

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

Βελτιστοποίηση Αεροδυναμικού Σχεδιασμού  
και Δικτυώματος Ιπτάμενου Χειριστή  
Προσανατολισμένη στο Πρόβλημα Ελέγχου

---

Συγγραφείς :

Στέφος Γεώργιος

Ποταμίτης-Κόμης Δ. Ελευθέριος

Επιβλέπων:

Καθηγητής Κωσταντίνος Ι.

Κυριακόπουλος

Μια διατριβή που υποβάλλεται για την εκπλήρωση των απαιτήσεων  
του διπλώματος των Μηχανολόγων Μηχανικών

στο

Εργαστήριο Αυτομάτου Ελέγχου και Ρυθμίσεως Μηχανών  
Τμήμα Μηχανολόγων Μηχανικών



---

NATIONAL TECHNOLOGICAL UNIVERSITY OF ATHENS

DIPLOMA THESIS

---

Control Oriented  
Aerodynamic Design Optimization and  
Truss Optimization  
for an  
Aerial Manipulator

---

*Authors:*

Stefos Georgios

Potamitis-Komis D. Elefterios

*Supervisor:*

Prof. Kostas J. Kyriakopoulos

*A thesis submitted in fulfillment of the requirements  
for the degree of Mechanical Engineer*

*in the*

Control System Laboratory  
Mechanical Engineering Department



## Περίληψη

Τα τελευταία χρόνια παρατηρείται μεγάλο ενδιαφέρον στον τομέα της εναέριας ρομποτικής και συγκεκριμένα στην αλληλεπίδραση με το περιβάλλον μέσω εναέριων χειριστών. Στην παρούσα διπλωματική εργασία αντιμετωπίζεται το πρόβλημα σχεδίασης ενός τέτοιου εναέριου χειριστή. Η σχεδίαση είναι προσανατολισμένη στα προβλήματα ελέγχου που παρουσιάζονται σε μια τέτοια κατασκευή όπως είναι η αεροδυναμική αλληλεπίδραση των ροών που παράγουν οι κινητήρες καθώς και η μη ευαισθησία του συστήματος σε μικρές μεταβολές των μεταβλητών κατάστασης. Ο εναέριος χειριστής θα έχει έναν επενεργητή, όπου ανεξάρτητα των βαθμών ελευθερίας του, ο ιπτάμενος χειριστής θα έχει τη δυνατότητα εφαρμογής δυνάμεων και ροπών προς όλες τις κατευθύνσεις.

Η εργασία αυτή αποτελεί συνέχεια σε μια σειρά εργασιών που έχουν γίνει συμβάλλοντας στην επίτευξη αυτού του στόχου. Αποτελεί κυρίως συνέχεια στην διπλωματική εργασία του Γεωργίου Γαβρίδη [1] στην οποία επιτεύχθηκε ελαχιστοποίηση του όγκου της κατασκευής ικανοποιώντας ταυτόχρονα τους περιορισμούς εύρωστης λειτουργίας.

Στην παρούσα διπλωματική εργασία πήγαμε ένα βήμα πιο πέρα μελετώντας τον υπάρχοντα αλγόριθμο και κάνοντας αλλαγές μειώνοντας κατά πολύ τον υπολογιστικό χρόνο. Έτσι μας δόθηκε η δυνατότητα εισαγωγής επιπλέον περιορισμών και αλλαγής της αντικειμενικής συνάρτησης ώστε να προκύψει μια πιο λειτουργική βέλτιστη λύση. Η αλλαγή της αντικειμενικής συνάρτησης έγινε διότι δεν υπήρχε περιορισμός που να αποτρέπει την απαίτηση οι κινητήρες να παράξουν δυνάμεις μεγαλύτερες από ότι μπορεί να παραχθεί από τα εμπορικά διαθέσιμα προϊόντα.

Στην συνέχεια μελετήθηκε ο σχεδιασμός του δικτύωματος που θα συγκρατεί τα διάφορα στοιχεία του εναέριου ρομποτικού χειριστή. Σε αυτή την ενότητα παρατέθηκαν δυο διαφορετικές μεθοδολογίες. Η πρώτη βελτιστοποιεί το δίκτυωμα μέσω ενός γενετικού αλγορίθμου που επικοινωνεί με το πρόγραμμα ANSYS. Η δεύτερη είναι μια καινοτόμος μεθοδολογία που αναπτύχθηκε στα πλαίσια της εργασίας αυτής για να αντιμετωπίσει το φαινόμενο του λυγισμού.



## Abstract

In the last few years great expansion has been noticed in the field of UAVs and especially towards aerial manipulation. In this thesis we confront the challenge of designing such an aerial manipulator. The design is oriented towards the control problems that arise in the operation of the aerial manipulator such as aerodynamic interaction of the flows produced by the thrusters and the sensitivity of the systems to small changes of the state-variables. This UAV will be equipped with a manipulator (end-effector), regardless of the degrees of freedom of the end-effector, the UAV will be able to exert forces and torques in all direction.

This thesis is a part of the work being done towards the construction of the aerial manipulator. It is meanly a continuation of the George Gavridis [1] thesis in which he accomplished to minimize the volume of the structure while fulfilling the constraints responsible for robust operation.

Here, in this thesis we went a step further by studying the previous algorithm and making changes thus making the algorithm a lot faster. Having done that we could focus on adding more constraints and changing the objective function so as to approach a more functional optimal solution. The change in the objective function was made because there was nothing to ensure that thrusters won't be needed to provide more thrust than it could be produced from the commercially available products.

Next came the design of the truss that would support the different parts of the aerial manipulator. Here, we used two different methodologies. The first one made use of a genetic algorithm from the programming environment Matlab interfacing the software ANSYS to find the optimum truss. The second was an innovative methodology which was produced in the context of this thesis in order to avoid buckling of the limbs of the truss.

## Ευχαριστίες

Πρώτα απ' όλα, θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή κ. Κωνσταντίνο Κυριακόπουλο, που πρώτον μας ανέθεσε την εν λόγω διπλωματική εργασία κάνοντάς μας μέρος της πορείας κατασκευής ενός τόσο καινοτόμου και πρωτότυπου μηχανισμού, και δεύτερον για τη συνεχή καθοδήγησή του καθ' όλη τη διάρκεια της πορείας αυτής. Ακόμη θα θέλαμε να ευχαριστήσουμε τα μέλη του εργαστηρίου και ιδιαίτερα τον Γεώργιο Ζογόπουλο. Επίσης θα θέλαμε να ευχαριστήσουμε το διευθυντή του εργαστηρίου προσωπικών υπολογιστών κ. Πετρόπουλο και τα υπόλοιπα μέλη της διοίκησης τον κ. Παύλου, την κ. Αγγέλη, τον κ. Βρανάκη και τον κ. Τελλάκη για τη διάθεση των ηλεκτρονικών υπολογιστών για τις ανάγκες της διπλωματικής εργασίας. Τέλος ευχαριστούμε τα μέλη του Κέντρου Ηλεκτρονικών Υπολογιστών του ΕΜΠ για τη βοήθεια τους, δίνοντας μας πρόσβαση στο σύστημα High Performance Computing (HPC) του ιδρύματος ώστε να εκτελέσουμε τις βελτιστοποιήσεις σε παράλληλο περιβάλλον υψηλής απόδοσης.



## Acknowledgements

First and foremost, we would like to express our deepest gratitude to our professor Mr. Konstantinos Kyriakopoulos, who confided us and made us a part in the construction of an innovative project such as this, as well as for his guidance throughout the course of this thesis. Secondly we would like to thank all the members of the laboratory and especially Mr. George Zogopoulos for his technical advices. In addition, we would like to thank Mr. Petropoulos and especially the staff of the Personal Computer Laboratory Mr. Pavlou, Mrs. Angeli, Mr. Vranakis and Mr. Tellakis. Also, we would like to thank the staff of Computer Center of NTUA, for giving us access to the High Performance Computing system of the institute that allowed us to perform optimizations on a parallel environment with multiple threads.

.

# Περιεχόμενα

<b>Περίληψη.....</b>	<b>5</b>
<b>Abstract .....</b>	<b>7</b>
<b>Ευχαριστίες .....</b>	<b>8</b>
<b>Acknowledgements .....</b>	<b>9</b>
<b>Περιεχόμενα .....</b>	<b>10</b>
<b>Λίστα Σχημάτων .....</b>	<b>12</b>
<b>Λίστα Πινάκων.....</b>	<b>13</b>
<b>Κεφάλαιο 1<sup>ο</sup>.....</b>	<b>14</b>
1.1 Εισαγωγή.....	14
<b>Κεφάλαιο 2<sup>ο</sup>.....</b>	<b>16</b>
2.1 Γενικά.....	16
2.2 Περιγραφή του προβλήματος.....	16
2.3 Βασικές Αρχές του Προβλήματος.....	17
2.3.1 Δυνάμεις.....	17
2.3.2 Ροπές .....	18
2.3.3 End-Effector.....	19
2.4 Αρνητική Δύναμη Ώσης .....	21
2.5 Αεροδυναμική Αλληλεπίδραση .....	26
2.6 Πρόβλημα Σχεδιασμού .....	31
2.6.1 Η Προηγούμενη Βελτιστοποίηση .....	32
2.6.2 Η Νέα βελτιστοποίηση .....	34
2.6.3 Εύρεση Καλύτερης Λύσης.....	38
2.7 Παραμετροποίηση .....	40
<b>43</b>	
<b>Κεφάλαιο 3<sup>ο</sup>.....</b>	<b>45</b>
3.2 Εισαγωγή στην Ανάλυση Δικτυωμάτων .....	45

3.3	Ορισμός του Προβλήματος.....	47
3.4	Επίλυση του Προβλήματος.....	50
3.4.1	Γενετικός Αλγόριθμος .....	50
3.4.2	ANSYS Apdl .....	53
3.5	Παραγωγή δικτύματος προσανατολισμένη στην αποτροπή λυγισμού.....	57
3.6	Επίλογος.....	60
<b>Κεφάλαιο 4<sup>ο</sup></b>	<b>.....</b>	<b>61</b>
4.1	Εισαγωγή.....	61
4.2	Αποτελέσματα Πρώτης Βελτιστοποίησης.....	61
4.2.1	Βέλτιστες Λύσεις .....	64
4.3	Αμφίρροπες προπέλες.....	102
4.3.1	Υπολογισμός ώσης για αμφίρροπες προπέλες.....	102
4.4	Επίλογος.....	113
<b>Κεφάλαιο 5<sup>ο</sup></b>	<b>.....</b>	<b>115</b>
5.1	Σύνοψη .....	115
5.2	Προτάσεις για μελλοντική εργασία.....	116
<b>Παραρτήματα.....</b>	<b>.....</b>	<b>118</b>
<b>Παράρτημα Α : Αλγόριθμοι Βελτιστοποίησης .....</b>	<b>.....</b>	<b>118</b>
<b>Παράρτημα Β.1 : Ανάλυση του Κώδικα APDL .....</b>	<b>.....</b>	<b>132</b>
<b>Παράρτημα Β.2 :.....</b>	<b>.....</b>	<b>138</b>
<b>Αλγόριθμος Δικτύματος για Αποφυγή Λυγισμού .....</b>	<b>.....</b>	<b>138</b>
<b>Βιβλιογραφία.....</b>	<b>.....</b>	<b>144</b>

## Λίστα Σχημάτων

<b>Σχήμα 2.4.1</b> Ισοδύναμο δύναμης ώσης	25
<b>Σχήμα 2.5.1</b> Πολυωνυμικό προφίλ αεροδυναμικής ροής.	27
<b>Σχήμα 2.5.2</b> Γραφική αναπαράσταση του αεροδυναμικού κώνου στο περιβάλλον προγραμματιστικό Matlab.	28
<b>Σχήμα 2.5.3</b> Πλάγια και προοπτική όψη του αποτελέσματος της βελτιστοποίησης των σφαιρών του κώνου για μήκος κώνου 0,9m και προπέλα 10 ιντσών.	30
<b>Σχήμα 2.6.1</b> Γεωμετρική απεικόνιση της γωνίας $\theta$	37
<b>Σχήμα 2.7.1</b> Διάγραμμα Ροής	43
<b>Σχήμα 2.7.2</b> Διάγραμμα κλήσεων συναρτήσεων	44
<b>Σχήμα 3.3.1</b> Προσεγγιστική απεικόνιση του end-effector, με τους απλοποιημένους κόμβους	48
<b>Σχήμα 3.4.1</b> Ιδιότητες στοιχείου PIPE288 από manual APDL	56
<b>Σχήμα 3.5.1</b> Απεικόνιση του διανύσματος δύναμης πάνω στον κόμβο	58
<b>Σχήμα 4.2.1:</b> Συγκριτική αναπαράσταση των λύσεων (15 και 5) με το μεγαλύτερο και με το μικρότερο όγκο αντίστοιχα (ανάμεσά τους φαίνεται με σφαίρα με διαστάσεις αυτές μιας μπάλας μπάσκετ). Επίσης σημειώνονται οι μεταβλητές των διαστάσεων των λύσεων (X,Y,Z).	63
<b>Σχήμα 4.4.1</b> Η προτεινόμενη λύση, με αεροδυναμικούς κώνους, τα μπλε διανύσματα δείχνουν την κατεύθυνση της δύναμης	113
<b>Σχήμα 4.4.2</b> Η προτεινόμενη λύση, με το δικτύωμα και τις προπέλες	114

## Λίστα Πινάκων

<i>Πίνακας 2.6.1: Η δομή του της παλαιάς βελτιστοποίησης.</i>	33
<i>Πίνακας 2.6.2: Η δομή της νέας βελτιστοποίησης.</i>	35
<i>Πίνακας 3.4.1: Συνδιασμοί για <math>n=7</math> κινητήρες</i>	55
<i>Πίνακας 3.5.1 Παράδειγμα αρίθμησης κόμβων</i>	59
<i>Πίνακας 4.2.1 Περιπτώσεις τρεξιμάτων</i>	62
<i>Πίνακας 4.2.2: Βέλτιστες λύσεις ως προς το συντελεστή ώσης υπολογισμένα με μέθοδο αναδιανομής ώσης.</i>	62
<i>Πίνακας 4.2.3: Βέλτιστες λύσεις ως προς το συντελεστή ώσης ευρισκόμενο μέσω της μεθόδου ελαχίστων τετραγώνων με μη αρνητικές μεταβλητές.</i>	64
<i>Πίνακας 4.3.1 Συγκριτική απεικόνιση των αποτελεσμάτων για αμφίρροπες προπέλες</i>	103
<i>Πίνακας 5.1.1 Πίνακας περιπτώσεων τρεξιμάτων</i>	115
<i>Πίνακας 5.1.2 Πίνακας περιπτώσεων τρεξιμάτων για αμφίρροπες προπέλες</i>	116

# Κεφάλαιο 1<sup>ο</sup>

## 1.1 Εισαγωγή

Τον τελευταίο καιρό τα UAVs ή Drones έχουν μπει στη καθημερινότητα μας με τον έναν ή με τον άλλο τρόπο. Χρησιμοποιούνται είτε για ψυχαγωγικούς σκοπούς όντας ένα απλό τηλεκατευθυνόμενο, είτε για επαγγελματικούς σκοπούς, σαν μέσω λήψης εναέριων φωτογραφιών ή βίντεο που θα ήταν πολύ δύσκολο τραβηχτούν αλλιώς, είτε για στρατιωτικούς- κατασκοπικούς σκοπούς, είτε για ερευνητικούς σκοπούς κυρίως στον τομέα της ρομποτικής.

Ένα μεγάλο κομμάτι των UAVs είναι τα λεγόμενα multi-copters, τα οποία σύμφωνα με την ονομασία τους είναι ελικοπτεράκια που αποτελούνται από ποικίλο αριθμό κινητήρων (quad-copter, hexa-copter, octo-copter). Η μέχρι τώρα συνηθέστερη διάταξη των ελίκων πάνω στο multi-copter είναι η επίπεδη, δηλαδή όλες οι έλικες βρίσκονται πάνω σε ένα επίπεδο και περιστρέφονται είτε δεξιόστροφα είτε αριστερόστροφα. Επίσης μέχρι τώρα όλα τα multi-copter που υπήρχαν, είχαν συγκεκριμένο προσανατολισμό πτήσης, δηλαδή μπορούσαν να αιωρούνται έχοντας μόνο ένα συγκεκριμένο προσανατολισμό στον χώρο.

Η εργασία Control Oriented Aerodynamic Design Optimization for an Aerial Manipulator [1] πρότεινε για πρώτη φορά ένα νέο μοντέλο multi-copter το οποίο θα μπορούσε να έχει 6 βαθμούς ελευθερίας και κατ' επέκταση να μπορεί να αιωρείται ως προς οποιαδήποτε κατεύθυνση, να μην έχει δηλαδή προσανατολισμό αιώρησης. Η παρούσα διπλωματική αποτελεί συνέχεια της προαναφερθείσας εργασίας, καθώς βασισμένοι σε αυτό το σκεπτικό θα αναλύσουμε περαιτέρω κάποιες παραμέτρους οι οποίες στο αρχικό στάδιο δεν είχαν ληφθεί υπόψη.

Αυτό που θέλουμε να πετύχουμε στην παρούσα διπλωματική εργασία είναι η μοντελοποίηση και ο σχεδιασμός ενός multi-copter το οποίο θα είναι προσανατολισμένο ώστε να λειτουργήσει ως Εναέριος Χειριστής (Aerial Manipulator). Για να επιτευχθεί αυτό εκτός από την χωρική βελτιστοποίηση με βάση την μαθηματική μοντελοποίηση που έγινε στην [1], θα πρέπει να ληφθούν υπόψη

παράγοντες οι οποίοι σχετίζονται με την υλοποιησιμότητα του θεωρητικού μοντέλου. Οι παράγοντες που δεν λήφθηκαν υπόψη είναι το βάρος της κατασκευής σε συνδυασμό με την περιορισμένη η δυνατότητα παραγωγής της απαιτούμενης ώσης από τους κινητήρες.

Επομένως έχοντας αυτά τα δεδομένα η νέα βελτιστοποίηση που έλαβε χώρα, ήταν μια χωρική βελτιστοποίηση ,δηλαδή ως μεταβλητές σχεδιασμού ήταν οι θέσεις και οι προσανατολισμοί των κινητήρων καθώς και η θέση του κουτιού που θα περιέχει τα ηλεκτρονικά. Η αντικειμενική συνάρτηση της βελτιστοποίησης , δηλαδή το κριτήριο του σχεδιασμού ήταν η μέγιστη απαιτούμενη ώση που χρειάζεται ώστε να μπορεί ο Aerial Manipulator να εκτελέσει αιώρηση. Ο παράγοντας του μεγέθους της κατασκευής δε μπήκε άμεσα στην αντικειμενική συνάρτηση , ο τρόπος με τον οποίο έγινε αυτό θα αναλυθεί εκτενέστερα στο κεφάλαιο 2.

## Κεφάλαιο 2<sup>ο</sup>

### Χωρικός-Αεροδυναμικός Σχεδιασμός:

Ορισμός Προβλήματος, Μοντελοποίηση,  
Βελτιστοποίηση.

#### 2.1 Γενικά

Στο παρόν κεφάλαιο θα περιγραφεί η δομή του προβλήματος, θα αναπτυχθούν οι εξισώσεις που το διέπουν, καθώς και οι περιορισμοί στους οποίους υπόκειται το πρόβλημα. Επίσης θα παρατεθεί το πρόβλημα βελτιστοποίησης που λύθηκε στο [1] καθώς και το νέο βελτιωμένο πρόβλημα βελτιστοποίησης.

#### 2.2 Περιγραφή του προβλήματος

Το πρόβλημα που μελετάμε είναι ο σχεδιασμός και η μοντελοποίηση ενός εναέριου χειριστή (Aerial Manipulator), στον οποίο από εδώ και στο εξής θα αναφερόμαστε ως AM. Ο AM θα αποτελείται από  $n$  αριθμό κινητήρων (thrusters) και έναν επενεργητή (end-effector), όπου  $n$  ο ακέραιος αριθμός των κινητήρων που θα καθοριστεί στη συνέχεια. Το χαρακτηριστικό αυτού του εναέριου οχήματος είναι ότι θα έχει 6 βαθμούς ελευθερίας, δηλαδή θα μπορεί να ασκεί δυνάμεις και ροπές προς οποιαδήποτε κατεύθυνση στον τρισδιάστατο χώρο. Αυτές οι δυνάμεις και οι ροπές θα εφαρμόζονται στο περιβάλλον του AM μέσω του end-effector. Συνεπάγεται, λοιπόν, ότι το AM δεν θα έχει συγκεκριμένο προσανατολισμό αιώρησης, αλλά θα μπορεί να αιωρείται προς οποιαδήποτε κατεύθυνση, καθώς και να εκτελεί κινήσεις στο χώρο προς οποιαδήποτε κατεύθυνση και οποιονδήποτε προσανατολισμό. Για να γίνει αυτό χρειάζεται να διαθέτει έναν αριθμό κινητήρων ικανό ώστε να έχει 6 βαθμούς ελευθερίας, άρα  $n \geq 6$ .



## 2.3 Βασικές Αρχές του Προβλήματος

Ξεκινώντας, ας ορίσουμε τα διανύσματα  $\mathbf{r}_i \in \mathbf{R}^3$  με  $i=1, \dots, n$  που αντιπροσωπεύουν τις θέσεις των κινητήρων σε σχέση με τον end-effector (ορίζουμε δηλαδή στο σύστημα συντεταγμένων που χρησιμοποιούμε ως αρχή των αξόνων την θέση του end-effector). Ορίζουμε επίσης και τα διανύσματα  $\hat{\mathbf{F}}_i \in \mathbf{R}^3$  με  $i=1, \dots, n$  που δείχνουν τον προσανατολισμό του κάθε κινητήρα στο συγκεκριμένο σύστημα συντεταγμένων. Λαμβάνοντας υπόψη τα παραπάνω εισάγουμε την έννοια του ‘συντελεστή ώσης’ και τον συμβολίζουμε με  $\lambda_i$ . Έτσι η δύναμη που παράγει κάθε κινητήρας ισούται με  $\lambda_i \cdot \hat{\mathbf{F}}_i$ . Ουσιαστικά το  $\lambda$  είναι το μέτρο και το  $\mathbf{F}$  το μοναδιαίο διάνυσμα κατεύθυνσης της δύναμης.

### 2.3.1 Δυνάμεις

Οι δυνάμεις που ασκούνται στο AM είναι δύο ειδών:

- Οι δυνάμεις που παράγονται από τους κινητήρες  $\lambda_i \cdot \hat{\mathbf{F}}_i$ .
- Και οι δυνάμεις που ασκούνται από το βαρυτικό πεδίο στη κατασκευή.

Επομένως οι συνολικές δυνάμεις που ασκούνται στο AM είναι:

$$\sum_{i=1}^n (\lambda_i \cdot \hat{\mathbf{F}}_i) + \hat{\mathbf{W}} = \hat{\mathbf{F}}_{act} \quad (2.1)$$

Όπου  $\hat{\mathbf{F}}_{act} \in \mathbf{R}^3$  είναι το διάνυσμα της δύναμης που ασκείται στον end-effector από την αλληλεπίδρασή του με το περιβάλλον και  $\hat{\mathbf{W}} \in \mathbf{R}^3$  είναι το διάνυσμα του βάρους του AM.

Το διάνυσμα του βάρους αποτελείται από το βάρος της κατασκευής και των επιμέρους εξαρτημάτων το οποίο ορίζουμε ως  $w_s$ , και το βάρος του κάθε κινητήρα που συμβολίζουμε με  $w$ .

Τελικά το ολικό βάρος του AM γράφεται ως:

$$\mathbf{W} = n \cdot w + w_s \quad (2.2)$$

Με απλή αντικατάσταση τις σχέσης (2.1) στην (2.2) προκύπτει:

$$\sum_{i=1}^n (\lambda_i \cdot \hat{F}_i) + n \cdot w + w_s = \hat{F}_{act} \quad (2.3)$$

Ο συντελεστής ώσης κάθε κινητήρα μπορεί να γραφεί στη μορφή:

$$\lambda = [\lambda_1 \lambda_2 \dots \lambda_n]^T$$

Και τα διανύσματα προσανατολισμού μπορούν να έχουν αυτή την μορφή:

$$\hat{F} \triangleq [\hat{F}_1 \hat{F}_2 \dots \hat{F}_n] \in R^{3 \times n}$$

Έτσι η 2.3 μπορεί να γραφεί:

$$\hat{F} \cdot \lambda + n \cdot w + w_s = F_{act} \quad (2.4)$$

Όπου  $F_{act} \in R^3$ .

### 2.3.2 Ροπές

Οι ροπές όπως και οι δυνάμεις χωρίζονται σε δυο τύπους:

- Τις ροπές που παράγονται από την εφαρμογή μιας δύναμης με κατεύθυνση που δεν τέμνει τον άξονα περιστροφής.
- Και τις ροπές που δημιουργούνται από την λεγόμενη ροπή αντίδρασης

Στον πρώτο τύπο ανήκουν οι ροπές που παράγονται από τις δυνάμεις που ασκούν οι κινητήρες και οι βαρυτικές δυνάμεις που ασκούνται στις διάφορες μάζες της κατασκευής (κινητήρες, ηλεκτρονικά, δικτύωμα, end-effector).

- Οι ροπές που αφορούν την δύναμη ώσης γράφονται:

$$r_i \times (\lambda_i \cdot \hat{F}_i)$$

- Οι ροπές που παράγονται από το βάρος της κατασκευής, μπορούν να εκφραστούν ως το εξωτερικό γινόμενο του κέντρου βάρους του Aerial Manipulator και του συνολικού του βάρους:

$$R_G \times W = -W \times R_G \triangleq -\left(\sum_{i=1}^n w \times r_i\right) - w_s \times r_s$$

Όσον αφορά τον δεύτερο τύπο ροπών, ας σκεφτούμε ότι ο κινητήρας γυρίζει ως προς μία κατεύθυνση, η κατασκευή (Aerial Manipulator) θα τείνει να περιστραφεί προς την αντίθετη κατεύθυνση σύμφωνα με τον τρίτο νόμο του Νεύτωνα. Αυτή είναι γνωστή ως ροπή αντίδρασης (reaction torque). Αυτή τη ροπή θα την συμβολίζουμε με  $\tau_i \in R^3$  η οποία είναι συγγραμμική με το διάνυσμα προσανατολισμού  $\hat{F}_i$  κάθε κινητήρα του συστήματος. Επομένως μπορούμε να συμπαιράνουμε ότι:

$$\tau_i = \mu \cdot (\lambda \cdot \hat{F}_i)$$

Όπου  $\mu$  είναι ένας συντελεστής που αντιπροσωπεύει τη σχέση μεταξύ του μέτρου της δύναμης που ασκεί ο κινητήρας και της ροπής αντίδρασης.

Έτσι η συνολική ροπή που ασκείται στον end-effector είναι:

$$\sum_{i=1}^n [r_i \times (\lambda_i \cdot \hat{F}_i) + \mu \cdot (\lambda_i \cdot \hat{F}_i)] + R_G \times W = M_{act} \quad (2.5)$$

Η (2.5) μπορεί να γραφεί και ως:

$$\sum_{i=1}^n [r_i \times (\lambda_i \cdot \hat{F}_i) + \mu \cdot (\lambda_i \cdot \hat{F}_i)] + \left( \sum_{i=1}^n r_i \right) \times w + r_s \times w_s = M_{act}$$

Όπου  $M_{act}$  είναι η ροπή που ασκεί ο end-effector στο περιβάλλον.

### 2.3.3 End-Effector

Σε αυτό το σημείο είναι βασικό να θυμίσουμε ότι το εξωτερικό γινόμενο δύο διανυσμάτων  $A$  και  $B$  μπορεί να γραφεί ως ο πολλαπλασιασμός ενός πίνακα  $3 \times 3$  και ενός διανύσματος ως ακολούθως:

$$A \times B = S(A) \cdot B = -S(B) \cdot A$$

Όπου  $S$  είναι ο skew-symmetric matrix, που είναι αντί-συμμετρικός και μη αναστρέψιμος πίνακας. Για ένα τυχαίο διάνυσμα  $A = [A_x \ A_y \ A_z]^T$  ο  $S$  είναι:

$$S(A) = \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix}$$

Επομένως το εξωτερικό γινόμενο της θέσης κάθε κινητήρα ( $r_i$ ) με την δύναμη ώσης ( $\lambda \cdot \hat{F}_i$ ) είναι :

$$r_i \times (\lambda_i \cdot \hat{F}_i) = \lambda_i \cdot (r_i \times \hat{F}_i) = \lambda \cdot S(r_i) \cdot \hat{F}_i \quad (2.6)$$

Με αντικατάσταση της (2.6) στην (2.5) έχουμε:

$$\sum_{i=1}^n [\lambda_i \cdot S(r_i) \cdot \hat{F}_i + \mu \cdot (\lambda_i \cdot \hat{F}_i)] + \left( \sum_{i=1}^n r_i \right) \times w + r_s \times w_s = M_{act}$$

Και τελικά παίρνουμε την εξίσωση:

$$\sum_{i=1}^n [\lambda_i \cdot S(r_i) \cdot \hat{F}_i] + \mu \cdot \sum_{i=1}^n (\lambda_i \cdot \hat{F}_i) + (\sum_{i=1}^n r_i) \times w + r_s \times w_s = M_{act} \quad (2.7)$$

Συνδυάζοντας τις εξισώσεις (2.4) και (2.7) έχουμε :

$$\sum_{i=1}^n [\lambda_i \cdot S(r_i) \cdot \hat{F}_i] + \mu \cdot (F_{act} - n \cdot w - w_s) + (\sum_{i=1}^n r_i) \times w + r_s \times w_s = M_{act}$$

Έχει ήδη σημειωθεί ότι

$$\lambda = [\lambda_1 \lambda_2 \dots \lambda_n]^T$$

Έτσι παράγεται η εξίσωση :

$$E(r, \hat{F}) \cdot \lambda = M_{act} - \mu \cdot (F_{act} - n \cdot w - w_s) - (\sum_{i=1}^n r_i) \times w - r_s \times w_s \quad (2.8)$$

Όπου

$$r \triangleq [r_1 \ r_2 \ \dots \ r_n]^T \in R^{3 \times n}$$

$$\hat{F} \triangleq [\hat{F}_1 \ \hat{F}_2 \ \dots \ \hat{F}_n]^T \in R^{3 \times n}$$

$$E(r, \hat{F}) \triangleq [S(r_1) \cdot \hat{F}_1 \ S(r_2) \cdot \hat{F}_2 \ \dots \ S(r_n) \cdot \hat{F}_n]^T \in R^{3 \times n}$$

Ας υποθέσουμε ότι ο αριθμός των κινητήρων  $n$  και οι θέσεις τους ( $r_i$ ) είναι γνωστές με  $i = 1, \dots, n$  οι εξισώσεις (2.4) και (2.7) ξαναγράφονται ως:

$$\left. \begin{aligned} \hat{F} \cdot \lambda + n \cdot w + w_s &= F_{act} \\ E(r, \hat{F}) \cdot \lambda &= M_{act} - \mu \cdot (F_{act} - n \cdot w - w_s) - (\sum_{i=1}^n r_i) \times w - r_s \times w_s \end{aligned} \right\} \quad (2.9)$$

Το σύστημα (2.9) γράφεται σε μητρική μορφή:

$$D(r, \hat{F}) \cdot \lambda = W_R \quad (2.10)$$

Όπου

$$W_R \triangleq \begin{bmatrix} F_{act} - n \cdot w - w_s \\ M_{act} - \mu \cdot (F_{act} - n \cdot w - w_s) - (\sum_{i=1}^n r_i) \times w - r_s \times w_s \end{bmatrix} \in R^6 \quad (2.11)$$

Που είναι ένα διάνυσμα που αντιπροσωπεύει τις δυνάμεις και τις ροπές που ασκούνται στον end-effector λόγω της βαρύτητας και λόγω των εξωτερικών δυνάμεων.

Και

$$D(r, \hat{F}) \triangleq \begin{bmatrix} \hat{F} \\ E(r, \hat{F}) \end{bmatrix} \in R^{6 \times n} \quad (2.12)$$

Ο οποίος θα λέγεται πίνακας μετασχηματισμού από το χώρο του κινητήρα στο χώρο του end-effector.

## 2.4 Αρνητική Δύναμη Ώσης

Είναι ξεκάθαρο ότι λύνοντας την εξίσωση (2.10), ο πίνακας που αντιπροσωπεύει τον συντελεστή ώσης  $\lambda$  μπορεί να πάρει οποιαδήποτε τιμή στο  $R^6$ . Ωστόσο οι προπέλλες των κινητήρων είναι βέλτιστα σχεδιασμένες για να περιστρέφονται ως προς μια συγκεκριμένη φορά, και επομένως η δύναμη ώσης θα έχει συγκεκριμένη κατεύθυνση, αυτή που ορίζεται από το διάνυσμα προσανατολισμού κάθε κινητήρα, για να ισχύει αυτό είναι επιτρεπτές μόνο θετικές τιμές του  $\lambda_i$ . Με σκοπό να αρθεί το πρόβλημα ύπαρξης αρνητικών συντελεστών  $\lambda_i$ , υιοθετήθηκε μία καινοτόμος μέθοδος που βασίζεται στην εισαγωγή ενός επιπλέον κινητήρα.

Ξεκινάμε γράφοντας την (2.10) στην μορφή:

$$\sum_{i=1}^n \lambda_i \cdot t_i = W_R \quad (2.13)$$

Όπου

$$D(r, \hat{F}) = [t_1 \dots t_n] \quad (2.14)$$

$$t_i = \begin{bmatrix} \hat{F}_i \\ S(r_i)\hat{F}_i \end{bmatrix} \in R^6 \quad (2.15)$$

Για όλα τα  $i = 1, \dots, n$  και εισάγουμε το διάνυσμα

$$t_a = -\sum_{i=1}^n t_i = -\sum_{i=1}^n \begin{bmatrix} \hat{F}_i \\ S(r_i)\hat{F}_i \end{bmatrix} = \begin{bmatrix} \hat{F}_a \\ S(r_a)\hat{F}_a \end{bmatrix} \quad (2.16)$$

Που αντιστοιχεί στον επιπλέον κινητήρα. Από την εξίσωση (2.16) το διάνυσμα θέσης  $r_a$  και κατεύθυνσης  $\hat{F}_a$  του καινούργιου κινητήρα πρέπει να ικανοποιούν τους εξής περιορισμούς:

$$\hat{F}_a = -\sum_{i=1}^n \hat{F}_i \quad (2.17)$$

$$S(\hat{F}_a) \cdot r_a = -\sum_{i=1}^n \{S(\hat{F}_i) \cdot r_i\} \quad (2.18)$$

Αν υποθέσουμε ότι λύνοντας την (2.12) ως προς  $\lambda$  προκύπτουν αρνητικές και θετικές τιμές, τότε ορίζουμε το σύνολο

$$\sigma_N = \{k: \lambda_k < 0, k = 1, \dots, n\}$$

που περιέχει όλες τις αρνητικές δυνάμεις ώσης, και το σύνολο

$$\sigma_P = \{1, 2, \dots, n\} - \sigma_N$$

Που περιέχει όλες τις θετικές δυνάμεις ώσης. Σημειώνοντας ότι

$$\lambda_k < 0 \Leftrightarrow (-\lambda_k) > 0 \quad \forall k \in \sigma_N$$

Η εξίσωση (2.13) χωρίζεται σε

$$\sum_{i \in \sigma_P} \lambda_i t_i + \sum_{k \in \sigma_N} \lambda_k t_k = W_R$$

$$\Leftrightarrow \sum_{i \in \sigma_P} \lambda_i t_i + \sum_{k \in \sigma_N} (-\lambda_k)(-t_k) = W_R \quad (2.19)$$

Τώρα από την (2.16) μπορούμε να εξάγουμε ότι:

$$\begin{aligned} t_a &= -\sum_{i=1}^n t_i = -\sum_{i \in \sigma_P} t_i - \sum_{j \in \sigma_N} t_j \\ \Leftrightarrow -\sum_{j \in \sigma_N} t_j &= t_a + \sum_{i \in \sigma_P} t_i \end{aligned} \quad (2.20)$$

Είναι προφανές ότι

$$-\sum_{j \in \sigma_N} t_j = -t_k - \sum_{\substack{j \in \sigma_N \\ j \neq k}} t_j, \forall k \in \sigma_N \quad (2.21)$$

Συνδυάζοντας τις (2.21) και (2.19) έχουμε

$$-t_k = t_a + \sum_{i \in \sigma_P} t_i + \sum_{\substack{j \in \sigma_N \\ j \neq k}} t_j, \forall k \in \sigma_N \quad (2.22)$$

Αντικαθιστώντας την (2.22) στην (2.19)

$$\begin{aligned} \sum_{i \in \sigma_P} \lambda_i t_i + \sum_{k \in \sigma_N} (-\lambda_k) \left[ t_a + \sum_{i \in \sigma_P} t_i + \sum_{\substack{j \in \sigma_N \\ j \neq k}} t_j \right] &= W_R \\ \sum_{i \in \sigma_P} \lambda_i t_i + \sum_{k \in \sigma_N} (-\lambda_k) t_a + \sum_{k \in \sigma_N} (-\lambda_k) \sum_{i \in \sigma_P} t_i + \sum_{k \in \sigma_N} \left[ (-\lambda_k) \sum_{\substack{j \in \sigma_N \\ j \neq k}} t_j \right] &= W_R \end{aligned} \quad (2.23)$$

Είναι γνωστό από μαθηματικές ιδιότητες ότι

$$\sum_{k \in \sigma_N} (-\lambda_k) \sum_{i \in \sigma_P} t_i = \sum_{i \in \sigma_P} \left[ \sum_{k \in \sigma_N} (-\lambda_k) \right] t_i \quad (2.24)$$

$$\sum_{k \in \sigma_N} \left[ (-\lambda_k) \sum_{\substack{j \in \sigma_N \\ j \neq k}} t_j \right] = \sum_{k \in \sigma_N} \left[ \sum_{\substack{j \in \sigma_N \\ j \neq k}} (-\lambda_k) \right] t_k \quad (2.25)$$

Έτσι η (2.23) συνδυάζοντας τις (2.24) και (2.25) μπορεί να γραφεί ως εξής

$$\sum_{i \in \sigma_P} \left[ \lambda_i + \sum_{k \in \sigma_N} (-\lambda_k) \right] t_i + \sum_{k \in \sigma_N} \left[ \sum_{\substack{j \in \sigma_N \\ j \neq k}} (-\lambda_j) \right] t_k + \left[ \sum_{k \in \sigma_N} (-\lambda_i) \right] t_a = W_R \quad (2.26)$$

Θέτοντας

$$\Delta = \sum_{k \in \sigma_N} (-\lambda_k) > 0$$

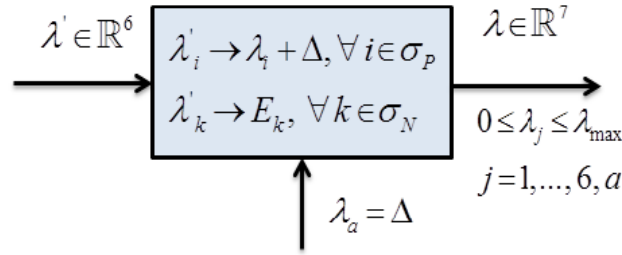
$$E_k = \sum_{\substack{j \in \sigma_N \\ j \neq k}} (-\lambda_j) > 0$$

Και αντικαθιστώντας στην (2.26) τελικά βγαίνει

$$\sum_{i \in \sigma_P} (\lambda_i + \Delta) t_i + \sum_{k \in \sigma_N} E_k t_k + \Delta \cdot t_a = W_R \quad (2.27)$$

Η (2.27) παρέχει την αναδιανομή ώσης σε όλους τους κινητήρες αφού προσθέσουμε τον καινούριο κινητήρα, έτσι ώστε να είναι όλα τα  $\lambda$  θετικά. Η παραπάνω εξίσωση μπορεί να φανεί καλύτερα στο Σχήμα 2.4.1 στο οποίο απεικονίζεται ο αλγόριθμος αναδιανομής της ώσης. Οι μεταβλητές  $\lambda', \lambda'_i, \lambda'_k$  υποδεικνύουν τις αρχικές δυνάμεις ώσης, και οι  $\lambda_j$  τις δυνάμεις ώσης μετά την αναδιανομή, συν την επιπρόσθετη δύναμη ώσης  $\lambda_a$ . Το  $\lambda'_i, \forall i \in \sigma_P$  υποδεικνύει τις αρχικά θετικές δυνάμεις ώσης και το  $\lambda'_k, \forall k \in \sigma_N$  υποδεικνύει τις αρχικά αρνητικές δυνάμεις ώσης. Αυτό σημαίνει ότι κάθε στιγμή, έξι δυνάμεις ώσης (όχι απαραίτητα όλες θετικές) είναι ισοδύναμες με επτά άλλες δυνάμεις ώσης, όλες θετικές.





**Σχήμα 2.4.1** Ισοδύναμο δύναμης ώσης

Την μέθοδο που αναλύσαμε παραπάνω θα την καλούμε στη παρούσα διπλωματική, *Μέθοδο Αναδιανομής Ώσης*. Για να γίνει καλύτερα αντιληπτός ο αλγόριθμος αναδιανομής των δυνάμεων, θα παρατεθεί ένα παράδειγμα. Ας υποθέσουμε ότι σε ένα τυχαίο χρόνο  $t_s \geq 0$ , προκύπτει από την (2.12) ότι το σύστημα χρειάζεται τις εξής δυνάμεις ώσης:

$$\begin{bmatrix} \lambda'_1(t_s) \\ \lambda'_2(t_s) \\ \lambda'_3(t_s) \\ \lambda'_4(t_s) \\ \lambda'_5(t_s) \\ \lambda'_6(t_s) \end{bmatrix} = \begin{bmatrix} -8 \\ 7 \\ -5 \\ 3 \\ -4 \\ 5 \end{bmatrix} [Newton]$$

Τα σύνολα  $\sigma_N, \sigma_P$  θα είναι  $\sigma_P = \{2,4,6\}, \sigma_N = \{1,3,5\}$ . Για το  $\Delta$  έχουμε  $\Delta = -(-8) - (-5) - (-4) = 17$ . Οπότε μετά τον αλγόριθμο αναδιανομής οι νέες δυνάμεις ώσης θα είναι:

$$\begin{bmatrix} \lambda'_1(t_s) \\ \lambda'_2(t_s) \\ \lambda'_3(t_s) \\ \lambda'_4(t_s) \\ \lambda'_5(t_s) \\ \lambda'_6(t_s) \\ \lambda'_\alpha(t_s) \end{bmatrix} = \begin{bmatrix} 9 \\ 24 \\ 12 \\ 20 \\ 13 \\ 22 \\ 17 \end{bmatrix} [Newton]$$

Στο παραπάνω παράδειγμα μιλήσαμε για  $n=6$  κινητήρες +1 επιπρόσθετο. Αυτό γιατί οι έξι κινητήρες είναι οι ελάχιστοι που μπορούν να μας δώσουν έξι βαθμούς ελευθερίας. Τα ίδια ακριβώς ισχύουν και για παραπάνω κινητήρες παραδείγματος χάρη 7+1, 8+1 κ.λ.π.

Κλείνοντας το θέμα των αρνητικών δυνάμεων ώσης ο επιπρόσθετος κινητήρας μετατρέπει την (2.11) σε :

$$W_R \triangleq \begin{bmatrix} F_{act} - (n+1) \cdot w - w_s \\ M_{act} - \mu \cdot (F_{act} - (n+1) \cdot w - w_s) - \left( \sum_{i=1}^{n+1} r_i \right) \times w - r_s \times w_s \end{bmatrix} \in R^6$$

όπου στον αριθμό των  $n$  κινητήρων προσθέτουμε τον επιπρόσθετο κινητήρα αλλάζοντας έτσι το βάρος του συστήματος.

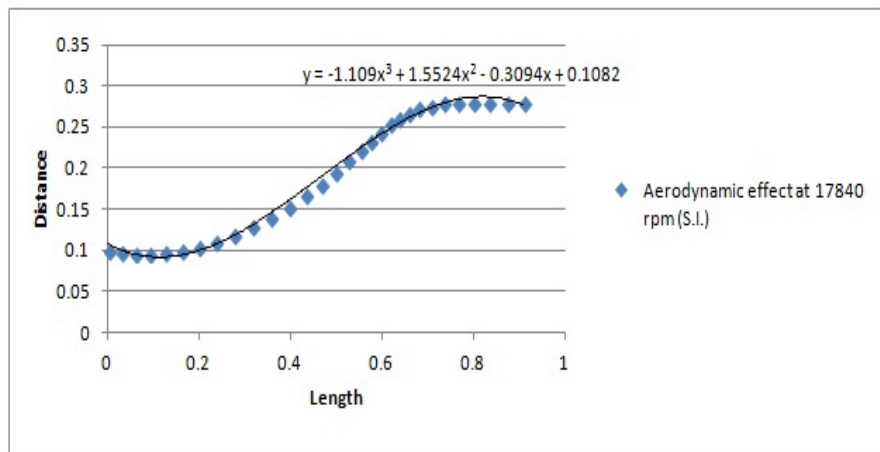
Συμπερασματικά ορίσαμε τις μεταβλητές του προβλήματος που είναι οι θέσεις και οι προσανατολισμοί των κινητήρων, διατυπώσαμε τη σχέση δυνάμεων και ροπών από το χώρο του κινητήρα στο χώρο του end-effector, και εν συνεχεία αντιμετωπίσαμε την εμφάνιση αρνητικών συντελεστών ώσης.

## 2.5 Αεροδυναμική Αλληλεπίδραση

Στην παρούσα υπό-ενότητα θα αναφερθούμε σε ένα σημαντικό πρόβλημα που προκύπτει σε μια τέτοια κατασκευή, που είναι η αεροδυναμική αλληλεπίδραση μεταξύ των ροών που παράγονται από τους κινητήρες. Η αλληλεπίδραση μεταξύ των ροών δεν είναι αποδεκτή διότι είναι δύσκολο να μοντελοποιηθεί. Όπως αναφέρεται και στη δημοσίευση [4] στην οποία δεν έχει ληφθεί υπόψη η αεροδυναμική αλληλεπίδραση, είναι δυνατή μεν η εφαρμογή δυνάμεων και ροπών προς όλες τις κατευθύνσεις, αλλά η μεταφορική και η περιστροφική κίνηση δεν είναι εντελώς αποσυνδεδεμένες. Για αποφυγή αυτού του φαινομένου απαιτούμε να μην υπάρχει καμία αεροδυναμική αλληλεπίδραση. Αυτό βέβαια αυξάνει τον όγκο της κατασκευής αλλά καθιστά πιο εύκολο τον έλεγχο.

Όπως και στην διπλωματική του Γεώργιου Γαυρίδη [1] έτσι και εδώ θα γίνει χρήση των δεδομένων της εργασίας της Ζωής Τραχανά[3]. Ένα από τα θέματα αυτής της εργασίας ήταν η εύρεση της γεωμετρίας της ροής στην είσοδο και στην έξοδο των κινητήρων για διάφορες ταχύτητες περιστροφής. Γνωρίζοντας το μέγεθος των αεροδυναμικών κώνων (θα αναφερόμαστε στην μόνιμη ροή που παράγεται από έναν

κινητήρα σε σταθερές στροφές ως «κώνο», διότι η ροή παραπέμπει σε αυτό το σχήμα) θα μπορέσουμε να ορίσουμε και την απαίτηση για μη αλληλεπίδραση. Σύμφωνα με τα δεδομένα από την εργασία της Ζωής Τραχανά [3] ο κώνος που παράγεται από έναν κινητήρα NEU 1902/2Y – 2035 με προπέλα 8x4.7 Slow Fly APC στα 17550 rpm εκφρασμένος στο σύστημα συντεταγμένων του κινητήρα φαίνεται στο Σχήμα 2.5.1. Επιλέχθηκε ο κώνος των 17550 rpm ώστε να μην έχουμε αλληλεπίδραση ακόμα και στη μεγαλύτερη δυνατή ροή.

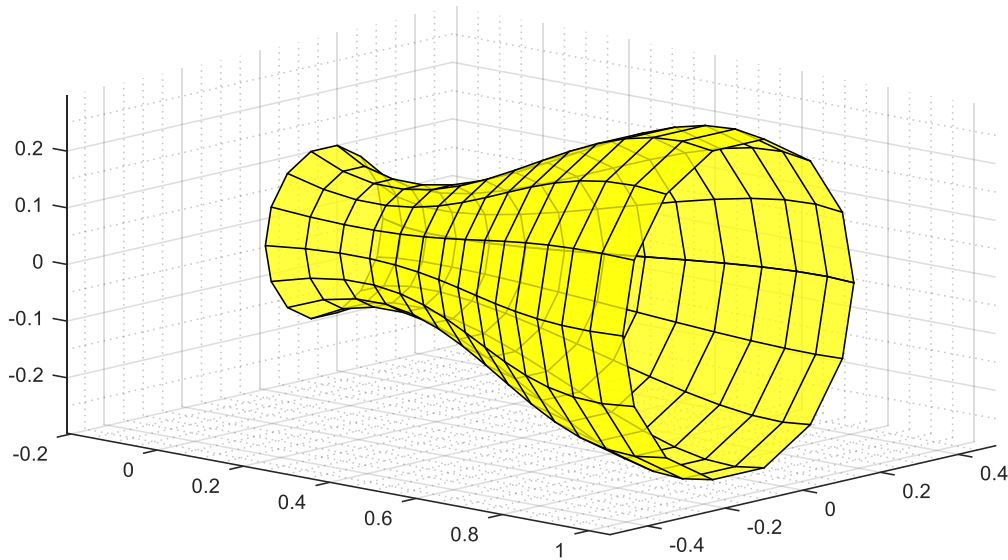


**Σχήμα 2.5.1** Πολυωνυμικό προφίλ αεροδυναμικής ροής.

Συγκεκριμένα ο κώνος εκτείνεται στην είσοδο του κατά 0,06 m, στην έξοδο του κατά 0,91 m, και το προφίλ του ακολουθεί ένα πολυώνυμο τρίτου βαθμού. Αν θεωρήσουμε ως άξονα συμμετρίας του κώνου τον άξονα των  $x$  τότε μπορούμε να διατυπώσουμε τα όρια του κώνου ως εξής:

$$\begin{cases} -0,06 \leq x'_i \leq 0,91 \\ (y'_i)^2 + (z'_i)^2 \leq [-1.11(x'_i)^3 + 1.56(x'_i)^2 - 0.31(x'_i) + 0.11]^2 \end{cases}$$

Στις παραπάνω εξισώσεις, η πρώτη αφορά την θέση πάνω στον άξονα συμμετρίας  $x$ , όπου για  $x=0$  είναι η θέση της προπέλας. Η δεύτερη δείχνει το κύκλο που διαγράφεται στο κάθετο επίπεδο του άξονα  $x$ , το επίπεδο  $yz$ , για την εκάστοτε τιμή του  $x$ .



**Σχήμα 2.5.2** Γραφική αναπαράσταση του αεροδυναμικού κώνου στο περιβάλλον προγραμματιστικό Matlab.

Στην διπλωματική εργασία του Γεώργιου Γαυρίδη η κύρια ιδέα για την ικανοποίηση αυτής της απαίτησης είναι ότι τα δυο πλησιέστερα σημεία που ανήκουν σε δυο διαφορετικούς κώνους θα πρέπει να έχουν απόσταση μεγαλύτερη από 0,01m. Για την εύρεση αυτής της απόστασης στην εν λόγω εργασία έγινε χρήση ενός αλγορίθμου βελτιστοποίησης με μεταβλητές τις συντεταγμένες των δυο σημείων, αντικειμενική συνάρτηση την απόσταση των σημείων, και περιορισμό το κάθε σημείο να ανήκει στον κώνο του. Έτσι μέσω αυτής της βελτιστοποίησης βρίσκεται η ελάχιστη απόσταση μεταξύ δυο αεροδυναμικών κώνων. Έπειτα η τιμή της αντικειμενικής συνάρτησης εισάγεται σαν περιορισμός στον κύριο αλγόριθμο βελτιστοποίησης. Η απαίτηση αυτή πρέπει να ικανοποιείται για όλους τους κώνους, που στην περίπτωση των 7 κινητήρων είναι  $\binom{7}{2} = 21$  ζευγάρια. Αν προσθέσουμε και τους περιορισμούς η ροή να μην αλληλεπιδρά με το κουτί των ηλεκτρονικών και με μια περιοχή κοντά στον end-effector τότε οι περιορισμοί γίνονται  $\binom{9}{2} = 36$ . Δηλαδή ο αλγόριθμος βελτιστοποίησης για την εύρεση της μικρότερης απόστασης μεταξύ δυο περιοχών, είτε αυτές είναι κώνοι ή το κουτί με τα ηλεκτρονικά ή η σφαίρα που περικλείει τον end-effector πρέπει να κληθεί 36 φορές σε κάθε επανάληψη του κύριου αλγορίθμου.

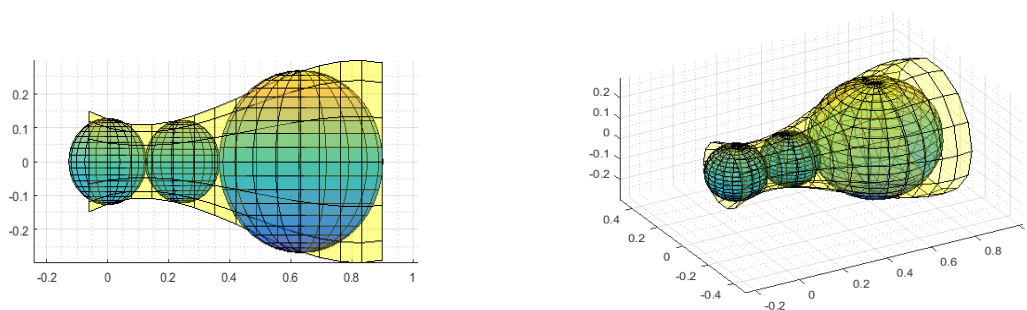
Ένα από τα πρώτα βήματα στα οποία προβήκαμε όταν ήρθαμε σε επαφή και έχοντας κατανοήσει τον αλγόριθμο βελτιστοποίησης της διπλωματικής εργασίας [1], ήταν να διεξάγουμε ένα profiling στον αλγόριθμο, δηλαδή να αναγνωρίσουμε πως κατανέμεται ο υπολογιστικός χρόνος μέσα στις διάφορες υπορουτίνες. Διαπιστώσαμε ότι μεγάλο μέρος του υπολογιστικού χρόνου καταναλωνόταν στον αλγόριθμο βελτιστοποίησης για την εύρεση της μικρότερης απόστασης μεταξύ των διαφόρων περιοχών. Χαρακτηριστικά για τον υπολογισμό των ελάχιστων αποστάσεων μεταξύ των κώνων χρειαζόταν 3.4 δευτερόλεπτα, όπου αν αναλογιστούμε ότι ο αλγόριθμος καλείται χιλιάδες φορές αυτό αυξάνει ραγδαία τον υπολογιστικό χρόνο. Συγκεκριμένα ένα τρέξιμο του κύριου αλγορίθμου το οποίο σύγκλινε μετά από 60 επαναλήψεις διήρκησε 1 ώρα και 26 λεπτά.

Επειδή γίνεται αναζήτηση τοπικού ελαχίστου, θέλουμε ο χρόνος τρεξίματος να είναι όσο το δυνατό μικρότερος ώστε να ελεγχθούν όσο το δυνατό περισσότερες αρχικές συνθήκες. Για το λόγω αυτό προσπαθήσαμε να ικανοποιήσουμε αυτή την απαίτηση με έναν λιγότερο χρονοβόρο τρόπο. Η αρχική μας σκέψη ήταν, αντί να έχουμε την γεωμετρία του κώνου, για την οποία είναι δύσκολο να υλοποιήσεις την απαίτηση σε μικρό χρόνο, αντίθετα να χρησιμοποιήσουμε πιο απλές γεωμετρίες. Επιλέξαμε λοιπόν να «παρεμβάλουμε» τον κώνο με ένα αριθμό σφαιρών ώστε να καλυφθεί όσο το δυνατόν μεγαλύτερος όγκος του κώνου. Έτσι αντί να απαιτούμε τα πλησιέστερα σημεία δυο κώνων να απέχουν τουλάχιστον 0,01m, απαιτούμε τα κέντρα δυο σφαιρών να απέχουν το άθροισμα των ακτινών τους συν 0,01m.

Γίνεται αντιληπτό ότι με αυτό τον τρόπο δεν πετυχαίνουμε ακριβώς τη γεωμετρία του κώνου, και ακόμα ότι οι περιορισμοί αυξάνονται αρκετά. Για παράδειγμα, για 3 σφαίρες ανά κώνο και 7 κινητήρες συν δυο περιοχές για το κουτί με τα ηλεκτρονικά και τον end-effector πρέπει να ελεγχθούν  $\binom{7+3+2}{2} - 7 * 3 = 253 - 21 = 232$  ζευγάρια (αφαιρούμε τα ζευγάρια που ανήκουν στον ίδιο κώνο). Όμως ο υπολογιστικός χρόνος για αυτούς τους περιορισμούς είναι 0,03 ms στον ίδιο επεξεργαστή που δοκιμάσαμε και την προηγούμενη μέθοδο. Ο αντίστοιχος χρόνος για να γίνουν 60 επαναλήψεις του κύριου αλγορίθμου είναι γύρω στα 10 δευτερόλεπτα έναντι της 1

ώρας και 26 λεπτών που χρειαζότανε πριν (οι δοκιμές έγιναν στον ίδιο υπολογιστή στις ίδιες παραμέτρους). Έχοντας μειώσει κατά πολύ τον υπολογιστικό χρόνο μας δόθηκε όπως θα δούμε και στη συνέχεια η δυνατότητα να προσθέσουμε περαιτέρω περιορισμούς.

Επειδή εμφανίστηκε η ανάγκη οι κώνοι να έχουν μεταβλητό μήκος, το οποίο θα ορίζεται από το χρήστη πριν την έναρξη του αλγορίθμου δημιουργήθηκε ένας αλγόριθμος που θα βελτιστοποιεί τη θέση και το μέγεθος των σφαιρών του κώνου. Αν θεωρήσουμε ότι έχουμε 3 σφαίρες στον κώνο τότε οι μεταβλητές αυτής της βελτιστοποίησης είναι συνολικά 4, 2 για τον ορισμό των θέσεων των σφαιρών και 2 για τον ορισμό της ακτίνας των σφαιρών (η μια σφαίρα έχει συγκεκριμένη θέση και μέγεθος για να εξασφαλίζεται η κάλυψη της προπέλας). Ακόμα η αντικειμενική συνάρτηση είναι το εμβαδόν του κώνου μείον το εμβαδόν των σφαιρών (χρησιμοποιούμε το εμβαδόν επειδή ο κώνος και οι σφαίρες είναι συμμετρικές ως προς τον άξονα X). Τους περιορισμούς αποτελούν η μη αλληλοκάλυψη των 2 σφαιρών μεταξύ τους όσο και με την πρώτη σφαίρα, τα κέντρα τους να μην ξεπερνούν το μέγιστο μήκος του κώνου και να μην πλησιάζουν την πρώτη σφαίρα πλησιέστερα από την ακτίνα της.



**Σχήμα 2.5.3** Πλάγια και προοπτική όψη του αποτελέσματος της βελτιστοποίησης των σφαιρών του κώνου για μήκος κώνου 0,9m και προπέλα 10 ιντσών.

## 2.6 Πρόβλημα Σχεδιασμού

Στη συγκεκριμένη υπό-ενότητα θα αναφερθούμε στο πρόβλημα σχεδιασμού το οποίο είναι προσανατολισμένο στον έλεγχο της κατασκευής.

Έχουμε την συγκεκριμένη κατασκευή που προσδιορίζεται από τον πίνακα  $D(r, \hat{F})$  ο οποίος περιέχει τις θέσεις και τους προσανατολισμούς των κινητήρων. Για κάθε σετ δυνάμεων και ροπών αλληλεπίδρασης  $(F_{act}, M_{act})$  που θέλουμε να ασκήσουμε είναι προφανές ότι χρειαζόμαστε τους αντίστοιχους συντελεστές ώσης  $\lambda_i$ .

Γνωρίζοντας ότι  $W_R \in R^6$  για να έχουμε λύση στην εξίσωση (2.10) ως προς τους συντελεστές

$$\lambda = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_n]^T$$

πρέπει να ισχύει

- $n \geq 6$  και
- ο πίνακας  $D(r, \hat{F})$  να έχει  $\text{rank}[D(r, \hat{F})]=6$

με αυτό τον τρόπο εξασφαλίζουμε ότι το σύστημα μας θα έχει 6 βαθμούς ελευθερίας το οποίο είναι και το πιο βασικό ζητούμενο του σχεδιασμού.

Ο βαθμός του πίνακα  $D(r, \hat{F})$  από μαθηματικής απόψεως είναι επαρκής, όμως από πρακτικής άποψης δεν είναι, διότι δεν εξασφαλίζει την ευρωστία του συστήματος. Για αυτό τον λόγο αντί να χρησιμοποιήσουμε το κριτήριο του βαθμού του πίνακα D, θα χρησιμοποιήσουμε το *condition number*, που ορίζεται ως εξής:

$$\kappa(D) = \frac{\sigma_{\max}(D)}{\sigma_{\min}(D)} \quad (2.28)$$

όπου τα  $\sigma_{\max}(D)$ ,  $\sigma_{\min}(D)$  είναι οι μέγιστη και η ελάχιστη singular value του  $D(r, \hat{F})$ . Αν απαιτήσουμε η τιμή του  $\kappa(D)$  να είναι μικρή, ιδανικά ίση με 1, τότε πετυχαίνουμε την ευρωστία του συστήματος, δηλαδή να μην είναι ευαίσθητο σε μικρές παραμετρικές αβεβαιότητες και μεταβολές. Επομένως η πρώτη απαίτηση/περιορισμός είναι,

$$\kappa(D) \leq K$$

όπου  $K$  είναι ένα όριο το οποίο ορίζουμε εμείς.

Παρατηρώντας την εξίσωση (2.28) βλέπουμε ότι οι singular values του πίνακα  $D(r, \hat{F})$  δεν μπορούν να είναι μηδέν, ώστε να μην μηδενίζεται ο παρονομαστής του κλάσματος. Εκτός αυτού, όταν οι singular values είναι κοντά στο μηδέν, παρότι το condition number θα παίρνει εφικτές τιμές, ο πίνακας  $D(r, \hat{F})$  μπορεί να είναι κοντά στο να γίνει ιδιόμορφος με μικρές παρεμβολές του συστήματος με αποτέλεσμα η ορίζουσα του να γίνει μηδέν. Επομένως εισάγεται άλλος ένας περιορισμός:  $\sigma(D) \geq e_1$ , όπου  $e_1$  είναι μια μικρή ποσότητα, την οποία εμείς ορίζουμε ίση με  $e_1 = 0.01$ .

Η λύση που θα προκύψει θέλουμε να ικανοποιεί, εκτός από τους παραπάνω περιορισμούς, που είναι η μη αλληλεπίδραση των ροών, το condition number και οι singular values του πίνακα  $D$ , αλλά και τους περιορισμούς που καθιστούν εφικτή τη μεθοδολογία μετατροπής των θετικών και αρνητικών συντελεστών ώσης σε μόνο θετικούς. Αυτοί οι επιπλέον περιορισμοί είναι οι σχέσεις (2.17), (2.18).

### 2.6.1 Η Προηγούμενη Βελτιστοποίηση

Στο πέμπτο κεφάλαιο της διπλωματικής του Γεώργιου Γαυρίδη [1] περιγράφεται η δομή της λύσης του προβλήματος, το οποίο παρουσιάστηκε στην παραπάνω υπό-ενότητα όπως και στο 4 κεφάλαιο της προαναφερθείσας διπλωματικής εργασία. Στην υπό-ενότητα αυτή θα παρουσιάσουμε τη δομή της λύσης και θα επισημάνουμε τις αλλαγές που έγιναν με σκοπό να ικανοποιηθούν οι επιπλέον περιορισμοί που εισήχθησαν.

Στο υποκεφάλαιο 5.2 [1] περιγράφονται τα διάφορα στοιχεία που χαρακτηρίζουν μια βελτιστοποίηση. Αν δηλαδή γίνεται αναζήτηση τοπικού ή ολικού ακροτάτου, αν οι μεταβλητές είναι πραγματικοί αριθμοί ή ακέραιοι, αν το πρόβλημα είναι γραμμικό ή μη-γραμμικό, αν είναι ντετερμινιστικό ή στοχαστικό ή αν το πρόβλημα έχει περιορισμούς ή όχι. Το πρόβλημα, όπως παρουσιάστηκε στην [1] είναι ένα ντετερμινιστικό πρόβλημα με μη-γραμμική αντικειμενική συνάρτηση και περιορισμούς, με πραγματικές μεταβλητές και όσον αφορά την βελτιστοποίηση γίνεται αναζήτηση τοπικού ελαχίστου. Στον Πίνακα 2.6.1 παρουσιάζεται η δομή του προβλήματος.



(P <sub>1</sub> )	$d_{ij}(r_i, \hat{F}_i, r_j, \hat{F}_j) = \min_{p_i, p_j} \ p_i - p_j\ $ $s.t. \quad G(r_i, \hat{F}_i, p_i) \leq 0$ $G(r_j, \hat{F}_j, p_j) \leq 0$
(P <sub>2</sub> )	$\min_{r, \hat{F}} J(r)$ $s.t. \quad \sigma(D) \geq \epsilon_1$ $d_{ij} \geq \epsilon_2, \forall i, j = 1, 2, \dots, n, \alpha$ $d_{ej} \geq R_e, \forall i = 1, 2, \dots, n, \alpha$ $\hat{F}_a = - \sum_{i=1}^n \hat{F}_i$ $S(\hat{F}_a)r_a = - \sum_{i=1}^n S(\hat{F}_i)r_i$ $1 \leq \kappa(D) \leq K$

**Πίνακας 2.6.1:** Η δομή του της παλαιάς βελτιστοποίησης.

Η οποία αποτελείται από δυο βελτιστοποιήσεις, μια μικρή (P<sub>1</sub>) και μια μεγάλη (P<sub>2</sub>).

Η βελτιστοποίηση P<sub>1</sub> έχει τα εξής χαρακτηριστικά :

- Αντικειμενική συνάρτηση : Απόσταση μεταξύ δυο σημείων στο χώρο.
- Περιορισμοί : Ανισοτικοί - Το καθένα ένα από αυτά τα σημεία να ανήκει σε ένα αεροδυναμικό κώνο, ο οποίος ορίζεται από τα διανύσματα θέσης και προσανατολισμού του κινητήρα.
- Οι αντικειμενική συνάρτηση όπως και οι περιορισμοί είναι μη-γραμμικοί.
- Έχει ομαλή (smooth) αντικειμενική συνάρτηση και περιορισμούς, αποτελεί όμως μη ομαλό (non\_smooth) περιορισμό στην βελτιστοποίηση P<sub>2</sub>.

Η βελτιστοποίηση P<sub>2</sub> έχει τα εξής χαρακτηριστικά :

- Αντικειμενική συνάρτηση : ευκλείδεια νόρμα του πίνακα γραμμή των θέσεων των κινητήρων.
- Περιορισμοί :
  1. Τα singular values του μητρώου D να είναι μεγαλύτερα από έναν αριθμό σε αυτή περίπτωση 0,001 (ανισοτικός).
  2. Η ελάχιστη απόσταση μεταξύ δύο αεροδυναμικών κώνων να είναι μεγαλύτερη από 0,01 m (ανισοτικός).
  3. Η απόσταση μεταξύ ενός κώνων και της σφαίρας εσωτερικά της οποίας βρίσκεται ο χειριστής να είναι μεγαλύτερη (ανισοτικός).

4. Ο προσανατολισμός του επιπλέον κινητήρα να είναι αντίθετος από το άθροισμα όλων των προσανατολισμών (ισοτικός).
  5. Ο μοχλοβραχίονας του επιπλέον κινητήρα να είναι ίσος και αντίθετος από τον άθροισμα των μοχλοβραχιόνων των υπόλοιπων κινητήρων (ισοτικός).
  6. Το condition number του μητρώου D να είναι μεγαλύτερο από 10 σε αυτή την περίπτωση (ανισοτικός).
- Η αντικειμενική συνάρτηση και οι περιορισμοί είναι μη γραμμικοί και μη ομαλοί.

### 2.6.2 Η Νέα βελτιστοποίηση

Έχοντας μειώσει τον υπολογιστικό χρόνο σε επίπεδα που μας επέτρεπαν να προσθέσουμε επιπλέον περιορισμούς όπως είδαμε στην υποενότητα 2.5, προβήκαμε σε μια προσθήκη που θεωρούσαμε αναγκαία για την λειτουργία της κατασκευής. Πιο συγκεκριμένα, παρατηρήσαμε ότι για κάποιες αρχικά καλές λύσεις (μικρός όγκος του AM) οι ζητούμενοι συντελεστές ώσης, μόνο για την αιώρηση σε διάφορους προσανατολισμούς του AM, ήταν μεγαλύτεροι από ότι θα μπορούσε να υποστηριχτεί τεχνολογικά με τα σημερινά δεδομένα. Αλλάξαμε λοιπόν την αντικειμενική συνάρτηση από την ευκλείδεια νόρμα του πίνακα των θέσεων, στην άπειρη νόρμα των μέγιστων συντελεστών ώσης κάθε προσανατολισμού του AM σε αιώρηση.

$$F_{obj} = \left\| \max_i \lambda_i \right\|_{\infty}$$

Οι συντελεστές ώσης για κάθε προσανατολισμό υπολογίζονται από την σχέση (2.10) επειδή μας ενδιαφέρει μόνο η αιώρηση του AM θα έχουμε  $M_{act}=0$  και  $F_{act}=0$ . Επειδή αυτή η σχέση είναι εκφρασμένη στο σωματόδετο σύστημα συντεταγμένων του AM, όταν ο AM αλλάζει προσανατολισμό στο αδρανειακό σύστημα συντεταγμένων στο σωματόδετο σύστημα το διάνυσμα του βάρους είναι αυτό που θα περιστρέφεται. Άρα ο πίνακας D παραμένει ο ίδιος ενώ ο πίνακας  $W_R$  προσαρμόζεται ανάλογα με τον προσανατολισμό. Έχουμε καταλήξει σε μια διακριτοποίηση 10 μοιρών έτσι ώστε να μην χάνουμε πληροφορία αλλά να μην κάνουμε και τον αλγόριθμο χρονοβόρο. Η διακριτοποίηση γίνεται στην περιστροφή των αξόνων Y και Z. Με δύο επαναλήψεις

όπου στην πρώτη περιστρέφουμε κατά τον Z άξονα και στη δεύτερη κατά τον Y άξονα (η περιστροφή κατά τον X άξονα είναι περιττή) έχουμε συνολικά  $36 \times 36 = 1236$  διαφορετικούς προσανατολισμούς. Σε κάθε ένα προσανατολισμό υπολογίζουμε τους απαιτούμενους συντελεστές ώσης συγκρίνουμε τότε τον μεγαλύτερο από αυτούς με τον μέχρι εκείνη τη στιγμή μεγαλύτερο συντελεστή ώσης από τους προηγούμενους προσανατολισμούς. Έτσι έχουμε για την συγκεκριμένη διάταξη την μέγιστη απαιτούμενη ώση η οποία είναι και η τιμή της αντικειμενικής συνάρτησης.

Για να διασφαλίσουμε το μικρό όγκο της κατασκευής έχουμε εισάγει άνω και κάτω όρια στις μεταβλητές των θέσεων των κινητήρων. Επίσης το βάρος της κατασκευής είναι συνάρτηση του όγκου και αυτό ώστε με μείωση του όγκου να έχουμε τελικά μείωση των απαιτούμενων δυνάμεων ώσης.

Οι περιορισμοί στη νέα βελτιστοποίηση παραμένουν οι ίδιοι με την παλαιά εκτός από τον περιορισμό της αεροδυναμικής αλληλεπίδρασης, όπου όπως αναφέραμε στην παράγραφο (2.5) αντικαταστάθηκε από μια άλλη μεθοδολογία. Έτσι δεν υπάρχει ανάγκη για τη βελτιστοποίηση ( $P_1$ ). Επίσης προστίθεται και ένας επιπλέον περιορισμός  $\cos(\theta)$  ο οποίος θα εξηγήσουμε στη συνέχεια τι εξυπηρετεί. Στον Πίνακα 2.6.2 παρουσιάζεται η δομή της νέας βελτιστοποίησης. Η νέα βελτιστοποίηση θα ονομαστεί ( $P_1$ ).

(P <sub>1</sub> )	$\min_{r, \hat{F}} \ \lambda\ _{\infty}$ $s.t. \quad lb \leq r_i^{x,y,z} \leq ub$ $1 \leq \kappa(D) \leq K$ $\sigma(D) \geq \epsilon_1$ $d_{ij} \geq r_i + r_j + 0,01,$ $\forall i, j = 1, 2, \dots, 3 * n + 2$ $\hat{F}_a = - \sum_{i=1}^n \hat{F}_i$ $S(\hat{F}_a)r_a = - \sum_{i=1}^n S(\hat{F}_i)r_i$ $1 \leq \kappa(D) \leq K$ $0.90 < \cos(\theta) < 1$
-------------------	---

**Πίνακας 2.6.2:** Η δομή της νέας βελτιστοποίησης.

Στη νέα βελτιστοποίηση έχουμε μόνο μια βελτιστοποίηση την P1.

Τα νέα χαρακτηριστικά της P1 είναι:

- lb,ub : lower και upper bound για τις συντεταγμένες θέσης των κινητήρων
- $d_{ij} \geq r_i + r_j + 0,01$  : η απόσταση των κέντρου μιας σφαίρας να είναι μεγαλύτερη από το άθροισμα των ακτινών συν 1 εκατοστό, από οποιαδήποτε άλλη σφαίρα.
- $\text{Cos}(\theta)$  : η γωνία που ορίζεται από το διάνυσμα προσανατολισμού της δύναμης και από το διάνυσμα με αρχή τη θέση του κουτιού των ηλεκτρονικών και πέρας τη θέση του εκάστοτε κινητήρα. Ο περιορισμός αυτός μπαίνει σε ισχύ μόνο όταν απαιτήσουμε οι προσανατολισμοί των διανυσμάτων ώσης να έχουν κατεύθυνση προς τα έξω.

Για να υπολογιστεί αυτή η γωνία εργαζόμαστε ως εξής:

Το σκεπτικό είναι το διάνυσμα της δύναμης να ασκεί εφελκυστικές δυνάμεις, άρα θέλουμε να έχει κατεύθυνση προς τα έξω από το κέντρο της κατασκευής που προσεγγιστικά είναι το κουτί μετά ηλεκτρονικά. Επομένως αφαιρώντας το διάνυσμα θέσης του κουτιού από το διάνυσμα θέσης του κινητήρα, παίρνουμε το διάνυσμα με αρχή τη θέση του κουτιού και τέλος τη θέση του κινητήρα.

$$r_{box_{thr}} = r_{thr} - r_{box}$$

Έπειτα παίρνοντας το εσωτερικό γινόμενο του μοναδιαίου  $\hat{r}_{box_{thr}}$  και του διανύσματος προσανατολισμού της δύναμης, δηλαδή του κινητήρα  $\hat{f}_{thr}$ :

$$\hat{r}_{box_{thr}} \cdot \hat{f}_{thr} = \|\hat{r}_{box_{thr}}\| \cdot \|\hat{f}_{thr}\| \cdot \cos(\theta)$$

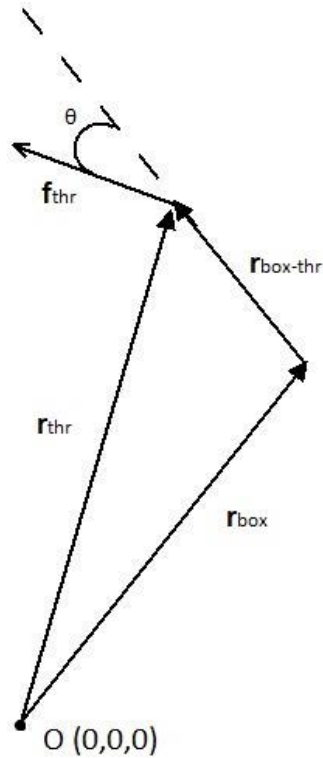
Άρα

$$\cos(\theta) = \frac{\hat{r}_{box_{thr}} \cdot \hat{f}_{thr}}{\|\hat{r}_{box_{thr}}\| \cdot \|\hat{f}_{thr}\|} \Rightarrow$$

$$\cos(\theta) = \hat{r}_{box_{thr}} \cdot \hat{f}_{thr}$$

Δηλαδή ο περιορισμός  $0.90 < \text{Cos}(\theta) < 1$  γίνεται ουσιαστικά

$$0.90 < \hat{r}_{box_{thr}} \cdot \hat{f}_{thr} < 1$$



**Σχήμα 2.6.1** Γεωμετρική απεικόνιση της γωνίας  $\theta$

Ο περιορισμός της προαναφερθείσας γωνίας μπήκε διότι υπήρξε η ανάγκη για διερεύνηση λύσεων στις οποίες να μην υπάρχουν εφελκυστικές τάσεις στις ράβδους του δικτυώματος. Με αυτό τον τρόπο αποφεύγουμε το φαινόμενο του λυγισμού στις ράβδους, αλλά κυριότερα αποφεύγουμε το φαινόμενο του ανεστραμμένου εκκρεμούς κατά την αιώρηση του ΑΜ.

Όταν ο προσανατολισμός των κινητήρων είναι ελεύθερος να ορισθεί από την βελτιστοποίηση χωρίς περιορισμούς, τότε τυχαίνει λόγο και της απαίτησης μικρού μεγέθους να στρέφονται οι κινητήρες προς το κέντρο, ώστε οι αεροδυναμικοί κώνοι να είναι στραμμένοι προς τα έξω και να ελαχιστοποιείται το μέγεθος της κατασκευής. Όταν όμως οι δυνάμεις έχουν κατεύθυνση προς το κέντρο έχουμε το φαινόμενο του ανεστραμμένου εκκρεμούς κατά την αιώρηση. Διότι καθώς οι δυνάμεις κοιτάνε προς το κέντρο, όταν θέλουμε να τα ισορροπήσουμε στον αέρα, οι δυνάμεις που θα το κρατάνε στον αέρα θα είναι αυτές που το σπρώχνουν από κάτω.

Ο αλγόριθμος βελτιστοποίησης παρατίθεται στο Παράρτημα Α με επαρκή σχόλια.

### 2.6.3 Εύρεση Καλύτερης Λύσης

#### 1) Πρόβλημα Ροπής Αντίδρασης

Όπως έχουμε αναφέρει κάθε προπέλα όταν περιστρέφεται παράγει μια ροπή αντίδρασης η οποία είναι παράμετρος που καθορίζει το διάνυσμα  $W_R$  ( βλ. εξίσωση 2.9) . Καθώς βλέπουμε όμως ότι αυτή η ροπής αντίδρασης έχει συγκεκριμένο πρόσημο, το οποίο είναι ίδιο για όλους τους κινητήρες, δηλαδή σημαίνει ότι όλες οι προπέλες στρέφονται προς την ίδια κατεύθυνση. Αυτό όμως δεν είναι το πιο αποδοτικό. Για παράδειγμα στα συμβατικά multi-copters οι προπέλες στρέφονται πάντα μισές αριστερόστροφα και μισές δεξιόστροφα, κι αυτό για να αλληλοαναιρούνται οι παραγόμενες ροπές αντίδρασης.

Στην περίπτωση του AM , όπου οι προπέλες δε βρίσκονται όλες στο ίδιο επίπεδο , δεν είναι εύκολο να γίνει αντίστοιχη απλοποιητική παραδοχή. Γι' αυτό κάθε αποτέλεσμα που βγαίνει από την ανωτέρω βελτιστοποίηση P1, το τρέχουμε ξανά δοκιμάζοντας όλους τους πιθανούς συνδυασμούς των φορών περιστροφής των προπελών. Ο αριθμός των εκάστοτε συνδυασμών βγαίνει από τον τύπο :  $N = 2^n$ , όπου  $n$  ο αριθμός των κινητήρων.

Για να γίνει αυτό τροποποιούμε την εξίσωση 2.10 ως εξής:

$$\begin{aligned}
 D(r, \hat{F}) \cdot \lambda &= W_R \Leftrightarrow \\
 \begin{bmatrix} \hat{F} \\ E(r, \hat{F}) \end{bmatrix} \cdot \lambda &= \begin{bmatrix} F_{act} - n \cdot w - w_s \\ M_{act} - \mu \cdot (F_{act} - n \cdot w - w_s) - \left( \sum_{i=1}^n r_i \right) \times w - r_s \times w_s \end{bmatrix} \Leftrightarrow \\
 \begin{bmatrix} \hat{F} \\ E(r, \hat{F}) \pm \mu \cdot \hat{F} \end{bmatrix} \cdot \lambda &= \begin{bmatrix} F_{act} - n \cdot w - w_s \\ M_{act} - \left( \sum_{i=1}^n r_i \right) \times w - r_s \times w_s \end{bmatrix} \Leftrightarrow \\
 \begin{bmatrix} \hat{F} \\ E(r, \hat{F}) + \widehat{Fm} \end{bmatrix} \cdot \lambda &= \begin{bmatrix} F_{act} - n \cdot w - w_s \\ M_{act} - \left( \sum_{i=1}^n r_i \right) \times w - r_s \times w_s \end{bmatrix} \quad (2.29)
 \end{aligned}$$

όπου  $\widehat{Fm} = \pm \mu \cdot \hat{F}$  και το πρόσημο αλλάζει για κάθε περίπτωση.

## 2) Πρόβλημα επίλυσης υπό-ορισμένου συστήματος

Όπως φαίνεται στη (2.29) το σύστημα εξισώσεων που αντιμετωπίζουμε είναι ένα σύστημα  $[6 \times n]$  εξισώσεων,  $n > 6$  δηλαδή είναι ένα υπό-ορισμένο σύστημα αφού έχουμε παραπάνω αγνώστους από ότι εξισώσεις. Αυτό το πρόβλημα λύθηκε με την μεθοδολογία που αναπτύχθηκε στο κεφάλαιο 2.4, όπου αναδιανέμονται οι δυνάμεις ώστε να είναι όλες θετικές. Η μεθοδολογία αυτή έχει ένα πλεονέκτημα και ένα μειονέκτημα. Το πλεονέκτημα είναι ότι είναι γρήγορη σε υπολογιστικό χρόνο, το μειονέκτημα ότι παράγει μεγάλες απαιτούμενες δυνάμεις, διότι προσθέτει το άθροισμα των αρνητικών δυνάμεων.

Για να ξεπεράσουμε αυτό το μειονέκτημα, θυσιάζοντας βέβαια υπολογιστικό χρόνο, χρησιμοποιούμε την μεθοδολογία των μη-αρνητικών ελαχίστων τετραγώνων (Least Square Non-Negative). Η οποία έχοντας μία εξίσωση της μορφής  $A \cdot x = b$  υπολογίζει τις παραμέτρους έτσι ώστε :

$$* \operatorname{argmin}_x \|Ax - b\|_2 \text{ subject to } x \geq 0$$

με τον τρόπο αυτό θυσιάζοντας σε υπολογιστικό χρόνο, αφού πραγματοποιείται βελτιστοποίηση, κερδίζουμε σε αποτέλεσμα αφού μέσω αυτής παίρνουμε την βέλτιστη διανομή θετικών δυνάμεων στους κινητήρες.

Επομένως χρησιμοποιώντας όσα είπαμε παραπάνω, κάθε λύση που έχει βγάλει η βελτιστοποίηση P1 της προηγούμενης παραγράφου, κρατάμε τις θέσεις και τους προσανατολισμούς που έχει βγάλει η βελτιστοποίηση P1, και ξανά υπολογίζουμε τη μέγιστη απαιτούμενη ώση, λαμβάνοντας υπόψη τις παραπάνω μεθοδολογίες, ώστε να βρούμε το βέλτιστο συνδυασμό φόρας περιστροφής κάθε προπέλας με το λιγότερο μέγιστο απαιτούμενο thrust υπολογισμένο με τη μεθοδολογία Least Square Nonneg. Δηλαδή για κάθε πιθανό συνδυασμό περιστροφής προπέλας, ελέγχουμε τη μέγιστη απαιτούμενη ώση με LSQ. Ο λόγος που δε χρησιμοποιούμε εξ αρχής αυτή την

μεθοδολογία, αλλά την εφαρμόζουμε εκ των υστέρων στα υπάρχοντα αποτελέσματα είναι ο εξής:

- Πρώτον επειδή γίνεται βελτιστοποίηση είναι αδύνατον βάση υπολογιστικού χρόνου να υπολογίζεται μέσα στην βελτιστοποίηση κάθε συνδυασμός φοράς περιστροφής προπέλας
- Και δεύτερον η μεθοδολογία Least Square που αναφέραμε, είναι μία μεθοδολογία curve fitting, δηλαδή αν υπάρχει λύση την προσεγγίζει με ακρίβεια, αλλά αν δεν υπάρχει λύση τότε δίνει μια προσεγγιστική λύση η οποία απέχει από την πραγματική. Επομένως όταν προσπαθήσαμε να τρέξουμε τη βελτιστοποίηση βάση αυτής της μεθοδολογίας, τα αποτελέσματα που παίρναμε ήταν μη ρεαλιστικά.

## 2.7 Παραμετροποίηση

Η νέα βελτιστοποίηση έχει παραμετροποιηθεί ως προς κάποιους παράγοντες για την εύρεση της βέλτιστης λύσης.

### Αριθμός Κινητήρων

Η πρώτη παραμετροποίηση αφορά στον αριθμό των κινητήρων. Αριθμός αυτός λόγω της φύσης του προβλήματος δεν μπορεί να είναι μικρότερος από 6 και για λόγους υπολογιστικής ισχύς δεν μπορεί να είναι μεγαλύτερος από 14. Στην παρούσα εργασία εξετάστηκαν οι περιπτώσεις για 7, 8 και 9 κινητήρες.

### Μέγεθος Αεροδυναμικών Κώνων

Η δεύτερη παραμετροποίηση αφορά στο μήκος των αεροδυναμικών κώνων που παράγονται από του κινητήρες. Το μήκος μπορεί να μεταβληθεί από την τιμή 0.3m έως 1.2m. Η παράμετρος αυτή εισάχθηκε για να διαπιστωθεί κατά πόσο επηρεάζει την λύση. Καθώς όσο μεγαλύτεροι είναι οι κώνοι τόσο μεγαλύτερη είναι η κατασκευή, το οποίο με τη σειρά του σημαίνει αύξηση του βάρους και τελικά αύξηση των



απαιτούμενων δυνάμεων ώσης από τους κινητήρες που είναι η αντικειμενική μας συνάρτηση.

### **Μέγεθος Προπέλας**

Επειδή δεν γνωρίζουμε ποιος θα είναι οι βέλτιστος ενεργειακά συνδυασμός κινητήρα προπέλας και επειδή αυτό ξεφεύγει από το όρια της παρούσας εργασίας. Θεωρήσαμε εύλογο και χρήσιμο για την περαιτέρω εξέλιξη της κατασκευής του AM να δίνεται η δυνατότητα προσδιορισμού της διαμέτρου της προπέλας. Ανάλογα με το μέγεθος της προπέλας ρυθμίζεται και αντίστοιχα η ακτινική διάσταση του κώνου.

### **Προσανατολισμός Κινητήρων**

Αυτή η παράμετρος έχει προστεθεί για δυο λόγους. Πρώτον, παρατηρήσαμε ότι για να μειωθεί ο όγκος της κατασκευής ο αλγόριθμος έστρεφε του κινητήρες έτσι ώστε η κατάθλιψη των αεροδυναμικών κώνων να είναι απομακρυσμένη από το κέντρο της κατασκευής, καθώς όσο είναι στραμμένη η κατάθλιψη προς τα μέσα δεν μπορούν να πλησιάσουν οι κινητήρες προς το κέντρο μειώνοντας έτσι τον όγκο. Αν οι κώνοι είναι στραμμένοι προς τα έξω τότε η κατεύθυνση της εφαρμοζόμενης δύναμης είναι προς τα μέσα. Αυτή η διάταξη παραπέμπει στο ανάστροφο εκκρεμές το οποίο θα επιφέρει προβλήματα στον μετέπειτα έλεγχο του AM. Για την σταθεροποίηση του AM στον αέρα (hovering) θα απαιτείται οι εκάστοτε πλευρικοί κινητήρες να εφαρμόζουν δυνάμεις με εναλλαγές σε συχνότητες που το σύστημα να μην μπορεί να αποκριθεί.

### **Επιπλέον Παράμετροι**

Κάποιοι επιπλέον παράμετροι που μπορούν να ρυθμιστούν εσωτερικά του αλγορίθμους βελτιστοποίησης:

- Συντελεστές του πολυωνύμου που εκ περιστροφής παράγει τον κώνο.
- Αριθμός σφαιρών συναρτήσει του μήκους του κώνου (για μήκος κώνου μικρότερο ή ίσο από 0.6m χρησιμοποιούνται 2 σφαίρες, για μεγαλύτερο 3 σφαίρες).

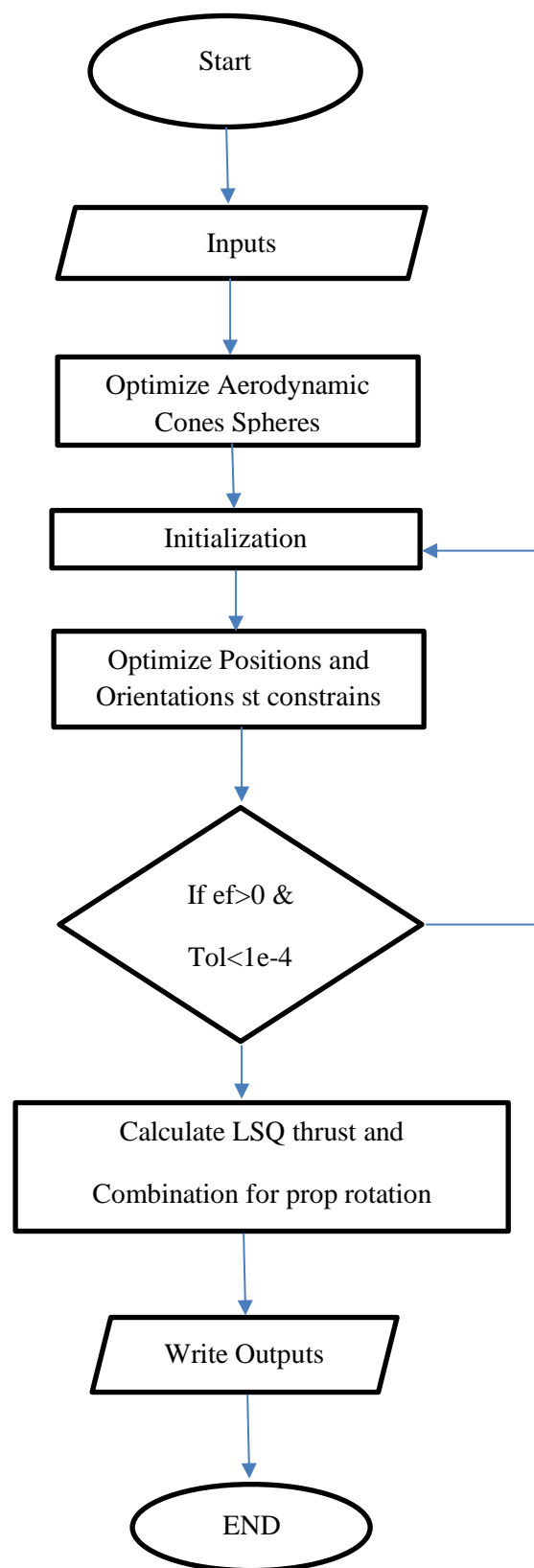
- Η ακμή του κύβου εσωτερικά του οποίου βρίσκονται οι αρχικές τιμές της βελτιστοποίησης.
- Το άνω και κάτω όριο των μεταβλητών της βελτιστοποίησης.
- Η γωνία της κορυφής του κώνου μέσα στον οποίο μπορεί να στραφεί το διάνυσμα προσανατολισμού του κινητήρα με τον κινητήρα να βρίσκεται στην κορυφή του κώνου, έτσι ώστε να επιτευχθεί ο περιορισμός του προσανατολισμού των κινητήρων.

Οι τέσσερις παράμετροι που όριζαν τις αρχικές συνθήκες είναι:

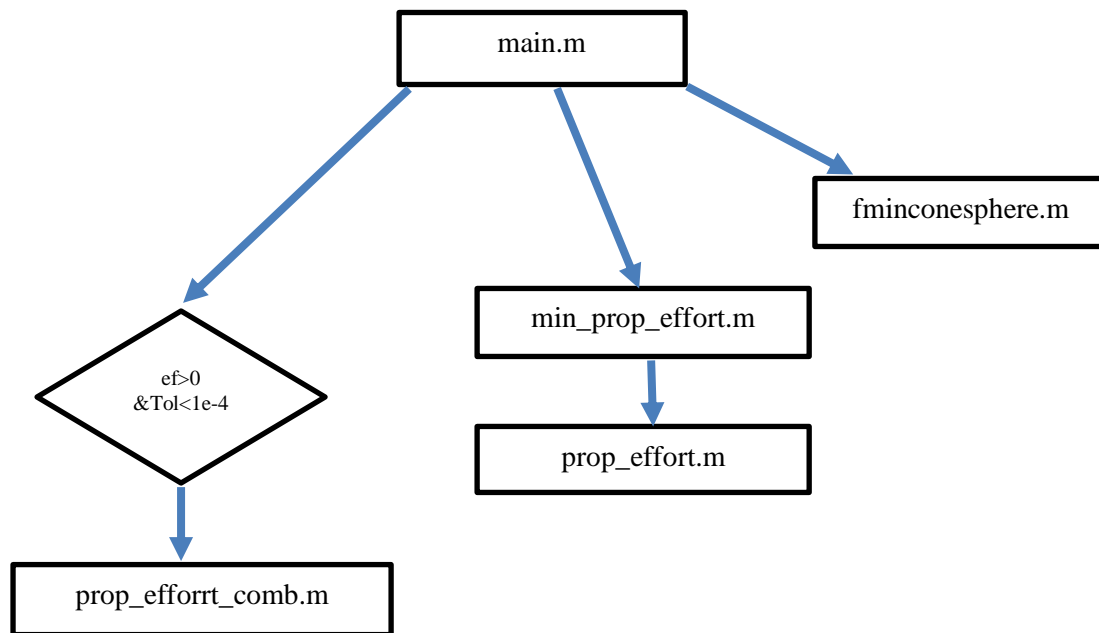
- Αριθμός κινητήρων.
- Μέγεθος αεροδυναμικού κώνου.
- Μέγεθος προπέλας.
- Προσανατολισμός κινητήρων.

Με βάση αυτές τις παραμέτρους «τρέξαμε» τον αλγόριθμο βελτιστοποίησης για όλους τους δυνατούς συνδυασμούς. Συγκεκριμένα, επιλέξαμε ο αριθμός κινητήρων να κυμαίνεται από 7 έως 9, το μήκος του αεροδυναμικού κώνου και αυτό να χωρίζεται σε τρεις τιμές 0.4, 0.6 και 0.9 m, η διάμετρος της προπέλας να μένει σταθερή στις 10 ίντσες και τελευταία ο προσανατολισμός των κινητήρων να παίρνει δυο τιμές «ελεύθερος» και «εσωτερικός». Οι συνολικοί συνδυασμοί είναι τελικά 18.

Για να γίνει καλύτερα αντιληπτή η πορεία του ακολουθούν οι κώδικες ακολουθεί ένα διάγραμμα ροής και ένα διάγραμμα που δείχνει με ποια σειρά καλούνται οι συναρτήσεις στον κώδικα. Σχήμα 2.7.1 φαίνεται η λογική πορεία του αλγορίθμου, όπου `ef` και `Tol` είναι Exit Flag και Tolerance αντίστοιχα που είναι output της συνάρτησης στοχαστικής βελτιστοποίησης `fmincon`. Στο Σχήμα 2.7.2 φαίνεται ποια συνάρτηση καλεί ποια και με ποια σειρά. Η χρονική σειρά είναι από πάνω προς τα κάτω.



Σχήμα 2.7.1 Διάγραμμα Ροής



Σχήμα 2.7.2 Διάγραμμα κλήσεων συναρτήσεων

## Κεφάλαιο 3<sup>ο</sup>

### Ανάλυση και Σχεδιασμός Δικτυώματος Ιπτάμενου Χειριστή

#### 3.1 Γενικά

Το επόμενο βήμα της παρούσας διπλωματικής εργασίας μετά την εύρεση των θέσεων και των προσανατολισμών των κινητήρων, είναι η εύρεση του τρόπου σύνδεσής τους στο χώρο. Για να διατηρηθεί το βάρος της κατασκευής σε χαμηλά επίπεδα υιοθετήθηκε η μέθοδος του δικτυώματος. Αρχικά γίνεται μια εισαγωγή στην ανάλυση δικτυωμάτων όπου παρατίθενται οι βασικές αρχές που διέπουν τα χαρακτηριστικά ενός τρισδιάστατου δικτυώματος. Έπειτα παρουσιάζονται δύο μεθοδολογίες εύρεσης κατάλληλου δικτυώματος, μια με χρήση των πακέτων Matlab και Ansys APDL στην οποία γίνεται βελτιστοποίηση, και μια με χρήση μιας μεθόδου εύρεσης του κατάλληλου δικτυώματος στο οποίο δεν ασκούνται θλιπτικές τάσεις στις ράβδους του.

#### 3.2 Εισαγωγή στην Ανάλυση Δικτυωμάτων

Ένα δικτύωμα αποτελείται από δυο στοιχεία, τα μέλη και τις συνδέσεις. Τα μέλη συνδέονται μεταξύ τους στα άκρα τους μέσω των συνδέσεων. Σε ένα δικτύωμα όλα τα φορτία και οι αντιδράσεις εφαρμόζονται μόνο στις συνδέσεις. Κάθε μέλος μπορεί δέχεται μόνο δυο ειδών φορτία εφελκυστικά ή θλιπτικά καθώς οι συνδέσεις στα άκρα τους είναι αρθρώσεις.

Σε ένα δικτύωμα τα μέλη του μπορούν να πάρουν διάφορες κατευθύνσεις, γι' αυτό είναι αναγκαίο να ορίσουμε τα τοπικά συστήματα συντεταγμένων κάθε μέλους και το ολικό σύστημα συντεταγμένων. Στο τοπικό σύστημα συντεταγμένων το μέλος κείται επάνω στον άξονα x και οι κόμβοι του αριθμούνται 1 και 2. Στο ολικό σύστημα συντεταγμένων κάθε σύνδεση έχει 3 βαθμούς ελευθερίας.

Έστω το διάνυσμα μετατόπισης ενός κόμβου στο τοπικό σύστημα συντεταγμένων όπως φαίνεται στο σχήμα 3.1.1

$$d' = [d'_1, d'_2]^T \quad (3.1)$$

Αντίστοιχα στο ολικό σύστημα συντεταγμένων το διάνυσμα μετατόπισης θα είναι

$$d = [d_1, d_2, d_3, d_4, d_5, d_6]^T \quad (3.2)$$

Γνωρίζουμε ότι η σχέση που συνδέει τις μετατοπίσεις είναι

$$\begin{cases} d'_1 = ld_1 + md_2 + nd_3 \\ d'_2 = ld_4 + md_5 + nd_6 \end{cases} \quad (3.3)$$

Όπου τα  $l = \cos(x', X)$ ,  $m = \cos(x', Y)$ ,  $n = \cos(x', Z)$  αποτελούν τα συνημίτονα κατευθύνσεις στον τοπικό άξονα  $x'$  ως προς τους ολικούς άξονες  $x, y$  και  $z$  αντίστοιχα.

Έτσι ο μετασχηματισμός τοπικών-ολικών συντεταγμένων είναι

$$d' = L * d \quad (3.4)$$

όπου το μητρώο μετασχηματισμού δίνεται από τη σχέση

$$L = \begin{bmatrix} l & m & n & 0 & 0 & 0 \\ 0 & 0 & 0 & l & m & n \end{bmatrix} \quad (3.5)$$

Το στοιχείο δικτυώματος είναι ένα μονοδιάστατο στοιχείο εάν θεωρηθεί σε ένα τοπικό σύστημα συντεταγμένων. Έτσι το μητρώο ακαμψίας του στοιχείου στο τοπικό σύστημα συντεταγμένων μπορεί να εκφραστεί ως

$$k' = \frac{E_e A_e}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3.6)$$

όπου  $E_e$  το μέτρο ελαστικότητας του υλικού,  $A_e$  το εμβαδόν της διατομής και  $l_e$  το μήκος του στοιχείου. Η έκφραση της ενέργειας παραμόρφωσης στο τοπικό σύστημα συντεταγμένων είναι

$$U_e = \frac{1}{2} d'^T k' d' \quad (3.7)$$

Αν αντικαταστήσουμε στην (3.7) την (3.4)

$$U_e = \frac{1}{2} d^T [L^T k' L] d \quad (3.8)$$

Η ενέργεια παραμόρφωσης σε ολικές συντεταγμένες μπορεί να γραφτεί ως

$$U_e = \frac{1}{2} d^T k d \quad (3.9)$$

Όπου  $k$  το μητρώο ακαμψίας σε ολικές συντεταγμένες

$$k = L^T k' L \quad (3.10)$$

Αντικαθιστώντας το  $\mathbf{L}$  παίρνουμε

$$\mathbf{k} = \frac{E_e A_e}{l_e} \begin{bmatrix} l^2 & lm & ln & -l^2 & -lm & -ln \\ lm & m^2 & mn & -lm & -m^2 & -mn \\ ln & mn & n^2 & -ln & -mn & -n^2 \\ -l^2 & -lm & -ln & l^2 & lm & ln \\ -lm & -m^2 & -mn & lm & m^2 & mn \\ -ln & -mn & -n^2 & ln & mn & n^2 \end{bmatrix} \quad (3.11)$$

### 3.3 Ορισμός του Προβλήματος

Το δικτύωμα που θα συνθέτει την κατασκευή του AM πρέπει να πληροί τα εξής χαρακτηριστικά:

- Να είναι όσο το δυνατόν πιο ελαφρύ, ούτως ώστε να διευκολύνει στη πτήση του AM.
- Να είναι όσο το δυνατόν πιο στιβαρό, ούτως ώστε να μπορεί να παραλαμβάνει τις δυνάμεις που ασκούνται, και να αποφεύγονται μικρές παραμορφώσεις στο σκελετό με αποτέλεσμα μετατοπίσεις στους κινητήρες και δημιουργία δυναμικών ταλαντώσεων αλλά και απώλεια ελέγχου του AM.

Τα πλεονεκτήματα του δικτύωματος τα οποία μας οδήγησαν να επιλέξουμε αυτό τον τρόπο κατασκευής έναντι άλλων είναι τα εξής:

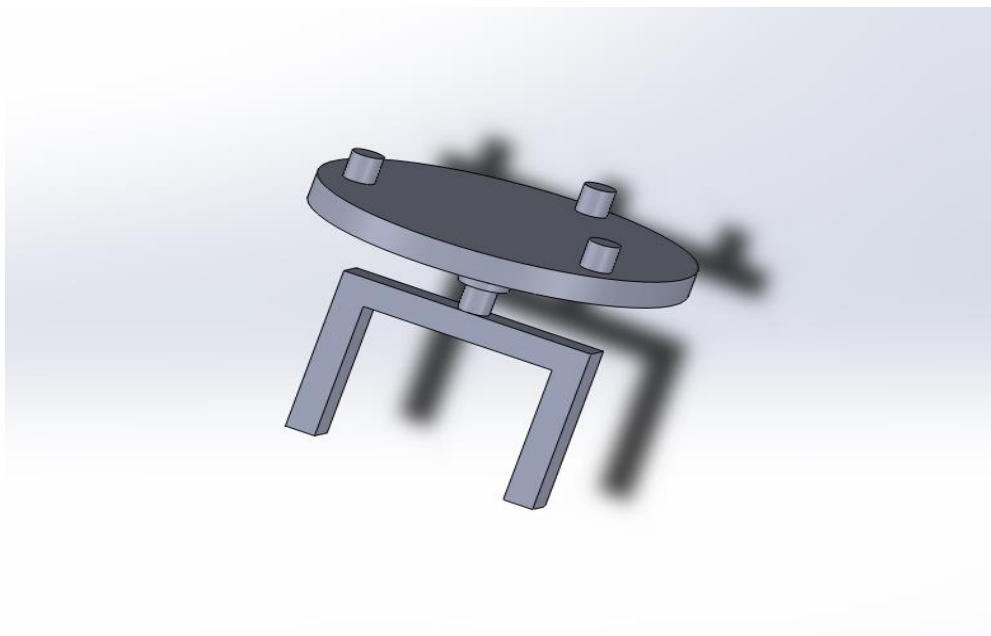
- Εύκολη κατασκευή και συναρμολόγηση, δεν απαιτεί περίπλοκες κατεργασίες.
- Με χρήση σύνθετων υλικών όπως ανθρακονήματα το δικτύωμα θα είναι πολύ ελαφρύ, κάτι το οποίο είναι ένα από τα ζητούμενα χαρακτηριστικά.
- Το κλάσμα στιβαρότητας-βάρους είναι μεγαλύτερο από άλλες επιλογές κατασκευής.
- Μπορούμε εύκολα να πετύχουμε περίπλοκες μορφές κατασκευής, κάτι που είναι ιδιαίτερα χρήσιμο διότι η μορφή του Aerial Manipulator δεν μας είναι γνωστή, και θέλουμε οι μεθοδολογία να προσαρμόζεται στα διάφορα αποτελέσματα.

Για να μπορέσουμε να σχεδιάσουμε ένα τρισδιάστατο δικτύωμα που να είναι κατάλληλο για να υλοποιήσει το Aerial Manipulator σύμφωνα με τις θέσεις που έχουν προκύψει από την βελτιστοποίηση, θα πρέπει να κάνουμε την εξής θεώρηση:

- Οι θέσεις των κινητήρων έτσι όπως έχουν προκύψει αποτελούν κόμβους του δικτύωματος.
- κόμβος του δικτύωματος θεωρείται και το κουτί το οποίο περιέχει τα απαραίτητα ηλεκτρονικά στοιχεία της κατασκευής

Επομένως έχουμε  $n + 1$  κόμβους, όπου  $n$  ο αριθμός των κινητήρων.

Εκτός από αυτούς τους κόμβους έχουμε και τον end-effector, όπως θα δούμε στη συνέχεια ο end-effector θα εδράζεται πάνω σε μία στρογγυλή πλάκα η οποία θεωρείται ένας υπερκόμβος διότι θα έχει 3 σημεία στήριξης.



**Σχήμα 3.3.1** Προσεγγιστική απεικόνιση του end-effector, με τους απλοποιημένους κόμβους

Στην σχήμα 3.2.1 φαίνεται ποια θα είναι η μορφή του end-effector. Βλέπουμε την κυκλική πλάκα στην οποία εδράζεται ο end-effector και παρατηρούμε τις 3 υποδοχές πάνω σε αυτή την πλάκα. Η πλάκα θα έχει 3 κόμβους διότι αυτός είναι ο ελάχιστος αριθμός σημείων έδρασης που χρειάζονται για να μην έχουμε μηχανισμό στον τρισδιάστατο χώρο.

Επομένως οι συνολικοί κόμβοι που θα έχουμε στον σχεδιασμό θα είναι:



$$Nodes = n + 1 + 3 = n + 4$$

όπου  $n$  ο αριθμός των κινητήρων.

Στους κόμβους που έχουμε θεωρήσει θέλουμε να δημιουργήσουμε χωρικό δικτύωμα το οποίο να μας εξασφαλίζει ότι η κατασκευή μας δεν αποτελεί μηχανισμό. Επομένως να είναι άκαμπτο. Για να πετύχουμε την ακαμψία ορίζουμε αρχικό περιορισμό ότι σε κάθε κόμβο θα πρέπει κατ' ελάχιστο να δένουν 3 ράβδοι, διότι όπως αναφέραμε και παραπάνω είναι ο ελάχιστος αριθμός για να μην δημιουργείται μηχανισμός.

Επίσης θέλουμε να πετύχουμε το ελάχιστο συνολικό μήκος ράβδων διότι έτσι θα πετύχουμε μικρότερο βάρος.

Γίνεται άρα αντιληπτό ότι πρέπει να λύσουμε ένα πρόβλημα βελτιστοποίησης. Το πρόβλημα αυτό θα έχει την εξής μορφή:

$$\begin{aligned} & \underbrace{\text{minimize}}_x F(q, m) \\ & \text{s. t.} \quad lb \leq x \leq ub \end{aligned}$$

όπου  $lb = 0$  και  $ub = 1$

όπου  $q$  και  $m$  αναφέρονται στην μετατόπιση των κόμβων και στη συνολική μάζα της κατασκευής αντίστοιχα.

Δηλαδή έχουμε ένα πρόβλημα βελτιστοποίησης όπου η αντικειμενική συνάρτηση έχει δύο στόχους, την μετατόπιση των κόμβων και το συνολικό βάρος της κατασκευής.

Η αντικειμενική συνάρτηση μπορεί να γραφτεί στην εξής μορφή:

$$F(q, m) = w \cdot \|q\| + (1 - w) \cdot m$$

όπου  $q$  είναι ο πίνακας των μετατοπίσεων των κόμβων που θα εξηγήσουμε παρακάτω πως τον βρίσκουμε,  $m$  είναι η συνολική μάζα του δικτύωματος της κατασκευής, η οποία υπολογίζεται βρίσκοντας το συνολικό μήκος των ράβδων του δικτύωματος και πολλαπλασιάζοντας τη μάζα ανά μονάδα μήκους της ράβδου με το μήκος αυτό, και  $w$  είναι παράγοντας ή βάρος που καθορίζει τη βαρύτητα της κάθε μεταβλητής βελτιστοποίησης. Πρέπει να παρατηρήσουμε ότι ο υπολογισμός αυτός της μάζας του

δικτύωματος προϋποθέτει ότι το δικτύωμα έχει φτιαχτεί από ενός τύπου ράβδους (δηλαδή ίδιων γεωμετρικών και υλικών χαρακτηριστικών).

### 3.4 Επίλυση του Προβλήματος

Για την επίλυση του ανωτέρω προβλήματος έχει αναπτυχθεί ένας κώδικας βελτιστοποίησης στο προγραμματιστικό περιβάλλον Matlab, όπου αλληλοεπιδρά με το λογισμικό ANSYS Apdl για να εξάγει τα δεδομένα που χαρακτηρίζουν την στιβαρότητα και την στατικότητα της κατασκευής.

#### 3.4.1 Γενετικός Αλγόριθμος

Όπως προαναφέραμε πρέπει να αναπτυχθεί ένας κώδικας βελτιστοποίησης ούτως ώστε να μπορέσουμε να εκπληρώσουμε τα κριτήρια που θέσαμε παραπάνω. Ο κώδικας έχει αναπτυχθεί στο περιβάλλον Matlab και χρησιμοποιεί εσωτερικές συναρτήσεις βελτιστοποίησης.

Συγκεκριμένα η βελτιστοποίηση θα γίνει με την μέθοδο των Γενετικών Αλγορίθμων (Genetic Algorithms). Αυτή η μέθοδος επιλέχθηκε γιατί έχει τα εξής πλεονεκτήματα:

- 1) Είναι Non-Gradient-Based αλγόριθμος κάτι που μας είναι απαραίτητο διότι το ένα σκέλος της αντικειμενικής συνάρτησης το οποίο αναφέρεται στις μετατοπίσεις των κόμβων, βγαίνει από την αλληλεπίδραση του κώδικα Matlab με το Ansys, κάτι που καθιστά αδύνατο τον υπολογισμό της παραγώγου της αντικειμενικής συνάρτησης.
- 2) Επίσης η μέθοδος των Γενετικών Αλγορίθμων λόγω του ότι δεν εξαρτάται από την παράγωγο της αντικειμενικής συνάρτησης, έχει πλεονέκτημα στα προβλήματα που είναι μη-γραμμικά, όπως στην προκειμένη περίπτωση.

#### Βήματα αλγορίθμου βελτιστοποίησης Matlab

##### I. Εισαγωγή Παραμέτρων

Σε πρώτη φάση καλούμε την συνάρτηση *truss3d.m* με είσοδο τον πίνακα  $x$   $[2n+1,3]$  ο οποίος περιέχει με τη σειρά τις θέσεις των κινητήρων στο χώρο, τους προσανατολισμούς τους, και τέλος την θέση του κουτιού που περιέχει τα ηλεκτρονικά.

## II. Ανάλυση Εισόδων και Παραγωγή Αρχείων για Ansys

Σε δεύτερη φάση, ο κώδικας ζητάει από τον χρήστη τον αριθμό των κινητήρων της συγκεκριμένης περίπτωσης. Αφού τον διαβάσει τότε χωρίζει την είσοδο που πήρε σε δύο πίνακες αυτόν των θέσεων και αυτόν των προσανατολισμών. Έπειτα τυπώνει αυτούς τους πίνακες σε αρχείο .txt (POSITION , ORIENTATION) τα οποία θα διαβάσει αργότερα το Ansys για να κάνει την ανάλυση του δικτυώματος.

## III. Συνάρτηση Βελτιστοποίησης

Στην τελική φάση ξεκινάει η βελτιστοποίηση. Αρχικά ξέροντας τον αριθμό των κινητήρων, βρίσκουμε τον αριθμό των πιθανών συνδυασμών των κόμβων ανά δύο από τον τύπο :

$$\binom{\nu}{k} = \frac{\nu!}{k! (\nu - k)!}$$

όπου,

$\nu$  : ο αριθμός των κόμβων

$k$ : ο αριθμός που δείχνει ανά πόσοι συνδέονται οι κόμβοι, στην περίπτωση μας ανά 2.

Έπειτα , φτιάχνουμε ένα πίνακα  $H [\nu,3]$  ο οποίος μας δίνει τον αύξοντα αριθμό του συνδυασμού και στις δυο τελευταίες στήλες μας δίνει τον συνδυασμό των κόμβων δηλαδή έχει αυτή την μορφή (για 7 κινητήρες):

$$H =$$

1	1	2
2	1	3
3	1	4
4	1	5
5	1	6
6	1	7
7	1	8
	⋮	
45	6	11
46	7	8
47	7	9
48	7	10
49	7	11

όπου αφαιρούνται πάντα οι τελευταίες 6 γραμμές διότι αντιστοιχούν σε συνδυασμούς μεταξύ κουτιού και end-effector οι οποίοι θα συνδέονται ούτως ή άλλως. Αυτόν τον πίνακα τον χρησιμοποιούμε σαν πίνακα αναφοράς για να μπορούμε εύκολα να ανατρέχουμε σε όποιο συνδυασμό θέλουμε.

### Περιορισμοί

Ένα βασικό χαρακτηριστικό είναι ότι θέλουμε οι μεταβλητές μας να παίρνουν τιμές 0 ή 1, αυτό επιτυγχάνεται εύκολα ορίζοντας ως κατώτατο όριο το 0 και ανώτατο το 1 και θέτοντας στη συνάρτηση βελτιστοποίησης μόνο ακέραιες τιμές για τις μεταβλητές σχεδιασμού.

Τέλος θέλουμε σε κάθε κόμβο να συνδέονται κατ' ελάχιστον 3 ράβδοι για λόγους στατικότητας. Αυτό επιτυγχάνεται χρησιμοποιώντας τον ανισωτικό περιορισμό  $A \cdot x \leq b$ , όπου  $A [n + 4, n]$  είναι ένας πίνακας που έχει στοιχεία 1 ή 0, και  $b [n + 4, 1]$  ένα διάνυσμα που κάθε στοιχείο του ισούται με 3. Συμπερασματικά ο πίνακας  $A$  περιέχει άσσους για κάθε πιθανό συνδυασμό κόμβων. Το διάνυσμα  $x [n + 4, 1]$  αποτελείται και αυτό από 1 ή 0 και όπου έχει άσσο ουσιαστικά “ενεργοποιεί” τον εκάστοτε συνδυασμό. Επομένως έστω ότι είχαμε την ισότητα  $A \cdot x = b$ , σύμφωνα με τα παραπάνω θα σήμαινε ότι κάθε κόμβος θα πρέπει να έχει “ενεργοποιημένους” 3 συνδυασμούς, άρα να

συνδέεται με 3 κόμβους. Επειδή όμως εμείς θέλουμε να είναι τουλάχιστον 3 και επειδή η συνάρτηση βελτιστοποίησης που χρησιμοποιούμε στη matlab υποστηρίζει περιορισμούς της μορφής  $A \cdot x \leq b$ , πολλαπλασιάζουμε τους A και b με -1, ώστε να επιτύχουμε τον περιορισμό:

$$A \cdot x \geq b$$

#### *Αντικειμενική Συνάρτηση*

Έχοντας ολοκληρώσει τα παραπάνω, μπαίνουμε στο σκέλος όπου πραγματοποιείται η βελτιστοποίηση. Η συνάρτηση βελτιστοποίησης που χρησιμοποιούμε είναι η *ga* της matlab. Καλώντας αυτή τη συνάρτηση γράφουμε σε ένα αρχείο ELEMENT.txt το χρωμόσωμα  $x$ , ώστε να το διαβάσει μετά το Ansys και να κατασκευάσει το δικτύωμα το οποίο θα αναλύσει. Έπειτα γίνεται ο υπολογισμός της αντικειμενικής συνάρτησης η οποία αποτελείται από δύο μέλη : το πρώτο μέλος αφορά τις μετατοπίσεις των κόμβων ( που τις διαβάζουμε από την ανάλυση του Ansys ) και το δεύτερο μέλος αφορά τη μάζα του δικτύωματος, που βγαίνει ουσιαστικά συναρτήσει του μήκους του.

### 3.4.2 ANSYS Apdl

Όπως είπαμε και παραπάνω μας ενδιαφέρει να ξέρουμε τις μετατοπίσεις των κόμβων όταν υπόκεινται στη μέγιστη καταπόνηση, διότι έτσι μπορούμε να εκτιμήσουμε αν είναι στιβαρή η κατασκευή ή όχι. Αν υπάρχουν μεγάλες μετατοπίσεις σημαίνει ότι η κατασκευή δεν είναι αρκετά στιβαρή, αν όχι σημαίνει ότι μπορεί να αντέξει τις καταπονήσεις οι οποίες της ασκούνται.

Ακόμα στην συγκεκριμένη περίπτωση όπου τα στοιχεία σύνδεσης θα έχουν μεγάλο μήκος και πολύ μικρή διατομή (για ελαχιστοποίηση του βάρους) μπορεί το δικτύωμα να είναι ευάλωτο σε λυγισμό.

Έχοντας λάβει υπ' όψη τα παραπάνω συντάχθηκε ένας αλγόριθμος για το περιβάλλον Ansys Apdl, ο οποίος δημιουργεί ένα τρισδιάστατο δικτύωμα, εφαρμόζει δυνάμεις στους κόμβους, ορίζει περιορισμούς και επιλύει το πρόβλημα εξάγοντας δεδομένα που αφορούν στην αντοχή, στην στιβαρότητα, και στο λυγισμό.

## Βήματα αλγορίθμου APDL

### I. *Εισαγωγή Παραμέτρων.*

Αρχικά ο αλγόριθμος διαβάζει τις παραμέτρους του δικτυώματος προς υλοποίηση. Δηλαδή, διαβάζει τους πίνακες  $R, F, C$  που αποτελούν τις θέσεις των κόμβων, την κατεύθυνση των εφαρμοζόμενων δυνάμεων και το σύνολο των ενεργοποιημένων συνδυασμών κόμβων.

Ο πίνακας  $F$  είναι ουσιαστικά ο πίνακας που δίνει τις κατευθύνσεις των κινητήρων. Ο πίνακας  $C$  είναι ένας μονοδιάστατος πίνακας ο οποίος αποτελείται μόνο από 0 και 1 και ονομάζεται χρωμόσωμα, θα εξηγήσουμε παρακάτω πως εξάγεται. Αυτό που πρέπει να ξέρουμε εδώ είναι ότι ουσιαστικά έχει μήκος όσοι και οι υπαρκτοί συνδυασμοί κόμβων και κάθε στοιχείο του αντιστοιχεί σε κάποιο συνδυασμό. Για παράδειγμα το στοιχείο  $C(1)$  αντιστοιχεί στο συνδυασμό του κόμβου 1 και 2, αν το στοιχείο αυτό είναι 1 τότε οι κόμβοι αυτοί συνδέονται με μια ράβδο, αν είναι 0 τότε δε συνδέονται.

	1	2	3	4	5	6	7	8	9	10	11
1		0	0	1	1	0	1	1	0	1	0
2			1	0	0	1	0	0	1	1	0
3				1	1	0	1	0	1	1	0
4					1	1	0	0	0	1	0
5						1	0	1	1	0	0
6							0	0	0	1	0
7								0	1	1	0
8									1	1	1
9										1	1
10											1

*Πίνακας 3.4.1: Συνδιασμοί για  $n=7$  κινητήρες*

Ο Πίνακας 1 δίνει τους συνδιασμούς για μια κατασκευή με  $n=7$  κινητήρες.

Από τον πίνακα 3.3.1 αν πάρουμε τα στοιχεία από τα αριστερά προς τα δεξιά και το θεωρήσουμε σαν στήλη θα έχουμε το εξής χρωμόσωμα:

0	0	1	1	0	1	1	0	1	0	...	1	1	1	1
1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10	1-11	...	8-11	9-10	9-11	10-11


από όπου και βλέπουμε ποια είναι η κωδικοποίηση του χρωμοσώματος.

Παρατηρούμε ότι στις θέσεις από 8-9 μέχρι 10-11 είναι όλα τα στοιχεία 1 γιατί οι κόμβοι 9-11 είναι οι κόμβοι πάνω στην πλάκα του end-effector επομένως θεωρούνται διαρκώς ενωμένοι και ο κόμβος 8 που είναι το κουτί με τα ηλεκτρονικά για λόγους που θα εξηγήσουμε παρακάτω θέλουμε να συνδέεται με ράβδους με τον end-effector .

## II. Καθορισμός Ιδιοτήτων Δικτύωματος

Αφού έχουμε εισάγει τις παραμέτρους του δικτύωματος, πρέπει τώρα να το δημιουργήσουμε στο Ansys. Αρχικά πρέπει να καθορίσουμε τις ιδιότητες των υλικών. Ως τύπο ράβδου επιλέγουμε από την βιβλιοθήκη του Ansys APDL το στοιχείο pipe288, το οποίο αντικατοπτρίζει τις ράβδους που θα χρησιμοποιήσουμε οι οποίες θα είναι tubes από ανθρακόνημα. Ένα άλλο

πλεονέκτημα του pipe288 είναι ότι μπορεί να αναλυθεί για λυγισμό , σε αντίθεση με άλλα απλούστερα στοιχεία της βιβλιοθήκης του ADPL που δεν αναλύονται για λυγισμό.

PIPE Elements	Pictorials
<b>PIPE288</b> 3-D 2-Node Pipe 2 nodes 3-D space DOF: UX, UY, UZ, ROTX, ROTY, ROTZ	

**Σχήμα 3.4.1** Ιδιότητες στοιχείου PIPE288 από manual APDL

Επίσης πρέπει να ορίσουμε το μέτρο ελαστικότητας και το λόγο Poisson. Στη θεώρηση μας για απλότητα και ταχύτητα των υπολογισμών θεωρούμε τις ράβδους μας σαν ισότροπο υλικό και ορίζουμε σαν μέτρο ελαστικότητας εκείνο για την δυσμενέστερη κατεύθυνση του ανθρακονημάτινου tube μας.

### III. Ορισμός Κόμβων, Δυνάμεων και Περιορισμοί

Αφού ορίσουμε τις ιδιότητες των υλικών πρέπει να φτιάξουμε το δικτύωμα στο Ansys. Αρχικά παίρνουμε τον πίνακα R που περιέχει τις θέσεις των κόμβων και μέσω κατάλληλης εντολής στο APDL ορίζουμε τους κόμβους. Έπειτα παίρνουμε τον πίνακα F, ο οποίος είναι ένας πίνακας με μοναδιαία διανύσματα κατεύθυνσης, έπειτα πολλαπλασιάζουμε κάθε στοιχείο του πίνακα με την μέγιστη απόλυτη δύναμη που μπορούν να δώσουν οι κινητήρες. Δηλαδή

$$F_{max} = F \cdot 15 \text{ [N]}$$

Έτσι θα έχουμε ως καταπόνηση τη μέγιστη που μπορεί να υπάρξει σε κάθε κόμβο. Μετά μέσω της κατάλληλης εντολής στο Apdl αντιστοιχίζουμε το κάθε διάνυσμα δύναμης με τον κατάλληλο κόμβο.

Το Aerial Manipulator που θέλουμε να κατασκευάσουμε είναι ιπτάμενο αντικείμενο, επομένως δεν έχει κάπου έδραση. Εφόσον όμως θα αλληλεπιδρά με το περιβάλλον του μέσω του end-effector θα θεωρήσουμε τον end-effector ως πάκτωση δηλαδή ορίζουμε στο Apdl μηδέν βαθμούς ελευθερίας για τους 3 κόμβους του end-effector που θεωρήσαμε πιο πάνω. Επίσης στο κουτί δεν ασκείται καμία εξωτερική δύναμη και ξέρουμε ότι αναγκαστικά θα συνδέεται με τους τρεις κόμβους του end-effector ειδικά θα στεκόταν στον αέρα, γι'



αυτό το λόγο θεωρούμε και τον κόμβο του κουτιού σαν μια πάκτωση. Επομένως έχουμε 4 πακτώσεις, δηλαδή 4 κόμβους με μηδεν βαθμούς ελευθερίας.

#### IV. Επίλυση

Τέλος μένει να γίνει επίλυση του δικτύωματος από το Apdl και να πάρουμε τα αποτελέσματα που χρειαζόμαστε. Αυτό γίνεται μέσω της εντολής Solve του Apdl η οποία λύνει πλήρως το δίκτυωμα. Όταν τελειώσει η επίλυση παίρνουμε τον πίνακα q που περιέχει όλες τις μετατοπίσεις των κόμβων και το γράφουμε σε ένα αρχείο txt. Αυτό το αρχείο txt το οποίο ονομάζουμε Output\_apdl.txt διαβάζει μετά η Matlab και υπολογίζει την αντικειμενική συνάρτηση.

### 3.5 Παραγωγή δικτύωματος προσανατολισμένη στην αποτροπή λυγισμού.

Στην παρούσα υποενότητα παρουσιάζεται μια καινοτόμος μεθοδολογία υπολογισμού δικτύωματος που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Η μεθοδολογία είναι προσανατολισμένη στην αντιμετώπιση του φαινομένου του λυγισμού. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο υπάρχουν κάποιες προϋποθέσεις ώστε να είναι δυνατή η υιοθέτηση της συγκεκριμένης μεθοδολογίας για μια διάταξη. Η προϋπόθεση αυτή όπως θα δούμε και παρακάτω είναι οι κινητήρες να έχουν τέτοιο προσανατολισμό που το διάνυσμα της δύναμης που παράγουν να «κοιτάει» προς τα «έξω». Δηλαδή, η εφαρμοζόμενη δύναμη να ασκεί μόνο εφελκυστικές τάσεις στα στοιχεία του δικτύωματος με αποτέλεσμα να αποφεύγεται ο λυγισμός.

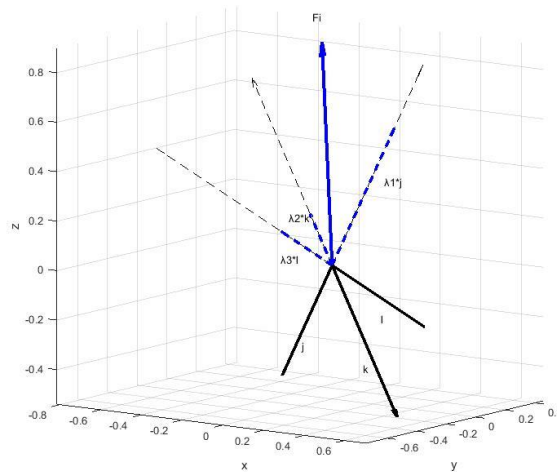
Θεωρούμε ότι η θέση κάθε κινητήρα είναι ένας κόμβος του δικτύωματος. Το πρόβλημα που θέλουμε να αντιμετωπίσουμε είναι ο προσδιορισμός του ελάχιστου αριθμού συνδέσμων ώστε να μην εμφανίζεται λυγισμός στους συνδέσμους. Γνωρίζουμε ότι ο ελάχιστος αριθμός συνδέσμων για τη επαρκή στήριξη ενός κόμβου στο χώρο είναι 3. Ποτέ όμως, ένας συνδυασμός 3 συνδέσμων είναι αποδεκτός. Ένας συνδυασμός είναι αποδεκτός όταν η δύναμη που παράγει η προπέλα ασκεί μόνο εφελκυστικές τάσεις στους συνδέσμους. Δηλαδή μια πρώτη συνθήκη που πρέπει να ικανοποιείται είναι η

κατεύθυνση της εφαρμοζόμενης δύναμης από την προπέλα να είναι προς τα έξω. Ας θεωρήσουμε τώρα ένα κόμβο  $i$  και τρεις ακόμα κόμβους  $j, k, l$  με τους οποίους θα συνδεθεί για να δημιουργηθούν οι τρεις σύνδεσμοι. Έχουμε λοιπόν τον συνδυασμό :

$$\sigma_m^i = \{(i, j), (i, k), (i, l)\}$$

Έστω τα διανύσματα  $\vec{j}, \vec{k}, \vec{l}$  που συνδέουν αντίστοιχα τους κόμβους  $j, k, l$  με τον κόμβο  $i$  και έχουν την αρχή τους στον κόμβο  $i$ . Θεωρούμε το μοναδιαίο διάνυσμα προσανατολισμού του κινητήρα :

$$\vec{F}_i = \begin{Bmatrix} F_i^x \\ F_i^y \\ F_i^z \end{Bmatrix}$$



**Σχήμα 3.5.1** Απεικόνιση του διανύσματος δύναμης πάνω στον κόμβο

Γράφοντας το μοναδιαίο διάνυσμα προσανατολισμού ως γραμμικό συνδυασμό των τριών διανυσμάτων των συνδέσμων παίρνουμε,

$$\vec{F}_i = \lambda_1 \vec{j} + \lambda_2 \vec{k} + \lambda_3 \vec{l}$$

και σε μητρωϊκή μορφή :

$$\begin{Bmatrix} F_i^x \\ F_i^y \\ F_i^z \end{Bmatrix} = \begin{bmatrix} j_x & k_x & l_x \\ j_y & k_y & l_y \\ j_z & k_z & l_z \end{bmatrix} * \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{Bmatrix}$$

η απαίτηση προσανατολισμού του κινητήρα (προς τα έξω) ικανοποιείται μόνο όταν οι συντελεστές των διανυσμάτων των συνδέσμων  $\{\lambda_1, \lambda_2, \lambda_3\}$  είναι αρνητικοί.

$$\lambda = I^{-1} * F_i \leq 0$$

Το πλήθος των συνδυασμών τριών συνδέσμων που υπάρχουν για έναν κόμβο προκύπτει από τη σχέση :  $\max m_i = \binom{n-1}{3} = \frac{n-1!}{(n-1-3)!3!}$  όπου  $n$  είναι ο αριθμός των κινητήρων. Αυτός ο αριθμός για τις τρεις περιπτώσεις 7,8 και 9 κινητήρων είναι 20,35 και 56 αντίστοιχα.

Θεωρούμε τη λίστα :

$$S_i = \{\sigma_m^i, \quad m = 1, \dots, m_i\}$$

που περιέχει όλους τους συνδυασμούς που ικανοποιούν τη συνθήκη, τότε ο μέγιστος αριθμός πιθανών συνδυασμών για  $n$  κόμβους προκύπτει από τη σχέση :

$$S = S_1 * S_2 * \dots * S_n \text{ (Cartesian product)}$$

Για 7, 8 και 9 κόμβους θα έχουμε αντίστοιχα :  $S \cong 128 * 10^7$ ,  $S \cong 2.25 * 10^{12}$ ,  $S \cong 5.42 * 10^{15}$ . Στις περισσότερες περιπτώσεις οι αριθμοί αυτοί είναι αρκετές τάξεις μεγέθους μικρότεροι.

Κάθε στοιχείο του  $S$  είναι ένας άνω-τριγωνικός πίνακας  $\Sigma_i$  ( $n \times n$ ), όπου  $n$  το πλήθος των κόμβων. Ο αριθμός κάθε στήλης και κάθε γραμμής αντιπροσωπεύει τον αντίστοιχο κόμβο. Ο πίνακας είναι ένα παράδειγμα με 7 κόμβους.

	1	2	3	4	5	6	7
1		0	1	1	1	1	0
2			0	0	1	1	1
3				1	0	0	1
4					0	0	2
5						1	1
6							0
7							

**Πίνακας 3.5.1** Παράδειγμα αρίθμησης κόμβων

Κάθε στοιχείο αυτού του πίνακα αποτελεί ένα σύνδεσμο μεταξύ δυο κόμβων. Για παράδειγμα αν ένα στοιχείο του πίνακα  $\Sigma_i$  έχει την τιμή μηδέν (0) αυτό σημαίνει ότι ο συγκεκριμένος σύνδεσμος δεν έχει επιλεγεί από κανένα συνδυασμό  $\sigma_m^i$ . Αν πάλι ένα στοιχείο έχει την τιμή ένα (1) τότε έχει επιλεγεί από ένα συνδυασμό μόνο, είναι φανερό ότι τα στοιχεία μπορούν να πάρουν μέχρι την τιμή δυο (2), αφού ένας σύνδεσμος

μπορεί να επιλεγεί μόνο από τους κόμβους στους οποίους ανήκει δηλαδή μόνο από δυο συνδυασμούς. Με αυτό τον τρόπο παράγονται όλοι οι δυνατοί συνδυασμοί συνδέσμων  $\Sigma_i$ .

Το επόμενο βήμα είναι η επιλογή της καλύτερης λύση  $\Sigma_i$  από αυτές που ικανοποιούν τη συνθήκη. Ένα κριτήριο είναι το βάρος, δηλαδή να επιλέξουμε τις λύσεις που έχουν τον ελάχιστο αριθμό συνδέσμων. Θα χρειαστεί λοιπόν να κατατάξουμε τις λύσεις σύμφωνα με τον αριθμό συνδέσμων που αυτές απαιτούν. Ο ελάχιστος αριθμός συνδέσμων για την περίπτωση των 7 κόμβων είναι 7 και ο μέγιστος  $\frac{n(n-1)}{2} = \frac{7(7-1)}{2} = 21$ . Έτσι, αυτός ο πίνακας λύσεων θα έχει 15 γραμμές και στις στήλες κάθε γραμμής θα υπάρχουν δείκτες που θα αναφέρονται στις λύσεις με το εκάστοτε πλήθος συνδέσμων. Ξεκινώντας από τις λύσεις με το μικρότερο πλήθος συνδέσμων θα εξετάσουμε περειαίρω χαρακτηριστικά όπως στιβαρότητα, αλληλεπίδραση συνδέσμων και προπελών, στηρίξεις επικίνδυνες ως προς την ταλάντωση, συνδεσιμότητα με τον επενεργητή και το κουτί με τα ηλεκτρονικά.

### 3.6 Επίλογος

Στο παρόν κεφάλαιο παρουσιάστηκαν δυο μεθοδολογίες παραγωγής δικτυώματος για τον εναέριο χειριστή. Η κάθε μεθοδολογία είχε το δικό της προσανατολισμό. Η πρώτη, που έγινε με τη χρήση του υπολογιστικού προγράμματος ANSYS, σκόπευε στην μείωση του βάρους και των μετατοπίσεων υπό την δυσμενέστερη περίπτωση φόρτισης. Η δεύτερη μεθοδολογία που αναπτύχθηκε για πρώτη φορά στην παρούσα εργασία σκόπευε στην αποφυγή λυγισμού κατά την εφαρμογή των φορτίων από τους κινητήρες. Όπως θα παρουσιαστεί και στο 4<sup>ο</sup> κεφάλαιο, έγιναν δύο ειδών τρεξίματα, είτε για ελεύθερο προσανατολισμό κώνων είτε για περιορισμένο (για λόγους που θα αναλυθούν εκτενέστερα στο επόμενο κεφάλαιο). Για κάθε τύπο τρεξίματος χρησιμοποιήθηκε διαφορετική μεθοδολογία παραγωγής δικτυώματος. Για αυτά με τον ελεύθερο προσανατολισμό χρησιμοποιήθηκε η μεθοδολογία Matlab-Ansys, ενώ για αυτά με τον περιορισμό στον προσανατολισμό χρησιμοποιήθηκε η μεθοδολογία εύρεσης δικτυώματος χωρίς θλιπτικές τάσεις στις ράβδους τους.

## Κεφάλαιο 4<sup>ο</sup>

### Αποτελέσματα Βελτιστοποίησης

#### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε τα αποτελέσματα των βελτιστοποιήσεων. Πρώτα τα αποτελέσματα της βελτιστοποίησης των θέσεων και των προσανατολισμών, και έπειτα αυτά της βελτιστοποίησης του δικτυώματος.

#### 4.2 Αποτελέσματα Πρώτης Βελτιστοποίησης

Ο αλγόριθμος βελτιστοποίησης των θέσεων και των προσανατολισμών, όπως παρουσιάστηκε στο κεφάλαιο 2 στην τελική του μορφή, εκτελέστηκε τουλάχιστον 1000 φορές για κάθε περίπτωση. Όσο μικρότερος ο αριθμός των κινητήρων, τόσο μικρότερος και ο υπολογιστικός χρόνος που απαιτείται. Γι' αυτό έχουμε περισσότερα αποτελέσματα στα τρεξίματα για 7 κινητήρες, λιγότερα για 8 και λιγότερα για 9 κινητήρες σε κάθε περίπτωση όμως δεν είναι λιγότερα από 1000. Οι εκτελέσεις του αλγορίθμου έγιναν στους ηλεκτρονικούς υπολογιστές του εργαστηρίου ηλεκτρονικών υπολογιστών της σχολής μηχανολόγων μηχανικών. Όπως επίσης και στον και στο σύστημα High Performance Computing του Κέντρου Ηλεκτρονικών Υπολογιστών του ΕΜΠ, όπου μας παρείχαν πολύ-πύρινο επεξεργαστή για τρέξιμο σε παράλληλο περιβάλλον.

Αρχικά τα αποτελέσματα θα παρουσιαστούν συνοπτικά με μορφή πινάκων για την άμεση σύγκριση των διαφόρων αρχικών παραμέτρων. Οι οποίες είναι:

- Ο αριθμός των κινητήρων.
- Το μήκος του αεροδυναμικού κώνου.
- Η διάμετρος της προπέλας.
- Ο προσανατολισμός των κινητήρων.

Συνολικά έγιναν 18 τρεξίματα του κώδικα με διαφορετικές παραμέτρους το κάθε τρέξιμο. Όπως έχουμε αναφέρει, διερευνήθηκαν 3 πλήθη κινητήρων  $k=7,8,9$ . Για κάθε ένα εξ αυτών είχαμε δύο διαφορετικές περιπτώσεις ως προς τον προσανατολισμό των κινητήρων, i) ο προσανατολισμός να είναι ελεύθερος (μη περιορισμένος) και να προκύπτει από την βελτιστοποίηση, ii) ο προσανατολισμός των κινητήρων να είναι περιορισμένος ώστε το διάλυμα της δύναμης να κοιτάει προς τα «έξω» αυτό γίνεται με τον τρόπο που αναφέραμε

στη παράγραφο 2.6.2. Τέλος για κάθε μία από τις ανωτέρω περιπτώσεις υπάρχουν άλλες 3 υποπεριπτώσεις όπου αλλάζει το μήκος αεροδυναμικού κώνου, ώστε όταν μετέπειτα διερευνηθεί το κατά πόσο επηρεάζει η αλληλεπίδραση των ροών ανάλογα με την απόσταση από την προπέλα, να μπορεί να επιλεγεί η κατάλληλη περίπτωση. Τα μήκη αεροδυναμικού κώνου που δοκιμάστηκαν είναι 0.4, 0.6 και 0.9 m.

Επομένως καταλήξαμε στην εξής δομή των τρεξιμάτων:

Αριθμός Κινητήρων	Προσανατολισμός Κινητήρων	Μήκος Αεροδυναμικού Κώνου
7	Inner / Free	0.4 / 0.6 / 0.9 m
8	Inner / Free	0.4 / 0.6 / 0.9 m
9	Inner / Free	0.4 / 0.6 / 0.9 m

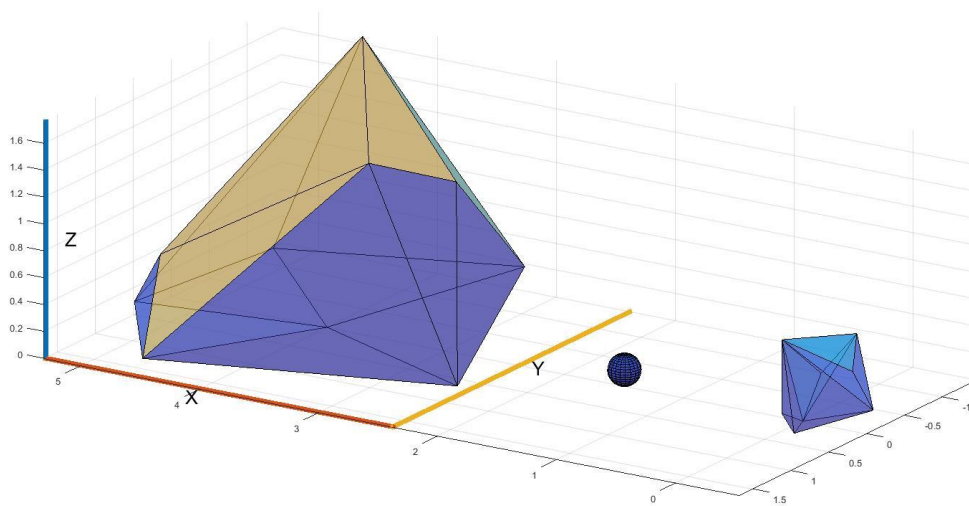
**Πίνακας 4.2.1** Περιπτώσεις τρεξιμάτων

Αρχικά (πίνακας 4.2.2) παραθέτουμε τα αποτελέσματα τα οποία έχουν επιλεγεί ως προς τον μικρότερο μέγιστο συντελεστή ώσης, οι οποίοι όμως έχουν προκύψει με τη μέθοδο αναδιανομής ώσης. Μετά στον πίνακα 4.2.3 παρουσιάζονται τα αποτελέσματα τα οποία έχουν επιλεγεί πάλι σύμφωνα με το μικρότερο μέγιστο συντελεστή ώσης, αλλά αυτή τη φορά οι ώσεις έχουν προκύψει από την μέθοδο των ελαχίστων τετραγώνων, όπως έχουμε αναλύσει στη παράγραφο 2.6.3. Στους ίδιους πίνακες δίπλα από τον συντελεστή ώσης βλέπουμε τις διαστάσεις της κατασκευής ως προς X,Y,Z καθώς και τον όγκο της. Επίσης βλέπουμε τον αριθμό της λύσης, δηλαδή σε ποιο αρχείο θα ανατρέξουμε για να την βρούμε. Ακόμα έχει προστεθεί ένας αύξων αριθμός σε κάθε λύση για την εύκολη αναφορά και εύρεσή της.

Διάμετρος Προπέλας (inches)	Αριθμός Κινητήρων	Προσανατολισμός Κινητήρων	Μήκος Αεροδυναμικού Κώνου (m)	Συντελεστής Ωσης	Διαστάσεις (m)			Όγκος (m <sup>3</sup> )	RUN	α/α
10	7	Inner	0.4	26.42	1.41	1.11	0.96	0.3642	333	1
			0.6	30.84	1.94	1.95	1	0.8290	191	2
			0.9	29.02	2.13	2.02	1.75	2.4300	505	3
		Free	0.4	22.58	0.67	0.88	0.53	0.0624	696	4
			0.6	22.94	0.72	0.96	0.56	0.0587	66	5
			0.9	22.54	0.78	0.80	0.59	0.0791	824	6
	8	Inner	0.4	27.57	1.53	1.68	0.96	0.7842	925	7
			0.6	27.37	2.20	1.87	0.98	1.0100	16	8
			0.9	28.69	3	2.64	1.63	3.9130	214	9
		Free	0.4	22.84	0.84	0.71	0.53	0.0859	748	10
			0.6	23.29	0.69	0.77	0.61	0.0914	108	11
			0.9	23.40	0.91	0.75	0.59	0.1195	149	12
	9	Inner	0.4	29.92	1.74	1.78	0.95	0.8701	26	13
			0.6	29.11	1.84	2.32	1.32	1.5844	64	14
			0.9	30.94	2.93	3.13	1.76	5.5000	62	15
		Free	0.4	23.43	1.13	0.99	0.54	0.1454	89	16
			0.6	23.95	1.20	1.02	0.75	0.1773	53	17
			0.9	24.51	1.18	0.83	0.53	0.1410	62	18

**Πίνακας 4.2.2:** Βέλτιστες λύσεις ως προς το συντελεστή ώσης υπολογισμένα με μέθοδο αναδιανομής ώσης.

Με μια πρώτη ματιά στον Πίνακα 4.2.2 παρατηρούμε ότι ο συντελεστής ώσης για κάθε αριθμό κινητήρων έχει μεγαλύτερες τιμές στον προσανατολισμό “inner” από ότι στο προσανατολισμό “free”. Το ίδιο ισχύει και για τον όγκο της κατασκευής. Έτσι, η καλύτερη λύση από όλες αυτές τις βέλτιστες λύσεις θα είναι μια με προσανατολισμό “free”. Όμως οι λύσεις με προσανατολισμό “inner” μας επιτρέπουν να χρησιμοποιήσουμε την πρωτότυπη μεθοδολογία παραγωγής δικτυώματος για αποφυγή λυγισμού και εκ φύσεως δεν έχουν το πρόβλημα του ανάστροφου εκκρεμούς. Παρατηρούμε ότι η καλύτερη λύση όσον αφορά το συντελεστή ώσης είναι η 6. Από τις λύσεις με προσανατολισμό inner η καλύτερη λύση είναι η 1.



**Σχήμα 4.2.1:** Συγκριτική αναπαράσταση των λύσεων (15 και 5) με το μεγαλύτερο και με το μικρότερο όγκο αντίστοιχα (ανάμεσά τους φαίνεται με σφαίρα με διαστάσεις αυτές μιας μπάλας μπάσκετ). Επίσης σημειώνονται οι μεταβλητές των διαστάσεων των λύσεων (X,Y,Z).

Διάμετρος Προπέλας (inches)	Μήκος Αεροδυναμικού Κώνου (m)	Προσανατολισμός Κινητήρων	Αριθμός Κινητήρων	Συντελεστής Ωσης	Διαστάσεις (m) X Y Z			Όγκος (m <sup>3</sup> )	RU N	α/α
10	7	Inner	04	19.66	1.79	1.69	0.94	0.9329	967	19
			0.6	20.27	2.03	2.45	1.26	1.2060	159	20
			0.9	22.04	2.13	2.02	1.75	2.43	505	21
		Free	04	16.57	0.76	0.90	0.62	0.0800	520	22
			0.6	15.33	0.58	1.04	0.53	0.0480	797	23
			0.9	17.13	0.78	0.80	0.59	0.0618	824	24
	8	Inner	04	16.91	1.87	1.89	0.97	2.7790	412	25
			0.6	17.80	2.02	2.31	1.28	1.6100	10	26
			0.9	20.70	3.21	2.54	1.73	4.4500	117	27
		Free	04	14.60	0.65	0.61	0.61	0.0830	67	28
			0.6	17.23	0.69	0.77	0.61	0.0910	108	29
			0.9	16.72	1.00	0.95	0.70	0.1660	327	30
	9	Inner	04	18.85	1.46	1.58	0.91	0.7524	24	31
			0.6	20.42	1.93	2.02	1.15	1.0222	80	32
			0.9	22.61	3.37	3.29	1.57	6.5078	63	33
		Free	04	17.32	1.13	0.99	0.54	0.1454	89	34
			0.6	16.32	0.55	0.89	0.83	0.1586	78	35
			0.9	21.77	1.18	0.83	0.53	0.1410	33	36

**Πίνακας 4.2.3:** Βέλτιστες λύσεις ως προς το συντελεστή ώσης ευρισκόμενο μέσω της μεθόδου ελαχίστων τετραγώνων με μη αρνητικές μεταβλητές.

Στο πίνακα 4.2.3 βλέπουμε αντίστοιχα τις βέλτιστες λύσεις όπως αυτές προκύπτουν από την μεθοδολογία των ελαχίστων τετραγώνων. Παρατηρούμε και εδώ όπως και πριν ότι οι λύσεις με προσανατολισμό "free" δίνουν καλύτερα αποτελέσματα. Συγκεκριμένα η καλύτερη λύση ως προς τον συντελεστή ώσης είναι η 28. Ακόμα, η καλύτερη λύση με προσανατολισμό "inner" είναι η 25 όμως έχει μεγάλο όγκος μια λύση που να συνδυάζει και τα δυο είναι η 31.

### 4.2.1 Βέλτιστες Λύσεις

Στην παρούσα υπό-ενότητα θα παρατεθούν όλες οι πληροφορίες των λύσεων που παρουσιάστηκαν προηγουμένως. Οι πληροφορίες αυτές είναι:

- Ο αριθμός των κινητήρων.
- Το μήκος του αεροδυναμικού κώνου.
- Ο προσανατολισμός των κινητήρων.
- Ο αύξων αριθμός της εκτέλεσης του αλγορίθμου βελτιστοποίησης.
- Ο συνδυασμός περιστροφής των προπελών που δίνει το μικρότερο συντελεστή ώσης.
- Ο συντελεστής ώσης.
- Ο όγκος της κατασκευής.
- Οι θέσεις και οι προσανατολισμοί των κινητήρων.
- Το χρωμόσωμα του δικτυώματος.



- Οι μετατοπίσεις των κόμβων που προέκυψαν για δύναμη 15N σε όλους τους κινητήρες

Να σημειωθεί ότι, οι συντεταγμένες θέσης έχουν αρχή των αξόνων (0,0,0) την βάση του end-effector. Επίσης τα διανύσματα των προσανατολισμών μπορούν να εκφρασθούν στο τοπικό σύστημα του κινητήρα, αν θεωρήσουμε ότι το τοπικό σύστημα του κινητήρα είναι μόνο μετατοπισμένο ως προς το σύστημα με αρχή τη βάση του end-effector, στο οποίο θα αναφερόμαστε και ως αδρανειακό σύστημα.

Όσον αφορά το χρωμόσωμα του βέλτιστου combination είναι ένα διάνυσμα αποτελούμενο από 0 και 1. Το 0 σημαίνει ότι το διάνυσμα της ροπής αντίδρασης της προπέλας έχει την ίδια φορά με το διάνυσμα της δύναμης, ενώ το 1 σημαίνει ότι το διάνυσμα της ροπής της δύναμης έχει αντίθετη φορά. Σε κάθε περίπτωση όμως η φορά της δύναμης δεν αλλάζει, αφού θα χρησιμοποιήσουμε αντίστοιχα η δεξιόστροφες ή αριστερόστροφες προπέλες.

Στο χρωμόσωμα του δικτυώματος, βλέπουμε ένα πίνακα που περιέχει μέσα το σύμβολο “+”, όπου υπάρχει αυτό το σύμβολο σημαίνει ότι στους αντίστοιχους κόμβους, που αναγράφονται στο πίνακα υπάρχει ράβδος που τα συνδέει. Για παράδειγμα αν στη θέση (4,6) του πίνακα υπάρχει το σύμβολο “+” τότε σημαίνει ότι ο κόμβος 4 θα ενώνεται με τον 6 με ράβδο.

Επίσης οι λύσεις inner δε περιλαμβάνουν μετατοπίσεις κόμβων γιατί δεν έχει γίνει τέτοιου είδους ανάλυση για τα δικτυώματά τους. Τα δικτυώματα των λύσεων inner έχουν βγει καθαρά και μόνο με την απαίτηση να μην ασκούνται θλιπτικές τάσεις στις ράβδους.

### Βέλτιστες Λύσεις με τη Μέθοδο Αναδιανομής Όσης

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)			
1	7	0.4	Inner		333	26.42		0.3642			
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)				
			-0.2257		0.3195		0.1095				
			-0.2297		-0.3505		0.0788				
			0.7793		0.3099		0.2314				
			0.1357		-0.5273		0.1071				
			-0.6350		0.5849		0.6002				
			0.4962		-0.0880		0.6021				
			-0.2137		-0.3498		0.9665				
			-0.1453		0.7997		-0.5825				
			-0.1342		-0.6052		-0.7846				
			0.7112		0.6111		-0.3474				
			-0.1014		-0.9244		-0.3676				
			-0.7025		0.5037		0.5028				
			0.7222		-0.2103		0.6590				
			-0.3500		-0.1746		0.9204				
			0		0		0.3953				
<div>Truss links</div> <div>Number: 11</div> <div>Total length: 9.467 m</div>				Thrusters							
				1	2	3	4	5	6	7	
			Thrusters	1	+			+	+		
				2				+			+
				3				+	+	+	
				4						+	
				5							+
				6							+
7											

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)			
2	7	0.6	Inner		191	30.84		0.8290			
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)				
			-0.0286		0.9474		0.4600				
			-1.0760		-0.2347		0.3142				
			0.0315		-1.005		0.09292				
			-0.1972		-0.3864		1.0040				
			0.8673		0.3232		0.1361				
			0.6884		-0.2944		0.07735				
			-0.7126		0.4836		0.7823				
			0.08393		0.975		0.2058				
			-0.8950		-0.3981		-0.2012				
			-0.0745		-0.8843		-0.4610				
			-0.1180		-0.4268		0.8966				
			0.7716		0.4070		-0.4889				
			0.9088		-0.3652		-0.2017				
			-0.6767		0.6923		0.2504				
			0		0		0.4347				
<div>Truss links</div> <div>Number: 11</div> <div>Total length: 13.011 m</div>				Thrusters							
				1	2	3	4	5	6	7	
			Thrusters	1		+			+		+
				2			+			+	+
				3				+		+	
				4					+		+
				5						+	
				6							
7											

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)			
3	7	0.9	Inner		505	29.02		2.43			
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)				
			0.1626		-0.5238		0.05259				
			0.626		-0.3772		1.723				
			0.785		0.4251		1.726				
			-1.067		-0.7868		1.753				
			-0.1773		1.235		1.021				
			-0.6652		0.8792		0.07642				
			-1.349		-0.6835		0.4604				
			0.4286		-0.4841		-0.7628				
			0.841		-0.437		0.319				
			0.5109		0.2753		0.8143				
			-0.7847		-0.3487		0.5125				
			0.2029		0.9573		-0.2057				
			-0.5391		0.738		-0.406				
			-0.6598		-0.7008		-0.2713				
			0		0		1.145				
<div>Truss links</div> <div>Number: 11</div> <div>Total length: 17.49 m</div>				Thrusters							
				1	2	3	4	5	6	7	
			Thrusters	1		+	+			+	+
				2			+	+			
				3					+		
				4					+		+
				5						+	
				6							+
7											

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)							
4	7	0.4	Free		696	22.58		0.062							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)								
			0.4304		0.05969		0.53								
			0.3787		0.2758		0.1591								
			-0.2474		-0.3965		0.06783								
			0.07356		-0.5503		0.1793								
			0.1063		0.333		0.1537								
			0.3409		-0.4887		0.1372								
			0.3009		-0.1462		0.312								
			-0.6286		0.7726		-0.08903								
			-0.06298		0.01734		0.9979								
			0.3303		-0.2592		0.9076								
			0.3958		-0.686		-0.6105								
			0.8539		0.4835		-0.1924								
			-0.7896		-0.6107		-0.05924								
			-0.09878		0.2824		-0.9542								
			0		0		0.7439								
<div>Truss links</div> <div>Number: 29</div>					Thrusters				Box	End-effector					
					1	2	3	4	5	6	7	8	9	10	11
			Thrusters	1		+	+			+	+	+	+	+	+
				2						+	+	+			
				3				+	+		+			+	
				4						+			+		+
				5						+	+	+	+		+
				6							+	+	+	+	+
				7										+	
<div>Nodal Displacement</div>			X(m)		Y(m)				Z(m)						
			0.002		-0.0014				-0.00002						
			0.0008		-0.0012				-0.00005						
			-0.0004		0.0002				0.00001						
			-0.0019		-0.0033				-0.0019						
			0.0011		-0.0004				0.00001						
			-0.0018		-0.0012				-0.00004						
			-0.00016		-0.001				-0.0002						

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration				Run	Objective Function				Volume (m³)			
5	7	0.6	Free				66	22.94				0.059			
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)				Y (m)				Z (m)				
			-0.2021				0.4339				0.06371				
			0.03092				0.2286				0.1146				
			-0.5301				-0.09575				0.147				
			0.002684				-0.2446				0.2888				
			0.1909				-0.02434				0.5656				
			-0.116				-0.5291				0.5267				
			-0.2788				-0.2178				0.3579				
			0.7401				-0.3275				-0.5874				
			0.2249				0.6003				0.7675				
			-0.01141				-0.7597				0.6502				
			-0.822				-0.5492				0.1503				
			-0.588				0.6399				-0.4947				
			0.249				-0.5256				-0.8135				
			0.2075				0.9218				0.3276				
			0				0				0.2399				
<div>Truss links</div> <div>Number: 31</div>					Thrusters							Box	End-effector		
					1	2	3	4	5	6	7	8	9	10	11
			Thrusters	1		+	+	+	+			+	+	+	
				2			+	+	+		+	+		+	+
				3					+		+			+	
				4							+	+			
				5						+	+	+	+	+	
				6							+	+	+	+	+
				7							+		+	+	+
<div>Nodal Displacement</div>			X(m)				Y(m)				Z(m)				
			0.0019				0.001				-0.00007				
			0.0009				-0.0012				-0.00008				
			-0.0037				0.0021				0.00068				
			-0.001				-0.001				-0.00001				
			-0.0008				-0.0005				0.00006				
			-0.0011				-0.00012				-0.00043				
			-0.0011				0.0015				-9e-6				

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)							
6	7	0.9	Free		824	22.54		0.079							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)								
			0.2981		0.02147		0.2593								
			0.3893		-0.3967		0.4814								
			-0.2649		-0.2606		0.1816								
			-0.3942		-0.4733		0.3071								
			0.07745		0.2476		0.5915								
			0.01338		0.3279		0.1848								
			0.3355		-0.1194		0.5079								
			-0.2535		0.9078		0.3341								
			-0.7234		-0.1521		-0.6734								
			-0.2153		-0.4475		0.868								
			0.6184		-0.7209		-0.3127								
			0.5727		0.3913		-0.7203								
			0.9634		0.1056		0.2463								
			-0.9625		-0.08412		0.2581								
			0		0		0.2245								
<div>Truss links</div> <div>Number: 26</div>				Thrusters							Box	End-effector			
				1	2	3	4	5	6	7	8	9	10	11	
			Thrusters	1				+			+	+	+	+	+
				2				+		+			+	+	+
				3				+				+	+	+	
				4								+	+		+
				5						+	+	+	+		
				6										+	+
7									+	+					
<div>Nodal Displacement</div>			X(m)		Y(m)		Z(m)								
			-0.001		0.0015		-0.00005								
			0.0064		0.0017		-0.0038								
			0.003		-0.0031		0.0001								
			0.0066		-0.0059		-0.001								
			0.0005		0.002		-0.0018								
			-0.00004		0.0006		-0.001								
			0.00023		0.0019		0.0004								

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration	Run	Objective Function	Volume (m³)						
7	8	0.4	Inner	925	27.57	0.78						
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)					
			0.07426		-0.4044		0.07764					
			0.4923		-0.05045		0.09406					
			-0.9219		0.3802		0.93					
			0.6096		0.4113		0.9449					
			0.2237		-0.2327		0.9666					
			-0.8079		0.7106		0.1017					
			-0.1415		-0.9767		0.164					
			-0.1623		0.4627		0.2296					
			0.4098		-0.7735		-0.4835					
			0.7297		0.07823		-0.6792					
			-0.8028		0.07114		0.592					
			0.6532		0.195		0.7316					
			0.4671		-0.1296		0.8746					
			-0.5459		0.7027		-0.4563					
			-0.4167		-0.8973		-0.1457					
			-0.4946		0.7533		-0.4336					
			0		0		0.3853					
			<div>Truss links</div> <div>Number: 13</div> <div>Total length: 12.427</div>					Thrusters				
1	2	3						4	5	6	7	8
Thrusters	1					+				+	+	
	2							+			+	+
	3								+	+		+
	4								+			+
	5										+	
	6											+
	7											
	8											



[illegible]

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function			Volume (m³)			
9	8	0.9	Inner		214	28.7			3.91			
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)			Y (m)			Z (m)			
			-0.6516			0.9761			1.633			
			1.088			-0.382			1.461			
			-1.556			-1.18			1.509			
			1.447			0.5732			0.1748			
			-0.8876			-1.192			0.3356			
			0.8042			-0.01545			0.1956			
			-0.2817			1.452			1.334			
			-1.113			-0.668			0.07819			
			-0.1023			0.7075			0.6993			
			0.6281			-0.3143			0.7118			
			-0.4863			-0.6413			0.5936			
			0.842			0.04227			-0.5378			
			-0.7215			-0.5102			-0.4681			
			0.8288			0.31			-0.4659			
			-0.4735			0.859			0.1947			
			-0.5152			-0.453			-0.7276			
			0			0			0.7603			
			<div>Truss links</div> <div>Number: 14</div> <div>Total length: 24.217</div>					Thrusters				
1	2	3						4	5	6	7	8
Thrusters	1						+	+			+	+
	2							+	+	+	+	
	3								+			+
	4									+		
	5											+
	6										+	+
	7											
	8											

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function				Volume (m³)									
10	8	0.4	Free		748	22.84				0.0859									
Thrusters Positions			X (m)		Y (m)				Z (m)										
			-0.2141		0.2214				0.2696										
			-0.08845		-0.2853				0.1164										
			0.1439		0.234				0.2493										
			0.5272		-0.3196				0.3549										
			0.3698		0.3978				0.1407										
			0.3128		-0.2464				0.5235										
			-0.1003		-0.1589				0.537										
			-0.3219		-0.1106				0.1879										
			-0.5248		-0.8037				0.2804										
Thrusters Orientations			-0.9838		0.133				-0.12										
			0.4955		0.08755				0.8642										
			-0.5409		0.7322				0.4139										
			0.402		-0.8645				-0.3018										
			0.4849		0.7151				-0.5034										
			-0.2707		-0.1346				-0.9532										
			0.9378		0.1348				0.3199										
			0		0				0.3641										
Box Position			0		0				0.3641										
			Truss links Number: 31					Thrusters								Box		End-effector	
								1	2	3	4	5	6	7	8	9	10	11	12
						1			+	+	+		+		+	+	+		
						2						+		+	+		+		
						3						+		+	+		+	+	
						4						+			+	+		+	
						5						+				+	+		
						6										+	+	+	
						7									+		+	+	
8													+		+				
Nodal Displacement			X(m)		Y(m)				Z(m)										
			0.0024		0.0024				1.4e-5										
			-0.0024		0.008				0.00033										
			0.0026		-0.0016				9e-5										
			0.0003		0.00012				-0.00027										
			0.0043		-0.0039				-0.00042										
			-0.0014		-0.0036				-0.0009										
			-0.0019		0.001				-0.00013										
			-0.00086		0.0029				0.00033										

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function	Volume (m³)									
11	8	0.6	Free		108	23.29	0.0914									
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)			Z (m)								
			0.1158		-0.2865			0.06075								
			0.202		0.3831			0.09372								
			-0.1547		-0.3242			0.5687								
			0.2679		0.006116			0.127								
			-0.1004		0.3036			0.3283								
			0.218		0.4519			0.4141								
			-0.4239		0.08137			0.6155								
			-0.2736		-0.2345			0.2342								
			-0.9253		0.3724			0.07152								
			-0.3028		-0.9526			0.02857								
			-0.3596		0.2031			-0.9107								
			-0.05399		0.3725			0.9265								
			0.9802		-0.05535			0.1902								
			0.01493		0.7218			-0.692								
			-0.03467		-0.9584			-0.2833								
			0.6813		0.2966			0.6692								
			0		0			0.4096								
<div>Truss links</div> <div>Number: 31</div>					Thrusters					Box	End-effector					
					1	2	3	4	5	6	7	8	9	10	11	12
			Thrusters	1		+		+						+	+	
				2							+	+		+	+	+
				3						+		+			+	+
				4					+	+	+			+		
				5									+	+		+
				6							+		+	+	+	+
				7										+	+	+
				8									+	+	+	
<div>Nodal Displacement</div>			X(m)		Y(m)			Z(m)								
			0,0028		0.0012			0.00004								
			-0.0039		0.0018			0.00013								
			0.0023		-0.0013			-0.0001								
			-0.0086		0.0037			0.0005								
			-0.0043		-0.0014			0.00005								
			-0.0059		0.0027			0.0002								
			-0.011		-0.0072			0.0018								
			0.0018		-0.0021			0.00001								

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)								
12	8	0.9	Free		149	23.4		0.1195								
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)									
			0.07718		0.1566		0.5775									
			-0.2212		0.4458		0.1419									
			0.1293		-0.2427		0.2541									
			-0.4173		-0.305		0.4432									
			0.4283		-0.05156		0.09393									
			-0.2147		0.1745		0.5647									
			0.4954		-0.2107		0.4919									
			0.1956		-0.1949		0.595									
			-0.2769		-0.8253		0.4922									
			0.4032		-0.9116		-0.0801									
			-0.5773		0.8069		-0.125									
			0.5146		-0.1261		-0.8481									
			-0.6996		-0.2327		-0.6756									
			0.8511		0.3602		0.382									
			-0.8201		0.1984		0.5368									
			0.6049		0.7302		0.3178									
			0		0		0.2992									
<div>Truss links</div> <div>Number: 33</div>					Thrusters								Box	End-effector		
					1	2	3	4	5	6	7	8	9	10	11	12
			Thrusters	1			+	+					+	+		+
				2					+		+			+	+	+
				3							+	+	+	+		+
				4					+				+		+	+
				5						+			+			
				6								+		+		+
				7								+	+	+	+	+
8										+	+	+	+			
<div>Nodal Displacement</div>			X(m)		Y(m)				Z(m)							
			0.0101		-0.0548				0.0013							
			0.0286		0.0140				0.0005							
			-0.01716		-0.0092				0.001							
			-0.0240		0.032				-0.00084							
			-0.0429		-0.0311				0.0038							
			0.0381		0.0433				0.0010							
			-0.0146		-0.0340				1.6e-5							
			-0.0134		-0.0138				2.9e-5							

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration	Run	Objective Function	Volume (m³)						
13	9	0.4	Inner	26	29.92	0.8701						
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)					
			0.1178		-0.1966		0.09261					
			0.4854		0.3742		0.2291					
			0.0733		0.3783		0.08034					
			-0.6495		0.0125		0.5294					
			0.4068		-0.1274		0.1275					
			-0.158		0.5787		0.1433					
			-0.1814		0.09439		0.3872					
			0.1128		-0.4084		0.5419					
			-0.2552		-0.417		0.386					
			0.1109		-0.292		-0.95					
			0.3548		0.7647		-0.5379					
			-0.6426		0.6061		0.4688					
			-0.5383		0.627		-0.5631					
			-0.9144		-0.3796		0.141					
			0.8157		0.4239		0.3937					
			0.08328		-0.9894		-0.1188					
			0.2207		0.07737		0.9723					
			0.51		-0.838		0.1939					
			0		0		0.5964					
<div>Truss links</div> <div>Number: 15</div> <div>Total length: 14.542</div>				Thrusters								
				1	2	3	4	5	6	7	8	9
			Thrusters	1	+	+				+		+
				2							+	+
				3				+	+	+		
				4						+	+	+
				5							+	+
				6								+
				7								
				8								
				9								







Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration				Run		Objective Function				Volume (m³)				
16	9	0.4	Free				89		23.43				0.1454				
Thrusters Positions			X (m)				Y (m)				Z (m)						
			-0.1522				-0.6685				0.268						
			0.5879				-0.383				0.5604						
			-0.2064				0.5896				0.4321						
			-0.1357				0.1603				0.9539						
			-0.6418				0.3365				0.08625						
			0.7976				0.9339				0.3444						
			-0.9454				-0.8506				0.7287						
			0.7125				0.4582				0.05424						
			0.1075				-0.6694				0.8735						
Thrusters Orientations			-0.3079				-0.7898				-0.5304						
			0.8575				-0.4223				0.2937						
			-0.3314				0.9359				0.1193						
			-0.5346				0.2931				0.7926						
			-0.6425				0.2826				-0.7123						
			0.8065				0.5853				-0.08359						
			-0.7186				-0.5747				0.3915						
			0.5535				0.5804				-0.5974						
			0.3176				-0.8903				0.3265						
			0				0				0.5148						
Box Position			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
			0				0				0.5148						
Truss links Number: 36				Thrusters									Box	End-effector			
				1	2	3	4	5	6	7	8	9		10	11	12	13
			Thrusters	1		+				+	+	+	+		+	+	
				2			+			+				+		+	
				3							+	+					+
				4					+	+	+		+	+			
				5							+		+	+	+	+	+
				6							+	+				+	+
				7										+			
				8									+		+	+	
9											+	+		+			
Nodal Displacement			X(m)				Y(m)				Z(m)						
			0.0024				0.0012				0.00007						
			-0.0041				0.0053				-0.00021						
			-0.40				0.0088				-0.0003						
			-0.0014				-0.00075				-0.0002						
			0.0001				0.0046				-1.34e-6						
			-0.0064				-0.0017				0.00015						
			-0.0109				-0.002				0.00016						
			0.0055				0.0013				-0.00019						
			0.0048				-0.0003				3.2e-5						

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration			Run	Objective Function				Volume (m³)							
17	9	0.6	Free			53	23.95				0.1773							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)			Y (m)				Z (m)								
			-1.201			-0.7642				0.05493								
			0.6332			-0.7825				0.8711								
			0.5785			0.3178				0.07799								
			0.02217			0.6864				0.1181								
			0.3284			-1.066				0.3051								
			-0.9932			-0.2614				0.3828								
			0.6456			-0.2995				1.325								
			-0.0933			0.6486				1.192								
			-0.4455			1.257				0.1789								
			-0.9175			-0.3913				-0.07199								
			0.4577			-0.6383				0.6189								
			0.5959			0.4377				-0.6733								
			0.2685			0.8688				-0.416								
			0.4557			-0.847				-0.2737								
			-0.9061			-0.3044				-0.2936								
			0.6228			-0.456				0.6358								
			-0.1586			0.5034				0.8494								
			-0.4184			0.8271				-0.3754								
			0			0				0.4535								
<div>Truss links</div> <div>Number: 36</div>					Thrusters									Box	End-effector			
					1	2	3	4	5	6	7	8	9	10	11	12	13	
			Thrusters	1			+	+		+	+							
				2			+				+					+	+	
				3				+					+		+			
				4					+				+					
				5							+		+	+	+	+		
				6							+		+	+		+		
				7								+	+		+	+		
				8									+	+	+	+		
9											+	+		+				
<div>Nodal Displacement</div>			X(m)			Y(m)				Z(m)								
			-0.0230			0.0034				-0.0009								
			-0.0235			-0.0196				-0.0002								
			0.0097			-0.0179				0.0002								
			0.020			-0.0061				0.0009								
			-0.0031			-0.0097				-5.1e-5								
			-0.0078			0.0297				-0.00089								
			-0.0087			-0.0185				0.0006								
			0.0198			0.0286				-5.2e-5								
			0.0038			0.0136				3.7e-5								

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration				Run	Objective Function				Volume (m³)					
18	9	0.9	Free				62	24.51				0.141					
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)				Y (m)				Z (m)						
			-1.628				0.834				0.07561						
			-0.81				0.7187				1.559						
			0.4804				1.677				0.1441						
			-0.625				-0.7261				0.9926						
			1.309				0.4847				0.09633						
			1.166				-1.124				0.07685						
			0.9772				0.6634				0.5548						
			0.4545				-1.182				1.763						
			-0.7336				-1.457				0.1799						
			-0.8699				0.1838				-0.4577						
			-0.4969				0.4137				0.7628						
			0.3963				0.9132				-0.09494						
			-0.7807				-0.6141				0.116						
			0.8973				0.4102				-0.1629						
			0.4996				-0.6443				-0.5791						
			0.6722				0.6775				-0.2988						
			0.2476				-0.5216				0.8165						
			-0.5655				-0.8184				-0.102						
			0				0				0.7201						
<div>Truss links</div> <div>Number: 40</div>				Thrusters									Box	End-effector			
				1	2	3	4	5	6	7	8	9	10	11	12	13	
			Thrusters	1		+	+	+					+				
				2				+			+			+	+	+	
				3					+				+	+	+	+	+
				4						+			+	+	+	+	
				5							+	+	+	+		+	+
				6							+	+	+			+	+
				7								+	+	+	+		
				8										+			+
				9											+	+	
<div>Nodal Displacement</div>			X(m)				Y(m)				Z(m)						
			-0.0305				-0.0059				-0.0003						
			-0.0263				-0.0299				-0.0003						
			-0.0060				0.0173				1.5e-6						
			0.0266				-0.0228				0.0001						
			-0.176				0.0047				-6.5e-5						
			0.0044				0.0039				-0.0001						
			-0.0243				0.0359				-0.0001						
			0.0046				0.0166				-6e-5						
			0.0054				-0.0271				2.7e-5						

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration	Run	Combination	Objective Function	Volume (m³)				
19	7	0.4	Inner	967	109	19.66	0.9329				
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)				
			-0.2257		0.3195		0.1095				
			-0.2297		-0.3505		0.0788				
			0.7793		0.3099		0.2314				
			0.1357		-0.5273		0.1071				
			-0.6350		0.5849		0.6002				
			0.4962		-0.0880		0.6021				
			-0.2137		-0.3498		0.9665				
			-0.1453		0.7997		-0.5825				
			-0.1342		-0.6052		-0.7846				
			0.7112		0.6111		-0.3474				
			-0.1014		-0.9244		-0.3676				
			-0.7025		0.5037		0.5028				
			0.7222		-0.2103		0.6590				
			-0.3500		-0.1746		0.9204				
			0		0		0.3953				
Combination			1		1		0		0		
<div>Truss links</div> <div>Number: 11</div> <div>Total length: 13.661</div>					Thrusters						
					1	2	3	4	5	6	7
			Thrusters	1			+			+	+
				2			+		+	+	
				3				+	+		
				4						+	+
				5							+
				6							
7											











Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Combination	Objective Function	Volume (m³)							
24	7	0.9	Free		824	21	17.13	0.0618							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)			Z (m)							
			0.2981		0.02147			0.2593							
			0.3893		-0.3967			0.4814							
			-0.2649		-0.2606			0.1816							
			-0.3942		-0.4733			0.3071							
			0.07745		0.2476			0.5915							
			0.01338		0.3279			0.1848							
			0.3355		-0.1194			0.5079							
			-0.2535		0.9078			0.3341							
			-0.7234		-0.1521			-0.6734							
			-0.2153		-0.4475			0.868							
			0.6184		-0.7209			-0.3127							
			0.5727		0.3913			-0.7203							
			0.9634		0.1056			0.2463							
			-0.9625		-0.08412			0.2581							
			0		0			0.2245							
Combination			0	0	1	0	1	0	0						
<div>Truss links</div> <div>Number: 26</div>					Thrusters							Box		End-effector	
					1	2	3	4	5	6	7	8	9	10	11
			Thrusters	1				+	+	+		+	+		+
				2			+				+	+		+	
				3				+			+		+	+	
				4					+			+		+	
				5						+			+		+
				6								+	+	+	+
				7										+	+
Nodal Displacement			X(m)			Y(m)			Z(m)						
			0.00007			0.00008			0.00007						
			-0.0082			-0.0083			-0.0002						
			-0.0023			-0.0059			-0.0121						
			0.0018			-0.016			-0.0003						
			0.0010			0.00018			-0.0015						
			0.00005			-0.00014			0.00005						
			-0.0109			-0.0093			0.0051						





[illegible]

---



Serial Number	Number of Thrusters	Aerodynamic cone (m)		Configuration			Run		Combination		Objective Function		Volume (m³)	
30	8	0.9		Free			327		38		16.72		0.166	
Thrusters' Positions  														





[illegible]









## 4.3 Αμφίρροπες προπέλες

Στη παραπάνω παράγραφο παρουσιάστηκαν τα αποτελέσματα από τα τρεξίματα που έγιναν. Η περίπτωση που μελετήθηκε ήταν ότι οι κινητήρες δίνουν ώση μόνο προς μια κατεύθυνση, εκμεταλλευόμενοι τον βέλτιστο σχεδιασμό των προπελών για μέγιστη αποδοτικότητα. Εκτός από αυτή την περίπτωση όμως υπάρχουν και προπέλες οι οποίες μπορούν να λειτουργήσουν αμφίρροπα, με μικρότερο βαθμό απόδοσης.

Για να καλύψουμε και αυτή την περίπτωση, πραγματοποιήθηκαν τρεξίματα προσομοιώνοντας την αμφίρροπη λειτουργία των προπελών. Αυτή τη φορά οι περιπτώσεις που διερευνήθηκαν ήταν οι εξής:

- Αριθμός κινητήρων 6, 7, 8
- Προσανατολισμός κινητήρων ελεύθερος (μη περιορισμένος)
- Μήκος αεροδυναμικού κώνου 0.4, 0.6 ή 0.9

Βλέπουμε ότι τώρα μπορούμε να χρησιμοποιήσουμε και 6 κινητήρες, που είναι ο ελάχιστος αριθμός κινητήρων ώστε να έχουμε 6 βαθμούς ελευθερίας. Αυτό συμβαίνει γιατί εφόσον μπορούμε να έχουμε και αρνητικές ώσεις δε χρειάζεται να εισάγουμε τον έξτρα κινητήρα που θα αντισταθμίσει τις αρνητικές ώσεις.

Επίσης ο προσανατολισμός εδώ είναι μόνο ελεύθερος διότι εφόσον οι κινητήρες δίνουν ώση και από τις δύο κατευθύνσεις δε θα είχε νόημα η απαίτηση οι δυνάμεις να κοιτάνε προς τα έξω, ώστε να έχουμε μόνο εφελκυστικές τάσεις.

### 4.3.1 Υπολογισμός ώσης για αμφίρροπες προπέλες

Σε αυτή την περίπτωση εφόσον κάθε κινητήρας μπορεί να παράξει και θετικές και αρνητικές ώσεις, οι μεθοδολογίες που αναλύθηκαν παραπάνω για την εύρεση μη αρνητικών ώσεων, εδώ δε μας είναι χρήσιμες.

Στην περίπτωση που έχουμε 6 κινητήρες ο υπολογισμός των ώσεων είναι απλός, αφού έχουμε ένα σύστημα 6x6 της μορφής  $A \cdot x = b$  το οποίο λύνεται ως εξής:

$$x = A^{-1} \cdot b$$

Στις περιπτώσεις των 7 και 8 κινητήρων, όπως έχουμε πει και στο 2<sup>ο</sup> κεφάλαιο, έχουμε ένα υπό-ορισμένο σύστημα εξισώσεων το οποίο έχει παραπάνω από μία λύσεις. Μια μέθοδος για να αποκτήσουμε μία μοναδική λύση ορισμένη στο πραγματικό σύνολο, είναι η χρήση του ψευδό-ανάστροφου πίνακα (*Moore–Penrose pseudoinverse*) που ορίζεται ως εξής:

$$A^+ = A^T \cdot (A \cdot A^T)^{-1}$$

Και η λύση δίνεται ως εξής:

$$x = A^+ \cdot b$$

Παρακάτω δίνεται ένας πίνακας που περιέχει συγκριτικά τα αποτελέσματα των τρεξιμάτων. Τα αποτελέσματα έχουν επιλεγθεί με βάση τη μικρότερη απαιτούμενη μέγιστη ώση. Στις υπόλοιπες στήλες φαίνονται οι διαστάσεις του ως προς X,Y,Z , ο όγκος της κάθε λύσης σε κυβικά μέτρα . Τέλος στις 2 τελευταίες στήλες αναγράφονται για ποιο RUN αντιστοιχεί κάθε λύση και ο αύξων αριθμός της , για διευκόλυνση της εύρεσης της στη παρούσα διπλωματική.

Διάμετρος Προπέλας (inches)	Αριθμός Κινητήρων	Προσανατολισμός Κινητήρων	Μήκος Αεροδυναμικού Κώνου (m)	Συντελεστής Ωσης	Διαστάσεις (m)			Όγκος (m <sup>3</sup> )	RUN	a/a
					X	Y	Z			
10	6	Free	0.4	12.57	1.12	1.46	0.43	0.2182	50	1
			0.6	16.39	1.05	1.04	0.41	0.1243	43	2
			0.9	12.62	0.78	1.08	0.54	0.0945	2	3
	7	Free	0.4	7.76	0.97	0.99	0.51	0.1096	9	4
			0.6	9.79	1.49	1.18	0.60	0.3339	28	5
			0.9	10.31	2.19	2.43	1.75	1.5684	18	6
	8	Free	0.4	7.61	1.00	1.01	0.51	0.2969	2	7
			0.6	8.21	1.09	1.17	0.93	0.2654	13	8
			0.9	9.19	2.28	2.27	1.25	2.3609	35	9

**Πίνακας 4.3.1** Συγκριτική απεικόνιση των αποτελεσμάτων για αμφίρροπες προπέλες

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)						
1	6	0.4	Free		50	12.57		0.2182						
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)							
			-0.1568		-0.02342		0.1038							
			-0.378		0.5389		0.1071							
			-0.3181		-0.5459		0.4323							
			0.2799		0.7421		0.11							
			0.5658		0.4727		0.4147							
			0.7431		-0.7256		0.1386							
			0.928		-0.3722		-0.01873							
			0.2127		0.005169		0.9771							
			-0.3612		-0.1239		-0.9242							
			-0.5378		-0.8427		0.02642							
			0.2909		0.734		-0.6137							
			-0.5165		0.6492		0.5584							
			0		0		0.4684							
<div>Truss links</div> <div>Number:</div>					Thrusters				Box	End-effector				
					1	2	3	4	5	6	7	8	9	10
			Thrusters	1		+		+	+	+		+	+	+
				2			+		+		+	+		+
				3							+		+	+
				4							+		+	+
				5							+	+		
				6							+	+	+	
<div>Nodal Displacement</div>			X(m)		Y(m)		Z(m)							
			0.001720062		-0.001127102		5.67523E-05							
			-0.003892929		-0.002712022		7.06797E-05							
			0.003873619		-0.002290738		-2.48866E-05							
			-0.005540799		0.002076849		-3.01333E-05							
			-0.003419181		0.004144708		4.90681E-05							
			0.005265155		0.00535698		0.000234538							
			0.001720062		-0.001127102		5.67523E-05							



Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function	Volume (m³)															
2	6	0.6	Free		43	16.39	0.1243															
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)			Y (m)			Z (m)													
			-0.1051			-0.3058			0.1051													
			0.03994			0.6175			0.4032													
			-0.668			0.7419			0.05233													
			-0.4946			-0.2253			0.3493													
			-0.472			0.3073			0.416													
			0.3896			-0.1293			0.1391													
			0.4946			0.7955			-0.35													
			-0.3853			0.03283			0.9222													
			0.7497			-0.6097			-0.2572													
			0.2086			0.7036			0.6792													
			-0.9438			0.1852			-0.2738													
			-0.2995			-0.8041			-0.5135													
			0			0			0.3605													
<div>Truss links</div> <div>Number:</div>											Thrusters		Box		End-effector							
					1		2		3		4		5		6		7		8		9	
			Thrusters	1			+					+			+		+	+		+		
				2					+	+				+			+	+		+		
				3										+		+	+					
				4										+		+	+	+				
				5								+	+			+	+					
6									+	+		+	+	+								
<div>Nodal Displacement</div>			X(m)			Y(m)			Z(m)													
			-0.000381446			-0.000128948			-0.00076416													
			0.005924131			-0.00387265			0.000185448													
			0.007125181			0.006347577			0.000218021													
			-0.002159212			0.00475808			2.37342E-05													
			-7.85549E-05			-0.000139078			-1.36089E-05													
			-0.000242426			-0.000689217			2.26286E-05													
			-0.000381446			-0.000128948			-0.00076416													



Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function	Volume (m³)							
4	7	0.4	Free		9	7.76	0.1096							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)		Y (m)		Z (m)							
			-0.207		-0.2033		0.1932							
			-0.4902		-0.02356		0.1473							
			0.3125		-0.1785		0.09781							
			-0.005691		0.4785		0.5111							
			0.4875		-0.36		0.5096							
			-0.1582		0.4399		0.06501							
			0.3978		-0.5246		0.07063							
			-0.09456		0.6472		0.7564							
			-0.5313		-0.8046		0.2652							
			0.7039		0.3665		0.6085							
			0.6215		-0.7834		-0.004089							
			-0.9962		-0.05692		0.06655							
			0.2366		-0.05727		-0.9699							
			0.06013		0.6885		-0.7227							
			0		0		0.191							
<div>Truss links</div> <div>Number:</div>			<div></div>		Thrusters				Box		End-effector			
					1	2	3	4	5	6	7	8	9	10
			Thrusters	1							+	+	+	+
				2				+			+	+	+	+
				3						+	+		+	+
				4					+		+	+		
				5						+	+	+		
				7							+	+	+	+
<div>Nodal Displacement</div>			X(m)		Y(m)		Z(m)							
			-0.00875576		-0.004563106		-7.9027E-06							
			0.000665376		0.000663444		-4.3528E-06							
			0.003318011		-0.003068535		3.86324E-05							
			-0.002485358		-0.004893894		-0.00013673							
			-0.005795512		0.001682522		5.26678E-05							
			-0.001470491		-0.005345112		4.00962E-05							
			-0.00875576		-0.004563106		-7.9027E-06							

Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function		Volume (m³)							
5	7	0.6	Free		28	9.79		0.3339							
<div>Thrusters Positions</div> <div>Thrusters Orientations</div> <div>Box Position</div>			X (m)			Y (m)			Z (m)						
			0.8436			0.6247			0.1287						
			-0.4265			-0.5673			0.3483						
			-0.4307			0.613			0.1254						
			-0.6596			-0.3888			0.3451						
			0.3977			0.1073			0.6098						
			-0.4344			-0.2067			0.2011						
			0.4558			-0.6972			0.1228						
			0.1124			0.7827			0.6122						
			0.05439			-0.1575			0.986						
			0.9854			0.1478			-0.08512						
			-0.1469			-0.9051			-0.3991						
			0.1189			0.7457			-0.6556						
			-0.9585			0.2814			-0.04465						
			-0.1662			-0.8951			-0.4138						
			0			0			0.1092						
<div>Truss links</div> <div>Number:</div>					Thrusters							Box		End-effector	
					1	2	3	4	5	6	7	8	9	10	11
			Thrusters	1			+		+		+			+	+
				2			+	+	+				+		+
				3						+	+				
				4							+	+	+	+	
				5							+	+	+	+	+
				6								+		+	+
				7								+	+	+	
<div>Nodal Displacement</div>			X(m)			Y(m)			Z(m)						
			0.001634593			-0.00078476			-0.006532406						
			0.000698855			0.002163716			0.004424188						
			0.001654105			0.001208579			-0.000868897						
			-0.000927512			-0.000091039			-0.002106732						
			0.003020108			0.001874972			-0.002427885						
			-0.000717505			0.00129767			-0.00028355						
			-0.000439988			-0.000215392			0.000529123						





Serial Number	Number of Thrusters	Aerodynamic cone (m)	Configuration		Run	Objective Function				Volume (m³)						
8	8	0.6	Free		13	8.21				0.2654						
Thrusters Positions			X (m)		Y (m)				Z (m)							
			-0.214		-0.2065				0.6877							
			0.2698		0.4192				0.8512							
			0.6106		-0.4165				0.2949							
			-0.4575		-0.08526				0.9383							
			-0.2088		0.1679				0.06784							
			0.02917		-0.7673				0.2935							
			0.6485		-0.1056				0.5991							
			-0.2954		0.1006				0.3486							
			0.2927		0.9382				0.1846							
Thrusters Orientations			-0.03119		0.124				0.9918							
			0.5707		0.2305				-0.7881							
			-0.4887		0.4432				-0.7515							
			-0.988		0.1241				0.09235							
			-0.5732		-0.8027				0.1645							
			0.9527		-0.1095				0.2834							
			0.2649		-0.9479				-0.1771							
			0		0				0.283							
Truss links  Number:					Thrusters								Box		End-effector	
					1	2	3	4	5	6	7	8	9	10	11	12
			Thrusters	1							+		+	+	+	
				2								+	+	+	+	
				3				+	+	+		+	+	+	+	+
				4							+	+				+
				5							+		+			+
				6								+	+		+	+
				7									+			+
				8										+		+
Nodal Displacement			X(m)		Y(m)				Z(m)							
			-0.02200959		0.02522126				0.00087379							
			0.02824143		-0.01990218				0.0011306							
			-0.03683314		-0.0534493				0.00078199							
			-0.07734693		0.03343411				-0.00716252							
			0.01052476		0.01363133				-0.00034142							
			-0.06701121		-0.02598056				-0.00074426							
			-0.01186413		-0.0583635				0.00405837							
			0.00621698		0.02301828				-0.00123823							

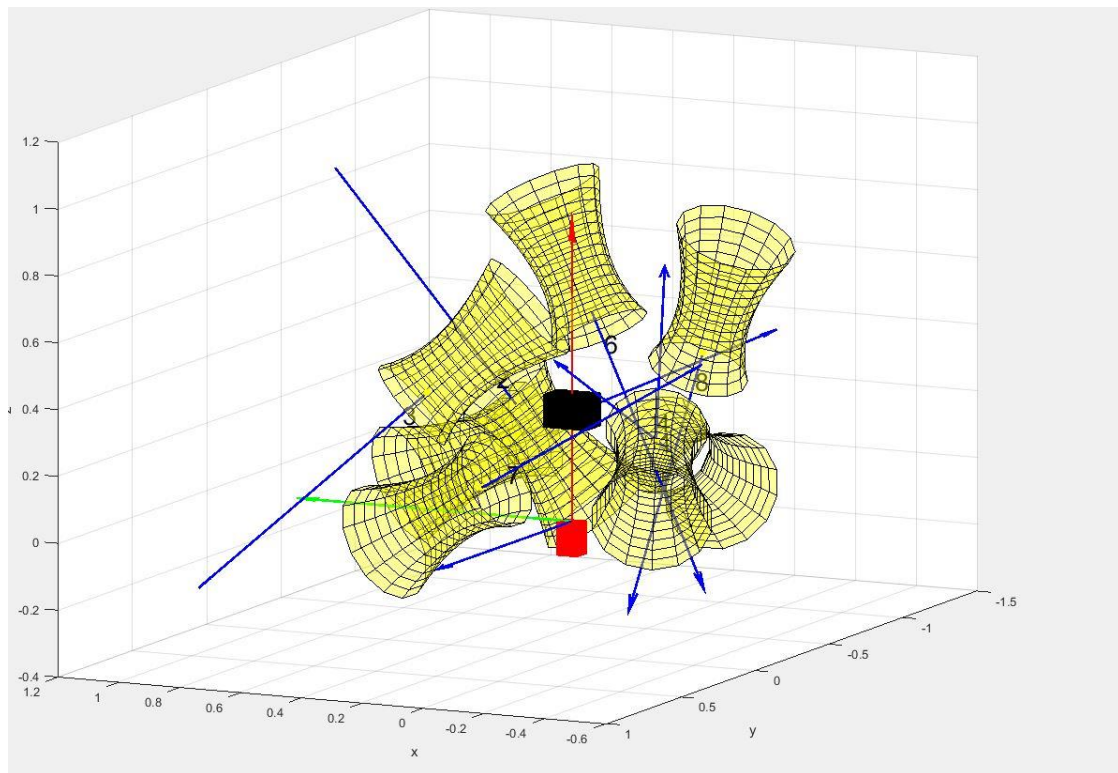




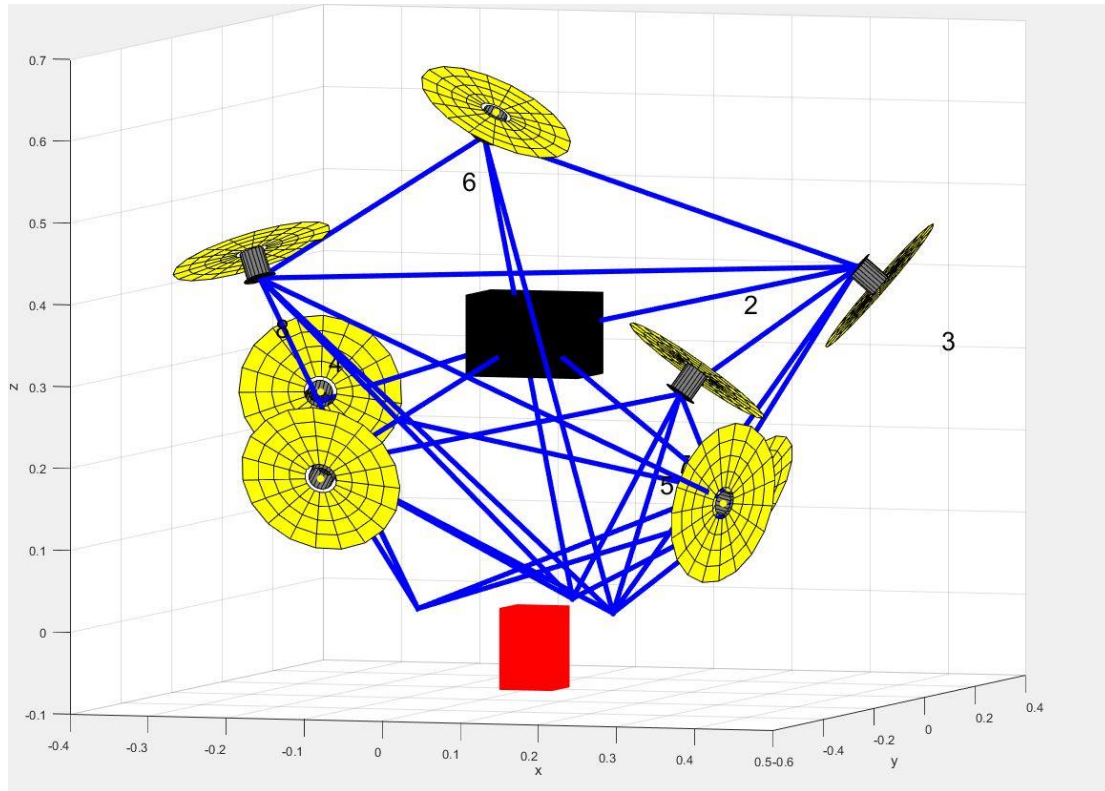
## 4.4 Επίλογος

Στο παρόν κεφάλαιο παρουσιάστηκαν τα αποτελέσματα των τρεξιμάτων που έγιναν για τις διάφορες περιπτώσεις που αναφέρθηκαν. Παρατηρώντας τα αποτελέσματα μπορούμε να δούμε οι βέλτιστες λύσεις διαφέρουν ανάλογα με το κριτήριο της σύγκρισης. Εμείς καθώς στην εν λόγω διπλωματική το κύριο μέλημα μας ήταν να έχουμε την μικρότερη μέγιστη απαιτούμενη ώση, και παίρνοντας αυτό σαν κριτήριο σύγκρισης βλέπουμε ότι η λύση με την καλύτερη ώση είναι η λύση με Serial No 28. Δηλαδή αυτή με 8 thrusters, μήκος αεροδυναμικού κώνου 0.4 και free προσανατολισμό κινητήρων, η οποία έχει μέγιστη απαιτούμενη ώση 14.60 N και όγκο 0.0830 κυβικά μέτρα.

Αυτή η λύση θα παρουσιαστεί γραφικά στο Matlab. Αρχικά θα παρουσιαστεί η λύση ως προς τους αεροδυναμικούς κώνους και έπειτα η λύση με το προτεινόμενο δικτύωμα και τις προπέλες.



**Σχήμα 4.4.1** Η προτεινόμενη λύση, με αεροδυναμικούς κώνους, τα μπλε διανύσματα δείχνουν την κατεύθυνση της δύναμης



**Σχήμα 4.4.2** Η προτεινόμενη λύση, με το δικτύωμα και τις προπέλες

Ο κατασκευαστής του AM , μπορεί για δικούς του λόγους , όπως για παράδειγμα λόγους κατασκευαστικούς ή λόγους ελεγχιμότητας του μοντέλου , να επιλέξει άλλη λύση. Επίσης παρατηρούμε ότι οι λύσεις για αμφίρροπες προπέλες στη παράγραφο 4.3 δίνουν καλύτερες ώσεις , όμως δεν επιλέχθηκαν διότι δεν έχει διερευνηθεί η αποδοτικότητα τους και η απόκριση τους στις ανάγκες του ελέγχου.

## Κεφάλαιο 5<sup>ο</sup>

### Συμπεράσματα

#### 5.1 Σύνοψη

Η παρούσα διπλωματική εργασία , αποτέλεσε συνέχεια της [1] . Η βασική προσέγγιση του προβλήματος του σχεδιασμού του Aerial Manipulator έμεινε η ίδια, δηλαδή θέλουμε να σχεδιάσουμε ένα UAV το οποίο να έχει προσαρτημένο πάνω του ένα χειριστή. Αυτό το UAV θέλουμε να έχει 6 βαθμούς ελευθερίας και η μαθηματική του μοντελοποίηση να είναι τέτοια ώστε να διευκολύνει στον έλεγχο του. Επίσης θέλουμε να είναι όσο το δυνατόν μικρότερο σε όγκο και οι θέσεις και οι προσανατολισμοί των κινητήρων του να είναι τέτοιοι ώστε να σηκώνουν το AM με το λιγότερο δυνατό έργο.

Για να εκπληρωθούν οι παραπάνω στόχοι, έγινε αρχικά η μαθηματική μοντελοποίηση του Aerial Manipulator , έπειτα μαθηματοποιήθηκε το πρόβλημα θέτοντας τους κατάλληλους περιορισμούς και έγιναν μια σειρά από βελτιστοποιήσεις στο περιβάλλον Matlab. Για να διερευνήσουμε περισσότερο το πλήθος των κινητήρων που χρειάζονται , το μέγεθος του αεροδυναμικού κώνου που κατ' επέκταση επηρεάζει το συνολικό μέγεθος της κατασκευής , τον προσανατολισμό των κινητήρων ( αν θέλουμε να ασκούν δυνάμεις μόνο προς τα «έξω» ή αν ο προσανατολισμός θα είναι ελεύθερος) , έγινε βελτιστοποίηση για κάποιες περιπτώσεις αλλάζοντας κάθε φορά μία παράμετρο από τις παραπάνω. Οι περιπτώσεις που διερευνήσαμε είναι οι εξής:

Προσανατολισμός	Αριθμός Κινητήρων	Μέγεθος αεροδυναμικού κώνου
Inner	7	0.4 , 0.6 , 0.9 [m]
	8	
	9	
Free	7	0.4 , 0.6 , 0.9 [m]
	8	
	9	

**Πίνακας 5.1.1** Πίνακας περιπτώσεων τρεξιμάτων

Τα παραπάνω ισχύουν για την περίπτωση που χρησιμοποιούμε κινητήρες και προπέλες που δίνουν ώση ως προς μία μόνο κατεύθυνση. Διερευνήθηκε και μία άλλη περίπτωση, όπου χρησιμοποιούνται ζευγάρια κινητήρων-προπελών τα οποία μπορούν να δώσουν ώση και προς τις δύο κατευθύνσεις. Σε αυτή την περίπτωση οι περιπτώσεις που διερευνήσαμε ήταν διαφορετικές. Αυτή τη φορά ο προσανατολισμός των κινητήρων

ήταν ελεύθερος, και ο αριθμός των κινητήρων κυμαινόταν από 6 μέχρι 8. Οι περιπτώσεις αυτές φαίνονται στον παρακάτω πίνακα :

Προσανατολισμός	Αριθμός Κινητήρων	Μέγεθος Αεροδυναμικού Κώνου
Free	6	0.4 , 0.6 , 0.9 [m]
	7	
	8	

**Πίνακας 5.1.2** Πίνακας περιπτώσεων τρεξιμάτων για αμφίρροπες προπέλες

Έπειτα , δημιουργήθηκε ένα λογισμικό σε περιβάλλον Matlab με αλληλεπίδραση του Ansys APDL , το οποίο παρήγαγε ένα χωρό-δικτύωμα βελτιστοποιημένο ως προς το βάρος του. Ούτως ώστε να έχουμε μία εικόνα του πως θα υλοποιηθεί κατασκευαστικά το Aerial Manipulator. Το εν λόγω λογισμικό μπορεί να χρησιμοποιηθεί για κάθε περίπτωση, συγκεκριμένα για την περίπτωση των inner όμως , χρησιμοποιήθηκε μια άλλη μεθοδολογία , η οποία μπορεί να εφαρμοστεί μόνο για τα inner, διότι αντιπροσωπεύει τον σκοπό που διερευνήθηκε η περίπτωση inner, να ασκούνται δηλαδή μόνο εφελκυστικές τάσεις στις ράβδους του δικτύωματος

## 5.2 Προτάσεις για μελλοντική εργασία

### Υλικό τεχνικός Εξοπλισμός

Το επόμενο βήμα που πρέπει να γίνει είναι, έχοντας τα αποτελέσματα της κάθε περίπτωσης, να γίνει έρευνα στο εμπόριο για να βρεθούν οι κατάλληλοι κινητήρες από άποψη βάρους και ισχύος σε συνδυασμό με τις κατάλληλες προπέλες που να ικανοποιούν τις προδιαγραφές.

Επίσης πρέπει να διερευνηθούν τα υλικά της κατασκευής, ράβδοι για το δικτύωμα, καλώδια, μπαταρίες και ούτω καθεξής, τα οποία να πληρούν τις προδιαγραφές αλλά και να είναι όσο το δυνατόν πιο ελαφριά.

### Δικτύωμα και κατασκευή

Πρέπει αφού επιλεγεί η καλύτερη περίπτωση, να διερευνηθεί ο τρόπος με τον οποίο θα υλοποιηθεί το χωρό-δικτύωμα. Πώς θα συνδέονται οι ράβδοι με τους κινητήρες-βάση-ηλεκτρονικά, τι συνδετικά εξαρτήματα θα χρησιμοποιηθούν.

### Δυναμική Ανάλυση

Στην παρούσα διπλωματική εργασία έγινε στατική ανάλυση του μοντέλου , διότι αφού γινόταν βελτιστοποίηση θα ήταν πολύ δύσκολο να βελτιστοποιήσουμε ένα πολύπλοκο

δυναμικό μοντέλο. Μια μελλοντική εργασία θα μπορούσε να είναι η δυναμική ανάλυση του μοντέλου, καθώς επίσης και η διερεύνηση ως προς τον έλεγχο του.

# Πα ρ α ρ τ ή μ α τ α

## Πα ρ ά ρ τ η μ α Α :

### Αλγόριθμοι Βελτιστοποίησης

Στο Παράρτημα Α θα παρουσιαστεί ο αλγόριθμος βελτιστοποίησης των θέσεων και των προσανατολισμών των κινητήρων και της θέσεις του κουτιού με τα ηλεκτρονικά. Θα παρατεθούν με την σειρά που καλούνται από τον αλγόριθμο.

Αλγόριθμος: **main.m**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               National Technological University of Athens                               %
%                               Mechanical Engineering Department                                         %
%                               Diploma thesis                                                            %
%                                                                                                         %
%                               Control Oriented Aerodynamic Design Optimization                             %
%                               for an Aerial Manipulator                                                  %
%                                                                                                         %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                                         %
% SUMMARY                                                                                                  %
%                                                                                                         %
% In this code we define the parameters of the optimization, and prepare %
% some of the constraints we will use in the optimization. First, the user %
% is requested to define the desired parameters shown below. Next, using an %
% optimization algorithm we find the best position and radius of the sphe- %
% res the wouldcover the volume of the cone. After that we create the com- %
% binations of the spheres that would serve as a constraint later. Elimi- %
% nating the combinations refering to the same sphere leaves us only with %
% the combinations that matter. Also, we create the vector containing the %
% distances the centers of the spheres should overcome so the won't inter- %
% sect one another. Finally, we come to the optimization where we create %
% the initial points using the latin hypercube function and we call the %
% function min_prop_effort.m which contains the optimization algorithm. If %
% the solution is feasible then another function is also called to access %
% if any spin combination of the thrusters can provide smaller values for %
% the propulsion effort. At the end, the information of the optimization are %
% written to a text file for later use. %
```

```

% %
% %
% VARIABLES USED %
% %
% num_thr : Number of Thrusters. %
% num_run : Number of times to run the Optimization Algorithm. %
% num_iter : Number of iterations inside Optimization Algorithm. %
% len_cone : Length of aerodynamic cone. %
% prop_size : Propeller's size in inches. %
% thr_ori : Thrusters orientation constraint. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clc

ses = input('Session number:');
num_run = input('Number of runs of the GB algorithm:');
num_iter = input('Number of iterations of the GB algorithm (0:default):');
num_thr = input('Number of thrusters : ');
len_cone = input('Aerodynamic Cone Length (m): ');
prop_size = input('Propeller size (in):');
thr_ori = menu('Thruster orientation.','Inner','Free')

bi = menu('Type','bi-directional','one-way')

global normR

if num_iter==0
    num_iter=1000;
end

% Sphere Optimization.
% Number of Spheres depends on the length of the cone.
if bi==2
    if len_cone<=0.6
        num_sphr=2;
    elseif len_cone>0.6
        num_sphr=3;
    end
elseif bi==1
    if len_cone<=0.6
        num_sphr=3;
    elseif len_cone>0.6
        num_sphr=5;
    end
end

prop_rad=prop_size*0.0254/2; % Propeller radius in meters.
cube_edge=0.25; % The edge of the cube
% containing the initial points.

sphr_rad=zeros(num_sphr,1);
sphr_pos=zeros(num_sphr-1,1);
sphr_rad(1)=prop_rad+0.01; % Radius of the first sphere.
ply=[-1.109 1.5524 -0.3094 prop_rad]; % Polynomial coefficients.
disp('Optimizing spheres.')
if bi==2

```

```

[x,fval]=fminconesphere(num_sphr-1,ply,len_cone,prop_rad);
elseif bi==1
    if num_sphr==3
        [x,fval]=fminconesphere(num_sphr-2,ply,len_cone,prop_rad);
    elseif num_sphr==5
        [x,fval]=fminconesphere(num_sphr-3,ply,len_cone,prop_rad);
    end
end

if bi==2
for j=1:num_sphr-1
    sphr_rad(j+1)=x(2*j);
    sphr_pos(j)=x(2*j-1);
end
elseif bi==1
    if num_sphr==3
        for j=1:num_sphr-2
            sphr_rad(j+1)=x(2*j);
            sphr_pos(j)=x(2*j-1);
        end
    elseif num_sphr==5
        for j=1:num_sphr-3
            sphr_rad(j+1)=x(2*j);
            sphr_pos(j)=x(2*j-1);
        end
    end
end
end
if bi==1
    if num_sphr==3
        sphr_rad(3)=sphr_rad(2);
    elseif num_sphr==5
        sphr_rad(4)=sphr_rad(2);
        sphr_rad(5)=sphr_rad(3);
    end
end
end

sphr_comb=nchoosek(1:num_sphr*num_thr+1,2); % Sphere combinations.
h=0; % Counter.

% Spheres in the same cone should not be considered,
% Same_cone saves these combination in order to delete them.
same_cone=zeros(num_thr*nchoosek(num_sphr,2),1);

% Every column represents a cone, the numbers in the first
% row belong to the first sphere of every cone, the second
% to the second sphere etc. (having 2 spheres per cone is
% not a problem here).
if bi==1
    L=[1:num_thr ;
        (num_thr+1):2*num_thr ;
        (2*num_thr+1):3*num_thr ;
        (3*num_thr+1):4*num_thr ;
        (4*num_thr+1):5*num_thr;];
elseif bi==2
    L=[1:num_thr ;
        (num_thr+1):2*num_thr ;
        (2*num_thr+1):3*num_thr;
end

```



```

% This for-loop finds the sphere combinations that belong to the cone.
for k=1:length(sphr_comb)
    [~,q1]=find(L(1:num_sphr,')==sphr_comb(k,1));
    [~,q2]=find(L(1:num_sphr,')==sphr_comb(k,2));
    if q1==q2
        h=h+1;
        same_cone(h)=k;
    end
end
% Here the previously found combinations are being deleted.
sphr_comb(same_cone,:)=[];
% Radius of the sphere belonging to the electronics box.
sphr_radbox=0.1;

% Minimum distance between spheres.
for o=1:num_sphr
    ds((o-1)*num_thr+1:o*num_thr,1:num_thr*num_sphr+1)=sphr_rad(o);
end
ds(num_thr*num_sphr+1,:)=sphr_radbox;
ds=ds+ds';
ds=triu(ds);
dist_sphr=ds(sub2ind(size(ds),sphr_comb(:,1),sphr_comb(:,2))));

disp('Optimization Started')
Iter=num_iter; % Maximum number of iterations.
y=0; % Counter.

while y<num_run
    % Creating the initial points for Gradient-Based algorithm.
    x0=lhsdesign(1,6*num_thr+1);
    x0=x0*cube_edge;
    % Shifting the points from the positive subspace so as
    % to have negative values as well.
    x0=x0-cube_edge/2;
    % Shifting the points in order to have only positive
    % values for the z position coordinates.
    for w=1:num_thr
        x0(w*3)=x0(w*3)+cube_edge/2+0.05;
    end
    x0(6*num_thr+1)=x0(6*num_thr+1)+cube_edge/2+0.05;
    tic;
    % fmincon optimization.
    [x1, fval1,ef,output1]=min_prop_effort(x0,Iter,sphr_comb,...
        dist_sphr,sphr_pos,num_sphr,num_thr,thr_ori,bi);

    time1=toc;
    time=time1/60;
    y=y+1; % Counter
    prnt1=0;
    P1=0;
    % If the solution of the optimization is feasible then the algorithm
    % prosides to find which of the thruster spin combinations
    % minimizes the propulsion effort.
    if bi==2
        if ef>=0 && output1.constrviolation<1e-4
            P1=prop_effort_comb(x1,num_thr,normR);
            prnt1=1;
        end
    end
end

```

```

% creating the orientation unit vectors.
x=[x1(1:6*num_thr) 0 0 x1(6*num_thr+1)];
for k=3*num_thr+1:3:6*num_thr
    x(k:k+2)=x(k:k+2)/norm(x(k:k+2));
end
% Saving results and optimization informations into a text file.
res2txt(num_thr,prop_size,len_cone,thr_ori,y,ses,...
        x,fval1,output1,P1,prnt1,time);
end

```

Αλγόριθμος: **fminconesphere.m**

```

function [x,fval,ef]=fminconesphere(num_sphr,ply,len_cone,prop_rad)
A=zeros(2*num_sphr,2*num_sphr); % Linear inequality Constraints.
for k=1:2:2*num_sphr
    A(k:k+1,k:k+1)=[1 1;-1 1];
end
b= repmat([len_cone; -prop_rad],num_sphr,1);

ub= repmat([len_cone inf],1,num_sphr); % Lower bounds.
lb= repmat([prop_rad 0 ],1,num_sphr); % Upper bounds.
x0=rand(1,num_sphr*2)*(len_cone-prop_rad)+prop_rad; % Initial points.
options=optimset('Display','final-detailed','Algorithm','interior-
point','MaxFunEvals',1500,'MaxIter',300); % Optimization options.
[x,fval,ef]=fmincon(@(x)min_area(x,num_sphr,ply,len_cone,prop_rad),x0,A,b,[],[],lb,ub,
    @(x)nonlc(x,num_sphr,ply),options);
end

function f=min_area(x,num_sphr,ply,len_cone,prop_rad)
% Because the geometry is symmetry about the x-axis we are going to
% manipulate the problem in 2D.
y=polyint(ply); % Integral of polynomial.
cone_area=polyval(y,len_cone)-polyval(y,prop_rad); % Area of the cone.
sphr_area=0;
for i=1:num_sphr % Total area of the circles.
    sphr_area=sphr_area+pi*x(i*2)^2;
end
f=cone_area-sphr_area; % Objective function : minimize the area of
the cone that is not covered from the circles
end

function [c,ceq]=nonlc(x,num_sphr,ply)

c=[];
% Nonlinear inequality constraints.
% Spheres should not : intersect each other, move farther than len_cone and intersect
the first sphere.
if num_sphr>=2
    nck=nchoosek(1:2:2*num_sphr,2);
    noint=zeros(length(nck),1);
    for l=1:size(nck,1)
        noint(l)=x(nck(l,1)+1)+x(nck(l,2)+1)-abs(x(nck(l,1))-x(nck(l,2)));
    end
end

```

```

c=noint;
end

% Nonlinear equality constraints.
d=zeros(num_sphr,1);
for j=1:num_sphr          % The radius of the sphere should not be greater than the
radius of the cone in this position.
    d(j)=x(2*j)-polyval(ply,x(2*j-1));
end

ceq=d;

end

```

Αλγόριθμος: **min\_prop\_effort.m**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           National Technological University of Athens           %
%           Mechanical Engineering Department                     %
%           Diploma thesis                                       %
%                                                                 %
%           Control Oriented Aerodynamic Design Optimization      %
%           for an Aerial Manipulator                             %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
% SUMMARY                                                         %
%                                                                 %
% In this algorithm takes place the optimization. Data from the main.m %
% algorithm are passed here to start the optimization. First the lower and %
% upper bounds of the optimization variables are created depending on %
% the size of the aerodynamic cones. Then the fmincon options are being %
% set and the optimization algorithm is being called.              %
%                                                                 %
% Next we see the objective function within which we call another %
% function that calculates the value of the objective function and it is %
% called prop_effort.m. Finally there is the constraints function in which %
% we define the equality and inequality non linear constraints.     %
%                                                                 %
% VARIABLES USED                                                  %
%                                                                 %
% num_thr   : Number of Thrusters.                                %
% len_cone  : Length of aerodynamic cone.                         %
% prop_size : Propeller's size in inches.                         %
% thr_ori   : Thrusters orientation constraint.                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x, fval,exitflag,output]=min_prop_effort(x0,MaxIter,sphr_comb,...
                                                    dist_sphr,sphr_pos,num_sphr,num_thr,thr_ori,bi)

if thr_ori==1
    max_edge=max(sphr_pos)*2;

```

```

else
    max_edge=max(sphr_pos);
end

lb=[repmat([-max_edge -max_edge 0.05],1,num_thr)...
    -inf*ones(1,num_thr*3) 0]; % Lower Bounds

ub=[repmat([max_edge max_edge max_edge],1,num_thr)...
    inf*ones(1,num_thr*3) max_edge]; % Upper Bounds

% Optimization Options.
options = optimset('Display','iter','Algorithm','interior-point',...
    'AlwaysHonorConstraints','bounds','InitBarrierParam',10,...
    'InitTrustRegionRadius',7,'SubproblemAlgorithm','cg',...
    'MaxIter',MaxIter,'TolX',1e-6,'TolFun',1e-6,'MaxFunEvals',15000,...
    'ScaleProblem','obj-and-constr','UseParallel',true);

[x, fval,exitflag,output] = fmincon(@(x)volume(x,num_thr,bi),x0,[],[],[],...
    [],lb,ub,@(x)nonlincon(x,sphr_comb,dist_sphr,sphr_pos,num_sphr,num_thr,...
    thr_ori,bi), options);

end

function f_obj = volume(x,num_thr,bi)
global normR

% This norm represents the volume of the structure,
% thus serving as a weight coefficient.
normR=norm([x(1:num_thr*3) [0 0 x(6*num_thr+1)]]);

f_obj=prop_effort(x,num_thr,normR,bi); % Objective function.
end

function [c, ceq]=
nonlincon(x,sphr_comb,dist_sphr,sphr_pos,num_sphr,num_thr,thr_ori,bi)

bc=[0 0 x(6*num_thr+1)]; % Electronics' box coordinates.
R=zeros(num_thr,3); % Size of the position matrix.
F=zeros(num_thr,3); % Size of the orientation matrix.
f=zeros(num_thr,3); % Size of the unit orientation matrix.
% Matrix containing the norms of the orientation vectors.
Fn=zeros(num_thr,1);
% Size of the cross product between position and orientation matrices.
SFr=zeros(3,num_thr);
Rs=zeros(num_sphr*num_thr+1,3); % Size of the sphere position matrix.
Rs(num_sphr*num_thr+1,:)=bc; % Setting the center of the box's sphere.
% Size of the dot product between position and orientation.
dt=zeros(num_thr,1);

for j=1:num_thr % Setting the optimization variables
    R(j,:)=x(3*j-2:3*j); % to the position and orientation matrices.
    F(j,:)=x(3*(num_thr+j)-2:3*(num_thr+j));
end
Rs(1:num_thr,:)=R; % The centers of the first spheres
% are the position of the propellers.

```

```

for i=1:num_thr
    Fn(i)=norm(F(i,:));
    f(i,:)=F(i,:)/Fn(i);           % Unit orientation vectors.
    r=(R(i,:)-bc)/norm(R(i,:)-bc); % Unit distance vectors (box to thruster).
    dt(i,1)=dot(r,f(i,:));         % Dot product of the 2 previous vectors.

    Rs(num_thr+i,:)=R(i,:)-f(i,:)*sphr_pos(1);           % Spheres' positions.
    if bi==2
        if num_sphr==3
            Rs(2*num_thr+i,:)=R(i,:)-f(i,:)*sphr_pos(2);
        end
    end

    if bi==1
        if num_sphr==3
            Rs(2*num_thr+i,:)=R(i,:)+f(i,:)*sphr_pos(1);
        elseif num_sphr==5
            Rs(3*num_thr+i,:)=R(i,:)+f(i,:)*sphr_pos(1);
            Rs(4*num_thr+i,:)=R(i,:)+f(i,:)*sphr_pos(2);
        end
    end

    SF=[0      -f(i,3) f(i,2);      % Skew-symmetric matrix of unit vector f.
        f(i,3) 0      -f(i,1);
        -f(i,2) f(i,1) 0      ];

    SFr(:,i)=-SF*R(i,:);           % Cross product of f and R.
end

D=[f';SFr];           % Transformation matrix from actuator to end-effector space.

s=svd(D);             % Singular value decomposition of the transformation matrix.
cn=cond(D);           % Condition number of the transformation matrix.

% Creating the sphere constraints.

spheres=Rs(sphr_comb(:,1),:)-Rs(sphr_comb(:,2),:);
spheres=spheres.^2;
spheres=sum(spheres,2);
spheres=spheres.^(1/2);
spheres=-spheres+dist_sphr;

% Nonlinear inequality constraints.
if thr_ori==1         % Inner orientation of the thrusters.
    c=[ cn-10      ; % Condition number should less than 10.
        -s+0.001 ; % Singular values must be no less than 0.001.
        spheres ; % Spheres' distance should be more than dist_sphr
        -dt+0.93 ; % dot product of vectors r and f (see above) should be
        dt-0.99]; % more than 0.93(for 7 thruster, 0.95 and 0.97 for 8 and 9
                  % thrusters respectively) and less than 0.99 for all
                  % thruster numbers.
else                 % Free orientation of the thrusters.
    c=[ cn-10      ;
        -s+0.001 ;

```

```

        spheres];
end

% Nonlinear equality constraints.
ceq=sum(D,2);      % Sum of the unit orientation vectors should be equal to zero.

end

```

#### Αλγόριθμος: **prop\_effort.m**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           National Technological University of Athens           %
%           Mechanical Engineering Department                     %
%           Diploma thesis                                       %
%                                                                 %
%           Control Oriented Aerodynamic Design Optimization      %
%           for an Aerial Manipulator                             %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
% SUMMARY                                                         %
%                                                                 %
% In this algorithm we calculate the forces needed to hover in each %
% orientation of the aerial manipulator and then find the maximum of all. %
% First, we read from the x the positions and orientation of the thrusters %
% as well the position of the box. Then we create the transformation matrix %
% to use later. The next step is to define the gravity vector and then start %
% the loops. The first loop rotate the orientation of the gravity vector %
% about the z-axis and the second about the y-axis (the second loop is %
% inside the first one). For each of the inner loops(for each orientation) %
% we calculate the propulsion effort. Finally when all the loops have %
% finished we save the maximum one define it as the output of the function. %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [l]=prop_effort(x,num_thr,normR,bi)
max_thrust=0;          % Defining variables and matrices.
pro=1:num_thr-1;
thrustl=zeros(num_thr,1);
f=zeros(num_thr,3);
SFr=zeros(3,num_thr);

r=zeros(num_thr+1,3);
F=zeros(num_thr,3);

if length(x)>2*num_thr+1      % Creating r and F matrices depending on
    for j=1:num_thr          % the size of matrix x.
        r(j,:)=x(3*j-2:3*j);
        F(j,:)=x(3*(num_thr+j)-2:3*(num_thr+j));
    end
    r(num_thr+1,:)=[0 0 x(6*num_thr+1)];
else
    r=[x(1:num_thr,:); x(2*num_thr+1,:)];
    F=x(num_thr+1:2*num_thr,:);

```

```

end

deg_int=10; % Degrees interval.
rad_int=degtorad(deg_int); % Rad interval.
% AM masses [thruster mass, end-effector mass, rest of structure mass].
mi=[0.1*ones(1,num_thr)';0.25;0.5+log(normR)*0.4];
m=sum(mi); % Total mass of AM.
% Centers of the individual masses.
ri=[r(1:num_thr,:);[0 0 -0.05];r(num_thr+1,:)]';
rg=ri*mi/sum(mi); % Center of mass of AM.

% In the next for-loop unit orientation vector(f), skew symmetric matrix of
% the unit orientation vector(SF) and the cross product matrix (SFr) are
% defined for each one of the thrusters.

for i=1:num_thr
    f(i,:)=F(i,:)/norm(F(i,:)); % Unit orientation vectors.

    SF=[0 -f(i,3) f(i,2); % Skew-symmetric matrix of unit vector f.
        f(i,3) 0 -f(i,1);
        -f(i,2) f(i,1) 0 ];

    SFr(:,i)=SF*r(i,:); % Cross product of f and R.
end
D=[f';SFr]; % Transformation matrix from actuator to end-effector space.

if num_thr==7
    D6=D(1:6,1:6);
    [L,U]=lu(D6); % Matlab suggests to use LU factorization in linear problems
                  % like A*x=b, where A does not change but b does.
else
    D_new=D(:,1:num_thr-1);
    id=pinv(D_new); % Pseudo-inverse of a non-square matrix.
end

g=[0;0;-9.81]; % Gravity vector.
Dpinv=pinv(D); %pseudo-inverse for bidirectional
% Calculating the propulsion efforts needed in its direction.
for ipsi=0:rad_int:2*pi
    for itheta=0:rad_int:2*pi
        Rz=[cos(ipsi) -sin(ipsi) 0; sin(ipsi) cos(ipsi) 0; 0 0 1];
        Ry=[cos(itheta) 0 sin(itheta); 0 1 0; -sin(itheta) 0 cos(itheta)];
        RR=Rz*Ry; % Rotation matrix.
        g1=RR*g;
        weight_force=m*g1; % AM weight vector.
        torque=cross(rg,weight_force); % Torque produce from the thrusters.
        % Reaction Torque produce from the thrusters' spin.
        reaction_torque=0.1417*(weight_force);
        wr=[-weight_force;-torque+reaction_torque];

        if bi==2
            if num_thr==7
                y=L\wr;
                thrust=U\y;
            else
                thrust=id*wr;
            end

            N=pro(thrust<0); % Finding negative propulsion efforts.

```





```

function [P1,l,lor,co,wl,P2]=prop_effort_comb(x,num_thr,normR)
max_thrust=0; % Setting max_thrust to zero.
max_res=0; % Setting max_res to zero.
f=zeros(num_thr,3); % Unit orientation vectors' matrix.
Fm=zeros(3,num_thr); % Unit orientation vectors of prop's spin.
SFr=zeros(3,num_thr);

r=zeros(num_thr+1,3); % Thruster position matrix.
F=zeros(num_thr,3); % Thruster orientation matrix.

if length(x)>2*num_thr+1 % Creating r and F matrices depending on
    for j=1:num_thr % the size of matrix x.
        r(j,:)=x(3*j-2:3*j);
        F(j,:)=x(3*(num_thr+j)-2:3*(num_thr+j));
    end
    r(num_thr+1,:)= [0 0 x(6*num_thr+1)];
else
    r=[x(1:num_thr,:); x(2*num_thr+1,:)];
    F=x(num_thr+1:2*num_thr,:);
end

% Defining the size of the matrix that will contain the combinations,
% the propulsion coefficients, and the residue.
P1=zeros(2^num_thr,num_thr+2);
% Defining the size of the matrix of the max thrust of each combination.
thrust_max=zeros(2^num_thr,1);
res_max=zeros(2^num_thr,1); % Residue matrix.
comb=dec2bin((0:2^num_thr-1)'); % Matrix of combinations.

deg_int=10; % Degrees interval.
rad_int=deg2rad(deg_int); % Rad interval.
% AM masses [thruster mass, end-effector mass, rest of structure mass].
mi=[0.1*ones(1,num_thr)';0.25;0.5+log(normR)*0.4];
m=sum(mi); % Total mass of AM.
% Centers of the individual masses.
ri=[r(1:num_thr,:);[0 0 -0.05];r(num_thr+1,:)'];
rg=ri*mi/sum(mi);

for j=1:2^num_thr

% In the next for-loop unit orientation vector(f), skew symmetric matrix of
% the unit orientation vector(SF) and the cross product matrix (SFr) are
% defined for each one of the thrusters.

for i=1:num_thr
    f(i,:)=F(i,:)/norm(F(i,:)); % Unit orientation vectors.

% Orientation of the reaction torque.
Fm(:,i)=f(i,:)'.*(1-2*bin2dec(comb(j,i)));

SF=[0 -f(i,3) f(i,2); % Skew-symmetric matrix of unit vector f.
    f(i,3) 0 -f(i,1);
    -f(i,2) f(i,1) 0 ];

SFr(:,i)=SF*r(i,:); % Cross product of f and R.

```

```

end
% Transformation matrix from actuator to end-effector space.
D=[f';SFr+Fm.*0.1417];
g=[0;0;-9.81]; % Gravity vector.

options=optimset('To1x',1e-4);

% Calculating the propulsion efforts needed in its direction.
for ipsi=0:rad_int:2*pi
    for itheta=0:rad_int:2*pi

        Rz=[cos(ipsi) -sin(ipsi) 0; sin(ipsi) cos(ipsi) 0; 0 0 1];
        Ry=[cos(itheta) 0 sin(itheta); 0 1 0; -sin(itheta) 0 cos(itheta)];
        RR=Rz*Ry; % Rotation matrix.
        g1=RR*g;
        weight_force=m*g1; % AM weight vector.
        torque=cross(rg,weight_force); % Torque produced from the thrusters.
        wr=[-weight_force;-torque];

% -----lsqnonneg-----
        [thrust]=lsqnonneg(D,wr,options);
        % Because lsqnonneg solves a minimization problem we want to ensure
        % that the solution provides the exact same forces as wr. So we
        % subtract D*(lamda) and wr. The residue of this subtraction will show how
        % close we are to the force needed.
        res=D*thrust-wr;
        res=max(abs(res));
        if max(thrust)>max_thrust % Saving the maximum propulsion effort.
            max_thrust=max(thrust);
        end
        if res>max_res % Saving the maximum residue.
            max_res=res;
        end
    end
end
res_max(j,1)=max_res;
thrust_max(j,1)=max_thrust;
max_thrust=0;
max_res=0;
end

for p=1:2^num_thr

    for k=1:num_thr
        P1(p,k)=bin2dec(comb(p,k)); % Saving the combinations.
    end
    P1(p,num_thr+1)=thrust_max(p,1); % Saving the max thrust of each combination.
    P1(p,num_thr+2)=res_max(p,1); % Saving the residue.
end
P1=[1:2^num_thr;P1']';
lor=P1(1,num_thr+2);
P2=sortrows(P1,num_thr+2);
l=P2(1,num_thr+2);
co=P2(1,2:num_thr+1);
wl=res_max(P2(1,1));

```

end

Αλγόριθμος: **res2txt.m**

```
function res2txt(num_thr,prop_size,len_cone,thr_ori,y,ses,x,fval,output,P1,prnt1,time)

% Writing the results of the optimization to a txt file.

global normR

for j=3*(num_thr)+1:3:6*num_thr-2
x(j:j+2)=x(j:j+2)/norm(x(j:j+2));
end

if prnt1==1
                                % Writing the table containing the combinations.
filename2=['tableP1_s' num2str(ses) '_run' num2str(y) '.txt'];
fid2=fopen(filename2,'wt');
txtP1=[repmat('%d\t',1,num_thr+1) '%7.4f\t%5.4e\n'];
fprintf(fid2, txtP1,P1');
fclose(fid2);
end

filename=['Results_s' num2str(ses) '_run' num2str(y) '.txt'];           % Writing
the results file.
fid=fopen(filename,'wt');

fprintf(fid, 'Optimization type : Gradient-Based Algorithm\n');
fprintf(fid, 'Number of thrusters : %d\n', num_thr);
fprintf(fid, 'Propeller size : %4.1f inches\n', prop_size);
fprintf(fid, 'Cone length : %4.3f m\n', len_cone);
if thr_ori==1
    fprintf(fid, 'Thruster orientation : inner\n');
else
    fprintf(fid, 'Thruster orientation : free\n');
end
fprintf(fid, 'Value of objective function : %7.4f\n', fval);
fprintf(fid, 'Positions Orientations\n');
fprintf(fid, '%5.3e\t%5.3e\t%5.3e\n', x);
fprintf(fid, 'Positions' norm : %7.4f\n', normR);
fprintf(fid, 'Number of iterations taken : %d\n', output.iterations);
fprintf(fid, 'Number of function evaluations : %d\n', output.funcCount);
fprintf(fid, 'Maximum of constraint functions : %5.3e\n', output.constrviolation);
fprintf(fid, 'Length of last displacement in x : %5.3e\n', output.stepsize);
fprintf(fid, 'Optimization algorithm used : %s \n', output.algorithm);
fprintf(fid, 'Total number of PCG iterations : %d \n', output.cgiterations);
fprintf(fid, 'Measure of first-order optimality : %5.3e\n', output.firstorderopt);
fprintf(fid, 'Run time (min) : %5.2f\n', time);
fprintf(fid, 'Exit message : %s', output.message);
fclose(fid);

end
```

## Παράρτημα Β.1 :

### Ανάλυση του Κώδικα APDL

Σε αυτή την παράγραφο, θα γίνει βήμα βήμα η ανάλυση και επεξήγηση του κώδικα του Ansys APDL. Το APDL (Ansys Parametric Design Language) είναι ουσιαστικά μια πλατφόρμα η οποία επιτρέπει στο χρήστη μέσω κώδικα να κάνει χρήση των λειτουργιών του πακέτου Ansys. Ο τύπος αρχείων που δέχεται το APDL ως script είναι τα αρχεία του τύπου “script.inp”. Τα αρχεία αυτά μπορούν να συνταχθούν σε ένα οποιοδήποτε editor και στη συνέχεια να αποθηκευθούν με την κατάληξη .inp.

#### Εισαγωγή Παραμέτρων

```
*DIM,R1,,11,1
*DIM,R2,,11,1
*DIM,R3,,11,1
*VREAD,R1(1,1),C:\Users\student\Desktop\truss3d\POSITION_x.txt
(F6.4)
*VREAD,R2(1,1),C:\Users\student\Desktop\truss3d\POSITION_y.txt
(F6.4)
*VREAD,R3(1,1),C:\Users\student\Desktop\truss3d\POSITION_z.txt
(F6.4)
R1(9)=0,0.125,-0.125
R2(9)=0.15,-0.075,-0.075
R3(9)=0,0,0
```

Σε αυτό το τμήμα του κώδικα εισάγουμε τους πίνακες R1,R2,R3 οι οποίοι διαβάζονται από τα αρχεία POSITION\_x,y,z.txt που έχουν παραχθεί μέσα από το προγραμματιστικό περιβάλλον Matlab. Οι πίνακες αυτοί αποτελούν τις συντεταγμένες των κόμβων του δικτυώματος. Παρακάτω ορίζουμε και τις συντεταγμένες των τριών κόμβων του end-effector τους οποίους έχουμε ορίσει εμείς. Όπου οι συντεταγμένες αυτές προκύπτουν ως τρία ισαπέχοντα σημεία ενός κύκλου με ακτίνα 15 cm και κέντρο το [0,0,0].

```
*DIM, F1, , 7, 1
*DIM, F2, , 7, 1
*DIM, F3, , 7, 1
*VREAD, F1(1), C:\Users\student\Desktop\truss3d\ORIENTATION_x.txt
(F6.4)
*VREAD, F2(1), C:\Users\student\Desktop\truss3d\ORIENTATION_y.txt
(F6.4)
*VREAD, F3(1), C:\Users\student\Desktop\truss3d\ORIENTATION_z.txt
(F6.4)
```

Σε αυτό το τμήμα σε αντιστοιχία με το προηγούμενο γίνεται η εισαγωγή των πινάκων F1,F2,F3 οι οποίοι διαβάζονται από τα αρχεία ORIENTATION\_x,y,z.txt που δημιουργούνται από το Matlab και αποτελούν τις συντεταγμένες των μοναδιαίων διανυσμάτων κατεύθυνσης κάθε δύναμης που θα εφαρμοστεί.

```
*DIM, C, , 49
*VREAD, C(1), C:\Users\student\Desktop\truss3d\ELEMENT.txt
(F1.0)
```

Κι εδώ διαβάζεται ο πίνακας C ο οποίος αποτελεί το χρωμόσωμα του δικτύωματος το οποίο εξηγήσαμε παραπάνω.

```

*DIM, CC,,55,2
CC(1,1)=1,1,1,1,1,1,1,1
CC(9,1)=1,1,2,2,2,2,2,2
CC(17,1)=2,2,2,3,3,3,3,3
CC(25,1)=3,3,3,4,4,4,4,4
CC(33,1)=4,4,5,5,5,5,5,5
CC(41,1)=6,6,6,6,6,7,7,7
CC(49,1)=7,8,8,8,9,9,10

CC(1,2)=2,3,4,5,6,7,8,9
CC(9,2)=10,11,3,4,5,6,7,8
CC(17,2)=9,10,11,4,5,6,7,8
CC(25,2)=9,10,11,5,6,7,8,9
CC(33,2)=10,11,6,7,8,9,10,11
CC(41,2)=7,8,9,10,11,8,9,10
CC(49,2)=11,9,10,11,10,11,11

```

Τέλος, δημιουργείται ο βοηθητικός πίνακας CC. Αποτελείται από 2 στήλες και γραμμές όσες και οι δυνατοί συνδυασμοί των κόμβων. Κάθε γραμμή περιέχει και έναν συνδυασμό. Παραδείγματος χάριν η γραμμή 1 περιέχει τον συνδυασμό 1-2. Αυτός ο πίνακας θα μας χρησιμεύσει για την μετέπειτα δημιουργία του δικτύωματος, διότι κάθε στοιχείο του χρωμοσώματος αντιστοιχεί σε μια γραμμή του πίνακα CC και επομένως σε ένα συνδυασμό.

### Καθορισμός Ιδιοτήτων Δικτύωματος

```

! DEFINE ELEMENT TYPE
ET,1,PIPE288
SECTYPE,1,PIPE288
SECDATA,0.01,0.0024

! DEFINE MATERIAL PROPERTIES
MP,EX,,65e9
MP,PRXY,,0.3

```

Στο ανωτέρω κομμάτι του κώδικα πραγματοποιείται η εκλογή του τύπου του στοιχείου που χρησιμοποιούμε, οι διαστάσεις τους, καθώς και το υλικό του.

```

! DEFINE NODES
N, 1, R1 (1) , R2 (1) , R3 (1)
N, 2, R1 (2) , R2 (2) , R3 (2)
N, 3, R1 (3) , R2 (3) , R3 (3)
N, 4, R1 (4) , R2 (4) , R3 (4)
N, 5, R1 (5) , R2 (5) , R3 (5)
N, 6, R1 (6) , R2 (6) , R3 (6)
N, 7, R1 (7) , R2 (7) , R3 (7)
N, 8, R1 (8) , R2 (8) , R3 (8)
N, 9, R1 (9) , R2 (9) , R3 (9)
N, 10, R1 (10) , 2 (10) , R3 (10)
N, 11, R1 (11) , R3 (11)

```

Έπειτα με την εντολή N, καθορίζουμε τις συντεταγμένες των κόμβων, από τους πίνακες R1, R2, R3 .

```

! DEFINE ELEMENT CONNECTION THRU NODES
*DO, I, 1, 49, 1
*IF, C (I) , EQ, 1, THEN
E, CC (I, 1) , CC (I, 2)
*ENDIF
*ENDDO

```

Σε αυτό το σημείο έχοντας το χρωμόσωμα στον πίνακα C και με ένα loop ελέγχουμε για ποιους συνδυασμούς κόμβων υπάρχει σύνδεση, και με την εντολή E εφαρμόζουμε μια ράβδο από τον έναν κόμβο στον άλλον.





```

! efarmogi twm dinamewn stous komvous
F,1,FX,F1(1)
F,1,FY,F2(1)
F,1,FZ,F3(1)

F,2,FX,F1(2)
F,2,FY,F2(2)
F,2,FZ,F3(2)

F,3,FX,F1(3)
F,3,FY,F2(3)
F,3,FZ,F3(3)

F,4,FX,F1(4)
F,4,FY,F2(4)
F,4,FZ,F3(4)

F,5,FX,F1(5)
F,5,FY,F2(5)
F,5,FZ,F3(5)

F,6,FX,F1(6)
F,6,FY,F2(6)
F,6,FZ,F3(6)

F,7,FX,F1(7)
F,7,FY,F2(7)
F,7,FZ,F3(7)

SOLVE ! solving ...

```

Τέλος πολλαπλασιάζουμε τους πίνακες F1, F2, F3 που περιέχουν τα μοναδιαία διανύσματα των κατευθύνσεων με 15 N , και με την εντολή F τις εφαρμόζουμε στους αντίστοιχους κόμβους.

## Παράρτημα Β.2 :

### Αλγόριθμος Δικτυώματος για Αποφυγή Λυγισμού

Παρατίθεται ο αλγόριθμος μόνο για την περίπτωση των 7 κινητήρων.

Αλγόριθμος: **buckling\_truss7.m**

```
function [sz,x,fval,exitflag,output,SS,SUM,H]=buckling_truss7(ses,run)
sz=0;
% Reading the needed information from the results file.
[x,num_thr,~,~,len_cone,fval,~,config,stat]=readresf(ses,run);
if stat==-1
    return
end
% Defining the appropriate variable and matrices to operate.
R=x(1:num_thr,:);
F=x(num_thr+1:num_thr*2,:); f=zeros(num_thr,3);
combs=nchoosek(num_thr-1,3);
S=zeros(combs,4); SS=zeros(combs,4,num_thr);
c=0; de=[]; sz=zeros(1,num_thr);

for i=1:7
    f(i,:)=F(i,:)/norm(F(i,:)); % Unit orientation vectors.
end

for p1=1:num_thr % Searching for the valid triplet combinations.
    nod=R(p1,:); % Choosing the thruster.
    pn=1:num_thr; % eliminating the thruster from the rest.
    pn(p1)=[];
    S(:,:)=ones(1,combs)*p1;nchoosek(pn,3)'; % Defining the triplet combinations.

    for p2=1:combs % Examining each triplet combination.
        j=R(S(p2,2),:)-nod;
        k=R(S(p2,3),:)-nod;
        l=R(S(p2,4),:)-nod;
        I=[j;k;l]';
        if det(I)==0
            continue
        end
        lam=I\f(p1,:);
        if lam(1)>=0 || lam(2)>=0 || lam(3)>=0
            c=c+1; % Saving the invalid combinations in order
            de(c)=p2; % to delete them.
        end
    end
    end
    sz(p1)=combs-c; % Number of valid triplet combinations.
    S(de,:)=[];
    SS(1:combs-c, :,p1)=S; % List of valid triplet combinations.
    S=zeros(combs,4);
    c=0;
    de=[];
end

if min(sz)~=0
```

```

        disp('Feasible combination')
    else
        return
    end

    lb=ones(1,num_thr);
    ub=sz;
    intcon=1:num_thr;
    opt=gaoptimset('Display','iter','UseParallel',true);
    [x,fval,exitflag,output]=ga(@(x)fobj(x,SS,num_thr,R),num_thr,[],[],[],[],lb,ub,[],intcon,opt);

    SUM=zeros(num_thr,num_thr);
    for i=1:num_thr
        SUM(i,SS(x(i),2:4,i))=SUM(i,SS(x(i),2:4,i))+1;
    end
    SUM=triu(SUM)+tril(SUM)';
    [II,JJ]=find(SUM~=0);

    H=[II';JJ']';

    mindistall(ses,run,H);
    hold on
    plot_AM(ses,run,1,2)
    axis equal
    grid on

    filename=['TrussResults_T' num2str(num_thr) upper(config(1)) num2str(len_cone) '_run'
    num2str(run) '.txt'];
    fid=fopen(filename,'wt');
    fprintf(fid, 'Buckling Oriented Truss Results\n');
    fprintf(fid, 'Number of Thrusters: %d\n',num_thr);
    fprintf(fid, 'Thruster Orientation: %s\n',config);
    fprintf(fid, 'Cone Length: %3.2f\n',len_cone);
    fprintf(fid, 'Run: %d\n',run);
    fprintf(fid, 'Active Links (Rows-Columns)\n');
    txt=[repmat('%d ',1,length(H)) '\n'];
    fprintf(fid, txt,H);
    fprintf(fid, 'Total length of truss links: %5.3f\n',fval);
    fprintf(fid, 'Number of generations : %d\n', output.generations);
    fprintf(fid, 'Number of function evaluations : %d\n', output.funccount);
    fprintf(fid, 'State of the MATLAB random number generator : %s\n',
    output.rngstate.type);
    fprintf(fid, 'Type of problem : %s \n', output.problemtype);
    fprintf(fid, 'Exit message : %s', output.message);
    fclose(fid);

end

function f=fobj(x,SS,num_thr,R)
    SUM=zeros(num_thr,num_thr);
    for i=1:num_thr
        SUM(i,SS(x(i),2:4,i))=SUM(i,SS(x(i),2:4,i))+1;
    end
end

```

```

SUM=triu(SUM)+tril(SUM)';
Norm=0;
[I,J]=find(SUM>0);
for c3=1:length(I)
Norm=Norm+norm(R(I(c3),:)-R(J(c3),:));
end
f=Norm;
end

```

Αλγόριθμος: **mindistall.m**

```

function [x,P]=mindistall(ses,run,H)

% In this algorithm we prepare various parameters that are needed for the
% optimization. First we determine if the combination of the links are
% provided or we should consider all of them. Next we calculate the
% rotation matrix of the orientation vector of each thruster. Then the
% rotation matrices, the position vectors, the unit orientation vectors,
% the active links and the number of thrusters are used as inputs to the
% solver.m algorithm in which takes place the optimization. Finally, using
% the results of the optimization we plot the optimized links.

[x,num_thr]=readresf(ses,run);
R=x(1:num_thr,:);
F=x(num_thr+1:2*num_thr,:);
RR=zeros(3,3,num_thr);
Fn=zeros(num_thr,3);
ang=zeros(num_thr,2);

if H==0
    K=nchoosek(1:num_thr,2);
else
    K=H;
end

for i=1:num_thr
    Fn(i,:)=(F(i,:)/norm(F(i,:)))*0.035; % 0.035 m is the distance from the propeller
    f=F(i,:)'; % to the mount plate of the thruster.
    if norm(R(i,:)+f')>norm(R(i,:))
        Fn(i,:)=-Fn(i,:);
    end
    xyco=[1 0 0;0 1 0;0 0 0];
    fxy=xyco*f;
    xnew=norm(fxy);
    fzxnew=[xnew 0 f(3)];

```

```

if xnew==0
    anglez=0;
else
    anglez=acos(f(1)/xnew);
end
angley=acos(xnew/norm(fzxnew));

if f(3)>0
    angley=-angley;
end

if f(2)<0
    anglez=-anglez;
end
ang(i,1:2)=[angley anglez];

Rz=[cos(ang(i,2)) -sin(ang(i,2)) 0;
sin(ang(i,2)) cos(ang(i,2)) 0;
0 0 1];
Ry=[cos(ang(i,1)) 0 sin(ang(i,1));
0 1 0;
-sin(ang(i,1)) 0 cos(ang(i,1))];
Rx=[1 0 0;
0 cos(0) -sin(0);
0 sin(0) cos(0)];
RR(:,:,i)=Rz*Ry*Rx;

end
r=R;

[x, ~]=solver(RR,r,Fn,K,num_thr);

disp(x)
P=zeros(length(K),6);
for i=1:length(K)
    ang=x(2*i-1);
    Rx=[1 0 0;
0 cos(ang) -sin(ang);
0 sin(ang) cos(ang)];
    RR1=Rx;
    t1=(RR1*[0;0.02;0]);
    ang=x(2*i);
    Rx=[1 0 0;
0 cos(ang) -sin(ang);
0 sin(ang) cos(ang)];
    RR2=Rx;
    t2=(RR2*[0;0.02;0]);

    x1=RR(:,:,K(i,1))*t1+(r(K(i,1),:)+Fn(K(i,1),:))';
    x2=RR(:,:,K(i,2))*t2+(r(K(i,2),:)+Fn(K(i,2),:))';
    P(i,:)=[x1' x2'];
end

for h=1:length(K)
    hold on
    link=[P(h,1:3);P(h,4:6)];
    plot3(link(:,1),link(:,2),link(:,3),'b','Linewidth',3);
end

```

```
end
```

Αλγόριθμος: **solver.m**

```
% In this algorithm takes place the optimization of the links. The
% objective function has two variables the one is the overall length of the
% links and the other is the overall distance of the connection points on
% each thruster. Also the optimization variables are the angles of rotation of the
% position vector of the connection point on the mount plate of the thruster.
% Connection points are the points where the the links are
% attached on the mount plate of the thruster. We want the distance of the
% connection points on the same thruster to be maximized because this would
% provide stability as the connection points would be as far as possible
% from one another.

% Optimization
function [x, fval] =solver(RR,r,f,K,n)
j=length(K)*2;
lb=zeros(1,j);
ub=ones(1,j)*2*pi;

options=gaoptimset('Display','iter','PopulationSize',500,'Generations',500,'PopInitRange',[0;2*pi],'UseParallel',true);

[x, fval] = ga(@(x)Distance(x,RR,r,K,f,j,n),j,[],[],[],[],lb,ub,[], options);

end

% Objctve Function
function f = Distance(x,RR,r,K,f,j,n)
h=j/2;
l=zeros(h,1);
P=zeros(h,6);
for i=1:h
    ang=x(2*i-1);
    Rx=[1 0      0;
```

```

        0 cos(ang)  -sin(ang);
        0 sin(ang)   cos(ang)];
    RR1=Rx;
    t1=(RR1*[0;0.02;0]);
    ang=x(2*i);
    Rx=[1 0      0;
        0 cos(ang) -sin(ang);
        0 sin(ang)  cos(ang)];
    RR2=Rx;
    t2=(RR2*[0;0.02;0]);

    x1=RR(:, :, K(i,1))*t1+(r(K(i,1),:)+f(K(i,1),:))';
    x2=RR(:, :, K(i,2))*t2+(r(K(i,2),:)+f(K(i,2),:))';
    P(i,:)=[x1' x2'];
    l(i)=norm(x1-x2);
end

for t=1:n
    [I,J]=find(K==t);
    for u=1:length(I)
        PP(u,:)=P(I(u),3*J(u)-2:3*J(u));
    end
    kh=nchoosek(1:length(I),2);
    for j=1:length(kh)
        N(j)=norm(PP(kh(j,1),:)-PP(kh(j,2),:));
    end
end
Na(t,1)=sum(N);
N=[];
end
NN=sum(Na);
f=0.5*sum(l)-0.5*NN;
end

```

## Βιβλιογραφία

- [1] G.Gavridis “Control Oriented Aerodynamic Design Optimization for an Aerial Manipulator” Diploma Thesis , NTUA 2014
- [2] A. Nikou “Analysis, Modelling and Nonlinear Control of a Novel Aerial Manipulator” Master Thesis 2014
- [3] Dario Brescianini and Raffaello D’Andrea “Design, Modeling and Control of an Omni-Directional Aerial Vehicle” 2015
- [4] X. Guo,G.D.Cheng, N.Olhoff, “optimum design of truss topology under buckling constraints” 2005
- [5]A.Torre, D.Mengoli, R.Naldi, F.Forte, A.Macchelli and L.Marconi “A prototype of Aerial Manipulator” 2012
- [6]T.R.Chandrupatla, A.D.Belegundu “Introduction to Finite Elements in Engineering” 2005
- [7]K.X Giannakoglou. “Optimization Methods for Aerodynamics” NTUA 2006
- [8] C.Castillo R.Minguez D.Ortigosa E.Castillo, A.J. Conejo. Perturbation approach to sensitivity analysys in mathematical programming. Technical report
- [9] Manolis I.A. Lourakis Theodore Papadopoulos. Estimating the jacobian of the singular value decomposition: theory and applications. Technical report 2004
- [10] A.Gauchia, B.L.Boada, M.J.L.Boada ,V.Diaz “Interration of Matlab and Ansys for Advanced Analysis of Vehicle Structures” 2014
- [11] M.Carvalho, J.L.Vale, L.T.Pinheiro, M.M.Neves “Tegchniques adopted in integrateing structural analysis with numerical optimization”
- [12] Arora, J.S.; Haug, E.J. (1979): *Methods of Design Sensitivity Analysis in Structural Optimization*. AIAA Journal
- [13] MATLAB 2016, The MathWorks Inc. 2016
- [14] ANSYS. *Structural Analysis Guide*. Ansys Inc 2011.
- [15] ANYS *Parametric Design Language Guide* 2011
- [16] ANSYS *Mechanical APDL Basic Analysis Guide* 2011
- [17] ANSYS *Mechanical APDL Command Reference* 2011
- [18] K.Liu, A.Tovar “An efficient 3D topology optimization code written in Matlab” 2014



[19] P.Sangeetha, P.Naveen Kumar, R.Senthil “ *Finite element analysis of space truss using Matlab*” 2015

[20] H.P.Gavin “*Mathematical Properties of Stiffness Matrices*” 2014

[21] Sztendel, S., Pislaru, Crinela, Longstaff, Andrew P., Myers, Alan, Fletcher, Simon and Ford, Derek G. “*Analysis of complex interactions between mechanical elements using ansys and Simulink*” 2012