POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Analysis, monitoring and control of morphing wing structures based on the inverse Finite Element Method

Relatore
prof. Marco Gherlone

Correlatore:
dott. Rinto Roy

Studente
Vincenzo Biscotti
matricola: 301898

Anno accademico 2022-2023

Abstract

Morphing capabilities allow for the design of a wing structure able to drastically change its external shape. The adaptive and flexible morphing wing is thus able to achieve high efficiency throughout the entire flight envelope reducing, for instance, consumption and pollution and improving aerodynamic performance.

In order to achieve its potential, a morphing wing necessitates the development of an effective controller and, in particular, of an accurate feedback system able to provide a real-time reconstruction of the multitude of possible shapes achievable by the morphing wing. In developing a feedback system, successful results have been collected by recreating the displacement field of the wing structure from the actual measurements of the strains of the structure itself. This approach is an inverse problem called shape sensing.

Notably, the shape sensing methodology known as iFEM (inverse Finite Element Method) has already shown promising applications in both open and closed-loop control strategies for morphing wing structures. However, closed-loop control architectures have not been tested thoroughly yet. Therefore, this master thesis focuses on the development of a closed-loop control system for the actuation of a morphing wing structure using the iFEM reconstructed shape as feedback. The objective is to observe the behavior of the iFEM results in a closed-loop environment as well as to evaluate the effectiveness of the iFEM code in providing feedback for the control scheme.

To simulate the actuation of the structure and the strain measurements, a FE model of the structure is created. The closed-loop architecture is implemented in a MATLAB environment. At each iteration, the MATLAB code calculates the difference between the target and iFEM reconstructed displacements. Following a least-squares approach this difference is used to compute the actuation loads. The least-squares problem is ill-posed thus, as the complexity of the structure (and, as a consequence, of its model) increases, regularization is needed. Therefore, the Tikhonov regularization method is introduced.

The results are presented for both a simple plate and a more realistic trailing edge model. In both structures, the morphing potential is exploited by modifying the camber of the structure so that it follows an arbitrary shape. The response of the control scheme and of the structure to different operational conditions and external disturbances is tested. For the trailing edge model, the external disturbance is simulated by the pressure distribution over the morphed structure, which is extracted at each iteration using the software Xfoil.

Contents

List of Figures 3

1 Introduction 5

1.1 Brief history of morphing wing designs 6

1.2 Morphing wing structures: state of the art 8

1.3 Control strategies for morphing structures: state of the art 15

1.3.1 PID control 15

1.3.2 Reinforcement Learning control 18

2 The inverse Finite Element Method 21

3 Control loop architecture 25

3.1 Notation introduction 25

3.2 Control loop flow chart 26

3.3 Target shape 28

3.4 Actuation loads 29

3.4.1 Least-squares approach 29

3.4.2 Regularization of the least-square method 32

3.5 FEM strains and displacements 34

3.6 Aerodynamic loads 35

3.7 Error definition 37

4 Results: plate structure 41

4.1 Plate structure model 41

4.2 Results 43

5 Results: trailing edge structure 49

5.1 Trailing edge model 49

5.1.1 3D Model 49

5.1.2 FE Model 51

5.2 Results 53

6 Conclusions 79

References 83

List of Figures

1.1 Timeline of fixed-wing aircraft implementing morphing technologies [1] . . 7
1.2 Spider plot comparing predicted performance of the baseline Firebee, a morphing airfoil Firebee, and a morphing planform Firebee [1] 8
1.3 The MIT Digital Morphing Wing [8] 9
1.4 Construction of shape-changing structures from discrete lattice building-block elements [9] 10
1.5 3D printing manufactured composite morphing wing [10] 11
1.6 Bio-inspired MAV [12] 12
1.7 SmartX morphing wing design [13] 12
1.8 Conceptual design of a UAV morphing wing [15] 13
1.9 Alternative conceptual morphing design of a UAV wing [16] 14
1.10 Clean Sky's prototype of a morphing wing for a civil aircraft [17] 14
1.11 Proportional-Integral-Derivative controller [18] 16
1.12 PID controller implemented in the closed loop control architecture of an adaptable morphing trailing edge concept [18] 16
1.13 Closed loop control of the shape morphing airfoil using a fuzzy logic PID hybrid controller [20] 17
1.14 Hybrid fuzzy logic proportional-integral-derivative and conventional on-off controller architecture [20] 17
1.15 Deep Reinforcement Learning control scheme [23] 19
2.1 Notation for the flat shell [25] 22
3.1 Closed-loop control architecture 27
3.2 Closed-loop control architecture - Added external disturbance 28
3.3 Target shape of the tested morphing structures 29
3.4 Reference cross-section reconstruction throughout the iterations 36
3.5 Refined aerodynamic paneling procedure 36
3.6 XFOIL reconstructed pressure distribution throughout iterations 37
3.7 Error plot on the morphed structure throughout the iterations 39
3.8 Zoom of the top picture of figure 3.7b 39
3.9 New target shape definition 40
4.1 Plate FE model 42
4.2 External disturbance model on the plate structure 43
4.3 Test on the correct implementation of the influence matrix 45
4.4 FEM and iFEM error along the iterations - $m = 0$ 45
4.5 FEM and iFEM error along the iterations - $m = 25$ 46

4.6 iFEM reconstructed shape throughout the closed-loop iterations - $m = 25$, $\lambda = 0$ . . . 47
4.7 iFEM reconstructed shape throughout the closed-loop iterations - $m = 25$, $\lambda = 10^{-6}$ . . . 47
4.8 iFEM reconstructed shape throughout the closed-loop iterations - $m = 50$, $\lambda = 10^{-6}$ . . . 47
4.9 FEM and iFEM error along the iterations - $m = 50$, $\lambda = 10^{-6}$ . . . 48
5.1 Trailing edge 3D model . . . 50
5.2 Trailing edge FE model . . . 51
5.3 Pressure distribution on the trailing edge FE model . . . 52
5.4 Target shapes . . . 53
5.5 Test of the correct implementation of the influence matrices relative to the two actuation schemes . . . 54
5.6 FEM and iFEM error plot along the iterations - $k = 0.4$, $V = 0$ m/s, sparse transverse forces actuation scheme . . . 54
5.7 Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, $V = 0$ m/s, sparse transverse forces actuation scheme . . . 55
5.8 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0$ m/s, sparse transverse forces actuation scheme . 60
5.9 Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, $V = 0$ m/s, distributed pressure actuation scheme . . . 60
5.10 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0$ m/s, $\lambda = 0$ distributed pressure actuation scheme 62
5.11 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0$ m/s, $\lambda = 10^{-16}$ distributed pressure actuation scheme . . . 64
5.12 Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 1$, $V = 0$ m/s . . . 64
5.13 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 1$, $V = 0$ m/s, discrete transverse forces actuation scheme . 66
5.14 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 1$, $V = 0$ m/s, distributed pressure actuation scheme . . . 69
5.15 Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, sparse transverse forces actuation scheme . . . 70
5.16 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 100$ m/s, sparse transverse forces actuation scheme 72
5.17 Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 100$ m/s, sparse transverse forces actuation scheme . . . 72
5.18 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 200$ m/s, sparse transverse forces actuation scheme 74
5.19 Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 200$ m/s, sparse transverse forces actuation scheme . . . 74
5.20 Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 300$ m/s, sparse transverse forces actuation scheme 77
5.21 Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 300$ m/s, sparse transverse forces actuation scheme . . . 77

Chapter 1

Introduction

The term 'morphing' is the short of metamorphose [1]. The etymology of the term is the Greek word 'metamórphōsis', which means transformation. In the dictionary, the possible definitions of 'metamorphosis' are the following: "A profound change in form from one stage to the next in the life history of an organism" or "A complete change of form, structure, or substance, as transformation by magic or witchcraft"[2]. These two definitions are related, as it is likely that the observation of the biological metamorphosis of plants and animals led to the creation of ancient myths and legends where human characters metamorphose, changing their appearance whilst keeping their identity [3].

In the aeronautical field, the word 'morphing' is usually considered to mean large shape change or transfiguration [4]. The term 'large' is intentionally vague as there is neither an exact definition nor an agreement between the researchers about the type or the extent of the geometrical changes necessary to qualify an aircraft for the title 'shape morphing' [1, 5]. The lack of a complete definition is in part related to the multiple efforts of designers and engineers who, since the very beginning of aeronautics, have tried to exploit the potential of a shape-morphing aircraft. Therefore, rather than attempting to produce a concise but incomplete definition, this chapter begins with a brief history of morphing designs to portray the main features and benefits of morphing when applied to air-vehicles.

The continuous efforts that, starting from the first aviation pioneers, were put in the development of morphing concepts and the increasing interest in the application of morphing in the aeronautical field laid the basis for the modern understanding and the modern designs of morphing structures. In today's literature, a multitude of papers regarding morphing aircraft and, notably, morphing wing designs can be found. A key concept underlying many of these papers is the development of enabling technologies for such structures: for instance, an intensively researched topic is the control strategy of the morphing wing.

In developing an effective controller, an accurate feedback system able to provide a real-time reconstruction of the shape of the structure is needed. The state-of-the-art feedback systems often revolve around sensor data measurements rather than relying on the actual shape of the structure. This observation was first made by the co-supervisor of this master thesis, Dr. Rinto Roy, in his paper [6], where he demonstrates how the shape of the structure can be accurately reproduced via the shape sensing methodology named 'inverse Finite Element Method' (iFEM). Exploiting this potential, in [6] a real-time shape-based feedback control strategy for morphing wing structures is developed. In his work [6], Dr. Roy describes both open and closed-loop control strategies and explores in depth the results of the open-loop control architecture. On the other hand, numerical results of the closed-loop control strategy are left as future work.

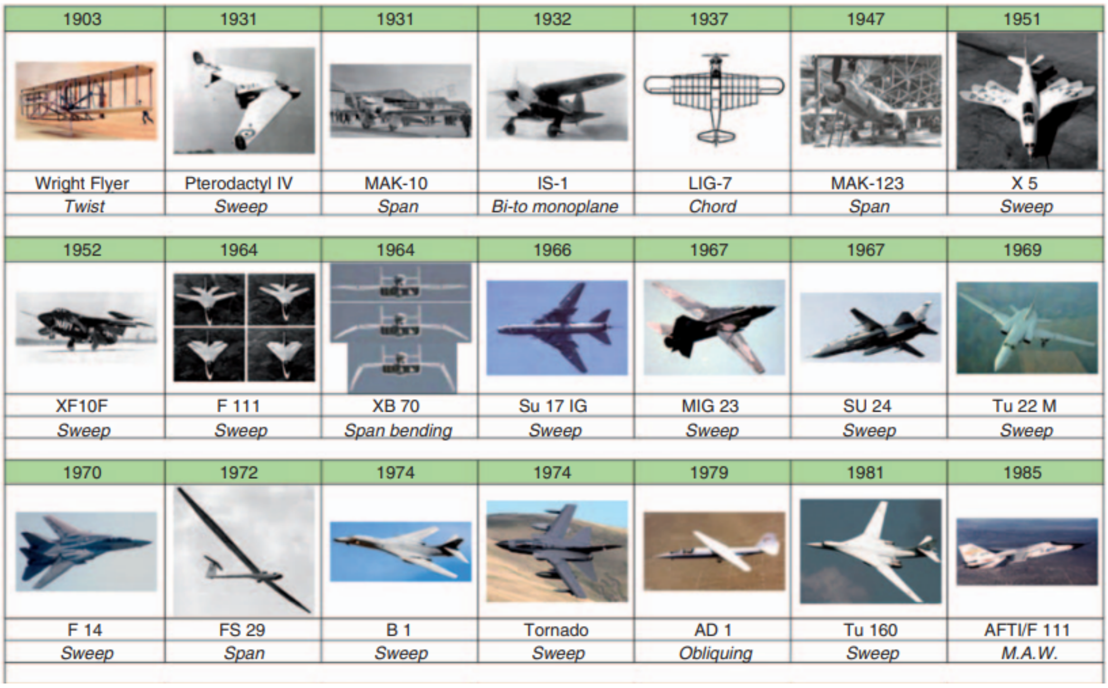
The paper of Dr. Roy paved the road for the efforts behind this master thesis. The objective is to complete the results already provided in [6] by implementing the inverse Finite Element Model shape feedback in a closed-loop control environment. Before discussing the actual control loop strategy, however, it is useful to build a general yet comprehensive framework of the modern morphing designs and their controllers. Therefore, following the introductory paragraph on the history of morphing in the aeronautical field, the state-of-the-art of both morphing wing structures and their closed-loop control strategies is provided.

1.1 Brief history of morphing wing designs

Since the very beginning of the history of the aeronautical field, morphing capabilities have always fascinated engineers and designers. The early pioneers found out that birds used warping of their wings to control flight. Wilbur Wright in 1900 wrote “My observation of the flight of buzzards leads me to believe that they regain their lateral balance, when partly overturned by a gust of wind, by a torsion of the tips of the wings. If the rear edge of the right wing tip is twisted upward and the left downward, the bird becomes an animated windmill and instantly begins to turn, a line from its head to its tail being the axis” (Wright, 1900) [7]. Following this observation, the Wright brothers created a glider that implemented a system of wires that could be used by the pilot to control lateral stability. This innovative morphing design represented a fundamental step in the creation of history’s first successful airplane, the Wright Glided. Bio-inspired designs were later exploited by the Austrian engineer Igo Etrich, with his Etrich Taube (’Taube’ means dove in German). The Taube was essentially a bio-mimetic airplane: the planform of the Taube accurately reproduced the shape of the Zanonia macrocarpa seed, which falls from trees in a similar spinning motion of the Disamare, commonly seen in Italy in mountain regions. Moreover, as for the Wright glider, the Taube exploited the warping of both the wing and the horizontal tail for lateral control.

As planes became faster and faster, metallic airplanes proved to be much more successful in sustaining the increasing structure loads. As a consequence, the morphing bio-inspired solutions were replaced by the now commonly used systems of hinges, pivots, and rails. However, engineers still experimented with morphing designs, now acting on the span of the wing to reduce wing loading (which is the ratio between the weight and the span of the wing) and, in turn, reduce induced drag. The first telescopic wing design, able to change its span, was conceived in France, by Ivan Makhonine, and applied to his aircraft, the Makhonine MAK-101 (1931). The idea of changing the geometry of the wing to adapt to different flight envelope conditions was pursued by NASA with the design in 1951 of the Bell X-5, which could change its wing sweep. The wing sweep could be set to 20, 45, and 60 degrees: high sweep angle values improved the take-off and landing performance. On the other hand, a low sweep angle allowed to achieve high speed and aerodynamic efficiency. The Bell X-5 inspired a series of military designs that went into large-scale production in the 1960s and 1970s.

The numerous studies on variable geometry aircraft reached a peek in 1981, with the development of NASA's AFTI F-111 Mission Adaptive Wing (MAW), an aircraft that coupled the mature variable geometry technologies with a shape morphing capability, similar to the one of the first aircraft designs. The objective was to create an aircraft that could adapt to off-flight conditions not only by changing its wing sweep angle but also by acting on the camber of its profile while maintaining a smooth skin surface.



1903	1931	1931	1932	1937	1947	1951
Wright Flyer	Pterodactyl IV	MAK-10	IS-1	LIG-7	MAK-123	X 5
Twist	*Sweep*	*Span*	*Bi-to monoplane*	*Chord*	*Span*	*Sweep*
1952	1964	1964	1966	1967	1967	1969
XF10F	F 111	XB 70	Su 17 IG	MIG 23	SU 24	Tu 22 M
Sweep	*Sweep*	*Span bending*	*Sweep*	*Sweep*	*Sweep*	*Sweep*
1970	1972	1974	1974	1979	1981	1985
F 14	FS 29	B 1	Tornado	AD 1	Tu 160	AFTI/F 111
Sweep	*Span*	*Sweep*	*Sweep*	*Obliquing*	*Sweep*	*M.A.W.*

Figure 1.1: Timeline of fixed-wing aircraft implementing morphing technologies [1]

Each of the designs briefly presented in this chapter (figure 1.1) represented a small step toward the modern concept of morphing wing designs. All these separate efforts were collected in the early years of the 21st century by two different projects: the NASA Morphing Aircraft Project and the DARPA Morphing Aircraft Structures (MAS) Program. Both programs contributed to the definition of 'morphing' and laid the basis for the state-of-the-art of morphing designs. The former project defined morphing as "efficient, multipoint adaptivity" [4]. The latter, on the other hand, stated: "Morphing is a capability to provide superior and/or new vehicle system performance while in flight by tailoring the vehicle's state to adapt to the external operational environment and multi-variable mission roles. In the context of this DTO, morphing aircraft are multi-role aircraft that change their external shape substantially to adapt to a changing mission environment during flight"[7]. As for the morphing designs developed within those projects, the past morphing solutions such as bio-inspiration, warping, shape-changing, and variable geometry were combined together leading to innovative designs with the potential to accommodate contradictory mission requirements; achieve resilience against unforeseen problems; reduce weight and drag and face mission scenarios previously deemed impossible [4].

The results of the NASA and DARPA projects kindled a renewed interest in morphing wing structures. Morphing wing designs were proven to greatly improve aerodynamic performance for a range of flight conditions, thus improving fuel savings and in turn reducing costs and pollution. An example of the performance enhancement produced with morphing is shown in the spider plot of figure 1.2. The spider plot compares, for different flight conditions, the baseline performance (in red) of the UAV BQM-34 to the performances of two different morphing strategies: a geometrical variation (in green), or an airfoil shape variation (in yellow). The outer circle represents the maximum performance value. As clearly shown in the figure, the performance of the aircraft is improved throughout all the flight envelope, highlighting the potential of morphing wing designs.

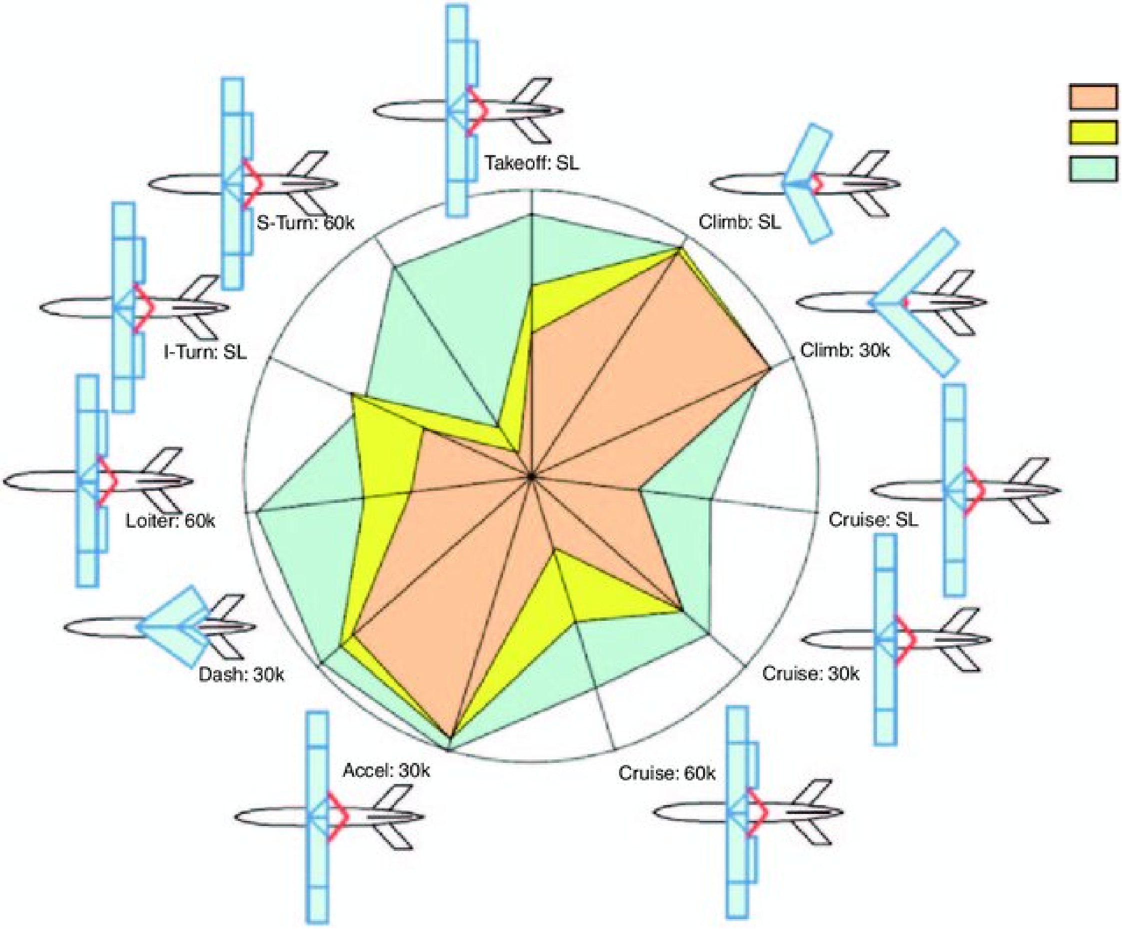


Figure 1.2: Spider plot comparing predicted performance of the baseline Firebee, a morphing airfoil Firebee, and a morphing planform Firebee [1]

1.2 Morphing wing structures: state of the art

The potential of morphing wing structures has attracted the attention of many researchers in the past decade. In today's literature, a plethora of morphing wing designs can be found.

This chapter contains a small collection of those designs together with a brief description of each one of them.

The first morphing wing structure presented is shown in figure 1.3. This morphing wing design was developed by the team of Neil Gershenfeld, director of MIT's Center for Bits and Atoms (CBA) in collaboration with NASA [8]. The internal structure of the wing is made of lattice carbon fiber reinforced polymer (CFRP) assembly forming a cellular solid structure. Cellular solids are materials made of a network of 1D and 2D elements. Such materials combine high specific properties with an extremely lightweight. Since the properties of cellular solid structures are related to the geometry of the lattice array, the macroscopic properties of the structure itself can be tuned to match the desired compliance and mass [9]. The skin of the wing is made of strips of Kapton. The flexible strips can overlap like the feathers of a bird, thus maintaining a smooth skin surface through morphing. Actuation is obtained via a small set of actuators placed at the wing tip. The motors allow the wing to twist for roll control.

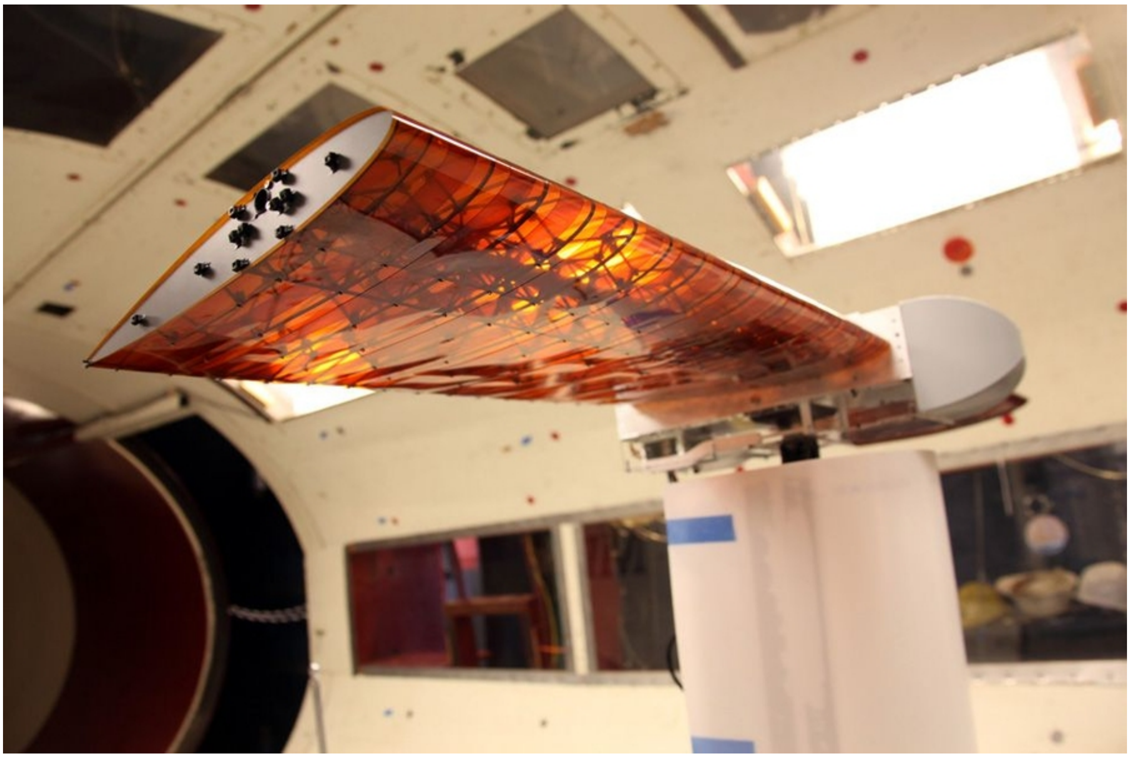


Figure 1.3: The MIT Digital Morphing Wing [8]

The peculiarity of the MIT wing design is shown in figure 1.4: the lattice structure is made of a repetitive unit, which Gershenfeld calls 'digital material' [8]. As shown in the top-right picture, the digital morphing structure can be assembled by combining the building block elements. This design strategy drastically reduces the manufacturing process complexity and allows to precisely tune the properties of the cellular solid.

The digital approach is not the only innovative manufacturing process found in the literature. At the ETH spin-off 9T Labs, a 3D-printed composite morphing wing has been

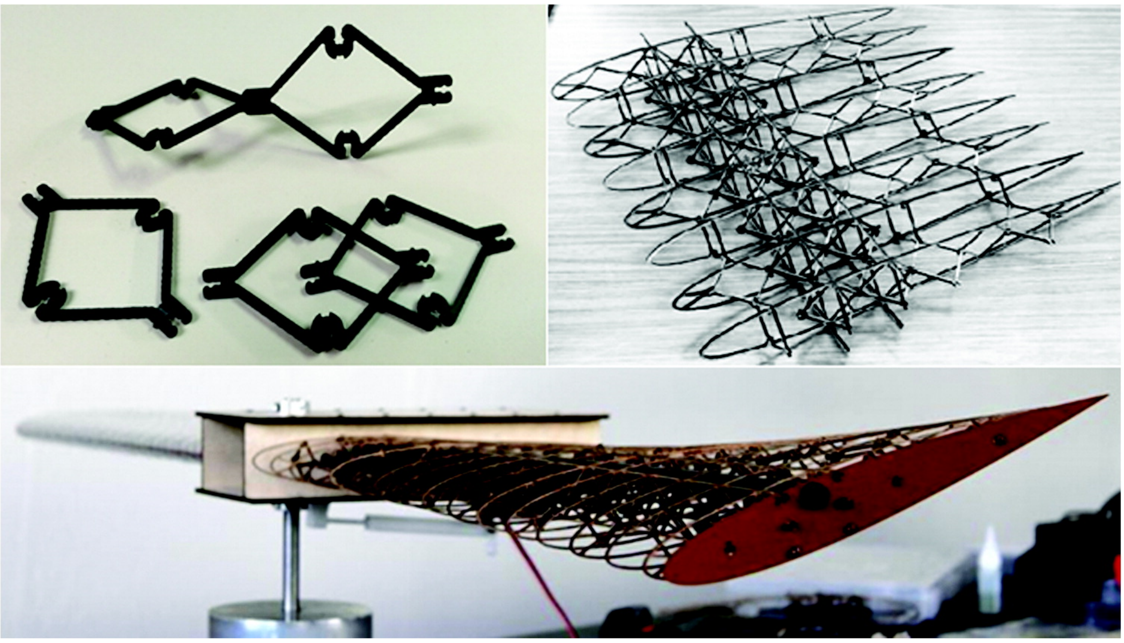


Figure 1.4: Construction of shape-changing structures from discrete lattice building-block elements [9]

developed [10]. To test the performance of the wing, the research team has produced a morphing composite drone, shown in figure 1.5. In the picture, the morphing parts of the drone are colored in green: the internal lattice structure of both the V-tail and the wing is actuated using 8 internal servo motors that allow trailing edge bending. As the actuators are independent from one another, the lift along the span can be varied, reducing the structural loads and potentially increasing the efficiency of the drone [11]. The carbon fiber and plastic internal structure, as mentioned before, is obtained via additive manufacturing. This innovative process, as waste material is minimized and the complexity of the printed part doesn't add to the cost of the process, reduces the cost of the process itself. Furthermore, the 3D printing technique allows to exploit the anisotropy of the carbon fiber reinforced polymer, by aligning the fibers in the direction of the load path [11].

In the field of drones, in particular of 'micro air-vehicles' (MAVs), an extremely interesting design is the one shown in figure 1.6. The primary flight feathers, highlighted in red in figure (a), allow birds to modify the planform area of the wing. The bio-inspired wing design presented mimics this mechanism: 8 artificial feathers are connected to the outer section of the leading edge and can be folded or deployed using respectively a pre-stretched linear spring or a servo motor [12]. This wing mechanism is used to achieve in a single efficient design conflicting mission requirements: when the drone has to maneuver among obstacles small turn radius is needed. This can be achieved by deploying the feather-like wings since the turn radius is directly proportional to the wing loading (W/S). On the other hand, when facing wind gust, or even breeze, by folding the wing, high wind resistance is achieved.

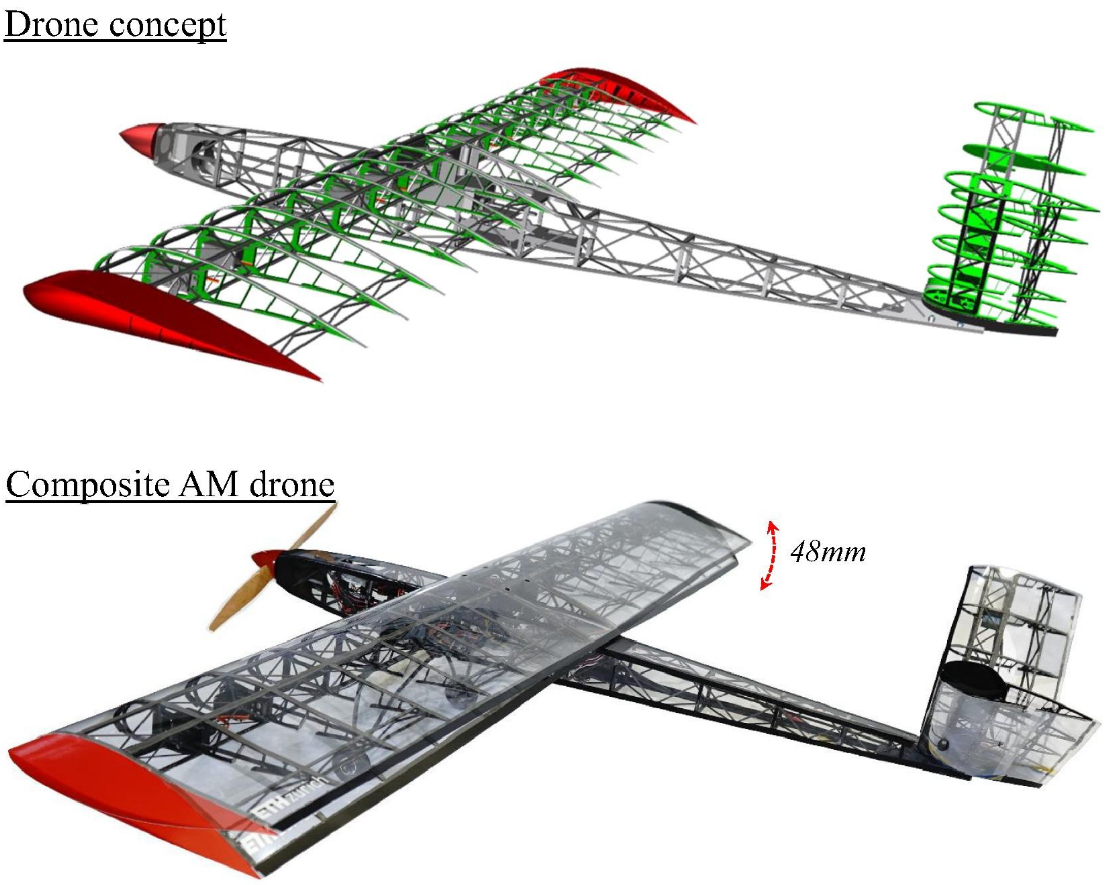


Figure 1.5: 3D printing manufactured composite morphing wing [10]

Moving away from morphing drones, many promising UAV morphing designs can be found in literature: here in this chapter two different projects are presented: the SmartX prototype morphing wing [13, 14], and the morphing concepts developed by G. Molinari and his team [15, 16]. The SmartX project is presented in figure 1.7. The morphing action takes place in the trailing edge of the structure, which is divided into 6 different morphing modules. Each module is actuated using a pair of electromechanical actuators placed, as shown for the first module in figure 1.7, in the wingbox. To guarantee continuity in the span direction, the gaps in the composite skin are filled with a flexible elastomeric skin. This actuation system is able to provide high deflections at a slow pace. To achieve faster morphing, the electromechanical actuators are coupled with piezoelectric bimorphs installed along the tip of each trailing edge module. To reconstruct the shape of the morphing structure, the trailing edge skin is bonded with FBG (Fiber Bragg Grating) sensors. The optical fibers run along both the span direction and the chord direction, assuming in this last configuration a U shape to measure both torsion a bending. The strain measurement is converted into displacements that allow for the reconstruction of

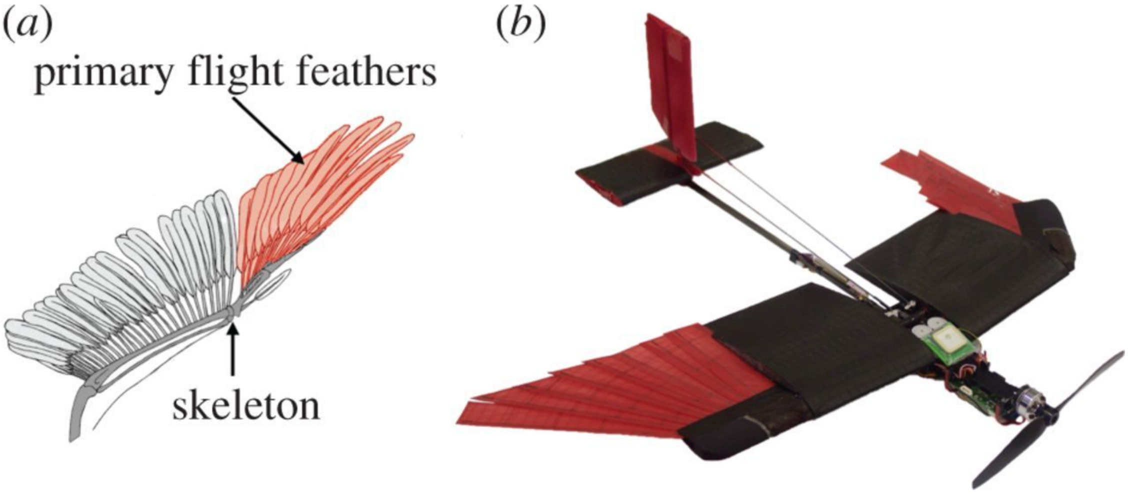


Figure 1.6: Bio-inspired MAV [12]

the shape using proper transfer functions that are obtained from pre-known morphing configurations [14]. Figure 1.7 shows how the skin is also embedded with piezoelectric sensors to measure boundary layer states.

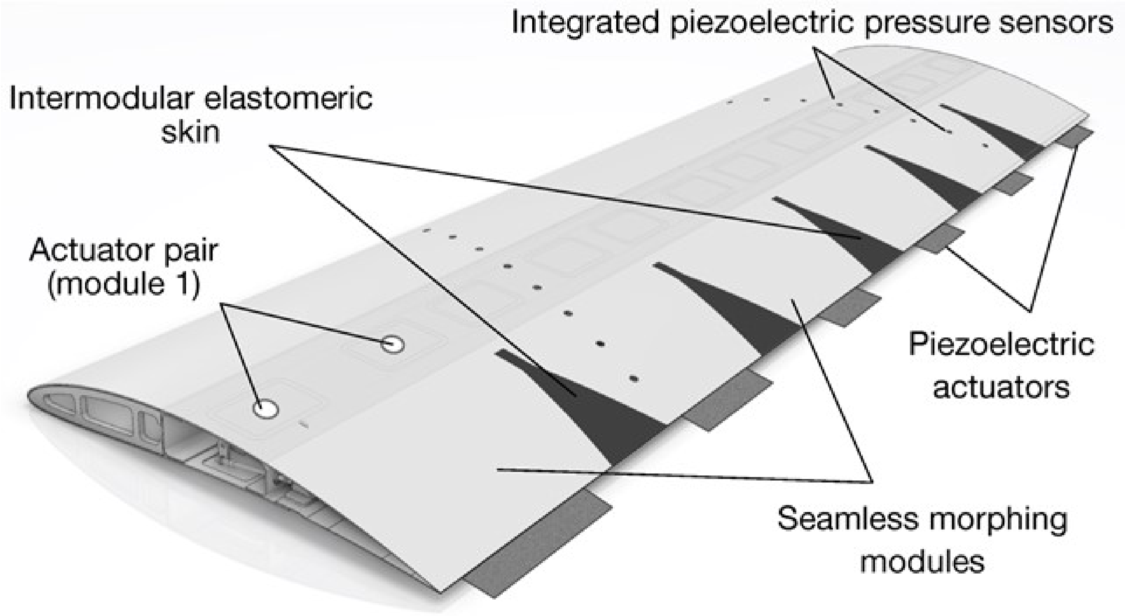

Integrated piezoelectric pressure sensors
Intermodular elastomeric skin
Actuator pair (module 1)
Piezoelectric actuators
Seamless morphing modules

Figure 1.7: SmartX morphing wing design [13]

Molinari's conceptual morphing design [15] is shown in figure 1.8. The wing is divided into three different sections. The leading edge of the structure is made of a stiff wing box and is considered infinitely rigid compared to the rest of the structure. Morphing occurs in the remaining part of the wing, which is divided into two sections: the intermediate one

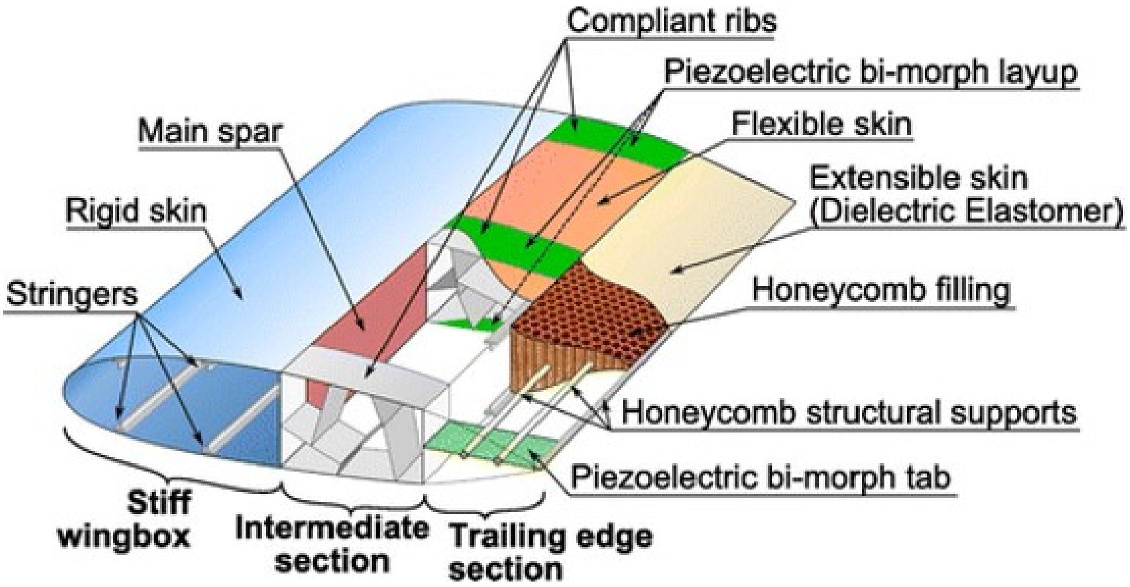


Figure 1.8: Conceptual design of a UAV morphing wing [15]

is made of compliant ribs with a truss-like structure and it is covered by a carbon fiber reinforced polymer (CFRP) skin. The actuation is obtained using piezoelectric bimorphs; the second section coincides with the trailing edge of the wing. The skin of this part is made of a dielectric elastomer (DE), which covers an inner honeycomb structure. The honeycomb core function is to withstand the pressure distribution acting on the wing. The actuation of the morphing trailing edge is obtained by combining the action of the DE skin with piezoelectric actuators placed on a cantilever plate. The actuation plate connects the intermediate section with stringers in the spanwise direction that support the honeycomb core. An alternative morphing design, proposed by G. Molinari and his team [16], is presented in figure 1.9. In this design, the compliant rib structure of the intermediate section is extended to the trailing edge. To guarantee an equivalent compliance of the rearmost part, the lower skin of the trailing edge is made of a corrugated glass fiber reinforced polymer (GFRP). To measure and reconstruct the shape of the wing as it morphs, strain gauges are bonded to the skin.

Morphing is not solely applied to unmanned air-vehicle design. Many researchers are studying how to apply morphing to civil aircraft. For instance, in figure 1.10 the morphing flap developed for a regional airplane in the Clean Sky Joint Technology Initiative (JTI) is shown. The single-slotted camber morphing flap is being designed to replace the double-slotted flap conventional architecture. Morphing is obtained through a rigid link-based mechanism: the ribs of the trailing edge are divided into four different blocks, connected in a "finger-like" configuration [17]. To reconstruct the shape, the relative rotations between the parts of the mechanism are measured using encoders.

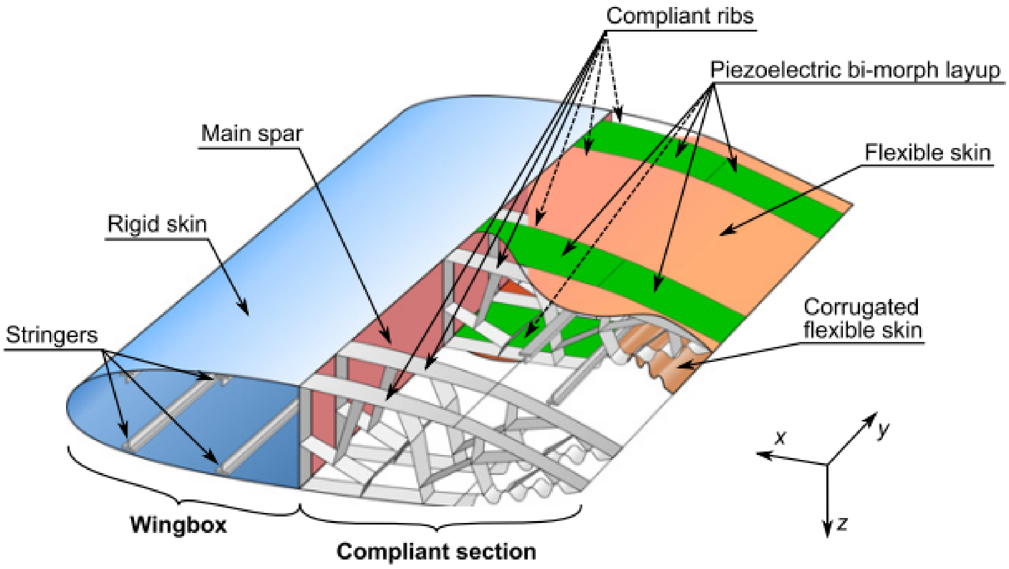
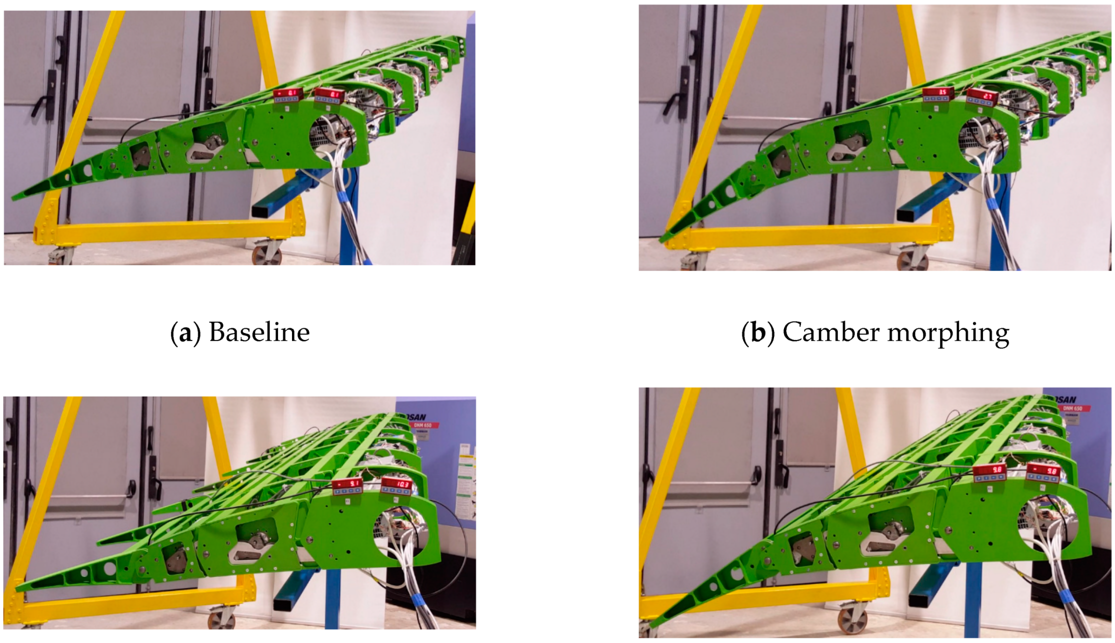


Figure 1.9: Alternative conceptual morphing design of a UAV wing [16]



(**a**) Baseline

(**b**) Camber morphing

(**c**) Tab up

(**d**) Tab down

Figure 1.10: Clean Sky's prototype of a morphing wing for a civil aircraft [17]

1.3 Control strategies for morphing structures: state of the art

In literature, two of the most frequently encountered control strategies for morphing wing designs are the following:

- Proportional-Integral-Derivative (PID) control [18–20];
- Reinforcement Learning (RL) control [21–23];

In this chapter, both are briefly presented together with examples found in the literature.

1.3.1 PID control

PID refers to a control strategy that combines the different control strategies (Proportional, Integral, and Derivative control) to obtain a unique and efficient controller.

In feedback control, the objective is to reduce the error signal between the output of the controlled system and its target value, the set point. To achieve this, the controller output (which is fed to the controlled system) can be computed using different strategies: in a proportional control scheme, the controller output is proportional to the error signal,

$$p(t) = K_c e(t) \tag{1.1}$$

Equation 1.1 implies that proportional controllers have an asymptotic behaviour, as the error $e(t)$, and as a consequence, the controller output $p(t)$, gradually decreases with each iteration. Proportional controllers are able to achieve fast convergence if the controller gain, K_c is properly adjusted, a practice which in control theory is referred to as 'tuning'. However, if an external sustained disturbance is acting on the system, or if a set change occurs, the proportional controller alone is not able to reduce the error to zero, and a steady-state error, or offset between the system's output and its target value is produced. To eliminate the steady-state error, integral control is introduced,

$$p(t) = K_I \int_0^t e(t^*)\, dt^* \tag{1.2}$$

where K_I is the integral gain. The integral controller output, roughly speaking, is produced by looking at the past, as the output is now proportional to the error integral (or sum in discrete cases) along the iterations. Therefore, in case of steady-state errors, while the proportional output reaches an asymptotic limit, the controller output of the integral controller will gradually increase with time until the error signal is zero. Implementing proportional and integral control yields effective controllers that, nonetheless, could have problems with oscillations and overshoots caused by sharp increases in the controller output. To avoid such problems, a prediction of the evolution of the error would be needed. This problem is solved with the implementation of a derivative controller,

$$p(t) = K_D \frac{de(t)}{dt} \tag{1.3}$$

where K_D is the derivative gain. Combining the three control strategies, a PID controller, as shown in figure 1.11, can be obtained.

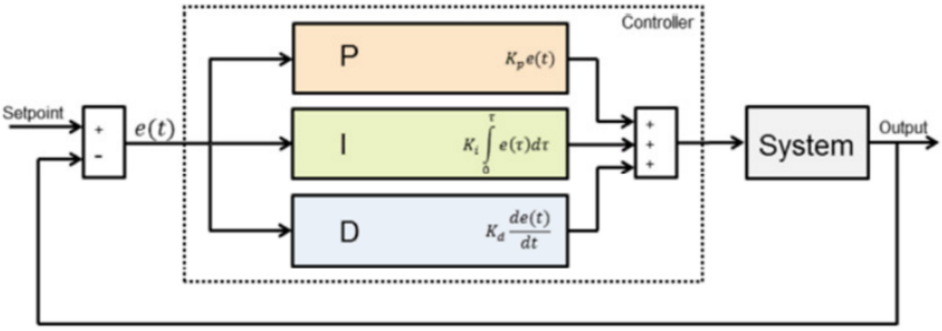


Figure 1.11: Proportional-Integral-Derivative controller [18]

The PID controller shown in figure 1.11 was implemented in a closed loop control architecture. The control scheme is presented in a Simulink environment in figure 1.12. The control scheme is used to monitor and actuate a morphing trailing edge structure made with an internal structure of segmented adaptive ribs [18]. The finger-like rib structure is actuated through rotary electromechanical actuators. The PID controller, highlighted in green in figure 1.12, is used to produce the input voltage command for the servo-actuator. The error at each iteration is obtained by the difference between the target deflection, highlighted in blue, and the angular position of the morphed trailing edge,

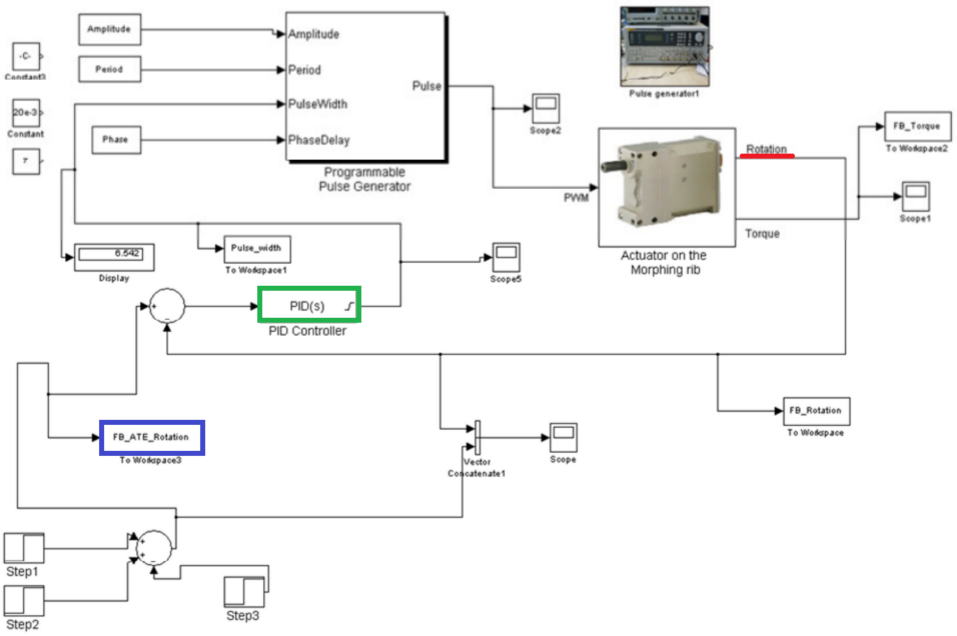


Figure 1.12: PID controller implemented in the closed loop control architecture of an adaptable morphing trailing edge concept [18]

which is measured using rotary encoders and is highlighted in red in figure 1.12.

A more complex application of PID control in closed loop control strategy is shown in figure 1.13. The closed-loop control architecture objective is to control the shape of a morphing profile in order to alter the position of the transition point between laminar and turbulent flow over the wing: moving the transition point towards the trailing edge reduces drag thus improving aerodynamic efficiency [20]. The structure is actuated with a spanwise distribution of shape memory alloy actuators. To monitor the displacements of the flexible morphing skin, two position transducers measure the displacements dY_{1real} and dY_{2real} in two specific control points. The measured displacements are then used as feedback for the controller, comparing them with the target displacements. In the previous example, a simple PID controller was sufficient for the electromechanical actuators as they were assumed to have linear behaviour. On the contrary, SMA actuators are highly non-linear and characterised by hysteresis. To deal with non-linearity, a fuzzy logic (FL) PID controller was implemented. The FL PID controller was then coupled with a conventional on-off controller to address the hysteretic behaviour between the cooling and heating phases of the SMA actuator. The resulting hybrid architecture is presented in figure 1.14.

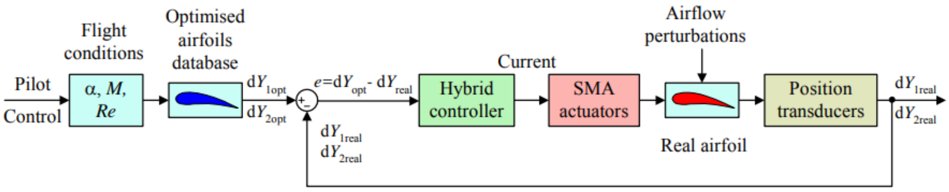


Figure 1.13: Closed loop control of the shape morphing airfoil using a fuzzy logic PID hybrid controller [20]

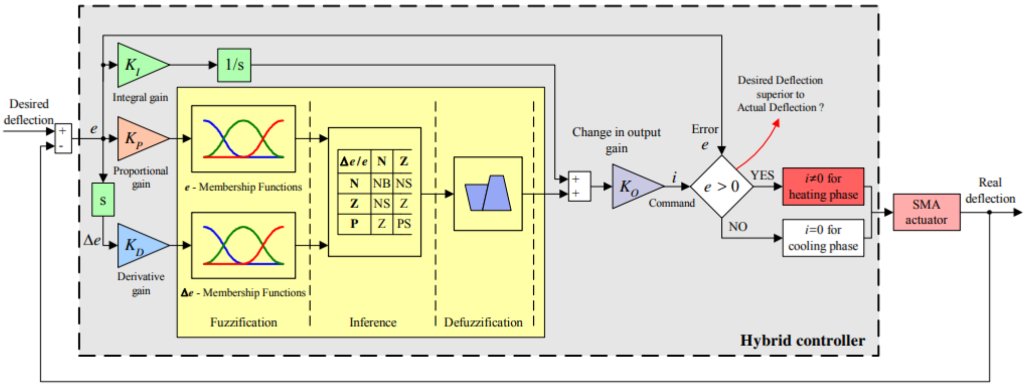


Figure 1.14: Hybrid fuzzy logic proportional-integral-derivative and conventional on-off controller architecture [20]

1.3.2 Reinforcement Learning control

Reinforcement Learning (RL) is an algorithm that is able to produce a control architecture exploiting a bio-inspired mechanism: the control scheme learns how to "behave" through trial and error interactions with the environment, in the same way as we, as humans, collect information and knowledge by interacting with our surroundings. As RL is a computational method, the reinforcement learning algorithm is trained using a goal-directed approach, that is using a reward system to achieve a certain goal. The RL problem is characterised by the interaction of a learning agent with its environment. The learning agent is defined by three main aspects [24]:

- sensation, as it must be able to sense the state of the environment;
- action, as it must be able to affect the state of the environment with its behaviour;
- goal, as it must relate goals to the state of the environment.

Sensation and action in a RL algorithm are devolved to a function called policy, that maps and determines the state-to-action behaviour of the actor. Goal is instead defined by a reward signal. The interaction of the agent with the environment produces a reward and the objective of the agent is to maximize such reward in the long run. Reward represents an immediate response to the action of the agent, thus it is the driving changing factor of the policy. However, choosing to guide the actor only with immediate feedback would lead to poor results. An initial state with low reward could result in a series of subsequent high reward states, amounting to an initially unexpected high total reward. Therefore, a key aspect in a RL method is the estimation of the value function [24], which, given a certain state, yields the expected reward to be received starting from that state. Lastly, a model of the environment, when the real environmental behaviour is unknown, is usually needed. This model, together with the policy, the reward signal, and the value function, constitutes the basis of the RL method [24].

In literature, different morphing controllers trained with RL can be found and frequently the control scheme is tested in a simulated environment [23]. However, recently, a Proximal Policy Optimization (PPO) algorithm was used to achieve control of a real morphing airfoil [23]. PPO is a Deep Reinforcement Learning algorithm (DRL): it combines Artificial Neural Networks with a RL algorithm to estimate both the state's value and the policy functions. PPO is also a model free method, as no modeling of the environment is needed. As for the last "ingredient" of the RL algorithm, the reward signal (in the paper [23], it is referred to as 'reward scheme') two definitions have been implemented: both are based on the relative errors between measured and achieved deflection of the structure, with one also implementing a penalty function to take into account unwanted overshoot. To distinguish between the controllers trained with a different reward scheme, the simplest one is represented as RL while MO (mitigate overshoot) points to the other.

The control scheme is shown in figure 1.15. On the right, the airfoil structure with its morphing trailing edge is shown. On the bottom left of the picture, the RL controllers are represented: the control architecture produces a voltage signal which is modulated and sent to the piezoelectric actuated skin of the morphing trailing edge. The deflection is reconstructed by feeding the real measurements of piezoelectric sensors to an inference model, which yields the true deflection, measured by a profilometer. The inference model was built using two different neural networks: a traditional linear (LIN) and a long short-term memory (LSTM) network.

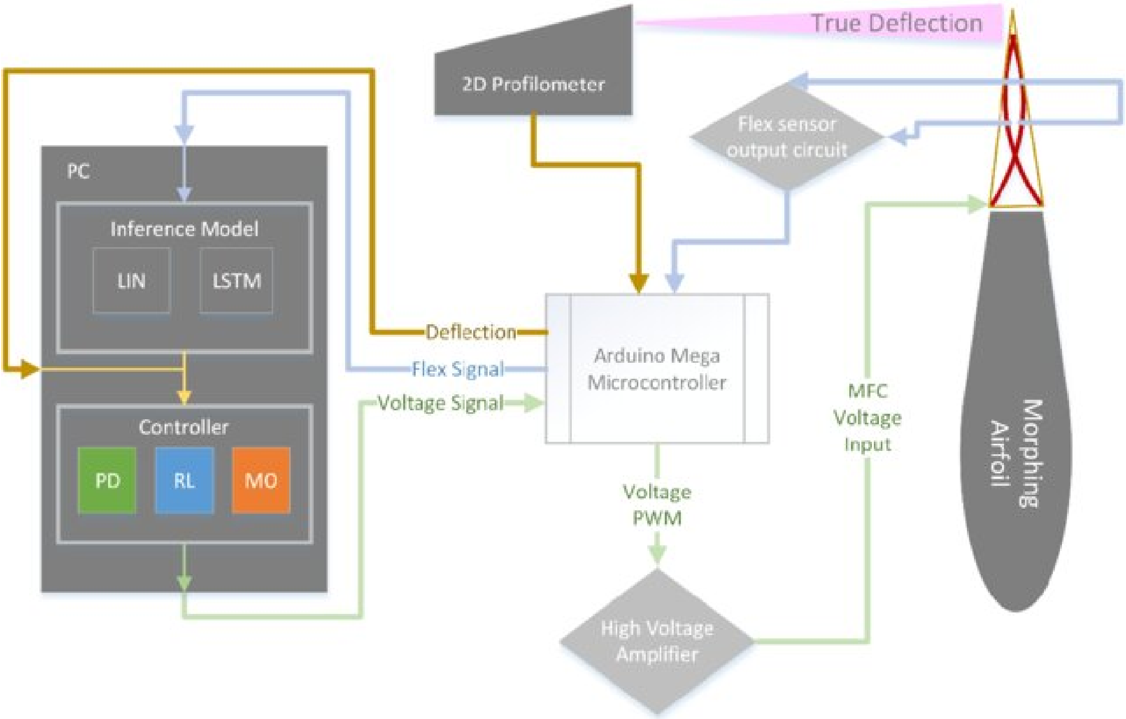


Figure 1.15: Deep Reinforcement Learning control scheme [23]

Chapter 2

The inverse Finite Element Method

The morphing designs and control strategies presented in the previous chapter implemented various feedback strategies:

- in [23] the feedback deflection was measured by coupling piezoelectric sensor results with inference models obtained through neural networks;
- in [17] encoders measuring relative rotation between the kinematic rigid link mechanism were used to monitor the shape of the structure;
- in [14] Fiber Bragg Grating strain measurements were fed to pre-known transfer functions to produce the control loop feedback.

Although very different, these strategies all rely on sensor data rather than actual shape measurements to reconstruct the shape of the morphing wing. In this master thesis, in contrast to the commonly used sensor feedback controllers, a shape-based feedback closed-loop control strategy is developed and tested using the shape sensing methodology known as the inverse Finite Element Method.

Shape sensing is defined as the inverse problem of reconstructing the deformed shape of a structure using surface strain measurements [25]. In the inverse Finite Element Method, the strain measurements are used to build a least-squares difference functional, Φ. Seeking an extremum of this functional, i.e. enforcing its first variation to be null ($\delta\Phi = 0$), yields Euler equations and consistent boundary conditions that relate the measured strains to the unknown displacement field. In other words, iFEM is a least-squares variational principle based method.

The structural framework model for the inverse Finite Element Method is the first-order, shear-deformation theory (FSDT or Mindlin theory). With reference to figure 2.1, the FSDT kinematic model relations are

$$\begin{cases} u_x(x,y,z) = u(x,y) + z\theta_y(x,y) \\ u_y(x,y,z) = v(x,y) + z\theta_x(x,y) \\ u_z(x,y,z) = w(x,y) \end{cases} \tag{2.1}$$

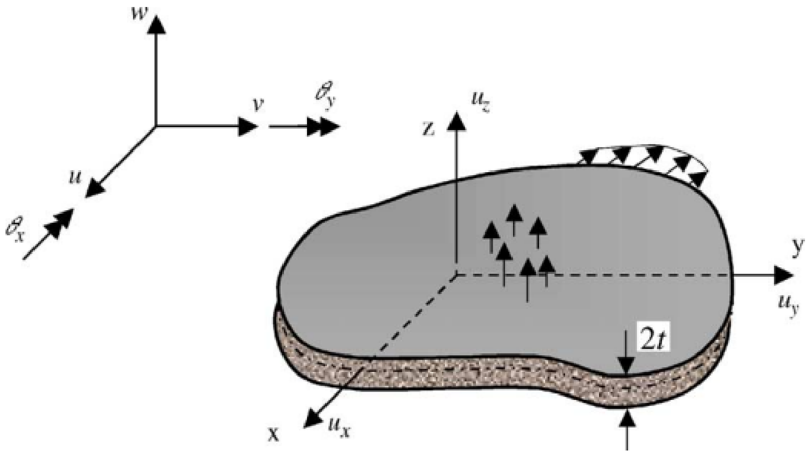


Figure 2.1: Notation for the flat shell [25]

where the five kinematic variables $u, v, w, \theta_x, \theta_y$ all refer to displacements and rotation of the middle plane of the shell structure. The kinematic model of Eqs. 2.1 leads to the following expression for the in-plane strains:

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix} + z \begin{Bmatrix} \kappa_{x0} \\ \kappa_{y0} \\ \kappa_{xy0} \end{Bmatrix} \equiv \mathbf{e}(\mathbf{u}) + z\mathbf{k}(\mathbf{u}) \tag{2.2}$$

where $\mathbf{u}$ is the vector of kinematic variables, $\mathbf{e(u)}$ is the vector of membrane strains, and $\mathbf{k(u)}$ the vector of curvatures. Both vectors are related to the five kinematic variables by 3×5 differential operator matrices:

$$\begin{aligned} \mathbf{e}(\mathbf{u}) &= \mathbf{L}_\varepsilon \mathbf{u} \\ \kappa(\mathbf{u}) &= \mathbf{L}_\kappa \mathbf{u} \end{aligned} \tag{2.3}$$

Similarly, transverse shear strains are related to $\mathbf{u}$ via a 2×5 differential operator matrix:

$$\begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \equiv \begin{Bmatrix} \gamma_{xz0} \\ \gamma_{yz0} \end{Bmatrix} \equiv \mathbf{g}(\mathbf{u}) = \mathbf{L}_\gamma \mathbf{u} \tag{2.4}$$

In the FSDT framework, the measured strains are to be correlated to the membrane strains and to the curvatures. Referring to the notation of figure 2.1, let the position of the i-th strain sensor pair (for instance, a couple FBG sensors or a strain rosettes) on the top or bottom surface of the plate be $\mathbf{x_i} = (x_i, y_i, \pm t)$. The strains measured by the sensor on the top-skin are $(\varepsilon^+_{xx}, \varepsilon^+_{yy}, \gamma^+_{xy})$. The strains measured on the bottom are $(\varepsilon^-_{xx}, \varepsilon^-_{yy}, \gamma^-_{xy})$. Having sensors on both the top and bottom surface, the measured strains can be converted into membrane strains and curvatures using the following equations:

$$\mathbf{e_i^\varepsilon} \equiv \begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix}_i = \frac{1}{2}\left(\begin{Bmatrix} \varepsilon_{xx}^+ \\ \varepsilon_{yy}^+ \\ \gamma_{xy}^+ \end{Bmatrix}_i + \begin{Bmatrix} \varepsilon_{xx}^- \\ \varepsilon_{yy}^- \\ \gamma_{xy}^- \end{Bmatrix}_i \right), \qquad (i = 1, n) \tag{2.5}$$

$$\mathbf{k_i^\varepsilon} \equiv \begin{Bmatrix} \kappa_{x0} \\ \kappa_{y0} \\ \kappa_{xy0} \end{Bmatrix}_i = \frac{1}{2t}\left(\begin{Bmatrix} \varepsilon_{xx}^+ \\ \varepsilon_{yy}^+ \\ \gamma_{xy}^+ \end{Bmatrix}_i - \begin{Bmatrix} \varepsilon_{xx}^- \\ \varepsilon_{yy}^- \\ \gamma_{xy}^- \end{Bmatrix}_i \right), \qquad (i = 1, n) \tag{2.6}$$

where n is the number of sensors, the i subscript stands for $\mathbf{x} = \mathbf{x_i}$, and the superscript ε represents the existence of errors in experimental measurements.

These measured strains, together with the FSDT strains of Eqs. 2.3 and 2.4, are used to formulate the least-squares difference variational principle that constitutes the basis of iFEM. The least-squares difference functional is presented in the following expression [26]:

$$\Phi_e^\lambda(\mathbf{u}) = \left\|\mathbf{e}(\mathbf{u}) - \mathbf{e}^\varepsilon\right\|^2 + \left\|\mathbf{k}(\mathbf{u}) - \mathbf{k}^\varepsilon\right\|^2 + \lambda\left\|\mathbf{g}(\mathbf{u}) - \mathbf{g}^\varepsilon\right\|^2 \tag{2.7}$$

where each norm is defined as follows:

$$\begin{aligned}
&\left\|\mathbf{e}(\mathbf{u}) - \mathbf{e}^\varepsilon\right\|^2 \equiv \frac{1}{A_e}\int_{A_e} \left[\mathbf{e}(\mathbf{u}) - \mathbf{e}^\varepsilon\right]^2 \, dA_e \\
&\left\|\mathbf{k}(\mathbf{u}) - \mathbf{k}^\varepsilon\right\|^2 \equiv \frac{(2t)^2}{A_e}\int_{A_e} \left[\mathbf{k}(\mathbf{u}) - \mathbf{k}^\varepsilon\right]^2 \, dA_e \\
&\left\|\mathbf{g}(\mathbf{u}) - \mathbf{g}^\varepsilon\right\|^2 \equiv \frac{1}{A_e}\int_{A_e} \left[\mathbf{g}(\mathbf{u}) - \mathbf{g}^\varepsilon\right]^2 \, dA_e \quad \textit{or} \quad \left\|\mathbf{g}(\mathbf{u}) - \mathbf{g}^\varepsilon\right\|^2 \equiv \int_{A_e} \mathbf{g}(\mathbf{u})^2 \, dA_e
\end{aligned} \tag{2.8}$$

In order to build the inverse Finite Element Model, an approximation of the distribution of the kinematic variables on the middle plane surface is introduced. Under the assumptions of the displacements finite element method, $\mathbf{u}$ is expressed as a function of $\mathbf{u_e}$, the nodal displacement vector of the inverse finite element:

$$\mathbf{u}(\mathbf{x}, \mathbf{y}) = \mathbf{N}(\mathbf{x}, \mathbf{y})\mathbf{u_e} \tag{2.9}$$

Equation 2.9 represents the approximation introduced by the finite element discretization as it is assumed that the displacement field over the element can be obtained by interpolating the nodal displacements using appropriate shape functions contained in the matrix $\mathbf{N}$. Two different inverse finite elements have been developed:

- a three-node triangular element, referred to as iMIN3 [26], with 5 DOF (degrees of freedom) at each node

$$\mathbf{u_e}^T = \{u_i, v_i, w_i, \theta_{xi}, \theta_{yi}\} \qquad i = 1{,}2{,}3 \tag{2.10}$$

- a four-node quadrilateral element, the iQS4 [27], with 6 DOF (degrees of freedom) at each node

$$\mathbf{u_e}^T = \{u_i, v_i, w_i, \theta_{xi}, \theta_{yi}, \theta_{zi}\} \qquad i = 1{,}2{,}3{,}4 \tag{2.11}$$

For both these inverse finite elements, the shape functions are always C^0-continuous anisoparametric functions. (For the sake of simplicity the shape functions are not presented in this work. Refer to [26] and [27] for further details). Introducing the inverse finite element approximation in Eqs. 2.3 and 2.4, the membrane and transverse strains, as well as the curvatures, can be expressed as functions of the nodal displacements:

$$\begin{aligned}
\mathbf{e}(\mathbf{u_e}) &= \mathbf{L}_\varepsilon \mathbf{N}\mathbf{u_e} = \mathbf{B^m}\mathbf{u_e} \\
\kappa(\mathbf{u_e}) &= \mathbf{L}_\kappa \mathbf{N}\mathbf{u_e} = \mathbf{B^b}\mathbf{u_e} \\
\mathbf{g}(\mathbf{u_e}) &= \mathbf{L}_\gamma \mathbf{u_e} = \mathbf{B^b}\mathbf{u_e}
\end{aligned} \tag{2.12}$$

where $\mathbf{B_m}$, $\mathbf{B_b}$, and $\mathbf{B_s}$ are matrices of shape function derivatives corresponding to the membrane, bending, and transverse shear strains [28].

Substituting Eqs. 2.12 and Eqs. 2.8 in the equation 2.7 of the least-squares functional, a resulting expression that depends only on the element nodal displacements is obtained. Therefore, minimizing the functional leads to the following problem:

$$\frac{\partial \Phi_e(\mathbf{u_e})}{\partial \mathbf{u_e}} = \mathbf{K_e}\mathbf{u_e} - \mathbf{f_e} = 0 \implies \mathbf{K_e}\mathbf{u_e} = \mathbf{f_e} \tag{2.13}$$

which is the element matrix equation, where $\mathbf{K_e}$ is a function of the positions of the strain sensors and the vector $\mathbf{f_e}$ depends on the measured strain data. Through appropriate rotations of the element vectors and coefficient matrices, and using finite element assembling techniques, the inverse Finite Element Model produces a system of algebraic equations

$$\mathbf{KU} = \mathbf{F} \tag{2.14}$$

whose solution yields the nodal displacements of the structure.

To obtain a non-singular coefficient matrix, boundary conditions are to be imposed. On the contrary, to solve for the displacements of the structure, elastic or inertial material properties can be unknown. This constitutes a critical advantage of the iFEM procedure, leading to its applications in both static and dynamic systems.

In the next chapter, it is shown how this potential was exploited to build a real-time shape-based feedback for closed-loop control of a morphing structure.

Chapter 3

Control loop architecture

The objective of this master thesis is the performance evaluation of an inverse Finite Element shape feedback system in a close-loop control strategy for a morphing structure. The actuation, as well as the measurements of the strains of the structure, were simulated in a FEM environment. The closed-loop control operations were run through an "in-house" developed MATLAB code. This chapter contains a description of each element of the closed-loop architecture and an explanation of how those elements were implemented together. Therefore, after a brief introduction to the notation used, the closed-loop flowchart is presented. The flow of information in the controller cycle is highlighted focusing, in particular, on the various inputs and outputs of the closed-loop building blocks, which are initially presented as "black boxes". Following these introductory paragraphs, the different aspects and the operating principles of each component of the closed-loop control architecture are detailed in a dedicated chapter. The covered topics are the following:

- how the target shape is obtained;
- how the actuation loads are computed: a least-squares inverse approach is adopted. As the problem at hand is ill-posed, regularization methods are introduced in a dedicated subparagraph;
- how the FEM strains and displacements are retrieved at each iteration;
- how the external disturbances caused by the aerodynamic loads are implemented in the closed-loop cycle;
- how the errors and convergence criteria are defined.

3.1 Notation introduction

Before starting with the description of the closed-loop control strategy, it is instrumental to explain the notation that will be used in the following paragraphs.

One of the most recurring input and output data in the closed-loop cycle is the shape of the structure, since the objective is its monitoring. The letter used for the different shapes is P. A few examples are presented:

- $\mathbf{P_{tgt}}$ is the target shape;
- $\mathbf{P_{iFEM}}$ is the inverse Finite Element Model reconstructed shape;
- $\mathbf{P_{FEM}}$ is the shape obtained using the Finite Element Model of the structure.

The shape, and thus $\mathbf{P}$, is defined in terms of displacements. In other words, $\mathbf{P}$ is a matrix containing the nodal displacements between the unmorphed shape and the shape $\mathbf{P}$ refers to. Each column of a $\mathbf{P}$ matrix corresponds to a displacement component. Therefore, $\mathbf{P}$ is always a $N \times 3$ matrix

$$\mathbf{P} = \begin{bmatrix} u_1 & v_1 & w_1 \\ \vdots & \vdots & \vdots \\ u_N & v_N & w_N \end{bmatrix} \tag{3.1}$$

where N is the number of nodes.

As explained in chapter 2, a fundamental input of the inverse Finite Element Method are the back-to-back in-plane strains of the surfaces of the structure. In the FE models used for the closed-loop control testing (an in detail description of these models will be presented in chapters 4.1 and 5.1.2), it was assumed that each element of the skin of the structure was instrumented on both the top and bottom surface with a strain rosette. Hence, the strain matrices, which will be always referred to as $\boldsymbol{\varepsilon}$, are $N_e \times 6$ matrices

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon^{+}_{xx,1} & \varepsilon^{-}_{xx,1} & \varepsilon^{+}_{yy,1} & \varepsilon^{-}_{yy,1} & \gamma^{+}_{xy,1} & \gamma^{-}_{xy,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varepsilon^{+}_{xx,N_e} & \varepsilon^{-}_{xx,N_e} & \varepsilon^{+}_{yy,N_e} & \varepsilon^{-}_{yy,N_e} & \gamma^{+}_{xy,N_e} & \gamma^{-}_{xy,N_e} \end{bmatrix} \tag{3.2}$$

where N_e is the number of instrumented elements, and $(\varepsilon^{+}_{xx}, \varepsilon^{+}_{yy}, \gamma^{+}_{xy})_i$ and $(\varepsilon^{-}_{xx}, \varepsilon^{-}_{yy}, \gamma^{-}_{xy})_i$ are respectively the top and bottom in plane strains measured at the centroid of the i-th element.

Lastly, the notation for the actuation load scheme is presented. For the models used to simulate the actuation of the morphing structure, a set of sparse concentrated transverse forces or element pressures was used. The value of each single load is contained in a column vector, $\mathbf{L}$, of N_L elements

$$\mathbf{L}^T = \{L_1, \cdots, L_i, \cdots, L_{N_L}\} \tag{3.3}$$

where N_L is the number of sparse actuation loads.

3.2 Control loop flow chart

In figure 3.1, the closed-loop control architecture is presented. Starting from the bottom-left corner, at each iteration the controller checks if the iFEM reconstructed shape has converged to the target shape. If convergence hasn't been reached, iteration is needed. Therefore the difference between the target shape and the reconstructed shape, $\mathbf{P_{tgt}} - \mathbf{P_{iFEM}}$, is used to compute the actuation load vector, $\mathbf{L}$, which in turn is fed to the finite element model of the initial undeformed structure. As an output, the FEM procedure yields the FEM simulated measurements of the strains of the morphed structure, $\boldsymbol{\varepsilon_{FEM}}$. These strains are inputted into the iFEM procedure, which reconstructs the morphed shape that, coherently with the notation of the previous chapter, is expressed in terms of displacements with respect to the unmorphed shape. The iFEM reconstructed shape is then sent back as feedback to the controller, thus completing the closed-loop cycle.

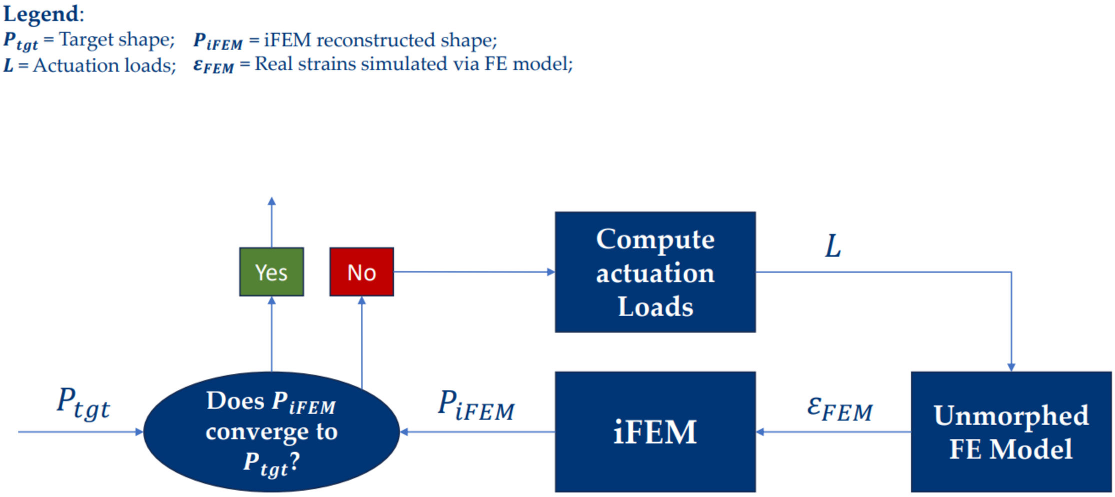

Legend:
P_{tgt} = Target shape; P_{iFEM} = iFEM reconstructed shape;
L = Actuation loads; ε_{FEM} = Real strains simulated via FE model;
Yes
No
Compute actuation Loads
L
P_{tgt}
Does P_{iFEM} converge to P_{tgt}?
P_{iFEM}
iFEM
ε_{FEM}
Unmorphed FE Model

Figure 3.1: Closed-loop control architecture

In the closed loop architecture presented in figure 3.1, the main source of error is the iFEM reconstructed shape. As iFEM has already proved its potential applications in monitoring the shape of a morphing structure in a control strategy [6], fast convergence is expected in this case study. This scenario, however, is far distinct from a real one: during flight, a number of external unknown disturbances act on the structure and modify its shape. An example of such unknown disturbances could be the pressure distribution acting on the morphing structure: as the shape of the morphing wing changes, the pressure distribution on the structure varies and can act both against or together with the actuation force. In both cases, if open-loop or fed-forward control were to be used the controller would not be able to achieve the target shape. In this case, that is when an external unknown action is present on the morphing structure, shape control can be obtained only with a closed-loop architecture.

To test the efficacy of the closed-loop architecture, therefore, an external disturbance, modeled as the pressure distribution acting on the structure, is introduced in the closed-loop cycle. The modified architecture is shown in figure 3.2. In the top-right corner of the flowchart, a new "black box" titled XFOIL is now present: the pressure distribution is computed by feeding the deformed FEM simulated shape to XFOIL, which produces the pressure distribution on the morphed profile. Then, in the same way as the actuation loads are fed to the FE model, the pressure distribution is introduced onto the initial unmorphed FEM structure to simulate its deformation. Varying the intensity of the pressure distribution, this new closed-loop architecture allows to test the iFEM feedback even under extreme external conditions.

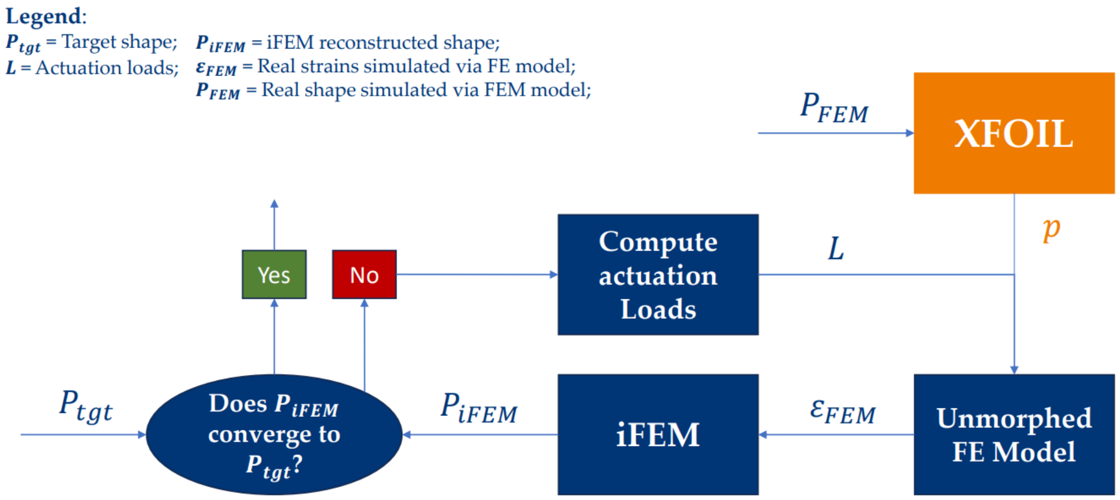

Legend:
P_{tgt} = Target shape;
L = Actuation loads;
P_{iFEM} = iFEM reconstructed shape;
ε_{FEM} = Real strains simulated via FE model;
P_{FEM} = Real shape simulated via FEM model;
P_{FEM}
XFOIL
p
Compute actuation Loads
L
Yes
No
P_{tgt}
Does P_{iFEM} converge to P_{tgt}?
P_{iFEM}
iFEM
ε_{FEM}
Unmorphed FE Model

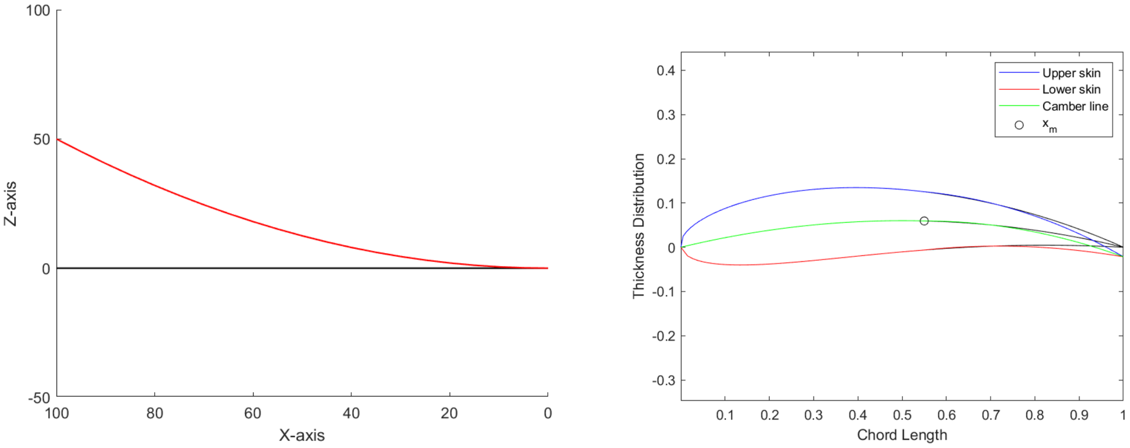
Figure 3.2: Closed-loop control architecture - Added external disturbance

3.3 Target shape

Now that the overall view of the closed-loop control strategy has been presented, an in-depth description of each of its elements will be provided. Starting from the target shape, it was decided to test the morphing structures only in bending, that is to concentrate the morphing action in the transverse direction. This choice is coherent with most of the designs [10, 15–17, 23] presented in chapter 1.2 and 1.3 as camber morphing has shown the most promising results when evaluated in both control authority and aerodynamic efficiency [23].

Two morphing structures were tested: a simple plate and a wing structure. The target shape of the morphing plate was obtained by imposing an arbitrary equation for the transverse deflection. It was assumed that $w(x) = x^2$. The target shape (in red) and the unmorphed shape (in black) are represented in figure 3.3a.

Between camber morphing wings, morphing trailing edge concepts have been proven to be superior to correspondent leading edge designs [15]. Therefore, morphing capabilities are concentrated in the trailing edge of the tested wing structure and morphing is used only to change the camber of the trailing edge profile. In figure 3.3b, the unmorphed shape (in black) and the target shape are portrayed. The target shape is obtained by modifying the typical equation of the camber line of the four-digit NACA profile to an arbitrary parabolic shape. The upper and lower skin of the profile, since their equations both depend on the equation of the camber line, morph accordingly. Once the equations of the arbitrary parabolic shape are obtained, they are used to compute the nodal coordinates of the morphed structure, according to the FE discretization. Then, by a simple subtraction of the baseline nodal coordinates of the unmorphed structure, the displacement matrix $\mathbf{P_{tgt}}$ is obtained.



(a) Target shape of the morphing plate structure

(b) Target shape of the morphing wing

Figure 3.3: Target shape of the tested morphing structures

3.4 Actuation loads

In chapter 1.3 the state-of-the-art control architectures for morphing wing structures were presented: the main employed strategies are PID (Proportional-Integral-Derivative) control and RL (Reinforcement learning) techniques. As the objective of this master thesis is the implementation of the inverse Finite Element Method as a feedback strategy, instead on focusing on the implementation of a complex modern control strategy, a simpler solution was chosen. Inspired by [6], the controller output is based on the minimisation of a least-squares difference functional between the required displacements and the displacements generated by the actuation loads. The least-squares approach yields an inverse problem, much similar to the one characterising the inverse Finite Element Method [25]. Inverse problems are ill-posed, meaning that they do not necessarily satisfy conditions of existence, uniqueness, and stability. In particular, instabilities can cause great changes in the output results in the presence of even small disturbances in the input data. This nature of ill-posed problems calls for solutions to improve the robustness of the method and damp the effect of disturbances. These solutions are called 'regularization methods'. Therefore, this paragraph is subdivided into two sections: in the first one, the least-squares approach used to compute the loads is presented. Subsequently, regularization methods, in particular Tikhonov Regularization, are introduced.

3.4.1 Least-squares approach

In chapter 3.2, it was explained how the function that computes the actuation loads receives as an input the difference between the target and iFEM reconstructed displacements, $\mathbf{P_{tgt}} - \mathbf{P_{iFEM}}$. This difference produces the required displacements, $\mathbf{D_{ref}}$, that the structure needs to achieve the target shape

$$\mathbf{D_{ref}} = \mathbf{P_{tgt}} - \mathbf{P_{iFEM}} \tag{3.4}$$

where the letter D was used because, differently from the **P** displacement matrices, the required displacements are not measured with respect to the unmorphed shape.

To compute the loads that produce $\mathbf{D_{ref}}$, a relation between the actuation scheme and the displacements is introduced.

Suppose that the number of sparse actuation points on the structure is N_L. Assume that at each actuation point, a concentrated load or moment can be produced in any direction of the reference coordinate system. If the subscript 'm' is used to point to the generic load point then the load vector repetitive unit is

$$\mathbf{L}_m^T = \{N_x, N_y, N_z, M_x, M_y, M_z\}_m, \qquad m = 1, ..., N_L \tag{3.5}$$

As a result, the actuation scheme load vector, **L** is a $6N_L \times 1$ column vector. To relate this load vector to the displacements of the structure the following steps are followed:

- in the FEM environment, (i.e. in Patran), for each actuation load (that is for all of the $6N_L$ loads) a load case is created where a unit value is assigned to that actuation load while all the others are considered null;
- the $6N_L$ load cases of a single unit load are employed in just the same amount of static analysis (run using Nastran);
- through this process $6N_L$ different nodal displacement fields are obtained. The generic nodal displacement field, $\mathbf{P_m}$, is a $N \times 3$ matrix of displacements components

$$\mathbf{P_m} = \begin{bmatrix} u_1 & v_1 & w_1 \\ \vdots & \vdots & \vdots \\ u_N & v_N & w_N \end{bmatrix}_m, \qquad m = 1, ..., 6N_L \tag{3.6}$$

  where N is the number of nodes of the FEM discretization;
- the generic displacement matrix $\mathbf{P_m}$ is rearranged, lining up in a column vector each of its rows:

$$\mathbf{P_m}^T = \{u_1, v_1, w_1, u_2, v_2, w_2, ..., u_N, v_N, w_N\}_m \qquad m = 1, ..., 6N_L \tag{3.7}$$

- lastly, assembling all the $\mathbf{P_m}$ column vectors in a single matrix, the matrix of influence coefficients, **A** is obtained:

$$\mathbf{A} = [\{P_1\} \cdots \{P_{6N_L}\}] \tag{3.8}$$

The matrix of influence coefficients is a $3N \times 6N_L$ matrix. For a given actuation scheme, and for any set of values of the actuation loads represented by the load vector **L**, the correlated displacements are obtained via this simple equation:

$$\mathbf{P} = \mathbf{AL} \tag{3.9}$$

The **P** matrix obtained in Eqn. 3.9, as explained in chapter 3.1, contains the displacements between the unmorphed shape and the shape **P** refers to. However, as explained previously, the function that computes the loads uses as an input the difference between two $\mathbf{P}$ matrices, the one related to the target shape and the one obtained via the iFEM reconstruction. The equation that relates the matrix of influence coefficients to the difference between two generic displacement matrices, $\mathbf{P_k}$ and $\mathbf{P_{k+1}}$, can be obtained using Eqn. 3.9 in the following procedure:

$$\mathbf{D_k} = \mathbf{P_{k+1}} - \mathbf{P_k} = \mathbf{AL_{k+1}} - \mathbf{AL_k} = \mathbf{AdL_k} \tag{3.10}$$

where $\mathbf{dL_k}$ is the load increment between the actuation loads that produce $\mathbf{P_{k+1}}$ and $\mathbf{P_k}$.

At each k-th iteration, Eqn. 3.10, together with the required displacement $\mathbf{D_{ref,k}}$, allows to build the least-squares difference functional

$$\Phi_k(\mathbf{dL_k}) = \left\|\mathbf{D_{ref,k}} - \mathbf{D_k}\right\|^2 = \left\|\mathbf{D_{ref,k}} - \mathbf{AdL_k}\right\|^2 \tag{3.11}$$

where the subscript 'k' is used for $\mathbf{D_{ref}}$ too, since a different iFEM reconstructed shape, $\mathbf{P_{iFEM,k}}$, can be obtained at each iteration.

The only independent variable of the least-squares functional is the load vector increment. Hence, the problem of minimizing Eqn. 3.11 leads to the following least-squares problem:

$$\min_{\mathbf{dL_k}} \left\|\mathbf{D_{ref,k}} - \mathbf{AdL_k}\right\|^2 \implies \mathbf{dL_k} = \mathbf{A}^\dagger \mathbf{D_{ref,k}} \tag{3.12}$$

whose solution, as shown in the implication, is the load vector increment itself. $\mathbf{dL_k}$, however, is not the final output of the process yet. $\mathbf{L}$, the actuation load vector that, as shown in 3.2, is the actual controller output, and that is used as an input for the FEM procedure, is obtained at the K-th iteration with the following expression:

$$\mathbf{L} = \sum_{k=1}^{K} \mathbf{dL_k} \tag{3.13}$$

As shown in equation 3.12 a powerful way of solving the least-squares functional problem is through the singular value decomposition (SVD) of the matrix $\mathbf{A}$, in particular using the pseudo-inverse matrix $\mathbf{A}^\dagger$ [29].

The singular value decomposition of a generic $m \times n$ $\mathbf{A}$ matrix is

$$\mathbf{A} = \mathbf{U\Sigma V^T} \tag{3.14}$$

where $\mathbf{U} \in \mathbb{R}^{m\times n}$ is an orthonormal column matrix, $\mathbf{V} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, and $\mathbf{\Sigma} \in \mathbb{R}^{n\times n} = diag(\sigma_1, ..., \sigma_n)$ is the diagonal matrix of the singular values of $\mathbf{A}$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are defined respectively as the right and left singular vectors of $\mathbf{A}$. It is common understanding that the columns of $\mathbf{U}$ and $\mathbf{V}$ are ordered and scaled such that the singular values are non-negative and are ordered by magnitude:

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \geq 0 \tag{3.15}$$

Using SVD the generic least-square minimisation problem

$$\min_{\mathbf{b}} \left\|\mathbf{b} - \mathbf{Ax}\right\|^2 \tag{3.16}$$

is solved using the following expression:

$$\mathbf{x} = \sum_{j=1}^{r} \frac{\mathbf{u_j^T b}}{\sigma_j} \mathbf{v_j} \tag{3.17}$$

where r is the rank of the **A** matrix. If A has full rank then the solution of Eqn. 3.17 can be easily obtained as

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b} \tag{3.18}$$

where $\mathbf{A}^{\dagger}$, the pseudo inverse matrix, is defined as

$$\mathbf{A}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{\dagger}\mathbf{U}^{\mathbf{T}}, \qquad where \quad \mathbf{\Sigma}^{\dagger} = diag(1/\sigma_i), \quad i = 1, n \tag{3.19}$$

3.4.2 Regularization of the least-square method

As mentioned in the introduction to this paragraph, the least-squares approach adopted to compute the loads produces an ill-posed problem. If the matrix of influence coefficients has full rank the conditions of existence and uniqueness of the solution are solved. However, the problem still remains afflicted by stability issues. To understand how even small disturbances can affect the solution of an arbitrary ill-posed problem such as the one in Eqn. 3.16, the random noise vector **s** is defined so that

$$\tilde{\mathbf{b}} = \mathbf{b} + \varepsilon\mathbf{s} \tag{3.20}$$

with ε small and greater than zero. Substituting Eqn. 3.20 in the solution obtained via SVD (Eqn. 3.17), yields the following expression:

$$\tilde{\mathbf{x}} = \sum_{j=1}^{r} \frac{\mathbf{u_j^T b}}{\sigma_j}\mathbf{v_j} + \varepsilon \sum_{j=1}^{r} \frac{\mathbf{u_j^T s}}{\sigma_j}\mathbf{v_j} \tag{3.21}$$

The solution presented in Eqn. 3.21 consists of two terms: the first one is the true solution of the least square problem. The second term contains in the numerator the projection of the noise vector onto the left singular vectors, $\mathbf{u_j^T s}$, and at the denominator the singular values of **A**. If the numerator stays roughly constant, since as shown in Eqn. 3.15 the singular values are in descending order of magnitude, the second term blows up with j, causing instabilities even for small values of ε. To prevent instabilities, the so-called 'regularization methods' are employed. Roughly speaking, the objective of regularization is to contain the negative and unstable effect of the high index singular values, i.e. the smallest ones, trying to maintain at the same time coherence with the model and adherence to the real solution. This is obtained by introducing in the regularized solution a diagonal matrix,$\mathbf{\Theta}$, of filter factors f_i:

$$\mathbf{x_{reg}} = \mathbf{V}\mathbf{\Theta}\mathbf{\Sigma}^{\dagger}\mathbf{U^T}\tilde{\mathbf{b}} = \sum_{j=1}^{r} \frac{\mathbf{u_j^T b}}{\sigma_j}\mathbf{v_j} + \varepsilon \sum_{j=1}^{r} f_j \frac{\mathbf{u_j^T s}}{\sigma_j}\mathbf{v_j} \tag{3.22}$$

Regularization is obtained in Eqn. 3.22 adjusting the values of the filter factor so that the unwanted components of the singular value decomposition are damped whereas the wanted components remain essentially unchanged.

The simplest regularization method is the truncated singular value decomposition (TSVD). This method consists in substituting the **A** matrix with its best rank-k approximation, which is obtained using the following expression:

$$\mathbf{A_k} = \sum_{j=1}^{k} \sigma_j \mathbf{u_j v_j^T} \tag{3.23}$$

where $k < r$, the rank of the $\mathbf{A}$ matrix. Introducing this expression in Eqn. 3.18, the regularized solution becomes

$$\mathbf{x_{reg}} = \mathbf{A_k^\dagger}\tilde{\mathbf{b}} = \sum_{j=1}^{k} \frac{\mathbf{u_j^T b}}{\sigma_j}\mathbf{v_j} + \varepsilon \sum_{j=1}^{k} \frac{\mathbf{u_j^T s}}{\sigma_j}\mathbf{v_j} \tag{3.24}$$

Comparing Eqn. 3.24 with Eqn. 3.22, it is easily deducible that the filter factors are

$$f_j = \begin{cases} 1, & \text{for } j \leq k \\ 0, & \text{for } j > k \end{cases} \tag{3.25}$$

The error associated with the TSVD method can be computed in the Frobenius norm as

$$\left\|\mathbf{A} - \mathbf{A_k}\right\|_2 = \sigma_{k+1} \tag{3.26}$$

Eqn. 3.26, together with Eqn. 3.25, perfectly portrays the compromise of regularization: a small value of k, which would lead to a highly regularized solution, would also result in a great error as the singular values decrease with k. On the other hand, a small error is attainable only with a high value of k which would in turn result in a poorly regularized solution. A method that allows for a better compromise between these two solutions is the Tikhonov regularization method. Tikhonov's method is one of the most refined and widespread regularization methods, finding possible applications even in the regularization of the inverse Finite Element least-squares problem [25]. In Tikhonov's approach the problem of Eqn. 3.16 is reformulated as

$$\min_{\mathbf{b}} \left\|\tilde{\mathbf{b}} - \mathbf{Ax}\right\|^2 + \lambda \left\|\mathbf{Lx}\right\| \tag{3.27}$$

Regularization is introduced in the solution to the least-squares problem of Eqn. 3.27 since the regularized solution, $\mathbf{x}_\lambda$, has to compromise between minimizing the residual $\tilde{\mathbf{b}} - \mathbf{Ax}$ and the penalty function $\mathbf{Lx}$. The parameter *lambda* is used to control the influence of the penalty functions and thus the "degree" of regularization. The solution to the regularized Tikhonov problem in the standard case, where $\mathbf{L} = \mathbf{I}$, is presented in the following expression:

$$\mathbf{x_{reg}} = \mathbf{x}_\lambda = \sum_{j=1}^{n} \frac{\sigma_j \mathbf{u_j^T}\tilde{\mathbf{b}}}{\sigma_j^2 + \lambda}\mathbf{v_j} \tag{3.28}$$

The filter factors are

$$f_j = \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \tag{3.29}$$

Eqn. 3.29 shows how Tikhonov regularization exploits the value of the regularization parameter to reduce possible instabilities in the solution: for singular values much larger than λ the filter factors are $f_J \approx 1$ whereas for singular values much smaller than λ it holds that $f_J \approx \sigma_J^2/\lambda \approx 0$. Hence, the influence of the smallest singular values, which cause the noise term to blow up in Eqn. 3.21, is damped.

Tikhonov's method was chosen to regularise the least-squares problem used to compute the loads. Referring to Eqn. 3.12, the main source of noise is related to $\mathbf{D_{ref}}$ since it contains the iFEM reconstructed shape, $\mathbf{P_{iFEM}}$. The noise is produced by the inherent errors of the inverse Finite Element Method as well as, in a real-case scenario, by the error on the measured strains. Using the notation of Eqn. 3.20, the notation $\mathbf{\tilde{D}_{ref}}$ is used to point to the required displacements containing the noised term. Applying Tikhonov regularization in the standard case ($\mathbf{L} = \mathbf{I}$), the problem of Eqn. 3.12 is reformulated as

$$\min_{\mathbf{dL}} \left\| \begin{bmatrix} \mathbf{A} \\ \lambda \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{\tilde{D}_{ref}} \\ \mathbf{0} \end{bmatrix} \right\|^2 \implies \mathbf{dL} = \mathbf{A}^{\dagger}_{\mathbf{reg}} \begin{bmatrix} \mathbf{\tilde{D}_{ref}} \\ \mathbf{0} \end{bmatrix} \tag{3.30}$$

where $\mathbf{A}^{\dagger}_{\mathbf{reg}}$ is the pseudo inverse of the "regularized" matrix of influence coefficients

$$\mathbf{A_{reg}} = \begin{bmatrix} \mathbf{A} \\ \lambda \mathbf{I} \end{bmatrix} \tag{3.31}$$

3.5 FEM strains and displacements

In the introduction to this chapter, as well as in figure 3.1 and 3.2, it was mentioned how the actuation loads are used as input to the Finite Element Model to simulate the strain measurements and the shape change of the structure. In theory, both strains and displacements could be calculated by running an analysis with a proper FEM software, such as, for instance, Nastran. However, this procedure would require a high computational cost, that would in turn result in time-consuming simulated runs of the closed-loop architecture. Therefore, under the simplifying assumption of small deformations of the structure, a superposition principle-based approach was adopted. The superposition principle method, able to produce a computationally fast algorithm, was implemented following a similar approach to the one presented in chapter 3.4.1, where it was explained how the matrix of influence coefficients was built: for each actuation point, where both concentrated forces and bending moments can be produced, in a FEM environment such as Patran, a single unit load replaces all the other loads of the actuation scheme. This single unit load case (for a total of $6N_L$, where N_L is the total number of actuation points) is used to run a Nastran static analysis. After these first common steps, the following deviate from the procedure used for the matrix of influence coefficients and, thus, are presented hereafter:

- the $6N_L$ Nastran analysis produce the displacement and strain fields caused by each of the single unit loads. The generic nodal displacement field, $\mathbf{P_m}$, is a $N \times 3$ matrix of displacements components

  $$\mathbf{P_m} = \begin{bmatrix} u_1 & v_1 & w_1 \\ \vdots & \vdots & \vdots \\ u_N & v_N & w_N \end{bmatrix}_m \quad , \qquad m = 1, ..., 6N_L \tag{3.32}$$

  where N is the number of nodes of the FEM discretization. On the other hand, the strain measurements, following the notation introduced in chapter 3.1, are collected in a $N_e \times 6$ matrix

  $$\boldsymbol{\varepsilon_m} = \begin{bmatrix} \varepsilon^+_{xx,1} & \varepsilon^-_{xx,1} & \varepsilon^+_{yy,1} & \varepsilon^-_{yy,1} & \gamma^+_{xy,1} & \gamma^-_{xy,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varepsilon^+_{xx,N_e} & \varepsilon^-_{xx,N_e} & \varepsilon^+_{yy,N_e} & \varepsilon^-_{yy,N_e} & \gamma^+_{xy,N_e} & \gamma^-_{xy,N_e} \end{bmatrix}_m \quad , \quad m = 1, ..., 6N_L \tag{3.33}$$

where N_e is the number of elements, and the elements of $\boldsymbol{\varepsilon_m}$ are the back-to-back strains on those elements;

- these strain and displacement data collected before the closed-loop iterations, are then used during each cycle to simulate the actuation and the strain measurements. Using the superposition principle, at each iteration both the displacements and the strains are reconstructed through the following expressions:

$$\begin{aligned}\mathbf{P_{FEM}} &= \sum_{m=1}^{6N_L} L_m \mathbf{P_m},\\ \boldsymbol{\varepsilon_{FEM}} &= \sum_{m=1}^{6N_L} L_m \boldsymbol{\varepsilon_m}\end{aligned} \tag{3.34}$$

where L_m is the m component of the actuation load vector computed with Eqn. 3.30.

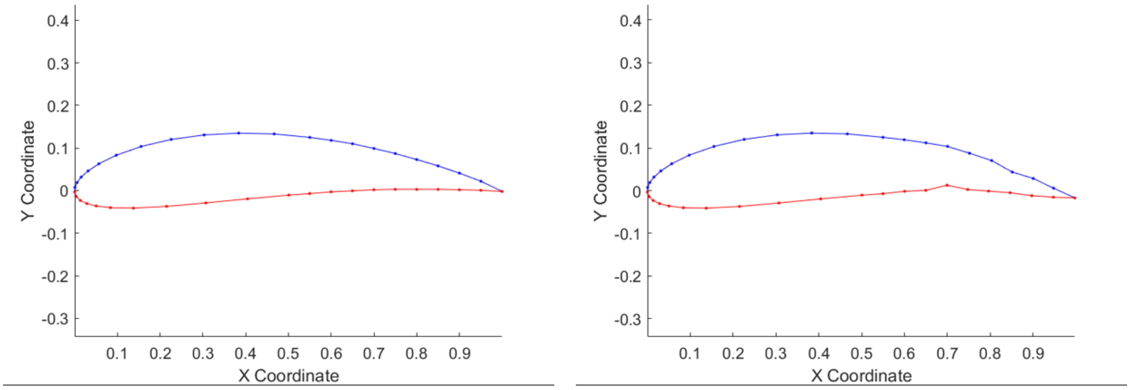
3.6 Aerodynamic loads

When presenting figure 3.2, it was explained that the closed-loop architecture was tested in presence of an unknown external load, which was modeled as the aerodynamic pressure distribution on the structure. To simplify the implementation of this external action on the structure some simplifying assumptions were made:

- the flow over the structure was considered inviscous. This assumption allowed to reduce the number of assumed parameters in the model;
- the pressure distribution along the span was considered constant. As explained in chapter 3.3, the target shape is obtained by modifying only the camber line on the trailing edge. Since the cross sections along the span, as the structure is subject to bending only, morph in the same way, a similar constant behaviour for the pressure distribution was assumed. This second assumption was fundamental in reducing the computational cost of the closed-loop strategy as, at each iteration, it was sufficient to evaluate the pressure distribution on a single reference cross-section, which was chosen as the middle one.

At each iteration of the closed-loop architecture, the effect of the pressure distribution on the structure is modeled through the following steps:

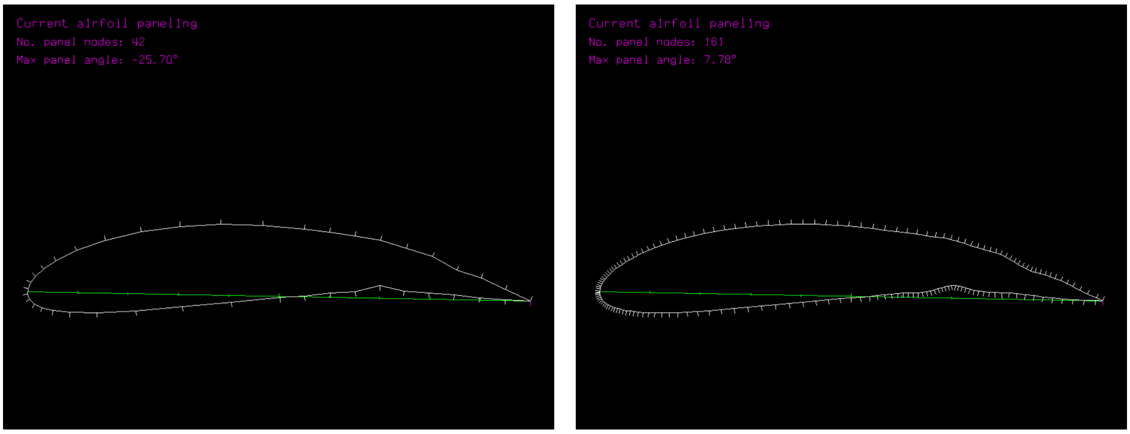
- the reference cross section is extracted by summing to its unmorphed baseline coordinates the nodal displacements of $\mathbf{P_{FEM}}$ obtained in the previous iteration. An example of the reconstructed cross-section along the iterations is represented in figure 3.4;
- the reconstructed morphed cross section is fed to XFOIL; in the XFOIL environment, the aerodynamic mesh is automatically constructed using a pre-determined number of aerodynamic panels. This number of panels, however, produced a very rough and uneven pressure distribution. Therefore, as shown in figure 3.5, the number of panels is adjusted to produce a smoother pressure distribution on the surface;



(a) Initial unmorphed configuration

(b) Final configuration

Figure 3.4: Reference cross-section reconstruction throughout the iterations



```
Current airfoil paneling
No. panel nodes: 42
Max panel angle: -25.70°
```

```
Current airfoil paneling
No. panel nodes: 161
Max panel angle: 7.78°
```

(a) $N_{panels} = 42$

(b) $N_{panels} = 161$

Figure 3.5: Refined aerodynamic paneling procedure

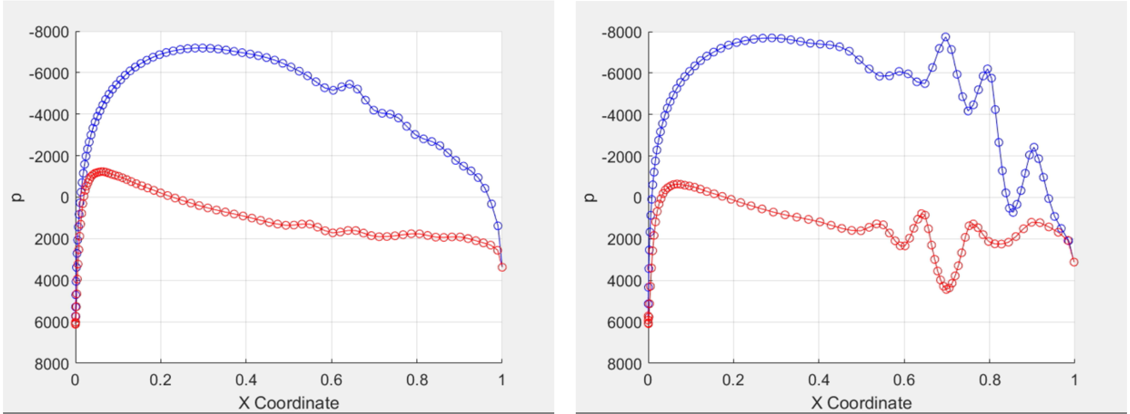
- XFOIL is run automatically to extract the pressure coefficient distribution on the cross-section. The inputs of the inviscous XFOIL analysis is only the angle of attack, which is set to a constant value of 0 degrees;

- the pressure coefficient distribution, $\mathbf{c_p}$ is converted into the pressure distribution on the structure using the following expression:

$$\mathbf{p} = \frac{1}{2}\rho V^2 \mathbf{c_p} \tag{3.35}$$

  Eqn. 3.35 allows to modulate the intensity of the external disturbance by simply acting on the relative velocity to the airflow. An example of the evolution of the pressure distribution on the airfoil is presented in figure 3.6;

- through spline interpolation of the pressure distribution on the reference cross-section, the pressure distribution at each element centroid of the FEM model is obtained.



(a) Initial unmorphed configuration

(b) Final unmorphed configuration

Figure 3.6: XFOIL reconstructed pressure distribution throughout iterations

Instead of running at each iteration a Nastran analysis to compute the displacements and the strains caused by the pressure distribution, the same approach presented in chapter 3.5 is exploited:

- before the close-loop cycle iterations, for each element of the external surface of the FEM model of the structure, a load case where a single unit pressure acts on that element is created;
- if N_e is the number of elements, then N_e Nastran static analysis are run to extract the displacements and strains distributions, $\mathbf{P_i}$ and $\boldsymbol{\varepsilon_i}$, caused by each of the single unit pressure load cases;
- the displacements and strains, relative to a specific pressure distribution, are computed at each iteration without any additional Nastran analysis. Instead, exploiting the superposition principle, the following expressions are employed:

$$\begin{aligned} \mathbf{P_p} &= \sum_{i=1}^{N_e} p_i \mathbf{P_i}, \\ \boldsymbol{\varepsilon_p} &= \sum_{i=1}^{N_e} p_i \boldsymbol{\varepsilon_i} \end{aligned} \tag{3.36}$$

  where p_i is the value of the pressure distribution in the centroid of the i-th element of the FE model.

3.7 Error definition

At each iteration, as shown in both figure 3.1 and figure 3.2, the controller has to verify if convergence has been reached, meaning that the controller has to check if the iFEM reconstructed shape has converged to the target shape. A proper convergence evaluation is obtained through the implementation of a coherent error definition. In the control architecture developed two error definitions were formulated.

As the actuation loads are computed using the difference between the iFEM reconstructed shape and the target shape, $\mathbf{P_{tgt}} - \mathbf{P_{iFEM}}$, the most natural definition for the error is a relative least-squares difference between the two matrices. Remembering that the shape matrices are always expressed in terms of displacements (see chapter 3.1) the formula for the relative least-squares error is

$$\begin{aligned} err_{iFEM} &= \\ &= \frac{1}{\sqrt{N}} \frac{\sqrt{\sum_{j=1}^{N}(u_{j,tgt} - u_{j,iFEM})^2 + \sum_{j=1}^{N}(v_{j,tgt} - v_{j,iFEM})^2 + \sum_{j=1}^{N}(w_{j,tgt} - w_{j,iFEM})^2}}{\max(\sqrt{\mathbf{u_{tgt}}^2 + \mathbf{v_{tgt}}^2 + \mathbf{w_{tgt}}^2})} \end{aligned} \tag{3.37}$$

where in the numerator there is the square root of the sum of the squared-differences between the target and iFEM nodal displacements and in the denominator there is the square root of the number of nodes of the FE model, N, multiplied by the maximum target shape magnitude. This error definition, although coherent with the model and the control loop architecture implemented, presented a problem for the trailing edge structure: when small transverse target deflections were considered, the definition yielded a high error even if the structure converged to the target shape. This can be seen, for instance, in figure 3.7, which represents with colors on the morphed structure the "u-component" of the least-squares relative error, that is the relative least-squares error computed with the following expression

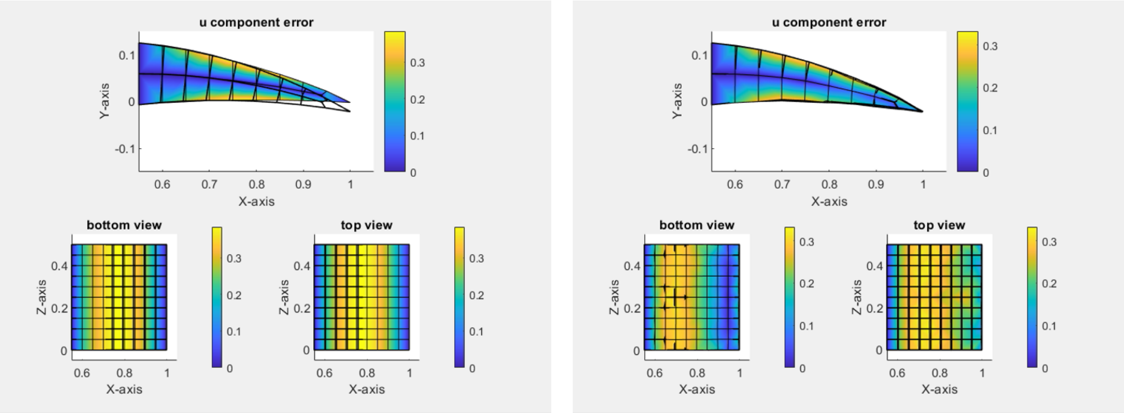
$$err^{u}_{iFEM} = \frac{1}{\sqrt{N}} \frac{\sqrt{\sum_{j=1}^{N}(u_{j,tgt} - u_{j,iFEM})^2}}{\max(\sqrt{\mathbf{u_{tgt}}^2 + \mathbf{v_{tgt}}^2 + \mathbf{w_{tgt}}^2})} \tag{3.38}$$

In the same figure, the target shape is represented with black outlines and a white filling. Looking at the top picture of figure 3.7b, it is evident that in the final configuration, the control scheme has achieved the target shape. However, looking at the error plot at the bottom of the same picture, the error relative to the displacement component along the x-axis, the u displacement component, has roughly the same value as the first initial configuration presented in figure 3.7a. The explanation of the high error, which is produced even at convergence, can be found by looking carefully again at the top picture of figure 3.7b: as it can be seen in figure 3.8, even though the nodes of the iFEM reconstructed shape lay on the top and bottom surface of the target shape, the x-coordinates of the nodes of the two shapes do not perfectly line up, resulting in the high error relative to the normal displacement component. To solve this problem a second error definition was implemented.

The new error is defined using only the transverse distance between the nodes of the skin at each iteration and the nodes on the curves that make up the skin of the final morphed structure. The 'target curves' are computed by spline interpolation, using a certain number of points of the target morphed shape, that is using the MATLAB command

$$\mathbf{y_{tgt}} = spline(\mathbf{x_{ref}}, \mathbf{y_{ref}}, \mathbf{x_{iFEM}}) \tag{3.39}$$

$\mathbf{x_{ref}}$ and $\mathbf{y_{ref}}$, as mentioned, are vectors containing the coordinates of the target shape. The spline command creates an interpolation curve function using these sets of data and



(a) Initial configuration

(b) Final configuration

Figure 3.7: Error plot on the morphed structure throughout the iterations

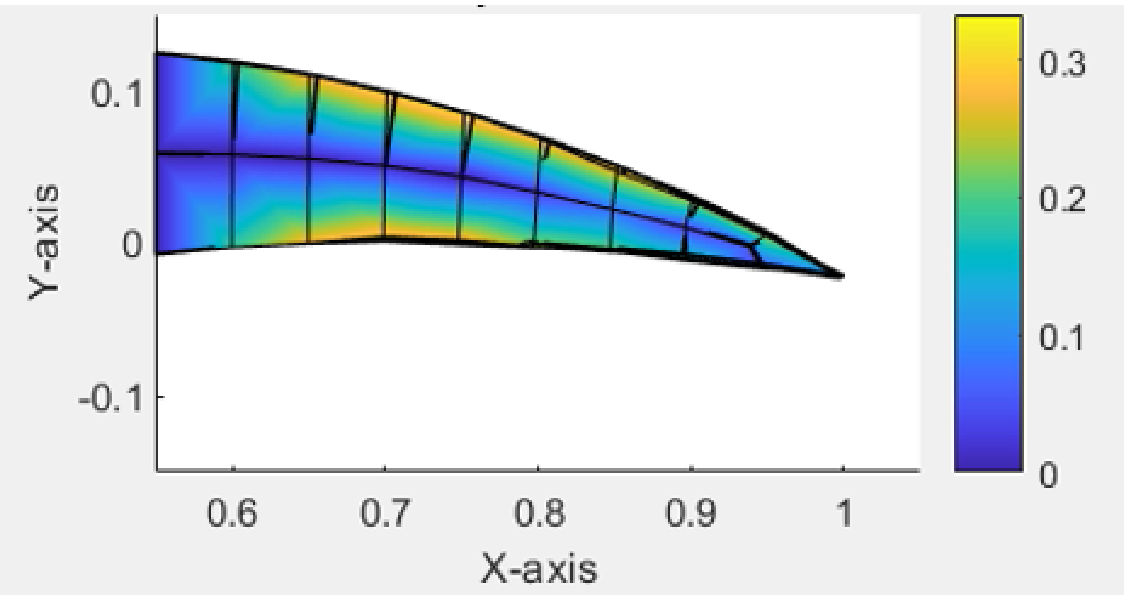


Figure 3.8: Zoom of the top picture of figure 3.7b

evaluates this function in $\mathbf{x_{iFEM}}$, the vector of x-coordinates of the iFEM reconstructed shape. In figure 3.9, the iFEM shape is represented in red while the target shape obtained via spline interpolation is colored with a black outline. Looking at figure 3.9, it is easy to understand why the target shape is obtained by introducing the iFEM coordinates in the spline evaluation: in this new target shape, the x-coordinates of the nodes of the inverse Finite Element shape and the target shape itself are the same. Thus, convergence is defined only by the difference between the y-coordinates of the nodes.

The error is computed following again a least square approach: for each node, the square transverse distance between the iFEM position of the node and the position that the node would occupy on the target curve is calculated. This definition results in the following

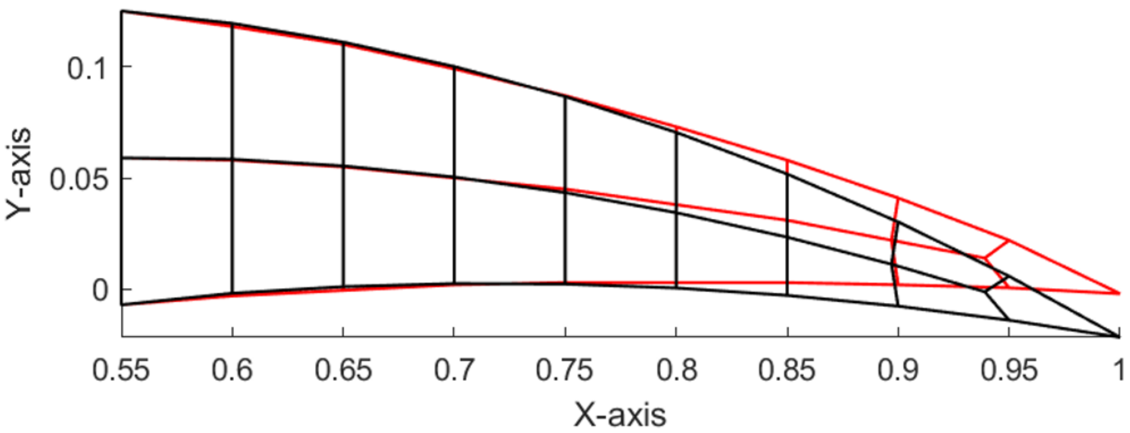


Figure 3.9: New target shape definition

expression:

$$err_{\Delta y} = \sqrt{\frac{1}{N} \frac{\sum_{j=1}^{N}(y_{j,target} - y_{j,iFEM})^2}{\max(\mathbf{y_{target}}^2)}} \tag{3.40}$$

Now that each one of the elements of the closed loop architecture has been explained in detail, the next chapter will focus on its application for the control of a simple morphing plate structure.

Chapter 4

Results: plate structure

The closed-loop control architecture presented in chapter 3 was implemented in a MATLAB environment. A simple plate structure was chosen for the first tests of the closed-loop scheme. Removing the complexity of the morphing structure was instrumental, as it allowed to investigate the following key aspects of the closed-loop control algorithm:

- the correct implementation of the matrix of influence coefficients;
- the effect of Tikhonov regularization;
- the influence of an unknown external load, modeled as a constant pressure distribution on the bottom surface of the plate;

These topics are all covered in this chapter, which contains a collection of the main results obtained in the tests of the closed-loop control of the morphing plate structure. The chapter begins with a brief introduction to the plate model, spanning over the geometry and the materials used, the direct and inverse finite element models of the structure, and the adopted actuation scheme.

4.1 Plate structure model

The model chosen for the structure is that of a cantilevered, square, thin, aluminum plate. The 3D model data are summarised in table 4.1 The FE model of the plate is presented in

Plate structure model

In-plane dimensions [mm×mm]	100×100
Thickness [mm]	2
Boundary conditions	Cantilevered
Material	Aluminum
Young modulus [MPa]	730000
Poisson coefficient	0.33

Table 4.1: Plate structure model

figure 4.1. For the direct model, the structure is modeled with only shell elements (Quad4), and a 20×20 mesh is adopted, resulting in a model with 441 nodes and 400 elements. For the inverse model, the same amount of inverse four-node elements (iQS4) is used. As a simplifying hypothesis, it is assumed that all the elements are instrumented with strain rosettes on both the top and bottom surfaces, so that back-to-back triaxial strains on all the elements are known.

In figure 4.1, the actuation scheme and the boundary conditions can also be observed: one side of the plate is clamped to obtain the cantilevered structure. As for the actuation scheme, each free node of the finite element model is loaded with a concentrated transverse force. The high number of load points increases the dimension of $\mathbf{A}$, the matrix of influence coefficients. As explained in chapter 3.4 the matrix of influence coefficients is a $3N \times 6N_L$ matrix, where N is the number of nodes, N_L is the number of actuation points and $6N_L$ is the total number of loads, assuming that a concentrated load or moment can be produced in any direction of the reference coordinate system. However, on the plate structure, only transverse concentrate forces are applied. Therefore, the matrix of influence coefficients in this case scenario has only a dimension of $3N \times N_L$. Nevertheless, since in this example $N = 441$ and $N_L = 420$, the matrix of influence coefficients is still a 1323×420 matrix. The dimension of the $\mathbf{A}$ matrix results in a high condition number. In particular, since $\mathbf{A}$ characterises the least-squares problem that is used to compute the loads, a high condition number results in a very badly conditioned problem, afflicted by instabilities. Therefore, although unrealistic, the exaggerated number of load points of the actuation scheme serves the purpose of testing the effectiveness of the Tikhonov regularization in maintaining a stable solution.

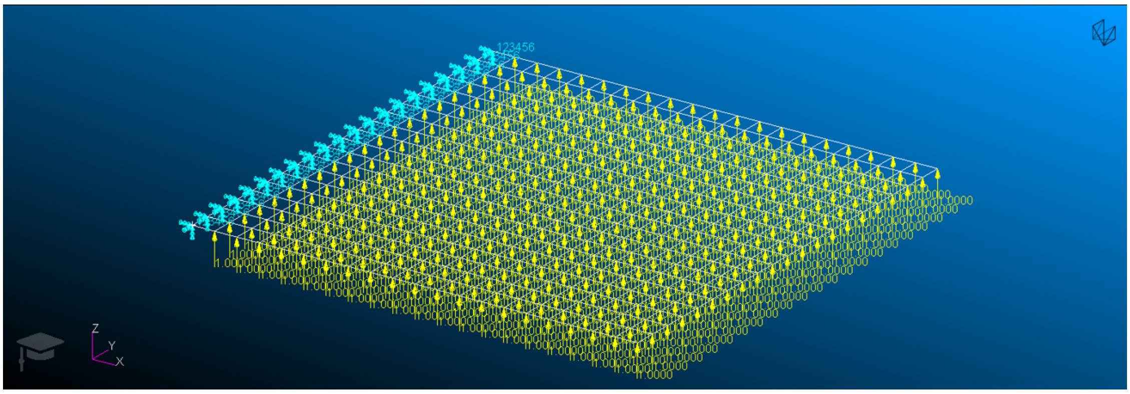


Figure 4.1: Plate FE model

In order to test the closed-loop control of the plate when subject to an external unknown disturbance, a constant unit pressure distribution on the bottom surface of the plate, shown in figure 4.2, is implemented in the FE model. Thus, the displacements and strains fields, respectively $\mathbf{P_p}$ and $\boldsymbol{\varepsilon_p}$, caused by this external action, can be extracted with a Nastran analysis before the closed-loop iterations. Referring to the closed-control loop flowchart presented in figure 3.2 of chapter 3.2, this procedure allows to skip the XFOIL run, as the effects of the aerodynamic load on the deformations, $\mathbf{P}$, and the strains, $\boldsymbol{\varepsilon}$, of the structure

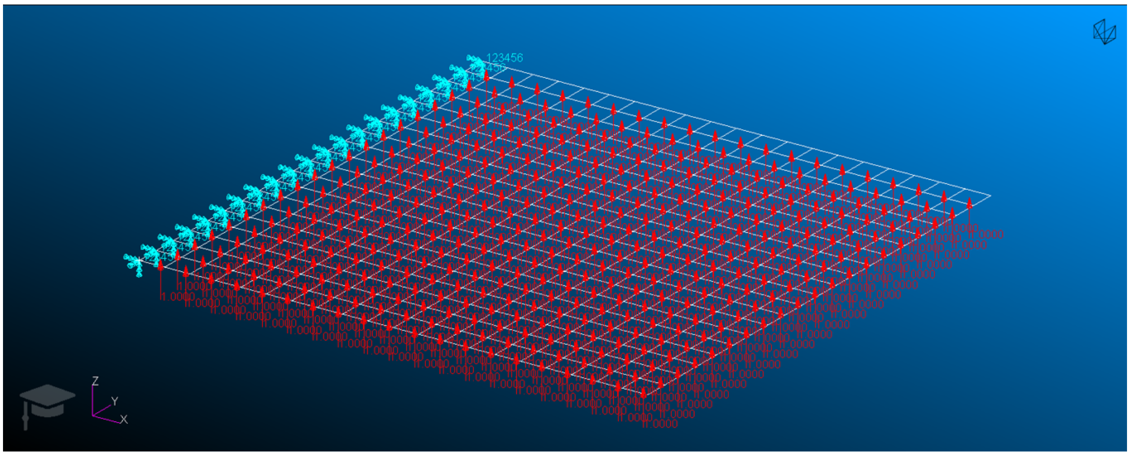


Figure 4.2: External disturbance model on the plate structure

are simply computed with the following expressions:

$$\begin{aligned} \mathbf{P} &= \mathbf{P_{FEM}} + m\mathbf{P_p} \\ \varepsilon &= \boldsymbol{\varepsilon_{FEM}} + m\boldsymbol{\varepsilon_p} \end{aligned} \tag{4.1}$$

where the parameter m is a constant used to modulate the amplitude of the effects of the pressure distribution. The approach presented with Eqs. 4.1 is clearly an extreme simplification of the procedure explained in chapter 3.6, but it is consistent with the other assumptions made, as the objective is not to produce a realistic example but rather testing the intricacies of the closed-loop architecture implemented.

4.2 Results

Before testing the close-loop control architecture, a procedure was developed to verify the correct implementation of the matrix of influence coefficients. The procedure is divided into two steps: the first one is the introduction of a new error, which is defined as the least-squares relative error between the shape obtained using the FE model, $\mathbf{P_{FEM}}$, and the target shape, $\mathbf{P_{tgt}}$. This definition is identical to the one presented in chapter 3.7, in Eqn. 3.37, with the exception of the iFEM displacements, which are replaced by the FEM displacement. The new error has the following expression:

$$\begin{aligned} &err_{FEM} = \\ &= \frac{1}{\sqrt{N}} \frac{\sqrt{\sum_{j=1}^{N}(u_{j,tgt} - u_{j,FEM})^2 + \sum_{j=1}^{N}(v_{j,tgt} - v_{j,FEM})^2 + \sum_{j=1}^{N}(w_{j,tgt} - w_{j,FEM})^2}}{\max(\sqrt{\mathbf{u_{tgt}}^2 + \mathbf{v_{tgt}}^2 + \mathbf{w_{tgt}}^2})} \end{aligned} \tag{4.2}$$

where the error is referred to as err_{FEM}, to distinguish it from err_{iFEM} which refers to the error presented in Eqn. 3.37. The second step to verify the correct implementation of $\mathbf{A}$ consists in running a closed-loop analysis, in which the target shape is the one obtained by setting the value of each load of the actuation scheme to one. In other words, the target shape is obtained by summing all the columns of the matrix of influence coefficients, $\mathbf{A}$:

$$\mathbf{P_{tgt}} = \sum_{i=1}^{N_L} \mathbf{P_i} \qquad \mathit{with} \quad \mathbf{A} = [\{\mathbf{P_1}\}, \cdots, \{\mathbf{P_i}\}, \cdots, \{\mathbf{P_{N_L}}\}] \tag{4.3}$$

Furthermore, the closed-loop cycles are run setting the Tikhonov regulation parameter, λ, to zero, as well as the pressure modulus coefficient, m. The regularization parameter was introduced in chapter 3.4.2 with Eqn. 3.27. As explained in that chapter, λ controls the extent of the regularization: if λ is set to zero, no regularization is applied, and the solution corresponds to the real solution of the least-squares problem. Similarly, if $m = 0$, as shown in Eqs. 4.1, no external action is considered on the structure and the deformations and the strains coincide with the FEM ones. In this scenario, the least-squares problem used to compute the actuation loads has an exact solution

$$\mathbf{L} = A^{\dagger} \sum_{i=1}^{N_L} \mathbf{P_i} = \{1 \quad \cdots \quad 1\}^T \tag{4.4}$$

where $\mathbf{P_i}$, the columns of the matrix of influence coefficients, are the displacements caused separately by each of the unit loads of the load vector. Introducing the unit load vector solution into Eqn. 3.34 of chapter 3.5, the superposition principle yields a FEM shape which is again the sum of the displacements caused by the single unit loads:

$$\mathbf{P_{FEM}} = \sum_{i=1}^{N_L} \mathbf{P_i} \tag{4.5}$$

Therefore, if the matrix of influence coefficients is implemented correctly, the result of this procedure is that the FEM error defined in Eqn. 4.2 is null, or close to machine precision, as $\mathbf{P_{FEM}} \equiv \mathbf{P_{tgt}}$.

In figure 4.3, the results of the procedure used to check the correct implementation of the influence matrix related to the the actuation scheme of figure 4.1, is shown. Looking at the data shown in the picture, it is evident that the FEM error is approximately zero. It is also worth noting that the iFEM error is very low (0.3% circa), meaning that the iFEM shape feedback is capable of accurately reconstructing the shape in this scenario.

To further analyze the performance of the iFEM feedback, the closed-loop architecture was tested using the target shape presented in chapter 3.3, in figure 3.3, that is the target shape where the transverse deflection follows the equation of a parabola, x^2. The first test was conducted by setting the regularization parameter and the module of the pressure distribution again to zero. The graph of figure 4.4a shows the evolution of the error with every iteration: it is evident that after the first iteration, the error starts diverging. This is caused, as explained in chapter 3.4.2, by the instability issues of the least-squares inverse problem: the small errors associated with the iFEM reconstruction are amplified with every iteration, leading to the error blow-up shown in the picture.

To solve the instability issues, the value of the Tikhonov regularization parameter, λ, was properly adjusted. However, as Hansen and Hanke stated in 1993: "No black-box procedures for choosing the regularization parameter λ are available, and most likely will never exist" [29]. Therefore, the regularization parameter was determined with a

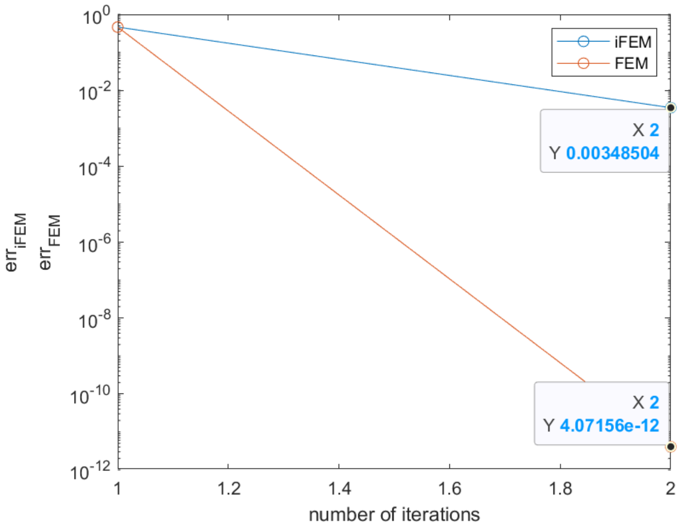
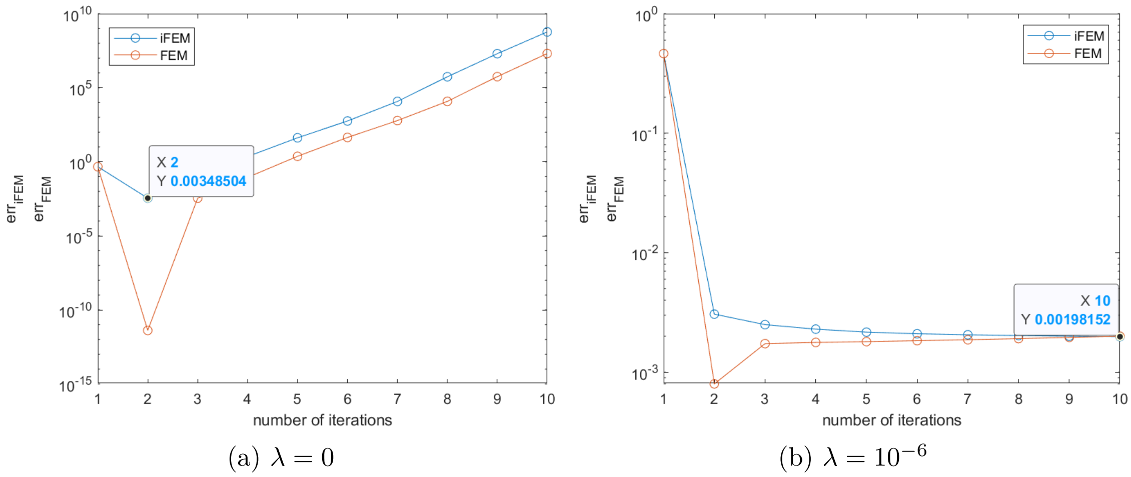


Figure 4.3: Test on the correct implementation of the influence matrix

heuristic approach. In particular, through a series of iterative attempts, the value of the regularization parameter was set to $\lambda = 10^{-6}$. The regularized problem and the closed-loop architecture were then tested one more time, still with no external disturbance acting on the structure. The error, plotted against the number of iterations, is shown in figure 4.4b. Comparing figure 4.4b with figure 4.4a, it is evident how the Tikhonov regularization stabilizes the solution after the second iteration, allowing to further reduce the error.

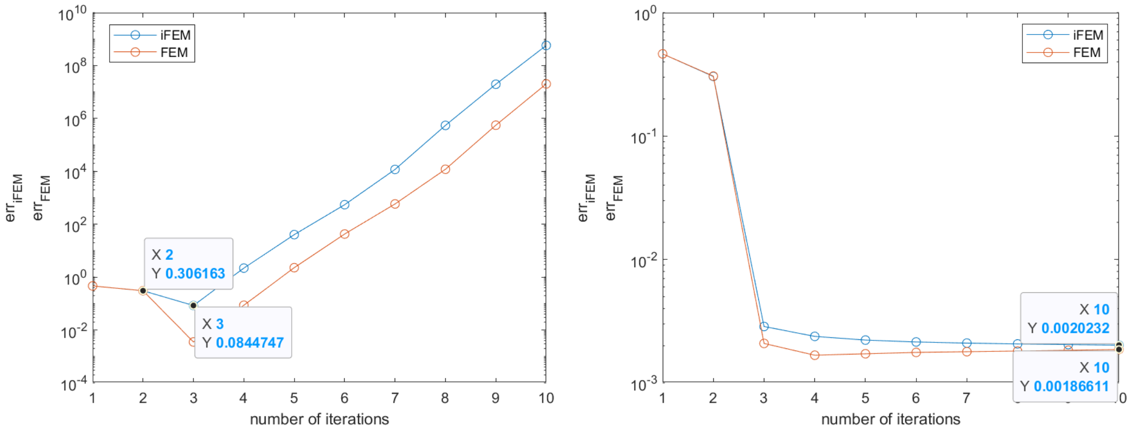


(a) $\lambda = 0$

(b) $\lambda = 10^{-6}$

Figure 4.4: FEM and iFEM error along the iterations - $m = 0$

One could argue that, in this example, the error reduction is so small (0.1% circa) that it would have been much easier to simply stop the iterations after the second one. This solution would have allowed to reach convergence with high accuracy without implementing Tikhonov regularization. This statement, although correct, holds true only as long as the closed-loop architecture is used as an open-loop strategy: in the case study of figure 4.4a, after the assessment of the shape of the structure in the first iteration, the second one is used to actuate the structure, which immediately morphs in the target shape. However, if more iterations were to be needed, for instance in the case where the the closed-loop architecture had to be used to control the shape under an unknown disturbance, the absence of a regularization method would lead to instabilities and convergence wouldn't be reached. This case scenario is represented in figure 4.5a: the external disturbance is, in this example, non-null as $m = 25$. This leads to an error of 30% circa in the second iteration (much higher than the error of 0.3% of the previous cases). The closed-loop controller attempts in the subsequent iterations to correct the error. However, the small disturbances, without any regularization, slowly start to affect the output, to a point where the error stops decreasing and instead starts to continuously rise. This behaviour is also captured in figure 4.6, where the iFEM reconstructed shape, in red, is compared to the target shape, plotted in black, at different morphing stages: in the first iteration (figure 4.6a) the iFEM feedback assesses that the structure is in its initial unmoprhed configuration; in the second iteration (figure 4.6b), the controller attempts to achieve the target shape, but the external load causes a shift of the structure. Therefore, another iteration is needed. However, at the third iteration (figure 4.6c), the instability of the inverse least-squares problem starts crippling the iFEM feedback, initiating the divergence of the error.

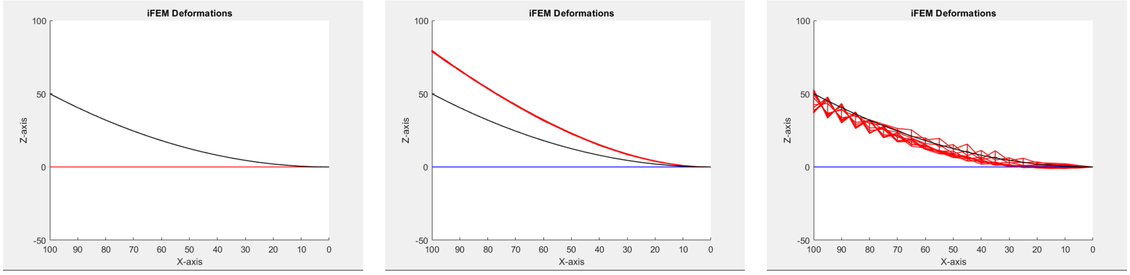


(a) $\lambda = 0$

(b) $\lambda = 10^{-6}$

Figure 4.5: FEM and iFEM error along the iterations - $m = 25$

Hence, regularization needed and it is applied in the case of $m = 25$. The results obtained using again a value of the regularization parameter $\lambda = 10^{-6}$ are presented in figure 4.5b. As clearly shown in the picture, the effect of regularization causes a drastic decrease in the error in the third iteration, as the instabilities that characterised the previous solution are drastically damped. This allows the closed loop controller to assess the effect of the unknown external load and correct it so that in the final iteration the error is once again in the order of 0.2%. This behaviour is also confirmed by the images of figure 4.7: comparing, in particular, figure 4.7c with figure 4.6c, it is evident how the regularization method removes the oscillating pattern in the iFEM reconstructed shape of figure 4.6c, producing, instead, a smooth surface and allowing the convergence to the target shape in the third iteration.

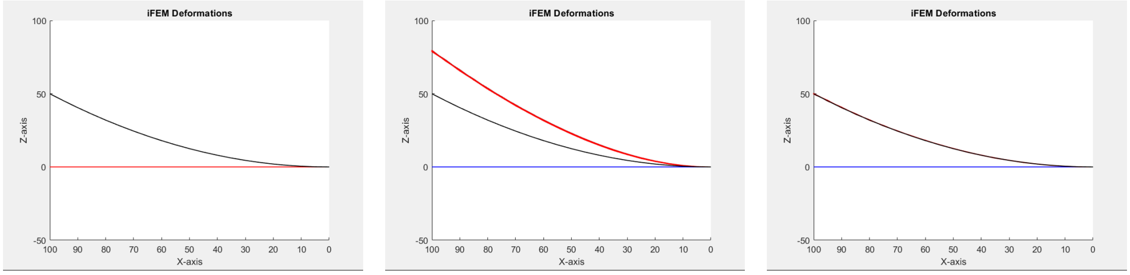


(a) First iteration

(b) Second iteration

(c) Third iteration

Figure 4.6: iFEM reconstructed shape throughout the closed-loop iterations - $m = 25$, $\lambda = 0$



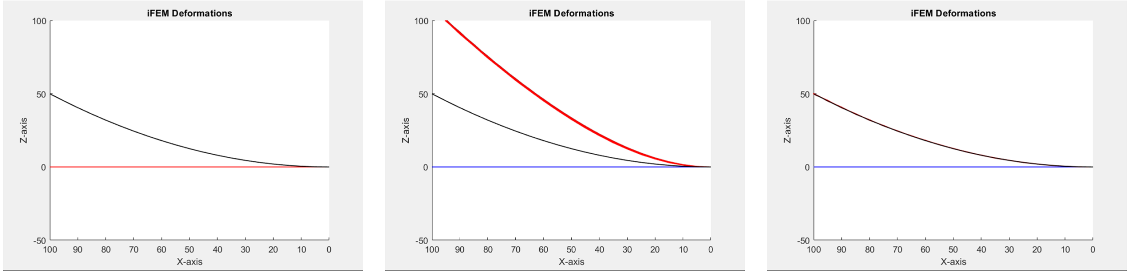
(a) First iteration

(b) Second iteration

(c) Third iteration

Figure 4.7: iFEM reconstructed shape throughout the closed-loop iterations - $m = 25$, $\lambda = 10^{-6}$

To test the robustness of the code, another test was run setting the pressure module $m = 50$. The results of the closed-loop control process are shown in figure 4.8. Looking at the pictures of the structure while it is morphing, the same pattern of the previous case study occurs: in the second iteration the effect of the external disturbance causes a noticeable deviation from the target shape (figure 4.8b). Nevertheless, in the third iteration, the controller is able to adjust the actuation loads to accommodate for the desired target



(a) First iteration

(b) Second iteration

(c) Third iteration

Figure 4.8: iFEM reconstructed shape throughout the closed-loop iterations - $m = 50$, $\lambda = 10^{-6}$

shape (figure 4.8c). This behaviour is reflected in the error plot, shown in figure 4.9. Because of the external load, in the second iteration the error instead of reducing itself, increases. Throughout the following iterations, however, the error converges to the same value of 0.2% of the previous cases.

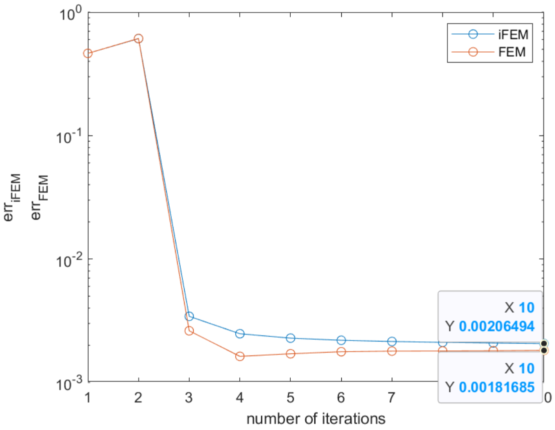


Figure 4.9: FEM and iFEM error along the iterations - $m = 50$, $\lambda = 10^{-6}$

The results collected in this chapter are summarised in table 4.2. For the simple plate structure, the inverse Finite Element method proved its capability of accurately reconstructing the shape along the iterations, and thus was successfully implemented in the closed-loop architecture. The controller, exploiting the iFEM feedback to assess the shape, and the regularized least-squares problem to compute the loads, achieved the arbitrary target shape with high precision even under the disturbance of an external unknown load.

Case study	Results
$\lambda = 0$ $m = 0$	The error converges to a value of 0.3% after the second iteration. If multiple iterations are requested, divergence occurs
$\lambda = 10^{-6}$ $m = 0$	The error converges to a value of 0.2%, even after multiple iterations
$\lambda = 0$ $m = 25$	The error diverges because multiple iterations are needed
$\lambda = 10^{-6}$ $m = 25$	The error converges to a value of 0.2%
$\lambda = 10^{-6}$ $m = 50$	The error converges to a value of 0.2%

Table 4.2: Closed-loop control test on the morphing plate structure: summary of the results

Chapter 5

Results: trailing edge structure

The results achieved in the previous chapter, although promising, were obtained employing the closed-loop control architecture with a structure, the morphing plate, that significantly differed from the morphing designs presented in chapter 1.2. To test the iFEM feedback controller in a state-of-the-art framework, a wing concept with a morphing trailing edge was developed. Furthermore, to simulate the effect of external unknown disturbances on the morphing structure in a realistic case scenario, such as during flight, a procedure to extract the pressure distribution on the wing while it is morphing was implemented.

This chapter contains the main results acquired testing the iFEM feedback control strategy to control and monitor the shape of the morphing trailing edge, both with or without the influence of the pressure disturbance.

5.1 Trailing edge model

Before presenting the results, the trailing edge model is introduced. The covered topics are the following:

- the morphing architecture;
- the materials;
- the direct and inverse FE models;
- the actuation schemes.

5.1.1 3D Model

As explained in chapter 3.3, camber morphing and, in particular, trailing edge morphing is the most recurring solution in the literature of morphing wing designs, since it provides excellent aerodynamic and control performance. The morphing trailing edge wing tested in this chapter was inspired by a particular promising design, the one developed by G. Molinari and his team [15], which has been already presented in detail in chapter 1.2. In particular, the model developed is a simplification of the conceptual morphing wing of [15].

In figure 5.1, the 3D model of the wing structure is shown. The wing structure is divided

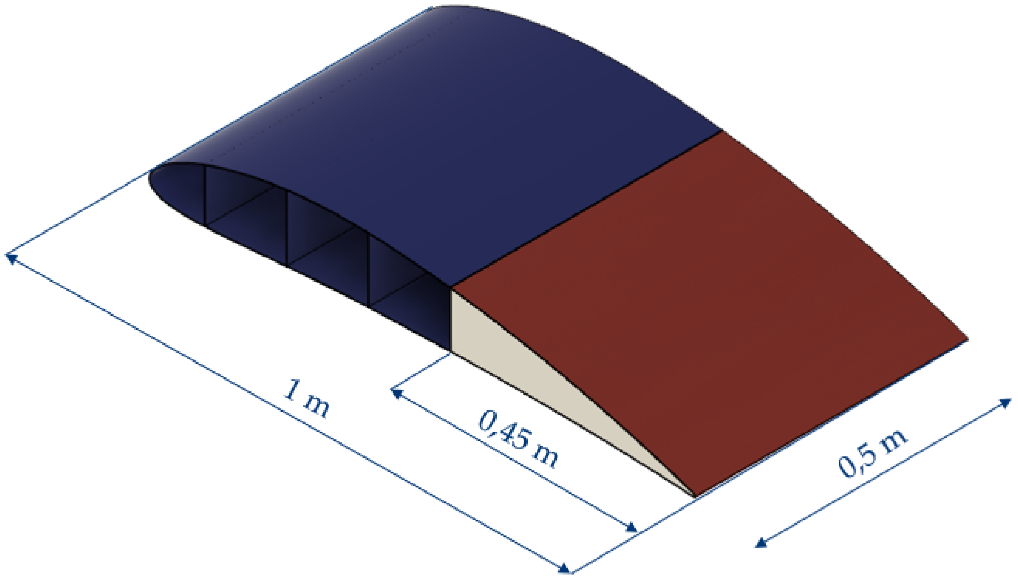


Figure 5.1: Trailing edge 3D model

into two parts: the first one is the leading edge (represented in blue), made of a very stiff, aluminum shell structure. This structure is considered infinitely rigid with respect to the other portion of the wing, the morphing trailing edge, which has a skin made of epoxy resin (represented in red) reinforced by a soft foam core. The materials used for the morphing portion of the wing are chosen so that the structure is compliant enough to exhibit morphing capabilities but, at the same time, has the robustness to withstand the applied loads. In particular, the foam core is used to counteract the pressure distribution on the flexible skin of the trailing edge, which would otherwise deform excessively under this transverse action. The material properties of both the stiff leading edge and the morphing trailing edge are summarised in table 5.1 Both the epoxy skin and the foam core, as clearly shown by the data in the table, are considered as isotropic materials, in order to facilitate the FEM implementation of the 3D model.

		Material	**E [MPa]**	ν
Leading edge	Shell Structure	Aluminum	73000	0.33
Trailing edge	Skin	Epoxy resin	3.2	0.35
	Core	Roachell WF 51	75	0.44

Table 5.1: Materials of the morphing trailing edge model

5.1.2 FE Model

The FE model of the structure is presented in figure 5.2. Only the trailing edge structure, whose dimensions are displayed in figure 5.1, is modeled in the FE environment. The leading edge, instead, is represented by the clamped rear end of the trailing edge. This choice is coherent with the assumption that the leading edge, made of a stiff, aluminum, shell structure, is infinitely rigid in comparison with the compliant, morphing trailing edge.

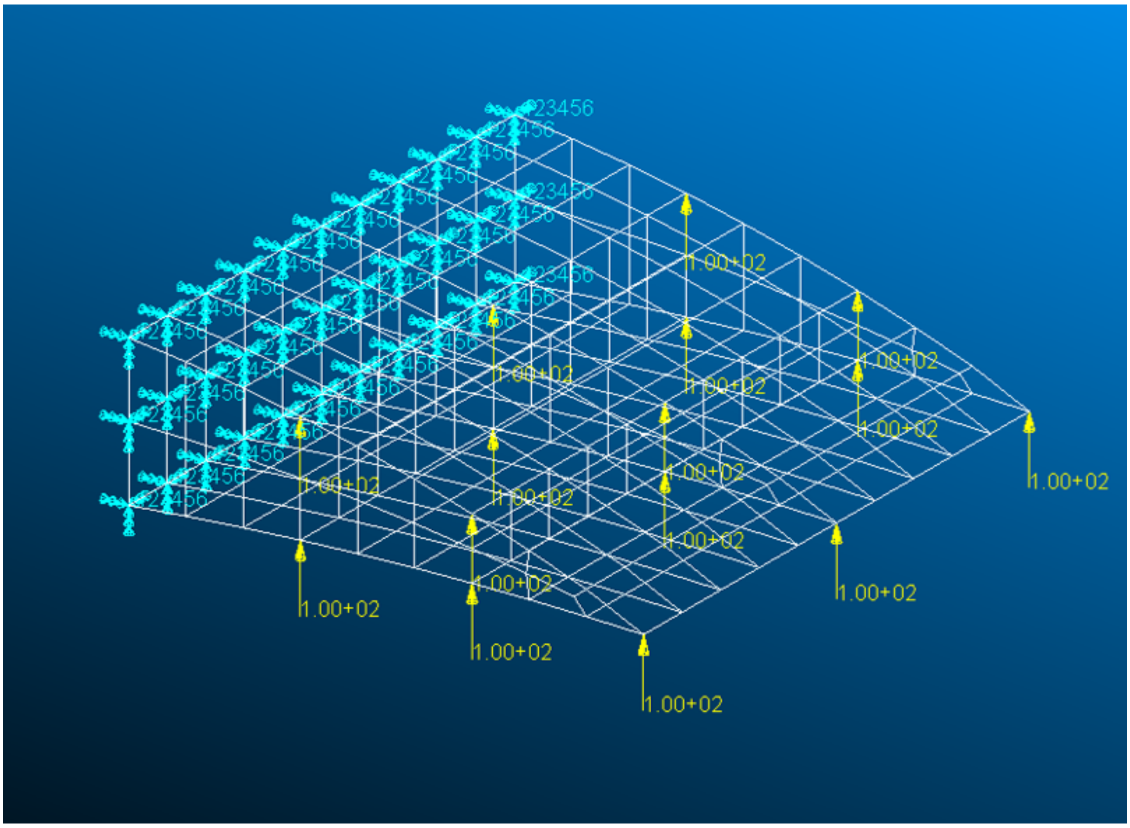


Figure 5.2: Trailing edge FE model

The skin of the trailing edge is modeled with 2D shell elements (Quad4), while, for the core, solid elements (Hex8) are used. The total number of nodes is 260, resulting in a total of 291 shell elements and 170 solid elements. On the other hand, the inverse finite element model consists solely of 2D inverse four-node elements (iQS4). The reason is that the iFEM feedback is used to monitor only the external shape of the structure, which can be obtained by simply reconstructing the displacements of the skin. To accomplish this task, it is assumed that every element of the skin is instrumented with a strain rosette on both its top and bottom surfaces. In reality, it would be very difficult to have strain sensors on the inner interface between the skin and the foam core (integrated FBG sensors would have to be used). Nevertheless, to validate the procedure, it is hypothesized that, at a numerical model level, this measurement is possible, so that the back-to-back tri-axial strains on all the elements are known. This allows for a full-field (i.e. at every material point) reconstruction of the displacements.

In figure 5.2, one of the two actuation schemes tested is also shown: it consists of a sparse set of transverse concentrated forces both in the chord and span-wise direction. This actuation scheme represents a simplification of the action exerted by the piezoelectric or shape memory alloy actuators frequently used in the modern morphing structures.

Another attempt at mimicking the state-of-the-art of morphing actuators is represented in figure 5.3, where the second actuation scheme employed for the trailing edge model is shown. It consists of a distributed pressure acting on both the top and bottom skin of the trailing edge. The pressure distribution is modeled by applying a pressure load on every element of the skin. This second actuation scheme was chosen as it corresponds to the model of the pressure distribution acting on the trailing edge. In particular, the procedure to extract, for each single element pressure, the displacement field to build the influence matrix (refer to chapter 3.4.1) is exactly the same procedure that is used to compute the effects, in terms of strains and displacements, of the aerodynamic pressure on the trailing edge, which was explained in detail in chapter 3.6.

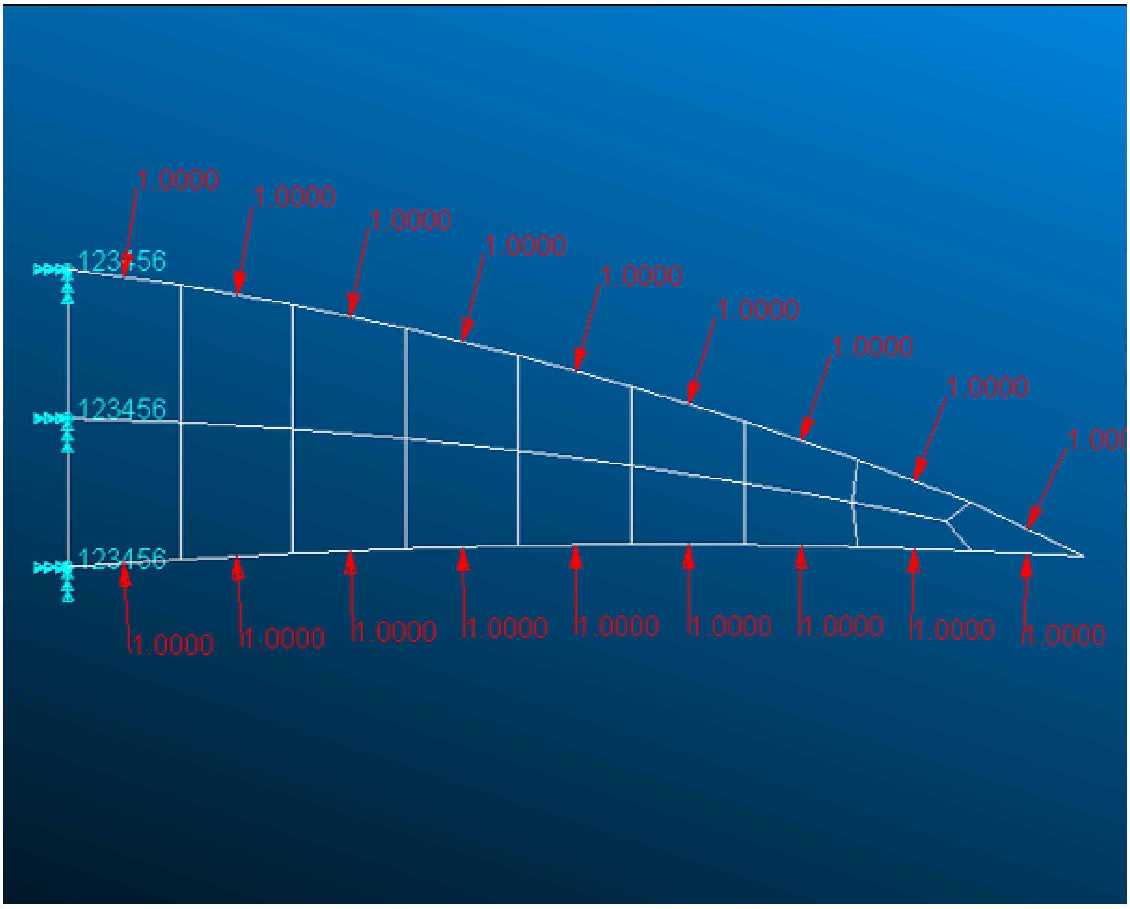


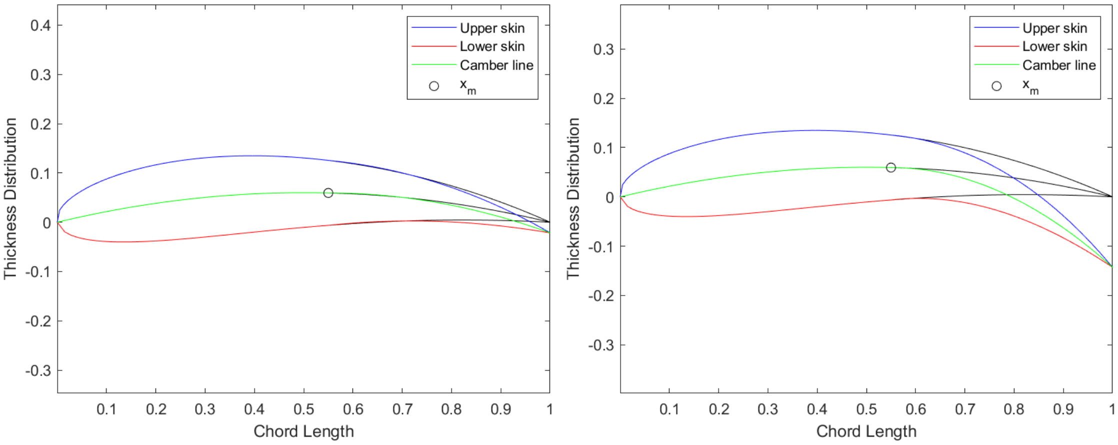
Figure 5.3: Pressure distribution on the trailing edge FE model

5.2 Results

As explained in chapter 3.3, the target shape for the morphing trailing edge was obtained by modifying only the camber line of the reference NACA profile (the NACA 6516), since the equations that define such profiles produce a congruent morphing of the skin of the structure. As morphing is concentrated in the trailing edge, only its camber line equation was altered. In particular, the camber line equation was modified so that it morphed into a parabolic shape. This led to the following expression of the transverse coordinate of the camber line:

$$y_c(x) = y_{c0} - k \cdot (x - x_m)^2 \tag{5.1}$$

where the term y_{c0} is used to guarantee the continuity of the equation with the unmorphed portion of the wing and $x_m = 0.55$ is the coordinate along the chord of the rear end of the morphing trailing edge. The term $-k$ represents an arbitrary constant that is used to control the extent of the downward transverse deflection of the profile. In the case studies presented in this chapter, the value of the constant was set to $k = 0.4$ and $k = 1$. The target shapes obtained are shown in figure 5.4.

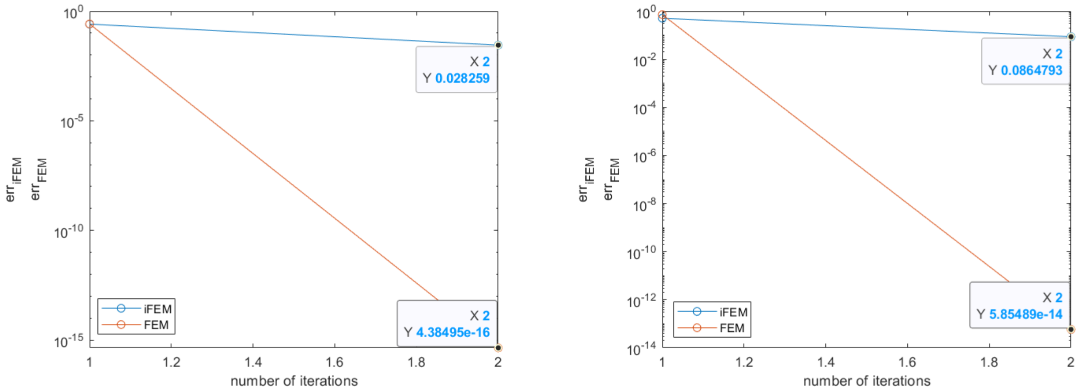


(a) $k = 0.4$

(b) $k = 1$

Figure 5.4: Target shapes

Before testing the control loop architecture, as for the plate structure, the correct implementation of the influence matrices relative to both actuation schemes was verified. The results are shown in figure 5.5: the least-squares error between the FEM shape and the target shape, which was defined in Eqn. 4.2 of chapter 4.2, is close to machine precision (i.e. close to zero) for the two actuation schemes. It is worth noting that even the iFEM error, that is the error between the target and iFEM reconstructed shape, is very small (less than 10% in both cases), highlighting the quality of the iFEM feedback. The FEM and iFEM errors presented in figure 5.5, are affected by a problem when used as convergence criteria for the closed loop iterations: even though the structure converges to the target shape, the FEM and iFEM errors remain very high. This issue was discussed previously in chapter 3.7, where the iFEM error was plotted with colors on the structure at different morphing stages (figure 3.7). In this chapter, to avoid repetition, the iFEM and FEM error

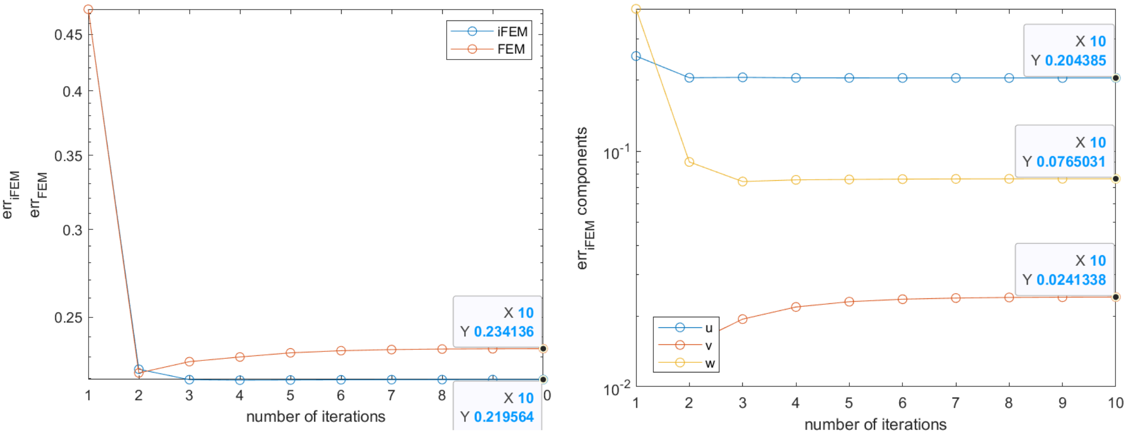


(a) Sparse transverse forces

(b) Pressure distribution

Figure 5.5: Test of the correct implementation of the influence matrices relative to the two actuation schemes

plots against the number of iterations are shown in figure 5.6. The graphs presented refer to the case where $k = 0.4$, no external disturbance is acting on the structure (remembering equation 3.35, this means $V = 0\ m/s$) and the actuation scheme is the one using sparse transverse forces. As it will be later shown in figure 5.8, with this set of parameters the



(a) FEM and iFEM error

(b) iFEM error components

Figure 5.6: FEM and iFEM error plot along the iterations - $k = 0.4$, $V = 0\ m/s$, sparse transverse forces actuation scheme

controller is more than capable of converging to the target shape. However, as clearly shown by the data highlighted in figure 5.6a, the iFEM and FEM errors have a value greater than 21% in the last iteration (the FEM error is higher because the closed-loop strategy is based on the iFEM feedback, thus it is to be expected that the associated iFEM error is the lowest one). The high error value, as shown in figure 5.6b, is related to the u displacement component, that is the displacement component along the x-axis, as the error related to this component has a value of 20% circa.

The solution to this problem was presented in chapter 3.7. To sum it up, a new error definition was introduced,$err_{\Delta y}$, measuring the relative least-squares difference between the transverse position of the nodes of the target and the iFEM reconstructed shape (Eqn. 3.40). The evolution of this new error with every iteration is shown in figure 5.7. Looking

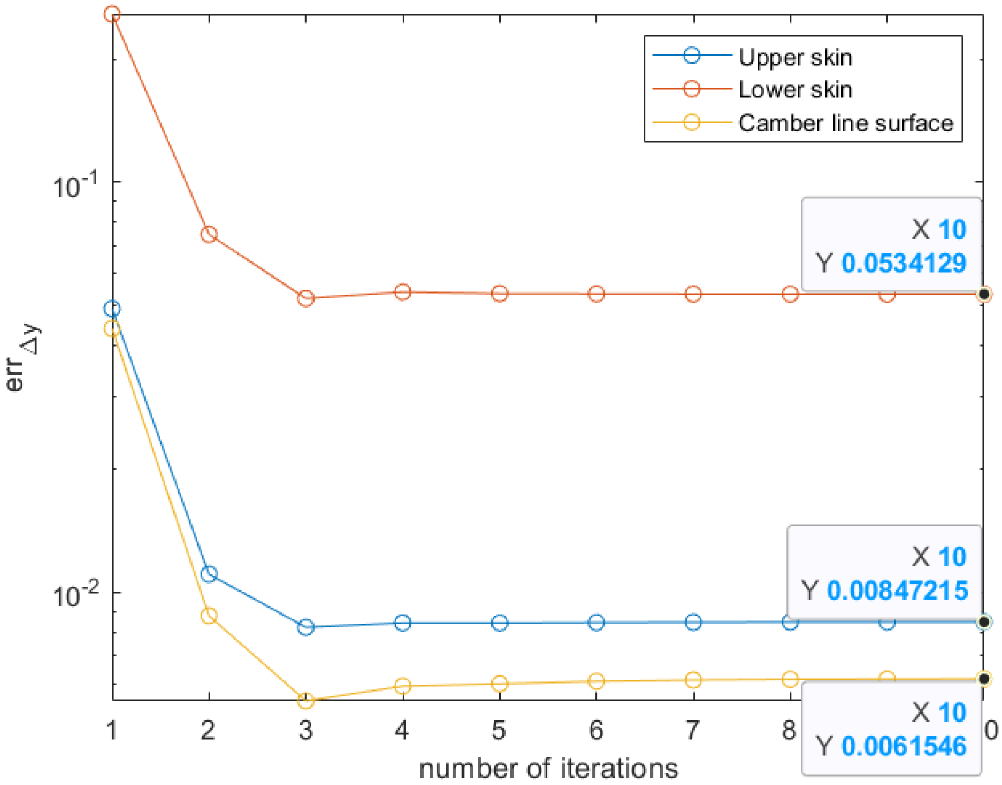


Figure 5.7: Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, $V = 0\ m/s$, sparse transverse forces actuation scheme

at the data in the figure, the new relative error produces in the last iteration a value of 5% for the lower skin and a value of 0.8% for the upper skin. To confirm the effectiveness of this new error definition, the 3D iFEM reconstructed shape plots that will be later presented, are colored using the nodal value of the squared difference:

$$\Delta y_j^2 = (y_{j,target} - y_{j,iFEM})^2 \tag{5.2}$$

that is the squared difference between the target transverse position of the j-th node and its transverse position measured by the iFEM feedback. The colored plots are presented in figure 5.8. Similarly to the plate structure (figure 4.4b), when no external disturbance acts on the structure, in the second iteration the controller is able to achieve, with high accuracy, the target shape, which is represented with black outlines in the picture. Coherently, the error values all over the structure (Δy_j^2) reduce themselves significantly, confirming the efficacy of the new error definition. From now on, therefore, $err_{\Delta y}$ and its graphs, both against the number of iterations and as color plots on the pictures of the morphing structure, will be used to represent the behaviour of the controller along the iterations.

Up until this point, the regularization of the least-squares problem used to compute the loads was not mentioned. This is because, for the transverse forces actuation scheme, no regularization is needed: the number of sparse loads is $N_L = 15$, which results in a small influence matrix compared to the one used for the plate structure, where $N_L = 420$. However, if the number of loads increases, as in the case of the distributed pressure actuation scheme, where $N_L = 180$, regularization becomes once again necessary to maintain a stable behaviour of the controller along the iterations. In order to demonstrate this assertion, the case of $k = 0.4$ and $V = 0\ m/s$ is considered once again, but now the distributed pressure actuation scheme is employed. The results, in terms of the transverse least-squares relative error value at each iteration, are presented in figure 5.9. In the labels of both figure 5.9a and figure 5.9b the value of the regularization parameter is shown: in the case of figure 5.9a, where $\lambda = 0$, no regularization is applied and as a consequence, after a fast convergence to the target solution, the error starts to continuously increase with every subsequent iteration. To maintain a stable behaviour, as the one of figure 5.9b, a value of $\lambda = 10^{-16}$ is sufficient. This value is much lower than the one that was needed for the plate structure ($\lambda = 10^{-6}$) but this is not concerning: the condition number of the $\mathbf{A}$ matrix of the plate structure, a 1323×420 matrix (please refer to chapter 4.1 for the procedure that led to this result), is much higher than the condition number of the matrix of influence coefficients related to the distributed pressure actuation scheme. In the latter case, in fact, $\mathbf{A}$ is a $3N \times N_L$ matrix, where N is the number of nodes. Since $N = 260$, this results in a 780×180 matrix, much smaller than the one used for the plate structure. The unstable behaviour of figure 5.9a, is confirmed by the pictures of the 3D iFEM reconstructed shape, collected in figure 5.10, throughout the iterations. In particular, the sequence of figure 5.10b and figure 5.10c, shows how the instabilities of the least-squares problem create an oscillating pattern on the displacements of the structure, that is amplified with every iteration. On the other hand, the stabilizing effect of the regularization parameter is evident in the pictures of figure 5.11, where at the same iterations shown in figure 5.10, the structure appears now with a smooth surface.

Observing the error plot of figure 5.9a, as well as the sequence of pictures of figure 5.10, it is clear that if the closed loop iterations were stopped at the third one, the target shape would have been achieved with high accuracy without needing any regularization. However, as it was explained and demonstrated for the plate structure in chapter 4.2, regularization is necessary if more iterations are needed: in the case of an external disturbance, for instance, the closed loop controller needs multiple iterations to converge to the target shape. If $\lambda = 0$ in that case scenario, the instabilities prevent the controller from ever achieving its objective.

For the target shape obtained with $k = 0.4$, the iFEM feedback-based controller was able to converge to the solution using both actuation schemes. To prove the potential of the controller in managing multiple and different target shapes, the closed-loop architecture was also tested with the objective of morphing the structure to the target shape when $k = 1$, that is when high transverse deflections are needed. In the case of again no external pressure distribution (i.e. $V = 0\ m/s$), the two actuation schemes were tested. The comparative results between the load cases are shown in figure 5.12. In both examples, the controller is able to converge to the target shape with a low error of approximately 1%. This behaviour can be also observed in the images of figure 5.13 and figure 5.14. In particular, looking carefully at the bottom view of figure 5.13c, it can be noticed that the transverse forces actuation scheme is not able to fully stretch the morphing structure along the x-axis to achieve the target shape. This same stretching issue can also be observed in the third iteration of figure 5.14. The pressure distribution, however, can exert a greater normal action (that is in the x-axis direction) than the transverse forces. Hence, by letting the controller run for multiple iterations, in the last one, shown in figure 5.14d, the tip of the morphing trailing edge matches almost perfectly the tip of the target structure.

The results gathered in this chapter so far were all collected under the hypothesis of $V = 0\ m/s$, which as previously mentioned, implies that the pressure distribution on the structure is null. On the other hand, if the wing was traveling with a non-null speed relative to the airflow, the aerodynamic pressure would be different from zero and its distribution would deform the structure, altering its shape and thus requiring the controller to modify its action to obtain the target shape. The last section of this chapter is dedicated to the investigation of the effect of such external action, considered as unknown, on the iFEM feedback-based controller.

To test the effect of the pressure distribution, only the target shape where $k = 0.4$ was considered. The case $k = 1$, besides producing an unrealistically high deflection, caused also the stall of the wing, impeding the retrieval of the pressure distribution along the iterations. Furthermore, for the sake of brevity, only the sparse transverse forces actuation scheme was used since, as shown in the previous example, the closed-loop controller produces very similar results for the two actuation schemes.

The results for $V = 100\ m/s$ are presented in figure 5.15a, where the transverse least-squares relative error, $err_{\Delta y}$, is plotted against the number of iterations.

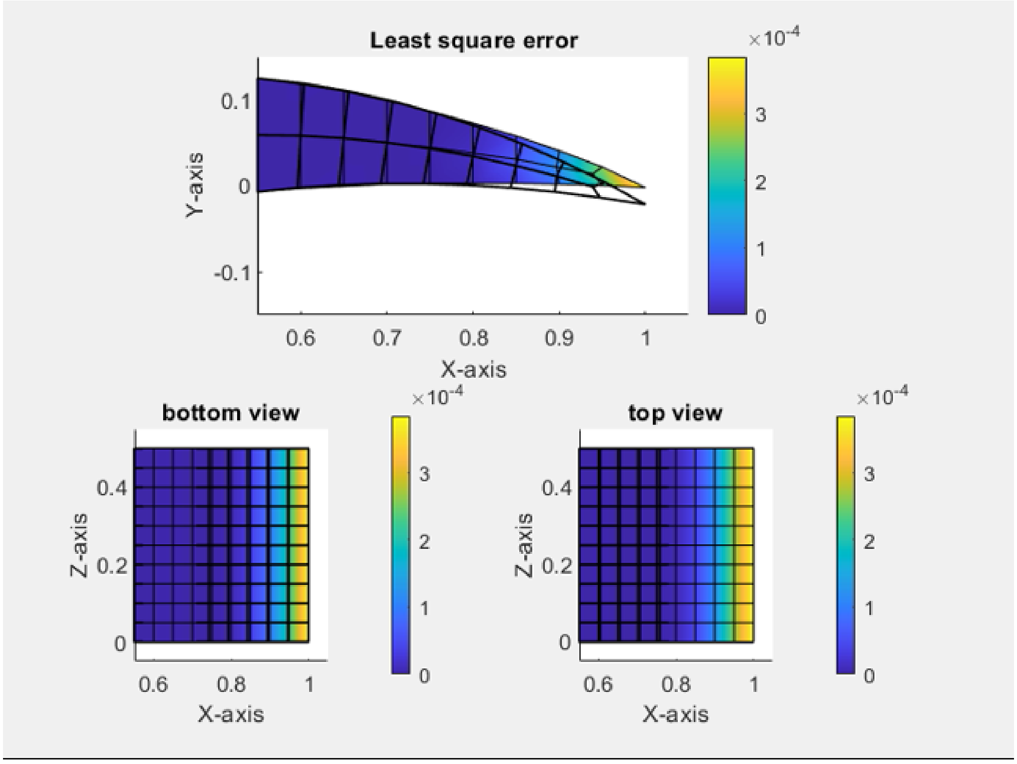
Comparing the graph of figure 5.15a with the one figure 5.7, the effect of the pressure distribution alters the error plot around the fifth iteration. However, this effect is barely noticeable and the controller is able to converge to the same error values of 5% for the bottom skin and 0.8% for the lower skin. This behaviour is also shown in figure 5.16: the pressure distribution, whose evolution is shown in figure 5.17, has little to no effect on the structure and the controller achieves the target shape on the third iteration.

To see a noticeable effect of the pressure distribution on the structure the velocity is increased to $V = 200\ m/s$. The maximum pressure value, which was 8000 Pa circa in the previous case study, now rises to 30000 Pa, as shown in figure 5.19. This increase of lifting action caused by the pressure does not create an evident change in the trend of the graph displayed in figure 5.15b: the error converges once again to the same error values of 5% the bottom surface and a value of less than 1% for the top one. On the other hand, looking at the sequence of morphing stages presented in figure 5.18, in particular in figure 5.18b, the result of the action of the pressure distribution is much more apparent, as it slows the convergence by slightly tilting in the upward direction the tip of the trailing edge. Thus far, the effects of the pressure distribution have affected the closed-loop iterations just barely. Conversely, when the velocity is augmented to $V = 300\ m/s$, the disturbance creates a high amplitude oscillating pattern, which is portrayed in figure 5.15c. As shown in the same image, however, the controller, is still able to converge to the usual 5% error value for the bottom skin and 0.8% for the lower skin, yet requiring 20 more iterations in the process. The response of the structure is caused by the oscillations of the pressure distribution and its peaks, that, as shown in figure 5.21, range now from 60000 to 80000 Pa. A few of the many iFEM reconstructed shapes along the iterations are collected in figure 5.20.

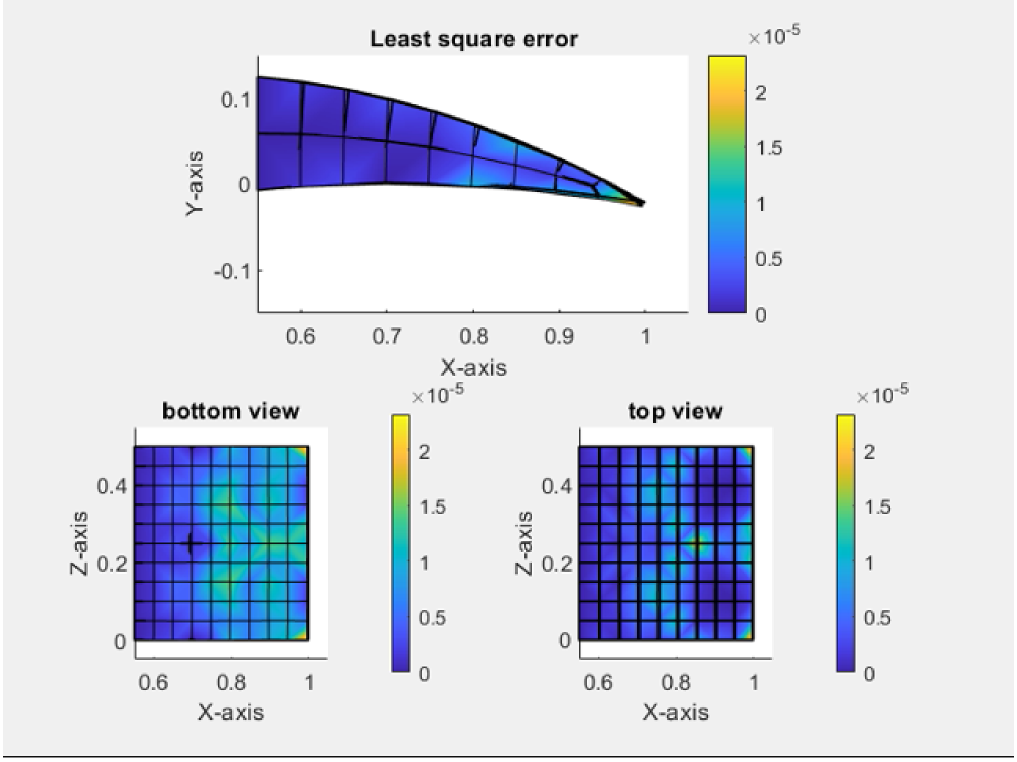
In table 5.2, the results presented in this chapter are summarised. As for the morphing plate structure, the closed-loop controller, thanks to the iFEM feedback, was able to accurately monitor the shape of the structure along the iterations. Furthermore, the control loop architecture was able to achieve different arbitrary target shapes with high precision, even in the presence of extreme external disturbances, proving once again the robustness of the iFEM feedback strategy.

		Convergence relative least-squares error	
		Transverse Forces $\lambda = 0$	Distributed pressure $\lambda = 10^{-16}$
k=0.4	V=0	5% bottom, 0.8% top	5% bottom, 0.8% top
k=1	V=0	2% bottom, 2% top	2% bottom, 1% top
k=0.4	V=100	5% bottom, 0.8% top	5% bottom, 0.8% top
	V=200	5% bottom, 0.8% top	5% bottom, 0.8% top
	V=300	5% bottom, 0.8% top	5% bottom, 0.8% top

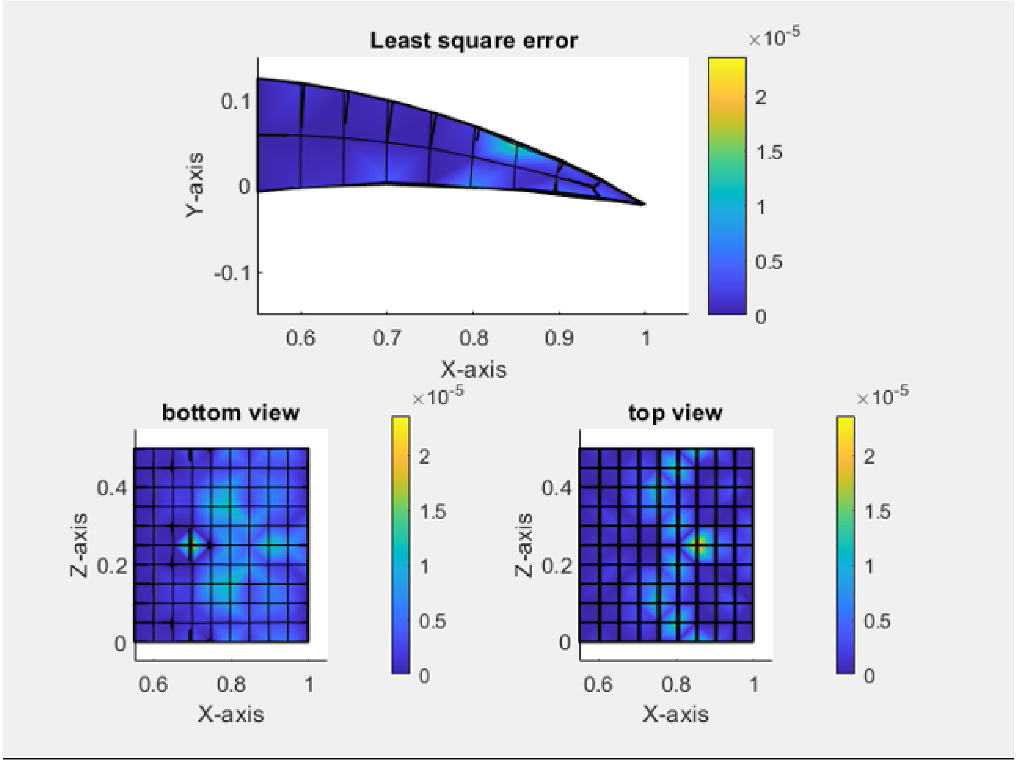
Table 5.2: Closed-loop control test on the morphing trailing edge: summary of the results



(a) First iteration

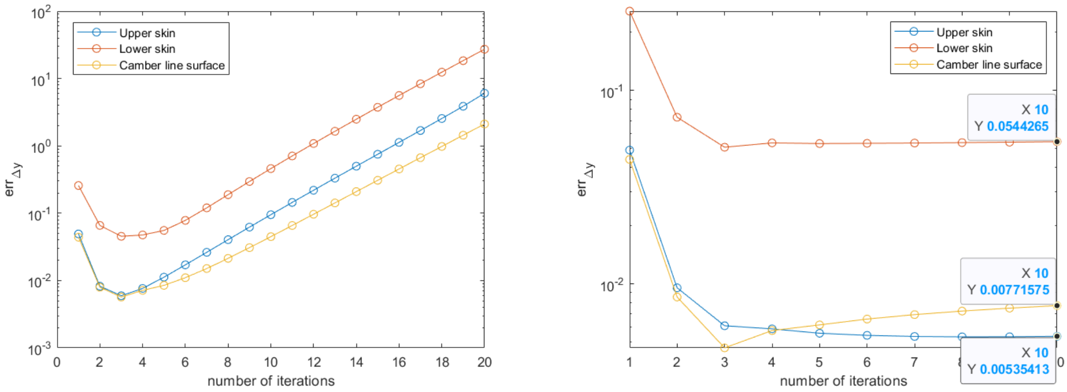


(b) Second iteration



(c) Last iteration

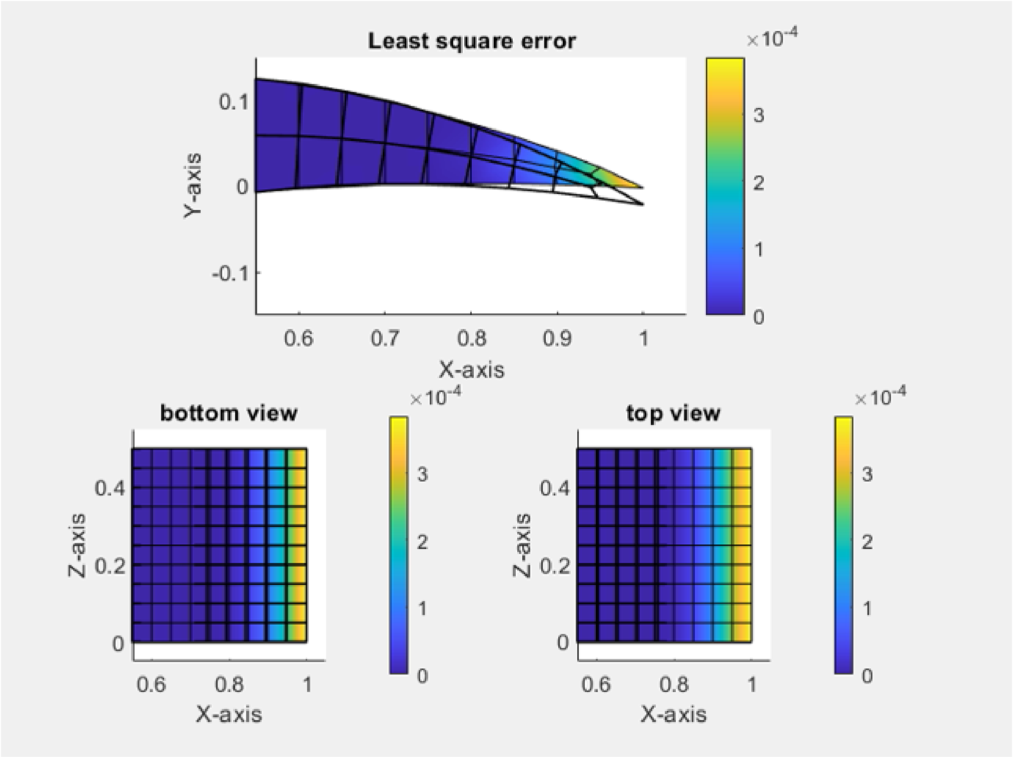
Figure 5.8: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0\ m/s$, sparse transverse forces actuation scheme



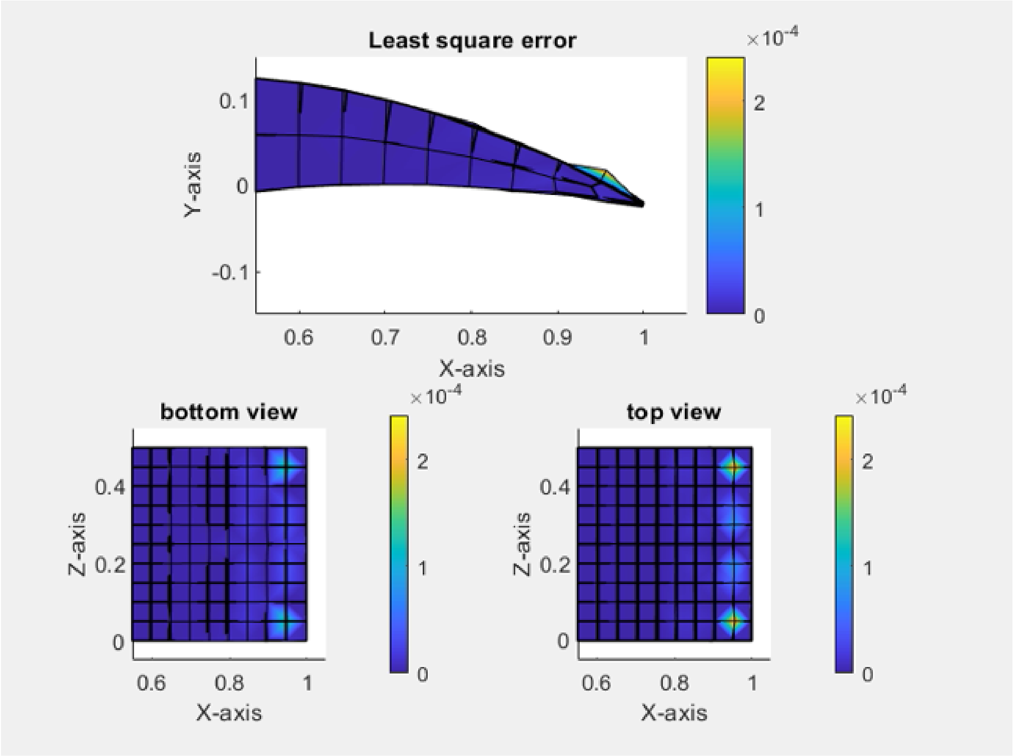
(a) $\lambda = 0$

(b) $\lambda = 10^{-16}$

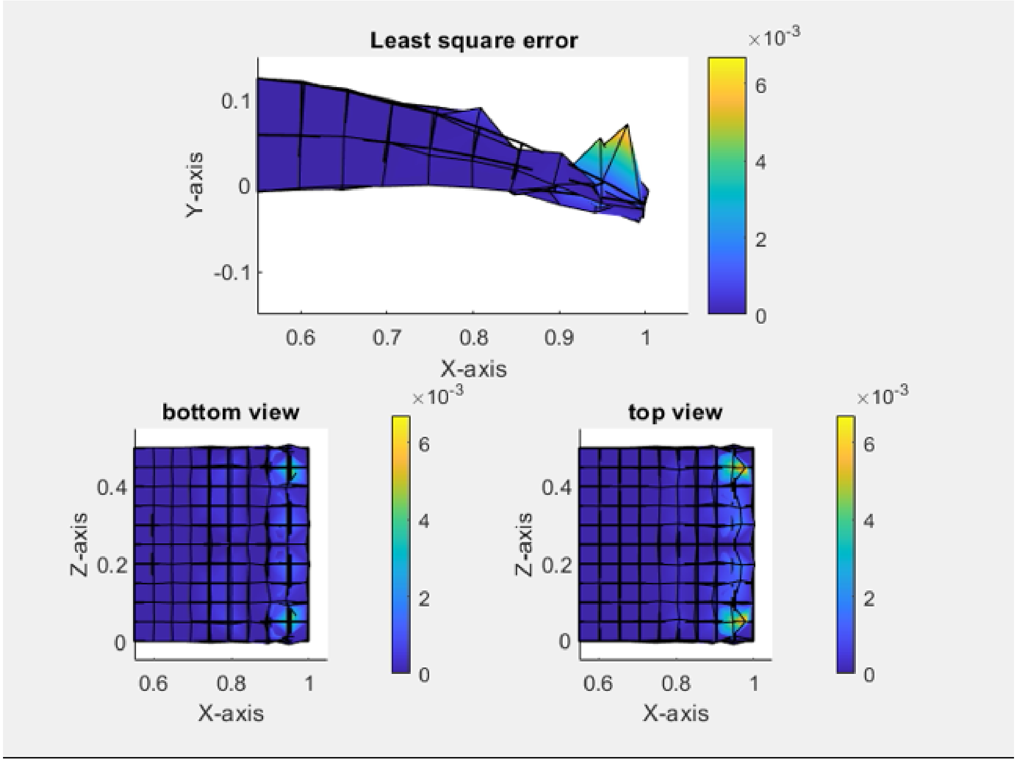
Figure 5.9: Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, $V = 0\ m/s$, distributed pressure actuation scheme



(a) First iteration

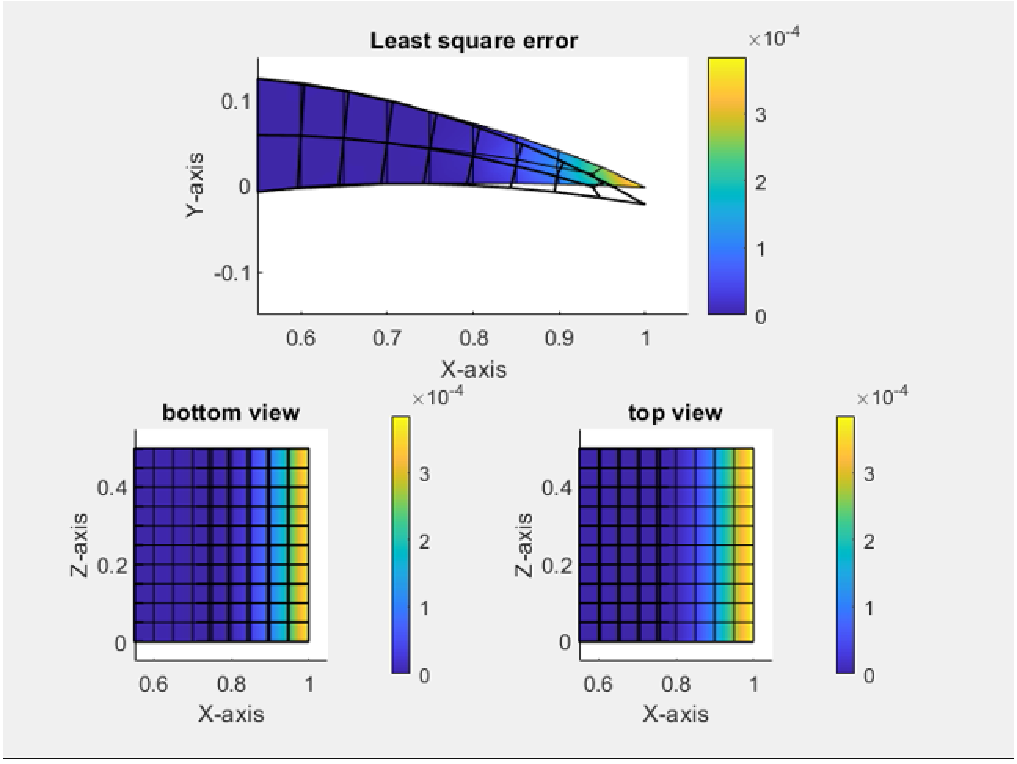


(b) Sixth iteration

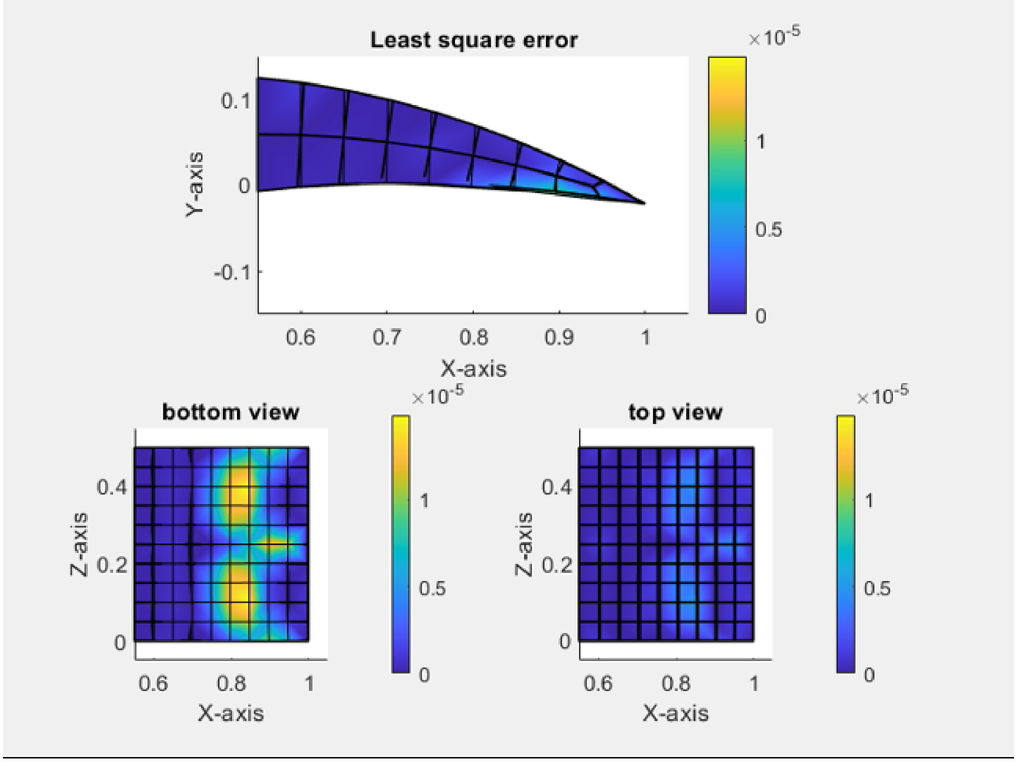


(c) Last iteration

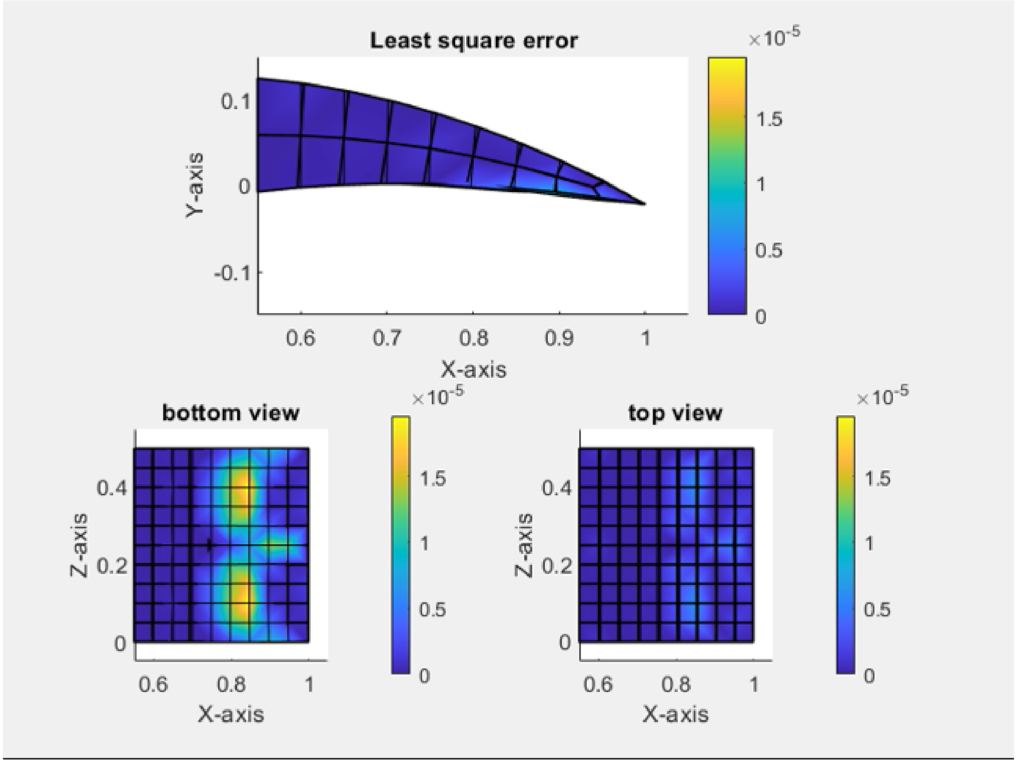
Figure 5.10: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0$ m/s, $\lambda = 0$ distributed pressure actuation scheme



(a) First iteration

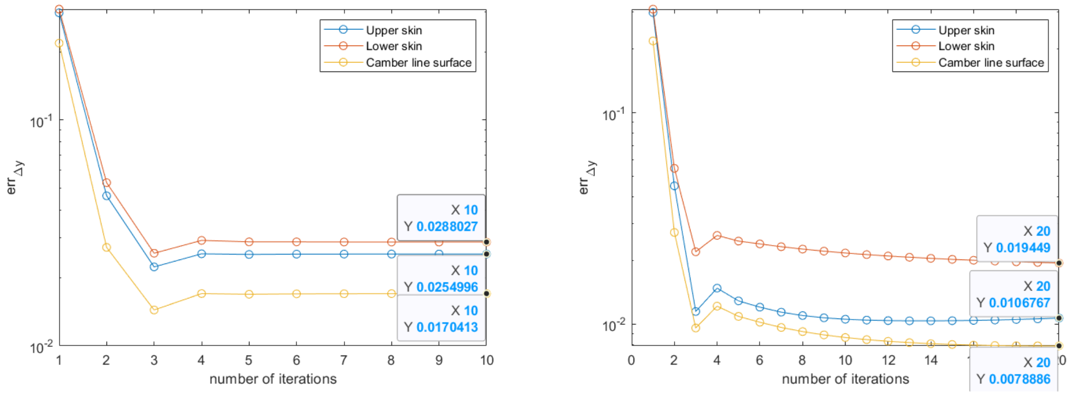


(b) Sixth iteration



(c) Last iteration

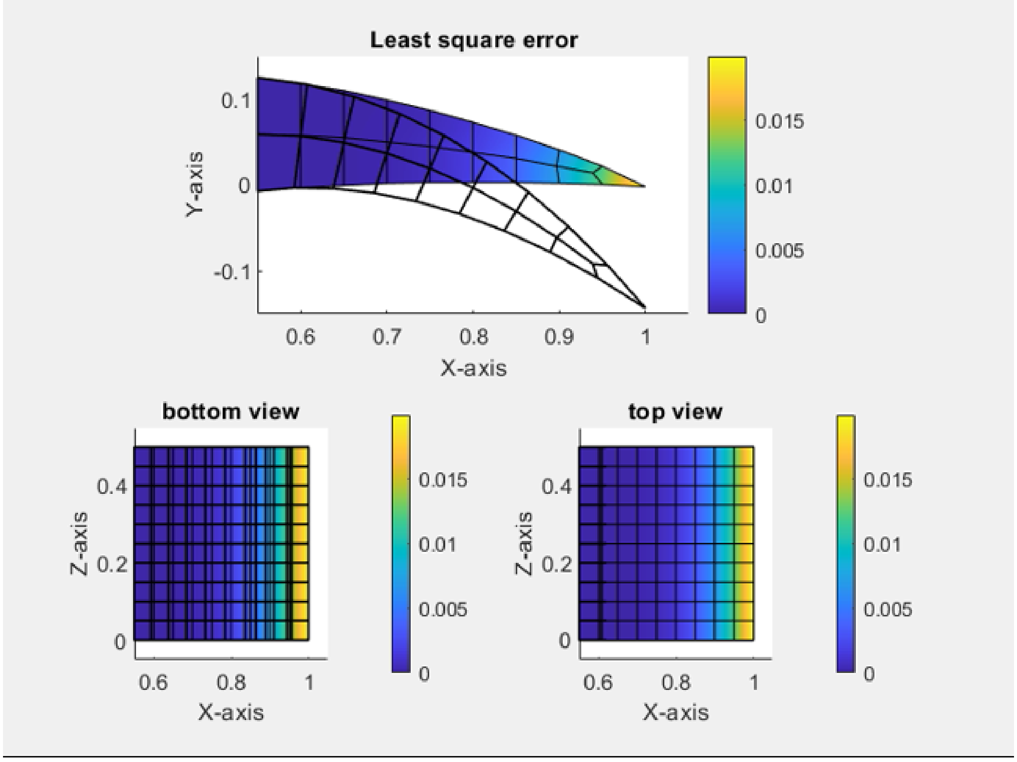
Figure 5.11: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 0\ m/s$, $\lambda = 10^{-16}$ distributed pressure actuation scheme



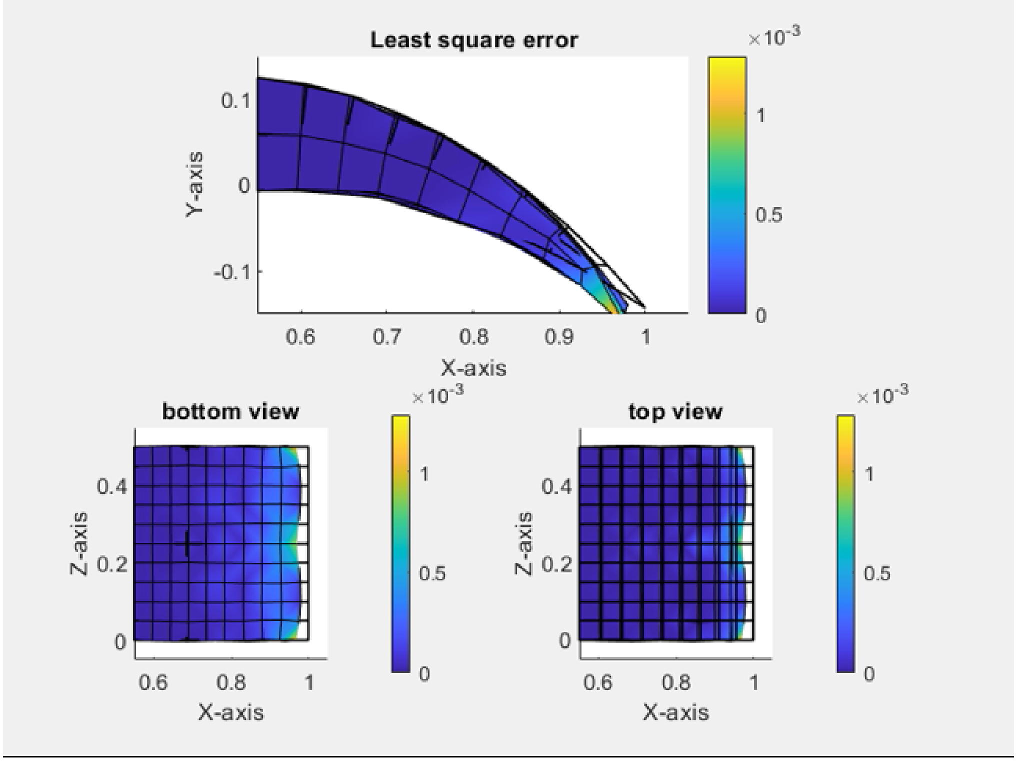
(a) Sparse transverse forces

(b) Distributed pressure

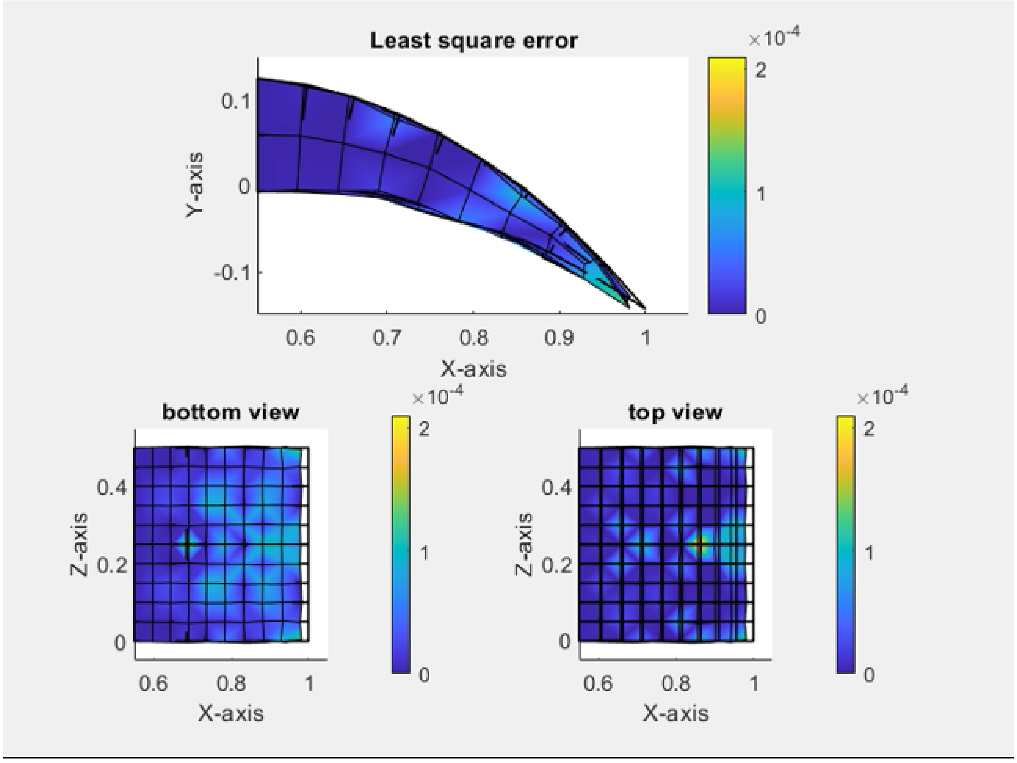
Figure 5.12: Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 1$, $V = 0\ m/s$



(a) First iteration

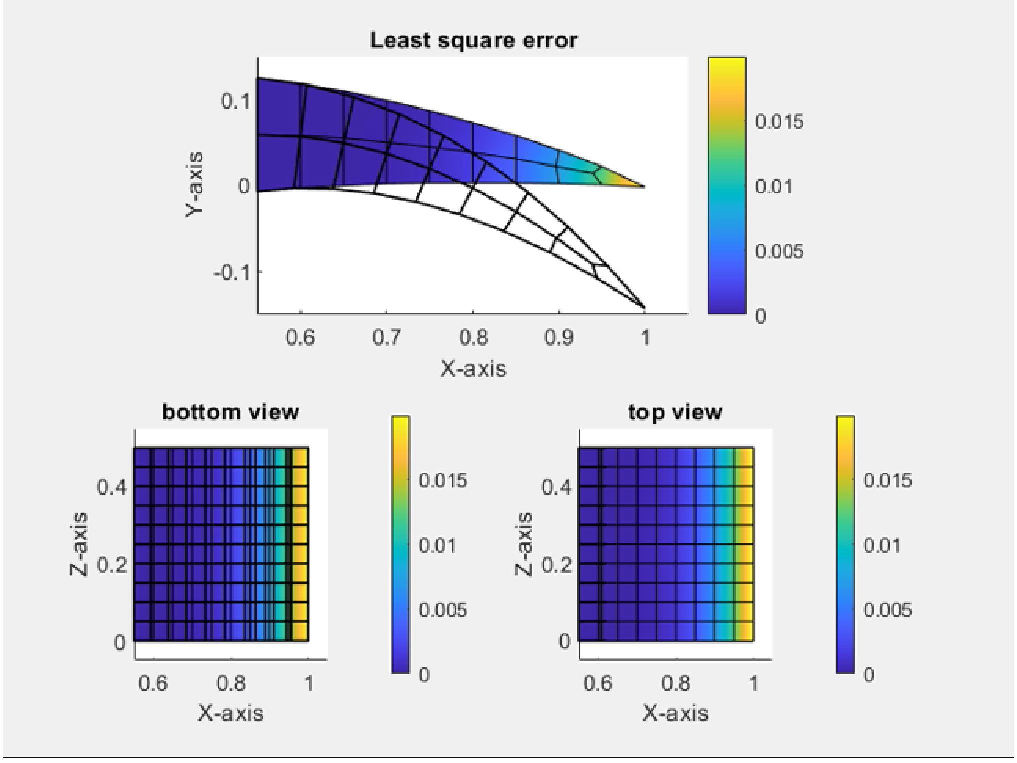


(b) Second iteration

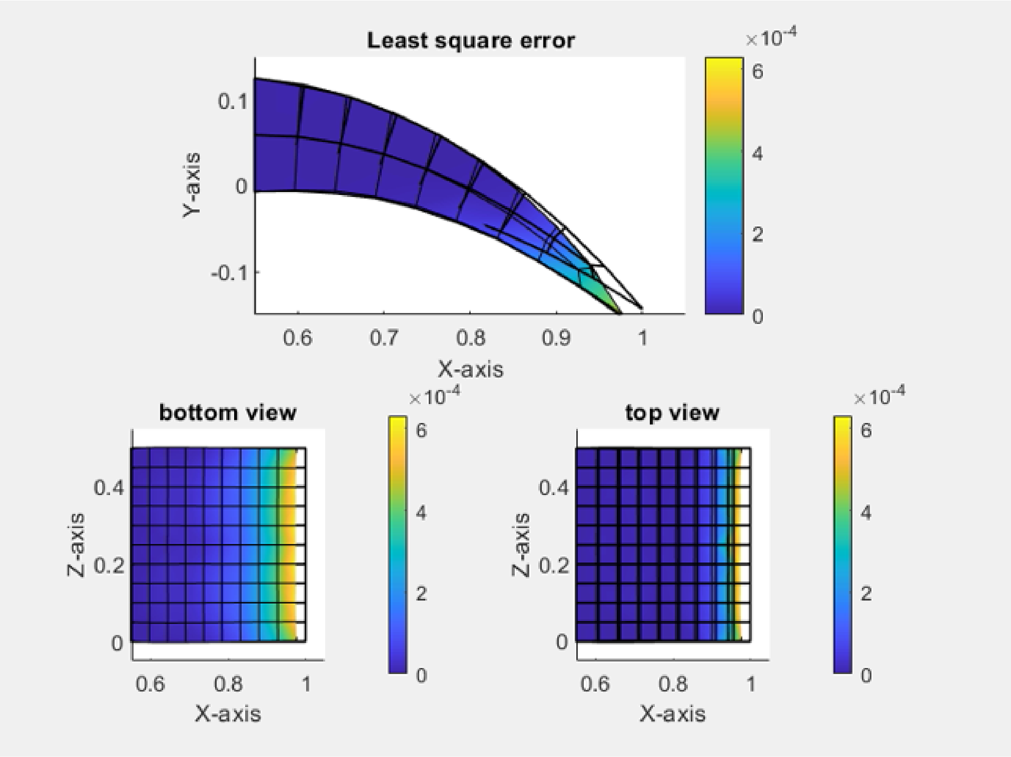


(c) Third iteration

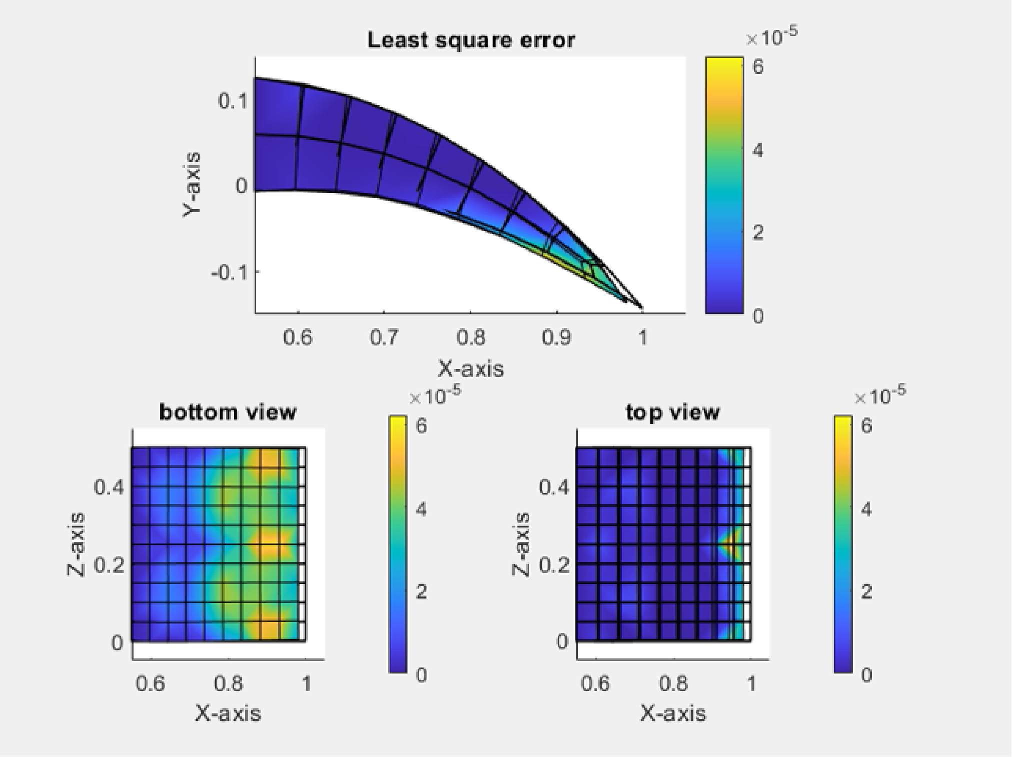
Figure 5.13: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 1$, $V = 0\ m/s$, discrete transverse forces actuation scheme



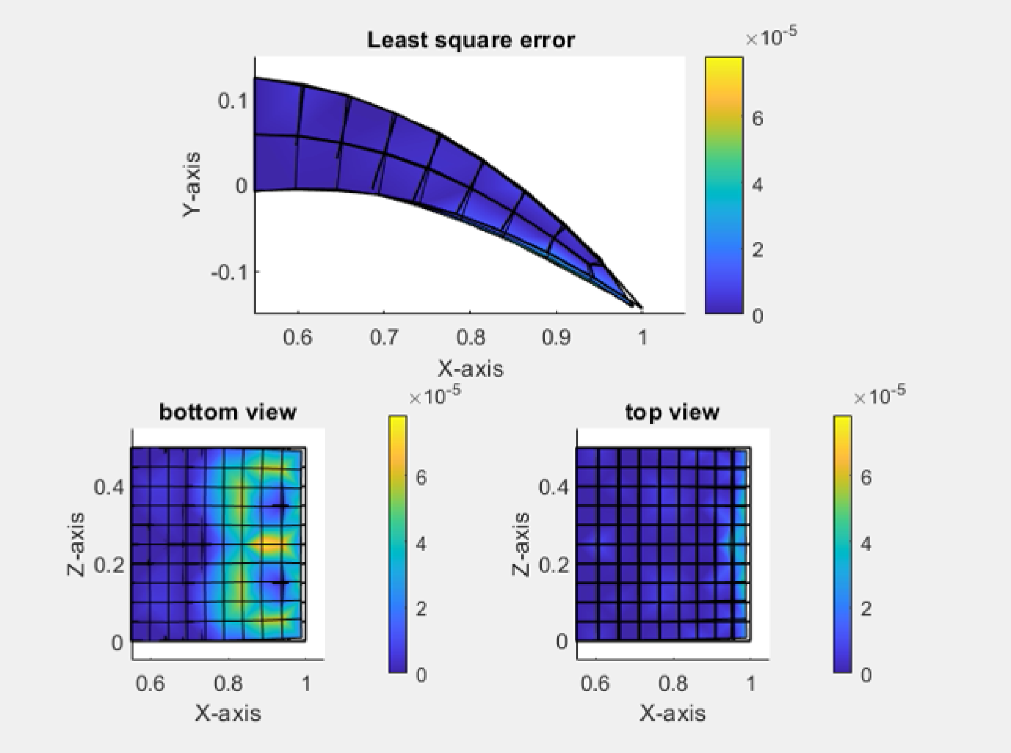
(a) First iteration



(b) Second iteration

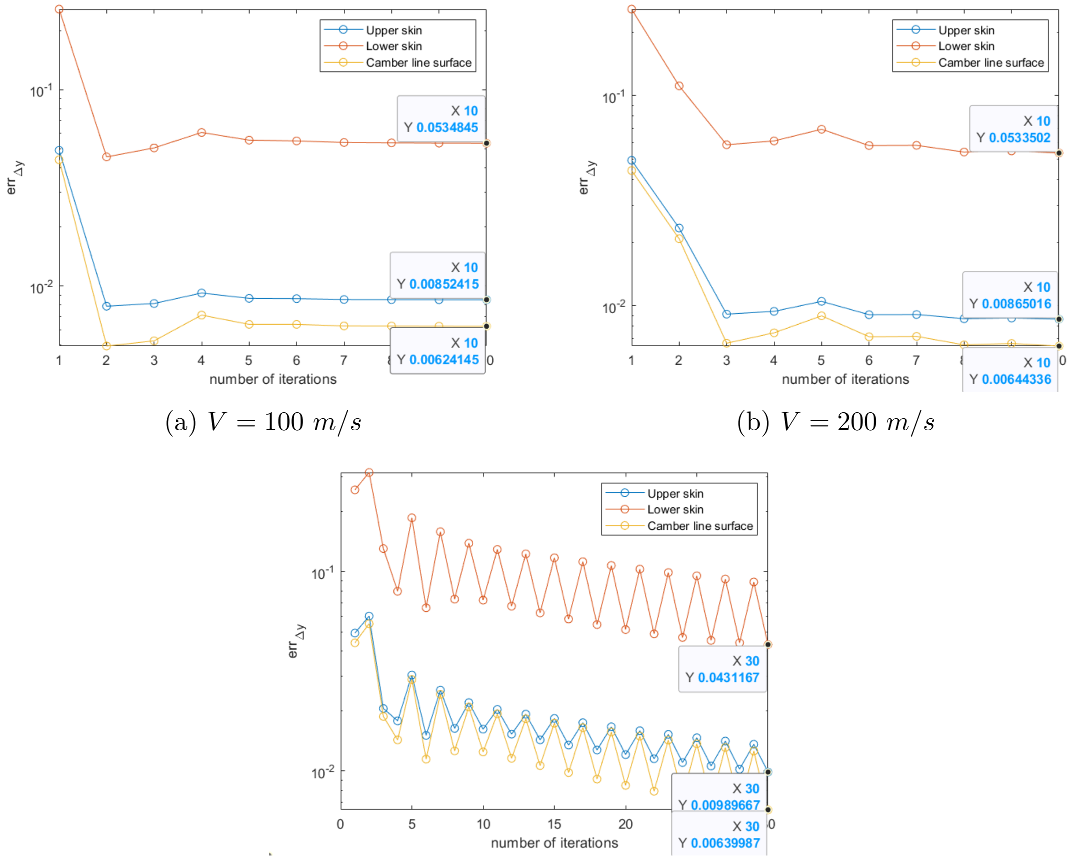


(c) Third iteration



(d) Last iteration

Figure 5.14: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 1$, $V = 0\ m/s$, distributed pressure actuation scheme

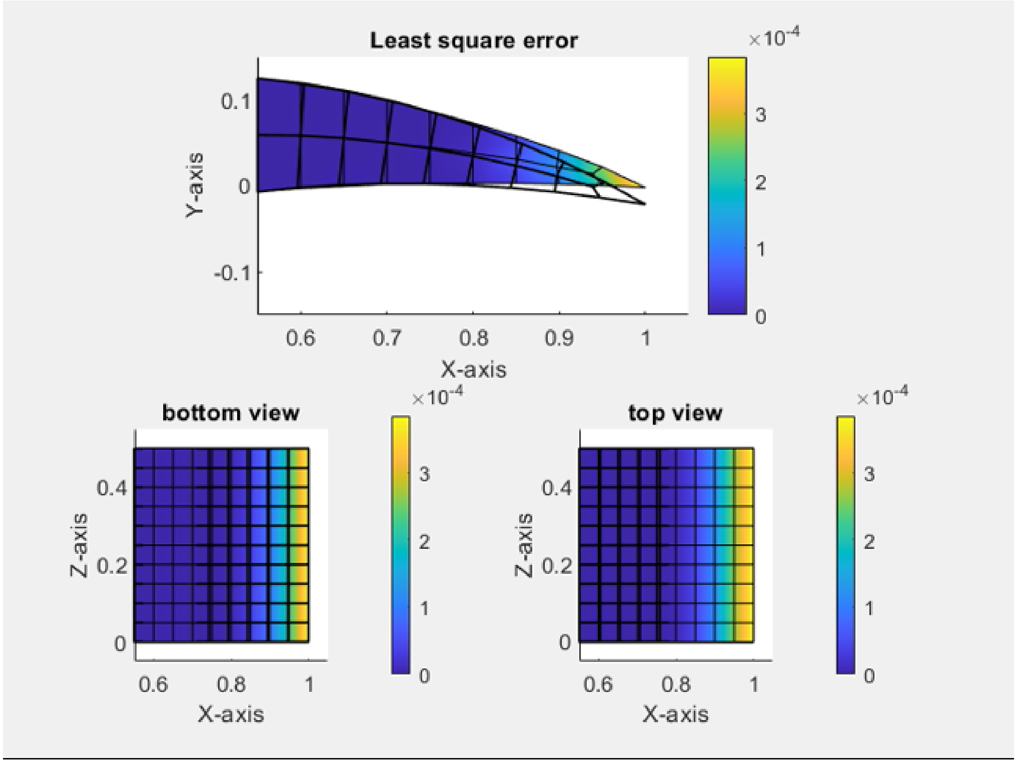


(a) $V = 100\ m/s$

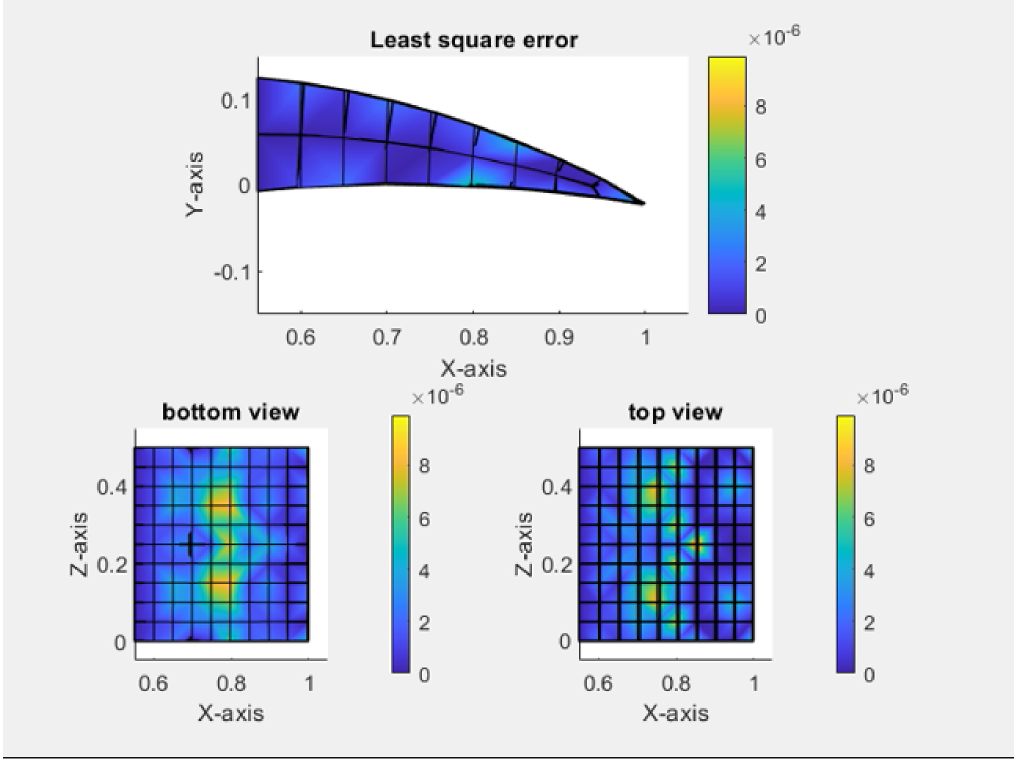
(b) $V = 200\ m/s$

(c) $V = 300\ m/s$

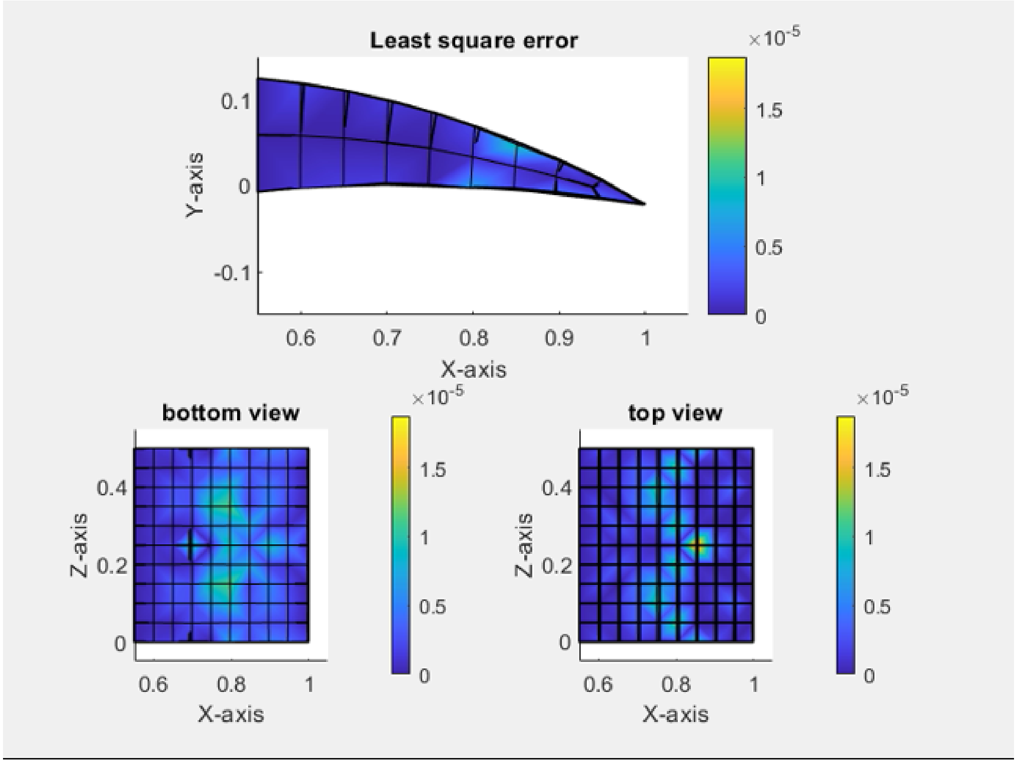
Figure 5.15: Transverse least-squares relative error, $err_{\Delta y}$, against the number of iterations - $k = 0.4$, sparse transverse forces actuation scheme



(a) First iteration

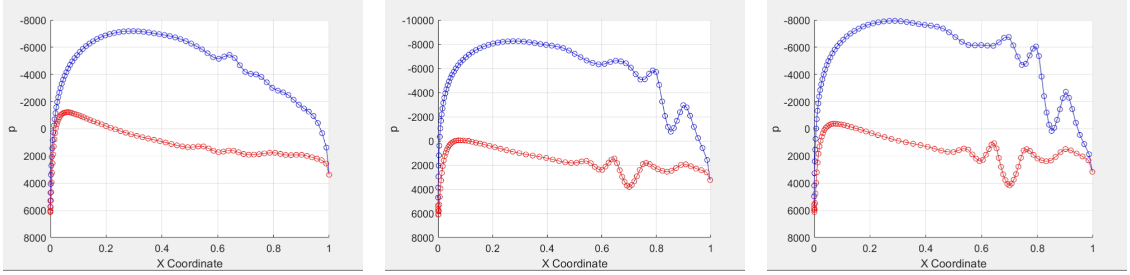


(b) Second iteration



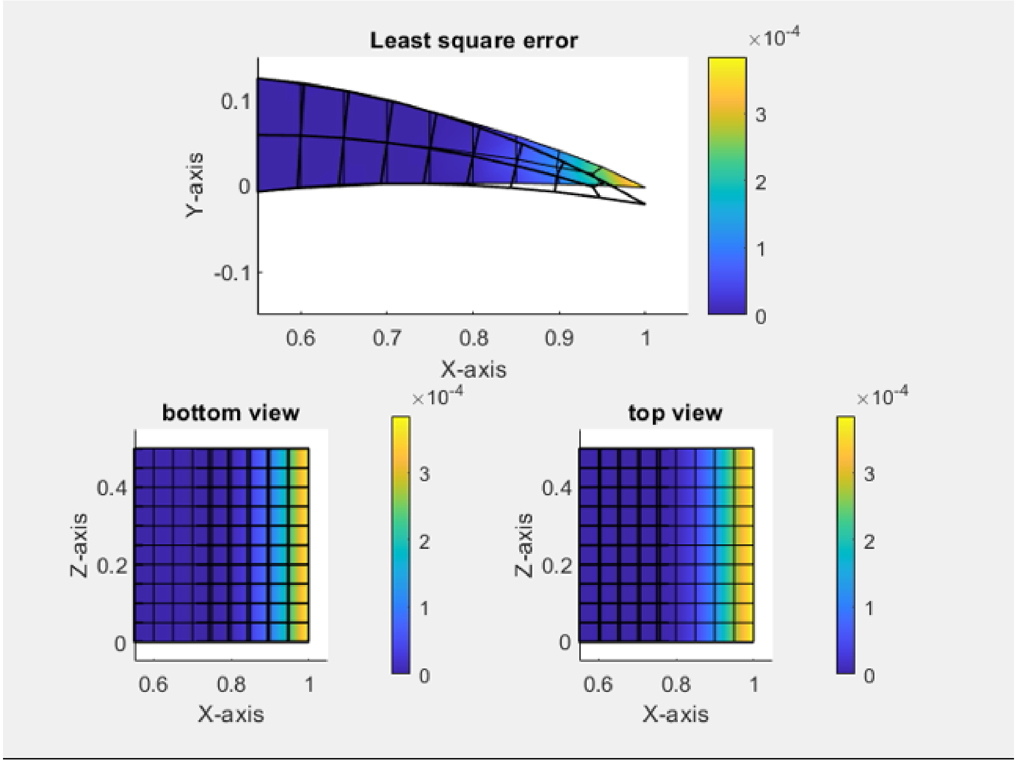
(c) Third iteration

Figure 5.16: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 100$ m/s, sparse transverse forces actuation scheme

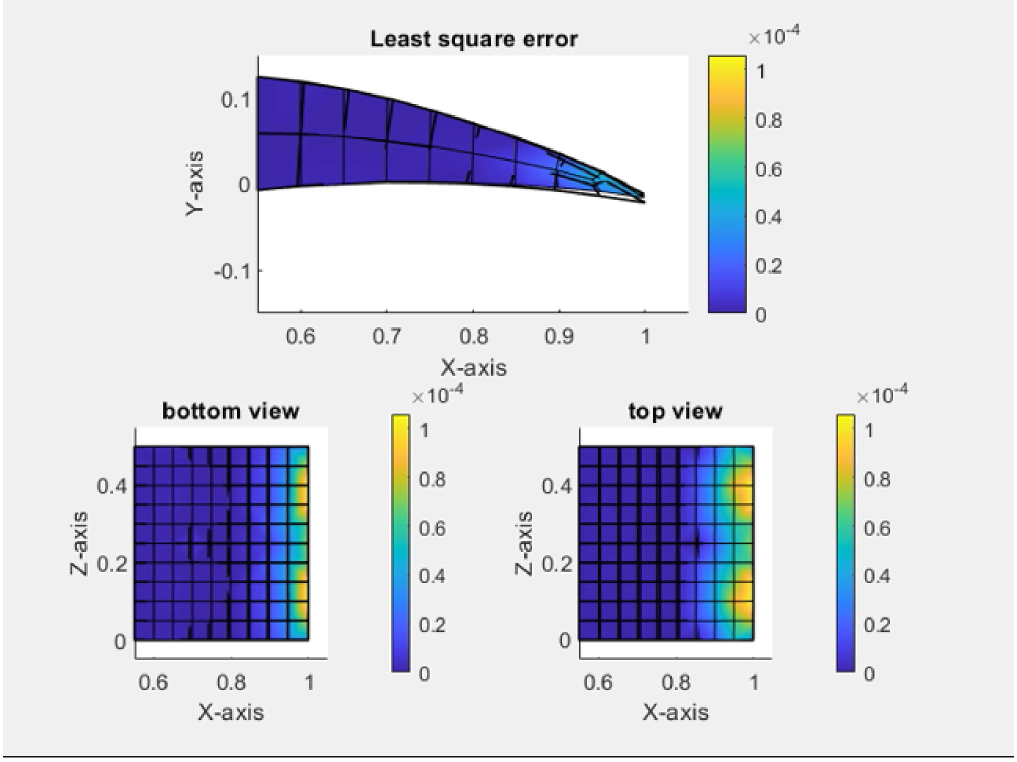


(a) First iteration (b) Second iteration (c) Third iteration

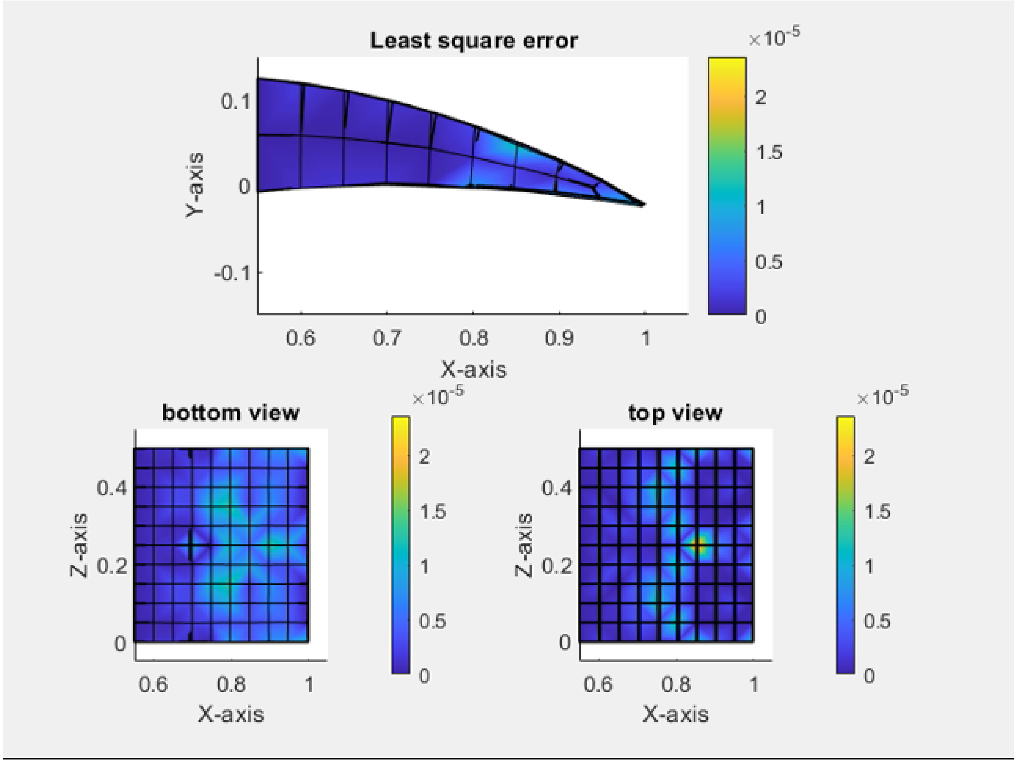
Figure 5.17: Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 100$ m/s, sparse transverse forces actuation scheme



(a) First iteration

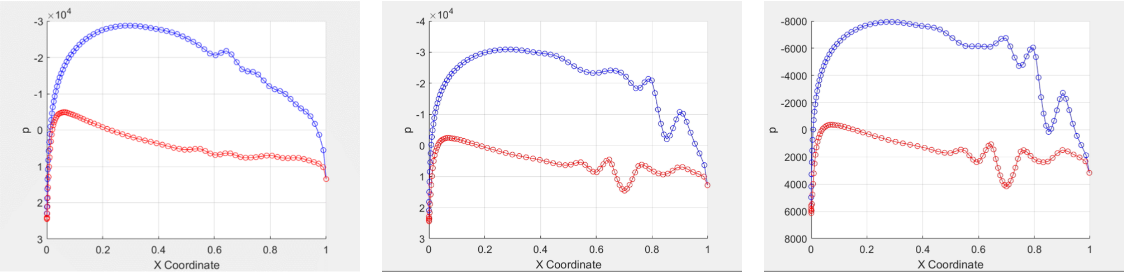


(b) Second iteration



(c) Third iteration

Figure 5.18: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 200$ m/s, sparse transverse forces actuation scheme

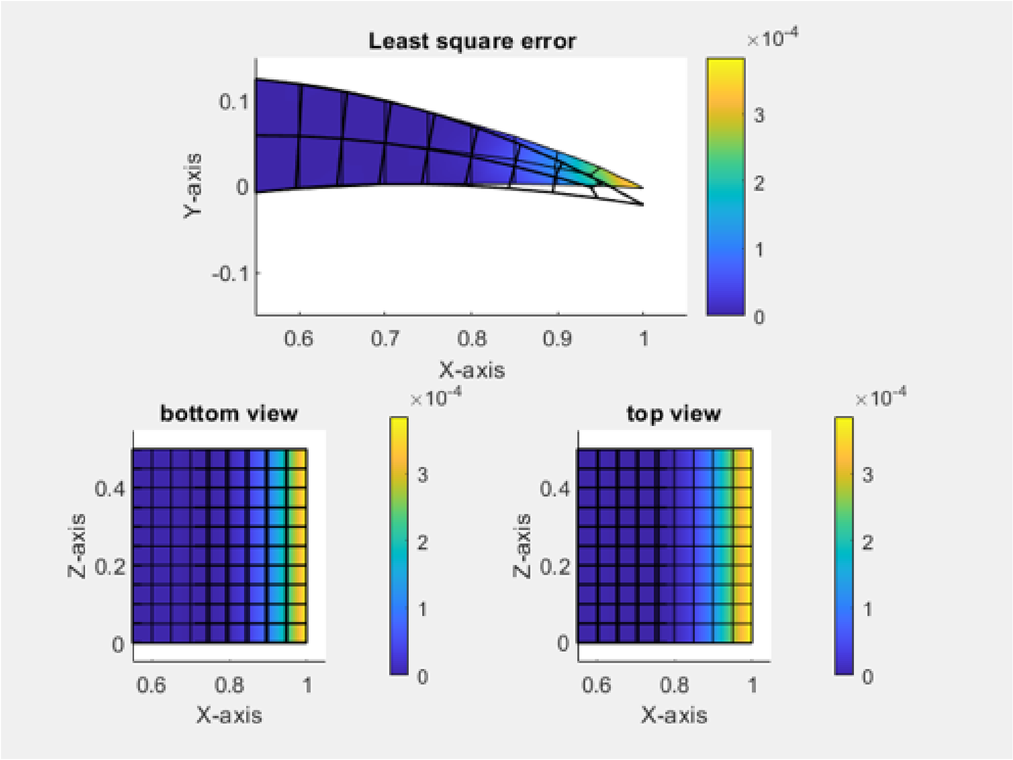


(a) First iteration

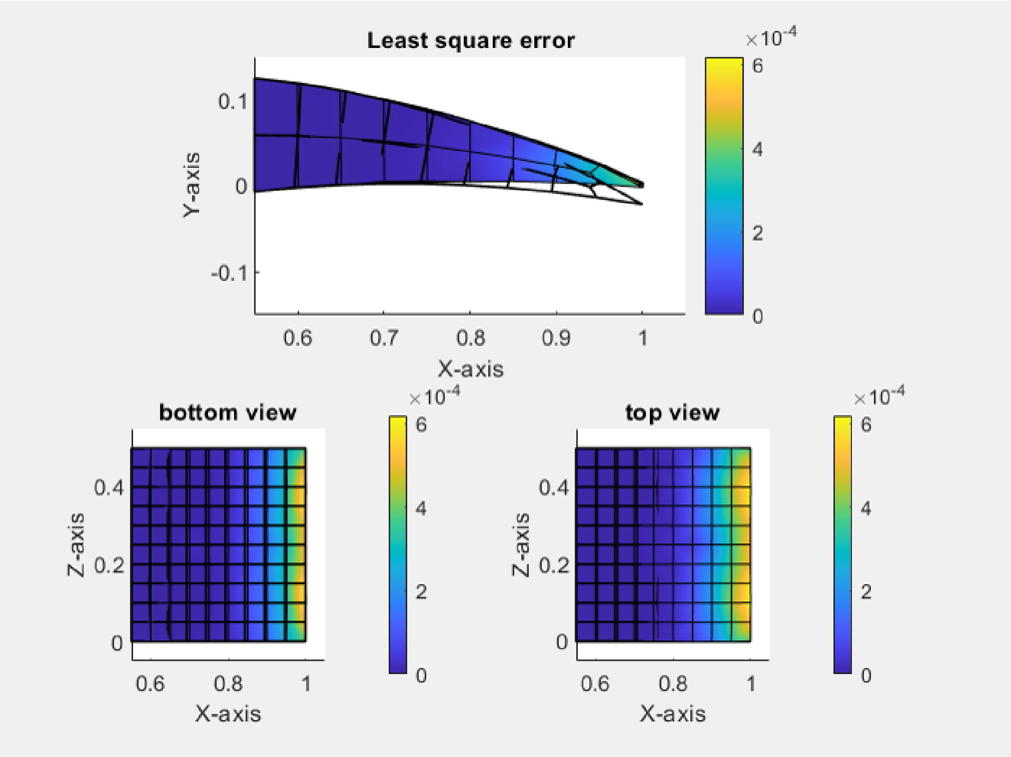
(b) Second iteration

(c) Third iteration

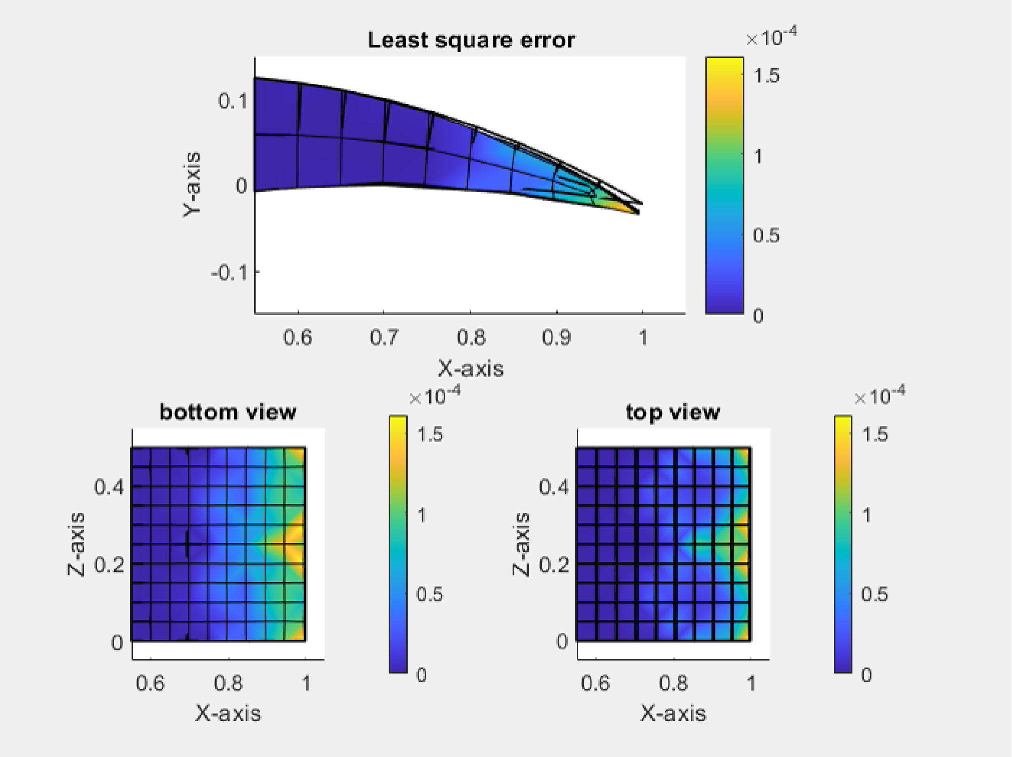
Figure 5.19: Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 200$ m/s, sparse transverse forces actuation scheme



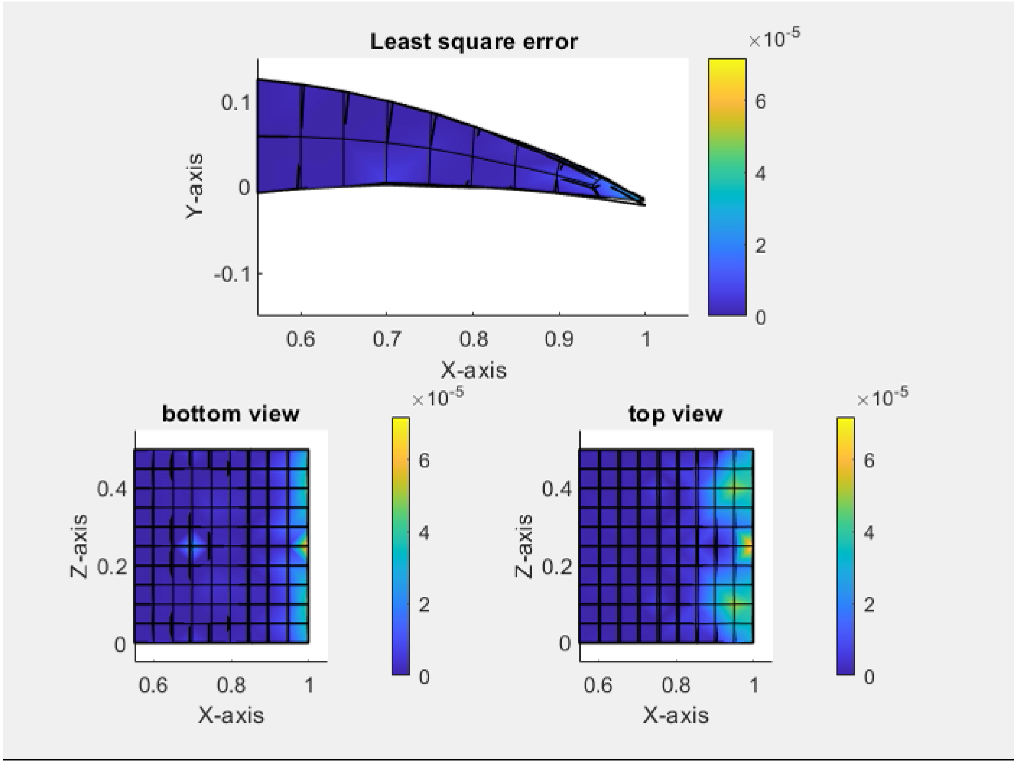
(a) First iteration



(b) Second iteration

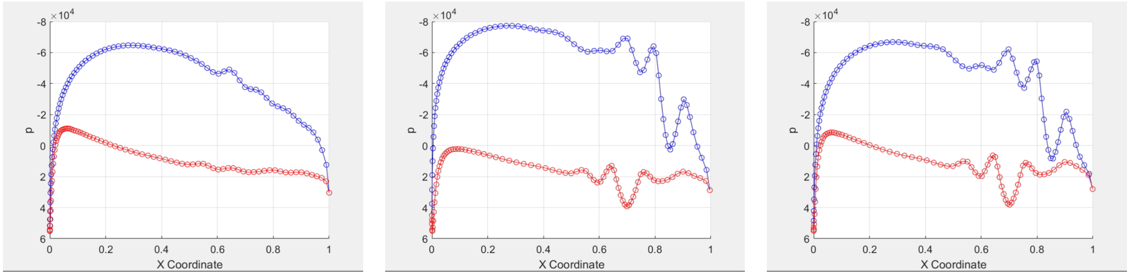


(c) Fifth iteration



(d) Tenth iteration

Figure 5.20: Δy_j^2 error plot on the iFEM reconstructed shape throughout the closed-loop iterations - $k = 0.4$, $V = 300$ m/s, sparse transverse forces actuation scheme



(a) First iteration

(b) Second iteration

(c) Third iteration

Figure 5.21: Pressure distribution throughout the closed-loop iterations - $k = 0.4$, $V = 300$ m/s, sparse transverse forces actuation scheme

Chapter 6

Conclusions

The objective of this master thesis was to develop and test a closed-loop control architecture based on an inverse Finite Element feedback strategy to actuate and monitor the shape of a morphing wing structure. This research was motivated by the disruptive potential of such structures, as well as that of the inverse Finite Element Method, which is envisioned as an enabling technology for future morphing designs [25].

The closed-loop control algorithm was implemented by combining the following strategies:

- simplifying the load control procedures found in the literature, a least-squares approach was used to compute the actuation loads at each iteration;
- the actuation itself and the strain measurements, instrumental for the iFEM feedback, were reproduced in a FEM environment;
- the inverse Finite Element Model was used as the core strategy to reconstruct the shape of the morphing structure, in order to provide an accurate feedback to the controller;
- lastly, different and specific error definitions were created to determine if the iFEM reconstructed shape was converging to the target one.

Using this control strategy, two different morphing concepts were studied: a simple plate structure and a morphing trailing edge. Both were fundamental in understanding the complexities of the various building blocks of the control algorithm. In particular, the simple plate model allowed to test

- the regularization methods, used to dampen the instabilities of the least-squares procedure;
- the superposition principle approach, exploited to simulate the FEM strain measurements of the structure while avoiding analysis runs with FEM software such as Nastran.

On the other hand, the trailing edge model enabled to study in a more realistic framework

- the effects of an external disturbance on the structure, modeled as the pressure distribution on the wing;
- the error definitions and their limits. In particular, as one definition proved unsuccessful in monitoring the convergence of the shape to the target one, the trailing edge studies allowed the development of a more suitable and effective error definition.

The results collected for both the plate structure and the trailing edge showed that

- the inverse Finite Element Method can be successfully implemented in a closed-loop architecture: for both the plate structure and the trailing edge model the control algorithm was able to converge to a variety of target solutions facing multiple and different external conditions. The regularization method employed, the Tikhonov regularization, proved essential, as it guaranteed process stability along the iterations;
- the iFEM reconstructed shape is able to provide accurate feedback to the controller that computes the actuation loads: in all case studies, for both the plate structure and the morphing trailing edge, the target shape was achieved with a least-squares relative error lower than 5%;
- the control loop implementing the iFEM feedback converges to the target shape even under extreme disturbance: for both models tested, even when the external loads produced displacements comparable or superior to those needed to morph the structure to the target shape, the controller was able to counteract such disturbances and achieved its objective.
- the iFEM feedback can be employed not only as a feedback strategy to control the shape of the structure: coupling the iFEM reconstructed shape with XFOIL, the algorithm was able to reconstruct the pressure distribution on the structure as it was morphing.

Nevertheless, there is still room for improvement. In future developments of this project, the objective will be to test the closed-loop architecture in a state-of-the-art framework. A few steps towards this goal would be

- changing the actuation scheme, to mimic the action of piezoelectric or SMA actuators, which are frequently used in morphing structure;
- changing the materials, to affect and produce a more realistic interaction between the structure and the external disturbances, such as the pressure distribution, acting on the structure itself.

Acknowledgements

I would like to express my sincerest and deepest gratitude to both my supervisors, Professor Gherlone and Dr. Rinto Roy. Working together with you was a wonderful experience. I have known Professor Gherlone since the second year of my bachelor's degree. He was the first teacher to introduce me to aerostructure engineering and I owe to him my passion for that discipline: he is an example of professionalism, always knowledgeable and available when needed, and it is not something I take for granted, so thank you. Thank you to Dr. Roy as well, for following me in this project step-by-step, always being able to come up with a solution to my problems. I hope to work again with you one day.

A te che stai leggendo, grazie, perché mi sei stato vicino in un momento così importante: chi mi conosce sa che questo è stato probabilmente il periodo più brutto e difficile della mia vita. Arrivare a questo traguardo, quest'anno, per me significa moltissimo, perché sono state molte le volte in cui credevo di non farcela e in cui avrei voluto mollare tutto. Non avrei mai potuto raggiungere questo obbiettivo senza le persone che mi hanno accompagnato in questi ultimi mesi e, per questo, vorrei dedicare a loro questi ringraziamenti.

Inizio dalla famiglia: grazie a zia Enza; a Franca, Marco e Paola; a Ezia, Leo, Andrea e Federica. Vi ringrazio per essermi stati vicino, anche solo con una telefonata, un invito a cena o una passeggiata in montagna, anche quando io non mi sono fatto sentire per mesi.

Ringrazio le ragazze della comitiva di Termoli: Laura Di Fabio e Di Giuseppe, Mariachiara, Martina, Linda, Nell e, in particolare, Antonella e Lucia. Grazie a voi due per avermi fatto sentire voluto bene e parte di un gruppo in cui temevo di non trovare più il mio posto.

Ringrazio i ragazzi della comitiva di Termoli: Massimiliano, Germano, Lorenzo, Andrea Corbo, Andrea Chiaromonte e Bondanese. Grazie perché siete la prova che la distanza e il tempo non possono spezzare la vera amicizia. Grazie soprattutto a Chiaro e Bondi, per essere stati la dimostrazione vivente del detto "Gli amici si vedono nel momento del bisogno".

Salendo su a Torino, un grande grazie va ai Power Ranger, che definire la mia squadra di pallavolo sarebbe riduttivo. Io so di amare la pallavolo, ma probabilmente questa passione non sarebbe tale se non avessi conosciuto voi. È davvero incredibile pensare che da una metà campo di 9m x 9m sia nata una amicizia così bella che va ben oltre quegli 81 m^2, e che mi fa pensare ora a Torino come la mia seconda casa. Ringrazio quindi Lucia, Marina, Eugenia, Davide, Gabriele, Matteo, Chiara, Claudia, Damiano e Marco.

Ho lasciato per ultime le persone a cui voglio dedicare i ringraziamenti più sentiti: grazie a Mattia, per avermi sopportato ed essere stato come un fratello in questi anni;

grazie, infine, ai miei genitori ai quali, ribaltando la mia solita frase, per questa volta dico: "Non vi sopporto, ma vi voglio molto bene".

References

[1] Silvestro Barbarino et al. “A Review of Morphing Aircraft”. In: *Journal of Intelligent Material Systems and Structures* 22.9 (2011), pp. 823–877. DOI: 10.1177/1045389X11414084.

[2] WordReference Random House Learner’s Dictionary of American English © 2023. URL: https://www.wordreference.com/definition/metamorphosis. (accessed: 08.10.2023).

[3] Emanuele Lelli - Enciclopedia per ragazzi (2006). *Metamorfosi.* URL: https://www.treccani.it/enciclopedia/metamorfosi_%28Enciclopedia-dei-ragazzi%29/. (accessed: 29.09.2023).

[4] Anna-Maria McGowan et al. “Recent results from NASA’s morphing project”. In: (lug. 2002).

[5] “Shape morphing of aircraft wing: Status and challenges”. In: *Materials & Design* 31.3 (2010), pp. 1284–1292. ISSN: 0261-3069. DOI: https://doi.org/10.1016/j.matdes.2009.09.011.

[6] R.Roy, C.Surace e M. Gherlone. “An Approach to Morphing Structure Control based on Real-time Displacement Reconstruction”.

[7] John Valasek. “Introduction”. In: *Morphing Aerospace Vehicles and Structures.* John Wiley e Sons, Ltd. Cap. 1, pp. 1–10. ISBN: 9781119964032. DOI: https://doi.org/10.1002/9781119964032.ch1.

[8] David L. Chandler - MIT News Office. *A new twist on airplane wing design.* URL: https://news.mit.edu/2016/morphing-airplane-wing-design-1103. (accessed: 30.09.2023).

[9] Benjamin Jenett et al. “Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures”. In: *Soft Robotics* 4.1 (2017). PMID: 28289574, pp. 33–48. DOI: 10.1089/soro.2016.0032.

[10] “Composite additive manufacturing of morphing aerospace structures”. In: *Manufacturing Letters* 23 (2020), pp. 85–88. ISSN: 2213-8463. DOI: https://doi.org/10.1016/j.mfglet.2019.12.004.

[11] Inken De Wit. *Take-off for a composite morphing aircraft.* URL: https://mavt.ethz.ch/news-and-events/d-mavt-news/2020/01/take-off-for-a-composite-morphing-aircraft.html. (accessed: 01.10.2023).

[12] M. Di Luca et al. “Bioinspired morphing wings for extended flight envelope and roll control of small drones”. In: *Interface Focus* 7.1 (2017), p. 20160092. DOI: 10.1098/rsfs.2016.0092.

[13] Roeland De Breuker et al. “Overview of the SmartX Wing Technology Integrator”. In: *Actuators* 11.10 (2022). DOI: 10.3390/act11100302.

[14] Nakash Nazeer, Xuerui Wang e Roger M. Groves. “Sensing, Actuation, and Control of the SmartX Prototype Morphing Wing in the Wind Tunnel”. In: *Actuators* 10.6 (2021). DOI: 10.3390/act10060107.

[15] Giulio Molinari, Andres F. Arrieta e Paolo Ermanni. “Aero-Structural Optimization of Three-Dimensional Adaptive Wings with Embedded Smart Actuators”. In: *AIAA Journal* 52.9 (2014), pp. 1940–1951. DOI: 10.2514/1.J052715.

[16] G Molinari et al. “Design, realization and structural testing of a compliant adaptable wing”. In: *Smart Materials and Structures* 24.10 (set. 2015), p. 105027. DOI: 10.1088/0964-1726/24/10/105027.

[17] Maurizio Arena et al. “Electro-Actuation System Strategy for a Morphing Flap”. In: *Aerospace* 6.1 (2019). DOI: 10.3390/aerospace6010001.

[18] Ignazio Dimino et al. “An Adaptive Control System for Wing TE Shape Control”. In: *Proc SPIE* 8690 (mar. 2013). DOI: 10.1117/12.2012187.

[19] Andrei Vladimir Popov et al. “Closed-Loop Control Simulations on a Morphing Wing”. In: *Journal of Aircraft* 45.5 (2008), pp. 1794–1803. DOI: 10.2514/1.37073.

[20] T. L. Grigorie et al. “A hybrid fuzzy logic proportional-integral-derivative and conventional on-off controller for morphing wing actuation using shape memory alloy Part 1: Morphing system mechanisms and controller architecture design”. In: *The Aeronautical Journal* 116.1179 (2012), pp. 433–449. DOI: 10.1017/S0001924000006977.

[21] Amanda Lampton, Adam Niksch e John Valasek. “Reinforcement Learning of Morphing Airfoils with Aerodynamic and Structural Effects”. In: *AIAA Infotech@Aerospace 2007 Conference and Exhibit*. DOI: 10.2514/6.2007-2805. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2007-2805. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2007-2805.

[22] Vinicius G. Goecks et al. “Control of Morphing Wing Shapes with Deep Reinforcement Learning”. In: gen. 2018. DOI: 10.2514/6.2018-2139.

[23] Kevin P. T. Haughn, Lawren L. Gamble e Daniel J. Inman. “Deep reinforcement learning achieves multifunctional morphing airfoil control”. In: *Journal of Composite Materials* 57.4 (2023), pp. 721–736. DOI: 10.1177/00219983221137644.

[24] Richard S. Sutton e Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.

[25] “A least-squares variational method for full-field reconstruction of elastic deformations in shear-deformable plates and shells”. In: *Computer Methods in Applied Mechanics and Engineering* 194.2 (2005). Selected papers from the 11th Conference on The Mathematics of Finite Elements and Applications, pp. 327–339. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2004.03.015.

[26] A. Tessler e Jan Spangler. "Title: Inverse FEM for Full-Field Reconstruction of Elastic Deformations in Shear Deformable Plates and Shells". In: (gen. 2004).

[27] "A quadrilateral inverse-shell element with drilling degrees of freedom for shape sensing and structural health monitoring". In: *Engineering Science and Technology, an International Journal* 19.3 (2016), pp. 1299–1313. ISSN: 2215-0986. DOI: `https://doi.org/10.1016/j.jestch.2016.03.006`.

[28] Rinto Roy et al. "Shape Sensing of Plate Structures Using the Inverse Finite Element Method: Investigation of Efficient Strain–Sensor Patterns". In: *Sensors* 20.24 (2020). DOI: `10.3390/s20247049`.

[29] Heinrich Voss. *Regularization of Least Squares Problems*. URL: `https://www.mat.tuhh.de/lehre/material/RegLS.pdf`. (accessed: 07.10.2023).