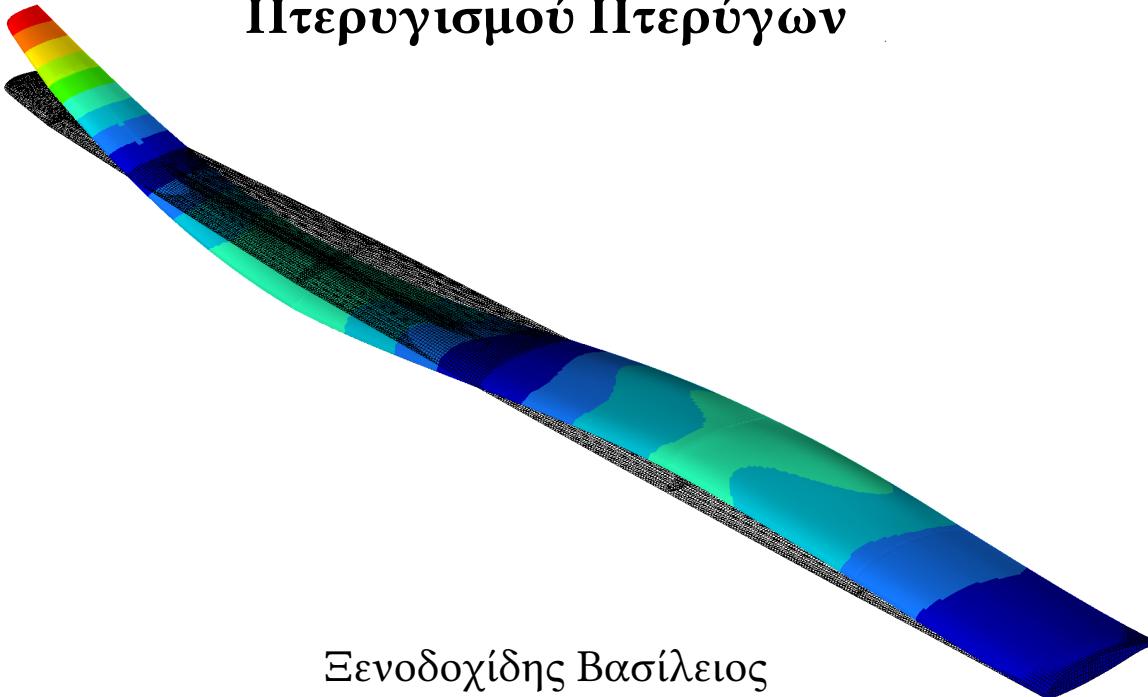




Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Μηχανολόγων Μηχανικών

Βελτιστοποίηση του Αεροελαστικού Πτερυγισμού Πτερύγων



Επιβλέποντες: Γιαγκόπουλος Δημήτριος, Αναπληρωτής Καθηγητής
Κουτσουπάκης Ιωσήφ, Υποψήφιος Διδάκτωρ

Θεσσαλονίκη, 20 Μαρτίου 2025

Περίληψη

Η παρούσα διπλωματική εργασία παρουσιάζει μια αεροελαστική ανάλυση μιας πτέρυγας, με στόχο τη βελτιστοποίηση του πολυστρωματικού σύνθετου υλικού της για τη μεγιστοποίηση της ταχύτητας πτερυγισμού ελαχιστοποιώντας παράλληλα τη μάζα. Η μελέτη χρησιμοποιεί τον επιλυτή OptiStruct της Altair για να αξιολογήσει την αεροελαστική σταθερότητα της δομής. Διάφοροι αλγόριθμοι βελτιστοποίησης, που υλοποιήθηκαν στην Python, χρησιμοποιήθηκαν για να εξερευνήσουν το χώρο σχεδιασμού και να εντοπίσουν βέλτιστους συνδυασμούς παραμέτρων που παράγουν αποδεκτά χαρακτηριστικά πτερυγισμού ενώ ταυτόχρονα ελαχιστοποιούν τη μάζα της κατασκευής. Επιπλέον, αναπτύχθηκε ένα μοντέλο νευρωνικού δικτύου για την πρόβλεψη της ταχύτητας πτερυγισμού με βάση βασικές δομικές και υλικές παραμέτρους, για να αναδειχθούν οι δυνατότητες που έχει ένα υποκατάστατο μοντέλο στην πρόβλεψη της ταχύτητας πτερυγισμού. Τα αποτελέσματα δείχνουν σημαντικές βελτιώσεις τόσο στην ταχύτητα πτερυγισμού όσο και στη μείωση βάρους, τονίζοντας τις δυνατότητες των προηγμένων τεχνικών βελτιστοποίησης και της μηχανικής μάθησης στον αεροελαστικό σχεδιασμό.

Abstract

This thesis presents an aeroelastic analysis of a wing structure with the objective of optimizing its composite material to maximize flutter speed while minimizing mass. The study utilizes Altair's OptiStruct solver to compute flutter curves and assess the aeroelastic stability of the structure. Various optimization algorithms, implemented in Python, were employed to explore the design space and identify optimal configurations that balance structural mass and aeroelastic performance. Furthermore, a neural network model was developed to predict flutter speed based on key structural and material parameters, to see the potential that a surrogate model has in predicting the flutter speed. The results demonstrate significant improvements in both flutter speed and weight reduction, highlighting the potential of advanced optimization techniques and machine learning in aeroelastic design.

Περιεχόμενα

1 Εισαγωγή	1
1.1 Διατύπωση Προβλήματος	2
1.2 Στόχος	3
1.3 Πεδίο Εφαρμογής και Περιορισμοί	4
1.4 Κίνητρο	5
2 Θεωρητικό Υπόβαθρο	6
2.1 Σύνθετα Πεπερασμένα Στοιχεία	7
2.1.1 Πεδίο Μετατοπίσεων	7
2.1.2 Διανύσματα Παραμορφώσεων	8
2.1.3 Σχέση Τάσεων - Παραμορφώσεων	10
2.1.4 Γενικευμένο Καταστατικό Μητρώο	12
2.1.5 Διακριτοποιημένη τάση και παραμόρφωση - Συναρτήσεις Μορφής	15
2.1.6 Μητρώο Στιβαρότητας	18
2.2 Η Μέθοδος Πλέγματος Στροβίλων (VLM)	23
2.2.1 Ο νόμος Biot Savart – το Vortex Filament	23
2.2.2 Ευθύγραμμο Τμήμα Δίνης	25
2.2.3 Υπολογιστική λύση επιφάνειας άντωσης με Vortex Ring Elements	27
2.3 Εξισώσεις Ανάλυσης Πτερυγισμού	32
2.3.1 Σύνδεση της κατασκευής με την αεροδυναμική – Infinite Plate splines	32

2.3.2	Η μέθοδος επίλυσης πτερυγισμού PK	35
2.4	Τεχνικές βελτιστοποίησης	38
2.4.1	Μέθοδος γραμμικής αναζήτησης Brent – Dekker	38
2.4.2	Η Μέθοδος του Powell	40
2.4.3	Γενετικός Αλγόριθμος	42
2.4.4	Νευρωνικά Δίκτυα	45
3	Μεθοδολογία	50
3.1	Εισαγωγή στο Πρόβλημα	51
3.2	Μοντέλο Κύριας Πτέρυγας ASW 28	54
3.2.1	Γεωμετρία Πτέρυγας & Διακριτοποίηση Γεωμετρίας	54
3.2.2	Ορισμός Ιδιοτήτων Υλικών	57
3.2.3	Συνοριακές Συνθήκες	59
3.2.4	Αεροδυναμικό Πλέγμα	59
3.2.5	Η διασυνδετική καμπύλη Spline	62
3.2.6	Setup Αεροελαστικού προβλήματος στο Optistruct	63
3.3	Optistruct – Διεπαφή με Python	69
3.3.1	Αποτελέσματα Ανάλυσης Πτερυγισμού & Python	69
3.3.2	Τροποποίηση της εισόδου του Optistruct χρησιμοποιώντας Python	71
3.4	Πρόβλημα Βελτιστοποίησης	73
3.4.1	Εφαρμογή της Μεθόδου Powell	73
3.4.2	Εφαρμογή του Γενετικού Αλγορίθμου	77
3.4.3	Πρόβλεψη Ταχύτητας πτερυγισμού μέσω Νευρωνικών Δικτύων	79
4	Αποτελέσματα	85
4.1	Ανάλυση Ιδιομορφών	86
4.2	Αρχική Ανάλυση Πτερυγισμού	89
4.3	Μέθοδος Βελτιστοποίησης Powell	92
4.4	Βελτιστοποίηση με τη μέθοδο Γενετικών Αλγορίθμων	101

4.5	Αποτελέσματα Πρόβλεψης μέσω Νευρωνικών Δικτύων	105
4.5.1	Εξέταση δεδομένων εκπαίδευσης	105
4.5.2	Νευρωνικό Δίκτυο με 1 Κρυφό Στρώμα	107
4.5.3	Νευρωνικό Δίκτυο με 2 Κρυφά Στρώματα	108
4.5.4	Νευρωνικό Δίκτυο με 4 Κρυφά Στρώματα	109
4.5.5	Νευρωνικό Δίκτυο με 6 Κρυφά Στρώματα	110
4.5.6	Νευρωνικό Δίκτυο με Ρυθμισμένες Υπερπαραμέτρους	111
5	Συμπεράσματα & Μελλοντική Εργασία	114
5.1	Βελτιστοποίηση	115
5.2	Πρόβλεψη με Νευρωνικά Δίκτυα	117
5.3	Μελλοντική Εργασία	118
A'	Κώδικας Βελτιστοποίησης Python	121

Κατάλογος Σχημάτων

2.1	Σύστημα συντεταγμένων σύνθετου στοιχείου	8
2.2	Ορισμός στρωμάτων σε composite laminated plate [11]	11
2.3	natural and physical coordinate space of quadrilateral plate element [1]	18
2.4	Σημεία Gauss για πλήρη και μειωμένη ολοκλήρωση στοιχείων τεσσάρων κόμβων	19
2.5	Ορισμός Local και Global συστημάτων συντεταγμένων [11]	22
2.6	Curved Three-dimensional vortex filament of strength Γ [12]	23
2.7	Ταχύτητα σε σημείο P λόγω ενός vortex distribution [7]	24
2.8	Επαγόμενη ταχύτητα από straight Vortex Segment [7]	25
2.9	Γεωμετρία Vortex segment [7]	27
2.10	Απεικόνιση στοιχείου Vortex Ring Element	28
2.11	Vortex ring elements σε πλέγμα [7]	29
2.12	Διάταξη από wing και wake panels [7]	30
2.13	Σύστημα συντεταγμένων Surface Spline [13]	33
2.14	Μια πιθανή διαμόρφωση σημείων [2]	39
2.15	Τυπική τερματική διαμόρφωση σημαντικών σημείων [2]	40
2.16	Η μέθοδος Cyclic Coordinate Search [8]	41
2.17	Η μέθοδος Powell [8]	42
2.18	Διασταύρωση (Crossover) ενός σημείου [8]	44
2.19	Διασταύρωση (Crossover) δύο σημείων [8]	44
2.20	Ομοιόμορφη διασταύρωση (Uniform Crossover) [8]	44

2.21	Η δομή ενός νευρώνα	46
2.22	Η δομή ενός νευρωνικού δικτύου [14]	47
3.1	Τεχνικά δεδομένα του ανεμοπλάνου ASW 28 [3]	52
3.2	Εμπρόσθια, πλάγια και άνω όψη του ανεμοπλάνου ASW 28 [3]	53
3.3	Εξωτερική Γεωμετρία της πτέρυγας του ASW 28 (length in meters)	54
3.4	Εσωτερική Γεωμετρία της πτέρυγας του ASW 28	55
3.5	Διακριτοποίηση Εσωτερικής Γεωμετρίας της πτέρυγας του ASW 28	56
3.6	Διακριτοποίηση Εξωτερικής Γεωμετρίας της πτέρυγας του ASW 28	57
3.7	Συνοριακές συνθήκες της πτέρυγας του ASW 28	59
3.8	Σύστημα συντεταγμένων ενός CAERO1 Aerodynamic panel [4]	60
3.9	Τα CAERO1 macro elements της πτέρυγας του ASW 28	61
3.10	SPLINE1 entries του μοντέλου της πτέρυγας ASW 28	63
3.11	Επεξήγηση του Optistruct .flt file	69
3.12	Παραδείγματα από V-g και V-f plots με τέσσερις ιδιομορφές [4]	70
3.13	Αντισυμμετρικά στρώματα σύνθετου υλικού	74
3.14	Δομή Νευρωνικού δικτύου με 1 κρυφό στρώμα	82
3.15	Δομή Νευρωνικού δικτύου με 2 κρυφά στρώματα	82
3.16	Δομή Νευρωνικού δικτύου με 4 κρυφά στρώματα	83
3.17	Δομή Νευρωνικού δικτύου με 6 κρυφά στρώματα	83
4.1	Ο πρώτες 6 ιδιομορφές του φτερού ASW 28	87
4.2	Αρχικό διάγραμμα πτερυγισμού για τις 4 πρώτες ιδιομορφές.	89
4.3	Διάγραμμα πτερυγισμού για την 3 ^η ιδιομορφή	90
4.4	Τιμή της αντικειμενικής συνάρτησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 1)	92
4.5	Τιμή κάθε μεταβλητής βελτιστοποίησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 1)	93
4.6	Διαγράμματα ανάλυσης πτερυγισμού από τη μέθοδο του Powell. (Περίπτωση 1)	94

4.7	Ανάλυση πτερυγισμού της 3 ^{ης} ιδιομορφής με τη μέθοδο του Powell. (Περίπτωση 1)	95
4.8	Τιμή της αντικειμενικής συνάρτησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 2)	96
4.9	Τιμή κάθε μεταβλητής βελτιστοποίησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 2)	97
4.10	Διαγράμματα ανάλυσης πτερυγισμού από τη μέθοδο του Powell. (Περίπτωση 2)	98
4.11	Ανάλυση πτερυγισμού της 3 ^{ης} ιδιομορφής με τη μέθοδο του Powell. (Περίπτωση 2)	99
4.12	Εξέλιξη του fitness function της καλύτερης λύσης κάθε γενιάς κατά τη διαδικασία βελτιστοποίησης	101
4.13	Εξέλιξη των τιμών των γονιδίων κατά τη διάρκεια της διαδικασίας βελτιστοποίησης.	102
4.14	Διάγραμμα πτερυγισμού από τον αλγόριθμο γενετικής βελτιστοποίησης.	103
4.15	Διάγραμμα πτερυγισμού της 1 ^{ης} ιδιομορφής από αλγόριθμο γενετικής βελτιστοποίησης	104
4.16	pair plot των δεδομένων εκπαίδευσης	106
4.17	Απόδοση Νευρωνικου δικτύου με 1 κρυφό στρώμα	107
4.18	Απόδοση Νευρωνικου δικτύου με 2 κρυφά στρώματα	108
4.19	Απόδοση Νευρωνικου δικτύου με 4 κρυφά στρώματα	109
4.20	Απόδοση Νευρωνικου δικτύου με 6 κρυφά στρώματα	110
4.21	Δομή βελτιστοποιημένου Νευρωνικού Δικτύου.	112
4.22	Απόδοση Hyperparameter tuned Νευρωνικού δικτύου	113

Κατάλογος Πινάκων

2.1	Συντελεστές συναρτήσεων μορφής	17
2.2	Βάρη και συντεταγμένες σημείων Gauss για ολοκλήρωση ενός και τεσσάρων σημείων ολοκλήρωσης Gauss	20
3.1	H bulk data entry κάρτα PCOMP του Optistruct	71
5.1	Σύνοψη των αποτελεσμάτων βελτιστοποίησης	115
5.2	Σύγκριση των αποτελεσμάτων των μεθόδων βελτιστοποίησης	115

Κεφάλαιο 1

Εισαγωγή

Η εισαγωγική ενότητα περιγράφει τον στόχο της διπλωματικής εργασίας. Επίσης, καθορίζει τα όρια και το πεδίο εφαρμογής της. Τέλος, παρουσιάζει το κίνητρο πίσω από αυτήν την ερευνητική εργασία.

1.1 Διατύπωση Προβλήματος

Η αεροελαστικότητα είναι ένας κλάδος της φυσικής και της μηχανικής που μελετά την απόκριση ελαστικών σωμάτων όταν εκτίθενται σε ροή ρευστού. Οι δυνάμεις που εμπλέκονται σε αυτήν την αλληλεπίδραση είναι η αδρανειακή, η ελαστική και η αεροδυναμική. Τα αεροελαστικά προβλήματα στη μηχανική μπορούν να ταξινομηθούν σε δύο βασικές κατηγορίες. Την στατική αεροελαστικότητα, η οποία εξετάζει την απόκριση σε κατάσταση μόνιμης ροής, και τη δυναμική αεροελαστικότητα, η οποία επικεντρώνεται κυρίως στη δονητική απόκριση του σώματος.

Τα πιο συνηθισμένα αεροελαστικά φαινόμενα που παρατηρούνται στα αεροσκάφη είναι:

- *Aerodynamic Divergence*, όπου η εκτροπή των ανυψωτικών επιφανειών ενός αεροσκάφους προκαλεί επιπλέον άνωση, η οποία με τη σειρά της οδηγεί σε περαιτέρω εκτροπή προς την ίδια κατεύθυνση, με αποτέλεσμα την εμφάνιση υπερβολικών τάσεων ή ακόμη και την αστοχία της κατασκευής.
- *Aeroelastic Control reversal*, όπου οι δυνάμεις που παράγονται από το πηδάλιο ελέγχου (το οποίο είναι υπεύθυνο για τον έλεγχο του αεροσκάφους στον διαμήκη του άξονα) είναι αρκετές για να προκαλέσουν στρέβλωση της πτέρυγας σε τέτοιο βαθμό ώστε να αλλάξουν τα χαρακτηριστικά άντωσης της πτέρυγας, καθιστώντας τις επιφάνειες ελέγχου αναποτελεσματικές ή ακόμη και παράγοντας το αντίθετο από το επιθυμητό αποτέλεσμα.
- *Aeroelastic Flutter* είναι μια δυναμική αστάθεια μιας κατασκευής που προκύπτει λόγω της αλληλεπίδρασης της ροής του ρευστού με τις ιδιομορφές της κατασκευής.

Στην παρούσα διπλωματική εργασία θα μελετηθούν τα χαρακτηριστικά ταλάντωσης (flutter) μιας ανυψωτικής επιφάνειας και θα προσαρμοστούν σε συγκεκριμένες απαιτήσεις με τη χρήση τεχνικών βελτιστοποίησης.

1.2 Στόχος

Ο στόχος αυτού του έργου είναι να αναπτυχθεί κατανόηση των χαρακτηριστικών του αεροελαστικού πτερυγισμού μιας ανυψωτικής επιφάνειας κατασκευασμένης από σύνθετο υλικό και των μεθόδων που χρησιμοποιούνται για την υπολογιστική πρόβλεψη της περιοχής πτερυγισμού χρησιμοποιώντας τον επιλυτή Optistruct της Altair. Στη συνέχεια, μελετάται η επίδραση καποιων κατασκευαστικών παραμέτρων για να προσδιοριστεί η επίδρασή τους στα χαρακτηριστικά του πτερυγισμού, και εξερευνείται η αποτελεσματικότητα διαφόρων τεχνικών βελτιστοποίησης για την προσαρμογή των χαρακτηριστικών του πτερυγισμού σε ένα σύνολο απαιτήσεων με τη χρήση Python.

1.3 Πεδίο Εφαρμογής και Περιορισμοί

Το έργο θα περιλαμβάνει:

- Μελέτη των σύνθετων υλικών και της υλοποίησής τους στον επιλυτή Optistruct.
- Μελέτη της μεθόδου Vortex Lattice και της χρήσης της κατά την ανάλυση αεροδυναμικού πτερυγισμού.
- Μελέτη της ανάλυσης αεροελαστικού πτερυγισμού σε θεωρητικό επίπεδο μέσω των εξισώσεων κίνησης και σε υπολογιστικό επίπεδο μελετώντας την εφαρμογή της θεωρίας σε εμπορικούς επιλυτές.
- Διερεύνηση της επίδρασης των ιδιοτήτων των σύνθετων υλικών στα χαρακτηριστικά του πτερυγισμού.
- Διερεύνηση διαφόρων τεχνικών βελτιστοποίησης για την προσαρμογή των χαρακτηριστικών του πτερυγισμού. Οι τεχνικές αυτές περιλαμβάνουν line search μεθόδους, γενετικούς αλγόριθμους και νευρωνικά δίκτυα.

1.4 Κίνητρο

Στη βιομηχανία της αεροδιαστημικής, η ελαχιστοποίηση του βάρους των κατασκευών έχει εξαιρετική σημασία, καθώς επιτρέπει την αύξηση του χρήσιμου φορτίου, βελτιώνει την αποδοτικότητα, τη ευελιξία και άλλα χαρακτηριστικά ελέγχου. Μία συνέπεια της ελαχιστοποίησης του βάρους των κατασκευών είναι η μείωση των συντελεστών ασφαλείας σε σύγκριση με άλλους τομείς της μηχανικής, γεγονός που δημιουργεί την ανάγκη για πιο ακριβείς υπολογισμούς για να εξασφαλιστεί η ασφάλεια.

Καθώς το κόστος των δοκιμών πρωτοτύπων στον τομέα της αεροδιαστημικής παραμένει υψηλό, οι υπολογιστικές προσομοιώσεις χρησιμοποιούνται ολοένα και περισσότερο και εξελίσουν τις δυνατότητές τους να μελετούν ευρύτερο φάσμα φαινομένων. Η παρούσα διπλωματική διατριβή θα βοηθήσει φοιτητές και τους ερευνητές να κατανήσουν τις βασικές αρχές της ανάλυσης ευστάθειας των πτερύγων, (wing flutter), και θα αναπτύξει τη διαδικασία βελτιστοποίησης για τέτοιες κατασκευές. Ο κώδικας που θα αναπτυχθεί σε αυτή την έργασία μπορεί επίσης να επεκταθεί και να τροποποιηθεί από οποιονδήποτε αναλάβει μια παρόμοια έργασία.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Σε αυτό το κεφάλαιο παρουσιάζεται η βασική θεωρία που θα χρησιμοποιηθεί κατά τη διάρκεια αυτού του έργου. Το κεφάλαιο αυτό χωρίζεται σε τέσσερις κύριες ενότητες. Οι τρεις πρώτες ενότητες αφορούν στον υπολογισμό των χαρακτηριστικών του αεροελαστικού πτερυγισμού μιας ανυψωτικής επιφάνειας, ενώ η τέταρτη ενότητα παρουσιάζει τις διάφορες τεχνικές βελτιστοποίησης που θα χρησιμοποιηθούν.

2.1 Σύνθετα Πεπερασμένα Στοιχεία

Η θεωρία πίσω από τα σύνθετα στοιχεία αναπτύσσεται σύμφωνα με τον E. Oñate, στο έργο του “Structural Analysis with the Finite Element Method” [11], το οποίο παρέχει τις βασικές εξισώσεις και υποθέσεις που απαιτούνται για την ανάπτυξη 2Δ στοιχείων (shell elements) τα οποία λαμβάνουν υπόψη την επίδραση των σύνθετων υλικών με πολλαπλά στρώματα. Το σύστημα συντεταγμένων που θα χρησιμοποιηθεί για την ανάλυση αυτών των στοιχείων φαίνεται στο Σχήμα 2.1

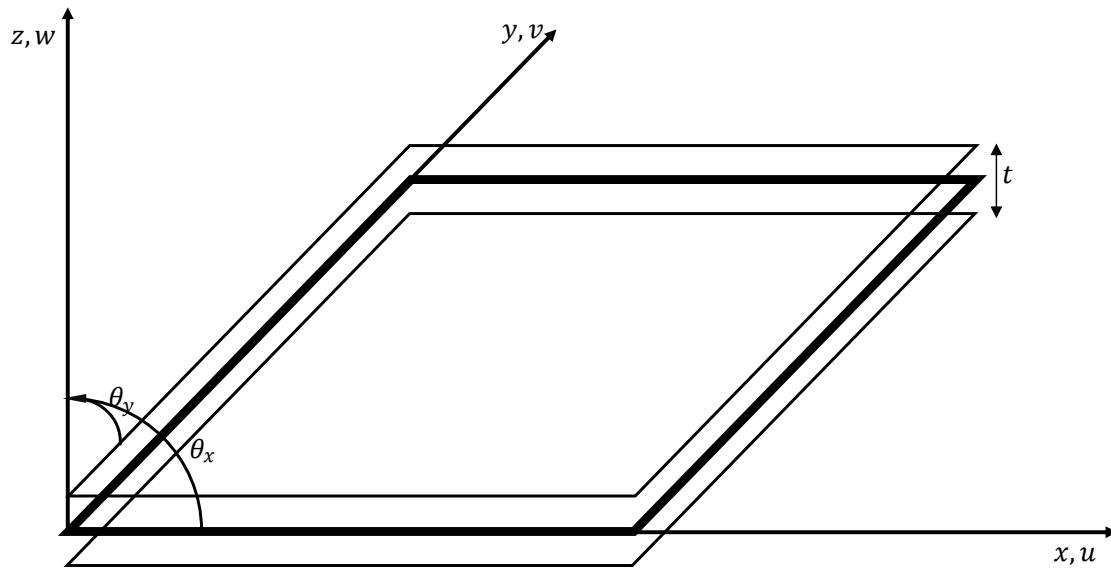
2.1.1 Πεδίο Μετατοπίσεων

Όταν μελετώνται τα σύνθετα στοιχεία πλακών με στρώματα, το κύριο πρόβλημα που προκύπτει είναι ότι, σε αντίθεση με τα ομογενή υλικά, σημεία που ανήκουν στο μέσο επίπεδο του στοιχείου μπορούν να μετακινηθούν «εντός του επιπέδου», κάτι που οδηγεί σε αξονικές δυνάμεις που δεν προκύπτουν σε ένα ομογενές υλικό. Για να ληφθεί υπόψη αυτό, εισάγουμε δύο αξονικές μετατοπίσεις $u_0(x, y)$ και $v_0(x, y)$ και έτσι η μετατόπιση οποιουδήποτε σημείου εντός της πλάκας μπορεί να υπολογιστεί ως εξής:

$$u(x, y, z) = u_0(x, y) - z\theta_x(x, y) \quad (2.1a)$$

$$v(x, y, z) = v_0(x, y) - z\theta_y(x, y) \quad (2.1b)$$

$$w(x, y, z) = w_0(x, y) \quad (2.1c)$$



Σχήμα 2.1: Σύστημα συντεταγμένων σύνθετου στοιχείου

2.1.2 Διανύσματα Παραμορφώσεων

Η παραμόρφωση εντός της πλάκας μπορεί να υπολογιστεί ως εξής:

$$\begin{aligned}
\boldsymbol{\epsilon} &= \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial v_0}{\partial y} \\ \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -z \frac{\partial \theta_x}{\partial x} \\ -z \frac{\partial \theta_y}{\partial y} \\ -z \left(\frac{\partial \theta_x}{\partial y} + \frac{\partial \theta_y}{\partial x} \right) \\ \frac{\partial w_0}{\partial x} - \theta_x \\ \frac{\partial w_0}{\partial y} - \theta_y \end{bmatrix} \\
&= \begin{bmatrix} \hat{\boldsymbol{\epsilon}}_m \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -z \cdot \hat{\boldsymbol{\epsilon}}_b \\ \hat{\boldsymbol{\epsilon}}_s \end{bmatrix} = \mathbf{S} \cdot \hat{\boldsymbol{\epsilon}}
\end{aligned} \tag{2.2}$$

Όπου:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \hat{\boldsymbol{\epsilon}}_m \\ \hat{\boldsymbol{\epsilon}}_b \\ \hat{\boldsymbol{\epsilon}}_s \end{bmatrix}, \quad \hat{\boldsymbol{\epsilon}}_m = \begin{bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial v_0}{\partial y} \\ \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \end{bmatrix}, \quad \hat{\boldsymbol{\epsilon}}_b = \begin{bmatrix} \frac{\partial \theta_x}{\partial x} \\ \frac{\partial \theta_y}{\partial y} \\ \frac{\partial \theta_x}{\partial y} + \frac{\partial \theta_y}{\partial x} \end{bmatrix}, \quad \hat{\boldsymbol{\epsilon}}_s = \begin{bmatrix} \frac{\partial w_0}{\partial x} - \theta_x \\ \frac{\partial w_0}{\partial y} - \theta_y \end{bmatrix}$$

είναι τα γενικευμένα διανύσματα τάσεων που οφείλονται σε παραμορφώσεις μεμβράνης (m), κάμψης (b) και εγκάρσιες παραμορφώσεις (s).

$$\mathbf{S} = \begin{bmatrix} I_3 & -zI_3 & O_2 \\ O_3^T & O_2 & I_2 \end{bmatrix} \tag{2.3}$$

Όπου, I_n είναι ο $n \times n$ μοναδιαίος πίνακας, και

$$O_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad O_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

από τη σχέση που προηγείται, μπορούμε να εξάγουμε τις εξής χρήσιμες εκφράσεις:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_p \\ \boldsymbol{\epsilon}_s \end{bmatrix} \quad (2.4)$$

Όπου: $\boldsymbol{\epsilon}_p = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix} = \hat{\boldsymbol{\epsilon}}_m - z \hat{\boldsymbol{\epsilon}}_b$, $\boldsymbol{\epsilon}_s = \begin{bmatrix} \gamma_{xy} \\ \gamma_{yz} \end{bmatrix} = \hat{\boldsymbol{\epsilon}}_s$

2.1.3 Σχέση Τάσεων - Παραμορφώσεων

Τώρα θα εξαχθεί η σχέση τάσεων-παραμορφώσεων σε μια σύνθετη πλάκα με στρώματα.

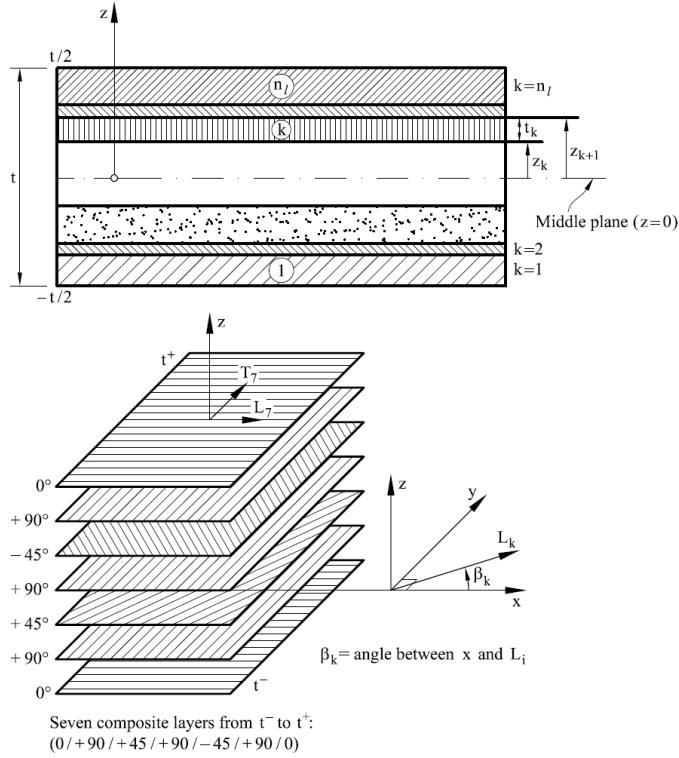
Θεωρούμε μια σύνθετη πλάκα με στρώματα που σχηματίζεται από την τοποθέτηση n_l ορθότροπων στρώσεων όπως φαίνεται στο Σχήμα 2.2, οι οποίες ονομάζονται plies, με άξονες ορθοτροπίας L, T, z και ισοτροπία στον άξονα L (στο επίπεδο Tz). Ο άξονας L είναι παράλληλος προς την κατεύθυνση των επιμήκων ινών του σύνθετου υλικού.

Θα υποτεθεί ότι:

- Κάθε στρώση (k) ορίζεται από το επίπεδο $z = z_k$ και $z = z_{k+1}$ με $z_k \leq z \leq z_{k+1}$.
- Οι άξονες ορθοτροπίας L και T μπορεί να μεταβάλλονται για κάθε στρώση και εκπροσωπούνται από τη γωνία β_i , η οποία ορίζεται ως η γωνία μεταξύ του global άξονα x και του άξονα L_i της i -οστής στρώσης.
- Κάθε στρώση ικανοποιεί την υπόθεση για την παραμόρφωση στο επίπεδο, δηλαδή $\sigma_z = 0$ και ο άξονας z είναι ο άξονας ορθοτροπίας που είναι κοινός για όλες τις στρώσεις.
- Το πεδίο μετατοπίσεων είναι συνεχές μεταξύ των στρώσεων και ικανοποιεί την εξίσωση (2.1).

Οι παραδοχές που αναφέρθηκαν παραπάνω, μας επιτρέπουν να εκφράσουμε τη σχέση μεταξύ των τάσεων εντός του επιπέδου $\sigma_x, \sigma_y, \tau_{xy}$ και των εγκάρσιων διατμητικών παραμορφώσεων τ_{xz}, τ_{yz} με τις αντίστοιχες παραμορφώσεις για κάθε στρώση k ως εξής:

$$\boldsymbol{\sigma}_p = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \mathbf{D}_p \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \mathbf{D}_p \boldsymbol{\epsilon}_p \quad (2.5)$$



Σχήμα 2.2: Ορισμός στρωμάτων σε composite laminated plate [11]

$$\boldsymbol{\sigma}_s = \begin{bmatrix} \tau_{xz} \\ \tau_{yz} \end{bmatrix} = \mathbf{D}_s \begin{bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \mathbf{D}_s \boldsymbol{\epsilon}_s \quad (2.6)$$

Και τέλος,

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_p \\ \boldsymbol{\sigma}_s \end{bmatrix} = \begin{bmatrix} \mathbf{D}_p & 0 \\ 0 & \mathbf{D}_s \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\epsilon}_p \\ \boldsymbol{\epsilon}_s \end{bmatrix} = \mathbf{D} \boldsymbol{\epsilon} \quad (2.7)$$

Τα καταστατικά μητρώα (constitutive matrices) \mathbf{D}_p , \mathbf{D}_s είναι συμμετρικά και τα στοιχεία τους εξαρτώνται από τις πέντε ανεξάρτητες ιδιότητες του υλικού καθώς και από τη γωνία β_k . Ο υπολογισμός των προαναφερθέντων μητρώων ξεκινά εκφράζοντας τις σχέσεις τάσεων-παραμορφώσεων στους άξονες ορθοτροπίας L, T, z .

$$\sigma_1 = \mathbf{D}_1 \epsilon_1, \quad \sigma_2 = \mathbf{D}_2 \epsilon_2 \quad (2.8)$$

Όπου:

$$\sigma_1 = \begin{bmatrix} \sigma_L \\ \sigma_T \\ \tau_{LT} \end{bmatrix}, \quad \epsilon_1 = \begin{bmatrix} \epsilon_L \\ \epsilon_T \\ \gamma_{LT} \end{bmatrix}, \quad \mathbf{D}_1 = \begin{bmatrix} D_{LL} & D_{LT} & 0 \\ D_{TT} & 0 \\ Sym. & G_{LT} \end{bmatrix}$$

$$\sigma_2 = \begin{bmatrix} \tau_{Lz} \\ \tau_{Tz} \end{bmatrix}, \quad \epsilon_2 = \begin{bmatrix} \gamma_{Lz} \\ \gamma_{Tz} \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} G_{Lz} & 0 \\ 0 & G_{Tz} \end{bmatrix}$$

$$D_{LL} = \frac{E_L}{a}, \quad D_{TT} = \frac{E_T}{a}, \quad a = 1 - \nu_{LT}\nu_{TL}, \quad D_{LT} = \frac{E_T \nu_{LT}}{a}$$

Οι πέντε πιο συχνά χρησιμοποιούμενες ανεξάρτητες παράμετροι του υλικού είναι:

$$E_L, \quad E_T, \quad \nu_{LT} \left(or \nu_{TL} = \frac{E_T}{E_L} \nu_{LT} \right), \quad G_{Lz} = G_{LT}, \quad G_{Tz}$$

Τέλος, η σχέση μεταξύ των μητρώων D_1 , D_2 και D_p , D_s μπορεί να εκφραστεί ως ένα απλό γινόμενο μητρώο με τους πίνακες μετασχηματισμού T_1 και T_2 ως εξής:

$$\mathbf{D}_p = T_1^T \cdot \mathbf{D}_1 \cdot T_1, \quad \mathbf{D}_s = T_2^T \cdot \mathbf{D}_2 \cdot T_2 \quad (2.9)$$

Όπου:

$$T_1 = \begin{bmatrix} \cos^2(\beta_k) & \sin^2(\beta_k) & \cos(\beta_k) \sin(\beta_k) \\ \sin^2(\beta_k) & \cos^2(\beta_k) & -\cos(\beta_k) \sin(\beta_k) \\ -2\cos(\beta_k) \sin(\beta_k) & 2\cos(\beta_k) \sin(\beta_k) & \cos^2(\beta_k) - \sin^2(\beta_k) \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \cos(\beta_k) & \sin(\beta_k) \\ -\sin(\beta_k) & \cos(\beta_k) \end{bmatrix}$$

2.1.4 Γενικευμένο Καταστατικό Μητρώο

$$\hat{\boldsymbol{\sigma}}_m = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{\sigma}_p dz, \quad membrane \ forces \quad (2.10)$$

$$\hat{\boldsymbol{\sigma}}_b = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = - \int_{-\frac{t}{2}}^{\frac{t}{2}} z \boldsymbol{\sigma}_p dz, \quad Bending \ moments \quad (2.11)$$

$$\hat{\boldsymbol{\sigma}}_s = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \int_{-\frac{t}{2}}^{\frac{t}{2}} z \boldsymbol{\sigma}_s dz \quad \text{Transverse shear forces} \quad (2.12)$$

$$\hat{\boldsymbol{\sigma}}_m = \hat{\mathbf{D}}_m \hat{\epsilon}_m + \hat{\mathbf{D}}_{mb} \hat{\epsilon}_b \quad (2.13)$$

$$\hat{\boldsymbol{\sigma}}_b = \hat{\mathbf{D}}_{mb} \hat{\epsilon}_m + \hat{\mathbf{D}}_b \hat{\epsilon}_b \quad (2.14)$$

$$\hat{\boldsymbol{\sigma}}_s = \hat{\mathbf{D}}_s \hat{\epsilon}_s \quad (2.15)$$

Όπου:

$$\hat{\mathbf{D}}_m = \int_{-t/2}^{t/2} \mathbf{D}_p dz \quad (2.16)$$

$$\hat{\mathbf{D}}_{mb} = \int_{-t/2}^{t/2} z \mathbf{D}_p dz \quad (2.17)$$

$$\hat{\mathbf{D}}_b = \int_{-t/2}^{t/2} z^2 \mathbf{D}_p dz \quad (2.18)$$

$$\hat{\mathbf{D}}_s = \begin{bmatrix} k_{11} \bar{D}_{s_{11}} & k_{12} \bar{D}_{s_{12}} \\ \Sigma \mu. & k_{22} \bar{D}_{s_{22}} \end{bmatrix}, \quad \mu \epsilon \quad \bar{D}_{s_{ij}} = \int_{-\frac{t}{2}}^{\frac{t}{2}} D_{s_{ij}} dz \quad (2.19)$$

Στην εξίσωση 2.19, οι όροι k_{ij} ονομάζονται συντελεστές διόρθωσης για τη διάτμηση.

Τα ολοκληρώματα στις εξισώσεις 2.16 - 2.19 μπορούν να εκφραστούν ως πεπερασμένο άθροισμα όταν η σύνθετη πλάκα laminate αποτελείται από n_l ορθότροπες στρώσεις. Για σύνθετες πλάκες, όπου οι x και y ταυτίζονται με τους άξονες ορθοτροπίας για όλες τις στρώσεις, ο πίνακας $\hat{\mathbf{D}}_s$ είναι διαγώνιος και μόνο οι όροι k_{11} και k_{22} χρειάζεται να υπολογιστούν.

Ο απλούστερος τρόπος υπολογισμού των συντελεστών διόρθωσης διάτμησης είναι η υπόθεση κυλινδρικής κάμψης. Αυτό σημαίνει ότι:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} = 0 \quad (2.20)$$

Υποθέτοντας περαιτέρω μια σταθερή κατανομή της εγκάρσιας διατμητικής τάσης στην κατεύθυνση του πάχους και μετά από μερικούς υπολογισμούς, τελικά προκύπτει:

$$k_{11} = \hat{D}_{b_{11}}^2 \left[\bar{G}_{xz} \int_{-t/2}^{t/2} \frac{g_1^2(z)}{G_{xz}} dz \right]^{-1} \quad (2.21)$$

$$k_{22} = \hat{D}_{b22}^2 \left[\bar{G}_{yz} \int_{-t/2}^{t/2} \frac{g_2^2(z)}{G_{yz}} dz \right]^{-1} \quad (2.22)$$

Όπου:

$$\begin{aligned} g_{1(z)} &= \int_{-t/2}^z z D_{p11} dz \\ g_{2(z)} &= \int_{-t/2}^z z D_{p22} dz \\ \bar{G}_{xz} &= \int_{-t/2}^{t/2} G_{xz} dz \\ \bar{G}_{yz} &= \int_{-t/2}^{t/2} G_{yz} dz \end{aligned}$$

Οι εξισώσεις (2.16) έως (2.19) μπορούν να εκφραστούν ως πεπερασμένο άθροισμα όταν το σύνθετο laminate αποτελείται από n_l ορθότροπες στρώσεις, εντός των οποίων οι ιδιότητες του υλικού είναι σταθερές.

$$\hat{\mathbf{D}}_m = \sum_{k=1}^{n_l} t_k \mathbf{D}_{pk} \quad (2.23)$$

$$\hat{\mathbf{D}}_{mb} = - \sum_{k=1}^{n_l} t_k \bar{z}_k \mathbf{D}_{pk} \quad (2.24)$$

$$\hat{\mathbf{D}}_b = \sum_{k=1}^{n_l} \frac{1}{3} (\bar{z}_{k+1}^3 - z_k^3) \mathbf{D}_{pk} \quad (2.25)$$

$$\hat{\mathbf{D}}_s = \begin{bmatrix} \tilde{k}_{11} & \sum_{k=1}^{n_l} t_k \mathbf{D}_{sk_{11}} & 0 \\ 0 & \tilde{k}_{22} \sum_{k=1}^{n_l} t_k \mathbf{D}_{sk_{22}} & \end{bmatrix} \quad (2.26)$$

Όπου:

$$t_k = z_{k+1} - z_k, \quad \bar{z}_k = \frac{1}{2} (z_{k+1} + z_k)$$

$$\tilde{k}_{ii} = \hat{\mathbf{D}}_{b_{ii}} \cdot \left[\sum_{k=1}^{n_l} t_k G_{xz} \cdot \sum_{k=1}^{n_l} \left(\frac{\sum_{l=1}^k (t_k \bar{z}_k \mathbf{D}_{p_{il}})}{G_{xz_k}} t_k \right) \right], \quad i = 1, 2$$

2.1.5 Διακριτοποιημένη τάση και παραμόρφωση - Συναρτήσεις Μορφής

Οι μετατοπίσεις εντός ενός σύνθετου τετραγωνικού στοιχείου τεσσάρων κόμβων μπορούν να εκφραστούν μέσω των συναρτήσεων μορφής.

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \end{bmatrix} = \sum_{i=1}^4 N_i a_i = \mathbf{N} \cdot \vec{\mathbf{a}} \quad (2.27)$$

Όπου:

$$\mathbf{N} = [\mathbf{N}_1 \mid \mathbf{N}_2 \mid \mathbf{N}_3 \mid \mathbf{N}_4]$$

$$\mathbf{N}_i = \begin{bmatrix} N_i & 0 & 0 & 0 & 0 \\ 0 & N_i & 0 & 0 & 0 \\ 0 & 0 & N_i & 0 & 0 \\ 0 & 0 & 0 & N_i & 0 \\ 0 & 0 & 0 & 0 & N_i \end{bmatrix}, \quad \text{for } i \in \{1, 2, 3, 4\}$$

$$\vec{a} = \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \\ \vec{a}_4 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \theta_{x_1} \\ \theta_{y_1} \\ u_2 \\ v_2 \\ w_2 \\ \theta_{x_2} \\ \theta_{y_2} \\ u_3 \\ v_3 \\ w_3 \\ \theta_{x_3} \\ \theta_{y_3} \\ u_4 \\ v_4 \\ w_4 \\ \theta_{x_4} \\ \theta_{y_4} \end{bmatrix}$$

Η σχέση τάσεων-παραμορφώσεων χρησιμοποιώντας την εξίσωση (2.2) μπορεί τώρα να εκφραστεί μέσω των συναρτήσεων μορφής:

$$\epsilon = \begin{bmatrix} \hat{\epsilon}_m \\ \hat{\epsilon}_b \\ \hat{\epsilon}_s \end{bmatrix} = \begin{bmatrix} \frac{\partial u_0}{\partial x} \\ \frac{\partial v_0}{\partial y} \\ \frac{\partial u_0}{\partial y} + \frac{\partial v_0}{\partial x} \\ \frac{\partial \theta_x}{\partial x} \\ \frac{\partial \theta_y}{\partial y} \\ \frac{\partial \theta_x}{\partial y} + \frac{\partial \theta_y}{\partial x} \\ \frac{\partial w_0}{\partial x} - \theta_x \\ \frac{\partial w_0}{\partial y} - \theta_y \end{bmatrix} = \sum_{i=1}^4 \begin{bmatrix} \frac{\partial N_i}{\partial x} u_i \\ \frac{\partial N_i}{\partial x} v_i \\ \frac{\partial N_i}{\partial y} u_i + \frac{\partial N_i}{\partial x} v_i \\ \frac{\partial N_i}{\partial x} \theta_{x_i} \\ \frac{\partial N_i}{\partial y} \theta_{y_i} \\ \frac{\partial N_i}{\partial y} \theta_{x_i} + \frac{\partial N_i}{\partial x} \theta_{y_i} \\ \frac{\partial N_i}{\partial x} w_i - N_i \theta_{x_i} \\ \frac{\partial N_i}{\partial y} w_i - N_i \theta_{y_i} \end{bmatrix} = [\mathbf{B}_1 \mid \mathbf{B}_2 \mid \mathbf{B}_3 \mid \mathbf{B}_4] \cdot \vec{a} = \mathbf{B} \cdot \vec{a} \quad (2.28)$$

Όπου:

$$\mathbf{B}_i = \begin{bmatrix} \mathbf{B}_{m_i} \\ \mathbf{B}_{b_i} \\ \mathbf{B}_{s_i} \end{bmatrix}$$

$$\mathbf{B}_{m_i} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 & 0 & 0 \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{b_i} = \begin{bmatrix} 0 & 0 & 0 & -\frac{\partial N_i}{\partial x} & 0 \\ 0 & 0 & 0 & 0 & -\frac{\partial N_i}{\partial y} \\ 0 & 0 & 0 & -\frac{\partial N_i}{\partial y} & -\frac{\partial N_i}{\partial x} \end{bmatrix},$$

$$\mathbf{B}_{s_i} = \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial x} & -N_i & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial y} & 0 & -N_i \end{bmatrix}$$

Οι συναρτήσεις μορφής που χρησιμοποιούνται ποικίλουν, αλλά η πιο απλή περίπτωση είναι η διγραμμική, η οποία χρησιμοποιεί γραμμικές συναρτήσεις μορφής για την παρεμβολή. Όλοι οι υπολογισμοί που περιγράφηκαν στο προηγούμενο κεφάλαιο πραγματοποιούνται σε ένα μετασχηματισμένο χώρο συντεταγμένων, όπου κάθε στοιχείο είναι ένα τέλειο τετράγωνο με μήκος πλευράς δύο μονάδες μήκους. Αυτός ο μετασχηματισμένος χώρος ονομάζεται “Natural” χώρος. Αντιθέτως ο πραγματικός (“Physical”) χώρος αντιπροσωπεύει την πραγματική γεωμετρία του στοιχείου. Μια απεικόνιση αυτών των χώρων γίνεται στο Σχήμα 2.3. Οι συναρτήσεις μορφής είναι:

$$N_i(\xi, \eta) = \frac{1}{4}(1 + a_i \cdot \xi)(1 + b_i \cdot \eta) \quad (2.29)$$

$$\frac{\partial N_i(\xi, \eta)}{\partial \xi} = \frac{1}{4}a_i(1 + b_i\eta) \quad (2.30)$$

$$\frac{\partial N_i(\xi, \eta)}{\partial \eta} = \frac{1}{4}b_i(1 + a_i\xi) \quad (2.31)$$

Στόν Πίνακα 2.1 παρουσιάζονται οι συντελεστές των συναρτήσεων μορφής.

i	1	2	3	4
a_i	-1	1	1	-1
b_i	-1	-1	1	1

Πίνακας 2.1: Συντελεστές συναρτήσεων μορφής

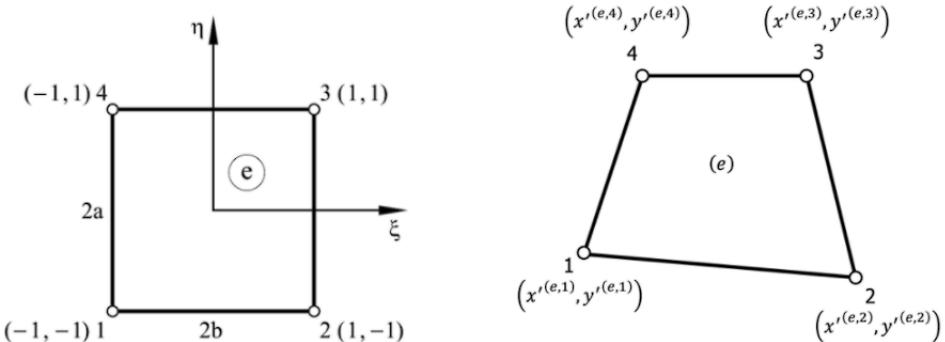
Η Ιακωβιανή αυτού του μετασχηματισμού από τον “Physical” στον “Natural” χώρο ορίζεται ως:

$$\mathcal{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (2.32)$$

Οι παράγωγοι των συναρτήσεων μορφής στον “Physical” χώρο μπορούν στη συνέχεια να υπολογιστούν ως:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathcal{J}^{-1} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (2.33)$$

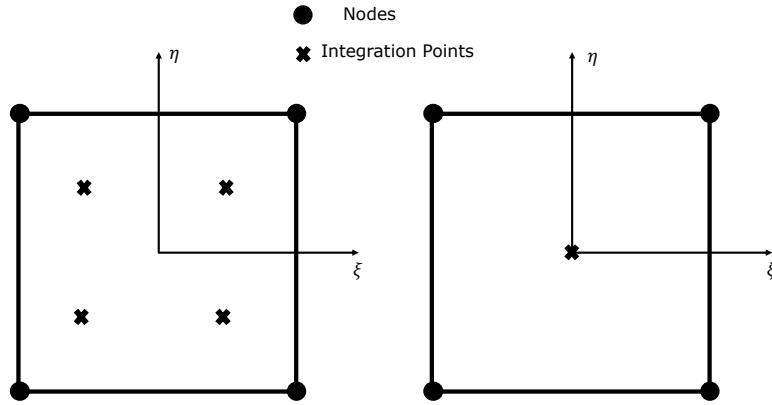
Τέλος, μια ακόμη ιδιότητα της Ιακωβιανής είναι ότι η ορίζουσά της αντιπροσωπεύει την κλίμακα του μετασχηματισμού.



Σχήμα 2.3: natural and physical coordinate space of quadrilateral plate element [1]

2.1.6 Μητρώο Στιβαρότητας

Το τελικό βήμα στον υπολογισμό των πεπερασμένων στοιχείων είναι ο υπολογισμός και η συναρμολόγηση του μητρώου δυσκαμψίας. Χρησιμοποιώντας την Αρχή του Εικονικού Έργου με το συνήθη τρόπο, τα τοπικά μητρώα στιβαρότητας μπορούν να γραφούν ως:



Σχήμα 2.4: Σημεία Gauss για πλήρη και μειωμένη ολοκλήρωση στοιχείων τεσσάρων κόμβων

$$\mathbf{K}_{m_{ij},[20 \times 20]} = \iint_{Area} \mathbf{B}_{m,i}^T \hat{\mathbf{D}}_m \mathbf{B}_{m,j} dA, \quad \text{membrane stiffness} \quad (2.34)$$

$$\mathbf{K}_{b_{ij},[20 \times 20]} = \iint_{Area} \mathbf{B}_{b,i}^T \hat{\mathbf{D}}_b \mathbf{B}_{b,j} dA, \quad \text{bending stiffness} \quad (2.35)$$

$$\mathbf{K}_{s_{ij},[20 \times 20]} = \iint_{Area} \mathbf{B}_{s,i}^T \hat{\mathbf{D}}_s \mathbf{B}_{s,j} dA, \quad \text{shear stiffness} \quad (2.36)$$

$$\mathbf{K}_{mb_{ij},[20 \times 20]} = \iint_{Area} \left(\mathbf{B}_{m,i}^T \hat{\mathbf{D}}_{mb} \mathbf{B}_{b,j} + \mathbf{B}_{b,i}^T \hat{\mathbf{D}}_{mb} \mathbf{B}_{m,j} \right) dA, \quad \text{membrane-bending stiffness} \quad (2.37)$$

Όλα τα ολοκληρώματα στις εξισώσεις (2.34) έως (2.37) υπολογίζονται χρησιμοποιώντας τη μέθοδο ολοκλήρωσης Gauss. Η πλήρης ολοκλήρωση Gauss για το στοιχείο πλάκας τεσσάρων κόμβων που αναπτύχθηκε, περιλαμβάνει τέσσερα σημεία ολοκλήρωσης, ενώ η μειωμένη μορφή ολοκλήρωσης αυτών των στοιχείων απαιτεί μόνο ένα σημείο Gauss όπως φαίνεται στο Σχήμα 2.4

Σύμφωνα με την ολοκλήρωση Gauss για δισδιάστατους τομείς χρησιμοποιώντας n σημεία Gauss :

$$\int_{-1}^1 \int_{-1}^1 f(x, y) dx dy \approx \sum_{j=1}^n \sum_{i=1}^n w_i w_j f(x_i, y_i), \text{ for every } i, j \leq n \quad (2.38)$$

Δεδομένου ότι το στοιχείο πλάκας έχει τέσσερις κόμβους, είναι δυνατές μόνο δύο ολοκληρώσεις:

- Χρησιμοποιώντας ένα σημείο Gauss, που οδηγεί στο σχήμα μειωμένης ολοκλήρωσης.
- Χρησιμοποιώντας τέσσερα σημεία Gauss, που οδηγεί στο πλήρες σχήμα ολοκλήρωσης.

Η χρήση πλήρους ολοκλήρωσης οδηγεί σε μεγαλύτερο υπολογιστικό χρόνο, αλλά βελτιώνει την ακρίβεια. Αντιθέτως, η μειωμένη ολοκλήρωση επιταχύνει τον υπολογισμό, απαιτώντας μόνο το ένα τέταρτο των πράξεων, αλλά οδηγεί σε λιγότερο ακριβή αποτέλεσματα και εισάγει τα λεγόμενα μηδενικά ενεργειακά modes, τα οποία είναι παραμορφωμένες καταστάσεις του στοιχείου με μηδενική ενέργεια παραμόρφωσης. Αυτό είναι φυσικά αδύνατο και μειώνει τη δυσκαμψία της δομής, καθώς για την επίτευξη αυτών των modes παραμόρφωσης δεν απαιτείται καμία ενέργεια. Αυτό το φαινόμενο είναι επίσης γνωστό ως το φαινόμενο της «κλεψύδρας» (hourglass effect) στη μέθοδο των πεπερασμένων στοιχείων (FEM).

Ο Πίνακας 2.2 παρουσιάζει τις συντεταγμένες των σημείων Gauss και τα βάρη που θα χρησιμοποιηθούν κατά την εκτέλεση ολοκλήρωσης με ένα ή τέσσερα σημεία Gauss.

N_G	k	w_k	ξ_k	η_k
1	1	2	0	0
4	1	1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
	2	1	$+\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
	3	1	$+\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$
	4	1	$-\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$

Πίνακας 2.2: Βάρη και συντεταγμένες σημείων Gauss για ολοκλήρωση ενός και τεσσάρων σημείων ολοκλήρωσης Gauss

Χρησιμοποιώντας την ολοκλήρωση Gauss, το τοπικό μητρώο δυσκαμψίας του στοιχείου πλάκας μπορεί να υπολογιστεί ως το άθροισμα όλων των επιμέρους μητρώων δυσκαμψίας:

$$\mathbf{K}_{local} = \mathbf{K}_m + \mathbf{K}_b + \mathbf{K}_s + \mathbf{K}_{mb} \quad (2.39)$$

Το τελικό βήμα για τον υπολογισμό του Global μητρώου στιβαρότητας είναι η μετατροπή του τοπικού συστήματος συντεταγμένων πίσω στο Global σύστημα συντεταγμένων. Τα συστήματα αυτά απεικονίζονται στο Σχήμα 2.5. Η μετατροπή αυτή γίνεται χρησιμοποιώντας τον πίνακα περιστροφής και την ορίζουσα του Ιακωβιανού ως συντελστή, ως ακολούθως:

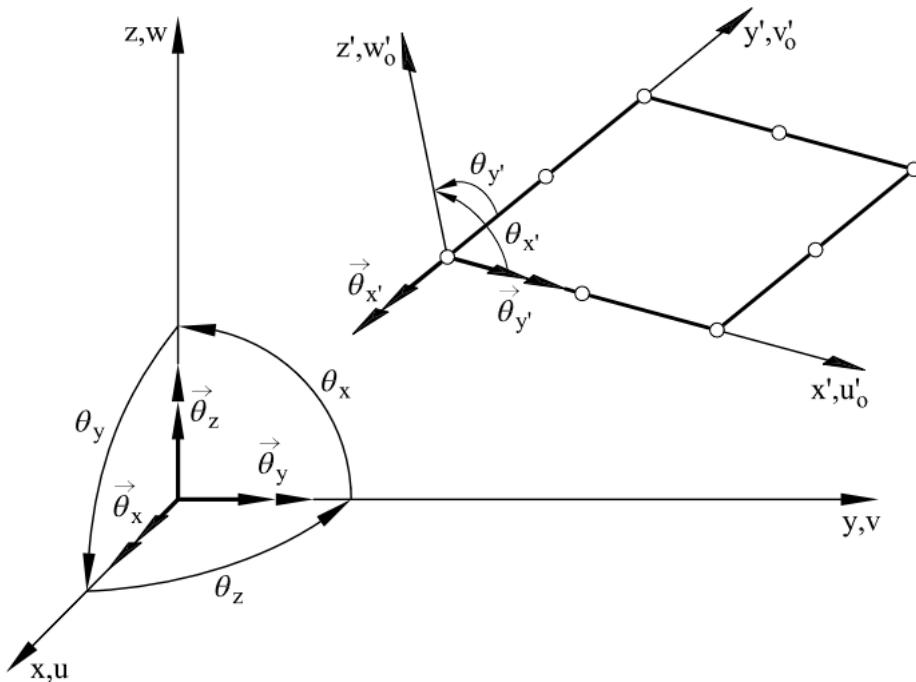
$$\mathbf{K}_{global} = \det(\mathcal{J}) \cdot \mathbf{R}^T \cdot \mathbf{K}_{local} \cdot \mathbf{R} \quad (2.40)$$

Όπου:

$$\mathbf{R}_{5n \times 6n} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_n \end{bmatrix}$$

$$\mathbf{L}_i = \begin{bmatrix} \lambda_{x'x} & \lambda_{x'y} & \lambda_{x'z} & 0 & 0 & 0 \\ \lambda_{y'x} & \lambda_{y'y} & \lambda_{y'z} & 0 & 0 & 0 \\ \lambda_{z'x} & \lambda_{z'y} & \lambda_{z'z} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda_{y'x} & -\lambda_{y'y} & -\lambda_{y'z} \\ 0 & 0 & 0 & \lambda_{x'x} & \lambda_{x'y} & \lambda_{x'z} \end{bmatrix}$$

Με $n = 4$ και $\lambda_{x'x}$ να είναι το συνημίτονο της γωνίας που σχηματίζεται από τους άξονες x' και x , κ.ο.κ.



Σχήμα 2.5: Ορισμός Local και Global συστημάτων συντεταγμένων [11]

Τα στοιχεία πλάκας που αναπτύχθηκαν έχουν μόνο 5 βαθμούς ελευθερίας ανά κόμβο. Για να επιτευχθούν οι πλήρεις έξι βαθμοί ελευθερίας ανά κόμβο, χρειάζεται να προστεθεί ένας όρος δυσκαμψίας στην περιστροφή γύρω από τον άξονα z , ϑ_z . Αυτός είναι ένας πλασματικός όρος δυσκαμψίας που μπορεί να προστεθεί για να φέρει το μητρώο δυσκαμψίας στο πλήρες μέγεθός του, 24×24 . Αυτή η τεχνική χρησιμοποιείται συχνά για να αποφευχθούν ενδεχόμενες μοναδικότητες στο μητρώο δυσκαμψίας.

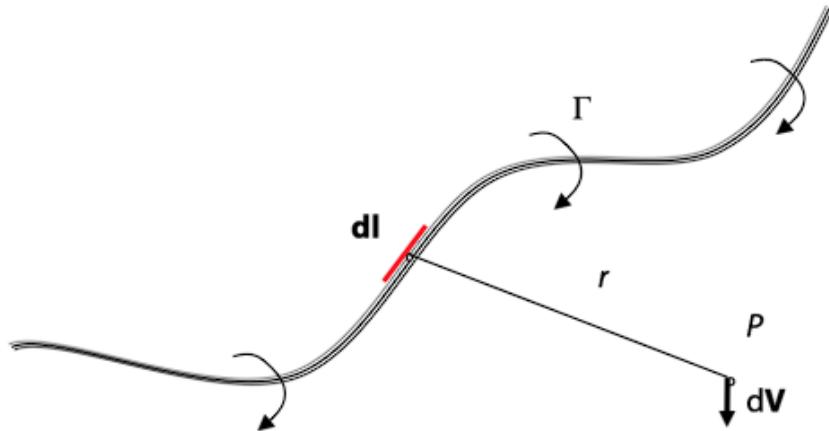
Μια κοινή τεχνική που χρησιμοποιείται για να αποφευχθεί το φαινόμενο του "shear locking" είναι η ολοκλήρωση των επιμέρους μητρώων δυσκαμψίας χρησιμοποιώντας διαφορετικό σχήμα ολοκλήρωσης Gauss.

2.2 Η Μέθοδος Πλέγματος Στροβίλων (VLM)

Η θεωρία της Μεθόδου Πλέγματος Στροβίλων Vortex Lattice Method (VLM) που αναπτύσσεται σε αυτό το κεφάλαιο ακολουθεί τις συμβάσεις του βιβλίου A. P. Joseph Katz, Low-Speed Aerodynamics [7].

2.2.1 Ο νόμος Biot Savart – το Vortex Filament

Στο Σχήμα 2.6 φαίνεται ενα τρισδιάστατο «νήμα» δίνης το οπόιο επάγει ένα πεδίο ταχυτήτων στο χώρο.



Σχήμα 2.6: Curved Three-dimensional vortex filament of strength Γ [12]

Η εξίσωση συνέχειας για ένα ασυμπίεστο ρευστό είναι:

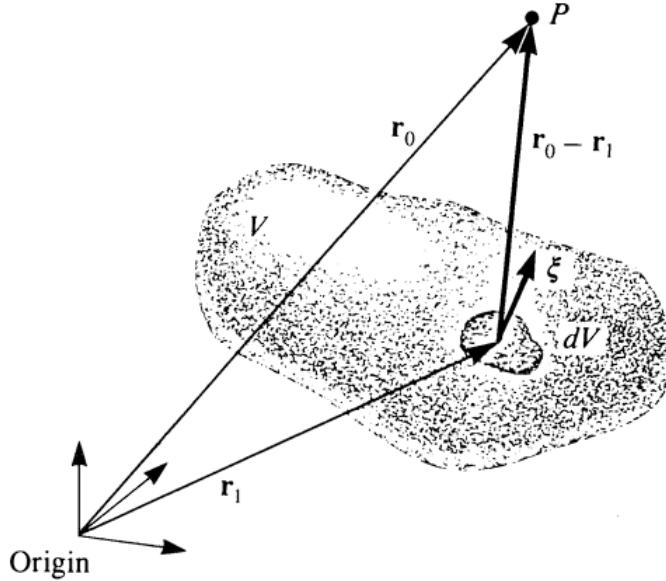
$$\nabla \cdot \vec{V} = 0 \quad (2.41)$$

Το δυναμικό του πεδίου ταχύτητας είναι ένα διανυσματικό πεδίο \vec{B} που ορίζεται από:

$$\nabla \times \vec{B} = \vec{V} \quad (2.42)$$

Η απόκλιση του διανυσματικού πεδίου \vec{B} είναι μηδέν και έτσι το curl αυτού μπορεί να εκφραστεί ως:

$$\zeta = \nabla \times \vec{V} = \nabla \times (\nabla \times \vec{B}) = \cancel{\nabla \cdot \vec{B}} - \nabla^2 \vec{B} = -\nabla^2 \vec{B} \quad (2.43)$$



Σχήμα 2.7: Ταχύτητα σε σημειο P λόγω vortex distribution [7]

Η γενική λύση σε αυτήν την εξίσωση που αντιπροσωπεύει μια γενική κατανομή από δίνες στο χώρο όπως φαίνεται στο Σχήμα 2.7, χρησιμοποιώντας το θεώρημα του Green, είναι:

$$\vec{B} = \frac{1}{4\pi} \int_V \frac{\vec{\zeta}}{|\vec{r}|} dV \quad (2.44)$$

$$\vec{V} = \frac{1}{4\pi} \int_V \nabla \times \frac{\vec{\zeta}}{|\vec{r}|} dV \quad (2.45)$$

Όπου:

$$\vec{r} = \vec{r}_0 - \vec{r}_1$$

Λαμβάνοντας υπόψη ένα απειροστό κομμάτι νηματοειδούς δίνης ζ :

$$d\mathbf{l} = \frac{\zeta}{\zeta} dl, \quad \Gamma = \zeta \cdot dS, \quad dV = dS \cdot dl \quad (2.46)$$

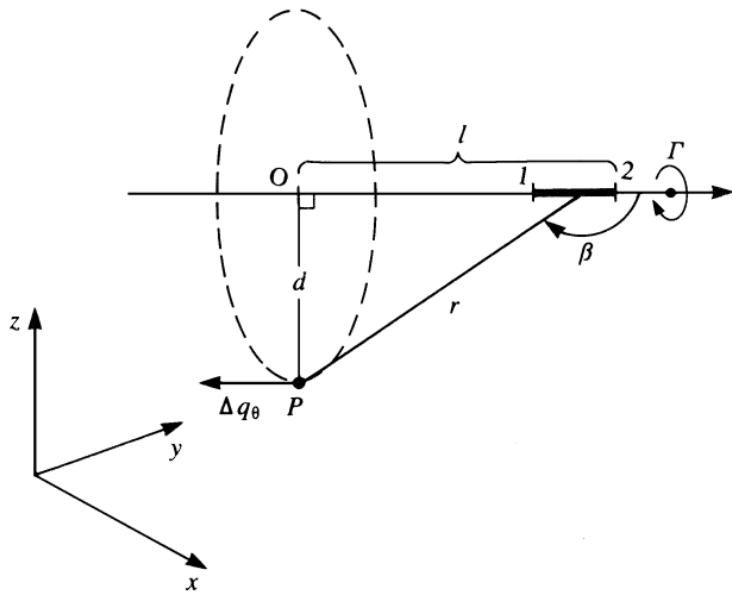
$$\nabla \times \frac{\zeta}{|\vec{r}|} d\vec{V} = \nabla \times \Gamma \frac{d\mathbf{l}}{|\vec{r}|} = \Gamma \frac{d\mathbf{l} \times \vec{r}}{|\vec{r}|^3} \quad (2.47)$$

Η αντικατάσταση της εξίσωσης (2.44) στην εξίσωση (2.47) οδηγεί σε

$$\vec{V} = \frac{1}{4\pi} \int_V \frac{d\mathbf{l} \times \vec{r}}{|\vec{r}|^3} dV \quad (2.48)$$

2.2.2 Ευθύγραμμο Τμήμα Δίνης

Το τμήμα της δίνης τοποθετείται σε αυθαίρετη κατεύθυνση με σταθερή κυκλοφορία και πεπερασμένο μήκος, όπως φαίνεται στο Σχήμα 2.8. Είναι φανερό οτι, η επαγόμενη ταχύτητα έχει μόνο μια εφαπτομενική συνιστώσα q_θ .



Σχήμα 2.8: Επαγόμενη ταχύτητα από straight Vortex Segment [7]

Από την ανάλυση του απειροστού νήματος δίνης, έχει αποδειχθεί ότι η επαγόμενη ταχύτητα είναι:

$$\Delta \mathbf{V} = \frac{\Gamma}{4\pi} \frac{d\mathbf{l} \times \vec{r}}{r^3} = \frac{\Gamma}{4\pi} \frac{\sin(\beta)}{r^2} dl \hat{e}_\theta \quad (2.49)$$

Σημειωτέων ότι:

$$d = r \sin(\beta) \quad \text{και} \quad \tan(\pi - \beta) = \frac{d}{l}$$

Για αυτό:

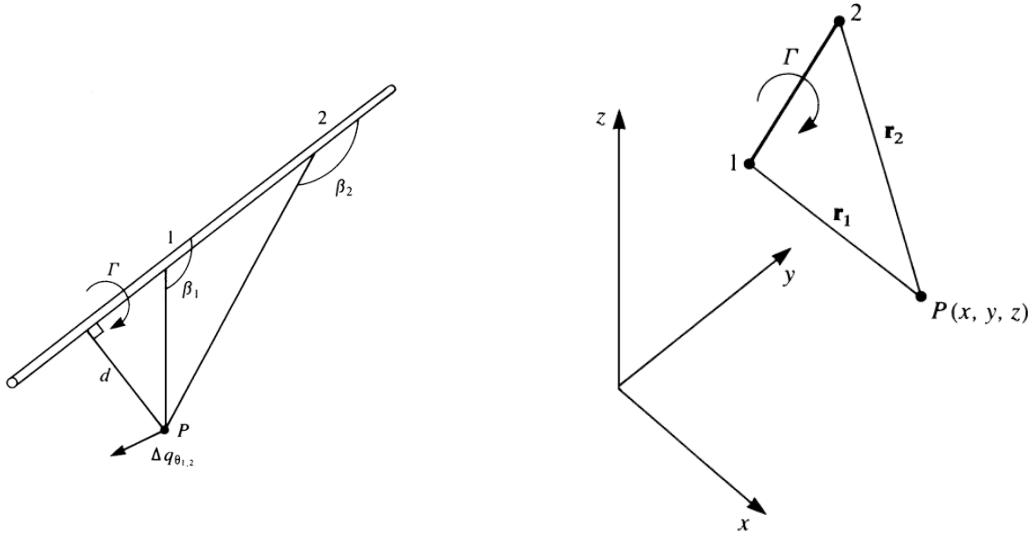
$$l = -\frac{d}{\tan(\beta)} \quad \text{και} \quad dl = \frac{d}{\sin^2(\beta)} d\beta$$

Αντικαθιστώντας αυτούς τους όρους στην εξίσωση (2.49) προκύπτει:

$$\Delta V_\theta = \frac{\Gamma}{4\pi d} \sin(\beta) d\beta \quad (2.50)$$

Αυτή η εξίσωση μπορεί να ολοκληρωθεί πάνω στο ευθύγραμμο τμήμα του νήματος δίνης όπως φαίνεται στο Σχήμα 2.9, οδηγώντας σε:

$$V_{\theta_1 \rightarrow 2} = \frac{\Gamma}{4\pi d} \int_{\beta_1}^{\beta_2} \sin(\beta) d\beta = \frac{\Gamma}{4\pi d} (\cos(\beta_1) - \cos(\beta_2)) \quad (2.51)$$



Σχήμα 2.9: Γεωμετρία Vortex segment [7]

Τέλος, σημειώνοντας ότι:

$$d = \frac{|\vec{r}_1 \times \vec{r}_2|}{|\vec{r}_0|}, \quad \cos \beta_1 = \frac{\vec{r}_0 \cdot \vec{r}_1}{|\vec{r}_0||\vec{r}_1|}$$

$$\cos \beta_2 = \frac{\vec{r}_0 \cdot \vec{r}_2}{|\vec{r}_0||\vec{r}_2|}, \quad \hat{e}_\theta = \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|}$$

Η επαγόμενη ταχύτητα γίνεται:

$$\vec{V}_{1 \rightarrow 2} = \frac{\Gamma}{4\pi} \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|} (\vec{r}_1 - \vec{r}_2) \cdot \left(\frac{\vec{r}_1}{|\vec{r}_1|} - \frac{\vec{r}_2}{|\vec{r}_2|} \right) \quad (2.52)$$

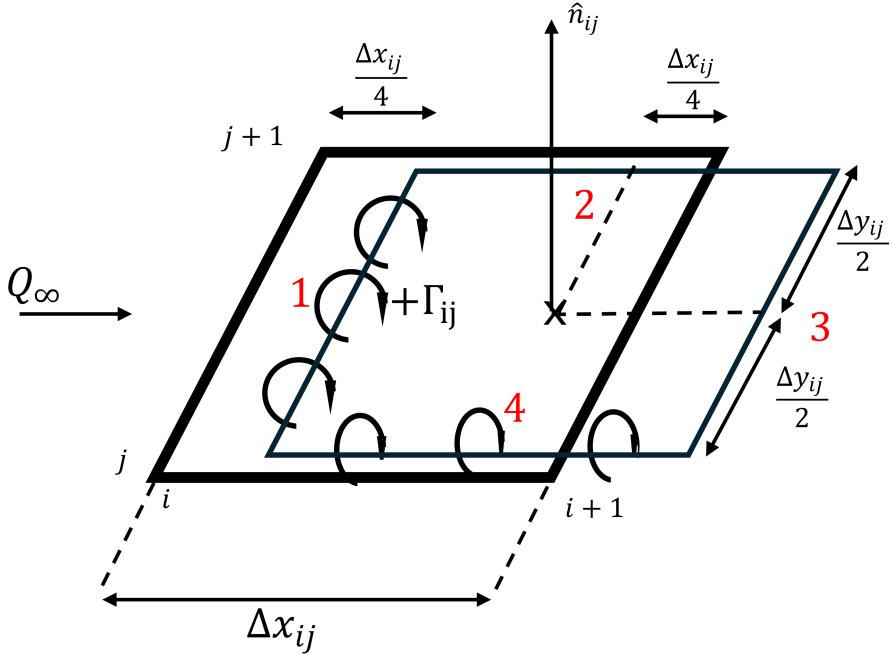
2.2.3 Υπολογιστική λύση επιφάνειας άντωσης με Vortex Ring Elements

Ο στόχος αυτής της ενότητας είναι ο υπολογισμός του πίνακα αεροδυναμικών συντελεστών επιρροής που είναι απαραίτητος για την πρόβλεψη της άντωσης μιας ταλαντευόμενης επιφάνειας πτέρυγας.

Η κύρια συνοριακή συνθήκη που πρέπει να ικανοποιηθεί είναι η μηδενική ροή κάθετα της στερεάς επιφάνειας της πτέρυγας, η οποία μπορεί να εκφραστεί χρησιμοποιώντας το δυναμικό ταχύτητας $\nabla \Phi = \vec{V}$ ως:

$$\nabla (\Phi + \Phi_\infty) \cdot \hat{n} = 0 \quad (2.53)$$

Η αριθμητική μέθοδος ξεκινά με τον ορισμό του τύπου του αεροδυναμικού στοιχείου που θα χρησιμοποιηθεί. Στην μέθοδο V.L.M., χρησιμοποιείται ένας δακτύλιος δινών. Ο δακτύλιος δινών είναι ένα τετραγωνικό στοιχείο που έχει ένα ευθύ τμήμα δίνης σε κάθε πλευρά. Η leading edge δίνη τοποθετείται στο ένα τέταρτο της χορδής του πάνελ, ενώ το σημείο ολοκλήρωσης collocation point είναι στο κέντρο της γραμμής των τριών τετάρτων της χορδής. Το normal διάνυσμα του πάνελ υπολογίζεται στο σημείο ολοκλήρωσης. Η θετική κυκλοφορία Γ ορίζεται σύμφωνα με τον κανόνα του δεξιού χεριού. Τοποθετώντας την leading edge δίνη στη γραμμή του ενός τετάρτου της χορδής ικανοποιείται η συνθήκη του Kutta στις δύο διαστάσεις.



Σχήμα 2.10: Απεικόνιση στοιχείου Vortex Ring Element

Οι αριθμοί 1 έως 4 στο Σχήμα 2.10, αντιπροσωπεύουν τα τέσσερα πεπερασμένα ευθύγραμμα τμήματα δινών που συνθέτουν το στοιχείο. Η επαγόμενη ταχύτητα του στοι-

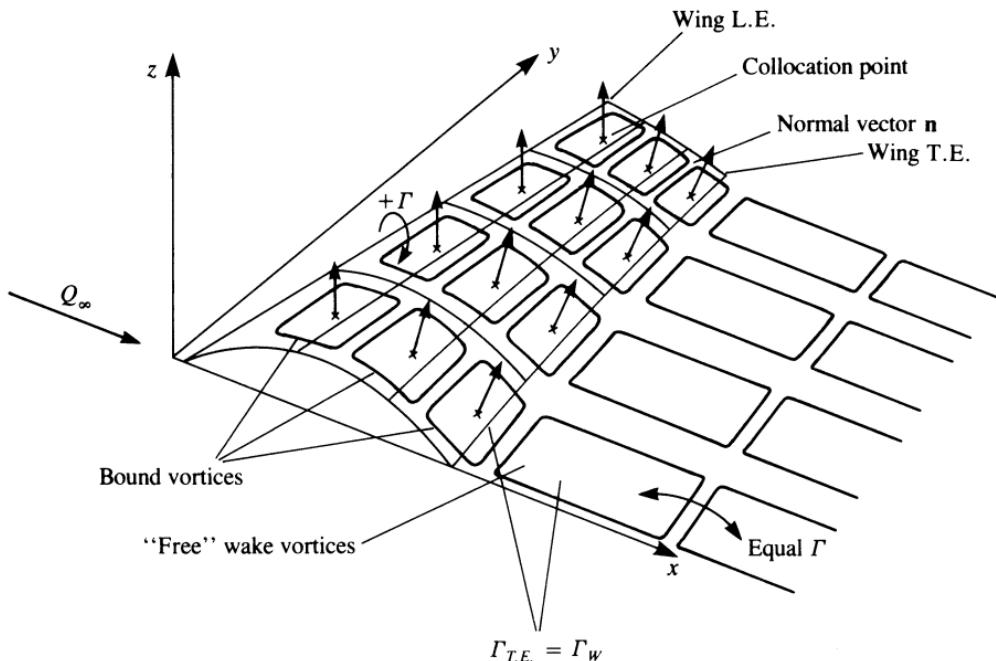
χείου μπορεί να υπολογιστεί χρησιμοποιώντας την εξίσωση (2.52) σε κάθε τμήμα ξεχωριστά. Για ένα αυθαίρετο σημείο στον χώρο $P(x, y, z)$, η επαγόμενη ταχύτητα είναι:

$$\vec{V}_P = \vec{V}_1 + \vec{V}_2 + \vec{V}_3 + \vec{V}_4 \quad (2.54)$$

Όπου:

$$\vec{V}_i = \frac{\Gamma}{4\pi} \frac{\vec{r}_{i,1} \times \vec{r}_{i,2}}{|\vec{r}_{i,1} \times \vec{r}_{i,2}|} (\vec{r}_{i,1} - \vec{r}_{i,2}) \cdot \left(\frac{\vec{r}_{i,1}}{|\vec{r}_{i,1}|} - \frac{\vec{r}_{i,2}}{|\vec{r}_{i,2}|} \right) \quad (2.55)$$

Όπως φάινεται στο Σχήμα 2.11, αυτά τα στοιχεία ταξινομούνται σε ένα δισδιάστατο πλέγμα για να καλύψουν το σχήμα της ανυψωτικής επιφάνειας. Για να ικανοποιηθεί η συνθήκη Kutta, η επίδραση της πίσω δίνης της τελευταίας σειράς πάνελ πρέπει να ακυρωθεί. Επομένως, η τελευταία σειρά πάνελ καθώς και όλα τα πάνελ του απόρρου πρέπει να έχουν την ίδια κυκλοφορία Γ .

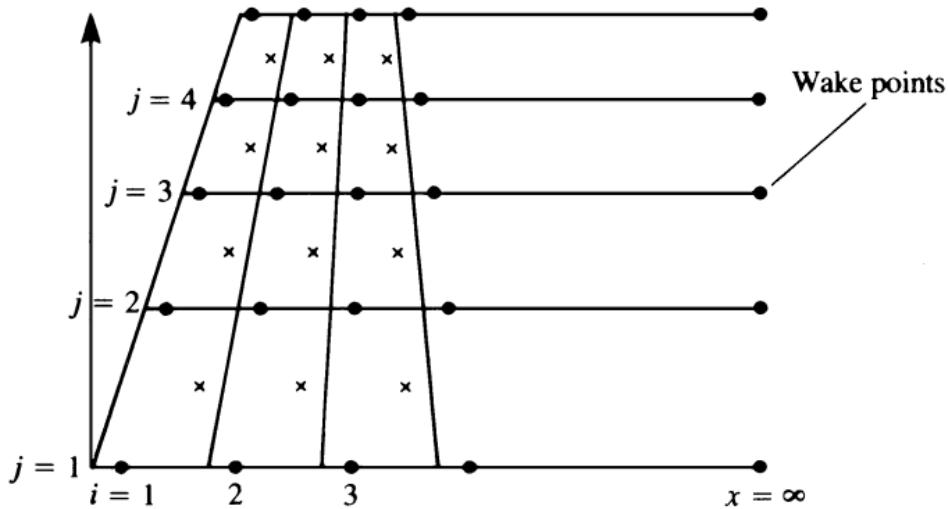


Σχήμα 2.11: Vortex ring elements σε πλέγμα [7]

Ο συντελεστής επιρροής α_{ij} είναι ουσιαστικά η επαγόμενη ταχύτητα του i -οστου στοι-

χείου δίνης με μοναδιαία κυκλοφορία στο j -οστό σημείο ολοκλήρωσης. Οι συντελεστές επιρροής μπορούν να υπολογιστούν εύκολα χρησιμοποιώντας την εξίσωση (2.55). Επαναλαμβάνοντας τον υπολογισμό για όλα τα πάνελ και όλα τα σημεία ολοκλήρωσης οδηγεί σε έναν πίνακα.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \quad (2.56)$$



Σχήμα 2.12: Διάταξη από wing και wake panels [7]

Για λόγους πληρότητας, το σύνολο των γραμμικών εξισώσεων που πρέπει να λυθεί προκειμένου να υπολογιστεί ή ένταση της κυκλοφορίας κάθε στοιχείου για μια δεδομένη γωνία προσβολής για κάθε πάνελ είναι:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_m \end{bmatrix} = \begin{bmatrix} -\vec{Q}_\infty \cdot \hat{n}_1 \\ -\vec{Q}_\infty \cdot \hat{n}_2 \\ \vdots \\ -\vec{Q}_\infty \cdot \hat{n}_m \end{bmatrix} \quad (2.57)$$

Η καθοδική ροή που προκαλείται σε κάθε στοιχείο δακτυλίου δίνης μπορεί να υπολογιστεί με ένα άλλο σύνολο γραμμικών εξισώσεων.

$$\begin{bmatrix} w_{ind1} \\ w_{ind2} \\ \vdots \\ w_{indm} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_m \end{bmatrix} \quad (2.58)$$

Η άντωση και η αντίσταση μπορούν στη συνέχεια να υπολογιστούν χρησιμοποιώντας τις σχέσεις:

$$L = \sum_{i=1}^M \sum_{j=1}^N \Delta L_{ij},$$

$$\text{Όπου } \Delta L_{ij} = \begin{cases} \rho Q_\infty (\Gamma_{i,j} - \Gamma_{i-1,j}) \Delta y_{ij}, & \text{γιά } i > 1 \\ \rho Q_\infty \Gamma_{ij} \Delta y_{ij}, & \text{γιά } i = 1 \end{cases} \quad (2.59)$$

$$D = \sum_{i=1}^M \sum_{j=1}^N \Delta D_{ij},$$

$$\text{Όπου } \Delta D_{ij} = \begin{cases} \rho w_{w\delta_{i,j}} (\Gamma_{i,j} - \Gamma_{i-1,j}) \Delta y_{ij}, & \text{γιά } i > 1 \\ \rho w_{w\delta_{i,j}} \Gamma_{ij} \Delta y_{ij}, & \text{γιά } i = 1 \end{cases} \quad (2.60)$$

2.3 Εξισώσεις Ανάλυσης Πτερυγισμού

Η γενική μορφή της αεροελαστικής εξίσωσης κίνησης είναι [6]

$$[A]\ddot{q} + (\rho V[B] + D)\dot{q} + (\rho V^2[C] + [E])q = [0] \quad (2.61)$$

Όπου:

- $[A]$, $[C]$, και $[E]$, είναι οι πίνακες που αντιστοιχούν στους $[M]$, $[C]$ και $[K]$ που είναι ο κλασικός συμβολισμός που χρησιμοποιείται στην ανάλυση ταλαντώσεων.
- $[B]$ και $[C]$ είναι οι αεροδυναμικοί πίνακες που εξαρτώνται από τον αριθμό Mach και τη μειωμένη συχνότητα reduced frequency. Ο τρόπος υπολογισμού αυτών των πινάκων μπορεί να ποικίλλει ανάλογα με την αεροδυναμική θεωρία που χρησιμοποιείται.
- q είναι οι γενικευμένες συντεταγμένες.

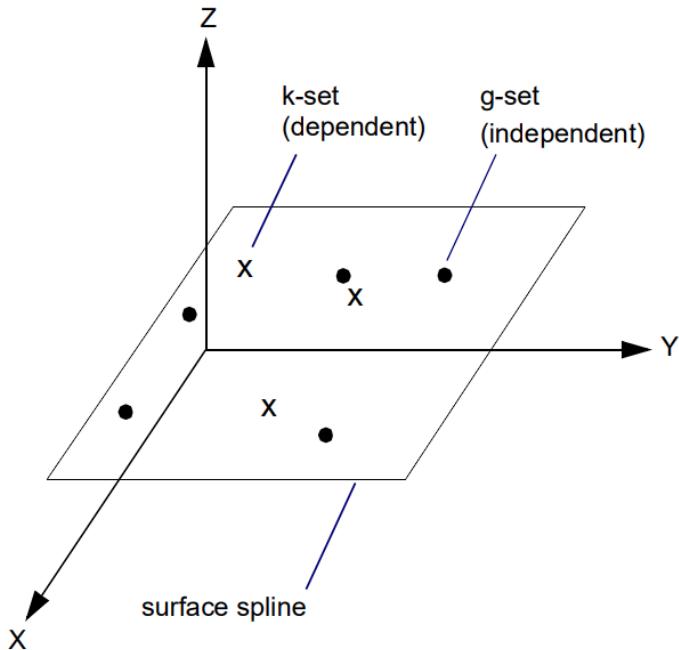
2.3.1 Σύνδεση της κατασκευής με την αεροδυναμική – Infinite Plate splines

Οι κατασκευαστικοί και οι αεροδυναμικοί βαθμοί ελευθερίας συνδέονται μέσω παρεμβολής. Αυτό επιτρέπει την ανεξάρτητη επιλογή των σημείων πλέγματος (κόμβων) των κατασκευαστικών και αεροδυναμικών πινάκων, αποσυνδέοντας έτσι τις δύο διακριτοποιήσεις της γεωμετρίας. Αυτή η μέθοδος παρεμβολής ονομάζεται splining. Το spline άπειρης πλάκας που φαίνεται στο Σχήμα 2.13 (Infinite Plate Spline) και χρησιμοποιείται εδώ, βασίζεται στην παραμόρφωση μιας θεωρητικά άπειρης πλάκας που υπόκειται σε σημειακά φορτία σε συγκεκριμένα σημεία. Τα splines χρησιμοποιούνται για δύο διαφορετικούς σκοπούς. Ως παρεμβολέας δυνάμεων για τον υπολογισμό της ισοδύναμης κατανομής δυνάμεων στην κατασκευή και ως παρεμβολέας μετατόπισης για τον υπολογισμό ενός συνόλου αεροδυναμικών μετατοπίσεων δεδομένου ενός συνόλου κατασκευαστικών μετατοπίσεων. Μαθηματικά αυτό σημαίνει ότι:

$$F_s = [G_{s \rightarrow a}] F_a, \quad U_a = [G_{a \rightarrow s}] U_s \quad (2.62)$$

Όπου:

- F_s και F_a είναι τα διανύσματα κατασκευαστικών και αεροδυναμικών δυνάμεων.
- U_a και U_s είναι τα αεροδυναμικά και κατασκευαστικά διανύσματα μετατόπισης.
- $G_{s \rightarrow a}$ και $G_{a \rightarrow s}$ είναι οι πίνακες spline που συνδέουν το κατασκευαστικό με το αεροδυναμικό πρόβλημα και αντίστροφα. Έχει αποδειχθεί με την αρχή του εικονικού έργου ότι ο ένας πίνακας spline είναι ο ανάστροφος του άλλου. Δηλαδή: $[G_{s \rightarrow a}] = [G_{a \rightarrow s}]^T$.



Σχήμα 2.13: Σύστημα συντεταγμένων Surface Spline [13]

To Infinite plate spline και όλα τα άλλα επιφανειακά splines χρησιμοποιούνται για να βρουν μια επιφάνεια της μορφής $w(x, y)$ για όλα τα σημεία (x, y) όταν το w είναι γνωστό σε ένα διακριτό σύνολο σημείων $w_i = w(x_i, y_i)$.

To Infinite plate spline μιμείται τη συμπεριφορά μιας άπειρης πλάκας υπό σημειακή φόρτιση. Η παραμόρφωση μιας τέτοιας πλάκας λόγω ενός ενιαίου συγκεντρωμένου φορτίου στο σημείο $(x_i = 0, y_i = 0)$ ονομάζεται θεμελιώδης λύση. Η κυβερνητική εξίσωση είναι:

$$D\nabla^4(w) = D \frac{1}{r} \frac{d}{dr} \left\{ r \frac{d}{dr} \left[\frac{1}{r} \frac{d}{dr} \left(r \frac{dw}{dr} \right) \right] \right\} = q \quad (2.63)$$

Η γενική λύση της ομογενούς μορφής της εξίσωσης 2.63 είναι:

$$w = C_1 + C_1 r^2 + C_2 \ln(r) + C_3 r^2 \ln(r) \quad (2.64)$$

Θέτοντας $C_2 = 0$ για να παραμείνει η λύση πεπερασμένη στο $r = 0$, τότε η ολοκλήρωση από το $r = 0$ στο $r = \epsilon$ (*small number*) οδηγεί στο $C_3 = \frac{P}{8\pi D}$. Έτσι, η θεμελιώδης λύση για συγκεντρωμένο φορτίο είναι:

$$w = A + Br^2 + \frac{P}{16\pi D} r^2 \ln(r^2) \quad (2.65)$$

Η θεμελιώδης λύση μπορεί να υπερτεθεί για την επίλυση όλου του προβλήματος της πλάκας:

$$w(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 y + \sum_{i=1}^N K_i(x, y) P_i \quad (2.66)$$

Όπου:

- $K_i(x, y) = \frac{1}{16\pi D} r_i^2 \ln(r_i^2)$
- $r_i^2 = (x - x_i)^2 + (y - y_i)^2$
- P_i concentrated load at (x_i, y_i)

Οι $N + 3$ άγνωστοι καθορίζονται από τις $N + 3$ εξισώσεις.

$$\sum P_i = 0 \quad (2.67)$$

$$\sum x_i P_i = 0 \quad (2.68)$$

$$\sum y_i P_i = 0 \quad (2.69)$$

$$w_j = a_0 + a_1 x_j + a_2 y_j + \sum_{i=1}^N K_{ij} P_i, \quad (j = 1, \dots, N) \quad (2.70)$$

Όπου $K_{ij} = K_i(x_j, y_j)$ και $K_{ij} = K_{ji}$, $K_{ij} = 0$ όταν $i = j$

Αυτές οι $N + 3$ εξισώσεις μπορούν να εκφραστούν σε μορφή πίνακα:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \hline w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & x_1 & \cdots & x_N \\ 0 & 0 & 0 & y_1 & \cdots & y_N \\ \hline 1 & x_1 & y_1 & 0 & \cdots & K_{1N} \\ 1 & x_2 & y_2 & K_{21} & \cdots & K_{2N} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_N & y_N & K_{N1} & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \hline P_1 \\ P_2 \\ \vdots \\ P_N \end{bmatrix} = [C][P] \quad (2.71)$$

Τώρα που τα a_i και P_i είναι γνωστά, η παρεμβολή μπορεί να εφαρμοστεί σε οποιοδήποτε μέγεθος για κάθε σημείο στο επίπεδο (x, y) χρησιμοποιώντας την ακόλουθη εξίσωση:

$$\{w\} = \begin{bmatrix} 1 & x_1 & y_1 & K_{1,1} & K_{1,2} & \cdots & K_{1,n} \\ 1 & x_2 & y_2 & K_{2,1} & K_{2,2} & \cdots & K_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_n & y_n & K_{n,1} & K_{n,2} & \cdots & K_{n,n} \end{bmatrix} \cdot [C]^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hline w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} \quad (2.72)$$

2.3.2 Η μέθοδος επίλυσης πτερυγισμού PK

Η θεμελιώδης εξίσωση για την ανάλυση πτερυγισμού χρησιμοποιώντας τη μέθοδο PK είναι [13]:

$$\left[M_{hh} p^2 + \left(B_{hh} - \frac{1}{4} \rho c V \frac{Q_{hh}^I}{k} \right) p + \left(K_{hh} - \frac{1}{2} \rho V^2 Q_{hh}^R \right) \right] \cdot \{u_h\} = 0 \quad (2.73)$$

Όπου:

- M_{hh} = πίνακας μάζας ιδιομορφών, συνήθως διαγώνιος.
- B_{hh} = πίνακας απόσβεσης ιδιομορφών.
- K_{hh} = πίνακας δυσκαμψίας ιδιομορφών, συνήθως διαγώνιος, μπορεί να είναι και μιγαδικός αν χρησιμοποιηθεί πραγματική απόσβεση.
- $Q_{hh}(M, k) =$ πίνακας αεροδυναμικής δύναμης που είναι συνάρτηση του αριθμού $Mach$ και της μειωμένης συχνότητας $k = \frac{\omega\bar{c}}{2V}$.
- $Q_{hh}R, Q_{hh}I$ είναι τα πραγματικά και φανταστικά μέρη του πίνακα αεροδυναμικής δύναμης Q_{hh} ή πίνακες αεροδυναμικής δυσκαμψίας και αεροδυναμικής απόσβεσης αντίστοιχα.
- $k = \frac{\omega\bar{c}}{2V}$ μειωμένη συχνότητα reduced frequency.
- $M =$ αριθμός $Mach$.
- $V =$ ταχύτητα.
- $\rho =$ πυκνότητα ρευστού.
- $\{u_h\}$ = modal participation vector.
- $p = \omega(\gamma \pm i)$ ιδιοτυπή.

Η εξίσωση επαναδιατυπώνεται στη μορφή state-space με μείωση τάξης.

$$[A - pI] \{\bar{u}_h\} = 0 \quad (2.74)$$

Όπου:

$$[A] = - \begin{bmatrix} [\mathbf{0}] & [\mathbf{I}] \\ -M_{hh}^{-1} \left(K_{hh} - \frac{1}{2} \rho V^2 Q_{hh}^R \right) & -M_{hh}^{-1} \left(B_{hh} - \frac{1}{4} \rho c V \frac{Q_{hh}^I}{k} \right) \end{bmatrix} \quad (2.75)$$

$$\bar{u}_h = \begin{bmatrix} u_h \\ \dot{u}_h \end{bmatrix} \quad (2.76)$$

Ο βασικός αλγόριθμος για τη μέθοδο PK περιλαμβάνει τα ακόλουθα βήματα:

1. Γίνεται μια αρχική εκτίμηση της ιδιοσυχνότητας για την ιδιομορφή.
2. Υπολογίζεται η μειωμένη συχνότητα k και η ταχύτητα V .
3. Προσδιορίζονται (με παρεμβολή) οι πίνακες αεροδυναμικής δυσκαμψίας και απόσβεσης $Q_{hh}R$, $Q_{hh}I$.
4. Προσδιορίζονται οι ιδιοσυχνότητες για το σύστημα σε αυτήν την πτητική συνθήκη χρησιμοποιώντας την εξίσωση (2.74).
5. Επιλέγεται η ιδιοσυχνότητα που ήταν πιο κοντά στην αρχική εκτίμηση και επαναλαμβάνονται τα βήματα 1-5 μέχρι να συγκλίνει η λύση για τη συχνότητα.
6. Αποθηκεύεται η τελική ιδιοσυχνότητα και η απόσβεση της ιδιομορφής.
7. Εξετάζεται η επόμενη ιδιομορφή ενδιαφέροντος χρησιμοποιώντας την ίδια διαδικασία μέχρι να έχουν διερευνηθεί όλες οι ιδιομορφές.

2.4 Τεχνικές βελτιστοποίησης

2.4.1 Μέθοδος γραμμικής αναζήτησης Brent – Dekker

Αυτός ο αλγόριθμος στοχεύει στην εύρεση του ελαχίστου μιας συνάρτησης μιας μεταβλητής χωρίς τη χρήση των παραγώγων της. Αυτός ο αλγόριθμος χρησιμοποιείται συνήθως για την εύρεση του ελαχίστου μιας συνάρτησης $f(x)$ πολλαπλών μεταβλητών [2]. Για το σκοπό αυτό συχνά απαιτείται η ελαχιστοποίηση της ακόλουθης συνάρτησης

$$\gamma(\lambda) = f(\mathbf{x}_0 + \lambda \mathbf{s}) \quad (2.77)$$

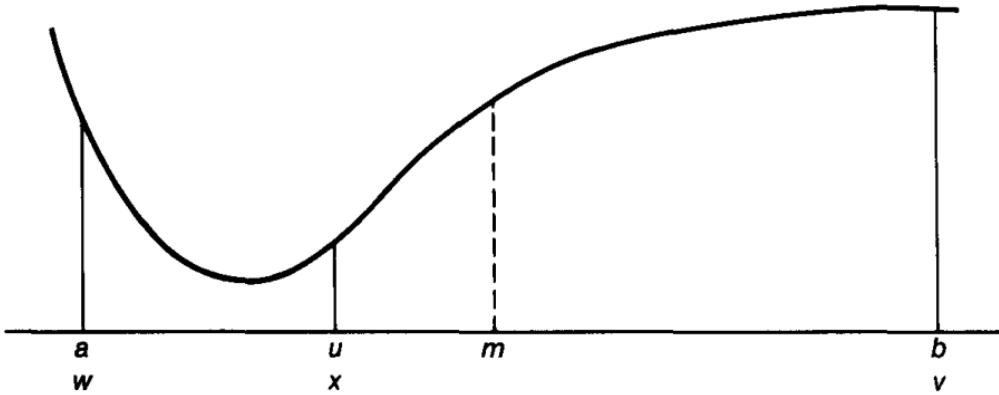
Όπου \mathbf{x}_0 και \mathbf{s} είναι σταθερά και το λ είναι μια βαυμωτή μεταβλητή. Το πρόβλημα αυτό περιγράφει ουσιαστικά μια μονοδιάστατη αναζήτηση ξεκινώντας από το \mathbf{x}_0 προς την κατεύθυνση του \mathbf{s} .

Ο αλγόριθμος βρίσκει μια προσέγγιση στο ελάχιστο μιας συνάρτησης f ορισμένης στο διάστημα $[a, b]$. Υπάρχουν έξι σημαντικά σημεία a, b, u, v, w και x τα οποία δεν είναι όλα διακριτά. Μία πιθανή διάταξη των σημείων αυτών φαίνεται στο Σχήμα 2.14. Αυτά τα σημεία αρχικοποιούνται ως εξής:

$$v = w = x = a + \left(\frac{3 - \sqrt{5}}{2} \right) (b - a) \quad (2.78)$$

Ο αριθμός $\left(\frac{3 - \sqrt{5}}{2} \right)$ προέρχεται από τον αλγόριθμο αναζήτησης χρυσής τομής και είναι αυθαίρετη επιλογή. Τα σημεία που ορίζονται στην εξίσωση (2.78) εξυπηρετούν ένα συγκεκριμένο σκοπό:

- Τα σημεία a και b ορίζουν το διάστημα μέσα στο οποίο βρίσκεται ένα τοπικό ελάχιστο.
- x : από όλα τα σημεία στα οποία έχει υπολογιστεί η f , το x είναι το σημείο με την ελάχιστη τιμή της f .
- w είναι το σημείο με την επόμενη χαμηλότερη τιμή του f .
- v είναι η προηγούμενη τιμή του w .
- u είναι το τελευταίο σημείο στο οποίο έχει υπολογιστεί η f .



Σχήμα 2.14: Μια πιθανή διαμόρφωση σημείων [2]

Μια τυπική επανάληψη του αλγόριθμου εξελίσσεται ως εξής:

1. Έστω $m = 1/2(a + b)$ το μέσο του διαστήματος.

α' . Αν $|x - m| \leq 2 \cdot tol - \frac{1}{2}(b - a)$, τότε ο αλγόριθμος τερματίζεται με το x ως ελάχιστο.
 β' . Διαφορετικά, οι αριθμοί p και q υπολογίζονται έτσι ώστε $x + p/q$ να είναι το ελάχιστο της παραβολής που διέρχεται από τα σημεία $(v, f(v))$, $(w, f(w))$, $(x, f(x))$.

2. Έστω e η τιμή του p/q .

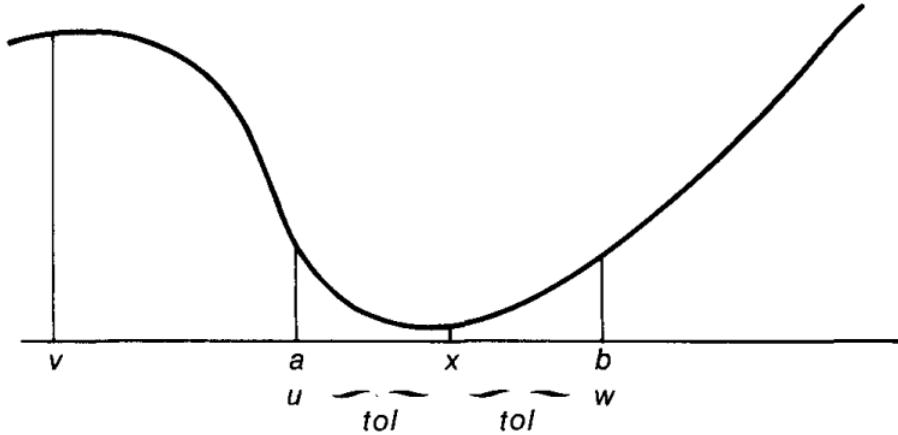
α' . Αν $|e| \leq tol$, $q = 0$, $x + p/q \notin [a, b]$ ή $|p/q| \geq 1/2|e|$, τότε πραγματοποιείται το βήμα της χρυσής τομής:

$$u = \begin{cases} \frac{\sqrt{5}-1}{2}x + \frac{3-\sqrt{5}}{2}a, & \text{αν } x \geq m \\ \frac{\sqrt{5}-1}{2}x + \frac{3-\sqrt{5}}{2}b, & \text{αν } x < m \end{cases} \quad (2.79)$$

β' . Διαφορετικά $u = x + p/q$ (οι αποστάσεις $|u - x|$, $u - a$, $b - u$ πρέπει να είναι τουλάχιστον tol).

3. Η f υπολογίζεται στο νέο σημείο u και τα σημεία a, b, v, w και x ενημερώνονται και ο κύκλος επαναλαμβάνεται.

Ο αλγόριθμος συνήθως τερματίζει όταν $x = b - tol$ ή $x = a + tol$ μετά από την πραγματοποίηση μιας παραβολικής παρεμβολής, όπου έχει εφαρμοστεί η συνθήκη ότι $|u - x| \geq tol$. Το επόμενο σημείο παραβολικής παρεμβολής βρίσκεται κοντά στα x και b , επομένως το u είναι αναγκαστικά $x - tol$. Μια τυπική κατάσταση στην οποία ο αλγόριθμος τερματίζει φαίνεται στο Σχήμα 2.15



Σχήμα 2.15: Τυπική τερματική διαμόρφωση σημαντικών σημείων [2]

2.4.2 Η Μέθοδος του Powell

Η μέθοδος του Powell είναι μια τροποποιημένη κυκλική αναζήτηση συντεταγμένων. Και οι δύο μέθοδοι στοχεύουν στην ελαχιστοποίηση μιας συνάρτησης πολλών μεταβλητών χωρίς να χρησιμοποιούν καμία από τις παραγώγους της. Γι' αυτό αποκαλούνται μέθοδοι μηδενικής τάξης.

Η μέθοδος κυκλικής αναζήτησης συντεταγμένων είναι στην ουσία μια σειρά βελτιστοποιήσεων γραμμικής αναζήτησης που εκτελούνται κυκλικά σε όλες τις παραμέτρους της συνάρτησης. Η αναζήτηση ξεκινά από ένα αρχικό $x^{(0)}$ και βελτιστοποιεί την πρώτη παράμετρο:

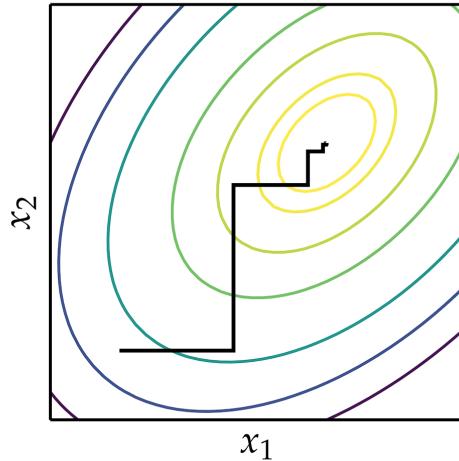
$$\mathbf{x}^{(1)} = \arg_{x_1} \min f(x_1, x_2^{(0)}, \dots, x_n^{(0)}) \quad (2.80)$$

Αφού βρεθεί αυτό, βελτιστοποιεί την επόμενη παράμετρο:

$$\mathbf{x}^{(2)} = \arg_{x_2} \min f(x_1^{(1)}, x_2, \dots, x_n^{(1)}) \quad (2.81)$$

Αυτό μπορεί επίσης να εκφραστεί με τη βοήθεια της εξίσωσης (2.77) αν s είναι το i -οστό διανύσμα βάσης για $i = 1, \dots, n$.

Μπορεί να χρησιμοποιηθεί οποιοσδήποτε αλγόριθμος γραμμικής αναζήτησης, αλλά ο αλγόριθμος του Brent βρίσκει ευρεία χρήση σε βιβλιοθήκες βελτιστοποίησης, συμπεριλαμβανομένης της υλοποίησης της μεθόδου Powell στήν SciPy.



Σχήμα 2.16: Η μέθοδος Cyclic Coordinate Search [8]

Όπως φαίνεται στο Σχήμα 2.16, ένα μειονέκτημα αυτής της μεθόδου είναι η αργή προσπέλαση των διαγώνιων κοιλάδων, όπου γίνονται επαναλαμβανόμενα μικρά βήματα σε κάθε κατεύθυνση. Η ιδέα του M.J.D. Powell ήταν να επεκτείνει την κυκλική αναζήτηση συντεταγμένων ώστε να μπορεί να αναζητά σε κατευθύνσεις που δεν είναι κάθετες μεταξύ τους και ούτε τα διανύσματα βάσης. Για να επιτευχθεί αυτή η νέα ικανότητα, ο αλγόριθμος του Powell διατηρεί μια λίστα κατευθύνσεων αναζήτησης $[\mathbf{u}^{(0)}, \dots, \mathbf{u}^{(n)}]$ οι οποίες αρχικά είναι τα διανύσματα βάσης συντεταγμένων $\mathbf{u}^{(i)} = e^{(i)}$ για όλα τα i .

Ξεκινώντας από το $\mathbf{x}^{(0)}$, η μέθοδος του Powell διεξάγει μια γραμμική αναζήτηση όπως και πριν, για κάθε κατεύθυνση αναζήτησης διαδοχικά:

$$\mathbf{x}^{(i+1)} \leftarrow \text{linesearch}(f, \mathbf{x}^{(i)}, \mathbf{u}^{(i)}) \text{ for } i \text{ in } [0, \dots, n] \quad (2.82)$$

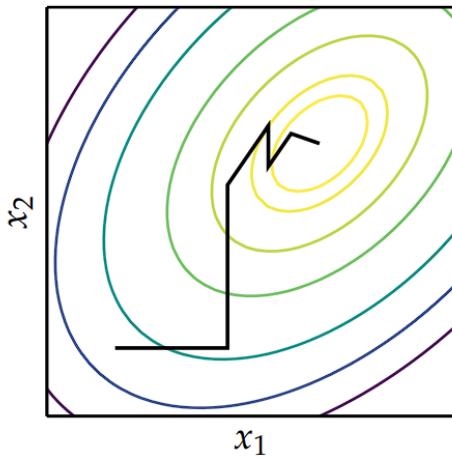
Στη συνέχεια, όλες οι κατευθύνσεις αναζήτησης μετατοπίζονται κατά έναν δείκτη, απορρίπτοντας την παλαιότερη κατεύθυνση αναζήτησης, $\mathbf{u}^{(0)}$

$$\mathbf{u}^{(i)} \leftarrow \mathbf{u}^{(i+1)} \text{ for } i \text{ in } [0, \dots, n-1] \quad (2.83)$$

Η τελευταία κατεύθυνση αναζήτησης αντικαθίσταται με την κατεύθυνση από το $\mathbf{x}^{(0)}$ στο $\mathbf{x}^{(n+1)}$, που είναι η συνολική κατεύθυνση της προόδου κατά τις τελευταίες n γραμμικές αναζητήσεις:

$$\mathbf{u}^{(n)} \leftarrow \mathbf{x}^{(n+1)} - \mathbf{x}^{(0)} \quad (2.84)$$

Αυτή η διαδικασία επαναλαμβάνεται μέχρι τη σύγκλιση.



Σχήμα 2.17: Η μέθοδος Powell [8]

Όπως φαίνεται στο Σχήμα 2.17, η μέθοδος του Πωελλ ξεκινά όπως η κυκλική αναζήτηση συντεταγμένων αλλά σταδιακά «μαθαίνει» την κατεύθυνση μέγιστης καθόδου.

2.4.3 Γενετικός Αλγόριθμος

Οι γενετικοί αλγόριθμοι μιμούνται τη λογική της Δαρβινικής εξέλιξης, όπου τα καταληλότερα «άτομα» είναι πιο πιθανό να μεταβιβάσουν τα γονίδιά τους στην επόμενη γενιά. Τα χρωμοσώματα κάθε ατόμου ορίζουν ένα μόνο σημείο σχεδίασης. Σε κάθε γενιά, τα χρωμοσώματα των καταληλότερων ατόμων μεταβιβάζονται στην επόμενη γενιά αφού περάσουν από τα στάδια της διασταύρωσης, (crossover) και της μετάλλαξης, (mutation).

Τα χρωμοσώματα συχνά αναπαρίστανται από ένα λογικό πίνακα, αλλά σε αυτήν την περίπτωση ένας πίνακας πραγματικών τιμών περιγράφει καλύτερα το χώρο σχεδίασης.

Αρχικοποίηση

Ο Γενετικός Αλγόριθμος ξεκινά με την αρχικοποίηση ενός πληθυσμού με τυχαία χρωμοσώματα.

Επιλογή Γονέων

Η διαδικασία επιλογής είναι η διαδικασία επιλογής των χρωμοσωμάτων που θα δημιουργήσουν απογόνους για την επόμενη γενιά μετά από την υποβολή στις γενετικές διαδικασίες της διασταύρωσης και της μετάλλαξης. Υπάρχουν πολλές μέθοδοι επιλογής των ατόμων που θα ζευγαρώσουν και θα συνδυάσουν τα γονίδιά τους. Μερικές από τις πιο δημοφιλείς περιλαμβάνουν:

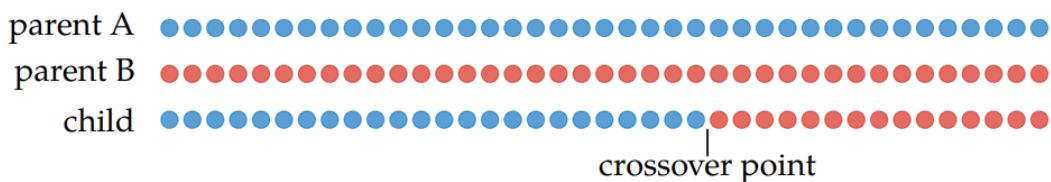
- Roulette Wheel selection: Όπου στο κάθε άτομο έχει ανατεθεί μέρος της φανταστικής ρουλέτας ανάλογη με την απόδοσή τους. Στη συνέχεια, η ρουλέτα περιστρέφεται μέχρι να επιτευχθεί ο επιθυμητός αριθμός γονέων. Αυτή η μέθοδος είναι απλή και αποτελεσματική στις περισσότερες περιπτώσεις. Μπορεί να επηρεαστεί ωστόσο από άτομα που είναι πολύ αποδοτικότερα από τον υπόλοιπο πληθυσμό.
- Tournament Selection: Ένα υποσύνολο ατόμων επιλέγεται τυχαία και το καλύτερο άτομο από αυτά επιλέγεται ως γονέας. Η διαδικασία επαναλαμβάνεται μέχρι να επιτευχθεί ο επιθυμητός αριθμός γονέων.
- Steady State Crossover: Ένα σταθερό μέρος του πληθυσμού αντικαθίσταται σε κάθε γενιά. Μόνο τα αδύναμα άτομα αντικαθίστανται από απογόνους, αυτή η μέθοδος διασφαλίζει ότι το καταλληλότερο άτομο διατηρείται τις περισσότερες φορές. Ένα μειονέκτημα αυτής της μεθόδου είναι ότι αργεί να προσαρμοστεί, καθώς μόνο ένα μικρό μέρος του πληθυσμού αντικαθίσταται σε κάθε γενιά.

Crossover

Η διασταύρωση (Crossover) είναι ο τρόπος με τον οποίο τα χρωμοσώματα των γονέων συνδυάζονται για τη δημιουργία απογόνων. Υπάρχουν διάφοροι τρόποι διασταύρωσης. Οι πιο κοινώς χρησιμοποιούμενοι είναι:

- Single point crossover

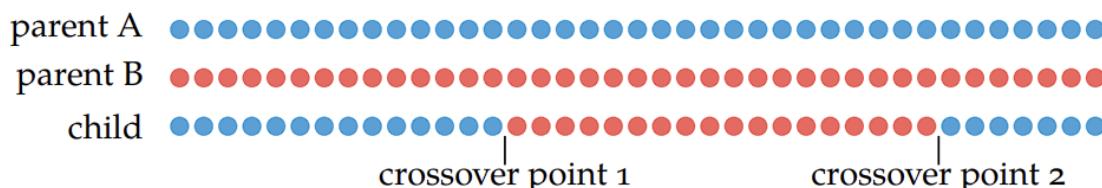
Με αυτόν τον τρόπο, το πρώτο τμήμα του χρωμοσώματος του γονέα Α σχηματίζει το πρώτο τμήμα του χρωμοσώματος του παιδιού και το τελευταίο τμήμα του χρωμοσώματος του γονέα Β σχηματίζει το τελευταίο τμήμα του χρωμοσώματος του παιδιού. Η διαδικασία αυτή φάίνεται στο Σχήμα 2.18



Σχήμα 2.18: Διασταύρωση (Crossover) ενός σημείου [8]

- Two point crossover

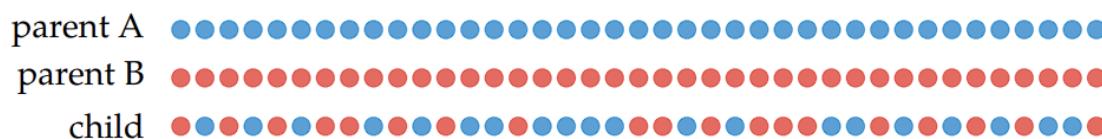
Όπως πριν, αλλά όπως φαίνεται στο Σχήμα 2.19 προστίθεται ένα δεύτερο σημείο διασταύρωσης.



Σχήμα 2.19: Διασταύρωση (Crossover) δύο σημείων [8]

- Uniform crossover

Κατά αυτόν τον τρόπο, όπως φάινεται στο Σχήμα 2.20 κάθε χρωμόσωμα έχει 50% πιθανότητα να προέρχεται από τον γονέα A ή τον γονέα B.



Σχήμα 2.20: Ομοιόμορφη διασταύρωση (Uniform Crossover) [8]

Mutation

Η μετάλλαξη είναι ένα απαραίτητο βήμα του γενετικού αλγορίθμου, καθώς επιτρέπει την ανάπτυξη νέων χαρακτηριστικών που δεν υπήρχαν στον αρχικό πληθυσμό. Στην ουσία, μια τυχαία τιμή προστίθεται σε ένα τυχαίο χρωμόσωμα ενός τυχαία επιλεγμένου απογόνου. Η πιθανότητα να συμβεί μια μετάλλαξη σε ένα άτομο ονομάζεται *mutation rate*.

Elitism

Ο ελιτισμός είναι μια στρατηγική που διασφαλίζει ότι ένας σταθερός αριθμός των καταλληλότερων ατόμων επιβιώνει στην επόμενη γενιά.

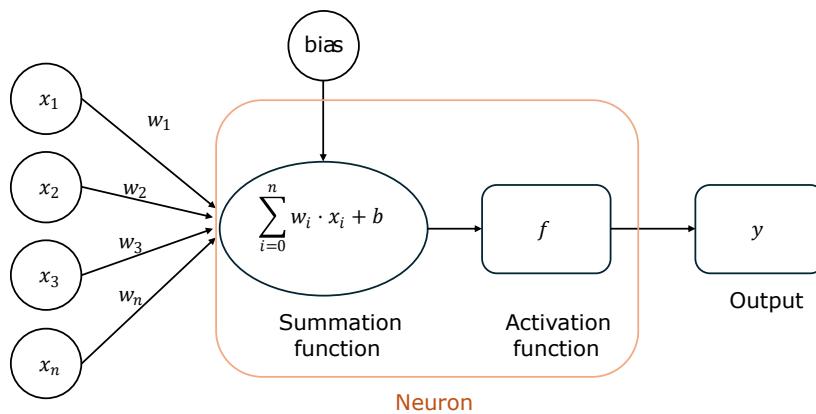
Οι γενετικοί αλγόριθμοι έχουν ένα πολύ ευρύ φάσμα εφαρμογών και είναι πολύ προσαρμόσιμοι σε πολλά προβλήματα. Όμως υπάρχει ένα πρόβλημα. Όλες αυτές οι παράμετροι και στρατηγικές επηρεάζουν σημαντικά τα αποτελέσματα και δεν υπάρχει καμία τυπική μέθοδος επιλογής κάθε παραμέτρου. Επομένως, είναι ευθύνη του ερευνητή να κατανοήσει το πρόβλημα και μέσω πειραματισμού και τεχνικών ρύθμισης υπερπαραμέτρων μοντέλου, να καθορίσει ένα αρκετά καλό σύνολο παραμέτρων.

2.4.4 Νευρωνικά Δίκτυα

Τι είναι τα Τεχνητά Νευρωνικά Δίκτυα (ANN's)

Ένα τεχνητό νευρωνικό δίκτυο είναι ένα υπολογιστικό μοντέλο που εμπνέεται από τον τρόπο λειτουργίας του βιολογικού εγκεφάλου. Η βασική δομή ενός ANN αποτελείται από διασυνδεδεμένους κόμβους (ή νευρώνες) οργανωμένους σε στρώματα.

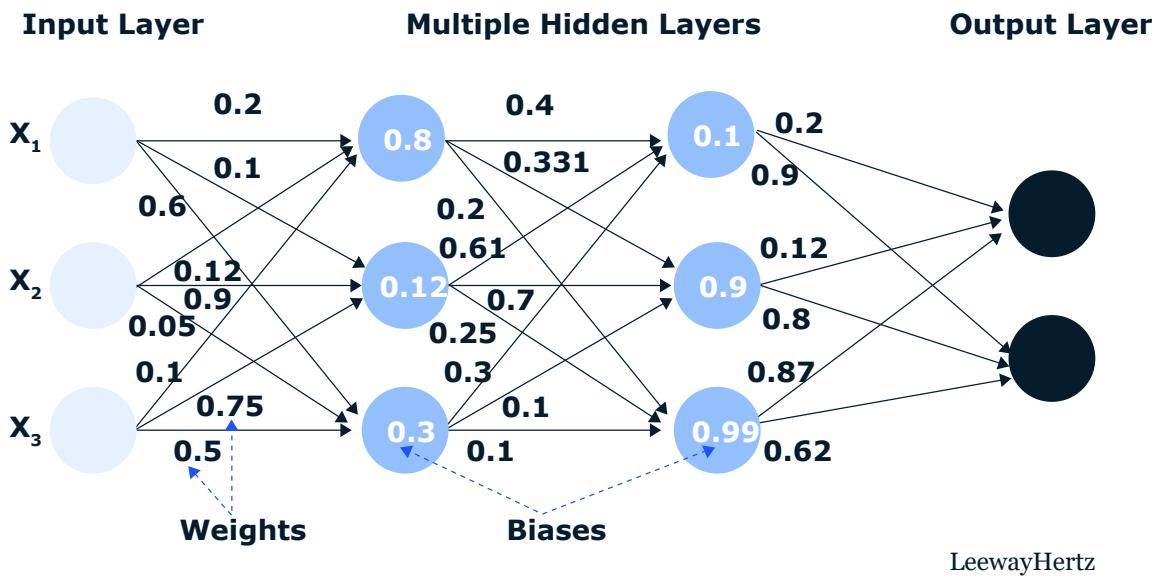
Η βασική μονάδα αυτής της δομής είναι ο νευρώνας, μια απεικόνιση του οποίου φαίνεται στο Σχήμα 2.21. Ένας νευρώνας έχει πολύ απλή δομή. Δέχεται έναν αυθαίρετο αριθμό εισόδων μαζί με μια σταθερά προσαύξησης (bias) και έχει μόνο μία έξοδο. Η έξοδος μπορεί να είναι λογική ή πραγματική. Εντός του νευρώνα, πραγματοποιείται ένα άθροισμα χρησιμοποιώντας το βάρος κάθε εισόδου και την σταθερά προσαύξησης. Το αποτέλεσμα κάθε αθροίσματος επεξεργάζεται περαιτέρω μέσω μιας «συνάρτησης ενεργοποίησης» (activation function) και προκύπτει η έξοδος. Η συνάρτηση ενεργοποίησης επιτρέπει τη μη γραμμική συμπεριφορά των νευρωνικών δικτύων. Μπορεί να είναι οποιαδήποτε συνάρτηση, αλλά ορισμένες συναρτήσεις έχουν επικρατήσει.



Σχήμα 2.21: Η δομή ενός νευρώνα

Η βασική αρχιτεκτονική ενός νευρωνικού δικτύου φαίνεται στο Σχήμα 2.22 και αποτελείται από:

1. Ένα στρώμα εισόδου το οποίο είναι υπεύθυνο για τη λήψη των αρχικών δεδομένων. Κάθε νευρώνας στο στρώμα εισόδου αντιπροσωπεύει ένα συγκεκριμένο χαρακτηριστικό των δεδομένων εισόδων.
2. Κρυφά στρώματα, τα οποία είναι ενδιάμεσα στρώματα μεταξύ των στρωμάτων εισόδου και εξόδου και είναι υπεύθυνα για την αναγνώριση σύνθετων χαρακτηριστικών.
3. Τα βάρη και οι σταθερές προσαύξησης που χρησιμοποιούνται στο νευρωνικό δίκτυο, και τα οποία είναι οι παράμετροι που πρέπει να εκπαιδευτούν για να μπορέσει το νευρωνικό δίκτυο να κάνει ουσιαστικές προβλέψεις.
4. Ένα στρώμα εξόδου το οποίο παράγει τις τελικές προβλέψεις. Η έξοδος μπορεί να είναι δυαδική για προβλήματα ταξινόμησης ή κλιμακωτή για προβλήματα παλινδρόμησης.



Σχήμα 2.22: Η δομή ενός νευρωνικού δικτύου [14]

Ο αριθμός των κρυφών στρωμάτων σε ένα νευρωνικό δίκτυο μπορεί να διαφέρει. Τα νευρωνικά δίκτυα που δεν έχουν κρυφά στρώματα ονομάζονται Ρηχά (shallow), ενώ τα νευρωνικά δίκτυα με κρυφά στρώματα ονομάζονται Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks).

Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

Όπως αναφέρθηκε προηγουμένως, για να είναι χρήσιμο ένα νευρωνικό δίκτυο πρέπει να εκπαιδευτεί. Η εκπαίδευση περιλαμβάνει τη συλλογή δεδομένων παρόμοιων με αυτά που το NN προορίζεται να προβλέψει, και την εκτέλεση μιας βελτιστοποίησης των βαρών και των προσαυξήσεων εντός του δικτύου για την ελαχιστοποίηση κάποιας συνάρτησης απώλειας loss function. Μια συνάρτηση απώλειας είναι ένα μέτρο του ποσού του σφάλματος πρόβλεψης. Υπάρχουν πολλές συναρτήσεις απώλειας που χρησιμοποιούνται συνήθως για τα νευρωνικά δίκτυα, αλλά όταν ένα νευρωνικό δίκτυο εκπαιδεύεται σε ένα πρόβλημα παλινδρόμησης, η συνάρτηση απώλειας είναι συνήθως μία από τις ακόλουθες:

1. Mean Squared Error (MSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (2.85)$$

2. Root Mean Squared Error

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \quad (2.86)$$

3. Mean Average Error (MAE)

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (2.87)$$

Όπου: \hat{y} είναι η προβλεπόμενη τιμή του y

Ο στόχος της εκπαίδευσης του νευρωνικού δικτύου είναι η ελαχιστοποίηση οποιασδήποτε από αυτές τις συναρτήσεις στα δεδομένα δοκιμής που το δίκτυο δεν έχει ξαναδεί.

Για να επιτευχθεί αυτός ο στόχος, πρώτα τα διαθέσιμα δεδομένα που περιέχουν τόσο την είσοδο όσο και την επιθυμητή έξοδο, χωρίζονται τυχαία σε δεδομένα εκπαίδευσης (training data) (αυτά που θα χρησιμοποιηθούν για την προσαρμογή των βαρών και των προσαυξήσεων) και δεδομένα δοκιμής (test data) (αυτά που θα χρησιμοποιηθούν για την αξιολόγηση της απόδοσης του νευρωνικού δικτύου). Στη συνέχεια, μέσω μιας διαδικασίας που ονομάζεται backpropagation (η ακριβής εξήγηση της οποίας είναι πέρα από το πεδίο αυτής της διατριβής) σε παρτίδες των δεδομένων εκπαίδευσης, τα βάρη ενημερώνονται με τέτοιο τρόπο ώστε να μειωθεί η τιμή της συνάρτησης απώλειας για αυτό το συγκεκριμένο σύνολο δεδομένων. Μόλις όλες οι παρτίδες δεδομένων έχουν περάσει από το δίκτυο και τα βάρη έχουν προσαρμοστεί ανάλογα, λέμε ότι το δίκτυο έχει ολοκληρώσει ένα πέρασμα μέσω των δεδομένων εκπαίδευσης. Αυτή η διαδικασία εκτιμά την κλίση της συνάρτησης απώλειας και έτσι το δίκτυο κατεβαίνει προς το ελάχιστο της συνάρτησης απώλειας. Ο αριθμός των επαναλήψεων αυτής της διαδικασίας ονομάζεται epochs. Ο αριθμός των epochs πρέπει να είναι τέτοιος ώστε να επιτευχθεί μια βέλτιστη λύση χωρίς την υπερπροσαρμογή (overfitting) του δικτύου στα δεδομένα εκπαίδευσης.

Hyperparameter Tuning

Όπως έχουμε δει, για να ορίσουμε ένα νευρωνικό δίκτυο, πρέπει να επιλέξουμε πολλές παραμέτρους εκ των προτέρων. Αυτές οι παράμετροι περιλαμβάνουν τον αριθμό των κρυφών στρωμάτων, τον αριθμό των νευρώνων σε κάθε στρώμα, και τη συνάρτηση ενεργοποίησης κάθε στρώματος, καθώς και πολλές άλλες. Το βέλτιστο σύνολο παραμέτρων δεν είναι γνωστό εκ των προτέρων και δεν υπάρχει μέθοδος επιλογής του.

Η διαδικασία βελτιστοποίησης των υπερπαραμέτρων ενός μοντέλου ονομάζεται *Hyperparameter Tuning*. Υπάρχουν πολλοί αλγόριθμοι βελτιστοποίησης που το επιτυγχάνουν αυτό. Ένας από τους πιο ενδιαφέροντες και αυτός που θα χρησιμοποιηθεί για αυτή την εφαρμογή είναι ο αλγόριθμος HyperBand [9]. Ο αλγόριθμος αυτός είναι μια βελτίωση του αλγόριθμου Successive Halving.

Η ιδέα πίσω από τον αλγόριθμο Successive Halving είναι αρκετά απλή. Κατανέμει ομοιόμορφα έναν προϋπολογισμό πόρων (υπολογιστικού χρόνου) σε ένα σύνολο υπερπαραμέτρων, αξιολογεί την απόδοση κάθε σετ υπερπαραμέτρων, απορρίπτει το χειρότερο 50% και επαναλαμβάνει μέχρι να απομείνει ένα μόνο σετ υπερπαραμέτρων. Το πρόβλημα με αυτόν τον αλγόριθμο είναι ότι δεν είναι γνωστό εάν η δοκιμή ενός μεγαλύτερου αριθμού διαμορφώσεων με περιορισμένους υπολογιστικούς πόρους η καθεμία ή ενός μικρότερου αριθμού διαμορφώσεων με περισσότερους πόρους, θα έχει καλύτερο αποτέλεσμα.

Αυτό είναι το πρόβλημα που αντιμετωπίζει ο αλγόριθμος HyperBand. Η αντιμετώπιση επιτυγχάνεται πραγματοποιώντας ένα grid search για τις εφικτές τιμές των n και r , όπου n είναι ο αριθμός των διαμορφώσεων που θα δοκιμαστούν και r είναι η ελαχιστή ποσότητα υπολογιστικής ισχύος που κατανέμεται σε όλες τις διαμορφώσεις πρωτού δύναται να απορριφθούν.

Κεφάλαιο 3

Μεθοδολογία

Σε αυτό το κεφάλαιο, το θεωρητικό υπόβαθρο που αναπτύχθηκε στο κεφάλαιο 2 θα εφαρμοστεί για την επίλυση ενός πραγματικού προβλήματος. Το κεφάλαιο αυτό εξηγήσει το πρόβλημα που πρέπει να επιλυθεί, τον τρόπο με τον οποίο δημιουργείται το μοντέλο αεροελαστικής ανάλυσης καθώς και τη διαδικασία που χρησιμοποιείται για τη βελτιστοποίηση του.

3.1 Εισαγωγή στο Πρόβλημα

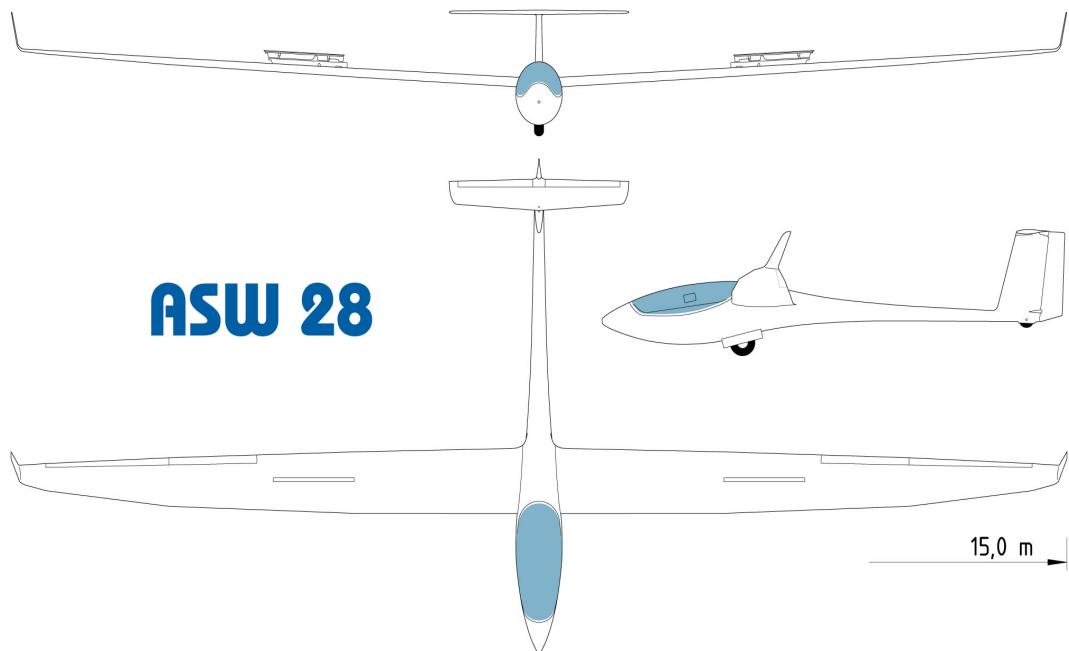
Το πρόβλημα που έχει επιλεγεί ως παράδειγμα εφαρμογής της θεωρίας που αναπτύχθηκε στο προηγούμενο κεφάλαιο είναι η μελέτη των χαρακτηριστικών πτερυγισμού της κύριας πτέρυγας του ανεμοπλάνου ASW 28 και η βελτιστοποίηση του σχεδιασμού, του σύνθετου υλικού του για να προσαρμοστούν τα χαρακτηριστικά του πτερυγισμού σε συγκεκριμένες απαιτήσεις. Η επιλογή αυτού του συγκεκριμένου αεροσκάφους έγινε λόγω του μεγάλου εκπετάσματος σε σχέση με το μήκος της χορδής του aspect ratio, γεγονός που το καθιστά πολύ ευλύγιστο. Επιπλέον, ο κύριος στόχος ενός ανεμοπλάνου είναι να επιτύχει το βέλτιστο λόγο καθόδου glide ratio, επομένως η μείωση του βάρους είναι υψίστης σημασίας για αυτά τα οχήματα κάτι που τα καθιστά ιδανικά για αυτήν την εφαρμογή καθώς συνδυάζουν τη μεγάλη πιθανότητα εμφάνισης πτερυγισμού με την ανάγκη για ελαχιστοποίηση βάρους.

“The ASW 28 is Schleicher’s high-performance glider for the FAI-Standard Class with 15m span.” [3]. Τα τεχνικά χαρακτηριστικά αυτού του αεροσκάφους συνοψίζονται στο Σχήμα 3.1 :

Technical Data

Span	15 m	49.2 ft
Wing area	10.5 m ²	113 sqft
Wing aspect ratio	21.4	
Winglet height	0.5 m	1.64 ft
Fuselage length	6.585 m	21.6 ft
Cockpit width	0.64 m	2.1 ft
Cockpit height	0.8 m	2.62 ft
Height at tail	1.3 m	4.26 ft
Empty mass	235 kg	518 lbs
Max. take-off mass	525 kg	1157 lbs
Min. wing loading	29 kg/m ²	5.93 lbs/sqft
Max. wing loading	50 kg/m ²	10.2 lbs/sqft
Useful load, max.	130 kg	286 lbs
Water ballast wing	180 l	397 lbs
Max. speed	285 km/h	154 kts
Min. sink	0.55 m/s	108 ft/min
Best glide ratio	45	

Σχήμα 3.1: Τεχνικά δεδομένα του ανεμοπλάνου ASW 28 [3]



Σχήμα 3.2: Εμπρόσθια, πλάγια και άνω όψη του ανεμοπλάνου ASW 28 [3]

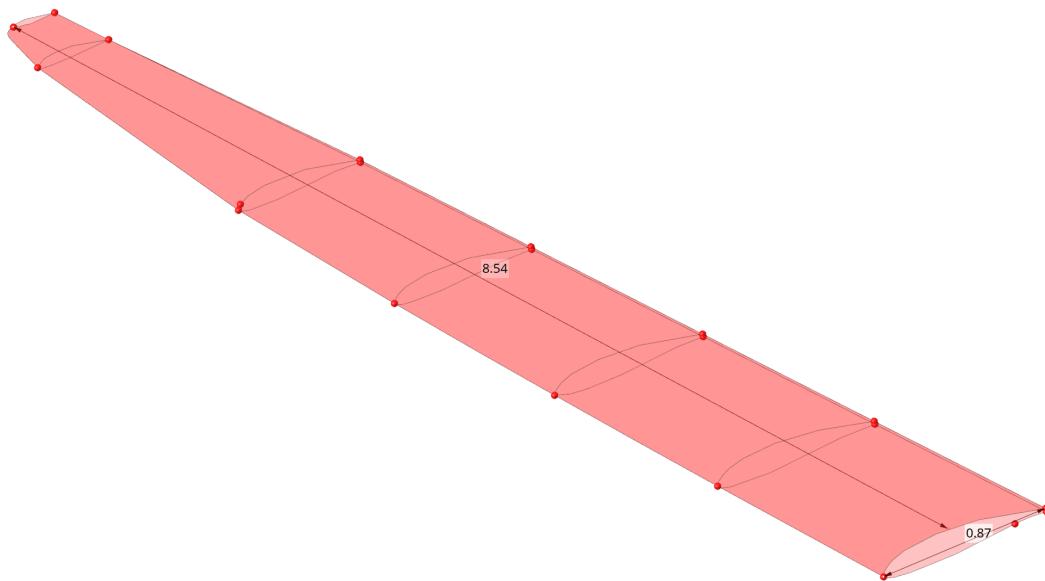
Η ανάλυση της κύριας πτέρυγας του ASW 28 θα περιλαμβάνει τα εξής βήματα:

1. Δημιουργία μοντέλου στο HyperMesh της Altair για ανάλυση ιδιομορφών και πτερυγισμού.
2. Δημιουργία κώδικα σε Python για το αυτόματο διάβασμα των αποτελεσμάτων πτερυγισμού από τον επιλυτή Optistruct.
3. Ορισμός του προβλήματος βελτιστοποίησης χρησιμοποιώντας διάφορες τεχνικές βελτιστοποίησης και υλοποίηση αυτών με Python.

3.2 Μοντέλο Κύριας Πτέρυγας ASW 28

3.2.1 Γεωμετρία Πτέρυγας & Διακριτοποίηση Γεωμετρίας

Η εξωτερική γεωμετρία της πτέρυγας εισήχθη στο HyperMesh από ένα αρχείο CAD. Η γεωμετρία καθαρίστηκε ώστε να περιλαμβάνει μόνο την κύρια πτέρυγα του αεροσκάφους και λόγω ορισμένων ανακριβειών και των μη μετρικών μονάδων μήκους που χρησιμοποιήθηκαν στο CAD, χρειάστηκε να γίνει μετασχηματισμός κλίμακας. Η εξωτερική γεωμετρία παρουσιάζεται στο Σχήμα 3.3

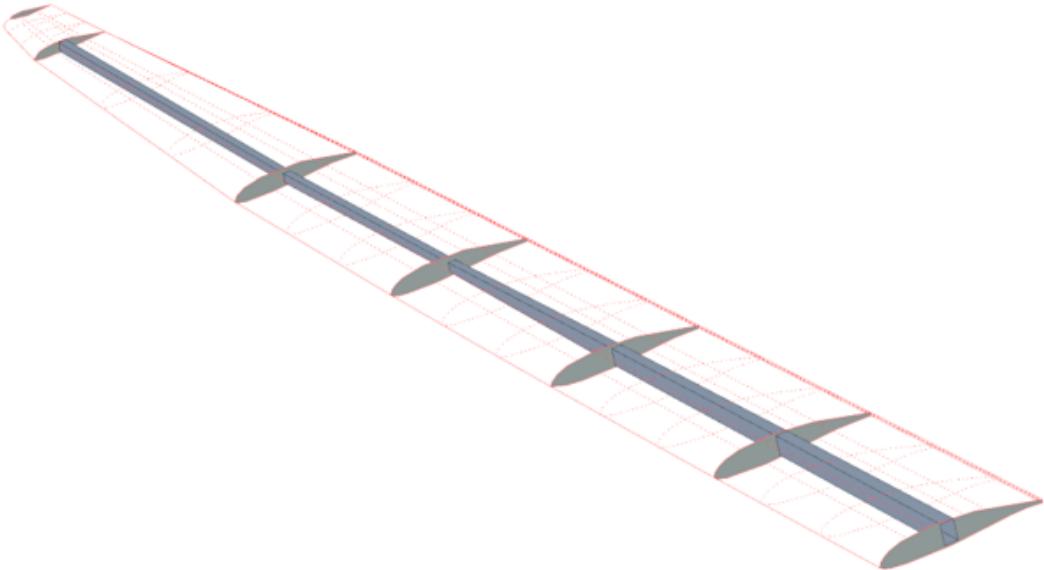


Σχήμα 3.3: Εξωτερική Γεωμετρία της πτέρυγας του ASW 28 (length in meters)

Μετά τη δημιουργία της εξωτερικής γεωμετρίας, η εσωτερική δομή της πτέρυγας έπρεπε να δημιουργηθεί από την αρχή. Εφόσον τα λεπτομερή σχέδια της εσωτερικής δομής δεν είναι διαθέσιμα, η γεωμετρία της εσωτερικής δομής υποτέθηκε με βάση κάποιες φωτογραφίες της αποσυναρμολογημένης πτέρυγας. Τα κύρια χαρακτηριστικά της εσωτερικής γεωμετρίας των περισσότερων σύγχρονων πτερύγων είναι τα spars και τα ribs. Τα spars είναι το κύριο δομικό στοιχείο της πτέρυγας που εκτείνονται κατά μήκος της πτέρυγας και είναι υπεύθυνα για τη μεταφορά των αεροδυναμικών φορτίων στο αεροπλάνο. Τα ribs είναι δομικά μέλη της πτέρυγας που εκτείνονται κατά μήκος της

χορδής και ο σκοπός τους είναι να υποστηρίζουν το εξωτερικό περίβλημα της πτέρυγας έτσι ώστε να διατηρεί το σωστό προφίλ αεροτομής.

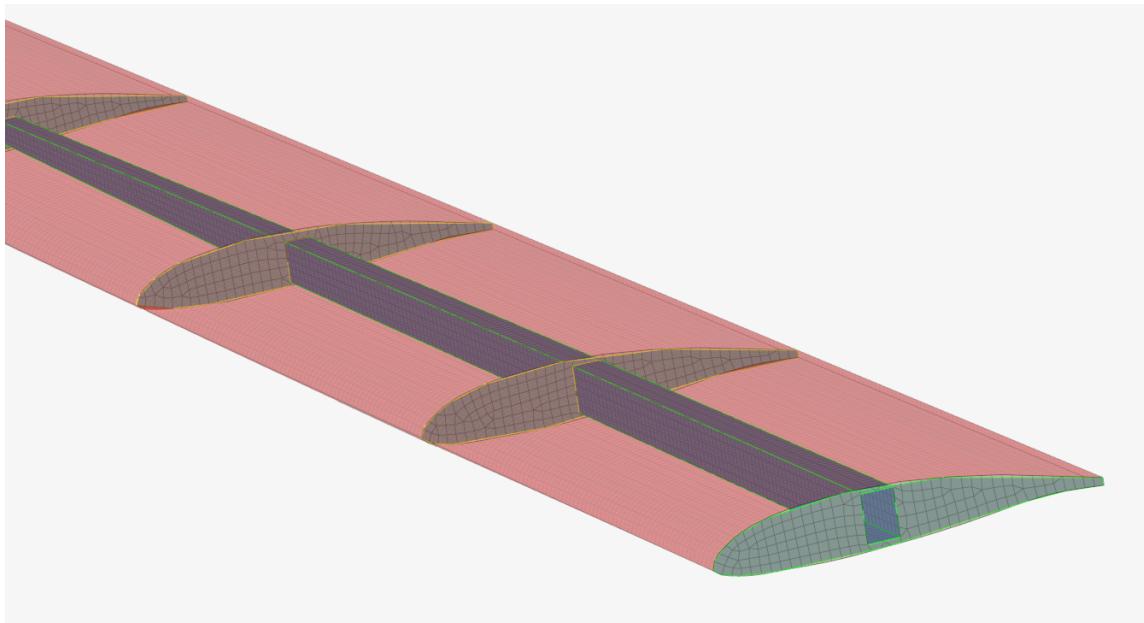
Οι σύγχρονες πτέρυγες ανεμόπτερων συνήθως έχουν μόνο ένα spar και αρκετά ribs. Για αυτήν την εφαρμογή χρησιμοποιούνται ένα spar ορθογώνιας διατομής που στενεύει προς την άκρη της πτέρυγας και 6 ribs που απεικονίζονται στο Σχήμα 3.4.



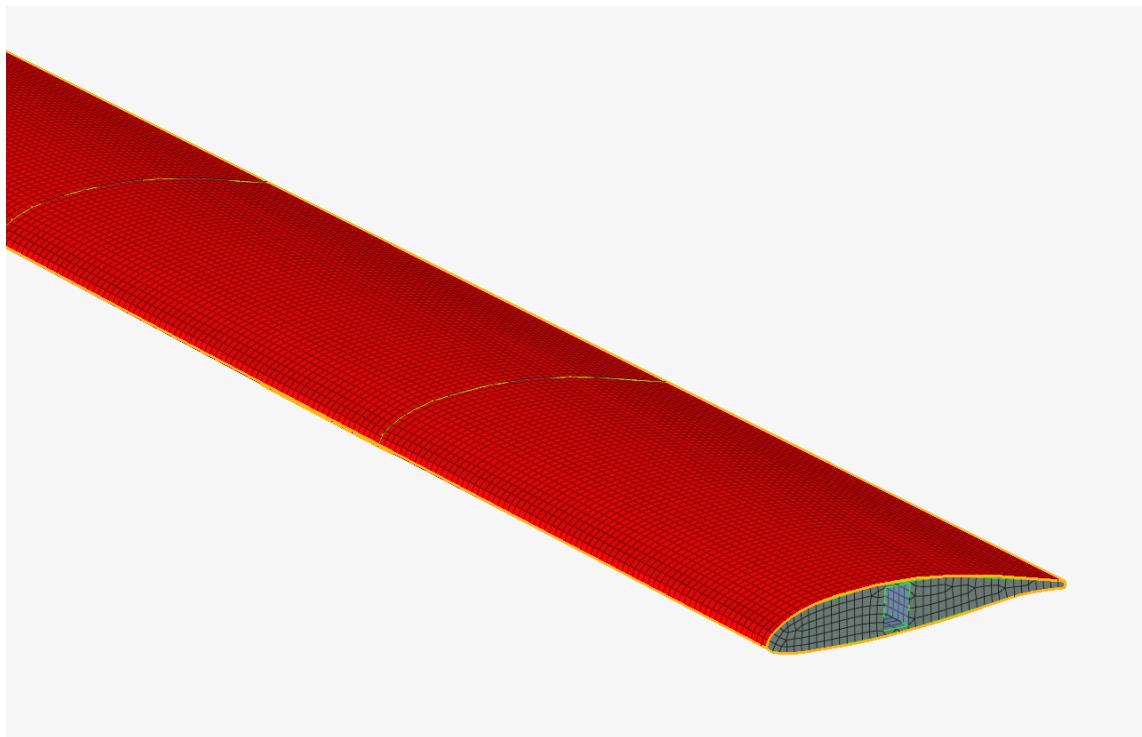
Σχήμα 3.4: Εσωτερική Γεωμετρία της πτέρυγας του ASW 28

Η διακριτοποίηση της γεωμετρίας επιτεύχθηκε χρησιμοποιώντας επιφανειακά στοιχεία που τοποθετήθηκαν σε ένα πλέγμα το οποίο δημιουργήθηκε με τη βοήθεια του εργαλείου “Panel Mesh” του HyperMesh για το περίβλημα της πτέρυγας, καθώς αυτή η λειτουργία είναι ιδιαίτερα κατάλληλη για αυτόν τον τύπο γεωμετρίας που μοιάζει με πάνελ, και του εργαλείου “General 2D Mesh” για τη εσωτερική γεωμετρία. Το μέσο μέγεθος των στοιχείων έχει μήκος πλευράς 0.02 [m].

Η διακριτοποίηση δεν χρειάζεται να είναι πολύ ακριβής, καθώς για την ανάλυση του πτερυγισμού χρειάζεται μόνο τόση λεπτομέρεια ώστε να περιγραφούν οι πρώτες 1-διομορφές ενώ δεν μας ενδιαφέρουν οι ακριβείς τάσεις σε συγκεκριμένα σημεία μέσα στη δομή. Η διακριτοποίηση της γεωμετρίας σε πεπερασμένα στοιχεία φαίνεται στα Σχήματα 3.5 και 3.6.



Σχήμα 3.5: Διακριτοποίηση Εσωτερικής Γεωμετρίας της πτέρυγας του ASW 28



Σχήμα 3.6: Διακριτοποίηση Εξωτερικής Γεωμετρίας της πτέρυγας του ASW 28

3.2.2 Ορισμός Ιδιοτήτων Υλικών

Οι ιδιότητες των υλικών που χρησιμοποιήθηκαν στην πτέρυγα είναι κάπως αυθαίρετες, καθώς αυτές οι πληροφορίες δεν είναι διαθέσιμες δημοσίως. Αυτό δε θα δημιουργήσει προβλήματα στο στόχο της παρούσας διατριβής, καθώς κατά τη διάρκεια της βελτιστοποίησης, οι ιδιότητες των υλικών είναι πιθανό να αλλάξουν ούτως ή άλλως ενώ κύριος σκοπός της διατριβής είναι η ανάπτυξη της μεθοδολογίας και όχι η συγκεκριμένη εφαρμογή.

Ένα κοινό σύνθετο υλικό για το εξωτερικό των ανεμόπτερων υψηλών επιδόσεων είναι το fiberglass ή τα ανθρακονήματα, καθώς αυτά τα υλικά προσφέρουν υψηλή αντοχή, χαμηλό βάρος και μια λεία αεροδυναμική επιφάνεια. Για αυτήν την εφαρμογή επιλέχθηκε ένα σύνθετο υλικό από ανθρακόνημα - Eroxxy. Για τον επιλυτή Optistruct, αυτός ο τύπος σύνθετου υλικού μπορεί να οριστεί χρησιμοποιώντας την κάρτα υλικού MAT8 με τις παρακάτω ιδιότητες [10]:

- $E_1 = 125 \text{ GPa}$

- $E_2 = 8.41 GPa$
- $\nu_{12} = 0.35$
- $G_{12} = 4.23 GPa$
- $\rho = 1517 \text{ kg/m}^3$

Αυτό το υλικό είναι ένα laminated composite και αν και οι ιδιότητες του υλικού έχουν καθοριστεί επαρκώς, η δομή του layup δεν είναι ακόμη ορισμένη. Για να οριστεί το layup, απαιτείται η κάρτα ιδιότητας PCOMP. Αυτή η κάρτα είναι ουσιαστικά μια λίστα που περιέχει τις ιδιότητες κάθε στρώσης του laminate. Οι ιδιότητες αυτές είναι:

1. Το υλικό της στρώσης του laminate ορίζεται ως κάρτα σύνθετου υλικού, στην προκειμένη περίπτωση μια κάρτα MAT8.
2. Το πάχος κάθε στρώσης του laminate ορίζεται ως πραγματικός αριθμός.
3. Η γωνία σε μοίρες του κύριου άξονα (άξονας 1) της στρώσης του laminate.

Για αυτήν την εφαρμογή θα δημιουργηθεί ένα σύνθετο υλικό με έξι στρώσεις laminate χρησιμοποιώντας carbon-fiber epoxy ορθοτροπικό υλικό με γωνίες που εναλλάσσονται μεταξύ $+45$ και -45 μοίρες και πάχος 0.5 mm για κάθε στρώση.

Για την εσωτερική δομή της πτέρυγας χρησιμοποιείται αλουμίνιο. Πιο συγκεκριμένα, χρησιμοποιείται το αλουμίνιο 6061, το οποίο έχει τις παρακάτω ιδιότητες:

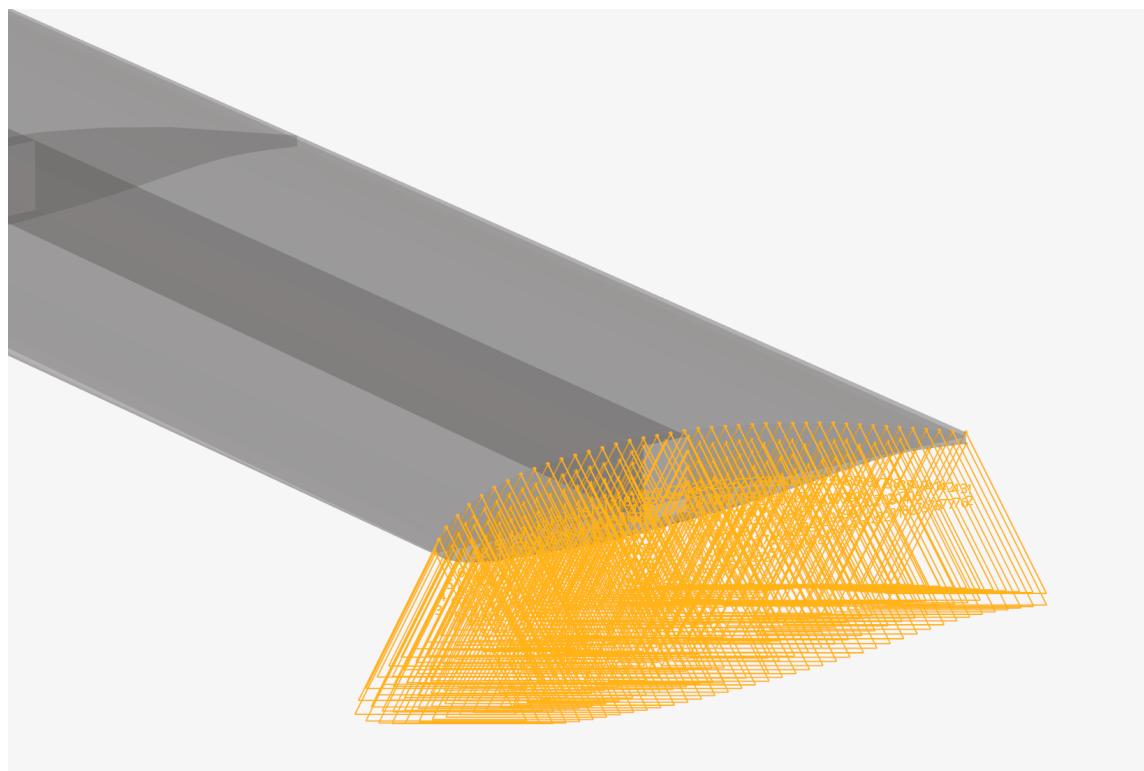
- $E = 69 GPa$
- $\nu = 0.33$
- $\rho = 2700 \text{ kg/m}^3$

Αυτό είναι ένα ισοτροπικό υλικό και ορίζεται στο Optistruct με μια κάρτα MAT1, η οποία ουσιαστικά παραθέτει αυτές τις παραμέτρους του υλικού με συγκεκριμένο τρόπο.

Ομοίως, μόνο ο ορισμός του υλικού δεν είναι επαρκής. Καθώς δουλεύουμε με επιφανειακά στοιχεία, πρέπει να οριστεί και το πάχος τους. Στο Optistruct αυτή η ιδιότητα ορίζεται χρησιμοποιώντας μια κάρτα ιδιότητας PSHELL, η οποία χρειάζεται να περιέχει μόνο μια αναφορά στο υλικό και σε μια τιμή πάχους. Στην προκειμένη περίπτωση, το πάχος είναι ομοιόμορφο παντού και ισούται με $t = 2\text{mm}$.

3.2.3 Συνοριακές Συνθήκες

Οι συνοριακές συνθήκες που ορίζονται για αυτό το πρόβλημα είναι παρόμοιες με αυτές μιας πακτωμένης – ελεύθερης δοκού. Οι κόμβοι στη ρίζα της πτέρυγας πακτώνονται σε κάθε βαθμό ελευθερίας με SPC's (Single Point Constraints). Αυτές οι συνοριακές συνθήκες υποθέτουν ότι η ρίζα της πτέρυγας είναι τελείως σταθερή, κάτι που λειτουργεί καλά για μια αποσπασμένη πτέρυγα που βρίσκεται σε πειραματική διάταξη, αλλά είναι μόνο μια προσέγγιση της πραγματικής κατάστασης κατά την πτήση, όπου η πτέρυγα είναι στερεωμένη στο αεροπλάνο, το οποίο δεν είναι στατικό κατά την πτήση. Οι συνοριακές συνθήκες φαίνονται στο Σχήμα 3.7.



Σχήμα 3.7: Συνοριακές συνθήκες της πτέρυγας του ASW 28

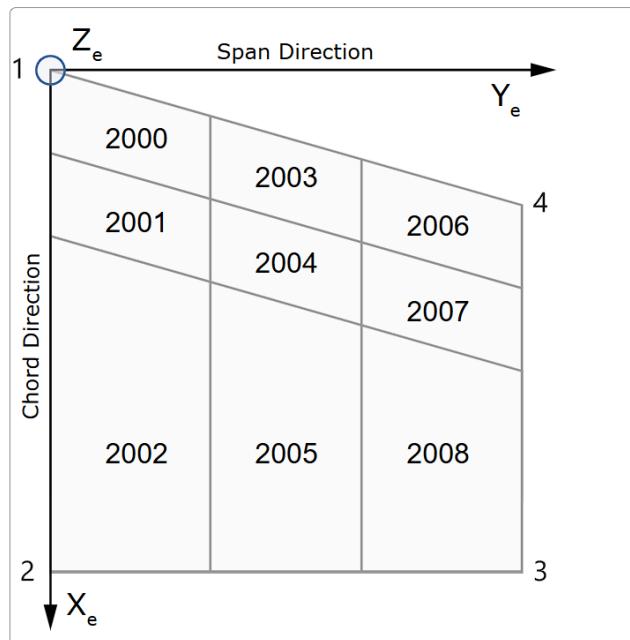
3.2.4 Αεροδυναμικό Πλέγμα

Για να δημιουργηθεί το πλέγμα vortex-lattice, η αεροδυναμική επιφάνεια της πτέρυγας πρέπει να διακριτοποιηθεί. Ο τρόπος με τον οποίο μοντελοποιείται αυτό στο Optistruct

είναι μέσω της κάρτας CAERO1, η οποία ορίζει ένα «αεροδυναμικό μακρο-στοιχείο» με απλή δισδιάστατη γεωμετρία τετραπλεύρου που χωρίζεται σε έναν προεπιλεγμένο αριθμό από boxes στις κατευθύνσεις του άξονα κατα μήκος της χορδής και του διαμήκη άξονα της πτέρυγας. Για να οριστεί η κάρτα CAERO1, πρέπει πρώτα να οριστούν τα τέσσερα σημεία γωνίας με συγκεκριμένη σειρά:

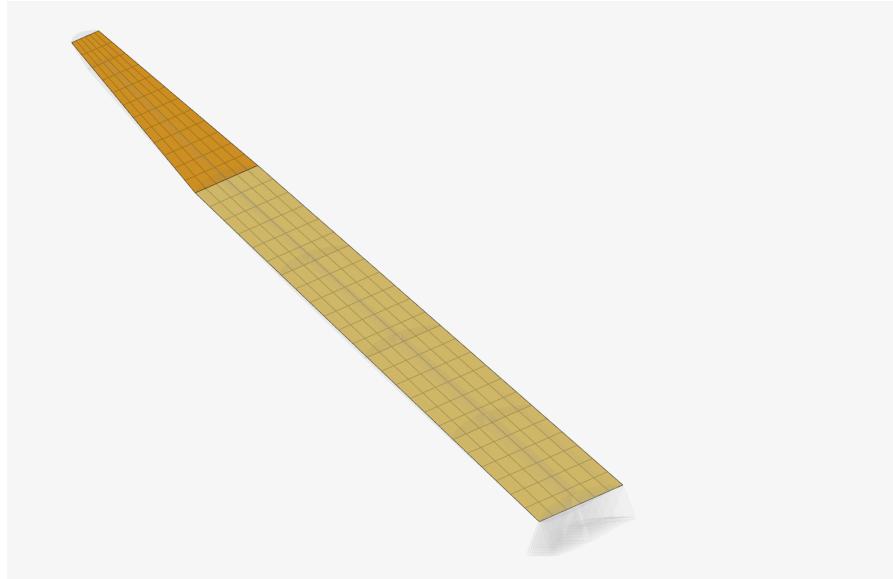
- Το πρώτο σημείο βρίσκεται στην ακμή προσβολής και στη ρίζα της πτέρυγας.
- Το δεύτερο σημείο έχει την ίδια συντεταγμένη y με το πρώτο, αλλά βρίσκεται στην ακμή φυγής της πτέρυγας.
- Το τρίτο σημείο βρίσκεται στην άκρη της πτέρυγας και στην ακμή φυγής.
- Το τέταρτο σημείο έχει την ίδια συντεταγμένη y με το τρίτο, αλλά βρίσκεται στην ακμή προσβολής της άκρης της πτέρυγας.

Τα σημεία αυτά φαίνονται στο Σχήμα 3.8



Σχήμα 3.8: Σύστημα συντεταγμένων ενός CAERO1 Aerodynamic panel [4]

Αυτό επιτρέπει τη μοντελοποίηση μόνο απλών γεωμετριών τραπεζοειδούς σχήματος. Επειδή η προβολή της πτέρυγας του ανεμόπτερου ASW28 στο επίπεδο XY δεν ταιριάζει καλά μέσα σε ένα μόνο τραπεζοειδές σχήμα, χρησιμοποιήθηκαν δύο από αυτά τα μακρο-στοιχεία για να αποτυπωθεί η μεταβαλλόμενη οπισθόκληση αυτής της πτέρυγας. Αυτά τα πάνελ φαίνονται στο Σχήμα 3.9.



Σχήμα 3.9: Τα CAERO1 macro elements της πτέρυγας του ASW 28

Το επόμενο βήμα για τον ορισμό των στοιχείων CAERO1 είναι ο ορισμός της διακριτοποίησης σε αεροδυναμικά boxes μέσω δύο ακέραιων τιμών $NSPAN$ και $NCHORD$, οι οποίες ορίζουν τον αριθμό των boxes κατά μήκος της πτέρυγας και της χορδής αντίστοιχα.

Για το εσωτερικό μακρο-στοιχείο CAERO1:

$$NSPAN = 24, \quad NCHORD = 6$$

Για το εξωτερικό μακρο-στοιχείο "AEPO1":

$$NSPAN = 12, \quad NCHORD = 6$$

Αυτή η διακριτοποίηση επιλέγεται έτσι ώστε ο λόγος διαστάσεων των boxes να είναι μικρότερος από τρία περίπου και το μήκος της χορδής κάθε κουτιού να είναι μικρότερο

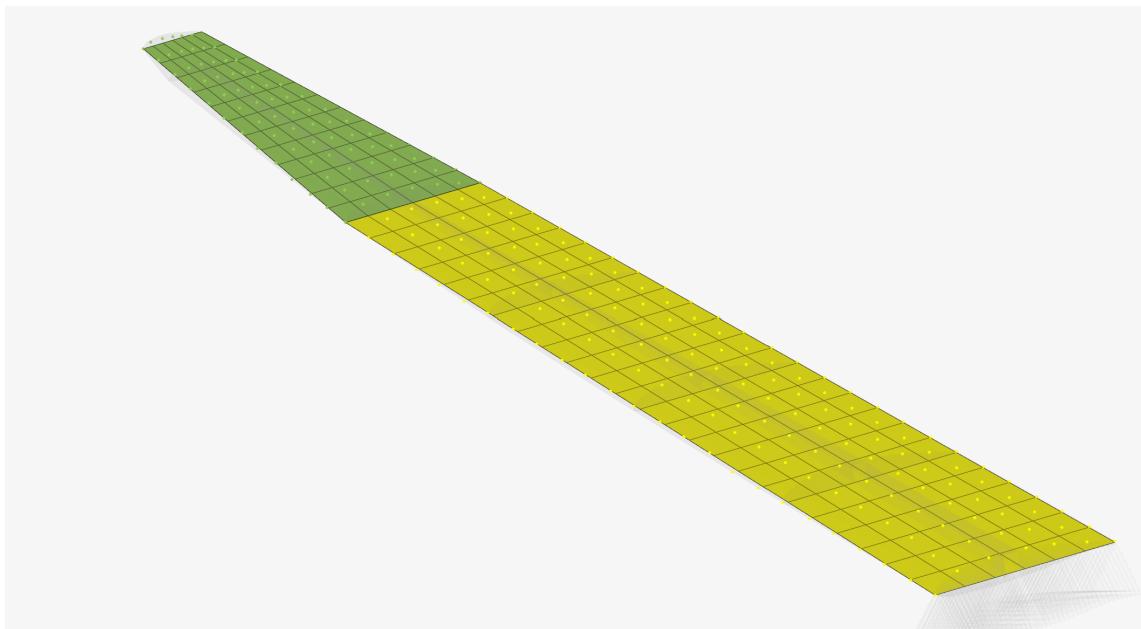
από $\Delta x < 0.08V_{\max} / f_{\max} \approx 0.65m$. Αυτές οι δύο συστάσεις για τη διακριτοποίηση των αεροδυναμικών πάνελ γίνονται στο εγχειρίδιο του Nastran [13].

3.2.5 Η διασυνδετική καμπύλη Spline

Στην ανάλυση πτερυγισμού, η κάρτα SPLINE χρησιμοποιείται για να συνδέσει τον κατασκευαστικό και τον αεροδυναμικό τομέα. Για αυτή την εφαρμογή χρησιμοποιείται η κάρτα SPLINE1, η οποία ορίζει ένα επιφανειακό spline (μονοδιάστατα spline είναι επίσης διαθέσιμα, αλλά δεν εφαρμόζονται σε αυτή την περίπτωση). Για να οριστεί η κάρτα SPLINE1 απαιτούνται οι εξής καταχωρίσεις:

- Το Id του CAERO που ορίστηκε στο προηγούμενο βήμα.
- Τα Id των πρώτων και τελευταίων αεροδυναμικών boxes που πρέπει να συμπεριληφθούν (όλα τα αεροδυναμικά πάνελ επιλέγονται για αυτή την ανάλυση).
- Ένα σύνολο κόμβων από το δομικό πλέγμα.

Η επιλογή των κατασκευαστικών κόμβων είναι σημαντική. Συνήθως, δεν επιλέγονται όλοι οι κόμβοι του μοντέλου. Επιλέγεται μόνο ένα υποσύνολο των κόμβων στην κάτω ή την άνω επιφάνεια της πτέρυγας. Αυτοί οι κόμβοι πρέπει να βρίσκονται στην περιοχή που καλύπτεται από τα CAERO μακρο-στοιχεία και τα αεροδυναμικά boxes που επιλέχθηκαν. Ιδανικά, κάθε αεροδυναμικό σημείο του πλέγματος έχει έναν αντίστοιχο κατασκευαστικό κόμβο ακριβώς από πάνω ή από κάτω του, αν και αυτό δεν είναι εφικτό στις περισσότερες περιπτώσεις.



Σχήμα 3.10: SPLINE1 entries του μοντέλου της πτέρυγας ASW 28

Όπως φαίνεται στο Σχήμα 3.10, δημιουργήθηκαν δύο ξεχωριστά SPLINE1 σε αυτό το μοντέλο, μία για κάθε CAERO μακρο-στοιχείο που ορίστηκε προηγουμένως. Σε αυτή την περίπτωση, επιλέγονται κόμβοι από την άνω επιφάνεια της πτέρυγας. Οι κόμβοι βρίσκονται σε σειρές κατά μήκος του άξονα της πτέρυγας σε διάφορα ποσοστά της χορδής, έτσι ώστε να ταιριάζουν καλά με τα σημεία του αεροδυναμικού πλέγματος.

3.2.6 Setup Αεροελαστικού προβλήματος στο Optistruct

Για το στήσιμο της ανάλυσης πτερυγισμού, πρέπει να οριστούν αρκετές παράμετροι.

The AERO card:

Η κάρτα δεδομένων AERO ορίζει βασικές παραμέτρους για την δυναμική αεροελαστικότητα που αφορούν κυρίως τις συνθήκες πτήσης. Οι εγγραφές αυτής της κάρτας συνοψίζονται ως εξής:

- **VELOCITY:** δεν έχει επίδραση στην ανάλυση Flutter, καθώς η ταχύτητα μεταβάλλεται κατά τη διάρκεια της ανάλυσης και ορίζεται αλλού, αλλά δεν μπορεί

να αφεθεί κενή, οπότε εισάγεται η τιμή μονάδας.

- **REFC:** Το μήκος της χορδής αναφοράς που χρησιμοποιείται για τον υπολογισμό της μειωμένης συχνότητας και των συντελεστών ανύψωσης και αντίστασης, αν ζητηθούν, $REFC = 0.92m$.
- **RHOREF:** Η πυκνότητα αναφοράς $RHOREF = 1.225 \text{ kg/m}^3$, η οποία είναι η πυκνότητα του αέρα στο επίπεδο της θάλασσας.
- **SYMXZ:** Ορίζει την συμμετρία για το επίπεδο XZ και μπορεί να έχει τρεις τιμές
 - -1: για αντισυμετρία.
 - 0: για καμία συμμετρία.
 - 1: για συμμετρία.

Για αυτή την εφαρμογή $SYMZ = 1$

- **SYMXY:** Ορίζει την συμμετρία για το επίπεδο XY με παρόμοιο τρόπο. Στην ανάλυση αυτή $SYMXY = 0$

The MKAERO1 card:

Η κάρτα MKAERO1 είναι μια bulk data entry κάρτα που χρησιμοποιείται για την εισαγωγή ενός πίνακα αριθμών Mach και μειωμένων συχνοτήτων για τις οποίες υπολογίζονται οι αεροδυναμικοί πίνακες.

Η μορφή της κάρτας MKAERO1 έχει δύο στήλες με μέγιστο αριθμό οκτώ στοιχείων η καθεμία. Η μία στήλη είναι για την εισαγωγή του αριθμού Mach, ενώ η άλλη για τις μειωμένες συχνότητες. Οι αεροδυναμικοί πίνακες υπολογίζονται για κάθε ζεύγος τιμών μειωμένης συχνότητας και αριθμού Mach.

Δεν υπάρχουν συγκεκριμένες συστάσεις για το εύρος των μειωμένων συχνοτήτων που πρέπει να καλύπτονται από αυτήν την κάρτα, αλλά καθώς γίνεται παρεμβολή μεταξύ των αεροδυναμικών πινάκων στην πραγματική τελική μειωμένη συχνότητα, η λογική υποδεικνύει ότι το εύρος των μειωμένων συχνοτήτων πρέπει να είναι τουλάχιστον μεγαλύτερο από το αποτέλεσμα του εύρους των μειωμένων συχνοτήτων. Φυσικά, αυτό δεν μπορούμε να το γνωρίζουμε εκ των προτέρων, καθώς οι τελικές μειωμένες συχνότητες καθίστανται γνωστές μόνο αφού εκτελεστεί η ανάλυση. Για αυτή την ανάλυση χρησιμοποιήθηκε ένα μεγάλο εύρος μειωμένων συχνοτήτων όπως προτάθηκε από πολλά παρόμοια παραδείγματα αυτού του τύπου ανάλυσης.

Σε περίπτωση που απαιτούνται περισσότερες από οκτώ τιμές για τη μειωμένη συχνότητα ή τον αριθμό Mach, μπορεί να εισχθεί μια δεύτερη κάρτα MKAERO1.

Οι τιμές που χρησιμοποιήθηκαν είναι οι εξής:

$$\overrightarrow{M} = 0.0, \quad \overrightarrow{K} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.8 \\ 1.6 \\ 3.2 \\ 6.4 \\ 10 \\ 14 \end{bmatrix}$$

Όπως φαίνεται, υπάρχει μόνο ένας αριθμός Mach $M = 0$, που σημαίνει ότι υποθέτουμε ασυμπίεστο ρευστό. Οι αεροδυναμικοί πίνακες υπολογίζονται σε κάθε σημείο (M_i, K_j)

The FFLFACT card:

Η κάρτα δεδομένων βυλκ bulk FFLFACT είναι μια κάρτα που καθορίζει μια σειρά αεροδυναμικών παραμέτρων. Αυτές οι παράμετροι χρησιμοποιούνται για να ορίσουν:

1. Τις αναλογίες πυκνότητας.
2. Τους αριθμούς Mach.
3. Τις μειωμένες συχνότητες ή ταχύτητες (μόνο για την μέθοδο PK).

Αυτοί οι παράγοντες μπορούν να οριστούν χρησιμοποιώντας δύο διαφορετικές μορφές.

1. Στη Μορφή 1, μια σειρά τιμών εισάγεται απευθείας στην κάρτα.
2. Στη Μορφή 2 χρησιμοποιείται η λεγόμενη μορφή THRU για τον ορισμό μιας σειράς τιμών, χρησιμοποιώντας:
 - α' . $F1$: Ο πρώτος παράγοντας.
 - β' . FNF : Ο τελικός παράγοντας.

γ'. NF : Ο αριθμός των παραμέτρων (ακέραιος).

δ'. $FMID$: Ο ενδιάμεσος αεροδυναμικός παράγοντας.

Η πραγματική σειρά τιμών που παράγεται όταν χρησιμοποιείται η μορφή THRU υπολογίζεται χρησιμοποιώντας τον παρακάτω τύπο:

$$\frac{F1(FNF - FMID)(NF - i) + FNF(FMID - F1)(i - 1)}{(FNF - FMID)(NF - i) + (FMID - F1)(i - 1)}, \text{ Όπου } i = 1, 2, \dots, NF \quad (3.1)$$

Σημειωτέον ότι όταν $FMID = \frac{F1+FNF}{2}$ οι παράγοντες κατανέμονται εξίσου μεταξύ $F1$ και FNF .

Για αυτήν την ανάλυση απαιτούνται τρεις κάρτες FFACT:

- **FFACT 1:** Είναι ο συντελεστής πυκνότητας και έχει τιμή 1. Αυτός ο συντελεστής είναι πολλαπλασιαστής της αναφορικής πυκνότητας που ορίζεται στην κάρτα AERO και υποδεικνύει ότι η ανάλυση θα εκτελεστεί με πυκνότητα $1 \times RHOREF$.
- **FFACT 2:** Είναι ο αριθμός Mach και έχει τιμή 0. Αυτός ο συντελεστής είναι ο αριθμός Mach στον οποίο θα εκτελεστεί η ανάλυση .
- **FFACT 3:** Είναι η ταχύτητα/ες στις οποίες θα εκτελεστεί η ανάλυση . Αυτή η FFACT ορίζεται χρησιμοποιώντας τη Μορφή THRU με παράγοντες:

- $F1 = 20m/s$
- $FNF = 320m/s$
- $NF = 30$
- $FMID = 160$

Η ανάλυση εκτελείται για κάθε συνδυασμό πυκνότητας, αριθμού Mach και ταχύτητας στις κάρτες FFACT.

The Flutter card:

Η bulk data entry card Flutter καθορίζει τη μέθοδο και τις παραμέτρους της αεροελαστικής ανάλυσης πτερυγισμού:

Τα πιο σημαντικά πεδία αυτής της κάρτας είναι:

1. **METHOD:** Η μέθοδος μπορεί να είναι μία από τις K , PK , $PKNL$, KE . Για αυτήν την ανάλυση χρησιμοποιείται η μέθοδος PK .
2. **DENS:** Αναφορά στην κάρτα δεδομένων FLFACΤ που καθορίζει τους πολλαπλασιαστές πυκνότητας.
3. **MACH:** Αναφορά στην κάρτα δεδομένων FLFACΤ που καθορίζει τον αριθμό Mach.
4. **VEL:** Αναφορά στην κάρτα δεδομένων FLFACΤ που καθορίζει τις ταχύτητες.
5. **IMETH:** Η μέθοδος παρεμβολής για τον αεροδυναμικό πίνακα, η οποία μπορεί να είναι είτε Λ είτε Σ για γραμμική ή επιφανειακή παρεμβολή αντίστοιχα. Η προεπιλεγμένη τιμή Λ διατηρείται για αυτήν την ανάλυση.

The EIGRL card:

Η bulk data entry card EIGRL ορίζει τα δεδομένα που απαιτούνται για την εκτέλεση ανάλυσης πραγματικών ιδιοτιμών με τη μέθοδο Lanczos. Τα κύρια πεδία αυτής της κάρτας είναι:

1. $V1, V2$ Εύρος συχνοτήτων της ανάλυσης.
2. ND Αριθμός επιθυμητών ιδιοσυχνοτήτων.

Για αυτήν την ανάλυση, χρησιμοποιούνται οι τιμές $V1 = 0.0 \text{ Hz}$ και $ND = 8$. Αυτές οι τιμές σημαίνουν ότι θα υπολογιστούν οι πρώτες οκτώ ιδιοσυχνότητες ξεκινώντας από 0 Hz .

Subcase Definition:

Για τον ορισμό του subcase, επιλέγεται ο τύπος ανάλυσης “Aeroelastic Flutter” και στη συνέχεια:

- Το πεδίο FMETHOD απαιτεί αναφορά στην κάρτα Flutter.
- Το πεδίο SPC απαιτεί αναφορά στον load collector SPC.
- Το πεδίο METHOD απαιτεί αναφορά στο EIGRL.
- Το πεδίο CMETHOD απαιτεί αναφορά σε μια κάρτα EIGC αλλά μπορεί να παραμείνει κενό αν δεν χρειάζεται να υπολογιστούν σύνθετες ιδιοτιμές (είναι κενό σε αυτή την περίπτωση καθώς δεν λαμβάνεται υπόψη η δομική απόσβεση).
- Το πεδίο SMETHOD απαιτεί αναφορά σε καμπύλη απόσβεσης TDMP αλλά μπορεί να παραμείνει κενό εφόσον δεν λαμβάνεται υπόψη η δομική απόσβεση (είναι κενό σε αυτή την περίπτωση καθώς δεν λαμβάνεται υπόψη η δομική απόσβεση).

Τα αποτελέσματα της αεροελαστικής ανάλυσης πτερυγισμού παρουσιάζονται στην ενότητα 4.2.

3.3 Optistruct – Διεπαφή με Python

3.3.1 Αποτελέσματα Ανάλυσης Πτερυγισμού & Python

Η έξοδος της ανάλυσης που ορίζεται στην προηγούμενη ενότητα είναι ένα αρχείο κειμένου .flt εκτός από το τυπικό αρχείο εξόδου .out του Optistruct. Έχει πολύ συγκεκριμένη μορφή. Είναι οργανωμένο σε μπλοκ πληροφοριών, κάθε ένα από τα οποία αντιστοιχεί σε ένα διαφορετικό συνδυασμό αριθμού Mach, πυκνότητας (ορισμένα στις κάρτες FLMFACT) και αριθμού ιδιομορφής. Τα πιο σημαντικά χαρακτηριστικά του αρχείου .flt περογράφονται στο Σχήμα 3.11.

Velocity, Mach Number and Density ratio defined in the FLMFACT entries						
Subcase =	2	POINT =	1	MACH NUMBER =	0.0000	Method of Flutter analysis
KFREQ	1./KFREQ	VELOCITY	DAMPING	FREQUENCY	COMPLEX	EIGENVALUE
0.10000	4.625000E+00	2.500000E+01	-5.231000E+01	1.400000E+00	-2.144000E+00	9.1440232E+00
0.1457	6.057000E+00	2.900000E+01	-1.453165E+00	3.671552E+00	-3.671552E+00	9.1440232E+00
0.1850	9.523122E+00	3.826807E+01	-1.109975E+00	1.398845E+00	-4.8739445E+00	8.7892046E+00
0.0880	1.258648E+00	4.7510917E+01	-1.4626258E+00	3.3227205E+00	-6.0778671E+00	8.3108980E+00
0.0618	1.6185957E+01	5.6255930E+00	-1.8968865E+00	1.2227797E+00	-7.2868435E+00	7.68279514E+00
0.0473	2.1127753E+01	7.9700000E+00	-2.4785468E+00	1.0919222E+00	-8.5023442E+00	6.8607494E+00
0.0377	2.7300000E+01	9.3700000E+00	-3.1227797E+00	9.0000000E+00	-9.0000000E+00	5.7577550E+00
0.0232	4.5017281E+01	8.5333333E+01	-5.2631474E+00	6.6000052E-01	-1.0951307E+01	4.4272398E+00
0.0000	1.0000000E+25	9.5000000E+01	-3.1550493E-01	1.5176463E-10	-1.1291130E+01	9.5356529E-10
0.0000	1.0000000E+25	1.0475336E+02	-2.2597893E-01	1.0432149E-14	-8.8819608E+00	6.5547125E-14
0.0000	1.0000000E+25	1.1459459E+02	-1.8858275E-01	-1.4898091E-14	-8.1409245E+00	-9.3607476E-14
0.0000	1.0000000E+25	1.2452489E+02	-1.6516519E-01	2.0723550E-14	-7.7478699E+00	1.3820990E-13
0.0000	1.0000000E+25	1.3450000E+02	-1.4465231E-01	3.8012312E-14	-7.0000000E+00	-1.3820990E-13
0.0000	1.0000000E+25	1.4455753E+02	-1.3579500E-01	-5.479574E-13	-7.4801786E+00	5.0000000E-12
0.0000	1.0000000E+25	1.5486239E+02	-1.2586548E-01	-2.5052247E-14	-7.3427747E+00	-5.7348791E-13
0.0000	1.0000000E+25	1.6516329E+02	-1.1783725E-01	-3.0895133E-14	-7.3315956E+00	-1.9411985E-13
0.0000	1.0000000E+25	1.755556E+02	-1.1197759E-01	-1.7973440E-15	-7.3358978E+00	-1.8727559E-14
0.0000	1.0000000E+25	1.869464E+02	-1.0559953E-01	7.4819355E-15	-7.4818127E+00	4.7810168E-14
0.0000	1.0000000E+25	1.983351E+02	-1.0000000E-01	3.0236667E-13	-7.4818127E+00	-2.2285000E-13
0.0000	1.0000000E+25	2.0732394E+02	-9.6625955E-02	1.9723895E-13	-7.5465232E+00	6.0000000E-13
0.0000	1.0000000E+25	2.1811321E+02	-9.2944545E-02	7.9466501E-13	-7.6368368E+00	4.9938275E-12
0.0000	1.0000000E+25	2.2900474E+02	-8.9661653E-02	-1.6609017E-13	-7.7349738E+00	-1.0435753E-12
0.0000	1.0000000E+25	2.4000000E+02	-8.6702139E-02	1.0814280E-13	-7.8387839E+00	6.7948127E-13
0.0000	1.0000000E+25	2.5110600E+02	-8.4005484E-02	1.1971308E-13	-7.9055690E+00	7.5216227E-13
0.0000	1.0000000E+25	2.6221190E+02	-8.1353548E-02	-9.2490902E-13	-8.0569053E+00	5.3880000E-13
0.0000	1.0000000E+25	2.7362159E+02	-7.9249090E-02	-3.9526834E-14	8.1587235E+00	2.4825442E-13
0.0000	1.0000000E+25	2.8504854E+02	-7.7119639E-02	-3.1594018E-14	-8.1821653E+00	-1.9851107E-13
0.0000	1.0000000E+25	2.9658537E+02	-7.5125653E-02	-3.8102498E-14	-8.3935487E+00	-2.3948499E-13
0.0000	1.0000000E+25	3.0823590E+02	-7.3477641E-02	-1.7144259E-14	-8.5053994E+00	1.8317529E-13
0.0000	1.0000000E+25	3.2000000E+02	-7.1477641E-02	-1.7144259E-14	-8.6164391E+00	-7.3791353E-14

Σχήμα 3.11: Επεξήγηση του Optistruct .flt file

Το τυπικό αρχείο .out, το οποίο επίσης εξάγεται, περιέχει πληροφορίες σχετικά με το αν ο επιλυτής ενέπεσε σε σφάλμα ή προειδοποιήσεις και, το πιο σημαντικό, πληροφορίες σχετικά με τη μάζα του μοντέλου.

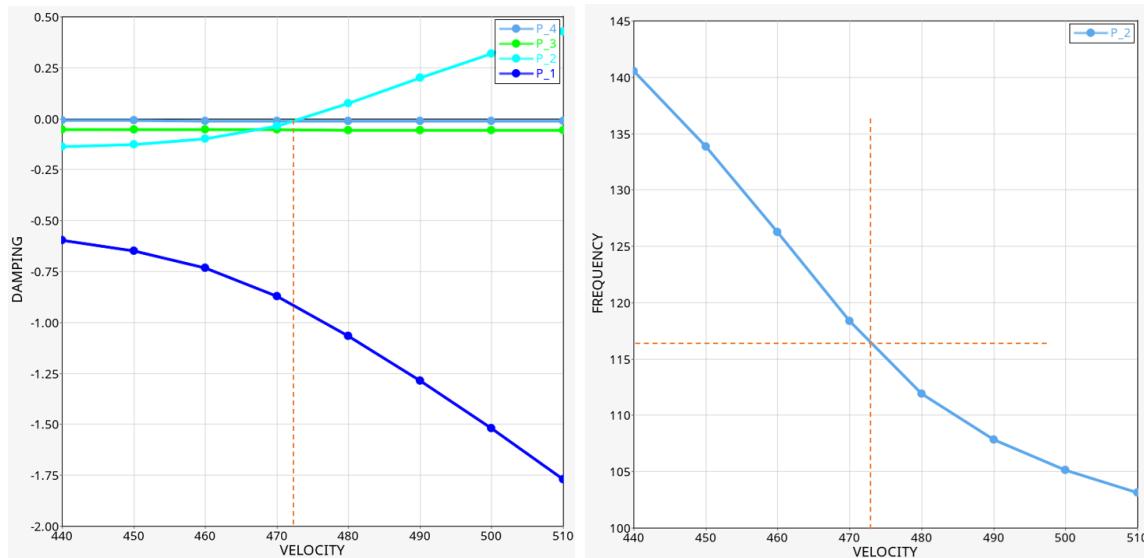
Από τα δεδομένα που περιέχονται στο αρχείο .flt, μπορούν να παραχθούν δύο πολύ χρήσιμα διαγράμματα.

1. Το διάγραμμα $V - g$ απεικονίζει την απόσβεση κάθε ιδιομορφής (λ ξονας y) ως

προς την Ταχύτητα (άξονας x).

2. Το διάγραμμα $V - f$ απεικονίζει τη συχνότητα κάθε ιδιομορφής (άξονας y) ως προς την Ταχύτητα (άξονας x).

Ένα τυπικό παράδειγμα αυτών των διαγραμμάτων φαίνεται στο Σχήμα 3.12



Σχήμα 3.12: Παραδείγματα από V-g και V-f plots με τέσσερις ιδιομορφές [4]

Από αυτά τα δεδομένα μπορεί να προσδιοριστεί η ταχύτητα πτερυγισμού εξετάζοντας το διάγραμμα $V - g$. Πιο συγκεκριμένα, μια ιδιομορφή αποκλίνει όταν η απόσβεσή της γίνεται από αρνητική, θετική. Πολλές ιδιομορφές μπορεί να αποκλίνουν, αλλά η ταχύτητα πτερυγισμού του μποντέλου προσδιορίζεται από την ιδιομορφή που αποκλίνει στη χαμηλότερη ταχύτητα.

Τα δεδομένα που περιέχονται στο αρχείο .flt διαβάζονται από ένα πρόγραμμα Python και αποθηκεύονται ως pandas data frames. Για να προσδιοριστεί η ταχύτητα πτερυγισμού, παρακολουθούνται οι αλλαγές του προσήματος της απόσβεσης κάθε ιδιομορφής μέχρι να ανιχνευτεί μια αλλαγή από αρνητικό σε θετικό πρόσημο. Στη συνέχεια, καταγράφονται όλα τα σημεία όπου παρατηρείται τέτοια αλλαγή και η ταχύτητα πτερυγισμού προσδιορίζεται από την ελάχιστη ταχύτητα στην οποία συμβαίνει αυτή η αλλαγή.

Μια άλλη συνάρτηση είναι υπεύθυνη για την ανάγνωση του αρχείου .out από όπου

καθορίζεται αν ο επιλυτής ολοκλήρωσε την ανάλυση με επιτυχία καθώς και τη μάζας του μοντέλου.

3.3.2 Τροποποίηση της εισόδου του Optistruct χρησιμοποιώντας Python

Για να γίνει βελτιστοποίηση του σύνθετου υλικού της πτέρυγας, είναι απαραίτητο να τροποποιηθούν οι ιδιοτήτες του σύνθετου υλικού προγραμματιστικά, έτσι ώστε να δοκιμαστούν πολλές διαφορετικές παραλλαγές και να φτάσουμε σε μια βέλτιστη λύση. Το Optistruct διευκολύνει αυτή τη διαδικασία αρκετά, διότι χρησιμοποιεί ένα αρχείο κειμένου για είσοδο, που κωδικοποιεί όλες τις πληροφορίες του μοντέλου. Αυτό το αρχείο ονομάζεται .fem και μπορεί να εξαχθεί από το HyperMesh με μορφή csv. Για να τροποποιηθούν οι ιδιότητες του σύνθετου υλικού, εντοπίζεται το σωστό τμήμα αυτού του αρχείου και αποκωδικοποιείται. Σύμφωνα με το εγχειρίδιο του Optistruct, οι ιδιότητες του σύνθετου υλικού κωδικοποιούνται με τον τρόπο που φαίνεται στον Πίνακα 3.1.

1	2	3	4	5	6	7	8	9	10
<i>PCOMP</i>	<i>PID</i>	<i>Z0</i>	<i>NSM</i>	<i>SB</i>	<i>FT</i>	<i>TREF</i>	<i>GE</i>	<i>LAM</i>	+
	<i>MID1</i>	<i>T1</i>	<i>THETA1</i>	<i>SOUT1</i>	<i>MID2</i>	<i>T2</i>	<i>THETA2</i>	<i>SOUT2</i>	+
	<i>MID3</i>	<i>T3</i>	<i>THETA3</i>	<i>SOUT3</i>	<i>Etc.</i>				

Πίνακας 3.1: Η bulk data entry κάρτα PCOMP του Optistruct

Όπου:

- *PCOMP*: είναι η λέξη-κλειδί που υποδεικνύει ότι ακολουθεί ορισμός ιδιοτήτων σύνθετου υλικού (string).
- *PID*: είναι το ID της ιδιότητας (integer).
- *NSM*: είναι η μη-δομική μάζα ανά μονάδα επιφάνειας (float).
- *SB*: επιτρεπτό μέγιστο τόξο ενδο-στρώματος (default 0.0) (float).
- *FT*: Επιλογή θεωρίας αστοχίας.
- *TREF*: Θερμοκρασία χωρίς τάση (float).
- *GE*: Συντελεστής απόσβεσης (float).
- *LAM*: Επιλογή laminate, διάφοροι τρόποι ορισμού του laminate. Στην προεπιλεγμένη περίπτωση, όλα τα plies πρέπει να οριστούν ένα προς ένα.

- $MIDi$: Το ID του υλικού του ply i (integer).
- Ti : Το πάχος του ply i.
- $THETAi$: Η γωνία του ply i.
- $SOUTi$: Αίτημα εξόδου τάσης, παραμόρφωσης, default False, (bool).

Οι μεταβλητές που πρέπει να τροποποιηθούν είναι κυρίως οι Ti και $THETAi$.

3.4 Πρόβλημα Βελτιστοποίησης

Σε αυτή την ενότητα θα συζητηθεί η εφαρμογή των αλγορίθμων βελτιστοποίησης που αναφέρθηκαν στην ενότητα 2.4.

3.4.1 Εφαρμογή της Μεθόδου Powell

Αυτός ο αλγόριθμος υλοποιήθηκε χρησιμοποιώντας τη συνάρτηση `minimize` της *SciPy* [15] επιλέγοντας την προαιρετική παράμετρο `method = "Powell"`. Για να οριστεί πλήρως το πρόβλημα βελτιστοποίησης, πρέπει να οριστούν και μερικές ακόμα συμαντικές παράμετροι.

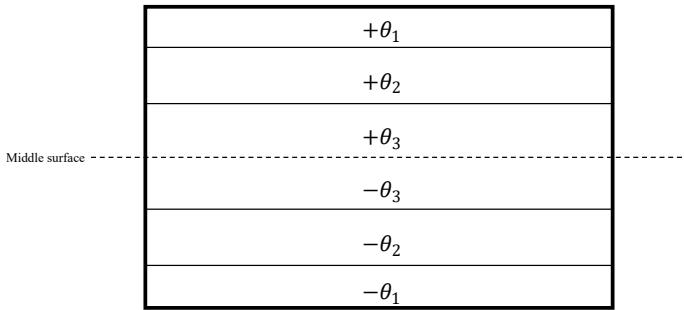
Μεταβλητές Απόφασης:

Το πρώτο βήμα είναι ο ορισμός των μεταβλητών απόφασης. Αυτές οι μεταβλητές είναι το πάχος και η γωνία κάθε στρώσης `ply` του σύνθετου υλικού. Επειδή το laminate αποτελείται από έξι στρώσεις, αυτό σημαίνει ότι θα πρέπει να χρησιμοποιηθούν 12 μεταβλητές απόφασης.

Λόγω του υπολογιστικού χρόνου που απαιτείται, πρέπει να γίνουν κάποιες υποθέσεις για να μειωθεί η πολυπλοκότητα του προβλήματος βελτιστοποίησης. Αυτές οι υποθέσεις είναι:

- Οι στρώσεις είναι αντισυμμετρικές.

Αυτό σημαίνει ότι για κάθε στρώση σε ύψος z_k με γωνία θ_k πάνω από τη μεσαία επιφάνεια, υπάρχει μια αντίστοιχη στρώση με αντίθετη γωνία $-\theta_k$ σε ύψος $-z_k$ από τη μεσαία επιφάνεια, όπως φαίνεται στο Σχήμα 3.13



Σχήμα 3.13: Αντισυμμετρικά στρώματα σύνθετου υλικού

Αυτή η υπόθεση μειώνει τον αριθμό των μεταβλητών απόφασης για τις γωνίες στρώσης στο μισό, διότι από τις αρχικές έξι ανεξάρτητες μεταβλητές, μόνο τρεις χρειάζεται να οριστούν για το laminate έξι στρώσεων $[\theta_1, \theta_2, \theta_3]$.

Αυτή η υπόθεση είναι λογική, καθώς τα περισσότερα σύνθετα υλικά είναι είτε συμμετρικά είτε αντισυμμετρικά προκειμένου να μειώσουν την επίδραση του membrane - bending coupling.

- Κάθε στρώση έχει το ίδιο πάχος.

Αυτή η υπόθεση μειώνει τον αριθμό των ανεξάρτητων μεταβλητών από έξι σε μία, αφού μόνο ένα πάχος $[t]$ χρειάζεται να οριστεί.

Αυτή η υπόθεση είναι επίσης αρκετά λογική, καθώς κατά την κατασκευή σύνθετων μελών, κάθε στρώση προέρχεται από τον ίδιο ρολό ανθρακονήματος, το οποίο έχει ομοιόμορφο πάχος σε όλη του την επιφάνεια.

Αυτές οι δύο υποθέσεις οδηγούν στις τελικές μεταβλητές απόφασης που είναι:

$$\vec{x} = [t, \theta_1, \theta_2, \theta_3]^T \quad (3.2)$$

Αντικειμενική Συνάρτηση:

Στη συνέχεια ορίζεται η αντικειμενική συνάρτηση. Εξετάστηκαν δύο διαφορετικές περιπτώσεις για αυτή τη συνάρτηση.

- Περίπτωση 1:** Η συνάρτηση στόχου εξετάζει τόσο τη μάζα όσο και την ταχύτητα πτερυγισμού. Η ιδέα πίσω από τον ορισμό αυτής της αντικειμενικής συνάρτησης είναι, να είναι δυνατός ο περιορισμός της μάζας της κατασκευής ενώ διατηρείται μια επαρκής ταχύτητα πτερυγισμού. Αυτό κανονικά θα ανήκε στην κατηγορία της πολυδιάστατης βελτιστοποίησης αλλά η μέθοδος Powell δεν έχει αυτή τη δυνατότητα, οπότε χρησιμοποιείται μια εναλλακτική λύση. Η μέθοδος ποινής πρέπει να εφαρμοστεί, οδηγώντας έτσι στον εξής ορισμό.

$$f_{obj}(\vec{x}) = \begin{cases} M, & V_{flutter} > V_{limit} \\ M + P \cdot (V_{limit} - V_{flutter}), & V_{flutter} < V_{limit} \end{cases} \quad (3.3)$$

Όπου:

- M : είναι η μάζα.
- P : είναι μια μεγάλη σταθερά που ονομάζεται ποινή (Penalty).
- V_{limit} : είναι το όριο κάτω από το οποίο η ταχύτητα πτερυγισμού της πτέρυγας θεωρείται ανεπίτρεπτα χαμηλή.
- $V_{flutter}$: είναι η ταχύτητα πτερυγισμού που προκύπτει από το Optistruct.

Αυτός ο ορισμός επιτρέπει την ελαχιστοποίηση της μάζας εξασφαλίζοντας ότι η ταχύτητα πτερυγισμού παραμένει πάνω από ένα ορισμένο όριο. Εάν η ταχύτητα πτερυγισμού πέσει κάτω από το καθορισμένο όριο, προστίθεται ένας όρος ποινής ανάλογος με την ποσότητα με την οποία παραβιάζεται ο περιορισμός, έτσι ώστε ο αλγόριθμος βελτιστοποίησης να αναγκαστεί να επιστρέψει σε μια περιοχή όπου ο περιορισμός δεν παραβιάζεται.

- Περίπτωση 2:** Το απλούστερο σενάριο είναι μια αντικειμενική συνάρτηση όπου η είσοδος είναι το διάνυσμα μεταβλητών απόφασης από την εξίσωση (3.2), χωρίς το πάχος t , και η έξοδος είναι η αρνητική ταχύτητα πτερυγισμού που υπολογίζεται από το Optistruct.

$$f_{obj}(\vec{x}) = -V_{flutter}, \quad \text{με } \vec{x} = [\theta_1, \theta_2, \theta_3]^T \quad (3.4)$$

Το αρνητικό πρόσημο στην ταχύτητα είναι απαραίτητο για να γίνει η ελαχιστοποίηση προς την κατεύθυνση της αύξησης της ταχύτητας πτερυγισμού. Εφόσον η μάζα δεν εξετάζεται στην αντικειμενική συνάρτηση, ο αλγόριθμος βελτιστοποίησης θα μπορούσε

απλώς να αυξήσει το πάχος του υλικού και να επιτύχει υψηλότερη ταχύτητα πτερυγισμού. Αυτή η συμπεριφορά είναι φυσικά ανεπιθύμητη και είναι ο λόγος που το πάχος αφαιρέθηκε από τις μεταβλητές βελτιστοποίησης και τώρα παραμένει σταθερό καθ' όλη τη διάρκεια της βελτιστοποίησης, μαζί με τη μάζα της πτέρυγας.

Search Space Boundaries

Για να οριστεί πλήρως το πρόβλημα βελτιστοποίησης, πρέπει να οριστεί και ο χώρος αναζήτησης. Ο ορισμός των ορίων είναι αρκετά απλός σε αυτήν την περίπτωση.

- **Για την Περίπτωση 1:**

Το πάχος και οι γωνίες πρέπει να περιορισθούν. Τα όρια για το πάχος ορίζονται να είναι εντός ενός λογικού εύρους από 0.1 έως 1mm ενώ για τις γωνίες περιορίζονται μεταξύ -90 και +90 μοιρών.

$$t \in [10^{-4}, 10^{-3}] \quad \text{and} \quad \theta_1, \theta_2, \theta_3 \in [-90^\circ, +90^\circ] \quad (3.5)$$

- **Για την Περίπτωση 2:**

Μόνο οι γωνίες πρέπει να περιορίζονται μεταξύ -90 και +90 μοιρών, οπότε οι περιορισμοί είναι:

$$\theta_1, \theta_2, \theta_3 \in [-90^\circ, +90^\circ] \quad (3.6)$$

Επιτάχυνση του αλγορίθμου

Λόγω της υπολογιστικά εντατικής φύσης της αντικειμενικής συνάρτησης, εφαρμόστηκε μια μέθοδος για τη μείωση του υπολογιστικού χρόνου. Αντί ο αλγόριθμος να μπορεί να επιλέξει οποιονδήποτε αυθαίρετο πραγματικό αριθμό εντός του καθορισμένου εύρους κάθε μεταβλητής βελτιστοποίησης, βρέθηκε μια συμβιβαστική λύση. Οι τιμές κάθε μεταβλητής στρογγυλοποιούνται εσωτερικά σε μια καθορισμένη ακρίβεια που δίνεται από τον χρήστη. Επιπλέον, τα αποτελέσματα κάθε επανάληψης αποθηκεύονται σε μνήμη cache έτσι ώστε σε περίπτωση που ο αλγόριθμος προσπαθήσει να χρησιμοποιήσει την ίδια είσοδο, οι υπολογισμοί να παραλείπονται και αντ' αυτών να χρησιμοποιούνται τα αποθηκευμένα αποτελέσματα. Η αποθήκευση στην cache σε συνδυασμό με τη στρογγυλοποίηση οδηγεί σε πολύ λιγότερες κλήσεις στον επιλυτή OptiStruct από ό,τι θα απαιτούνταν διαφορετικά. Για την εφαρμογή αυτή, οι γωνίες στρογγυλοποιούνται

στον πλησιέστερο ακέραιο, ενώ το πάχος στρογγυλοποιείται στο πλησιέστερο δέκατο του χιλιοστού.

Για να γίνει πιο σαφής αυτή η έννοια, ακολουθεί ένα παράδειγμα:

Ας υποθέσουμε ότι το αρχικό διάνυσμα είναι $\vec{x}_0 = [0.0005, 45, -45, 45]^T$

1. Για την πρώτη επανάληψη ας υποθέσουμε ότι ο αλγόριθμος προσπαθεί

$$\vec{x}_1 = [0.00043, -20.3, -69.6, 42.14]^T$$

αυτό θα στρογγυλοποιηθεί εσωτερικά σε

$$\vec{x}_1 = [0.0004, 20, -70, 42]^T$$

και το αποτέλεσμα θα αποθηκευτεί στην cache

2. Αν στη δεύτερη επανάληψη ο αλγόριθμος προσπαθήσει

$$\vec{x}_2 = [0.00041, -20.4, -69.8, 42.47]^T$$

θα στρογγυλοποιηθεί και αυτό σε

$$\vec{x}_2 = [0.0004, 20, -70, 42]^T$$

και το ίδιο αποθηκευμένο αποτέλεσμα θα χρησιμοποιηθεί.

Τα αποτελέσματα της μεθόδου Powell παρουσιάζονται στην ενότητα 4.3

3.4.2 Εφαρμογή του Γενετικού Αλγορίθμου

Για την εφαρμογή του γενετικού αλγορίθμου χρησιμοποιείται η βιβλιοθήκη PyGAD [5]. Για να οριστεί το πρόβλημα βελτιστοποίησης για το γενετικό αλγόριθμο, πρέπει να οριστούν πολλές παράμετροι. Δυστυχώς, δεν υπάρχει τρόπος να βρεθούν οι βέλτιστες παράμετροι για κάθε μεταβλητή σε κάθε συγκεκριμένο πρόβλημα. Επομένως, οι παράμετροι επιλέχθηκαν μετά από κάποιο πειραματισμό χρησιμοποιώντας ένα μικρότερο αριθμό γενεών που μπορούν να εκτελούνται πιο γρήγορα. Οι παράμετροι που επιλέχθηκαν πιθανότατα δεν είναι οι βέλτιστοι, αλλά λειτουργούν αρκετά καλά.

1. Αρχικά, πρέπει να αποφασιστούν τα **γονίδια** (genes) και ο **χώρος των γονιδίων** (gene space).

Τα γονίδια είναι αντίστοιχα με τις μεταβλητές βελτιστοποίησης των κλασικών αλγορίθμων βελτιστοποίησης και επιλέγονται οι τρεις γωνίες και το πάχος των στρώσεων, δηλαδή τέσσερα γονίδια συνολικά.

Ακόμη, πρέπει να οριστεί ένα εύρος δυνατών τιμών για κάθε γονίδιο. Το εύρος για κάθε γονίδιο γωνίας είναι:

$$\vartheta_{RANGE} = [-90, +90, \ step = 1]$$

και για το πάχος:

$$t_{RANGE} = [0.0001, 0.001, \ step = 0.0001]$$

αυτά τα εύρη ορίζουν το χώρο των γονιδίων.

2. Πρέπει να οριστεί το fitness function . Το fitness function είναι αντίστοιχο με την αντικειμενική συνάρτηση των κλασικών αλγορίθμων βελτιστοποίησης. Λόγω των προηγμένων δυνατοτήτων αυτού του αλγορίθμου, εκτελείται μια βελτιστοποίηση πολλαπλών στόχων , όπου οι δύο στόχοι είναι:

- Η ελαχιστοποίηση της μάζας της δομής του πτερυγίου ASW 28.
- Η μεγιστοποίηση της ταχύτητας πτερυγισμού.

Για να επιτευχθούν αυτοί οι στόχοι, δημιουργείται μια συνάρτηση με την εξής μορφή:

$$f_{fitness} \begin{pmatrix} \begin{bmatrix} t \\ \vartheta_1 \\ \vartheta_2 \\ \vartheta_3 \end{bmatrix} \end{pmatrix} = \begin{bmatrix} -Mass \\ V_{flutter} \end{bmatrix} \quad (3.7)$$

Το αρνητικό πρόσημο στη μάζα οφείλεται στο γεγονός ότι αυτός ο αλγόριθμος λειτουργεί με σκοπό τη μεγιστοποίηση της συνάρτησης καταλληλότητας.

Μετά την οριστικοποίηση αυτών των βασικών και υποχρεωτικών εισόδων, ορίζονται αρκετές άλλες παράμετροι που ελέγχουν τον τρόπο λειτουργίας του αλγορίθμου.

- $Num_generations = 1000$, ελέγχει τον αριθμό των γενεών στο διάστημα των οποίων θα πραγματοποιηθεί η εξέλιξη. Αυτή η παράμετρος επιλέγεται έτσι ώστε να διατηρείται ένας λογικός χρόνος υπολογισμού.
- $sol_per_pop = 10$, ο αριθμός των λύσεων που θα παραχθούν (ο αριθμός των χρωμοσωμάτων) μετά από κάθε γενιά.
- $parent_selection_type = steady\ state\ selection$
- $keep_elitism = 4$, που σημαίνει ότι οι τέσσερις καλύτερες λύσεις κάθε γενιάς μεταφέρονται στην επόμενη.
- $crossover_type = "single\ point"$
- $crossover_probability = 0.7$, η πιθανότητα επιλογής ενός γονέα για την εφαρμογή της γενετικής διασταύρωσης. Για κάθε γονέα, δημιουργείται μια τυχαία τιμή μεταξύ 0.0 και 1.0. Αν αυτή η τυχαία τιμή είναι μικρότερη ή ίση με την τιμή που έχει ανατεθεί στην παράμετρο crossover_probability, τότε ο γονέας επιλέγεται.
- $mutation_type = "random"$
- $mutation_probability = 0.1$, η πιθανότητα επιλογής ενός γονιδίου για την εφαρμογή της μετάλλαξης. Σε κάθε γονίδιο σε μια λύση, αναθέτεται μια τυχαία τιμή μεταξύ 0.0 και 1.0. Αν αυτή η τυχαία τιμή είναι μικρότερη ή ίση με την τιμή που έχει ανατεθεί στην παράμετρο mutation_probability, τότε το γονίδιο επιλέγεται.
- $mutation_by_replacement = True$, σημαίνει ότι το γονίδιο αντικαθίσταται από την τυχαία τιμή που δημιουργείται αντί να προστεθεί η μετάλλαξη στην υπάρχουσα τιμή του.

Τα αποτελέσματα του Γενετικού Αλγορίθμου παρουσιάζονται στήν ενότητα 4.4

3.4.3 Πρόβλεψη Ταχύτητας πτερυγισμού μέσω Νευρωνικών Δικτύων

Ο σκοπός αυτής της ενότητας είναι η ανάπτυξη ενός υποκατάστατου μοντέλου (surrogate model) χρησιμοποιώντας Νευρωνικά Δίκτυα για να επιταχυνθεί η διαδικασία της βελτιστοποίησης. Τα υποκατάστατα μοντέλα χρησιμοποιούνται συχνά στη μηχανική όταν το αποτέλεσμα που μας ενδιαφέρει δεν μπορεί να μετρηθεί εύκολα ή όταν η υπολογιστική προσπάθεια που απαιτείται για την κατάλληλη προσομοίωση είναι πολύ μεγάλη. Η

ανάπτυξη των νευρωνικών δικτύων γίνεται στην Python χρησιμοποιώντας τη βιβλιοθήκη Keras από το TensorFlow.

Η ανάπτυξη ενός νευρωνικού δικτύου ακολουθεί κάποια βήματα:

1. Τα δεδομένα εκπαίδευσης για τα μοντέλα πρέπει να συλλεχθούν και να οργανωθούν με τρόπο που να είναι εύχρηστος.
2. Η δομή και οι παράμετροι του μοντέλου πρέπει να οριστούν.
3. Το μοντέλο πρέπει να εκπαιδευτεί με τα δεδομένα εκπαίδευσης.
4. Τέλος, πρέπει να αξιολογηθεί η απόδοση του δικτύου.

Δεδομένα Εκπαίδευσης

Για να αποκτηθούν τα δεδομένα εκπαίδευσης, ο επιλυτής εκτελείται επανειλημμένα με τυχαίες παραμέτρους του laminte εντός καθορισμένων ορίων. Οι παράμετροι που μεταβάλλονται είναι οι ίδιες με αυτές που χρησιμοποιούνται για τη βελτιστοποίηση.

Πιο συγκεκριμένα, οι παράμετροι αυτοί είναι:

Η γωνία του κάθε στρώματος, με την υπόθεση ότι η κατασκευή είναι αντισυμμετρική

$$\theta_i \in [-90^\circ, +90^\circ], \quad \text{για } i = 1, 2, 3 \quad (3.8)$$

Το πάχος κάθε στρώματος, με την υπόθεση ότι όλα τα στρώματα έχουν το ίδιο πάχος

$$t \in [0.1, 0.9] \text{ mm} \quad (3.9)$$

Συλλέχθηκαν 3000 τυχαία σημεία δεδομένων και οργανώθηκαν σε ένα data frame, όπου κάθε γραμμή αντιπροσωπεύει ένα σημείο σχεδίασης και περιέχει πληροφορίες για τις μεταβλητές εισόδου που χρησιμοποιούνται και την προβλεπόμενη ταχύτητα flutter από το Optistruct, η οποία είναι και ο στόχος πρόβλεψης του Νευρωνικού Δικτύου. Επίσης καταγράφηκε και η μάζα της πτέρυγας, αλλά αυτή δεν αποτελεί στόχο για τα νευρωνικά δίκτυα που πρόκειται να αναπτυχθούν, καθώς η μάζα είναι ευθέως ανάλογη με το πάχος των στρωμάτων. Μάζα $\propto t$. Ο απαιτούμενος υπολογιστικός χρόνος για τη συλλογή όλων των δεδομένων ήταν περίπου 23 ώρες.

Δομή Μοντέλου και Παράμετροι

Η δομή των στρωμάτων είναι μια πολύ σημαντική πτυχή των νευρωνικών δικτύων. Δεν υπάρχει τρόπος να γνωρίζουμε εκ των προτέρων τη βέλτιστη δομή του νευρωνικού δικτύου ώστε να γίνει καλή πρόβλεψη και να επιτευχθεί ικανοποιητικός χρόνος εκπαίδευσης. Για αυτόν τον λόγο, θα δημιουργήθουν πολλά διαφορετικά μοντέλα και η απόδοσή τους θα αξιολογηθεί για να βρεθεί το καλύτερο.

Το πρώτο στρώμα κάθε Νευρωνικού Δικτύου είναι το στρώμα κανονικοποίησης. Αυτό είναι ένα σημαντικό βήμα που διασφαλίζει μια συνεπή κλίμακα μεταξύ των διαφορετικών τύπων δεδομένων εισόδου. Αυτό διασφαλίζει ότι κανένα χαρακτηριστικό δεν κυριαρχεί στη διαδικασία μάθησης. Επιπλέον, η κανονικοποίηση μπορεί να βελτιώσει τη σύγκλιση διατηρώντας τα βάρη και τις ενεργοποιήσεις εντός μιας λογικής περιοχής. Τέλος, η κανονικοποίηση μπορεί να βοηθήσει στη μείωση του overfitting και να καταστήσει το μοντέλο πιο ανθεκτικό.

Για αυτό το παράδειγμα, οι δύο διαφορετικοί τύποι εισόδων (οι γωνίες και το πάχος) έχουν πολύ διαφορετικά όρια, καθώς οι γωνίες κυμαίνονται μεταξύ -90 και $+90$, ενώ τα πάχη είναι τέσσερις τάξεις μεγέθους μικρότερα, αφού κυμαίνονται μεταξύ 0.0001 και 0.001 μέτρα.

Το στρώμα κανονικοποίησης στο Keras μετατοπίζει και κλιμακώνει τα δεδομένα εισόδου σε μια κατανομή που έχει κέντρο το 0 και τυπική απόκλιση 1 , αφαιρώντας τον μέσο όρο των δεδομένων εκπαίδευσης και διαιρώντας με τη ρίζα της διακύμανσης των δεδομένων.

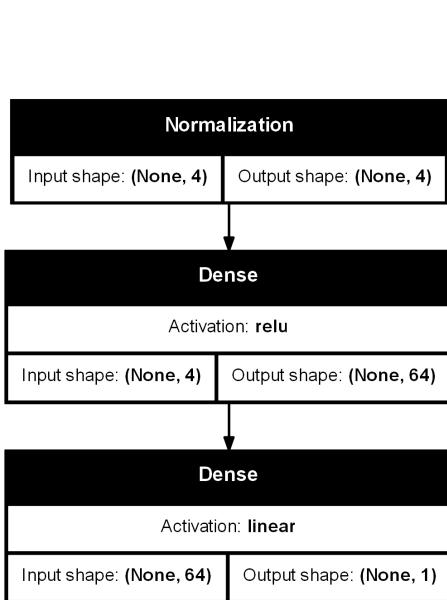
$$f_{norm}(x) = \frac{x - mean(\vec{x})}{\sqrt{var(\vec{x})}} \quad (3.10)$$

Ο μέσος όρος και η διακύμανση των δεδομένων υπολογίζονται με βάση τα δεδομένα εκπαίδευσης. Εάν τα δεδομένα εκπαίδευσης δεν είναι αντιπροσωπευτικά των πραγματικών δεδομένων, στα οποία το δίκτυο θα κληθεί να κάνει προβλέψεις, η κανονικοποίηση μπορεί να αποτύχει, παράγοντας μια μη κεντραρισμένη κατανομή με διακύμανση που διαφέρει σημαντικά από το 1 .

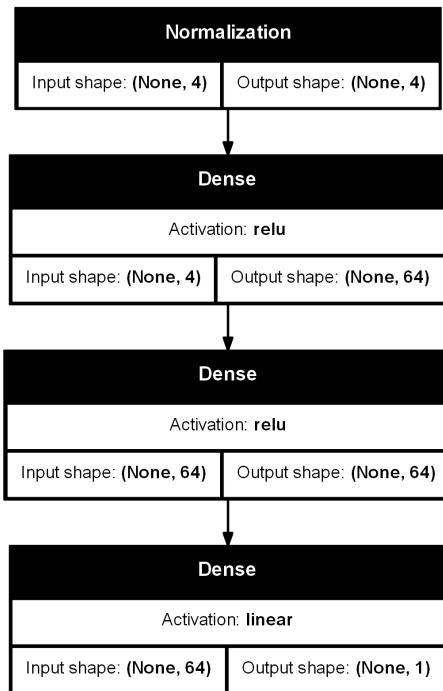
Μετά το στρώμα κανονικοποίησης ακολουθούν τα κρυφά στρώματα. Αυτά τα στρώματα διαφέρουν σε αριθμό από μοντέλο σε μοντέλο. Στη συγκεκριμένη εργασία, θα δοκιμαστούν Νευρωνικά Δίκτυα με $1, 2, 4$ και 6 κρυφά στρώματα όπως φαίνεται στα Σχήματα 3.14 έως 3.17. Κάθε στρώμα περιέχει 64 Νευρώνες και χρησιμοποιεί τη συνάρτηση ενεργοποίησης RELU.

Τέλος, προστίθεται το στρώμα εξόδου το οποίο χρησιμοποιεί μια γραμμική συνάρτηση

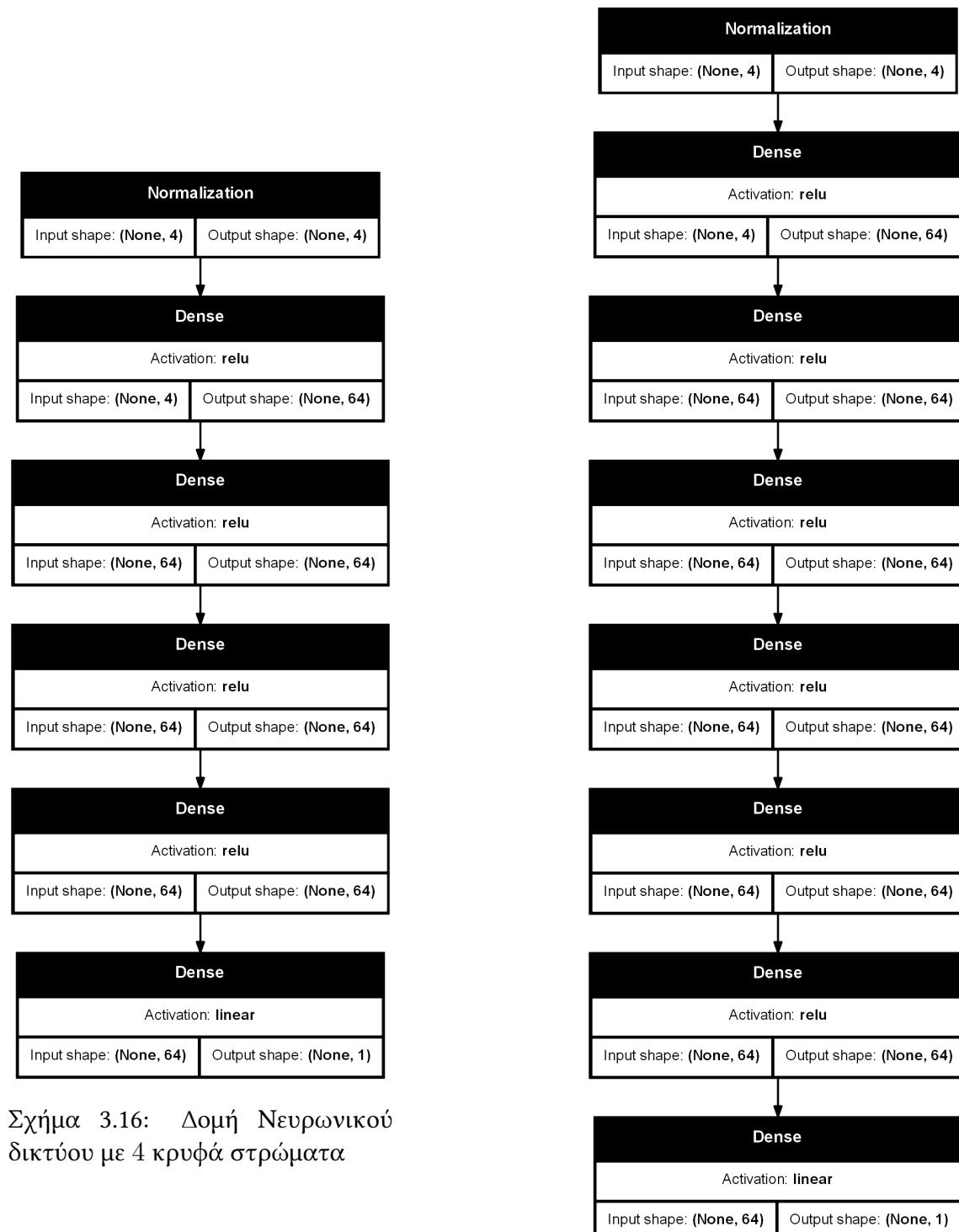
ενεργοποίησης, καθώς πρόκειται για πρόβλημα παλινδρόμησης και έχει μόνο 1 Νευρώνα. Η μοναδική πρόβλεψη που κάνουν αυτά τα νευρωνικά δίκτυα είναι η ταχύτητα πτερυγισμού της πτέρυγας ASW 28.



Σχήμα 3.14: Δομή Νευρωνικού δικτύου με 1 κρυφό στρώμα



Σχήμα 3.15: Δομή Νευρωνικού δικτύου με 2 κρυφά στρώματα



Σχήμα 3.16: Δομή Νευρωνικού δικτύου με 4 κρυφά στρώματα

Σχήμα 3.17: Δομή Νευρωνικού δικτύου με 6 κρυφά στρώματα

Συνάρτηση Απώλειας (Loss Function)

Η συνάρτηση απώλειας που επιλέχθηκε για αυτήν την εφαρμογή είναι το Mean Average Error, που περιγράφεται στην Εξίσωση (2.87), καθώς είναι εύκολη στην ερμηνεία της, έχει τις ίδιες μονάδες με τη μεταβλητή εξόδου και είναι ανθεκτική σε outlier δεδομένα.

Optimizer

Ο βελτιστοποιητής που επιλέχθηκε είναι ο Adam, ο οποίος είναι ευρέως χρησιμοποιούμενος και θεωρείται ένας από τους καλύτερους βελτιστοποιητές, με ρυθμό εκμάθησης 0.01.

hyperparameter Tuning

Τέλος, κατασκευάζεται ένα μοντέλο του οποίου η δομή καθορίζεται από τον αλγόριθμο HyperBand. Το μοντέλο είναι ελεύθερο να επιλέξει τις ακόλουθες υπερ-παραμέτρους:

- Τον αριθμό των Κρυφών Στρωμάτων μεταξύ 1 και 10.
- Τον αριθμό των Νευρώνων για κάθε στρώμα από ένα διακριτό σύνολο τιμών

$$\{2^i\}, \text{ με } i = 1, 2, \dots, 10$$

- Τη συνάρτηση ενεργοποίησης κάθε στρώματος: ReLu, tanh, ή Leaky ReLu.
- Τον ρυθμό εκμάθησης του βελτιστοποιητή.

Η δομή αυτού του μοντέλου δεν είναι γνωστή σε αυτή τη φάση, καθώς είναι προϊόν της διαδικασίας βελτιστοποίησης. Γι' αυτό το λόγο θα παρουσιαστεί στην ενότητα αποτελεσμάτων.

Τα αποτελέσματα των Νευρωνικών Δικτύων παρουσιάζονται στην ενότητα 4.5

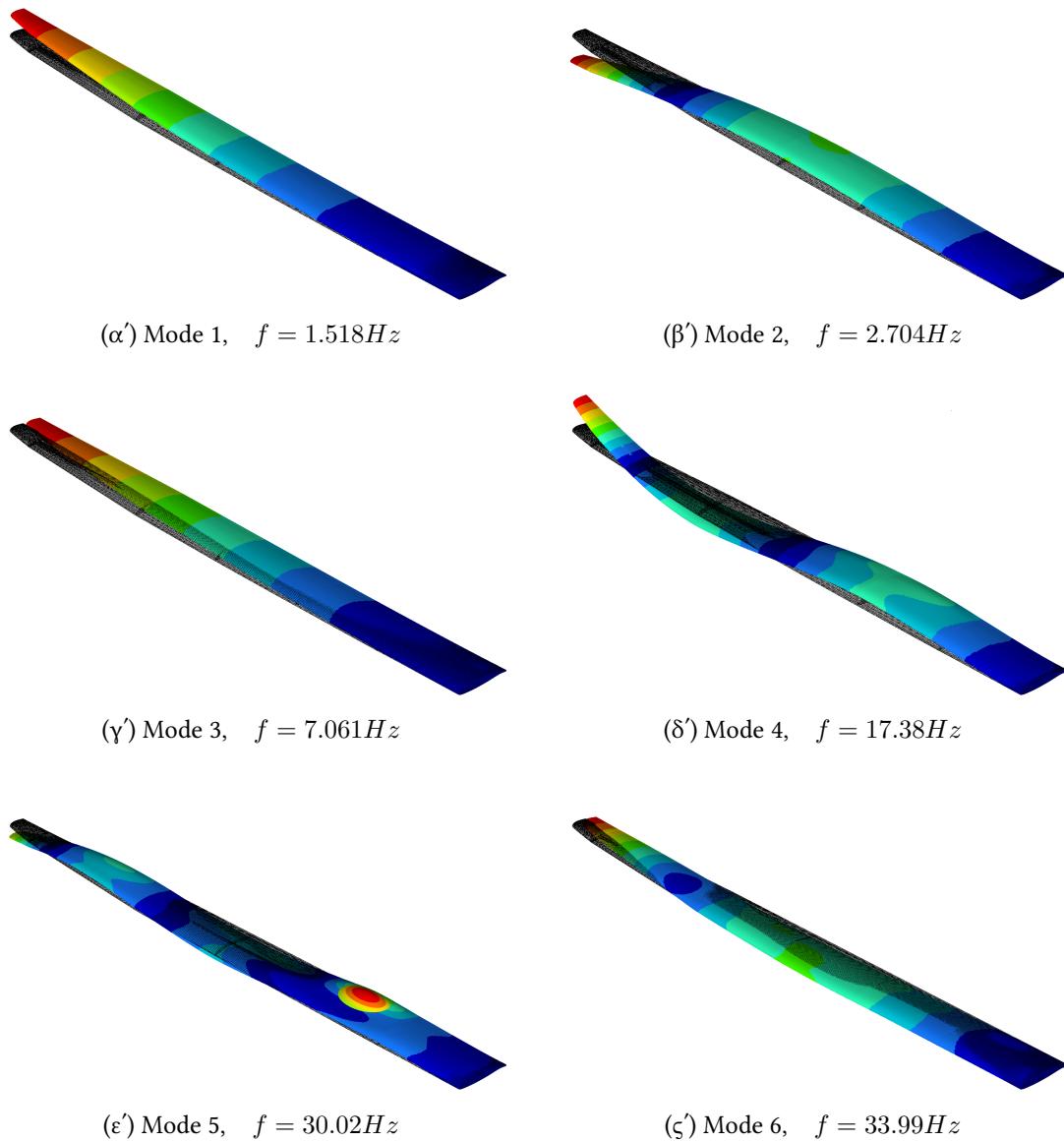
Κεφάλαιο 4

Αποτελέσματα

Το κεφάλαιο των Αποτελεσμάτων μελετά τα δυναμικά χαρακτηριστικά πτερυγισμού (*flutter*) του φτερού ASW28. Επιπλέον, παρουσιάζονται τα αποτελέσματα των τεχνικών βελτιστοποίησης που εφαρμόστηκαν. Τέλος, αποδεικνύεται η δυνατότητα που έχουν τα Νευρωνικά Δίκτυα να προβλέπουν την ταχύτητα πτερυγισμού του φτερού.

4.1 Ανάλυση Ιδιομορφών

Για μια καλύτερη κατανόηση των τρόπων με τους οποίους ταλαντώνεται ή πτέρυγα, πραγματοποιείται μια απλή ανάλυση ιδιομορφών. Οι ιδιομορφές και τα ιδιοδιανύσματα που προκύπτουν από αυτήν την ανάλυση είναι εκείνα που θα προέκυπταν σε μια ανάλυση πτερυγιμού για μηδενική ταχύτητα αέρα. Οι πρώτες 6 ιδιομορφές του φτερού παρουσιάζονται στο Σχήμα 4.1:



Σχήμα 4.1: Ο πρώτες 6 ιδιομορφές του φτερού ASW 28

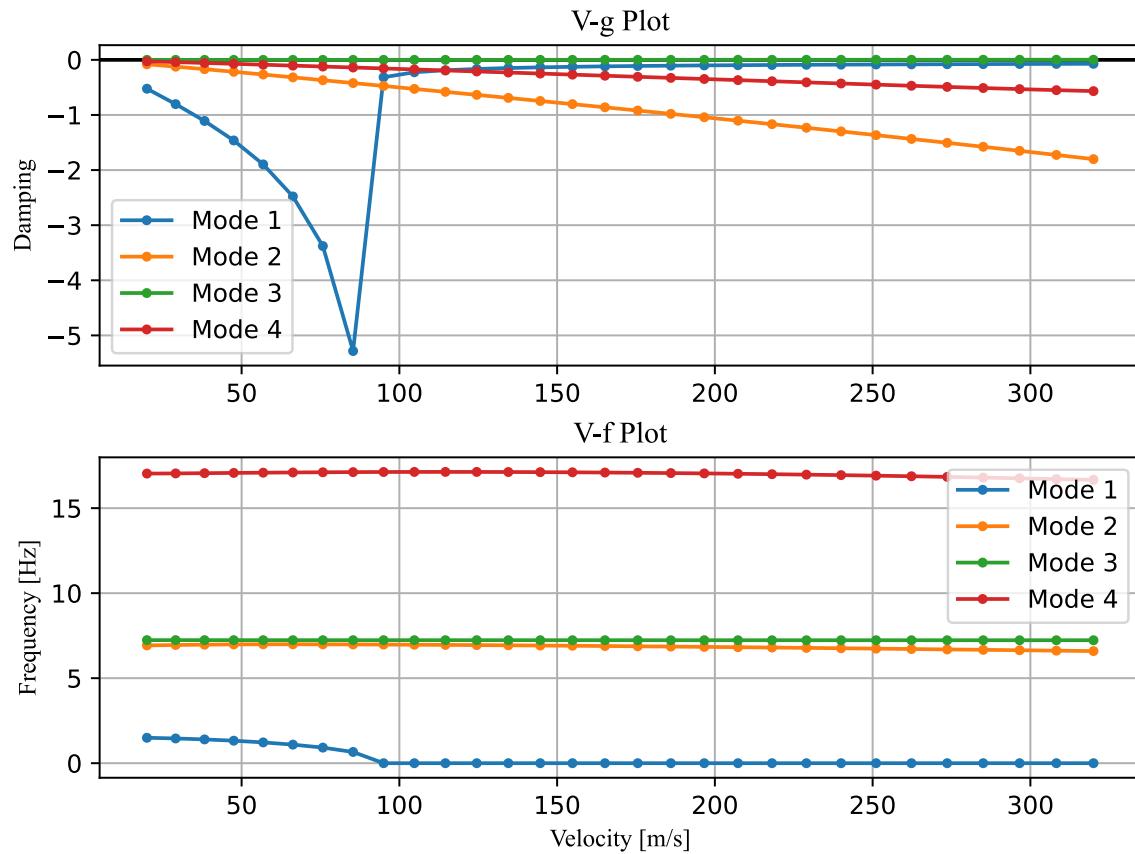
Από το Σχήμα 4.1 μπορούν να παρατηρηθούν τα εξής:

- Η πρώτη ιδιομορφή έχει ιδιαίτερα χαμηλή ιδιοσυχνότητα, υποδεικνύοντας ότι το φτερό είναι αρκετά εύκαμπτο σε αυτήν την κατεύθυνση.

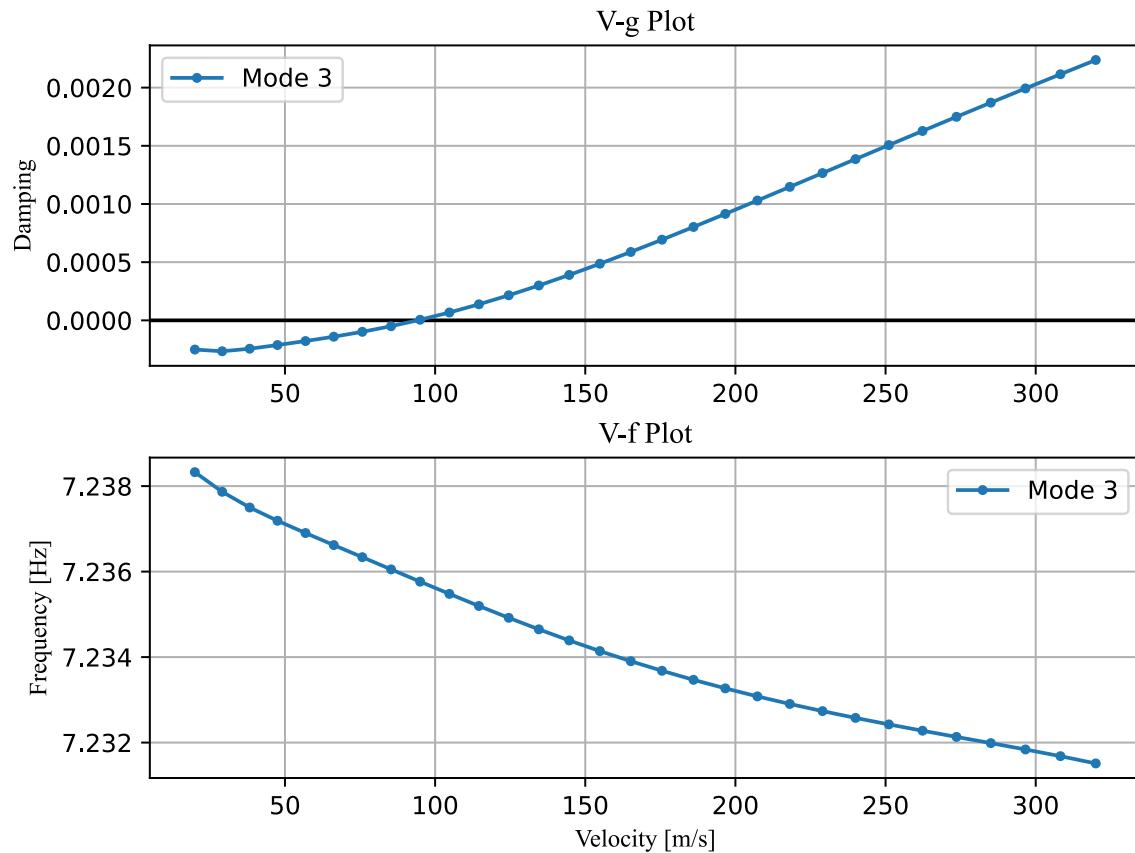
- Οι ιδιομορφές 1, 2, 4 και 5 είναι όλες καμπτικές γύρω από τον άξονα x , με αυξανόμενο αριθμό κόμβων (σταθερών σημείων) στο φτερό.
 - Η ιδιομορφή 1 είναι η πιο απλή με ένα μόνο κόμβο στη ρίζα του φτερού. Καθώς η ιδιοσυχνότητα αυξάνεται, αυξάνεται και η πολυπλοκότητα της καμπτικής κίνησης καθώς και ο αριθμός των κόμβων.
 - Η ιδιομορφή 2 έχει δύο κόμβους, ένα στη ρίζα του φτερού και έναν περίπου στα τρία τέταρτα του εκπετάσματος της πτέρυγας.
 - Η ιδιομορφή 4 έχει τρεις κόμβους, ένα στη ρίζα του φτερού, ένα στη μέση και έναν περίπου στο 5/6 του εκπετάσματος της πτέρυγας.
 - Η ιδιομορφή 5 έχει τέσσερις κόμβους κατανεμημένους κατά μήκος του εκπετάσματος της πτέρυγας, αλλά παρουσιάζει επίσης κάποιες ενδείξεις ταλάντωσης πλάκας στην άνω επιφάνεια της πτέρυγας, κοντά στη ρίζα, όπου τα ribs είναι πιο απομακρυσμένα και υπάρχει ένα μεγάλο τμήμα επιφάνειας του φτερού χωρίς στήριξη.
- Οι ιδιομορφές 3 και 6 είναι καμπτικές γύρω από τον άξονα z . Η ιδιομορφή 3 έχει μόνο έναν κόμβο, ενώ η ιδιομορφή 6 έχει δύο.

4.2 Αρχική Ανάλυση Πτερυγισμού

Εφαρμόζοντας στο μοντέλο τη μεθοδολογία που αναπτύχθηκε στην ενότητα 3.2, προκύπτουν τα εξής αποτελέσματα:



Σχήμα 4.2: Αρχικό διάγραμμα πτερυγισμού για τις 4 πρώτες ιδιομορφές.



Σχήμα 4.3: Διάγραμμα πτερυγισμού για την 3^η ιδιομορφή

Από τα Σχήματα 4.2 και 4.3 μπορούμε να δούμε ότι το Flutter συμβαίνει στην 3^η ιδιομορφή με:

$$V_{flutter} = 94.11 \text{ m/s} \quad (4.1)$$

Παρατηρείται ότι η 1^η ιδιομορφή είναι πολύ πιθανό να αποκλίνει επίσης περίπου στην ίδια ταχύτητα, αν και η απόσβεσή της δεν αλλάζει πρόσημο. Η απότομη αύξηση της απόσβεσης σε μια τιμή κοντά στο μηδέν (αλλά ακόμη αρνητική), μαζί με τη μείωση της ιδιοσυχνότητας σχεδόν στο μηδέν, υποδεικνύουν την ύπαρξη στατικής αεροελαστικής απόκλισης (aeroelastic divergence) που δεν καταγράφεται από τον αλγόριθμο. Η συμπεριφορά αυτή είναι αναμενόμενη για πολύ ευκαμπτες πτέρυγες με μεγάλα εκπετάσματα όταν μελετούνται με γραμμική ανάλυση πτερυγισμού, όμως όπως έχουν δείξει μελέτες μέσω μη γραμμικών προσομοιώσεων και πειραμάτων, αυτό δεν αντικατοπτρίζει κατ'

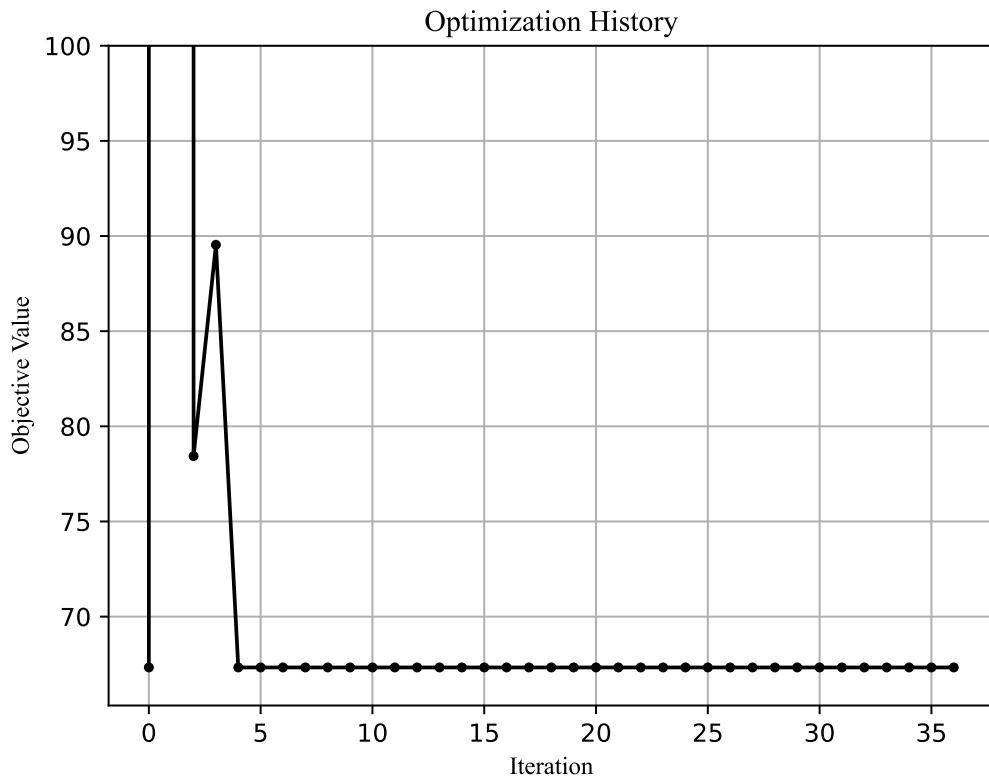
ανάγκη την πραγματικότητα, διότι τα μη γραμμικά φαινόμενα υπερισχύουν σε μεγάλες μετατοπίσεις [16].

4.3 Μέθοδος Βελτιστοποίησης Powell

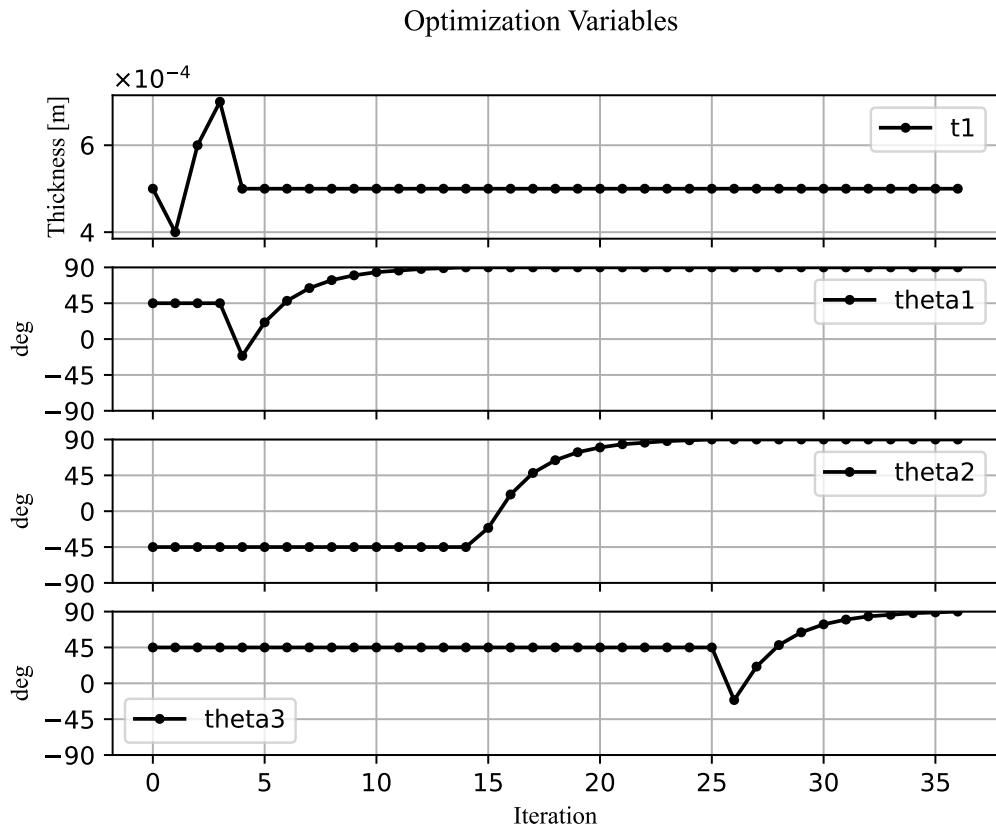
Μετά την εφαρμογή της μεθόδου βελτιστοποίησης Powell όπως περιγράφεται στήν υποενότητα 3.4.1, προκύπτουν τα εξής αποτελέσματα:

Περίπτωση 1:

Αυτή ήταν η πρώτη αντικειμενική συνάρτηση που διατυπώθηκε στην εξίσωση (3.3), καθώς ορίζει ένα πιο ολοκληρωμένο πρόβλημα βελτιστοποίησης. Η βελτιστοποίηση ολοκληρώθηκε με μερική επιτυχία. Τα αποτελέσματα παρουσιάζονται στο Σχήμα 4.4:



Σχήμα 4.4: Τιμή της αντικειμενικής συνάρτησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 1)



Σχήμα 4.5: Τιμή κάθε μεταβλητής βελτιστοποίησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 1)

Όπως φαίνεται ο αλγόριθμος αποτυγχάνει να βελτιώσει την αντικειμενική συνάρτηση πέρα από το αρχικό σημείο. Στην πρώτη επανάληψη, ο αλγόριθμος προσπαθεί να μειώσει τη μάζα της πτέρυγας μειώνοντας το πάχος των στρωμάτων κατά ένα δέκατο του χιλιοστού, αλλά η ταχύτητα πτερυγισμού μειώνεται υπερβολικά και επιβάλλεται η ποινή, με αποτέλεσμα μια πολύ υψηλή τιμή της αντικειμενικής συνάρτησης.

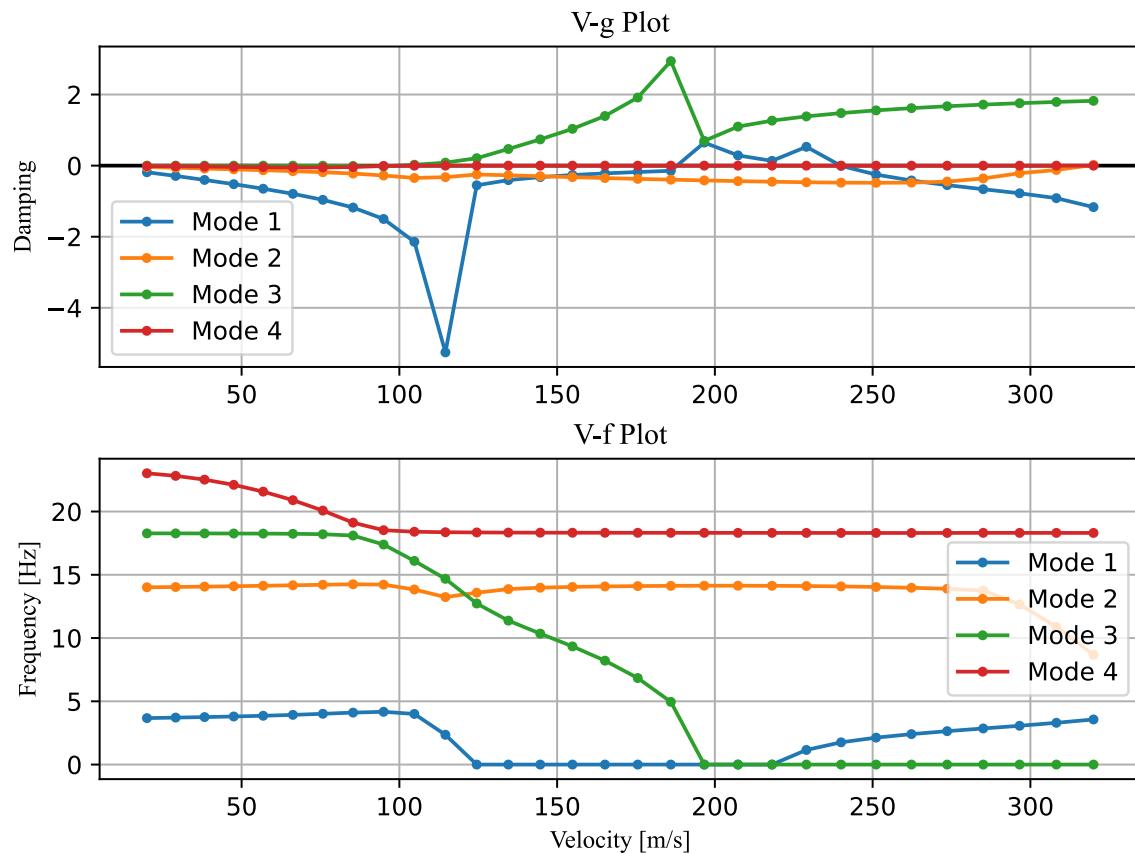
Από εκείνο το σημείο, ο αλγόριθμος πειραματίζεται με μεγαλύτερες τιμές πάχους, οι οποίες φυσικά οδηγούν σε μεγαλύτερη μάζα, αλλά με αρκετά υψηλή ταχύτητα πτερυγισμού ώστε να μην επιβληθεί η ποινή.

Τελικά, ο αλγόριθμος μειώνει το πάχος των στρωμάτων σταδιακά μέχρι να φτάσει στο αρχικό πάχος.

Ο αλγόριθμος δεν αντιλαμβάνεται ότι αλλάζοντας πρώτα τις γωνίες των στρωμάτων

ώστε η ταχύτητα πτερυγισμού να αυξηθεί και στη συνέχεια μειώνοντας το πάχος, η ταχύτητα πτερυγισμού παραμένει αρκετά υψηλή ώστε να μην επιβληθεί η ποινή ενώ ταυτόχρονα μειώνεται η μάζα. Αυτή είναι μια περίπλοκη λύση, καθώς οι γωνίες δεν επηρεάζουν άμεσα τη μάζα, αλλά μόνο την ταχύτητα πτερυγισμού. Αν ο αλγόριθμος είχε εξερευνήσει καλύτερα τις γωνίες ενώ βρισκόταν στη ζώνη ποινής, θα μπορούσε να είχε αυξήσει την ταχύτητα του πτερυγισμού επαρκώς για να βγει από τη ζώνη ποινής και τελικά να βρει μια καλύτερη λύση. Δυστυχώς, αυτή η συμπεριφορά είναι αρκετά σύνθετη και ξεπερνά τις δυνατότητες αυτού του σχετικά απλού αλγορίθμου.

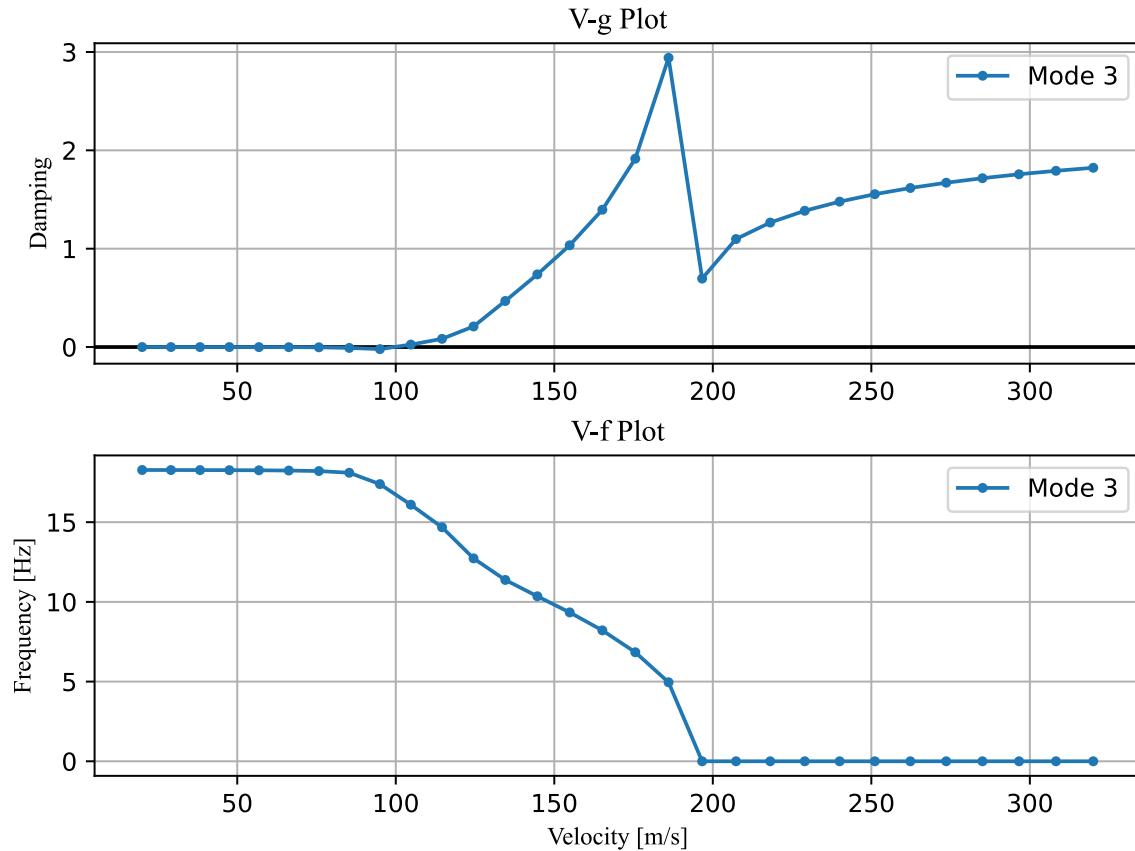
Τα αποτελέσματα της ανάλυσης πτερυγισμού της τελικής λύσης αυτού του αλγορίθμου παρουσιάζονται στα σχήματα 4.6 και 4.7:



Σχήμα 4.6: Διαγράμματα ανάλυσης πτερυγισμού από τη μέθοδο του Powell.
(Περίπτωση 1)

Η πρώτη ιδιομορφή που παρουσιάζει απόκλιση είναι η ιδιομορφή 3, η οποία απεικο-

νίζεται μόνη της στο επόμενο σχήμα ώστε να φανεί καθαρά πότε αλλάζει πρόσημο η απόσβεση.



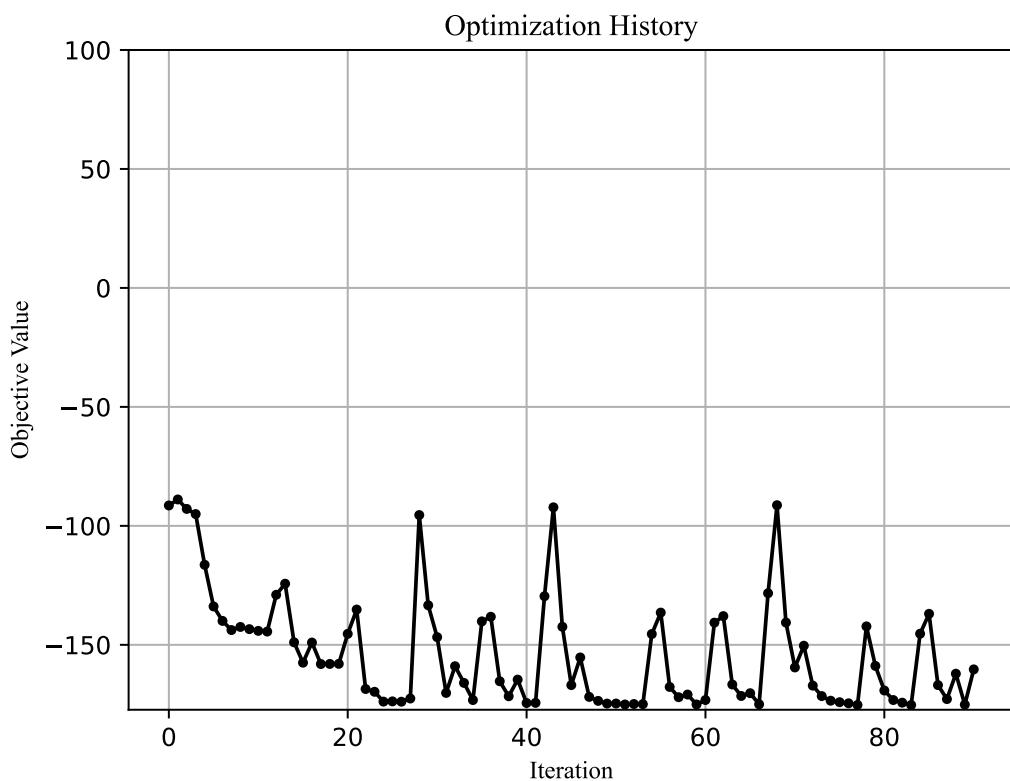
Σχήμα 4.7: Ανάλυση πτερυγισμού της 3^{ης} ιδιομορφής με τη μέθοδο του Powell.
(Περίπτωση 1)

Η ταχύτητα με την οποία η ιδιομορφή αυτή αποκλίνει για πρώτη φορά είναι η ταχύτητα πτερυγισμού και στην προκειμένη περίπτωση:

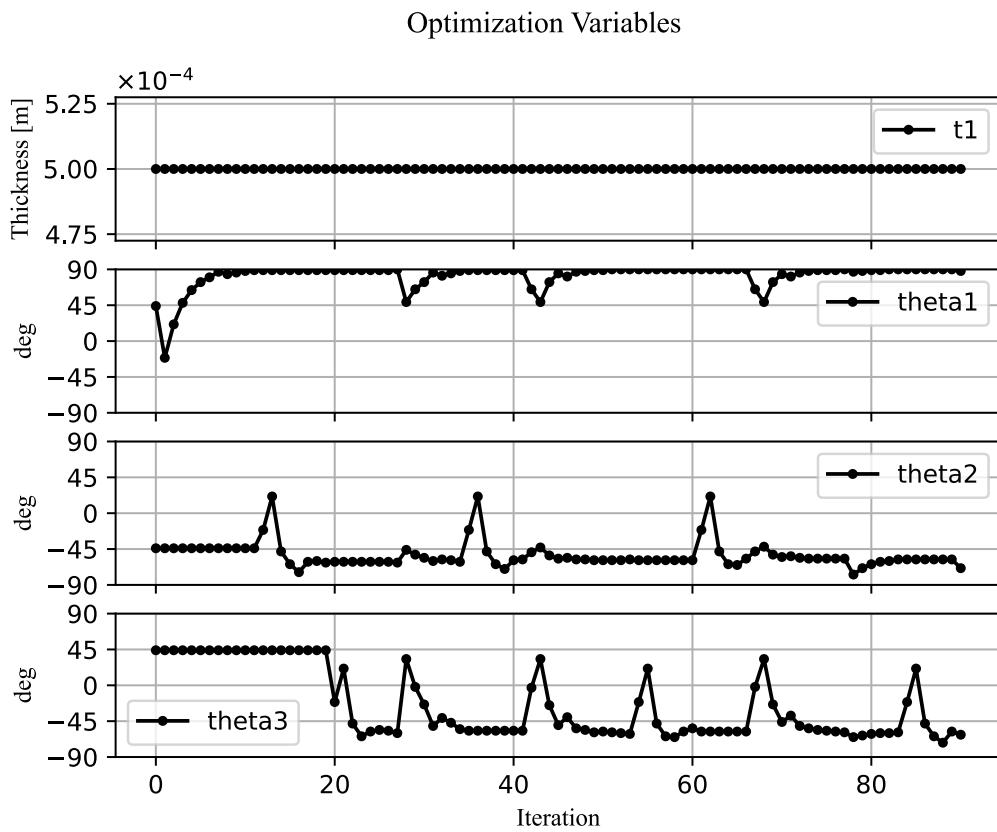
$$V_{flutter} = 99.48 \text{ m/s} \quad (4.2)$$

Περίπτωση 2:

Για να επιτευχθεί ένα καλύτερο αποτέλεσμα, χρησιμοποιείται η απλοποιημένη αντικειμενική συνάρτηση της εξίσωσης (3.4). Αυτή τη φορά, ο αλγόριθμος εξερευνά το χώρο των πιθανών λύσεων με μεγαλύτερη επιτυχία. Τα αποτελέσματα παρουσιάζονται στο Σχήμα 4.8:



Σχήμα 4.8: Τιμή της αντικειμενικής συνάρτησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 2)

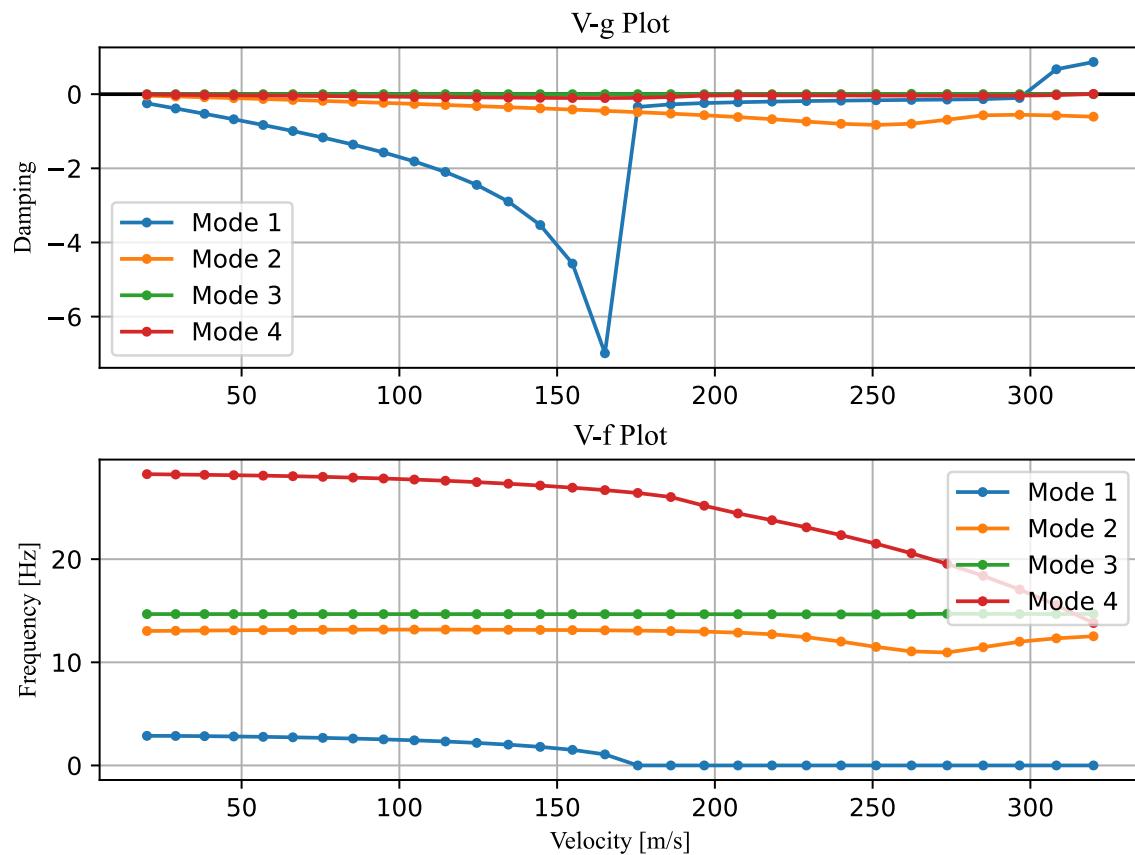


Σχήμα 4.9: Τιμή κάθε μεταβλητής βελτιστοποίησης κατά τη διάρκεια της βελτιστοποίησης (Περίπτωση 2)

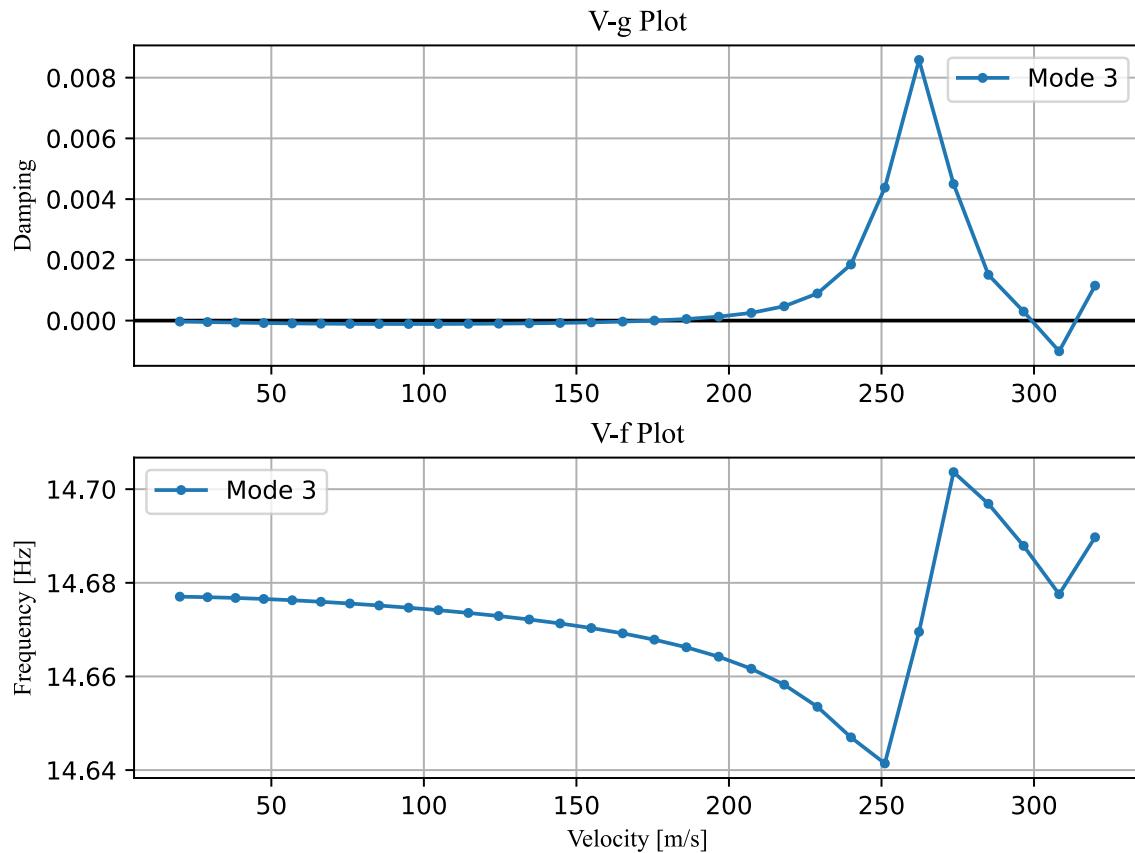
Παρατηρείται από το Σχήμα 4.8 ότι η αντικειμενική συνάρτηση, η οποία τώρα αποσκοπεί στην ελαχιστοποίηση της αρνητικής ταχύτητας πτερυγισμού, έχει βελτιωθεί σημαντικά.

Επιπλέον, σύμφωνα με τον ορισμό αυτού του προβλήματος, η μεταβλητή πάχους παραμένει σταθερή και ίση με την αρχική τιμή των 0.5mm. Αυτό σημαίνει ότι ενώ η ταχύτητα πτερυγισμού αυξάνεται σημαντικά, η μάζα παραμένει σταθερή.

Τα διαγράμματα πτερυγισμού για την βέλτιστη λύση αυτής της μεθόδου παρουσιάζονται στα σχήματα 4.10 και 4.11 .



Σχήμα 4.10: Διαγράμματα ανάλυσης πτερυγισμού από τη μέθοδο του Powell.
(Περίπτωση 2)



Σχήμα 4.11: Ανάλυση πτερυγισμού της 3^{ης} ιδιομορφής με τη μέθοδο του Powell.
(Περίπτωση 2)

Για άλλη μια φορά, η πρώτη ιδιομορφή που αποκλίνει είναι η ιδιομορφή 3. Αυτή τη φορά όμως, η ταχύτητα είναι πολύ μεγαλύτερη από πριν.

$$V_{flutter} = 175.28 \text{ m/s} \quad (4.3)$$

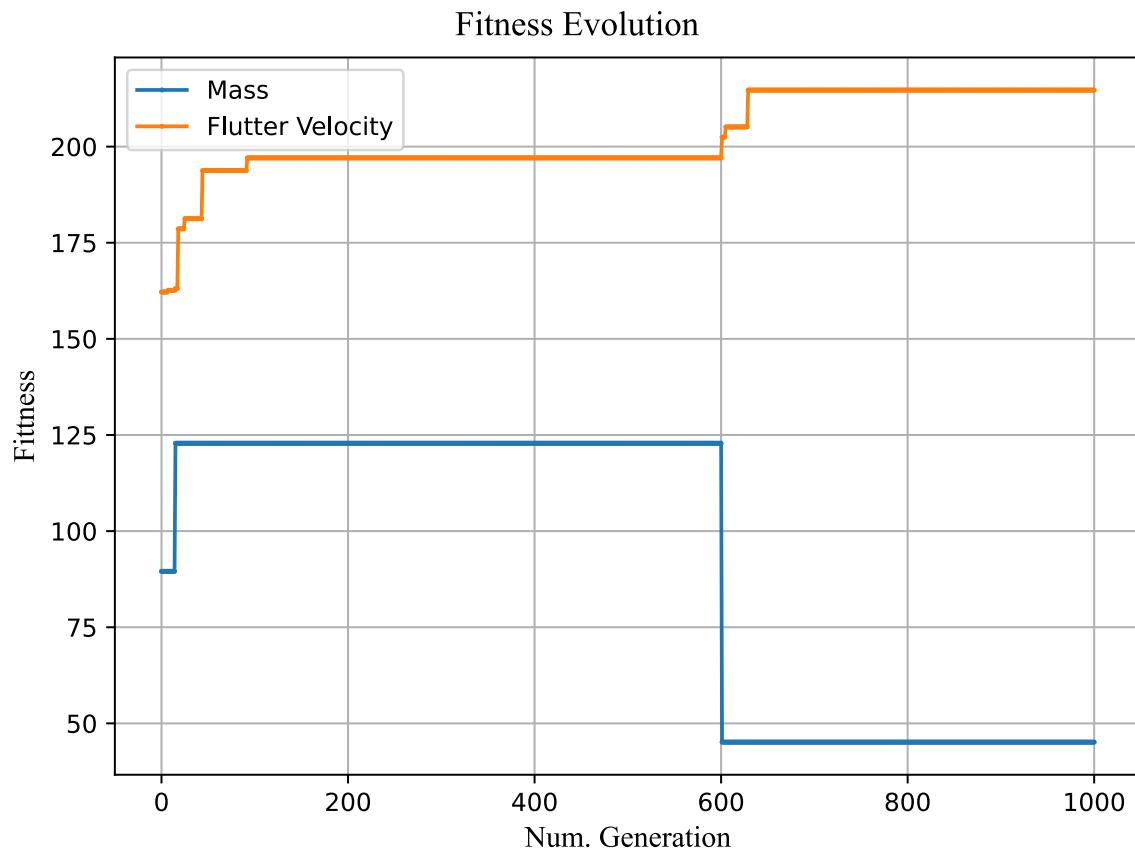
Οι ιδιομορφές 1 και 4 αποκλίνουν επίσης στα 267.09 και 177.14 m/s αντίστοιχα. Όπως είναι προφανές, αυτή είναι μια εξαιρετική βελτίωση της ταχύτητας πτερυγισμού του φτερού σε σχέση με την αρχική λύση, χωρίς επιπλέον υλικό και άρα με την ίδια μάζα με το αρχικό φτερό.

Το διάνυσμα που δίνει τη βέλτιστη λύση είναι:

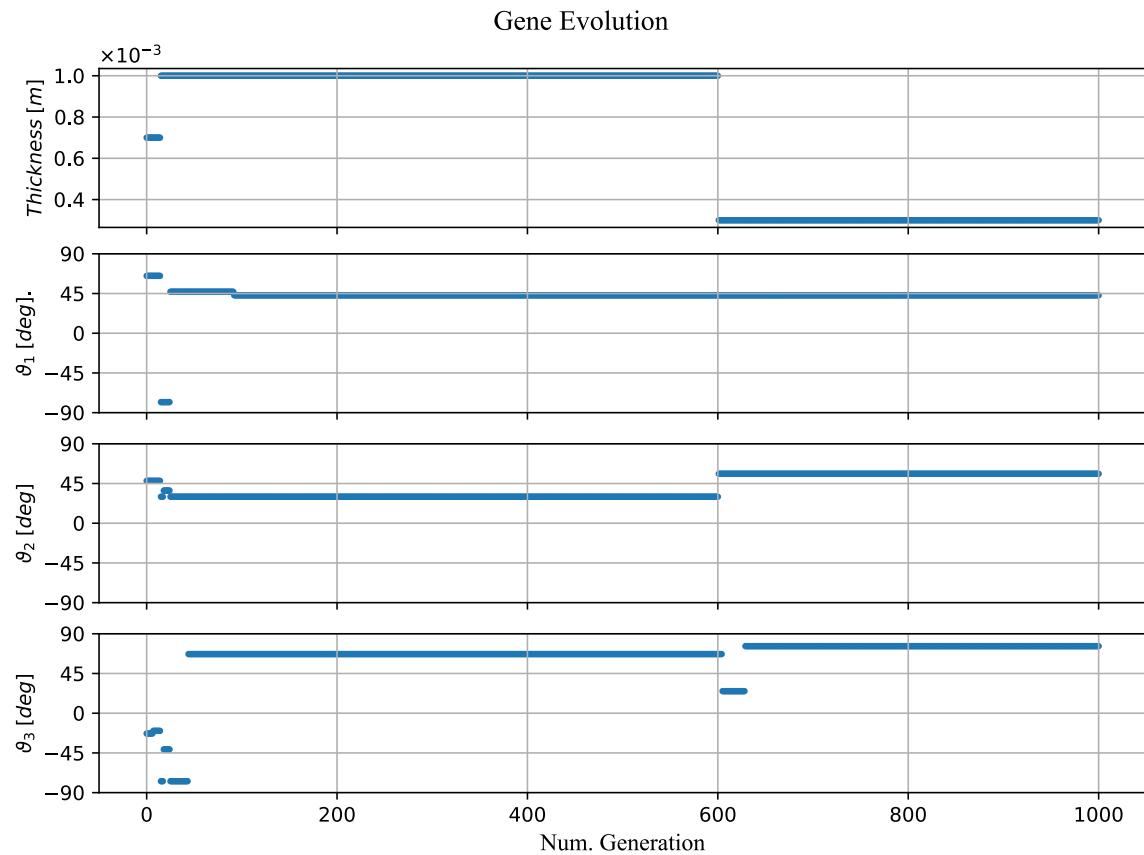
$$\vec{x}_{opt} = [0.0005, 90, -58, -59]^T \quad (4.4)$$

4.4 Βελτιστοποίηση με τη μέθοδο Γενετικών Αλγορίθμων

Μετά την εφαρμογή της μεθόδου γενετικών αλγορίθμων όπως περιγράφεται στήν υπο-ενότητα 3.4.2, προκύπτουν τα εξής αποτελέσματα:



Σχήμα 4.12: Εξέλιξη του fitness function της καλύτερης λύσης κάθε γενιάς κατά τη διαδικασία βελτιστοποίησης



Σχήμα 4.13: Εξέλιξη των τιμών των γονιδίων κατά τη διάρκεια της διαδικασίας βελτιστοποίησης.

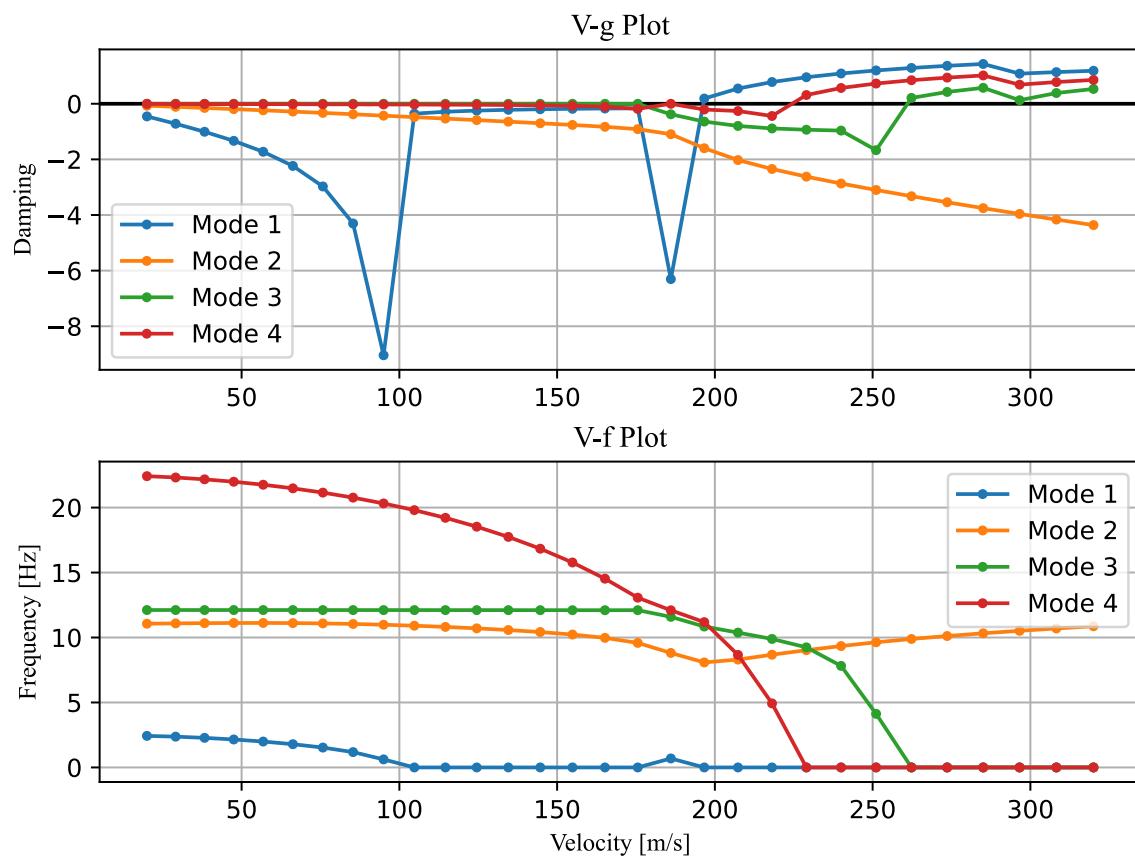
Από το Σχήμα 4.12, όπου απεικονίζεται η καλύτερη λύση που βρέθηκε μέχρι στιγμής σε κάθε γενιά, παρατηρείται ότι και οι δύο στόχοι βελτιώνονται μέχρι το τέλος της βελτιστοποίησης, παρόλο που αυτοί είναι αντιφατικοί μεταξύ τους. Από τα αρχεία καταγραφής της βελτιστοποίησης, η καλύτερη λύση έχει τα εξής χαρακτηριστικά:

$$V_{flutter} = 214.69 \text{ m/s}, \quad Mass = 45.1 \text{ kg} \quad (4.5)$$

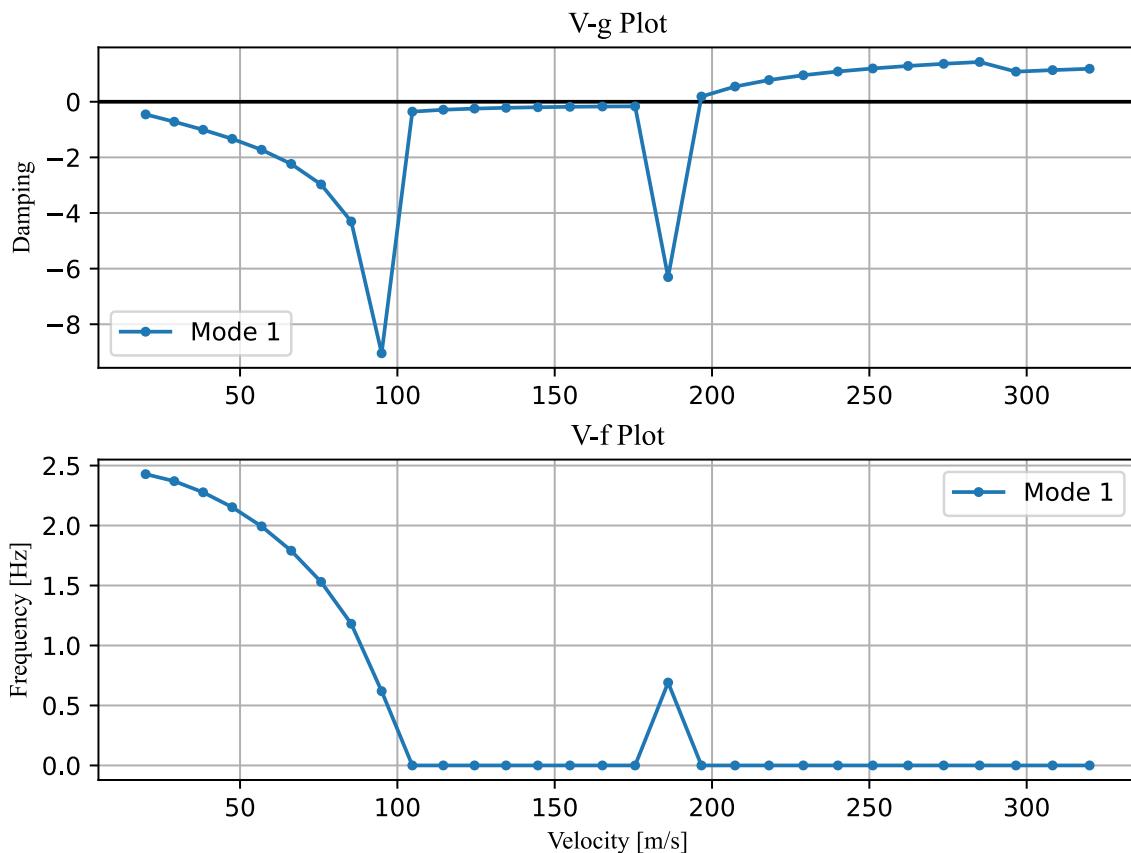
Οι τιμές των γονιδίων που οδηγούν στη βέλτιστη λύση είναι:

$$t = 0.0003 \text{ mm}, \quad \vartheta_1 = 43^\circ, \quad \vartheta_2 = 56^\circ, \quad \vartheta_3 = 76^\circ \quad (4.6)$$

Το διάγραμμα πτερυγισμού για αυτήν τη λύση παρουσιάζεται στο Σχήμα 4.14:



Σχήμα 4.14: Διάγραμμα πτερυγισμού από τον αλγόριθμο γενετικής βελτιστοποίησης.



Σχήμα 4.15: Διάγραμμα πτερυγισμού της 1^{ης} ιδιομορφής από αλγόριθμο γενετικής βελτιστοποίησης

Αν και η λύση που προκύπτει από τη μέθοδο των γενετικών αλγορίθμων φαίνεται ιδανική, εξετάζοντας το Σχήμα 4.15 μπορούμε να δούμε ότι πριν από την αναφερόμενη ταχύτητα πτερυγισμού των 214m/s , η απόσβεση της 1^{ης} ιδιομορφής αλλάζει σημαντικά στα 110m/s και η ιδιοσυχνότητα μειώνεται στο μηδέν, το οποίο είναι ένδειξη στατικής αεροελαστικής απόκλισης (static aeroelastic divergence). Είναι πιθανό αυτό να μη συμβαίνει στην πραγματικότητα, διότι όπως έχουμε δει σε άλλες μελέτες [16], όταν το φτερό εκτρέπεται με μεγάλες μετατοπίσεις, τα μη γραμμικά φαινόμενα γίνονται σημαντικά και απαιτούνται μη γραμμικές αναλύσεις στατικής αερολαστικότητας για τον εντοπισμό τέτοιων φαινομένων.

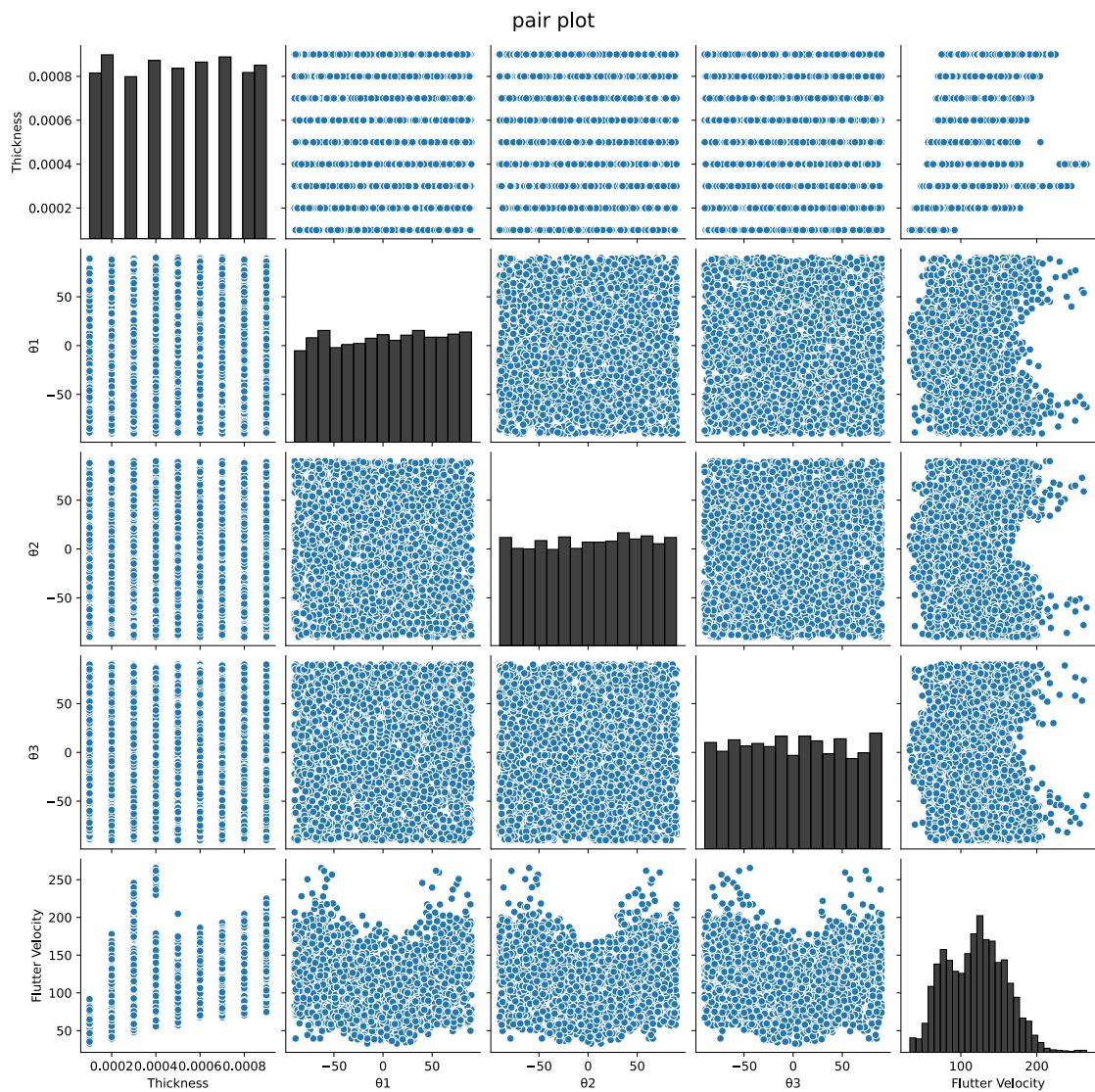
4.5 Αποτελέσματα Πρόβλεψης μέσω Νευρωνικών Δικτύων

Σε αυτή την ενότητα, θα παρουσιαστούν τα αποτελέσματα κάθε Νευρωνικού Δικτύου που αναπτύχθηκε στήν υποενότητα 3.4.3. Παρουσιάζονται τρία βασικά διαγράμματα για την κατανόηση της απόδοσης κάθε δικτύου.

1. Το πρώτο διάγραμμα δείχνει την τιμή της συνάρτησης απώλειας στα δεδομένα εκπαίδευσης και επικύρωσης σε κάθε εποχή της εκπαίδευσης.
2. Το δεύτερο διάγραμμα χρησιμοποιείται για την αξιολόγηση της απόδοσης του δικτύου με τη χρήση του διαγράμματος διασποράς y_{pred} vs y_{true} . Ιδανικά, αυτές οι τιμές θα ταυτίζονται, παράγοντας μια ευθεία γραμμή με γωνία 45° .
3. Το τρίτο διάγραμμα εμφανίζει τις διαφορές ως προς το y_{true} , $(y_{true} - y_{pred})$ vs y_{true} , διευκολύνοντας την κατανόηση του μεγέθους του σφάλματος καθώς και τυχόν τάσεων που μπορεί να υπάρχουν σε συγκεκριμένες ταχύτητες πτερυγισμού.

4.5.1 Εξέταση δεδομένων εκπαίδευσης

Ένας προκαταρκτικός έλεγχος των δεδομένων πραγματοποιείται χρησιμοποιώντας ένα διάγραμμα συσχέτισης (pair plot), το οποίο είναι ένα πλέγμα γραφημάτων που επιτρέπει την οπτικοποίηση της σχέσης μεταξύ κάθε ζεύγους μεταβλητών στο σύνολο δεδομένων.



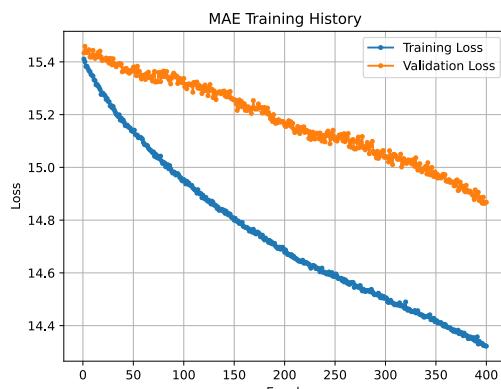
Σχήμα 4.16: pair plot των δεδομένων εκπαίδευσης

Από το Σχήμα 4.16 μπορούμε να παρατηρήσουμε τις εξής σχέσεις στα συγκεντρωμένα δεδομένα:

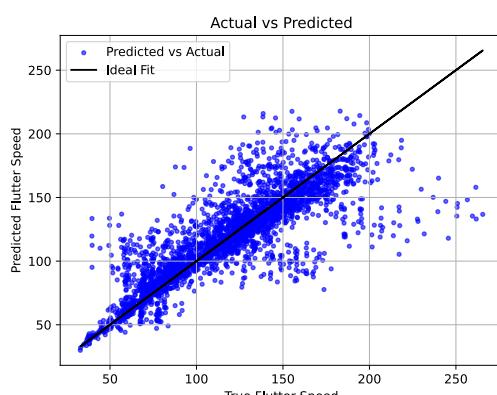
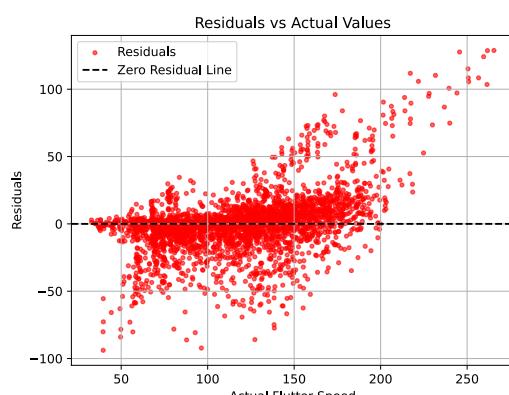
1. Καθώς αυξάνεται το πάχος, αυξάνεται και η προβλεπόμενη ταχύτητα πτερυγισμού.

2. Οι γωνίες $\theta_1 - \theta_3$ έχουν μια ισχυρά μη γραμμική συσχέτιση με την ταχύτητα πτερυγισμού χωρίς εμφανείς τάσεις.
3. Κατά μήκος της διαγωνίου των ιστογραμμάτων που αντιπροσωπεύουν τη συσχέτιση μιας μεταβλητής με τον εαυτό της, παρατηρείται ότι τα δεδομένα είναι κατανεμημένα τυχαία και ομοιόμορφα σε όλο το εύρος των πιθανών τιμών.

4.5.2 Νευρωνικό Δίκτυο με 1 Κρυφό Στρώμα



(α') training history

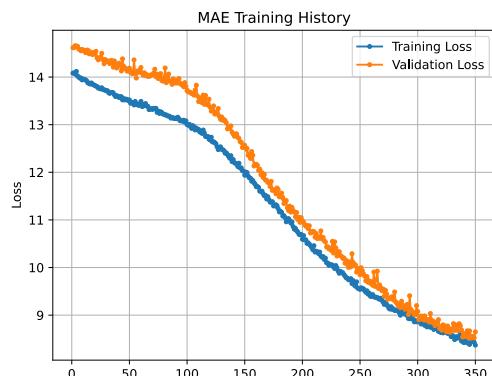
(β') Y_{true} vs Y_{pred} 

(γ') Residuals

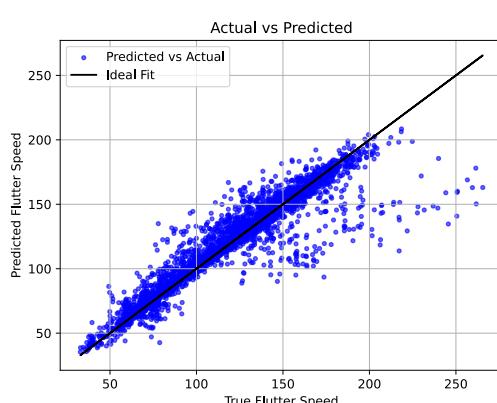
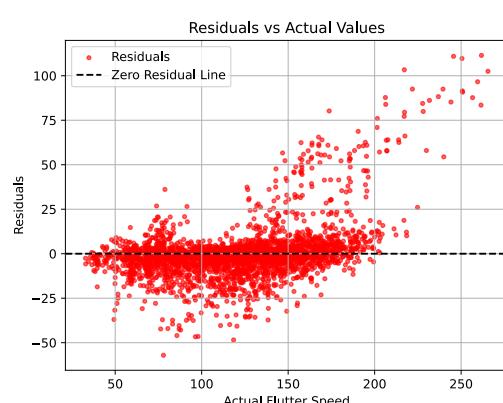
Σχήμα 4.17: Απόδοση Νευρωνικού δικτύου με 1 κρυφό στρώμα

Από το Σχήμα 4.17 καθίσταται προφανές ότι αν και αυτό το μοντέλο με μόνο ένα κρυφό στρώμα εκπαιδεύτηκε για 400 epochs, οι οποίες είναι αρκετές όπως θα φανεί από τα επόμενα αποτελέσματα, το Mean Average Error των δεδομένων επικύρωσης είναι πολύ μεγάλο για να είναι χρήσιμο το δίκτυο. Η διακύμανση πάνω και κάτω από τη μηδενική γραμμή στο Σχήμα 4.17 (γ') είναι πολύ μεγάλη, με πολλά σημεία να υπερεκτιμούνται ή να υπερεκτιμούνται κατά περισσότερο από $50m/s$.

4.5.3 Νευρωνικό Δίκτυο με 2 Κρυφά Στρώματα



(α') training history

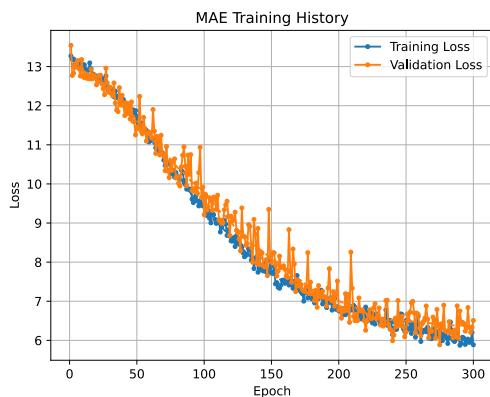
(β') Y_{true} vs Y_{pred} 

(γ') Residuals

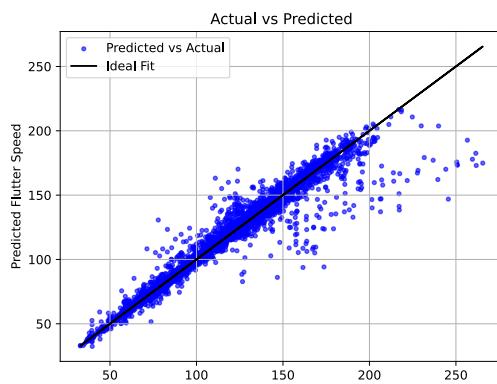
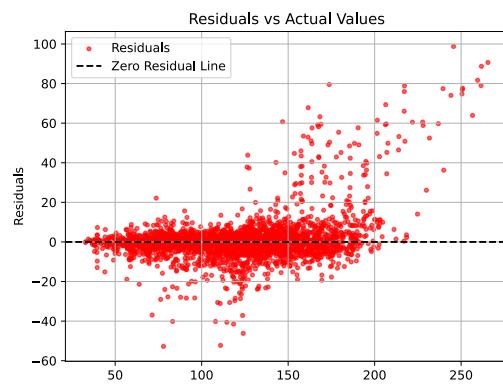
Σχήμα 4.18: Απόδοση Νευρωνικού δικτύου με 2 κρυφά στρώματα

Αυτό το μοντέλο εκπαιδεύτηκε για 350 epochs. Τα 2 κρυφά στρώματα αποδίδουν πολύ καλύτερα από ένα μόνο στρώμα. Το Mean Average Error στα δεδομένα επικύρωσης στο τέλος της εκπαίδευσης είναι περίπου $7m/s$. Το μόνο πρόβλημα είναι ότι εξακολουθεί να υπάρχει μεγάλη διακύμανση στις προβλέψεις, όπως φαίνεται από το Σχήμα 4.18.

4.5.4 Νευρωνικό Δίκτυο με 4 Κρυφά Στρώματα



(α') training history

(β') Y_{true} vs Y_{pred} 

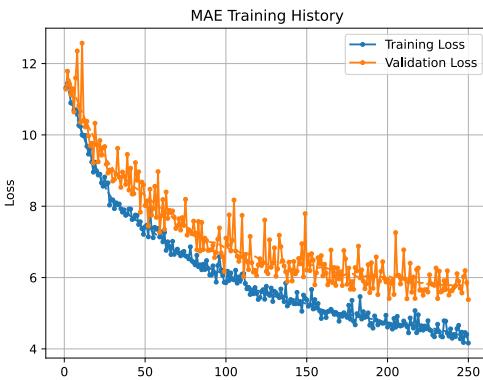
(γ') Residuals

Σχήμα 4.19: Απόδοση Νευρωνικού δικτύου με 4 κρυφά στρώματα

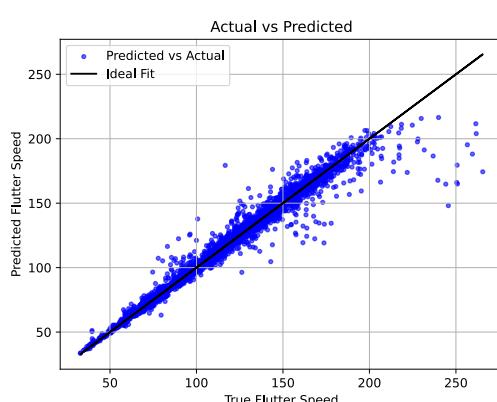
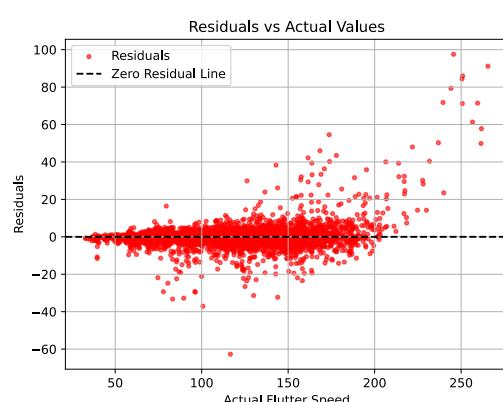
Το νευρωνικό δίκτυο με τέσσερα κρυφά στρώματα εκπαιδεύτηκε για 300 epochs, διότι το validation loss αρχίζει να εξομαλύνεται, όπως φαίνεται από το Σχήμα 4.19 (α'). Η

κατανομή των δεδομένων γύρω από τη γραμμή μηδενικής διαφοράς δεν είναι πολύ πιο στενή σε σχέση με το προηγούμενο μοντέλο, όπως φαίνεται στο Σχήμα 4.19 (γ').

4.5.5 Νευρωνικό Δίκτυο με 6 Κρυφά Στρώματα



(α') training history

(β') Y_{true} vs Y_{pred} 

(γ') Residuals

Σχήμα 4.20: Απόδοση Νευρωνικού δικτύου με 6 κρυφά στρώματα

Το νευρωνικό δίκτυο με 6 κρυφά στρώματα εκπαιδεύτηκε για 250 epochs για να αποφεύχθεί η υπερεκπαίδευση (overfitting), καθώς η απώλεια επικύρωσης παραμένει στάσιμη μετά την εποχή 220. Αυτό φαίνεται στο Σχήμα 4.20.

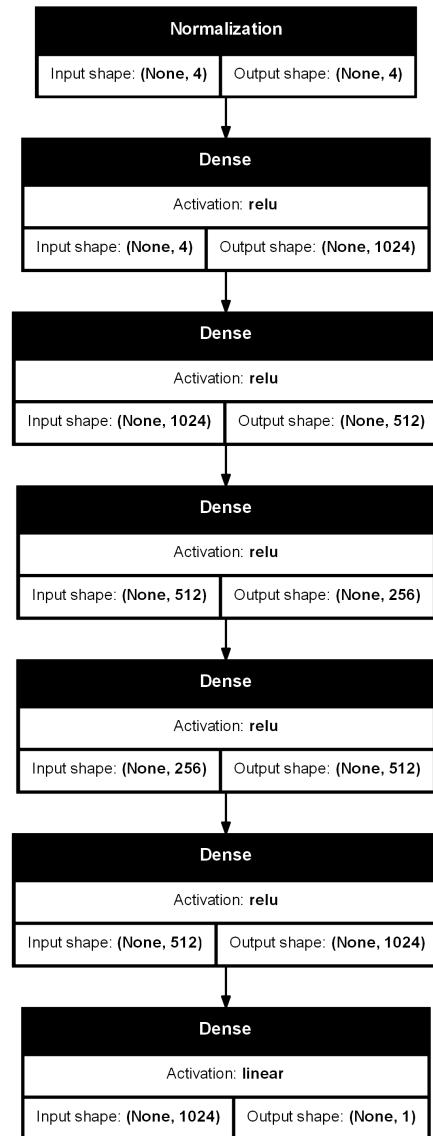
Η κατανομή των residuals είναι καλύτερη αλλά εξακολουθεί να είναι υψηλή, αν και βελτιώνεται σε σχέση με το προηγούμενο μοντέλο, όπως φαίνεται στο Σχήμα Σχήμα 4.20 (γ').

4.5.6 Νευρωνικό Δίκτυο με Ρυθμισμένες Υπερπαραμέτρους

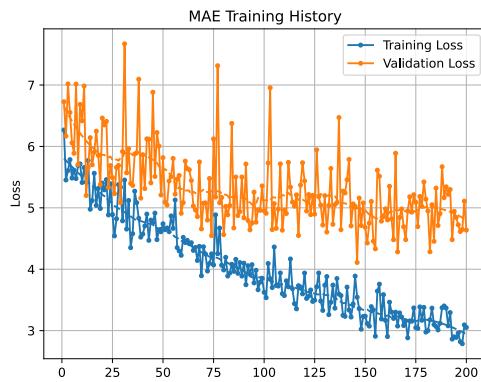
Οι υπερπαράμετροι που επιλέχθηκαν από τον αλγόριθμο HyperBand, ο οποίος περιγράφηκε στην υποενότητα 3.4.3, είναι οι εξής:

- Αριθμός κρυφών στρωμάτων: 5.
- Αριθμός νευρώνων για κάθε στρώμα: 1024, 512, 256, 512, και 1024 αντίστοιχα
- Συνάρτηση ενεργοποίησης για κάθε στρώμα: ReLu.
- Ρυθμός μάθησης του Adam Optimizer: 0.001.

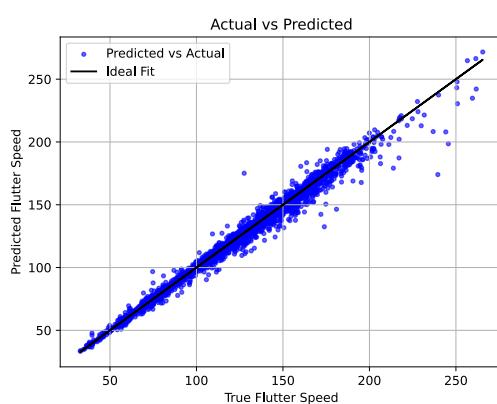
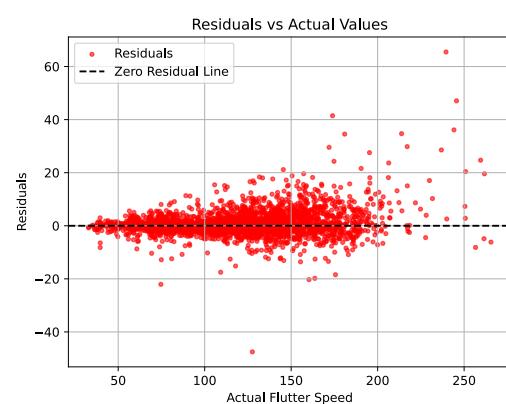
Η προκύπτουσα δομή του μοντέλου μπορεί να φανεί στο Σχήμα 4.21.



Σχήμα 4.21: Δομή βελτιστοποιημένου Νευρωνικού Δικτύου.



(α') training history

(β') Y_{true} vs Y_{pred} 

(γ') Residuals

Σχήμα 4.22: Απόδοση Hyperparameter tuned Νευρωνικού δικτύου

Όπως φαίνεται από το Σχήμα 4.22, αυτό το μοντέλο εκπαιδεύτηκε για 200 epochs, καθώς το MAE επικύρωσης σταματά να μειώνεται μετά από αυτό το σημείο. Αυτό το μοντέλο παρουσιάζει το χαμηλότερο MAE από όλα τα υπόλοιπα μοντέλα, με $MAE < 5m/s$ στην επικύρωση.

Επιπλέον, η κατανομή των δεδομένων γύρω από τη Γραμμή Μηδενικής Διαφοράς στο Σχήμα 4.22 (γ'), είναι η πιο στενή από όλα τα μοντέλα, παράγοντας έτσι τα καλύτερα αποτελέσματα μέχρι στιγμής. Αξιοσημείωτο είναι επίσης ότι αυτό το μοντέλο έχει τον υψηλότερο αριθμό παραμέτρων που μπορούν να εκπαιδευτούν και απαιτεί τον περισσότερο χρόνο εκπαίδευσης.

Κεφάλαιο 5

Συμπεράσματα & Μελλοντική Εργασία

Σε αυτό το κεφάλαιο παρουσιάζεται μια περίληψη των αποτελεσμάτων, εξάγονται γενικά συμπεράσματα και προτείνονται ιδέες για περαιτέρω ανάπτυξη.

5.1 Βελτιστοποίηση

Στο παρόν κεφάλαιο, παρουσιάζεται μια περίληψη των αποτελεσμάτων που παράχθηκαν σε αυτή την εργασία και εξάγονται γενικά συμπεράσματα επ' αυτών.

Στον Πίνακα 5.1 παρατίθεται μια περίληψη των αποτελεσμάτων που αποκτήθηκαν με την εκάστοτε μέθοδο βελτιστοποίησης που χρησιμοποιήθηκε στην παρούσα εργασία, καθώς και ένας πίνακας, Πίνακα 5.2, με την ποσοστιαία διαφορά από την αρχική ανάλυση.

Method	Flutter Velocity (m/s)	Mass (Kg)	Divergent Mode
Initial Flutter Characteristics	94.11	67.33	3
Powell's Method Scenario 1	99.48	67.33	3
Powell's Method Scenario 2	175.28	67.33	3
Genetic Algorithm	214.69	45.1	1

Table 5.1: Σύνοψη των αποτελεσμάτων βελτιστοποίησης

Method	Velocity Change (%)	Mass Change (%)
Powell's Method Scenario 1	5.71%	0.00%
Powell's Method Scenario 2	86.25%	0.00%
Genetic Algorithm	128.13%	-33.02%

Table 5.2: Σύγκριση των αποτελεσμάτων των μεθόδων βελτιστοποίησης

Αυτό που είναι σαφές από τη σύνοψη των αποτελεσμάτων είναι ότι υπάρχει σημαντικός χώρος για βελτίωση σε σχέση με την αρχική λύση. Η επίδραση των γωνιών των στρώσεων είναι αξιοσημείωτη και κάτι που πρέπει να λαμβάνεται υπ' όψιν κατά το σχεδιασμό ενός φτερού αεροπλάνου με composite laminate υλικά. Αυτού του είδους η βελτιστοποίηση αξίζει να πραγματοποιείται, διότι μπορεί ενδεχομένως να μειώσει τη μάζα του αεροσκάφους, με αποτέλεσμα βελτιωμένες επιδόσεις πτήσης, χειρισμό και οικονομία καυσίμου. Αξιοσημείωτο είναι, ωστόσο, ότι η δυναμική αστάθεια πτερυγισμού δεν είναι ο μόνος περιοριστικός παράγοντας για τη δομή της κύριας πτέρυγας και υπάρχουν άλλοι παράγοντες που πρέπει να ληφθούν υπ' όψιν. Ένας κύριος εξ αυτών είναι η παραμόρφωση της πτέρυγας και η δυναμική απόκριση υπό διαφορετικά φορτία και γωνίες προσβολής. Ένας άλλος παράγοντας είναι τα φαινόμενα στατικής αεροελαστικότητας και η αντιστροφή του ελέγχου πτήσης.

Από τον Πίνακα 5.2 φαίνεται ότι το καλύτερο αποτέλεσμα επιτυγχάνεται με τη χρήση του γενετικού αλγορίθμου, ο οποίος καταφέρνει να μειώσει τη μάζα της πτέρυγας κατά ένα διόλου αμελητέο 33%, ενώ ταυτόχρονα καταφέρνει να αυξήσει την ταχύτητα πτερυγισμού κατά 128%. Αυτό το εντυπωσιακό αποτέλεσμα πρέπει να αντιμετωπιστεί με πολύ προσοχή, διότι όπως έχει συζητηθεί στην ενότητα 4.4, η λύση φαίνεται να καθίσταται ασταθής νωρίτερα, αν και το πρόσημο της απόσβεσης της πρώτης ιδιομορφής δεν αλλάζει μέχρι τα $214m/s$.

5.2 Πρόβλεψη με Νευρωνικά Δίκτυα

Από την ανάπτυξη Νευρωνικών Δικτύων για την πρόβλεψη της ταχύτητας πτερυγισμού αυτής της γεωμετρίας, καταλήξαμε στο συμπέρασμα ότι απαιτείται ένα αρκετά περίπλοκο νευρωνικό δίκτυο για μία ικανοποιητική ακρίβεια στις προβλέψεις.

Επιπρόσθετα, τα δεδομένα που απαιτούνται για την εκπαίδευση του νευρωνικού, είναι υπολογιστικά ασύμφορο να αποκτηθούν και, στις περισσότερες περιπτώσεις, η υπολογιστική πολυπλοκότητα για τη δημιουργία των δεδομένων εκπαίδευσης ξεπερνά κατά πολύ την αντίστοιχη για την πραγματοποίηση μιας άμεσης βελτιστοποίησης.

Η ακρίβεια των προβλεπόμενων αποτελεσμάτων είναι αρκετά καλή στις περισσότερες περιπτώσεις, αλλά όχι αρκετή για ακριβή πρόβλεψη πτερυγισμού σε κάθε περίπτωση, διότι υπάρχουν μερικά outliers στα δεδομένα όπου η πρόβλεψη δεν είναι ικανοποιητική.

Αξιοσημείωτο είναι ότι μετά την εκπαίδευση του μοντέλου, απαιτείται σχεδόν μηδενικό υπολογιστικό κόστος για να πραγματοποιηθεί μια πρόβλεψη σχετικά με την ταχύτητα πτερυγισμού. Το Νευρωνικό Δίκτυο μπορεί στη συνέχεια να χρησιμοποιηθεί για τη βελτιστοποίηση της κατασκευής, αν και η ακρίβεια των αποτελεσμάτων δεν είναι εγγυημένη.

5.3 Μελλοντική Εργασία

Η παρούσα μελέτη παρέχει κάποιες εφαρμοσμένες πρακτικές και ερμηνείες στην ανάλυση πτερυγισμού και τις τεχνικές προσαρμογής πτερυγισμού χρησιμοποιώντας μεθόδους βελτιστοποίησης και Python, αλλά παραμένουν αρκετές ανεξερεύνητες περιοχές που θα μπορούσαν να διαλευκανθούν με περαιτέρω μελέτη.

Οι μελλοντικές προσπάθειες θα πρέπει να περιλαμβάνουν τουλάχιστον μια πειραματική επικύρωση των αποτελεσμάτων της προσομοίωσης. Με αυτόν τον τρόπο, η επίδραση των γωνιών των στρώσεων του σύνθετου υλικού μπορεί να επιβεβαιωθεί στην πράξη.

Αν και πολλοί αλγόριθμοι βελτιστοποίησης χρησιμοποιήθηκαν σε αυτή τη μελέτη με υποσχόμενα αποτελέσματα, παραμένουν ανεξερεύνητες πολλές προηγμένες τεχνικές. Κάποια μελλοντική εργασία θα μπορούσε να εξετάσει αυτούς τους πιο πολυδιάστατους αλγορίθμους, καθώς και την ενδεχόμενη εφαρμογή μεθόδων βασισμένων στη μηχανική μάθηση για περαιτέρω ενίσχυση των διαδικασιών βελτιστοποίησης.

Το φαινόμενο του πτερυγισμού που μελετήθηκε σε αυτή τη διατριβή, μοντελοποιημένο με την αεροδυναμική θεωρία Vortex Lattice σε συνδυασμό με έναν επιλυτή βασισμένο στην ανάλυση ιδιομορφών, θα μπορούσε επίσης να προσομοιωθεί μέσω μιας transient ανάλυσης αλληλεπίδρασης Ρευστού-Κατασκευής (FSI). Αυτό θα επέτρεπε μια σύγκριση των αποτελεσμάτων μεταξύ των δύο προσεγγίσεων, παρέχοντας περαιτέρω επικύρωση και σύγκριση των αποτελεσμάτων διαφορετικών μεθόδων.

Τέλος, η ενσωμάτωση των Νευρωνικών Δικτύων Ενημερωμένων από Φυσική (PINNs) σε αυτή την εφαρμογή, πρόκειται για μια υποσχόμενη κατεύθυνση για μελλοντική έρευνα. Τα PINNs είναι ένας αναδυόμενος κλάδος της υπολογιστικής μηχανικής, που συνδυάζει την εκμάθηση βασισμένη σε δεδομένα με τους φυσικούς νόμους που διέπουν την εκάστοτε εφαρμογή, όπως οι εξισώσεις της ροής ρευστού και της δομικής δυναμικής. Ενσωματώνοντας τη φυσική άμεσα στη διαδικασία εκπαίδευσης του νευρωνικού δικτύου, τα PINNs μπορούν να προσφέρουν μια πιο αποτελεσματική και ακριβή λύση σε περίπλοκα προβλήματα αλληλεπίδρασης ρευστού-κατασκευής. Η δυνατότητά τους να παρακάμψουν κάποιους από τους περιορισμούς των παραδοσιακών αριθμητικών μεθόδων, όπως η δημιουργία πλέγματος και τα ζητήματα σύγκλισης του επιλυτή, τα καθιστά ιδιαίτερα ελκυστικά για τη μελέτη μηχανικών συστημάτων.

Bibliography

- [1] G.A. Bolla. Aeroelastic study of the flutter conditions of an aircraft wing. Master's thesis, Universitat Politecnica de Catalunya, 2022.
- [2] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice Hall, New Jersey, 1973.
- [3] Alexander Schleicher GmbH & Co. Asw 28 glider information, 2025.
- [4] Altair Community. Flutter analysis setup - tips & tricks, 2025. Accessed: 2025-03-01.
- [5] Ahmed Fawzy Gad. Pygad: An intuitive genetic algorithm python library. *Multimedia Tools and Applications*, pages 1–14, 2023.
- [6] Jonathan E. Cooper Jan R. Wright. *Dynamic Aeroelasticity – Flutter*, chapter 11, pages 167–198. John Wiley & Sons, Ltd, 2014.
- [7] Joseph Katz. *Low-Speed Aerodynamics*. Cambridge University Press, second edition, 2001.
- [8] Mykel L. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. The MIT Press, London, 2019.
- [9] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization, 2018.
- [10] LLC MatWeb. Matweb material property data, 2025. Accessed: 2025-03-01.
- [11] E. O nate. *Structural Analysis with the Finite Element Method, vol. 2*. Springer, Barcelona, 2013.

- [12] S. Pinzon. Introduction to vortex lattice theory. *Ciencia y Poder Aereo*, 10:39–48, 2015.
- [13] H. M. Software. *MSC Nastran 2021.1 Aeroelastic User's Guide*, 2021.
- [14] A. Takyar. Neural networks: Architecture, types, working, applications, case studies, development and implementation, 2025.
- [15] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.
- [16] Changchuan Xie, Yi Liu, Chao Yang, and J. E. Cooper. Geometrically nonlinear aeroelastic stability analysis and wind tunnel test validation of a very flexible wing. *Shock and Vibration*, 2016(1):5090719, 2016.

Παράρτημα Α'

Κώδικας Βελτιστοποίησης Python

Στο παράρτημα αυτό παρουσιάζεται ο κύριος κώδικας για την απλούστερη περίπτωση βελτιστοποίησης χρησιμοποιώντας τη μέθοδο Powell. Ο κώδικας για όλες τις άλλες εφαρμογές είναι τροποποιημένος, αλλά η βασική λογική και πολλά μέρη παραμένουν ίδια σε όλη τη διαδικασία. Όλος ο κώδικας, συμπεριλαμβανομένου του κώδικα που χρησιμοποιήθηκε για την επεξεργασία των αποτελεσμάτων και την παραγωγή των γραφημάτων σε αυτή τη διατριβή, είναι διαθέσιμος στο GitHub: [vas xen/Aeroelastic_Flutter](https://github.com/vas xen/Aeroelastic_Flutter) (github.com).

Listing A.1: Κώδικας Python για βελτιστοποίηση με τη μέθοδο Powell

```
1 import os
2 import subprocess
3 import pandas as pd
4 import numpy as np
5 from enum import Enum
6 from scipy.optimize import minimize, Bounds
7 from numpy.typing import NDArray
8 from dataclasses import dataclass
9 from typing import List, Tuple, Callable, Any, Dict
10 from matplotlib.axes import Axes
11 from matplotlib.lines import Line2D
12 from matplotlib.figure import Figure
13 from matplotlib import pyplot as plt
14 import pickle
15 # from time import time
16
17
18 # ----- Classes -----
19 @dataclass
```

```

20 class MAT8():
21     '''A MAT8 class which contains all the information of a MAT8 orthotropic
22         material bulk data entry in optistruct'''
23     LineIndex: int
24     MID: int
25     E1: float
26     E2: float
27     NU12: float
28     G12: float
29     G1Z: str
30     G2Z: str
31     RHO: float
32
33     def to_string(self) -> str:
34         ''' This methods returns the information contained in the class in a string
35             format identical to the one found in a .fem input file'''
36         s = f'MAT8,{self.MID},{self.E1},{self.E2},{self.NU12},{self.G12},{self.G1Z}
37             },{self.G2Z},{self.RHO},'
38         return s
39
40 @dataclass
41 class Ply():
42     '''A Ply class which contains all the information of a Ply of an PCOMP composite
43         shell property bulk data entry in optistruct'''
44     index: int
45     MID: int
46     Thickness: float
47     Theta: float
48
49     def to_string(self, SOUT: str = 'NO') -> str:
50         ''' This methods returns the information contained in the class in a string
51             format identical to the one found in a .fem input file'''
52         s = f'{self.MID},{self.Thickness},{self.Theta},{SOUT}'
53         return s
54
55 class PCOMP():
56     '''A PCOMP class which contains all the information of a Ply of an PCOMP
57         composite shell property bulk data entry in optistruct'''
58
59     def __init__(self, Id: int, Plies: List[Ply], original_txt_lines: List[str],
60                  Indeces: Tuple[int,int]):
61         self._Id = Id
62         self._Plies = Plies
63         self._original_txt_lines = original_txt_lines
64         self._Indeces = Indeces

```

```

59
60 #===== PROPERTIES =====
61
62 # ----- Id -----
63 @property
64 def Id(self):
65     return self._Id
66
67 @Id.setter
68 def Id(self, val):
69     raise AttributeError('Id is immutable and cannot be changed')
70 # ----- // -----
71
72 # ----- Plies -----
73 @property
74 def Plies(self):
75     return self._Plies
76
77 @Plies.setter
78 def Plies(self, val: List[Ply]):
79     self._Plies = val
80 # ----- // -----
81
82 # ----- Original_txt -----
83 @property
84 def original_txt_lines(self):
85     return self._original_txt_lines
86
87 @original_txt_lines.setter
88 def original_txt_lines(self, val):
89     raise AttributeError('original_txt_lines is immutable and cannot be changed')
90 # ----- // -----
91
92 # ----- Indeces -----
93 @property
94 def Indeces(self):
95     return self._Indeces
96
97 @Indeces.setter
98 def Indeces(self, val: Tuple[int, int]):
99     raise AttributeError('Indeces is immutable and cannot be changed')
100
101 # ----- // -----
102
103

```

```

104     # ----- NumPlies -----
105     @property
106     def NumPlies(self) -> int:
107         return len(self.Plies)
108     # ----- // -----
109
110
111     #===== METHODS =====
112
113     def to_string(self) -> List[str]:
114         ''' This methods returns the information contained in the class in a string
115             format identical to the one found in a .fem input file'''
116
117         Lines: List[str] = []
118         Lines.append(self.original_txt_lines[0][:-1])
119
120         for i in range(0, self.NumPlies - 1, 2):
121             ply1 = self.Plies[i]
122             ply2 = self.Plies[i + 1]
123             line = f'+,{ply1.to_string()},,{ply2.to_string()},'
124             Lines.append(line)
125
126         if i + 1 < self.NumPlies - 1:
127             line = f'+,{self.Plies[-1].to_string()},'
128             Lines.append(line)
129
130         return Lines
131
132 @dataclass
133 class FlutterAnalysisPoint():
134     ModeNumber : int
135     MachNumber : float
136     DensityRatio: float
137     Method: str
138     Data: pd.DataFrame
139
140     def DetectFlutter(self) -> Tuple[List[float], List[Tuple[int, int]]]:
141         Stable = self.Data['DAMPING'] < 0
142         FlutterIndeces: List[Tuple[int, int]] = []
143         for i in range(1, Stable.shape[0]):
144             if (Stable[i-1]) and (not Stable[i]):
145                 FlutterIndeces.append((i-1, i))
146
147         FlutterSpeed = []
148         for ind in FlutterIndeces:

```

```

149     D2 = self.Data[ 'DAMPING' ][ind[1]]
150     V1 = self.Data[ 'VELOCITY' ][ind[0]]
151     V2 = self.Data[ 'VELOCITY' ][ind[1]]
152     m = (D2 - D1) / (V2 - V1)
153     Vflutter = V1 - D1/m
154     FlutterSpeed.append(Vflutter)
155
156     return FlutterSpeed, FlutterIndeces
157
158 def Plot(self,ax: Axes, ix: int, iy: int, label: str = '') -> Line2D:
159     line, = ax.plot(self.Data.iloc[:,ix], self.Data.iloc[:,iy], marker = '.', 
160                     label = label )
161     return line
162
163 def PlotDamping(self) -> Figure:
164     fig, ax = plt.subplots()
165     self.Plot(ax, 2, 3, f'MODE {self.ModeNumber}')
166     ax.set_xlabel('VELOCITY')
167     ax.set_ylabel('DAMPING')
168     ax.legend()
169     return fig
170
171
172 def PlotFrequency(self) -> Figure:
173     fig, ax = plt.subplots()
174     self.Plot(ax, 2, 4, f'MODE {self.ModeNumber}')
175     ax.set_xlabel('VELOCITY')
176     ax.set_ylabel('FREQUENCY')
177     ax.legend()
178     return fig
179
180
181
182 @dataclass
183 class FlutterSubcase():
184     SubcaseId: int
185     XY_Symmetry: bool
186     XZ_Symmetry: bool
187     Method: str
188     NumPoints: int
189     Points: List[FlutterAnalysisPoint]
190
191     def FlutterInfo(self):
192         FlutterInfo: Dict[int, float] = {}
193         for point in self.Points:

```

```

194         Vel, _ = point.DetectFlutter()
195         if Vel: FlutterInfo[point.ModeNumber] = min(Vel)
196
197     return FlutterInfo
198
199 class FlutterSummary():
200
201     @staticmethod
202     def __split_subcases(lines: List[str]) -> List[List[str]]:
203         subcase_id = []
204         for item in lines:
205             if item.startswith('Subcase'):
206                 split = item.split()
207                 subcase_id.append(int(split[2]))
208
209         subcase_id = np.array(subcase_id)
210         Subcaseschange: List = (np.where(subcase_id[:-1] != subcase_id[1:])[0] + 1).
211             tolist()
212         Subcaseschange.insert(0, 0)
213         Subcaseschange.append(len(lines))
214         subcase_indeces = Subcaseschange
215
216         Subcases = []
217         for i in range(len(subcase_indeces) - 1):
218             Subcases.append(lines[subcase_indeces[i]:subcase_indeces[i+1]])
219         return Subcases
220
221     @staticmethod
222     def __split_points(lines: List[str]) -> List[List[str]]:
223         point_indeces: List[int] = []
224         for ind, item in enumerate(lines):
225             if item.startswith('Subcase'):
226                 point_indeces.append(ind)
227
228         point_indeces.append(len(lines))
229
230         points: List[List[str]] = []
231         for i in range(len(point_indeces) - 1):
232             points.append(lines[point_indeces[i]: point_indeces[i+1]])
233         return points
234
235     @staticmethod
236     def __read_analysis_point(point: List[str]) -> FlutterAnalysisPoint :
237         def remove_letters(input_string: str) -> str:
238             tempresult = ''

```

```

239     for char in input_string:
240         # Check if the character is not a letter
241         if not char.isalpha():
242             tempresult += char
243
244     return tempresult.replace('+', 'E+')
245
246 def read_header(header: str) -> Tuple[int, float, float, str]:
247     split = header.strip().split('=')
248     # assert len(split) == 12, 'length of header of analysis point incorrect
249
250     headerdata =[remove_letters(item) for item in split[1:-1]]
251     headerdata.append(split[-1].strip())
252     out = (int(headerdata[0]), float(headerdata[1]), float(headerdata[2]),
253            str(headerdata[3]))
254     return out
255
256
257 def read_data(data: List[str]) -> pd.DataFrame:
258     # Initialize an empty list to store rows
259     rows = []
260
261     # Iterate over each string in the input list
262     for string in data:
263         # Split the string into numbers
264         numbers = string.split()
265         row = [float(num) for num in numbers]
266         rows.append(row)
267
268
269     # Create a DataFrame from the list of rows
270     df = pd.DataFrame(rows, columns = ['KFREQ', '1/KFREQ', 'VELOCITY',
271                        'DAMPING', 'FREQUENCY', 'COMPLEX', 'EIGENVALUE'])
272     return df
273
274
275     header = point[3]
276     data = point[5:]
277
278     header = read_header(header)
279     data = read_data(data)
280     return FlutterAnalysisPoint(*header, data)
281
282 @staticmethod
283 def __read_subcase_header(subcase_header: List[str]) -> Tuple[int, bool, bool]:
284     Id = int(subcase_header[0].split('=')[1])
285     row3 = subcase_header[2]
286     row3eqsplit = row3.split('=')
287     row3split = []

```

```

282     for s in row3eqsplit:
283         row3split.extend(s.split())
284
285     XY_Symmetry: bool = False
286     XZ_Symmetry: bool = False
287     if row3split[2] == 'SYMMETRIC':
288         XY_Symmetry = True
289
290     if row3split[5] == 'SYMMETRIC':
291         XZ_Symmetry = True
292     return (Id, XY_Symmetry, XZ_Symmetry)
293
294 def __init__(self, filepath: str):
295     self.title: str = ''
296     self.Filepath: str = ''
297     self.NumSubcases: int = 0
298     self.Subcases: List[FlutterSubcase] = []
299
300     assert os.path.isfile(filepath), f'File at {filepath} does not exist'
301
302     #Title
303     self.title = os.path.basename(filepath).split('.')[0]
304
305     #Filepath
306     self.Filepath = filepath
307
308     #Subcases
309     with open(filepath, 'r') as file:
310         lines = file.readlines()
311     stripedlines = []
312     for line in lines:
313         if line.isspace():
314             lines.remove(line)
315         else:
316             stripedlines.append(line.strip())
317
318     # Split on Subcase level
319     subcases = self.__split_subcases(stripedlines)
320     Subcases: List[FlutterSubcase] = []
321     for subcase in subcases:
322         subcase_header = subcase[0:3]
323         SubcaseHeader = self.__read_subcase_header(subcase_header)
324         points = self.__split_points(subcase)
325         Points: List[FlutterAnalysisPoint] = []
326         for point in points:
327             Points.append(self.__read_analysis_point(point))

```

```

328
329     SubcaseMethod = Points[0].Method
330     Subcases.append(FlutterSubcase(*SubcaseHeader, SubcaseMethod, len(Points
331         ), Points))
332
333     self.Subcases = Subcases
334
335     #Number of Subcases
336     self.NumSubcases = len(Subcases)
337
338     def FlutterInfo(self) -> str:
339         s = f'\n{n***** FLUTTER INFORMATION *****\n\n'
340         for subcase in self.Subcases:
341             s += f'----- SUBCASE {subcase.SubcaseId} ----- |\n'
342             s += '| '
343             for point, vel in subcase.FlutterInfo().items():
344                 s += f'| POINT {point} -> {round(vel, 2)} m/s |\n'
345         return s
346
347 class PlySymmetry(Enum):
348     AntiSymmetric = -1
349     NoSymmetry = 0
350     Symmetric = 1
351
352     class ToleranceWrapper():
353         def __init__(self, func: Callable,
354                      ThicknessTolerance: float,
355                      AngleTolerance: float,
356                      FlutterVelocityConstraint: float,
357                      inputfile: str,
358                      solverpath: str,
359                      Penalty: float = 1E10,
360                      PlySymmetry: PlySymmetry = PlySymmetry.AntiSymmetric
361                      ):
362             self.func = func
363             self.ThicknessTolerance = ThicknessTolerance
364             self.AngleTolerance = AngleTolerance
365             self.FlutterVelocityConstraint = FlutterVelocityConstraint
366             self.inputfile = inputfile
367             self.solverpath = solverpath
368             self.Penalty = Penalty
369             self.PlySymmetry = PlySymmetry
370             self.cache: Dict[Tuple[float], float] = {}
371             self.history: Dict[Tuple[float], float] = {}
372
373         def __call__(self, x: NDArray[np.float64]) -> float:

```

```

372     t = x[0]
373     t = np.round(t / self.ThicknessTolerance) * self.ThicknessTolerance
374
375     a = x[1:]
376     a = (np.round(a / self.AngleTolerance) * self.AngleTolerance).tolist()
377     NumLayers = len(a)
378     t = NumLayers * [t]
379     rounded_input = (*t, *a)
380     if rounded_input in self.cache:
381         result = self.cache[rounded_input]
382         self.history[rounded_input] = result
383         return result
384     else:
385         result = self.func(t, a, self.inputfile, self.solverpath, self.
386                             FlutterVelocityConstraint, self.PlySymmetry, self.Penalty)
387         self.history[rounded_input] = result
388         self.cache[rounded_input] = result
389         return result
390
391     def __str__(self) -> str:
392         s = f'Thickness Tolerance: {self.ThicknessTolerance}\n'
393         s += f'Angle Tolerance: {self.AngleTolerance}\n'
394         s += '\nCached Data:\n'
395         for k, v in self.cache.items():
396             s += f'x = {k} -> f(x) = {v}\n'
397         return s
398
399     def savecache(self, file: str) -> None:
400         assert file.endswith('.xlsx'), f'The file must be an .xlsx file not a .{file
401             .split('.')[1]} file'
402         Dataframe = pd.DataFrame(list(self.cache.items()), columns = ['Input Vector',
403                                   'Function Value'])
404         Dataframe.to_excel(file)
405         return
406
407     def savehistory(self, file: str) -> None:
408         assert file.endswith('.xlsx'), f'The file must be an .xlsx file not a .{file
409             .split('.')[1]} file'
410         Dataframe = pd.DataFrame(list(self.history.items()), columns = ['Input
411             Vector', 'Function Value'])
412         Dataframe.to_excel(file)
413         return
414
415 # ----- Functions -----
416 def ReadFem(inputfile: str) -> Tuple[List[PCOMP], List[MAT8]]:
417     '''This function reads a .fem optistruct input file and detects any PCOMP and

```

```

any MAT8 entries. This function assumes that PCOMP refers exclusively to MAT8
material types.
413    ### Parameters:
414    - inputfile: a file path to a .fem file given as a string
415    ### Returns:
416    - A List of PCOMP objects
417    - A list of MAT8 objects ''
418    def readply(PlyIndex: int, input: List[str]) -> Ply:
419        Mid = int(input[0])
420        Thickness = float(input[1])
421        Theta = float(input[2])
422        return Ply(PlyIndex, Mid, Thickness, Theta)
423
424    def readPCOMP(StartIndex: List[int]) -> List[PCOMP]:
425        numPCOMP = len(StartIndex)
426        EndIndex: List[int] = []
427        for index in StartIndex:
428            i = 1
429            while lines[index + i].startswith('+'):
430                i += 1
431
432            EndIndex.append(index + i)
433
434        PCOMPs: List[PCOMP] = []
435
436        for i in range(numPCOMP):
437            Section = lines[StartIndex[i] : EndIndex[i]]
438            Id = int(Section[0].split(',') [1])
439            Plies_str: List[List[str]] = []
440            for line in Section[1:]:
441                SplitLine = line.split(',')
442                lenSplit = len(SplitLine)
443                if lenSplit == 10:
444                    if SplitLine[8] == 'YES' or SplitLine[8] == 'NO':
445                        Plies_str.append(SplitLine[1:5])
446                        Plies_str.append(SplitLine[5:9])
447                    else:
448                        pass
449                elif lenSplit == 6:
450                    if SplitLine[4] == 'YES' or SplitLine[4] == 'NO':
451                        Plies_str.append(SplitLine[1:5])
452
453            Plies: List[Ply] = []
454            for Plyindex, ply in enumerate(Plies_str):
455                Plies.append(readply(Plyindex, ply))
456

```

```

457     PCOMPs.append(PCOMP(Id, Plies, Section, (StartIndex[i], EndIndex[i]) ))
458
459
460 def readMAT8(Indeces: List[int]) -> List[MAT8]:
461     MAT8s: List[MAT8] = []
462     for index in Indeces:
463         line = lines[index]
464         split = line.split(',') [1:]
465         MID = int(split[0])
466         E1 = float(split[1])
467         E2 = float(split[2])
468         NU12 = float(split[3])
469         G12 = float(split[4])
470         G1Z = split[5]
471         G2Z = split[6]
472         RHO = float(split[7])
473         M = MAT8(index, MID,E1,E2,NU12,G12,G1Z,G2Z,RHO)
474         MAT8s.append(M)
475
476
477     return MAT8s
478
479 if not inputfile.endswith('.fem'):
480     raise ValueError('Incorrect file type. File must be of type .fem')
481
482 with open(inputfile, 'r') as f:
483     lines = f.readlines()
484
485     StartIndex: List[int] = []
486     Mat8Indeces: List[int] = []
487
488     for index, line in enumerate(lines):
489         if line.startswith('PCOMP'):
490             StartIndex.append(index)
491         elif line.startswith('MAT8'):
492             Mat8Indeces.append(index)
493
494     PCOMPs = readPCOMP(StartIndex)
495     MAT8s = readMAT8(Mat8Indeces)
496
497     return PCOMPs, MAT8s
498
499 def WriteFem(Properties: List[PCOMP], Materials: List[MAT8], inputfile: str) -> None
500     :
501         '''Writes a back to a .fem file the modified PCOMP and MAT8 objects given in the
502             input. It is designed to be used on the same file as ReadFem function
503             ### Parameters:

```

```

501     - Properties: A list of PCOMP objects
502     - Materials: A list of MAT 8 objects
503     - inputfile: A path to a file as a string to which the objects will be returned
504     ''
505 def replace_lines_in_file(file_path: str, line_numbers : List[int], new_lines:
506     List[str]):
507     if len(line_numbers) != len(new_lines):
508         raise ValueError("The number of line numbers must match the number of
509             new lines.")
510
511     try:
512         # Read all lines from the file
513         with open(file_path, 'r') as file:
514             lines = file.readlines()
515
516             # Replace the specified lines
517             for line_number, new_line in zip(line_numbers, new_lines):
518                 if 0 <= line_number < len(lines):
519                     lines[line_number] = new_line + '\n'
520                 else:
521                     Warning(f"Warning: Line number {line_number} is out of range.
522                         Skipping this replacement.")
523
524             # Write the modified lines back to the file
525             with open(file_path, 'w') as file:
526                 file.writelines(lines)
527
528             print("Lines replaced successfully.")
529
530     except FileNotFoundError:
531         print(f"Error: The file {file_path} was not found.")
532     except Exception as e:
533         print(f"An error occurred: {e}")
534
535     if Properties:
536         for Pcomp in Properties:
537             LineIndeces = list(range(Pcomp.Indeces[0], Pcomp.Indeces[1]))
538             replace_lines_in_file(inputfile, LineIndeces, Pcomp.to_string())
539
540     if Materials:
541         for Mat in Materials:
542             replace_lines_in_file(inputfile, [Mat.LineIndex], [Mat.to_string()])
543
544 def readmass(outputfile: str) -> float:
545     ''' Reads a .out file and checks if it contains Mass information if it does it
546         returns the mass as a float otherwise it throws a Value Error

```

```

543     """ Parameters:
544     - outputfile: A path as a string to a .out file """
545     if not outputfile.endswith('.out'):
546         raise ValueError('Incorrect file type. File must be of type .out')
547
548     with open(outputfile, 'r') as f:
549         lines = f.readlines()
550
551     Mass: float = -1.0
552     for line in lines:
553         if 'Mass' in line:
554             MassString = line.split('Mass')[1]
555             MassString = MassString.replace('=', '')
556             MassString = MassString.strip()
557             Mass = float(MassString)
558             break
559
560     if Mass == -1.0: raise ValueError('Provided file does not contain any Mass
561                             information')
562
563     return Mass
564
564 def CallSolver(inputfile: str, solverpath: str, options: str) -> subprocess.
CompletedProcess:
    '''This function calls the optistruct solver on the input file.
    The function works by writing a temporary batch file the excecuting it and
    finally deleting it.
    It returns the std out to the consoleand the completedProcess object provided by
    the subprocess module
    """
    Parameters:
    - inputfile: a path as a string to a .fem solver input file
    - solverpath: the path to the altair yperwors solver scripts as a string
    - options: a string of options exactly as they would be set in the command
        window or in the altair compute consosle'''
    inputfile = f'{inputfile.replace('/', '\\\\)}'
    solverpath = f'{solverpath.replace('/', '\\\\)}'
    574
    575
    576    lines = ['@echo off\n',
    577              'optistruct ' + inputfile + ' ' + options]
    578
    579    with open('temp.bat', 'w') as file:
    580        file.writelines(lines)
    581
    582
    583    s = subprocess.run([f'temp.bat'])

```

```

584     # os.remove('temp.bat')
585     return s
586
587 def ObjectiveFunction(thicknesses: List[float], angles: List[float], inputfile: str,
588     solverpath: str, FlutterVelocityConstraint: float, sym: PlySymmetry, penalty:
589     float) -> float:
590     assert len(thicknesses) == len(angles), f'Thicknesses and angles must have the
591     same length'
592     assert all([e > 0 for e in thicknesses]), 'All thicknesses must be strictly
593     positive'
594
595     Properties, _ = ReadFem(inputfile)
596     assert len(Properties) == 1, f'Function expected only one PCOMP to optimize, {
597         len(Properties)} found'
598     Property = Properties[0]
599     # Property.to_string()
600
601     match sym.value:
602         case -1: #Antisymmetric
603             assert Property.NumPlies / 2 == len(thicknesses), f'for antisymmetric
604                 laminates the length of the inputs should be half the number of
605                 plies'
606             thicknesses.extend(thicknesses)
607             angles.extend([-e for e in angles])
608
609             for i in range(Property.NumPlies):
610                 Property.Plies[i].Thickness = thicknesses[i]
611                 Property.Plies[i].Theta = angles[i]
612
613         case 0: # No symmetry
614             assert Property.NumPlies == len(thicknesses), f'length of the number of
615                 inputs should be half the number of plies'
616             for i in range(Property.NumPlies):
617                 Property.Plies[i].Thickness = thicknesses[i]
618                 Property.Plies[i].Theta = angles[i]
619
620         case 1: # Symmetric
621             assert Property.NumPlies / 2 == len(thicknesses), f'for symmetric
622                 laminates the length of the inputs should be half the number of
623                 plies'
624             thicknesses.extend(thicknesses)
625             angles.extend(angles)
626
627             for i in range(Property.NumPlies):
628                 Property.Plies[i].Thickness = thicknesses[i]
629                 Property.Plies[i].Theta = angles[i]

```

```

620
621
622 WriteFem([Property], [], inputfile)
623 CallSolver(inputfile, solverpath, '-nt 6')
624 outfile = inputfile.replace('.fem', '.out')
625 fltfile = inputfile.replace('.fem', '.flt')
626
627 Mass = readmass(outfile)
628 Flutter = FlutterSummary(fltfile)
629 assert len(Flutter.Subcases) == 1, f'Analysis should include only one subcase
   not {len(Flutter.Subcases)}'
630 Subcase = Flutter.Subcases[0]
631
632 Velocities: List[float] = []
633 for point in Subcase.Points:
634     Vel, _ = point.DetectFlutter()
635     if Vel: Velocities.append(Vel[0])
636
637 P: float = 0
638 if Velocities:
639     FlutterVelocity = min(Velocities)
640     if FlutterVelocity < FlutterVelocityConstraint:
641         P = penalty * (FlutterVelocityConstraint - FlutterVelocity)
642
643 Objective = Mass + P
644
645 return Objective
646
647 def DeleteUnessesaryFiles(directory: str, FileExtensions: Tuple[str, ...]) -> None:
648     '''This function deletes all file with certain extension within a directory USE
       CAREFULLY
649     ### Parameters:
650     - directory: path a string to a directory (folder)
651     - FileExtensions: A tuple of string containing the extension that will be
       deleted including the dot eg: ('.txt', '.exe') '',
652     files = os.listdir(directory)
653
654     for file in files:
655         if file.endswith(FileExtensions):
656             os.remove(os.path.join(directory, file))
657
658 def main():
659     # Define optimization problem parameters
660     inputFile = "C:/Users/vasxen/OneDrive/Thesis/code/ASW28 Wing.fem"
661     solverpath = "C:/My_Programms/Altair/hwsolvers/scripts"
662     x0 = np.array([0.0005, 45, -45, 45], dtype = np.float64)# initial solution

```

```

    vector
663 lower_bounds = [0.0002] + 3 * [-90]                      # lower constraints of
   thickness and ply angles
664 upper_bounds = [0.0008] + 3 * [+90]                      # Upper constraints of
   thickness and ply angles
665 bounds = Bounds(lower_bounds, upper_bounds)             # type: ignore
666 options = {'disp' : True,
667     'maxfev' : 1000,
668     'return_all' : True}
669 WrappedObj = ToleranceWrapper(ObjectiveFunction, 0.0001, 1, 90, inputFile,
   solverpath)
670 Min = minimize(WrappedObj, x0 = x0, method = 'powell', bounds = bounds, options =
   options)
671 DeleteUnessesaryFiles(os.path.dirname(inputFile), FileExtensions = ('.out', '.stat',
   '.mvw'))
672 DeleteUnessesaryFiles(os.getcwd(), FileExtensions = ('.bat', ))
673
674
675 WrappedObj.savecahce('FunctionEvaluations.xlsx')
676 WrappedObj.savehistory('OptimizationHistory.xlsx')
677 with open('minimization.pkl', 'wb') as f:
678     pickle.dump(Min, f)
679
680 with open('Optimization Summary.txt', 'w') as f:
681     print('\n\n===== OPTIMIZATION SUMMARY =====', file = f)
682     print(Min, file = f)
683     print(FlutterSummary(inputFile.replace('.fem', '.flt')).FlutterInfo(), file
   = f)
684
685
686 if __name__ == '__main__':
687     main()

```