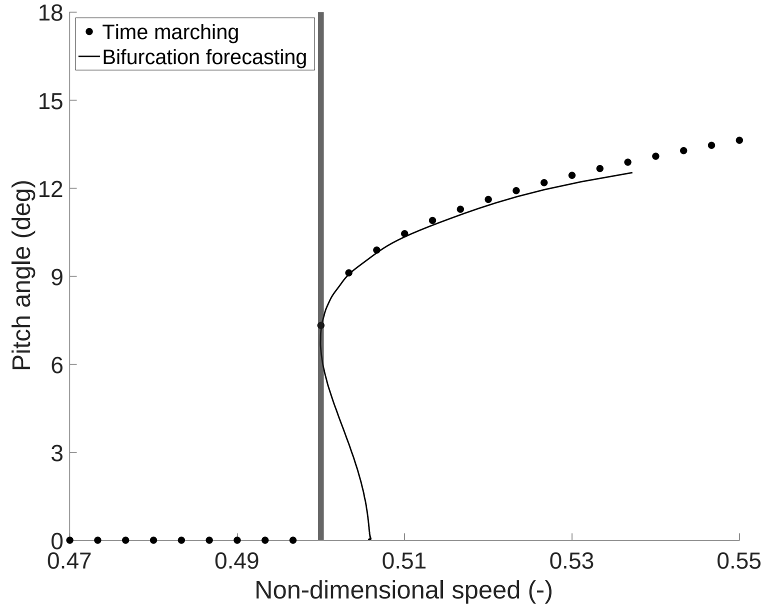


Figure 6.2: Optimized bifurcation diagram for the three-variable case.

value markedly by over 1. This represents an approximately 14.6% improvement in the value for the two-variable case. Furthermore, the number of iterations taken was slightly increased, indicating that the higher complexity of the optimization problem does not adversely impact the optimization time in a significant way (although the number of iterations does seem to scale with the problem's complexity). The results are identical for the cases where the change in elastic axis offset is and is not included in the objective function, which may originate from the low assumed value of the corresponding weight.

At this design point, both the flutter and LCO constraints are exactly satisfied. The design variables, although bounded by the established limits, could be manipulated by the optimizer to remove any LCO behavior below a non-dimensional speed of 0.5. The optimized bifurcation diagram is shown in Figure 6.2 for completeness.

REFERENCES

[1] F. Afonso, J. Vale, É. Oliveira, F. Lau, and A. Suleman, "A review on non-linear aeroelasticity of high aspect-ratio wings," *Progress in Aerospace Sciences*, vol. 89, no. Supplement C, pp. 40–57, 2017.

[2] E. Jonsson, C. Riso, C. A. Lupp, C. E. S. Cesnik, J. R. R. A. Martins, and B. I. Epureanu, "Flutter and post-flutter constraints in aircraft design optimization," *Progress in Aerospace Sciences*, vol. 109, p. 100 537, 2019.

[3] E. Jonsson, C. A. Mader, G. J. Kennedy, and J. R. R. A. Martins, "Computational modeling of flutter constraint for high-fidelity aerostructural optimization," in *AIAA SciTech Forum*, AIAA-2019-2354, San Diego, CA, Jan. 2019, pp. 1–27.

[4] R. E. Bartels and B. K. Stanford, "Aeroelastic optimization with an economical transonic flutter constraint using Navier–Stokes aerodynamics," *Journal of Aircraft*, vol. 55, no. 4, pp. 1522–1530, 2018.

[5] S. He, E. Jonsson, and J. R. R. A. Martins, "Wing aerodynamic shape optimization with time spectral limit-cycle oscillation adjoint," in *AIAA AVIATION Forum*, AIAA-2022-3357, Chicago, IL, Jun. 2022, pp. 1–23.

[6] K. E. Jacobson and B. K. Stanford, "Flutter-constrained optimization with the linearized frequency-domain approach," in *AIAA SciTech Forum*, San Diego, CA & Online, 2022, pp. 1–17.

[7] C. E. S. Cesnik, R. Palacios, and E. Y. Reichenbach, "Reexamined structural design procedures for very flexible aircraft," *Journal of Aircraft*, vol. 51, no. 5, pp. 1580–1591, 2014.

[8] N. P. Werter and R. De Breuker, "A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation," *Composite Structures*, vol. 158, pp. 369–386, 2016.

[9] A. Variyar, T. D. Economon, and J. J. Alonso, "Design and optimization of unconventional aircraft configurations with aeroelastic constraints," in *2017 AIAA SciTech Forum*, AIAA-2017-0463, Grapevine, TX, Jan. 2017, pp. 1–13.

[10] C. Xie, Y. Meng, F. Wang, and Z. Wan, "Aeroelastic optimization design for high-aspect-ratio wings with large deformation," *Shock and Vibration*, pp. 1–16, 2017.

[11] C. A. Lupp and C. E. S. Cesnik, "A gradient-based flutter constraint including geometrically nonlinear deformations," in *AIAA SciTech Forum*, San Diego, CA, Jan. 2019, pp. 1–22.

[12] E. Jonsson, C. Riso, B. Bahia Monteiro, A. C. Gray, J. R. R. A. Martins, and C. E. S. Cesnik, "High-fidelity gradient-basedwing structural optimization including a geometrically nonlinear flutter constraint," *AIAA Journal*, 2023.

[13] A. C. Gray, C. Riso, E. Jonsson, J. R. R. A. Martins, and C. E. S. Cesnik, "High-fidelity aerostructural optimization with a geometrically nonlinear flutter constraint," *AIAA Journal*, pp. 1–14, 2023.

[14] E. Dowell, J. Edwards, and T. Strganac, "Nonlinear aeroelasticity," *Journal of Aircraft*, vol. 40, no. 5, pp. 857–874, 2003.

[15] D. Tang and E. H. Dowell, "Experimental and theoretical study on aeroelastic response of high-aspect-ratio wings," *AIAA Journal*, vol. 39, pp. 1430–1441, 2001.

[16] B. Stanford and P. Beran, "Direct flutter and limit cycle computations of highly flexible eings for efficient analysis and optimization," *Journal of Fluids and Structures*, vol. 36, pp. 111–123, 2013.

[17] A. Drachinsky, O. Avin, D. E. Raveh, Y. Ben-Shmuel, and M. Tur, "Flutter tests of the pazy wing," *AIAA Journal*, pp. 1–8, 2022, article in advance.

[18] C. Riso, C. E. S. Cesnik, and B. I. Epureanu, "Bifurcation-diagram-free postflutter response constraint for design optimization," *AIAA Journal*, 2023.

[19] G. Dimitriadis, *Introduction to Nonlinear Aeroelasticity*. John Wiley and Sons, 2017.

[20] D. Tang and E. H. Dowell, "Experimental and theoretical study for nonlinear aeroelastic behavior of a flexible rotor blade," *AIAA Journal*, vol. 31, no. 6, pp. 1133–1142, 1993.

[21] M. J. Patil, D. H. Hodges, and C. E. S. Cesnik, "Limit-cycle oscillations in high-aspect-ratio wings," *Journal of Fluids and Structures*, vol. 15, no. 1, pp. 107–132, 2001.

[22] A. Nayfeh and B. Balachandran, *Applied Nonlinear Dynamics*. John Wiley and Sons, 1995.

[23] R. Kapania and S. Park, "Nonlinear transient response and its sensitivity using finite elements in time," *Computational Mechanics*, vol. 17, no. 5, pp. 306–317, 1996.

[24] P. Bar-Yoseph, F. D., and G. O., "Spectral element methods for nonlinear spatio-temporal dynamics of an euler-bernoulli beam," *Computational Mechanics*, vol. 19, no. 1, pp. 136–151, 1996.

[25] M. Kurdi and P. Beran, "Spectral element method in time for rapidly actuated systems," *Journal of Computational Physics*, vol. 227, no. 3, pp. 1809–1835, 2008.

[26] G. Sewell, *Numerical Solution of Ordinary and Partial Differential Equations*. John Wiley and Sons, 2005.

[27] A. Raghothama and S. Narayanan, "Non-linear dynamics of a two-dimensional airfoil by incremental harmonic balance method," *Journal of Sound and Vibration*, vol. 226, no. 3, pp. 493–517, 1999.

[28] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Nonlinear inviscid aerodynamic effects on transonic divergence, flutter, and limit-cycle oscillations," *AIAA Journal*, vol. 40, no. 4, pp. 638–646, 2002.

[29] K. C. Hall, J. P. Thomas, and W. S. Clark, "Computation of unsteady nonlinear flows in cascades using a harmonic balance technique," *AIAA Journal*, vol. 40, no. 5, pp. 879–886, 2002.

[30] L. Liu and E. H. Dowell, "Harmonic balance approach for an airfoil with a freeplay control surface," *AIAA Journal*, vol. 43, no. 4, pp. 802–815, 2005.

[31] K. C. Hall, K. Ekici, J. P. Thomas, and E. H. Dowell, "Harmonic balance methods applied to computational fluid dynamics problems," *International Journal of Computational Fluid Dynamics*, vol. 27, no. 2, pp. 52–67, 2013.

[32] A. Gopinath and A. Jameson, "Time spectral method for periodic unsteady computations over two- and three- dimensional bodies," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan. 2005, pp. 1–14.

[33] L. L. Smith and L. Morino, "Stability analysis of nonlinear differential autonomous systems with applications to flutter," *AIAA Journal*, vol. 14, no. 3, pp. 333–341, 1976.

[34] A. Vakakis, "Non-linear normal modes (nnms) and their applications in vibration theory: An overview," *Mechanical Systems and Signal Processing*, vol. 11, no. 1, pp. 3–22, 1997.

[35] A. H. Nayfeh, *Perturbation Methods*. New York: John Wiley and Sons, 2000.

[36] A. H. Nayfeh, *The Method of Normal Forms*. John Wiley & Sons, 2011.

[37] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*. Springer Science & Business Media, 2012, vol. 13.

[38] M. Berci and G. Dimitriadis, "A combined multiple time scales and harmonic balance approach for the transient and steady-state response of nonlinear aeroelastic systems," *Journal of Fluids and Structures*, vol. 80, pp. 132–144, 2018.

[39] J. Lim and B. I. Epureanu, "Forecasting a class of bifurcations: Theory and experiment," *Physical Review E*, vol. 83, pp. 016 203–016203-9, 1 2011.

[40] R. Lind, "Flight-test evaluation of flutter prediction methods," *Journal of Aircraft*, vol. 40, no. 5, pp. 964–970, 2003.

[41] G. Dimitriadis and J. E. Cooper, "Flutter prediction from flight flutter test data," *Journal of Aircraft*, vol. 38, no. 2, pp. 355–367, 2001.

[42] K. E. Jacobson, J. F. Kiviaho, G. J. Kennedy, and M. J. Smith, "Evaluation of time-domain damping identification methods for flutter-constrained optimization," *Journal of Fluids and Structures*, vol. 87, pp. 174–188, 2019.

[43] C. Riso, A. Ghadami, C. E. S. Cesnik, and B. I. Epureanu, "Data-driven forecasting of postflutter responses of geometrically nonlinear wings," *AIAA Journal*, vol. 58, no. 6, pp. 2726–2736, 2020.

[44] C. Riso, C. E. S. Cesnik, and B. I. Epureanu, "Output-based flutter bifurcation forecasting in nonlinear fluid-structural systems with multiple varying parameters," *Journal of Fluids and Structures*, vol. 101, p. 103 201, 2021.

[45] S. Missoum, C. Dribusch, and P. Beran, "Reliability-based design optimization of nonlinear aeroelasticity problems," *Journal of Aircraft*, vol. 47, pp. 992–998, 3 2010.

[46] B. Stanford, P. Beran, and M. Kurdi, "Adjoint sensitivities of time-periodic nonlinear structural dynamics via model reduction," *Computers and Structures*, vol. 88, no. 19–20, pp. 1110–1123, 2010.

[47] B. Stanford and P. Beran, "Computational strategies for reliability-based structural optimization of aeroelastic limit cycle oscillations," *Structural and Multidisciplinary Optimization*, vol. 45, no. 1, pp. 83–99, 2012.

[48] S. He, E. Jonsson, and J. R. R. A. Martins, "Adjoint-based limit cycle oscillation instability sensitivity and suppression," *Nonlinear Dynamics*, vol. 111, pp. 3191–3205, 2023.

[49] J. P. Thomas and E. H. Dowell, "Discrete adjoint approach for nonlinear unsteady aeroelastic design optimization," *AIAA Journal*, vol. 57, no. 10, pp. 4368–4376, 2019.

[50] S. He, "Aerodynamic shape optimization using a time-spectral approach for limit cycle oscillation prediction," University of Michigan, Ph.D. Thesis, 2021.

[51] G. Kreisselmeier and R. Steinhauser, "Systematic control design by optimizing a vector performance index," in *Computer Aided Design of Control Systems*, Pergamon, 1980, pp. 113–117.

[52] G. A. Wrenn, "An indirect method for numerical optimization using the kreisselmeier–steinhauser function," NASA Langley Research Center, Hampton, VA, Tech. Rep. CR-4220, 1989.

[53] N. M. K. Poon and J. R. R. A. Martins, "An adaptive approach to constraint aggregation using adjoint sensitivity analysis," *Structural and Multidisciplinary Optimization*, vol. 34, pp. 61–73, 2007.

[54] C. Riso, C. E. S. Cesnik, and B. I. Epureanu, "Estimating flutter bifurcation diagrams using the state velocity," *AIAA Journal*, vol. 60, no. 9, pp. 1–13, 2022.

[55] J. F. Kiviaho, K. E. Jacobson, and G. J. Kennedy, "Flutter boundary identification from time-domain simulations using the matrix pencil method," *AIAA Journal*, vol. 57, no. 8, pp. 3639–3645, 2019.

[56] D. A. Peters, M. C. A. Hsieh, and A. Torrero, "A state-space airloads theory for flexible airfoils," *Journal of the American Helicopter Society*, vol. 52, no. 4, pp. 329–342, 2007.

[57] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, 4 2019.

[58] R. Falck, J. S. Gray, K. Ponnapalli, and T. Wright, "Dymos: A python package for optimal control of multidisciplinary systems," *Journal of Open Source Software*, vol. 6, no. 59, p. 2809, 2021.