

ΠΡΟΣΟΜΟΙΩΣΗ ΑΓΩΝΙΣΤΙΚΟΥ ΟΧΗΜΑΤΟΣ FORMULA STUDENT ΜΕ ΧΡΗΣΗ ΛΟΓΙΣΜΙΚΟΥ

από

Καραθανάσης Δημήτριος
Α.Μ. 18389093

Διατριβή για λήψη Βασικού Διπλώματος

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής

Πανεπιστήμιο Δυτικής Αττικής

Αιγάλεω, Μάρτιος 2024

SIMULATION OF A FORMULA STUDENT RACE CAR USING SOFTWARE

by

Karathanasis Dimitrios
R.N. 18389093

Diploma Thesis

Department of Industrial Design and Product Engineering

University of West Attica

Egaleo, March 2024

**Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του
Εισηγητή**

Η πτυχιακή/διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική
Επιτροπή:

<p>ΖΑΧΑΡΙΑ ΠΑΡΑΣΚΕΥΗ Επίκουρη Καθηγήτρια</p>	
<p>ΧΑΤΖΟΠΟΥΛΟΣ ΑΒΡΑΑΜ Λέκτορας Εφαρμογών</p>	
<p>ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ Εργαστηριακό Διδακτικό Προσωπικό (ΕΔΙΠ)</p>	

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Καραθανάσης Δημήτριος του Ιωάννη, με αριθμό μητρώου 18389093 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Πίνακας Περιεχομένων

5

Περίληψη.....	7
Κεφάλαιο 1: Εισαγωγή.....	9
Κεφάλαιο 2: Εισαγωγή στο Formula Student.....	10
1.1 Ο διαγωνισμός.....	10
1.2 Τα αγωνίσματα.....	11
1.2.1 Στατικά αγωνίσματα.....	11
1.2.2 Δυναμικά αγωνίσματα.....	12
Κεφάλαιο 3: Εισαγωγή στο MATLAB.....	18
3.1 Το λογισμικό.....	18
3.2 Στόχοι χρήσης του λογισμικού.....	18
Κεφάλαιο 4: Το όχημα και τα χαρακτηριστικά του.....	19
4.1 Μονοθέσια οχήματα.....	19
4.2 Κανονισμοί.....	20

Κεφάλαιο 5: Δομή κώδικα	29
5.1 Εισαγωγή.....	29
5.2 Γενικότερη δομή κώδικα.....	30
5.2.1 Εισαγωγή και επεξεργασία δεδομένων.....	30
5.2.2 Δημιουργία μοντέλου οχήματος.....	33
5.2.3 Δημιουργία μοντέλου πίστας.....	40
5.2.4 Η συνάρτηση simulate.....	47
5.2.5 Η συνάρτηση max_speed.....	52
5.2.6 Η συνάρτηση vehicle_model.....	56
5.2.7 Η συνάρτηση next_point.....	59
5.2.8 Η συνάρτηση other_points.....	60
5.2.9 Κώδικας πίνακα αποτελεσμάτων.....	60
Κεφάλαιο 6: Αποτελέσματα προσομοίωσης	66
6.1 Πίνακες αποτελεσμάτων.....	66
6.2 Σύγκριση αποτελεσμάτων με διαγωνισμούς.....	70
Κεφάλαιο 7: Συμπεράσματα και περιορισμοί	74
Κατάλογος Αναφορών	76

Περίληψη

Η Διπλωματική Εργασία έχει ως σκοπό την μοντελοποίηση και προσομοίωση ενός πρωτότυπου αγωνιστικού οχήματος. Το όχημα κατασκευάστηκε στα πλαίσια του διαγωνισμού Formula Student και για τον σκοπό της εργασίας θα προσομοιωθεί αποκλειστικά με τη χρήση λογισμικού. Ύστερα, τα αποτελέσματα που θα εξαχθούν από την προσομοίωση μπορούν να χρησιμοποιηθούν για την βελτιστοποίηση του οχήματος σε διάφορους τομείς.

Abstract

The objective of this Thesis is to model and simulate a prototype race car built for the purpose of participating in the Formula Student competition. The car will be simulated exclusively using software. Afterwards, the results can be used to optimize the car in various fields of performance.

Κεφάλαιο 1

Εισαγωγή

Αντικείμενο της παρούσας Διπλωματικής Εργασίας είναι η προσομοίωση ενός μονοθέσιου αγωνιστικού οχήματος, με στόχο την δημιουργία αναλυτικών αναφορών σχετικά με την συμπεριφορά του οχήματος. Η εργασία αποσκοπεί στην χρήση των αποτελεσμάτων για την πρόβλεψη της απόδοσης ενός οχήματος, χωρίς να χρειαστεί να δεσμευθούν πόροι για να κατασκευαστεί κάποιο φυσικό μοντέλο, εξοικονομώντας αρκετό χρόνο στη σχεδιαστική διαδικασία. Συμπληρωματικά, υπάρχει δυνατότητα μετατροπής των παραμέτρων του μοντέλου, κάτι που οδηγεί σε εύκολη εξαγωγή συμπερασμάτων, σε σύγκριση με ένα φυσικό μοντέλο. Αφού συζητηθεί ο τύπος και τα χαρακτηριστικά του οχήματος τα οποία θα προσομοιωθούν, θα αναλυθεί εκτενώς ο κώδικας που χρησιμοποιήθηκε. Για την διεκπεραίωση της εργασίας αναπτύχθηκε κώδικας στην προγραμματιστική γλώσσα MATLAB με χρήση εργαλείων που ήταν ήδη διαθέσιμα στο διαδίκτυο. Στα αρχικά κεφάλαια θα γίνει αναφορά στον διαγωνισμό Formula Student στον οποίον συμμετάσχουν οχήματα τέτοιου είδους, ενώ στα επακόλουθα κεφάλαια θα αναπτυχθεί ο τύπος οχήματος και ο κώδικας που χρησιμοποιήθηκε και θα σχολιαστούν τα αποτελέσματα της προσομοίωσης.

Κεφάλαιο 2

Εισαγωγή στο Formula Student

2.1 Ο διαγωνισμός

Ο διαγωνισμός Formula Student είναι ένας διεθνής διαγωνισμός στον οποίο φοιτητές από εκατοντάδες πανεπιστήμια παγκοσμίως καλούνται να σχεδιάσουν, να κατασκευάσουν και να τρέξουν με αγωνιστικά μονοθέσια. Οι διαγωνισμοί λαμβάνουν χώρα σε πολλές χώρες σε όλες τις ηπείρους κατά την διάρκεια των καλοκαιρινών μηνών κάθε έτους. Εκεί, κάθε ομάδα διαγωνίζεται και βαθμολογείται ανάλογα με την εργασία της. Ο διαγωνισμός δεν κερδίζεται από την ομάδα με το πιο γρήγορο μονοθέσιο, αλλά με τον καλύτερο συνδυασμό κατασκευής, απόδοσης και οικονομικού σχεδιασμού που μπορεί να πραγματοποιήσει η εκάστοτε ομάδα. Επιπροσθέτως, οι ομάδες καλούνται να αποδείξουν στην πράξη τα αγωνιστικά χαρακτηριστικά του μονοθέσιου μέσα από μία σειρά δυναμικών αγωνισμάτων τα οποία επιδεικνύουν την επιτάχυνση, επιβράδυνση και απόδοση στις στροφές. Όλα τα αγωνίσματα ενός διαγωνισμού γίνεται να προσφέρουν έως 1000 πόντους στην ομάδα. Νικήτρια είναι η ομάδα που θα βαθμολογηθεί με το μεγαλύτερο σύνολο πόντων σε όλα τα αγωνίσματα.

2.2 Τα αγωνίσματα

Τα αγωνίσματα χωρίζονται σε 2 γενικές κατηγορίες: Τα Static Events (στατικά αγωνίσματα), στα οποία βαθμολογείται η προσπάθεια της ομάδας στον σχεδιασμό, τον

διαμοιρασμό πόρων, την κατασκευή και την επιχειρηματικότητα της ομάδας στην αγορά. Τα Dynamic Events (δυναμικά αγωνίσματα), στα οποία το όχημα καλείται να τρέξει σε πίστες οριοθετημένες από τον διαγωνισμό με συγκεκριμένες προδιαγραφές. Σε αυτό το σημείο οφείλεται να αναφερθεί ότι για να αγωνιστεί το οποιοδήποτε όχημα πανεπιστημίου στα δυναμικά αγωνίσματα, αρχικά περνάει εκτενή έλεγχο από τους κριτές του διαγωνισμού για να βεβαιωθεί ότι πληροί όλους τους κανονισμούς. Οι συγκεκριμένοι κανονισμοί είναι αναρτημένοι πάντα στην ιστοσελίδα του αντίστοιχου διαγωνισμού. Οι κανονισμοί ενδέχεται να διαφέρουν από τον διαγωνισμό της μίας χώρας με άλλης, όπως και μίας χρόνιας με άλλης. Η πλειοψηφία των διαγωνισμών που διεξάγονται στην Ευρώπη υπάγονται στους κανονισμούς της διοργάνωσης του Formula Student Germany, και ακολουθούν τους κανόνες που αναρτώνται κάθε χρόνο στην ομόνυμη ιστοσελίδα με πολύ λίγες αλλαγές.

2.2.1 Στατικά αγωνίσματα

Σύμφωνα με τους κανονισμούς του Formula Student Germany για το έτος 2024, τα στατικά αγωνίσματα χωρίζονται ως εξής:

- Business Plan Presentation (Παρουσίαση Επιχειρηματικού Σχεδίου) με μέγιστο έπαθλο 75 πόντους
- Engineering Design (Μηχανική Σχεδίαση) με μέγιστο έπαθλο 150 πόντους
- Cost and Manufacturing (Κόστος και Κατασκευή) με μέγιστο έπαθλο 100 πόντους.

Η ομάδα που παίρνει την 1^η θέση στα παραπάνω αγωνίσματα κερδίζει το μέγιστο ποσό πόντων.

2.2.2 Δυναμικά αγωνίσματα

Στην παρούσα εργασία έμφαση θα δοθεί στα δυναμικά αγωνίσματα και στην προσομοίωση αυτών, οπότε χρειάζεται περαιτέρω εμβάθυνση στην βαθμολόγηση και τους περιορισμούς τους.

Τα τέσσερα δυναμικά αγωνίσματα ενός διαγωνισμού Formula Student είναι:

- Acceleration (Επιτάχυνση) με μέγιστο έπαθλο 50 πόντους
- Skidpad (Ειδική πίστα με σχήμα «8») με μέγιστο έπαθλο 50 πόντους
- Autocross (Ειδική πίστα ενός γύρου) με μέγιστο έπαθλο 100 πόντους
- Endurance (Πίστα αντοχής περίπου 25 γύρων) με μέγιστο έπαθλο 250 πόντους

Η διαδικασία του Acceleration έχει ως εξής: το όχημα ξεκινάει από την γραμμή εκκίνησης και πρέπει επιταχύνοντας να διανύσει μια απόσταση 75 μέτρων από την εκκίνηση. Ο χρόνος μετράει μέχρι να περάσει την γραμμή τερματισμού. Κάθε ομάδα έχει δύο προσπάθειες για το αγώνισμα, με τον ίδιο οδηγό. Η ομάδα έχει το δικαίωμα να φέρει εις πέρας την δεύτερη προσπάθεια της αμέσως μετά την πρώτη. Η βαθμολογία του Acceleration υπολογίζεται ως εξής:

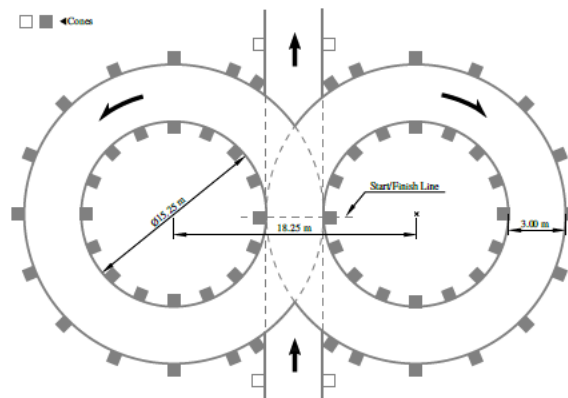
$$M_ACCELERATION_SCORE = 0.95 P_{\max} \left(\frac{T_{\max} - 1}{T_{\text{team}} - 0.5} \right)$$

Εικόνα 1: Η βαθμολόγηση του Acceleration όπως εμφανίζεται στο βιβλιάριο κανονισμών του FSG.

Όπου P_{\max} είναι οι μέγιστοι πόντοι που μπορεί να πάρει η ομάδα στο αγώνισμα, T_{\max} ο χρόνος της γρηγορότερης ομάδας επί 1,5 (συν τυχόν ποινές που έχει δεχθεί η ομάδα) και T_{team} ο χρόνος της ομάδας συμπεριλαμβανομένων των ποινών τους. Επιπροσθέτως, το 5% των μέγιστων πόντων που μπορούν να βραβευτούν για το αγώνισμα δίνεται σε όποια ομάδα τερματίσει τουλάχιστον σε μία προσπάθεια χωρίς DNF (did not finish, δηλαδή μη τερματισμός λόγω κάποιου θέματος) ή DSQ (αποκλεισμός λόγω

παραβίασης κανονισμών).

Το αγώνισμα του Skidpad είναι από τις πιο απαιτητικές όσον αφορά τις ικανότητες του οχήματος κατά τη στροφή: το όχημα καλείται να διασχίσει δύο κύκλους ακτίνας 9,125 μέτρων όσο πιο γρήγορα μπορεί. Οι πλευρικές επιταχύνσεις που δέχεται το όχημα ξεπερνάνε συχνά το 1G ($9,81\text{m/s}^2$) οπότε οι ομάδες πρέπει να βρουν την χρυσή τομή ανάμεσα στην επιτάχυνση στις ευθείες και την ευκινησία στις στροφές. Όπως και στο Acceleration, η ομάδα μπορεί να κάνει και δεύτερη προσπάθεια αμέσως μετά την πρώτη της, με τον ίδιο οδηγό.



Εικόνα 2: Η πίστα του Skidpad, όπως εμφανίζεται στο βιβλιάριο κανονισμών FSG.

Η βαθμολογία του Skidpad υπολογίζεται ως εξής:

$$M_SKIDPAD_SCORE = 0.95 P_{\max} \left(\frac{\left(\frac{T_{\max}}{T_{\text{team}}} \right)^2 - 1}{0.5625} \right)$$

Εικόνα 3: Η βαθμολόγηση του Skidpad όπως εμφανίζεται στο βιβλιάριο κανονισμών FSG.

Όπου P_{\max} είναι οι μέγιστοι πόντοι που μπορεί να πάρει η ομάδα στο αγώνισμα, T_{\max} ο χρόνος της γρηγορότερης ομάδας επί 1,25 (συν τυχόν ποινές που έχει δεχθεί η ομάδα) και T_{team} ο χρόνος της ομάδας συμπεριλαμβανομένων των ποινών τους. Επιπροσθέτως, το 5% των μέγιστων πόντων που μπορούν να βραβευτούν για το αγώνισμα δίνεται σε όποια ομάδα τερματίσει τουλάχιστον σε μία προσπάθεια χωρίς DNF (did not finish,

δηλαδή μη τερματισμός λόγω κάποιου θέματος) η DSQ (αποκλεισμός λόγω παραβίασης κανονισμών).

Η πίστα του Autocross είναι μια πίστα σχεδιασμένη για να ελέγχει τον έλεγχο του αμαξιού. Κάθε πίστα autocross έχει συγκεκριμένες προδιαγραφές. Σύμφωνα με τους κανονισμούς του Formula Student Germany 2024, αυτές είναι:

- Οι ευθείες είναι το πολύ 80 μέτρα σε μήκος
- Οι συνεχείς στροφές φτάνουν τα 50 μέτρα σε διάμετρο.
- «Φουρκέτες» με εξωτερική διάμετρο 9 μέτρα.
- Υπάρχουν σλάλομ κατασκευασμένα από κώνους με απόσταση από 7,5 έως 12 μέτρα
- Επίσης συμπεριλαμβάνονται στροφές τύπου «chicane» (σχήμα S), αλληπάλληλες στροφές, στροφές με φθίνουσα ακτίνα, κ.α..
- Το ελάχιστο πλάτος της πίστας είναι τα 3 μέτρα.

Το ολικό μήκος της πίστας είναι λιγότερο από 1,5 χιλιόμετρο. Κάθε ομάδα έχει έως 4 προσπάθειες, με 2 οδηγούς να πραγματοποιούν 2 προσπάθειες ο καθένας. Ο κάθε οδηγός έχει την δυνατότητα να επιχειρήσει την δεύτερη προσπάθεια του αμέσως μετά την πρώτη. Για τον πρώτο γύρο, το όχημα ξεκινάει από μία δευτερεύουσα γραμμή η οποία βρίσκεται πίσω από την γραμμή εκκίνησης – τερματισμού. Ο χρόνος μετράει μόνο αφού το όχημα προσπεράσει την γραμμή εκκίνησης.

Η βαθμολογία του Autocross υπολογίζεται ως εξής:

$$AUTOCROSS_SCORE = 0.95 P_{max} \left(\frac{\frac{T_{max}}{T_{team}} - 1}{0.25} \right)$$

Εικόνα 4: Η βαθμολόγηση του Autocross όπως εμφανίζεται στο βιβλιάριο κανονισμών του FSG.

Όπου P_{max} είναι οι μέγιστοι πόντοι που μπορεί να πάρει η ομάδα στο αγώνισμα, T_{max} ο

χρόνος της γρηγορότερης ομάδας επί 1,25 (συν τυχόν ποινές που έχει δεχθεί η ομάδα) και T_{team} ο χρόνος της ομάδας συμπεριλαμβανομένων των ποινών τους. Επιπροσθέτως, το 5% των μέγιστων πόντων που μπορούν να βραβευτούν για το αγώνισμα δίνεται σε όποια ομάδα τερματίσει τουλάχιστον σε μία προσπάθεια χωρίς DNF (did not finish, δηλαδή μη τερματισμός λόγω κάποιου θέματος) ή DSQ (αποκλεισμός λόγω παραβίασης κανονισμών).

Τελευταίο είναι το αγώνισμα του Endurance (αντοχής) και Efficiency (απόδοση), το οποίο θεωρείται και το πιο απαιτητικό. Η χάραξη της πίστας του Endurance ακολουθεί προδιαγραφές παρόμοιες με του Autocross:

- Οι ευθείες είναι το πολύ 80 μέτρα σε μήκος
- Οι συνεχείς στροφές φτάνουν τα 50 μέτρα σε διάμετρο.
- «Φουρκέτες» με εξωτερική διάμετρο 9 μέτρα.
- Υπάρχουν σλάλομ κατασκευασμένα από κώνους με απόσταση από 9 έως 15 μέτρα
- Επίσης συμπεριλαμβάνονται στροφές τύπου «chicane» (σχήμα S), αλλεπάλληλες στροφές, στροφές με φθίνουσα ακτίνα, κ.α..
- Το ελάχιστο πλάτος της πίστας είναι τα 3 μέτρα.

Ένας γύρος του Endurance φτάνει περίπου το 1 χιλιόμετρο. Το ολικό μήκος του Endurance είναι περίπου 22 χιλιόμετρα. Ως αποτέλεσμα, οι ομάδες καλούνται να φτιάξουν ένα όχημα που θα αντέξει πάνω από 20 γύρους σε συνθήκες αγώνα, καθώς όλοι οι γύροι χρονομετρούνται. Όπως και στο Autocross, το αμάξι στήνεται σε μία γραμμή πριν την γραμμή εκκίνησης – τερματισμού, και ο χρόνος μετράει όταν περάσει την γραμμή εκκίνησης στον πρώτο γύρο. Κάθε ομάδα έχει μόνο μια προσπάθεια στο Endurance. Οδηγούν δύο οδηγοί. Ο πρώτος οδηγός για τα πρώτα 11 χιλιόμετρα, και

στη συνέχεια θα πρέπει να μεταβεί στην ειδική περιοχή αλλαγής οδηγού για να αλλάξει οδηγό το όχημα για τα υπόλοιπα έντεκα. Το Endurance και το Efficiency αντιμετωπίζονται ως ένα αγώνισμα.

Η βαθμολογία του Endurance υπολογίζεται ως εξής:

$$ENDURANCE_SCORE = 0.9 P_{max} \left(\frac{T_{max} - 1}{0.333} \right)$$

Εικόνα 5.1: Η βαθμολόγηση του Endurance όπως εμφανίζεται στο βιβλιάριο κανονισμών FSG.

Όπου P_{max} είναι οι μέγιστοι πόντοι που μπορεί να πάρει η ομάδα στο αγώνισμα, T_{max} ο χρόνος της γρηγορότερης ομάδας επί 1,333 (συν τυχόν ποινές που έχει δεχθεί η ομάδα) και T_{team} ο χρόνος της ομάδας συμπεριλαμβανομένων των ποινών τους. Επιπροσθέτως, το 10% των μέγιστων πόντων που μπορούν να βραβευτούν για το αγώνισμα δίνεται σε όποια ομάδα τερματίσει τουλάχιστον σε μία προσπάθεια χωρίς DNF (did not finish, δηλαδή μη τερματισμός λόγω κάποιου θέματος) ή DSQ (αποκλεισμός λόγω παραβίασης κανονισμών).

Το Efficiency μετράει την απόδοση του οχήματος όσον αφορά την κατανάλωση ενέργειας. Για να ληφθεί υπόψιν ένα όχημα για το αγώνισμα του Efficiency, πρέπει πρώτα να έχει λάβει βαθμούς στο Endurance. Δεύτερον, ο χρόνος του οχήματος στο Endurance (χωρίς ποινές) θα πρέπει να είναι λιγότερος από τον γρηγορότερο χρόνο όλων των ομάδων στην κατηγορία επί 1,333. Η ενέργεια που καταναλώθηκε στο Endurance μετράται ως το ολοκλήρωμα της συνολικής ισχύος του οχήματος, όπως μετρήθηκε από τους αισθητήρες που παρέχει ο διαγωνισμός. Η βαθμολογία του Efficiency προκύπτει από την εξίσωση:

$$EFFICIENCY_SCORE = P_{\max} \left(\frac{EF_{\max} - EF_{team}}{EF_{\max} - EF_{\min}} \right)$$

Εικόνα 5.2: Υπολογισμός βαθμολογίας Efficiency

Όπου P_{\max} οι μέγιστοι πόντοι για το αγώνισμα, EF_{\max} ο μεγαλύτερος συντελεστής απόδοσης όλων των ομάδων, EF_{team} ο συντελεστής απόδοσης της ομάδας και EF_{\min} ο μικρότερος συντελεστής απόδοσης όλων των ομάδων. Ο συντελεστής απόδοσης κάθε ομάδας υπολογίζεται ως εξής:

$$EF = T^2 \cdot E$$

Εικόνα 5.3: Υπολογισμός συντελεστή απόδοσης

Όπου T ο χρόνος κάθε ομάδας και E η ενέργεια που καταναλώθηκε.

Η πλειοψηφία των διαγωνισμών αναρτεί σε ηλεκτρονική μορφή τα αποτελέσματα για κάθε δυναμικό αγώνισμα στην αντίστοιχη ιστοσελίδα τους. Οποιοσδήποτε μπορεί να δει και να συγκρίνει αποτελέσματα των ομάδων από ένα μεγάλο εύρος ετών που έγιναν οι διαγωνισμοί. Με όλες αυτές τις πληροφορίες διαθέσιμες, αρκετά γρήγορα γίνεται προφανές ότι εάν κάποια ομάδα θέλει να διασφαλίσει την επιτυχία της στον διαγωνισμό, πρέπει με κάποιο αξιόπιστο τρόπο να προβλέψει την απόδοση της στα αγωνίσματα αυτά, χρησιμοποιώντας κάποιου είδους προσομοίωση για τους χρόνους της.

Κεφάλαιο 3

Εισαγωγή στο MATLAB

3.1 Το λογισμικό

Το MATLAB είναι προγραμματιστικό περιβάλλον της ομώνυμης γλώσσας προγραμματισμού, που παρέχεται από την Mathworks από το 1984 μέχρι και σήμερα. Είναι εξαιρετικά δημοφιλές σε μηχανικούς, λόγω της ενσωμάτωσης του με αλγεβρικούς πίνακες, κάτι που το καθιστά πολύ εύχρηστο για μαθηματικούς υπολογισμούς και μοντέλα. Από την χρήση των πινάκων εκείνων προκύπτει και το όνομα του προγράμματος, MATLAB (Matrix Laboratory – Εργαστήριο Πινάκων). Σε αυτό το σημείο αξίζει να σημειωθεί ότι το MATLAB επιλύει τα προβλήματα με αριθμητική πεπερασμένης ακρίβειας, δηλαδή αντί για την ακριβή λύση, το πρόγραμμα προσεγγίζει την λύση ενός προβλήματος. Συμπληρωματικά, το λογισμικό του MATLAB προσφέρει δυνατότητες γραφικής παρουσίασης των αποτελεσμάτων που παράχθηκαν με πληθώρα τρόπων, ιδιότητα που χρησιμοποιήθηκε κατά την εκπόνηση αυτής της εργασίας, έτσι ώστε να γίνει αποτελεσματικά και κατανοητά η εμφάνιση των αποτελεσμάτων.

3.2 Στόχοι χρήσης του λογισμικού

Στόχος της εργασίας είναι η χρήση του MATLAB για την ανάπτυξης ενός αλγορίθμου που θα ικανοποιεί τις ανάγκες της προσομοίωσης. Συγκεκριμένα, θα πρέπει να:

- Εισάχθουν δεδομένα σχετικά με το αυτοκίνητο και την πίστα
- Δημιουργηθεί ένα μοντέλο του οχήματος από τα δεδομένα
- Δημιουργηθούν αποτελέσματα σχετικά με τον χρόνο περάσματος (lap time) και την κατανάλωση ενέργειας (για χρήση στο αγώνισμα του Efficiency).

Συγκεκριμένα, απαιτείται η χρήση ενός αλγορίθμου που υπολογίζει την κατάσταση του οχήματος για κάθε θέση μέσα στην πίστα, και η εύρεση του βέλτιστου συνδυασμού καταστάσεων, έτσι ώστε να υπολογιστεί ο χαμηλότερος χρόνος περάσματος.

Λαμβάνοντας υπόψιν χαρακτηριστικά του οχήματος όπως η μάζα, οι συντελεστές τριβής των ελαστικών, η οπισθέλκουσα κ.α. γίνεται με χρήση φυσικής και αλγεβρικών εξισώσεων να υπολογιστεί η ταχύτητα του για κάθε θέση του μέσα στην πίστα. Όλες οι πιθανές θέσεις του οχήματος στην πίστα μπορούν να οριστούν από ένα πλέγμα (mesh) το οποίου η πυκνότητα ορίζεται από μεταβλητή στον κώδικα.

Ο συνδυασμός αυτός των χαρακτηριστικών του MATLAB το καθιστά ιδανικό εργαλείο για να την ανάπτυξη μίας προσομοίωσης περάσματος γύρου (laptime simulation).

Κεφάλαιο 4

Το όχημα και τα χαρακτηριστικά του

4.1 Μονοθέσια

Όλα τα οχήματα που συμμετέχουν στον διαγωνισμό Formula Student πρέπει να είναι τύπου formula. Συγκεκριμένα, αυτά τα οχήματα έχουν εκτεθειμένο το κεφάλι του οδηγού (συνήθως αναφέρεται και ως πιλότος από το ιταλικό pilota), σε διάταξη ανοικτού πιλοτηρίου (cockpit) όπως και τους 4 τροχούς τους (open wheel). Κατά την συντριπτική τους πλειοψηφία έχουν τους πίσω τροχούς ως κινητήριους (πισωκίνητα), ενώ πάντα ο κινητήρας βρίσκεται πίσω από την θέση του οδηγού, στη μέση του αμαξώματος, για λόγους ορατότητας. Βάσει ονόματος, αυτά τα οχήματα δεν είναι επιβατηγά, καθώς προορίζονται μόνο για αγωνιστική χρήση σε ειδικές πίστες.

Παρέχουν λοιπόν, μονάχα μία θέση για τον οδηγό. Οι βασικοί αυτοί περιορισμοί αναφέρονται στο βιβλιάριο κανονισμών Formula Student Germany.

4.2 Κανονισμοί

Υπάρχουν πρόσθετοι περιορισμοί που επιβάλλονται στις ομάδες Formula Student από τον διαγωνισμό. Συγκεκριμένα, αυτοί οι κανονισμοί διέπουν τις διαδικασίες εγγραφής της ομάδας στον διαγωνισμό, την συμπεριφορά όλων των μελών, τα έγγραφα που πρέπει να κατατεθούν, τα κατασκευαστικά χαρακτηριστικά ολόκληρου του οχήματος και την βαθμολόγηση των ομάδων, όπως αναφέρθηκε προηγουμένως. Το εκτενέστερο κομμάτι του βιβλίου κανονισμών πραγματεύεται τα κατασκευαστικά χαρακτηριστικά, από τις διαστάσεις όλου του οχήματος μέχρι και τις προδιαγραφές που πρέπει να χρησιμοποιούνται για βίδες του οχήματος σε συγκεκριμένα υποσυστήματα.

Το βιβλίο των κανονισμών έχει έκταση γύρω στις 130 σελίδες. Για τις ανάγκες αυτής της εργασίας, θα γίνει αναφορά μόνο στους κανονισμούς που αφορούν είτε τα αγωνίσματα, είτε το ίδιο το όχημα.

Από το 2024 και μετά, όπως αναφέρεται στο handbook του διαγωνισμού Formula Student Germany (έγγραφο ξεχωριστό από το βιβλιάριο κανονισμών, το οποίο αφορά αποκλειστικά και μόνο τον διαγωνισμό που γίνεται στην Γερμανία) ο διαγωνισμός θα δέχεται μόνο ηλεκτροκίνητα μονοθέσια. Η πλειοψηφία των διαγωνισμών δέχεται και οχήματα με μηχανές εσωτερικής καύσης, αλλά όποια ομάδα επιθυμεί να λάβει μέρος στο Formula Student Germany, που θεωρείται ο πιο απαιτητικός διαγωνισμός παγκοσμίως, είναι περιορισμένη σε ηλεκτροκίνητα μονοθέσια. Στην εργασία αυτή, θα προσομοιωθεί ένα ηλεκτροκίνητο μονοθέσιο.



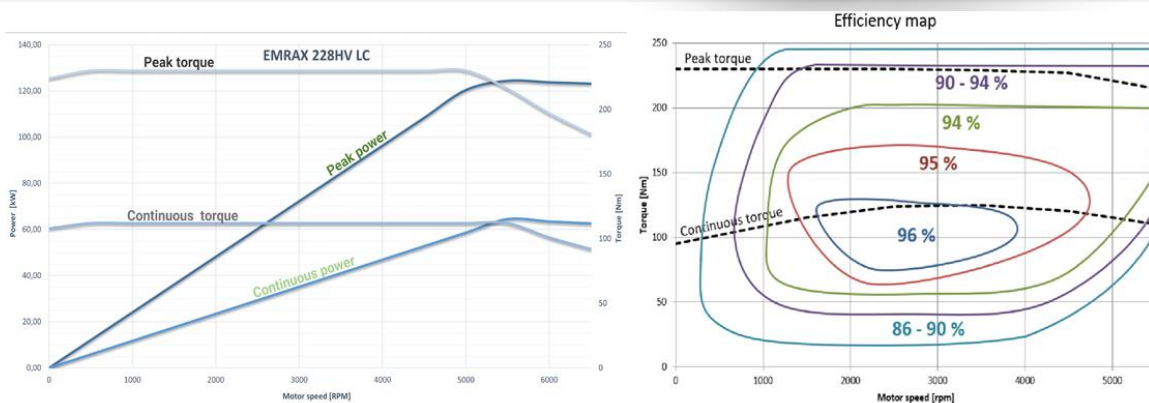
Εικόνα 6: Ηλεκτρικό μονοθέσιο Formula Student στον διαγωνισμό Formula Student East

Ηλεκτροκινητήρας

Σύμφωνα με τους κανονισμούς για το έτος 2024, οι ηλεκτροκινητήρες έχουν ανώτατο όριο ισχύος τα 80Kw. Ένας ηλεκτροκινητήρας όπως ο Emrax 228, είναι δημοφιλής επιλογή για πολλές ομάδες καθώς καλύπτει ικανοποιητικά τις ανάγκες του διαγωνισμού. Έχει βάρος γύρω στα 13 κιλά, ισχύ στα 75kw, συνεχής ροπή γύρω στα 130Nm (διαφέρει από αμάξι σε αμάξι λόγω κατασκευής) και διάμετρο 22,8 εκατοστών. Είναι εξαιρετική επιλογή για εφαρμογές με μέτριες προς υψηλές απαιτήσεις ενέργειας, με μικρές χωροταξικές απαιτήσεις. Στην παρακάτω εικόνα φαίνονται οι πληροφορίες του κινητήρα όπως είναι αναρτημένες στην ιστοσελίδα της Emrax:

EMRAX 228	
DIAMETER LENGTH	228 mm 86 mm
WEIGHT	12,9-13,5 kg
COOLING	air / water / combined
PEAK CONTINUOUS POWER	124 kW 75 kW*
PEAK CONTINUOUS TORQUE	230 Nm 130 Nm*
MAXIMUM SPEED	6500 RPM
OPERATING VOLTAGE	50 - 830 V
EFFICIENCY	up to 96%*
POSITION SENSOR	resolver / encoder

*Subject to motor configuration, drive cycle, thermal conditions, and controller capability.

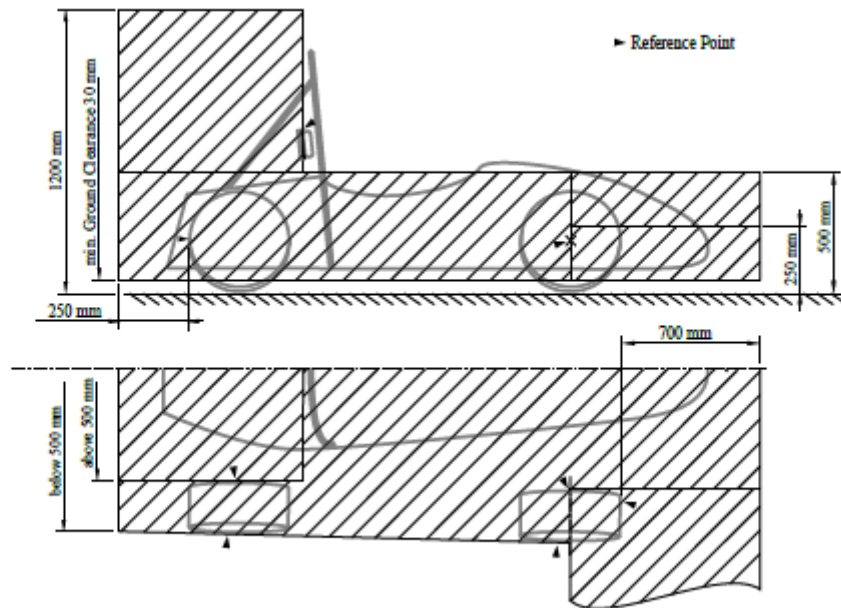
Εικόνα 7: Ηλεκτροκινητήρας Emrax 228

Για τις ανάγκες αυτής της εργασίας, θα προσομοιωθούν τα χαρακτηριστικά αυτού του ηλεκτροκινητήρα για το όχημα.

Αεροδυναμική

Αεροδυναμικά στοιχεία όπως αεροτομές και πατώματα που χειραγωγούν την ροή του αέρα κάτω από τα οχήματα ήταν ανέκαθεν κρίσιμα στοιχεία κάθε μονοθέσιου. Στον διαγωνισμό Formula Student, όλα τα (αεροδυναμικά και μη) στοιχεία του αυτοκινήτου πρέπει να είναι σε ύψος τουλάχιστον 30 χιλιοστών από το έδαφος (εκτός από τους τροχούς) όταν το όχημα είναι σε ακινησία και ο οδηγός είναι στη θέση του.

Επιπροσθέτως, όλα τα αεροδυναμικά στοιχεία πρέπει να βρίσκονται στην παρακάτω περιοχή, όπως απεικονίζεται στο βιβλιάριο κανονισμών.



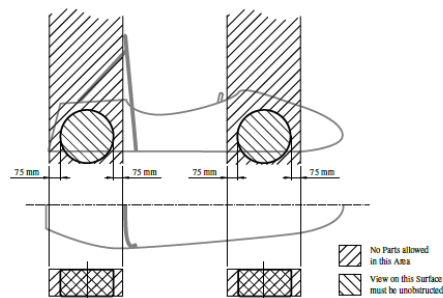
Εικόνα 8: Τα μέγιστα όρια αεροδυναμικών στοιχείων ενός οχήματος Formula Student, σύμφωνα με το βιβλιário κανονισμών για το έτος 2024

Υπάρχει μεγάλος αριθμός διατάξεων και συνδυασμών αεροδυναμικών στοιχείων που μπορεί να συνδυάσει κάποια ομάδα για να επιφέρει το επιθυμητό αποτέλεσμα στην απόδοση του οχήματος της. Στην παρούσα εργασία, θα γίνει προσομοίωση οχήματος με πλήρες αεροδυναμικό πακέτο, και κατανομή 45/55. Αυτό σημαίνει ότι τα αεροδυναμικά φορτία θα είναι κατανομημένα 45% στο μπροστά μέρος του οχήματος, και 55% προς το πίσω. Ένα πλήρες αεροδυναμικό πακέτο περιλαμβάνει τουλάχιστον μπροστά και πίσω «φτερό» και μια συσκευή που κατευθύνει την ροή του αέρα κάτω από το όχημα, γνωστή ως διαχύτης (diffuser). Ο συνδυασμός των στοιχείων αυτών, μαζί με το υπόλοιπο αμάξι, παράγουν σημαντικά φορτία αρνητικής άντωσης, τα οποία συνήθως μετρούνται σε κιλά, όταν το αμάξι πηγαίνει με 60 χιλιόμετρα την ώρα, και «σπρώχνουν» το αμάξι προς τα κάτω. Ως αποτέλεσμα, το αμάξι μπορεί να αναπτύξει μεγαλύτερες πλευρικές δυνάμεις, δηλαδή να διανύσει μία στροφή με μεγαλύτερη ταχύτητα. Το μειονέκτημα είναι ότι τα στοιχεία αυτά αυξάνουν την επιφάνεια του αυτοκινήτου που έρχεται σε επαφή με τον αέρα, αυξάνοντας την αντίσταση του αέρα,

γνωστή και ως οπισθέλκουσα. Ένα όχημα με πλήρες αεροδυναμικό πακέτο έχει συνήθως εμβαδόν επιφάνειας 1 τετραγωνικό μέτρο, συντελεστή άντωσης που ξεπερνά το -2 και συντελεστή οπισθέλκουσας 0,8. Αυτή η τιμή θα χρησιμοποιηθεί και στην προσομοίωση.

Ελαστικά

Δεν υπάρχει κάποια δέσμευση σχετικά με το μέγεθος ή τον κατασκευαστή ελαστικού που μπορεί να φέρει ένα όχημα. Προφανώς, τα ελαστικά πρέπει να ταιριάζουν στην αντίστοιχη ζάντα τους. Όμως οι ομάδες είναι υποχρεωμένες να αφήνουν 75 χιλιοστά κενό χώρο μπροστά και πίσω από κάθε ελαστικό.



Εικόνα 9: Σχεδιάγραμμα του αδέσμευτου χώρου σύμφωνα με το βιβλιάριο κανονισμών για το 2024
Αξίζει επίσης να σημειωθεί ο κανονισμός T.2.9.1: Το όχημα πρέπει να έχει μεταξόνιο μήκους τουλάχιστον 1525 χιλιοστά. Μεταξόνιο είναι η απόσταση που μετρείται από τον μπροστά άξονα μέχρι τον πίσω άξονα του οχήματος. Γενικότερα, οι ομάδες προσπαθούν να προσεγγίσουν το ελάχιστο μεταξόνιο, καθώς ένα αμάξι με μικρότερο μεταξόνιο είναι μικρότερο, άρα φθηνότερο, και ελαφρύτερο. Ο συνδυασμός αυτών των δύο απαιτήσεων έχει ως αποτέλεσμα τις ομάδες να καταλήγουν σε ζάντες 13 ιντσών, με τα αντίστοιχα ελαστικά που ταιριάζουν. Υπάρχουν ομάδες που με βελτιστοποίηση μείωσαν το μέγεθος (αρά και το βάρος) των ζαντών σε 10 ίντσες. Για τις ανάγκες αυτής της εργασίας, θα χρησιμοποιηθεί ένα ελαστικό με διάμετρο 260,34 χιλιοστά, που

ταιριάζει

δηλαδή σε ζάντα 10 ιντσών (254 χιλιοστά).



Εικόνες 10 & 11: Οχήματα Formula Student με ζάντες διαμέτρου 13 ιντσών και 10 ιντσών αντίστοιχα

Εκτός από το μέγεθος του ελαστικού, ένα από τα πιο σημαντικά μεγέθη είναι ο συντελεστής τριβής των ελαστικών. Καθώς το ελαστικό είναι το μόνο μέρος του οχήματος που έρχεται σε επαφή με την άσφαλτο, και χρησιμοποιεί την τριβή για να διατηρεί το όχημα στην πορεία του, τα ελαστικά είναι τα σημαντικότερα στοιχεία του οχήματος που μπορούν να μελετηθούν για να προβλεφθεί την απόδοση του, μετά τον κινητήρα του. Τα ελαστικά έχουν διαφορετικό συντελεστή τριβής για διαμήκης (δηλαδή δυνάμεις που ασκούνται κατά μήκος του ελαστικού, από εμπρός προς τα πίσω) και πλευρικές δυνάμεις, ο οποίος μεταβάλλεται ανάλογα με την θερμοκρασία και τις φορτίσεις που δέχονται τα ελαστικά. Για να παραμείνει το μοντέλο απλό, θέτεται ένας σταθερό συντελεστή τριβής 2 για διαμήκης δυνάμεις και 2 για πλευρικές δυνάμεις. Ακόμα ένας συντελεστής είναι η αντίσταση του ελαστικού στην κύλιση. Υπάρχει σε κάθε ελαστικό, και προκαλείται από την ίδια την ελαστικότητα του ελαστικού. Σύμφωνα με τον Thomas Gillespie, είναι ένας συντελεστής της τάξης του χιλιοστού της μονάδας. Όσο μεγαλύτερος ο συντελεστής, τόσο μεγαλύτερη απώλεια ενέργειας θα υπάρχει κατά την κίνηση του ελαστικού.

Μετάδοση

Για να μεταδοθεί η κίνηση από τον κινητήρα στους τροχούς, χρησιμοποιούνται γρανάζια, αλυσίδες και διαφορικά για να υπάρχει ο βέλτιστος συνδυασμός ροπής και τελικής ταχύτητας. Θα χρησιμοποιηθεί ένας ολικός λόγος μείωσης για τις ανάγκες αυτής της εργασίας, που θα είναι 4. Η μείωση αυτή είναι δημοφιλής επιλογή σε πολλές ομάδες Formula Student, καθώς οι ανάγκες για υψηλή τελική ταχύτητα είναι περιορισμένες, διότι η μακρύτερη ευθεία θα το πολύ 80 μέτρα. Στην προσομοίωση του οχήματος, θα φανεί ότι η τελική ταχύτητα του οχήματος είναι σχετικά χαμηλή (μικρότερη από 140 χιλιόμετρα την ώρα), αλλά η επιτάχυνση είναι αρκετά ισχυρή λόγω του υψηλού λόγου μείωσης.

Βάρος οχήματος

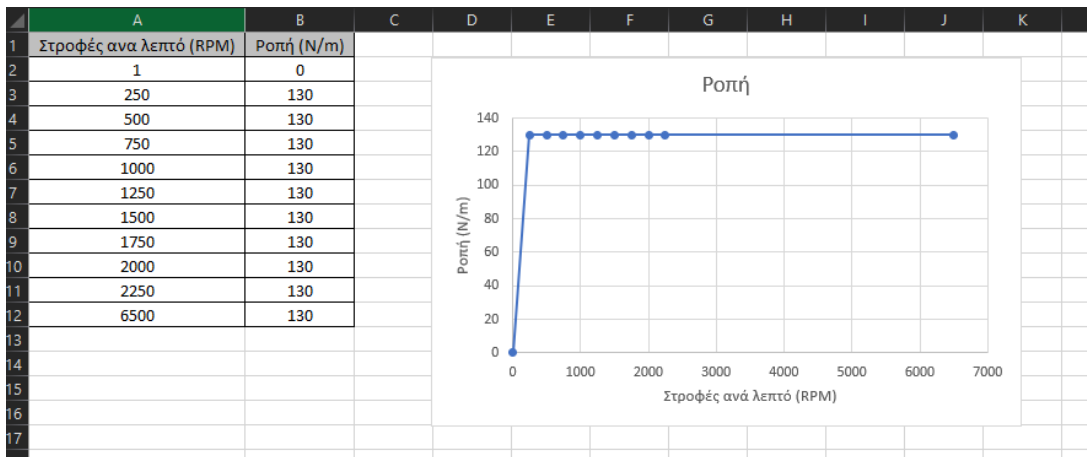
Το μόνο μέγεθος στο οποίο δεν έχει γίνει ακόμα αναφορά είναι το βάρος του οχήματος. Παρόλο που το βάρος δεν διέπεται από κάποιο κανονισμό, οι παράγοντες που επηρεάζουν τη μάζα ενός αμαξιού είναι πολυπληθείς, αλλά οι σημαντικότεροι είναι το σασί, ο τύπος κινητήρα και ο ίδιος ο οδηγός. Τα ηλεκτρικά μονοθέσια είναι ελαφρώς βαρύτερα από τα βενζινοκίνητα, λόγω των μπαταριών. Ένα όχημα με οδηγό στη θέση του και γεμισμένο με όλα τα απαραίτητα υγρά μπορεί εύκολα να ξεπεράσει τα 300 κιλά. Η κατανομή μάζας πάντα είναι βεβαρημένη προς τα πίσω, λόγω της αναγκαστικής θέσης του κινητήρα πίσω από τον οδηγό.

Μεταξόνιο, ύψος κέντρου βαρύτητας και γεωμετρία αναρτήσεων

Σύμφωνα με τον κανονισμό T.2.10.2, το μεταξόνιο και το μετατρόχιο του οχήματος πρέπει να συνεργάζονται με τρόπο που να αποτρέπει το αναποδογύρισμα (rollover) του

οχήματος. Εάν το μετατρόχιο είναι πολύ κοντό, το όχημα διατρέχει κίνδυνο να εκτραπεί σε περίπτωση απότομης μεταβολής του βάρους – εάν είναι πολύ μακρύ, το όχημα δυσχεραίνεται λόγω μεγαλύτερου βάρους, κόστους και μεγαλύτερης ακτίνας στροφής. Γίνεται η υπόθεση ότι το όχημα έχει το ίδιο μετατρόχιο στον μπροστά και τον πίσω άξονα. Σύμφωνα με τον κανονισμό T.2.2.1, το ύψος του χαμηλότερου μέρους του οχήματος πρέπει να είναι τουλάχιστον 30 χιλιοστά από το έδαφος. Το ύψος αυτό μετριέται με το όχημα σε ακινησία, με τον οδηγό στη θέση του. Ως σκληρότητες και γωνίες των ελατηρίων χρησιμοποιούνται κάποιες ενδεικτικές τιμές που θα μπορούσαν να συναντηθούν σε συστήματα αναρτήσεων Formula Student. Το ίδιο ισχύει για το ύψος του κέντρου βαρύτητας. Ως λόγο κίνησης και των δύο αναρτήσεων εισάγεται το 1. Αυτό σημαίνει ότι για κάθε 1 χιλιοστό που μετατοπίζεται το ελατήριο της ανάρτησης, ο τροχός θα μετατοπίζεται ακριβώς την ίδια απόσταση. Η εξακρίβωση των μεγεθών αυτών δεν γίνεται χωρίς εξειδικευμένο λογισμικό σχεδίασης και κινηματικής ανάλυσης αναρτήσεων.

Όλα τα χαρακτηριστικά του οχήματος που συζητήθηκαν στο κεφάλαιο αυτό φαίνονται στο παρακάτω υπολογιστικό φύλλο Microsoft Excel. Θα γίνει εισαγωγή στο MATLAB και με τις τιμές που παρέχονται θα υπολογιστούν τα χαρακτηριστικά του οχήματος, όσον αφορά την τελική ταχύτητα, τις επιταχύνσεις και τις επιβραδύνσεις του. Με αυτά ως γνώμονα υπολογίζεται η απόδοση ενός οχήματος κατά τη διάρκεια ενός γύρου.



A	B	C	D	E	F
1	Παράμετρος	Τιμή	Μέγεθος		Τύπος
2	Μάζα	300	kg		Formula Student
3	Εμπρόσθια κατανομή μάζας	45	%		Κινητήρας
4	Συντελεστής άντωσης (CL)	-2	--		Ηλεκτρικός
5	Συντελεστής οπισθέλκουσας (CD)	-0.8	--		
6	Εμπρόσθια Αεροδυναμική κατανομή	45	%		
7	Frontal area	1	m ²		
8	Πυκνότητα αέρα (ρ)	1,225	kg/m ³		
9	Διάμετρος ελαστικού	260.34	mm		
10	Αντίσταση κύλισης	-0,001	--		
11	Διαμήκης συντελεστής τριβής	2	--		
12	Πλευρικός συντελεστής τριβής	2	--		
13	Μείωση γραναζιού	4	--		
14	Απόδοση κινητήρα	96	%		
15	Μεταξόνιο	1525	mm		
16	Μετατρόχιο	1200	mm		
17	Λόγος κίνησης μπροστά ανάρτησης	1	--		
18	Σκληρότητα ελατήριων μπροστά ανάρτησης	80	N/mm		
19	Γωνία ελατηρίων μπροστά ανάρτησης	45	μοίρες		
20	Λόγος κίνησης πίσω ανάρτησης	1	--		
21	Σκληρότητα ελατήριων πίσω ανάρτησης	100	N/mm		
22	Γωνία ελατηρίων πίσω ανάρτησης	45	μοίρες		
23	Στατικό ύψος οχήματος	30	mm		
24	Ύψος κέντρου βαρύτητας	300	mm		

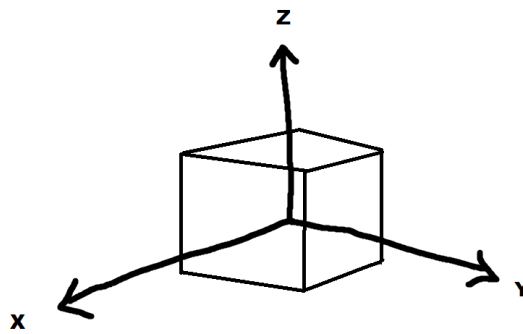
Εικόνες 12 & 13: Τα χαρακτηριστικά του οχήματος στο Microsoft Excel

Κεφάλαιο 5

Δομή κώδικα

5.1 Εισαγωγή

Για τις ανάγκες της εργασίας, θα γίνονται αναφορές σε 3 άξονες κίνησης οι οποίοι έχουν κέντρο το κέντρο του οχήματος: Ο άξονας x , ή αλλιώς διαμήκης (longitudinal) άξονας διατρέχει το όχημα κατά μήκος. Ο άξονας y , ή αλλιώς πλευρικός (lateral) άξονας, και ο άξονας z , η αλλιώς κάθετος άξονας.



Πίνακας 1: Άξονες κίνησης

Αξίζει επίσης να σημειωθεί πως το όχημα αντιμετωπίζεται ως σημειακή μάζα. Η κάθε κατάσταση του οχήματος θεωρείται σταθερή, και γίνεται να επιταχύνει μόνο σε έναν άξονα την φορά.

Για τις ανάγκες της εργασίας χρησιμοποιήθηκε η βασική δομή του λογισμικού OpenLap και τροποποιήθηκε ανάλογα με τις ανάγκες της εργασίας. Το OpenLap είναι λογισμικό open source και βρίσκεται στις παρακάτω ιστοσελίδες:

<https://www.mathworks.com/matlabcentral/fileexchange/75063-openlap-lap-time-simulator>

<https://github.com/mc12027/OpenLAP-Lap-Time-Simulator>

5.2 Γενικότερη δομή κώδικα

Στην αρχή του αλγορίθμου, εισάγονται τα δεδομένα από Microsoft Excel και ύστερα υπολογίζονται οι παράμετροι του οχήματος και η διάταξη της πίστας. Επακολούθως, αποθηκεύονται σε δομές δεδομένων που ονομάζονται structures και εισάγονται στις σχετικές συναρτήσεις για υπολογισμό της ταχύτητας του οχήματος σε κάθε σημείο της πίστας, η οποία πλεγματοθετείται με όση ακρίβεια χρειάζεται. Υπολογίζεται ο χρόνος που χρειάστηκε για την μετάβαση από το ένα σημείο στο επόμενο, και το συνολικό άθροισμα των χρόνων αυτών είναι ο χρόνος περάσματος (lap time). Παρόμοια μεθοδολογία χρησιμοποιείται για τον υπολογισμό της κατανάλωσης ενέργειας καθώς είναι δεδομένα τα χαρακτηριστικά του ηλεκτροκινητήρα και η ταχύτητα του οχήματος σε κάθε σημείο της πίστας.

5.2.1 Εισαγωγή και επεξεργασία δεδομένων

```
1      %% Εκκίνηση
2      clear
3      clc
4      close all force
5      diary('off')
6      fclose('all') ;
7
8      %% Επιλογές
9      %files
10     VehicleFilename = 'vehicle.xlsx' ; % Αρχείο οχήματος
11     TrackFilename = 'Autocross.xlsx' ; % Αρχείο πίστας
12     % meshing
13     mesh = 1; % meshing πίστας σε μέτρα (meters)
14     kappa = 1000 ; % μεταβλητή μακρών στροφών σε μοίρες (deg)
15
16     disp('Εκκίνηση αλγορίθμου επιτυχής.')
17
18     %% Εισαγωγή αρχείων
19     opts = spreadsheetImportOptions;
20     opts.Sheet = 'Parameters';
21     opts.DataRange = 'C2:C23';
22     opts.MissingRule = 'omitrow';
23     parameters = readtable(VehicleFilename,opts);
24     opts2 = spreadsheetImportOptions;
25     opts2.Sheet = 'Motor';
26     opts2.VariableNames = {'RPM','Torque'};
27     opts2.DataRange = 'A2:B12';
28     opts2.MissingRule = 'omitrow';
29     torque = readtable(VehicleFilename,opts2);
30     opts3 = spreadsheetImportOptions('NumVariables',3);
31     opts3.DataRange = 'A2:C999';
32     opts3.MissingRule = 'omitrow';
33     opts3.VariableNames = ["Type", "Length", "Radius"];
34     opts3.VariableTypes = ["categorical", "double", "double"];
35     shape = readtable(TrackFilename, opts3);
36     veh = struct;
37     tr = struct;
38     disp('Εισαγωγή δεδομένων επιτυχής.')
39
```

Εικόνα 14: Εισαγωγή δεδομένων στον κώδικα

Αρχικά, με τις πρώτες 5 εντολές γίνεται εκκαθάριση της μνήμης, του workspace, του

command line και κλείνουν τυχόν ανοιχτά αρχεία και figures που ήταν ανοικτά πριν στο matlab. Αμέσως μετά επιλέγονται τα αρχεία Excel που θα χρησιμοποιηθούν στην προσομοίωση. Πρέπει να σημειωθεί ότι τα αρχεία πρέπει να βρίσκονται στο ίδιο directory με το αρχείο .mat, αλλιώς το πρόγραμμα θα αποπειραθεί να αλλάξει το directory προορισμού για να τα βρεί. Στις γραμμές 13 και 14 δίνονται τιμές σε 2 μεταβλητές σχετικά με την ανάλυση της πίστας, την meshing και την kappa. Η πρώτη έχει αναφερθεί ήδη στην εργασία ως το «πλέγμα» της πίστας, δηλαδή μια ακολουθία διακριτών σημείων τα οποία το αμάξι διανύει για να φθάσει στον τερματισμό. Όσο υψηλότερο είναι το νούμερο της μεταβλητής, η ακρίβεια (ή αλλιώς «ανάλυση») της πίστας μικραίνει. Η δεύτερη μεταβλητή λειτουργεί παρόμοια με πλέγμα, αλλά για τμήματα της πίστας που υπάγονται σε στροφές. Εάν το τόξο που διαγράφεται από την στροφή είναι μεγαλύτερο από την μεταβλητή, τότε ο αλγόριθμος εισάγει συμπληρωματικά σημεία στην πορεία του οχήματος στην στροφή με σκοπό την βελτίωση της ακρίβειας. Θα γίνει περαιτέρω εμβάθυνση στην λειτουργία αυτή στην ενότητα δημιουργίας της πίστας. Στην γραμμή 17 εμφανίζεται η συμβολοσειρά «Εκκίνηση αλγορίθμου επιτυχής» για να ενημερωθεί ο χρήστης ότι δεν υπήρχε κάποιο σφάλμα κατά την εισαγωγή των αρχείων.

Αφού γίνει η εισαγωγή των αρχείων, γίνεται χρήση της SpreadsheetImportOptions για να οριοθετηθούν τα κελιά του αρχείου Excel. Το αρχείο Excel του οχήματος έχει δύο φύλλα, οπότε ονοματίζονται το πρώτο και το δεύτερο και αποθηκεύονται τα αντίστοιχα δεδομένα. Ο πίνακας με τα χαρακτηριστικά του ηλεκτροκινητήρα αποθηκεύεται σε δύο πίνακες, τον RPM και τον Torque (ροπή). Εάν υπάρχει ανάγκη για επέκταση οποιουδήποτε πίνακα, τα όρια των δεδομένων μπορούν να επεξεργαστούν από τον κώδικα.

Παρόμοια μεθοδολογία ακολουθείται για το αρχείο με το σχήμα της πίστας. Οι τρεις μεταβλητές που περιγράφουν το σχήμα της πίστας φαίνονται στο παρακάτω φύλλο Excel:

	A	B	C	D
1	Τύπος	Μήκος	Ακτίνα	
2	Left	1	9,125	
3	Left	55	9,125	
4	Left	1	9,125	
5	Right	1	9,125	
6	Right	55	9,125	
7	Right	1	9,125	
8				
9				

Εικόνα 15: Αρχείο Excel πίστας

Τα κελιά Left και Right περιγράφουν την κατεύθυνση της στροφής (αριστερή ή δεξιά). Το μήκος και η ακτίνα κάθε στροφής αποθηκεύονται στα διπλανά κελιά. Οι ευθείες θα πρέπει να αποθηκεύονται με μηδενική ακτίνα. Ο τύπος κάθε τμήματος αποθηκεύεται ως μεταβλητή τύπου categorical. Όπως θα φανεί και από τον κώδικα αργότερα, τα κελιά αυτά μπορούν να έχουν ως τιμή μία από τρεις συμβολοσειρές. Μαζί με «Left» και «Right» υπάρχει και η επιλογή για «Straight», δηλαδή ευθεία.

Αφού εισαχθούν και τα δύο αρχεία και γίνει η αποθήκευση των δεδομένων, γίνεται η αρχικοποίηση των structures με το όνομα «veh» και «struct», στα οποία θα αποθηκευτούν τα δεδομένα για το όχημα και την πίστα αντίστοιχα.

Στο τέλος των εντολών, εκτυπώνεται η συμβολοσειρά «Εισαγωγή δεδομένων επιτυχής» για την ενημέρωση του χρήστη. Στις γραμμές 41 έως 64 γίνεται η αποθήκευση και μετατροπή όλων των μεταβλητών του οχήματος στο SI.


```

40 %% Επεξεργασία μεταβλητών οχήματος
41 i = 1;
42 M = str2double(table2array(parameters(i,1))); i = i+1; veh.M = M; % μάζα (kg)
43 WtDist = str2double(table2array(parameters(i,1)))/100; i = i+1; veh.WtDist = WtDist; % κατανομή βάρους προς τα μπροστά (%)
44 Lift = str2double(table2array(parameters(i,1))); i = i+1; veh.Lift = Lift; % συντελεστής άντωσης (--)
45 Drag = str2double(table2array(parameters(i,1))); i = i+1; veh.Drag = Drag; % συντελεστής οπισθέλκουσας (--)
46 AeroDist = str2double(table2array(parameters(i,1)))/100; i = i+1; veh.AeroDist = AeroDist; % αεροδυναμική κατανομή (%)
47 FrontArea = str2double(table2array(parameters(i,1))); i = i+1; veh.FrontArea = FrontArea; % εμπρόσθιο εμβαδόν (m^2)
48 AirDens = str2double(table2array(parameters(i,1))); i = i+1; veh.AirDens = AirDens; % πυκνότητα αέρα (kg*m^3)
49 TirRadius = str2double(table2array(parameters(i,1)))/1000; i = i+1; veh.TirRadius = TirRadius; % διάμετρος ελαστικού (m)
50 RollRes = str2double(table2array(parameters(i,1))); i = i+1; veh.RollRes = RollRes; % συντελεστής αντίστασης κύλισης ελαστικού (--)
51 Mux = str2double(table2array(parameters(i,1))); i = i+1; veh.Mux = Mux; % διαμήκης συντελεστής τριβής ελαστικού (--)
52 Muy = str2double(table2array(parameters(i,1))); i = i+1; veh.Muy = Muy; % πλευρικός συντελεστής τριβής ελαστικού (--)
53 Pgear = str2double(table2array(parameters(i,1))); i = i+1; veh.Pgear = Pgear; % λόγος μείωσης (--)
54 Eff = str2double(table2array(parameters(i,1)))/100; i = i+1; veh.Eff = Eff; % απόδοση κινητήρα (%)
55 WB = str2double(table2array(parameters(i,1)))/1000; i = i+1; veh.WB = WB; % μεταξόνιο (m)
56 TR = str2double(table2array(parameters(i,1)))/1000; i = i+1; veh.TR = TR; % μετατόχιτο (m)
57 FrontMR = str2double(table2array(parameters(i,1))); i = i+1; veh.FrontMR = FrontMR; % λόγος κίνησης μπροστά ανάρτησης (--)
58 FStiff = str2double(table2array(parameters(i,1)))*1000; i = i+1; veh.FStiff = FStiff; % σκληρότητα μπροστά ελατηρίων (N/m)
59 FAngle = str2double(table2array(parameters(i,1))); i = i+1; veh.FAngle = FAngle; % γωνία ελατηρίων μπροστά ανάρτησης (μοίρες)
60 RearMR = str2double(table2array(parameters(i,1))); i = i+1; veh.RearMR = RearMR; % λόγος κίνησης πίσω ανάρτησης (--)
61 RStiff = str2double(table2array(parameters(i,1)))*1000; i = i+1; veh.RStiff = RStiff; % σκληρότητα πίσω ελατηρίων (N/m)
62 RAngle = str2double(table2array(parameters(i,1))); i = i+1; veh.RAngle = RAngle; % γωνία ελατηρίων πίσω ανάρτησης (μοίρες)
63 RideHgt = str2double(table2array(parameters(i,1)))/1000; i = i+1; veh.RideHgt = RideHgt; % ύψος οχήματος (m)
64 Cog = str2double(table2array(parameters(i,1)))/1000; veh.Cog = Cog; % κέντρο βαρύτητας (m)
65
66

```

Εικόνα 6: Εισαγωγή και αποθήκευση μεταβλητών

Στο επόμενο βήμα ο κώδικας δημιουργεί το μοντέλο του οχήματος.

5.2.2 Δημιουργία μοντέλου οχήματος

```

56 %% Μοντέλο οχήματος
57 |
58 rpm_curve = torque(:,1); % καμπύλη ταχύτητας κινητήρα (στροφές κινητήρα) (rpm)
59 torque_curve = torque(:,2); % καμπύλη ροπής κινητήρα (n*m)
60 rpm_curve = str2double(table2array(rpm_curve));
61 torque_curve = str2double(table2array(torque_curve));
62 en_power_curve = torque_curve.*rpm_curve*2*pi/60; % ισχύς κινητήρα (watt)
63 wheel_speed = zeros(length(rpm_curve),1); % ταχύτητα τροχού για κάθε τιμή στροφών κινητήρα
64 velocity = zeros(length(rpm_curve),1); % ταχύτητα οχήματος για κάθε τιμή στροφών κινητήρα
65 wheel_torque = zeros(length(torque_curve),1); % ροπή τροχού για κάθε τιμή στροφών κινητήρα
66 wheel_speed(:,1) = rpm_curve./Pgear; % ταχύτητα τροχού για κάθε τιμή στροφών κινητήρα
67 velocity(:,1) = wheel_speed(:,1)*2*(pi/60)*TirRadius; % υπολογισμός ταχύτητας οχήματος για κάθε ταχύτητα τροχού
68 wheel_torque(:,1) = torque_curve.*Pgear/Eff; % υπολογισμός ροπής τροχού με συντελεστή απόδοσης κινητήρα
69 % ελάχιστη και μέγιστη ταχύτητα οχήματος
70 v_min = min(velocity,[], 'all');
71 v_max = max(velocity,[], 'all');
72 % πλέγμα ταχυτητών με βήμα 1 m/s ανάμεσα στην ελάχιστη και μέγιστη ταχύτητα
73 dv = 1;
74 vehicle_speed = linspace(v_min,v_max,(v_max-v_min)/dv); % πίνακας ταχύτητας
75 fx_engine = zeros(length(vehicle_speed),1); % δύναμη κινητήρα
76 % υπολογισμός δύναμη κινητήρα για κάθε τιμή ταχύτητας στο πλέγμα
77 for i=1:length(vehicle_speed)
78     fx_engine(i) = interp1(velocity(:),wheel_torque(:)/TirRadius,vehicle_speed(i),'linear',0);
79 end
80 vehicle_speed = [0;vehicle_speed]; % εισάγουμε μηδενικό ως πρώτη τιμή για καλύτερη ακρίβεια στο διάγραμμα
81 fx_engine = [fx_engine(1);fx_engine]; % εισάγουμε μηδενικό ως πρώτη τιμή για καλύτερη ακρίβεια στο διάγραμμα
82 veh.vehicle_speed = vehicle_speed; % αποθήκευση στο structure
83 veh.fx_engine = fx_engine; % αποθήκευση στο structure
84 engine_rpm = Pgear.*vehicle_speed/TirRadius*60/2/pi; % στροφές κινητήρα
85 wheel_torque = fx_engine*TirRadius; veh.wheel_torque = wheel_torque; % ροπή τροχού
86 engine_torque = wheel_torque/Pgear; % ροπή κινητήρα
87 engine_power = engine_torque.*engine_rpm*2*pi/60; veh.engine_power = engine_power; % ισχύς κινητήρα
88 disp('Μοντελοποίηση του οχήματος επιτυχής.')
89

```

Εικόνα 17: Αρχική επεξεργασία μεταβλητών οχήματος

Χρησιμοποιώντας μια μεταβλητή i , δίνεται σε κάθε μεταβλητή που χρειάζεται την αντίστοιχη τιμή της μεταβλητής από το αρχείο Excel. Καθώς το MATLAB αναγνωρίζει

τα κελία Excel ως string (συμβολοσειρές), μετατρέπονται όλες οι μεταβλητές σε τύπου double. Τα ποσοστά διαιρούνται με το εκατό για να αποθηκευθεί μια μεταβλητή μικρότερη του ένα για να χρησιμοποιηθεί ως συντελεστής από τον κώδικα. Η διάμετρος του ελαστικού αποθηκεύεται σε μέτρα, άρα διαιρείται με το χίλια. Όλες οι μεταβλητές αποθηκεύονται στο structure «veh».

Σειρά έχει ο υπολογισμός της ροπής του κινητήρα, η τελική ταχύτητα του οχήματος και η ισχύς του κινητήρα. Με τη χρήση των δεδομένων του φύλλου Excel, η μηχανική ισχύς του κινητήρα υπολογίζεται σε έναν πίνακα που περιγράφεται από την ακόλουθη εξίσωση:

$$T R 2 \frac{\pi}{60} \quad (1.1)$$

Όπου T η ροπή του κινητήρα και R οι στροφές ανά λεπτό του κινητήρα..

```

56 %% Μοντέλο οχήματος
57
58 rpm_curve = torque(:,1); % καμπύλη ταχύτητας κινητήρα (στροφές κινητήρα) (rpm)
59 torque_curve = torque(:,2); % καμπύλη ροπής κινητήρα (n*m)
60 rpm_curve = str2double(table2array(rpm_curve));
61 torque_curve = str2double(table2array(torque_curve));
62 en_power_curve = (torque_curve.*rpm_curve*2*pi/60); % ισχύς κινητήρα (watt)
63 wheel_speed = zeros(length(rpm_curve),1); % ταχύτητα τροχού για κάθε τιμή στροφών κινητήρα
64 velocity = zeros(length(rpm_curve),1); % ταχύτητα οχήματος για κάθε τιμή στροφών κινητήρα
65 wheel_torque = zeros(length(torque_curve),1); % ροπή τροχού για κάθε τιμή στροφών κινητήρα
66 wheel_speed(:,1) = rpm_curve./Pgear; % ταχύτητα τροχού για κάθε τιμή στροφών κινητήρα
67 velocity(:,1) = wheel_speed(:,1)*2*(pi/60)*TirRadius;% υπολογισμός ταχύτητας οχήματος για κάθε ταχύτητα τροχού
68 wheel_torque(:,1) = torque_curve/Pgear; % υπολογισμός ροπής τροχού
69 % ελάχιστη και μέγιστη ταχύτητα οχήματος
70 v_min = min(velocity,[],'all');
71 v_max = max(velocity,[],'all');
72 % πλέγμα ταχυτητών με βήμα 1 m/s ανάμεσα στην ελάχιστη και μέγιστη ταχύτητα
73 dv = 1;
74 vehicle_speed = linspace(v_min,v_max,(v_max-v_min)/dv)'; % πίνακας ταχύτητας
75 fx_engine = zeros(length(vehicle_speed),1);% δύναμη κινητήρα
76 % υπολογισμός δύναμη κινητήρα για κάθε τιμή ταχύτητας στο πλέγμα
77 for i=1:length(vehicle_speed)
78     fx_engine(i) = interp1(velocity(:,1),wheel_torque(:)/TirRadius,vehicle_speed(i),'linear',0);
79 end
80 vehicle_speed = [0;vehicle_speed]; % εισάγουμε μηδενικό ως πρώτη τιμή για καλύτερη ακρίβεια στο διάγραμμα
81 fx_engine = [fx_engine(1);fx_engine]; % εισάγουμε μηδενικό ως πρώτη τιμή για καλύτερη ακρίβεια στο διάγραμμα
82 veh.vehicle_speed = vehicle_speed; % αποθήκευση στο structure
83 veh.fx_engine = fx_engine; % αποθήκευση στο structure
84 engine_rpm = Pgear.*vehicle_speed/TirRadius*60/2/pi; % στροφές κινητήρα
85 wheel_torque = fx_engine*TirRadius; veh.wheel_torque = wheel_torque; % ροπή τροχού
86 engine_torque = wheel_torque/Pgear; % ροπή κινητήρα
87 engine_power = engine_torque.*engine_rpm*2*pi/60; veh.engine_power = engine_power; % ισχύς κινητήρα
88 disp('Μοντελοποίηση του οχήματος επιτυχής.')
89

```

Εικόνα 18: Μοντελοποίηση του κινητήρα

Ως αποτέλεσμα θα δημιουργηθεί ένας πίνακας 2 στηλών και με τον ίδιο αριθμό σειρών

με τον πίνακα τιμών ροπής του οχήματος. Η μία στήλη θα έχει τις τιμές ροπής και η άλλη τις τιμές ισχύς του κινητήρα. Η σχέση αυτή θα είναι ανάλογη.

Με την χρήση της εντολής `length()` δημιουργούνται τρεις παρόμοιοι μηδενικοί πίνακες στους οποίους θα αποθηκευτούν τα υπόλοιπα δεδομένα. Οι πίνακες θα έχουν ίδιο αριθμό σειρών με τον πίνακα τιμών ροπής του οχήματος. Από την μηχανική υπολογίζεται ότι η ταχύτητα του τροχού (σε περιστροφές ανά λεπτό), είναι οι περιστροφές του κινητήρα διαιρεμένες με ολόκληρο τον λόγο μείωσης του οχήματος, που είναι η μεταβλητή `Pgear`. Το αποτέλεσμα αποθηκεύεται στον πίνακα `wheel_speed`. Η ταχύτητα όλου του οχήματος μπορεί να υπολογιστεί από τον τύπο της γραμμικής ταχύτητας κυκλικής κίνησης.

$$v = \frac{2\pi}{60} \omega R \quad (1.2)$$

Όπου ω ουσιαστικά είναι η γραμμική ταχύτητα του τροχού (η μεταβλητή `wheel_speed`) και R η διάμετρος του (η μεταβλητή `TirRadius`). Η διαίρεση με το 60 χρειάζεται για να μετατραπούν οι στροφές ανά λεπτό σε περίοδο μίας περιστροφής. Η τελική ροπή του τροχού υπολογίζεται με το πηλίκο της ροπής του κινητήρα με την μείωση των γραναζιών. Με τις εντολές `min()` και `max()` αποθηκεύονται η ελάχιστη και η μέγιστη ταχύτητα του οχήματος αντίστοιχα. Στην γραμμή 74, με την εντολή `linspace` δημιουργείται μια ακολουθία όλων των τιμών ανάμεσα στο `v_min` και `v_max`, με βήμα την μεταβλητή `dv`, η οποία έχει οριστεί ως 1. Αφού αποθηκευτεί και άλλος ένας μηδενικός πίνακας για την δύναμη που παράγει ο κινητήρας, υπολογίζεται η δύναμη που παράγει ο κινητήρας για κάθε τιμή ταχύτητας με χρήση της εντολής `interp1` ανάμεσα στους πίνακες `velocity` και `wheel_torque`, ο δεύτερος διαιρεμένος με τη διάμετρο του τροχού, μέσα σε ένα βρόγχο. Μετά τον τερματισμό του βρόγχου, εισάγεται η τιμή 0 ως ο πρώτος όρος στον πίνακα `vehicle_speed` (χωρίς να αντικαθιστά

τον πρώτο ορο) και αντίστοιχα ο πρώτος όρος της f_{x_engine} ξαναεισάγεται ως ο πρώτος όρος της. Τα παραπάνω έγιναν με σκοπό την καλύτερη ακριβεία στην δημιουργία ενός διαγράμματος επιταχύνσεων για το όχημα.

Αφού είναι διαθέσιμες οι μέγιστες και ελαχίστες ταχύτητες του οχήματος, υπολογίζονται οι στροφές του κινητήρα σε σχέση με τις ταχύτητες, πολλαπλασιάζοντας τον συντελεστή μείωσης με την εξίσωση 1.2. Επίσης ξαναυπολογίζεται η ροπή του τροχού για κάθε τιμή του πίνακα f_{x_engine} χρησιμοποιώντας την εξίσωση της ροπής.

$$T = F R \quad (1.3)$$

Όπου F είναι η κάθε τιμή του πίνακα f_{x_engine} και R η μεταβλητή $TirRadius$, δηλαδή η διάμετρος του τροχού. Έχει γίνει ήδη η υποθέση ότι η δύναμη φτάνει από τον κινητήρα στο κέντρο του τροχού χωρίς απώλειες. Αποτελέσμα της παραπάνω εξίσωσης είναι ο πίνακας $engine_torque$, που έχει μία στήλη και όσες σειρές έχει ο πίνακας $wheel_torque$. Ως τελευταίο βήμα, με χρήση της εξίσωσης 1.1, υπολογίζεται η ισχύς του κινητήρα για κάθε τιμή του πίνακα $engine_torque$.

```

90  %% Δυνάμεις
91  g = 9.81;
92  % κατανομές βάρους και αεροδυναμικής
93  factor_drive = (1-wtDist) ; % κατανομή βάρους
94  factor_aero = (1-AeroDist) ; % κατανομή αεροδυναμικής
95  driven_wheels = 2 ; % αριθμός κινητήριων τροχών
96  % άξονας z (κατακόρυφος άξονας)
97  fz_mass = -M*g ; % βάρος οχήματος
98  fz_aero = 1/2*AirDens*Lift*FrontArea*vehicle_speed.^2; % αεροδυναμική αρνητική άντωση (downforce ή κάτω δύναμη)
99  fz_total = fz_mass+fz_aero ; veh.fz_total = fz_total; % ολική δύναμη στον άξονα z
100 fz_tyre = (factor_drive*fz_mass+factor_aero*fz_aero)/driven_wheels;
101 % άξονας x (διαμήκης άξονας)
102 fx_aero = 1/2*AirDens*Drag*FrontArea*vehicle_speed.^2;% οπισθέλκουσα
103 fx_roll = RollRes*abs(fz_total); % αντίσταση κύλισης
104 fx_tyre = driven_wheels*(Mux.*abs(fz_tyre)); % διαμήκης δύναμη στο ελαστικό
105

```

Εικόνα 19: Υπολογισμός δυνάμεων

Το επόμενο βήμα είναι ο υπολογισμός των δυνάμεων που διέπουν το όχημα. Αφού αρχικοποιηθεί η μεταβλητή g (η επιτάχυνση της βαρύτητας που θεωρείται $9,81 \text{ m/s}^2$) και οι συντελεστές βάρους, αεροδυναμικής και σημειωθεί ο αριθμός κινούμενων

τροχών (σε αυτή την εργασία προσομοιώνεται όχημα με τους δύο πίσω τροχούς ως κινητήριους) υπολογίζονται τα υπόλοιπα φορτία στο αμάξι, σε κάθε άξονα. Στον άξονα z, δηλαδή τον κάθετο άξονα, υπολογίζεται το βάρος του οχήματος (η δύναμη που του ασκείται από την Γη) και η τιμή της αεροδυναμικής αρνητικής άντωσης. Η αεροδυναμική άντωση μετριέται σε Newton και υπολογίζεται ως εξής:

$$1/2 \rho C_L A v^2 (1.4)$$

Όπου ρ η πυκνότητα του αέρα (η μεταβλητή AirDens), C_L ο συντελεστής άντωσης (η μεταβλητή Lift), A η μεταβλητή εμπρόσθιας περιοχής (η μεταβλητή FrontArea) και v η ταχύτητα του οχήματος. Το άθροισμα της fz_mass και fz_aero μας δίνουν την fz_total . Η fz_tyre , δηλαδή η δύναμη του ελαστικού, είναι το φορτίο του οχήματος επί τον αριθμό των κινούμενων τροχών, αθροισμένο με τα αεροδυναμικά φορτία, δια τον αριθμό των κινητήριων τροχών. Στον άξονα x υπολογίζονται οι εξής δυνάμεις. Η οπισθέλκουσα, που υπολογίζεται με τρόπο παρόμοιο της άντωσης.

$$1/2 \rho C_D A v^2 (1.5)$$

Όπου ρ η πυκνότητα του αέρα (η μεταβλητή AirDens), C_D ο συντελεστής οπισθέλκουσας (η μεταβλητή Drag), A η μεταβλητή εμπρόσθιας περιοχής (η μεταβλητή FrontArea) και v η ταχύτητα του οχήματος. Η δύναμη που χρειάζεται το όχημα για να κυλίσει fx_roll , που είναι ο συντελεστής κύλισης δια την fz_tyre , και η δύναμη στο ελαστικό κατά τον άξονα x.

$$2 * (\mu x |Fz|) (1.6)$$

Όπου 2 είναι ο αριθμός των κινητήριων τροχών, μx ο διαμήκης συντελεστής τριβής και Fz ο πίνακας fz_tyre (οι δυνάμεις υπολογίζονται για όλες τις τιμές στον πίνακα $vehicle_speed$) σε απόλυτη τιμή.

Ένα διάγραμμα «G-G-V» είναι από τους πιο αξιόπιστους τρόπους για να εξεταστεί η

απόδοση ενός αγωνιστικού οχήματος. Αποτελείται από την τομή 2 καμπυλών, οι οποίες περιγράφουν τις μέγιστες πλευρικές και διαμήκης επιταχύνσεις που μπορεί να αναπτύξει ένα όχημα. Ύστερα, αυτές οι επιταχύνσεις υπολογίζονται για κάθε ταχύτητα που μπορεί να αναπτύξει το όχημα και συνδυάζονται για να δημιουργήσουν ένα τρισδιάστατο διάγραμμα κυλινδρικής μορφής.

```

106 %% Διάγραμμα G-G-V
107 % δυνάμεις
108 Wz = M*g; % βάρος στον z άξονα
109 Wx = -M*g; % βάρος στον y άξονα
110 Wx = M*g; % βάρος στον x άξονα
111 % βήμα διαγράμματος (μεγαλύτερη τιμή κάνει το διάγραμμα πιο πυκνό)
112 dv = 5;
113 v = (0:dv:v_max)';
114 if v(end)~=v_max
115     v = [v;v_max];
116 end
117 % σημεία καμπύλης
118 N = 50;
119 % αρχικοποίηση πίνακα σημείων
120 GGv = zeros(length(v),2*N-1,3);
121 for i=1:length(v)
122     % αεροδυναμικά φορτία
123     Aero_Df = 1/2*AirDens*Lift*FrontArea*v(i)^2;
124     Aero_Dr = 1/2*AirDens*Drag*FrontArea*v(i)^2;
125     Roll_Dr = RollRes*abs(-Aero_Df+Wz); % αντίσταση κύλισης
126     Wd = (factor_drive*Wz+(-factor_aero*Aero_Df))/driven_wheels; % φορτίο στους κινητήριους τροχούς
127     ax_drag = (Aero_Dr+Roll_Dr+Wx)/M; % επιτάχυνση οπισθέλκουσας
128     ay_max = 1/M*Muy*(Wz-Aero_Df); % μέγιστη πλευρική επιτάχυνση διαθέσιμη από τα ελαστικά
129     ax_tyre_max_acc = 1/M*Mux*Wd*driven_wheels; % μέγιστη διαμήκης επιτάχυνση διαθέσιμη από τα ελαστικά
130     ax_tyre_max_dec = -1/M*Mux*(Wz-Aero_Df); % μέγιστη διαμήκης επιβράδυνση διαθέσιμη από τα ελαστικά
131     ax_power_limit = 1/M*(interp1(vehicle_speed,fx_engine,v(i))); % μέγιστη ισχύ κινητήρα
132     ax_power_limit = ax_power_limit*ones(N,1);
133     ay = ay_max*cosd(linspace(0,180,N))'; % πίνακας πλευρικής επιτάχυνσης
134     % πίνακας διαμήκης επιτάχυνσης
135     ax_tyre_acc = ax_tyre_max_acc*sqrt(1-(ay/ay_max).^2); % καμπύλη επιταχύνσεων
136     ax_acc = min(ax_tyre_acc,ax_power_limit)+ax_drag; % οριοθέτηση από τα όρια του κινητήρα
137     ax_dec = ax_tyre_max_dec*sqrt(1-(ay/ay_max).^2)+ax_drag; % καμπύλη επιβραδύνσεων
138     % αποθήκευση διαγράμματος
139     GGv(i,:,1) = [ax_acc',ax_dec(2:end)]';
140     GGv(i,:,2) = [ay',flipud(ay(2:end))']';
141     GGv(i,:,3) = v(i)*ones(1,2*N-1);
142 end
143

```

Εικόνα 110: Δημιουργία διαγράμματος G-G-V

Αρχικά, υπολογίζεται το βάρος του οχήματος σε κάθε άξονα και επιλέγεται το πόσο πυκνά θα είναι τα αποτελέσματα του διαγράμματος (με βήμα). Δημιουργείται πίνακας v με όλες τις ταχύτητες του οχήματος. Ο κενός πίνακας GGv έχει τον ίδιο αριθμό σειρών με τον πίνακα v , στήλες ίσες με το διπλάσιο των σημείων των καμπυλών (που επιλέχθηκε από την μεταβλητή N) πλην ένα και βάθος 3. Χρησιμοποιούνται οι ίδιες εξισώσεις που συζητήθηκαν παραπάνω για τον υπολογισμό των δυνάμεων. Στην γραμμή 127, η επιβράδυνση στο όχημα λόγω της αντίστασης του αέρα περιγράφεται από την παρακάτω εξίσωση:

$$\frac{(Dr + Roll + Wx)}{M} \quad (1.7)$$

Όπου Dr η αντίσταση του αέρα, $Roll$ η αντίσταση κύλισης και Wx το βάρος του οχήματος στον x άξονα. M είναι η μάζα του οχήματος. Η μέγιστη πλευρική επιτάχυνση του οχήματος είναι:

$$\frac{1}{M}\mu y(Wz - Df) \quad (1.8)$$

Όπου M η μάζα του οχήματος, μy ο πλευρικός συντελεστής τριβής των ελαστικών, Wz το βάρος του οχήματος στον z άξονα, και Df η τιμή της αρνητικής άντωσης. Επόμενες είναι η επιτάχυνση και επιβράδυνση στον x άξονα που μπορούν να προσφέρουν τα ελαστικά. Η επιτάχυνση υπολογίζεται ως:

$$\frac{1}{M}\mu x Wd \times 2 \quad (1.9)$$

Όπου M η μάζα του οχήματος, μx ο διαμήκης συντελεστής τριβής των ελαστικών, Wd το μέγιστο φορτίο του οχήματος στους κινητήριους τροχούς (ουσιαστικά, είναι η μέγιστη τιμή του πίνακα `fz_tyre`) επί τον αριθμό των κινητήριων τροχών του οχήματος. Με παρόμοιο τρόπο υπολογίζεται η μέγιστη επιβράδυνση που μπορούν να προσφέρουν τα ελαστικά:

$$-\frac{1}{M}\mu x(Wz - Df) \quad (1.10)$$

Όπου M η μάζα του οχήματος, μx ο διαμήκης συντελεστής τριβής των ελαστικών, Wz το βάρος του οχήματος στον z άξονα, και Df η τιμή της άντωσης.

Το όριο επιτάχυνσης του κινητήρα υπολογίζεται με την εντολή `interp1` ανάμεσα στους πίνακες `vehicle_speed` και `fx_engine`, δια τη μάζα του οχήματος. Για τον υπολογισμό του διαγράμματος, χρειάζεται να πολλαπλασιαστεί το όριο αυτό με έναν μοναδιαίο πίνακα που έχει τον ίδιο αριθμό σειρών με τα σημεία της καμπύλης (που τα ρυθμίζει η μεταβλητή N). Για να δημιουργηθεί η καμπύλη πλευρικών επιταχύνσεων, ay ,

πολλαπλασιάζεται την μέγιστη επιτάχυνση με το συνημίτονο όλων των μοιρών από 0 έως 180 χρησιμοποιώντας την εντολή linspace. Η επιτάχυνση των ελαστικών είναι:

$$ax_tyre \sqrt{1 - \left(\frac{ay}{ay_max}\right)^2} \quad (1.10)$$

όπου ax_tyre η μέγιστη επιτάχυνση που μπορούν να προσφέρουν τα ελαστικά, ay η πλευρική επιτάχυνση του οχήματος και ay_max η μέγιστη επιτάχυνση του οχήματος. Με τον ίδιο ακριβώς τρόπο υπολογίζεται η επιβράδυνση του οχήματος στην γραμμή 137, με μοναδική διαφορά την πρόσθεση της αντίστασης του αέρα ax_drag. Η τελική τιμή της επιτάχυνσης του οχήματος (γραμμή 136) είναι η μικρότερη τιμή ανάμεσα στο όριο ισχύος του κινητήρα και το όριο επιτάχυνσης των ελαστικών, μαζί με την αντίσταση του αέρα. Ολοκληρώνεται η μοντελοποίηση του οχήματος αποθηκεύοντας τους πίνακες στον πίνακα GGV.

5.2.3 Δημιουργία μοντέλου πίστας

```

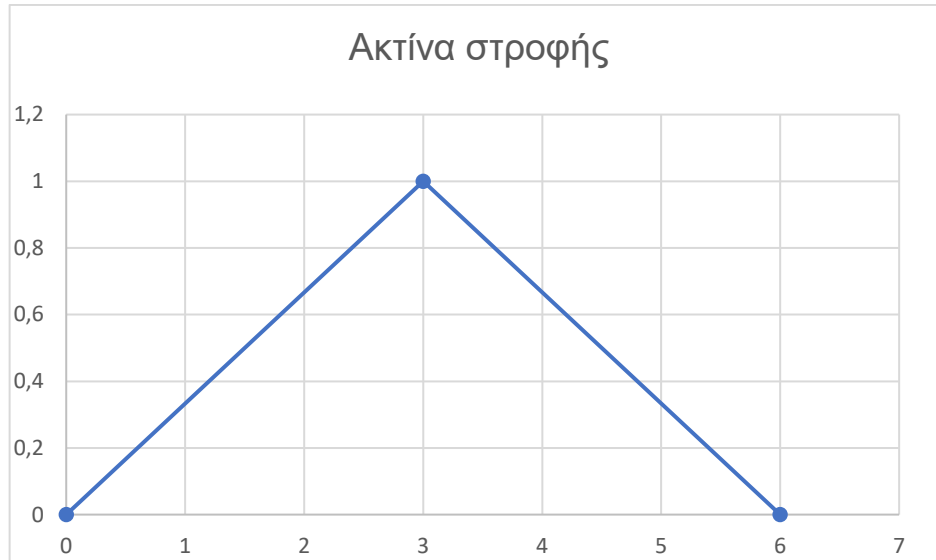
144 %% Σχήμα πίστας
145
146 Radius = table2array(shape(:,3)); % ακτίνα στροφών
147 l = table2array(shape(:,2)); % μήκος κελιών
148 type_tmp = table2array(shape(:,1)); % τύπος κελιών
149 Radius(Radius==0) = inf; % θέτουμε όλες τις ευθείες με μηδέν ακτίνα ως ευθείες με άπειρη ακτίνα
150 L = sum(l); % συνολικό μήκος πίστας
151 % θέτηση αριθμών για κάθε τύπο στροφής η ευθεία
152 Type = zeros(length(l),1);
153 Type(string(type_tmp)=="Straight") = 0;
154 Type(string(type_tmp)=="Left") = 1;
155 Type(string(type_tmp)=="Right") = -1;
156 % αφαιρούμε κελιά με μηδενικό μήκος
157 Radius(l==0) = [];
158 Type(l==0) = [];
159 l(l==0) = [];
160

```

Εικόνα 20: Αποθήκευση σχήματος πίστας

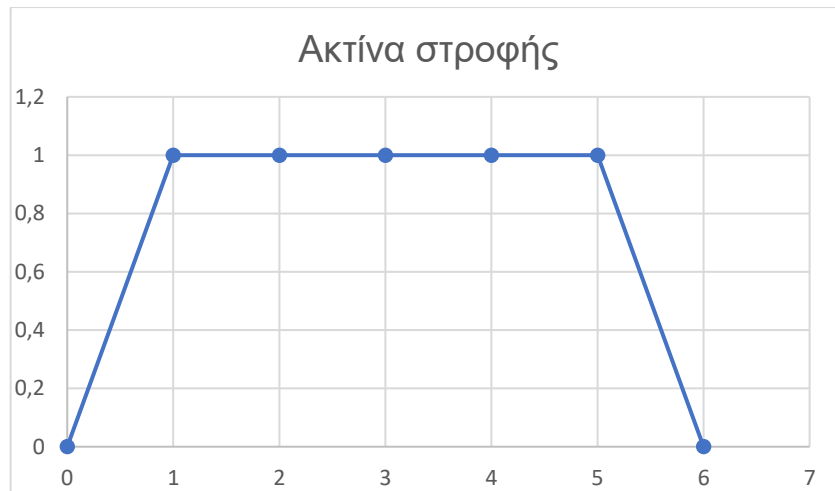
Το παραπάνω απόσπασμα κώδικα αποθηκεύει το σχήμα της πίστας σε μορφή που μπορεί να επεξεργαστεί. Αφού εισαχθούν και οι 3 στήλες του πίνακα Excel σε μορφή array, υπολογίζεται το συνολικό μήκος της πίστας. “Υστερα, για κάθε τύπο τμήματος της πίστας θέτονται 3 μεταβλητές. 1 για τις αριστερές στροφές και -1 για τις δεξιές. Εάν το μήκος οποιουδήποτε τμήματος είναι 0, αφαιρείται από τον πίνακα.

Στο παρακάτω διάγραμμα φαίνεται η ακτίνα της πίστας προς την ανάλογη θέση στην πίστα, σε ένα υποθετικό σενάριο. Έστω ότι από το σημείο 0 στον άξονα x έως το σημείο 6 υπάρχει μία στροφή.



Πίνακας 2: Ακτίνα στροφής σε σχέση με την θέση στην πίστα

Από το διάγραμμα φαίνεται ότι ο κώδικας αναγνωρίζει μόνο το κεντρικό κομμάτι κάθε τμήματος, κάτι που σε στροφές μεγάλου τόξου μειώνει την ακρίβεια. Καθώς η υποθετική στροφή έχει μήκος 6 και ακτίνα 1, το εμβαδόν κάτω από την μπλε γραμμή θα έπρεπε να καλύπτει περίπου ένα παραλληλόγραμμο ύψους 1 και μήκους 6. Όμως στο παρακάτω διάγραμμα (που απεικονίζεται το πως θα ήταν αποθηκευμένη στη μνήμη) το εμβαδόν είναι κοντά στο μισό του κανονικού. Η λύση σε αυτό το πρόβλημα είναι η εισαγωγή παραπάνω σημείων στην καμπύλη.



Πίνακας 3: Ακτίνα στροφής μετά την εισαγωγή στοιχείων

Το εμβαδόν κάτω από την καμπύλη προσεγγίζει το παραλληλόγραμμο με πολύ μεγαλύτερη ακρίβεια, χωρίς να έχει αλλάξει το συνολικό μήκος του τμήματος. Η παραπάνω λειτουργία εκτελείται από το παρακάτω κομμάτι κώδικα:

```

161 % προσθέτουμε σημεία σε μακρίες στροφές για μεγαλύτερη ακρίβεια
162 angle_seg = rad2deg(1./Radius);
163 j = 1; % σημείο
164 RR = Radius; % νέος πίνακας με τα ένθετα σημεία
165 ll = l; % νέος πίνακας με τα ένθετα σημεία
166 tt = Type; % νέος πίνακας με τα ένθετα σημεία
167 for i=1:length(l)
168     if angle_seg(i)>kappa
169         l_inj = min([ll(j)/3,deg2rad(kappa)*Radius(i)]);
170         ll = [
171             ll(1:j-1);...
172             l_inj;...
173             ll(j)-2*l_inj;...
174             l_inj;...
175             ll(j+1:end)...
176         ];
177         RR = [...
178             RR(1:j-1);...
179             RR(j);...
180             RR(j);...
181             RR(j);...
182             RR(j+1:end)...
183         ];
184         tt = [...
185             tt(1:j-1);...
186             tt(j);...
187             tt(j);...
188             tt(j);...
189             tt(j+1:end)...
190         ];
191         j = j+3;
192     else
193         j = j+1;
194     end
195 end
196 Radius = RR;
197 l = ll;
198 Type = tt;
199

```

Εικόνα 21: Εισαγωγή σημείων σε στροφές με μεγάλο τόξο

Τα τελικά τμήματα της πίστας υπολογίζονται ως παρακάτω:

```

200      %% Τελικός υπολογισμός τμημάτων
201
202      X = cumsum(1); % τελική θέση κάθε σημείου
203      XC = cumsum(1)-1/2; % κεντρική θέση κάθε σημείου
204      j = 1; % σημείο
205      x = zeros(length(X)+sum(Radius==inf),1); % αρχικοποίηση
206      r = zeros(length(X)+sum(Radius==inf),1); % αρχικοποίηση
207      for i=1:length(X)
208          if Radius(i)==inf % ένθετο σημείο σε κάθε τέλος ευθείας
209              x(j) = X(i)-1(i);
210              x(j+1) = X(i);
211              j = j+2;
212          else % κέντρο κυκλικού τμήματος
213              x(j) = XC(i);
214              r(j) = Type(i)./Radius(i);
215              j = j+1;
216          end
217      end
218
219      % αποθήκευση σημείων
220      xx = x;
221

```

Εικόνα 22: Υπολογισμός γωνιών και δημιουργία χάρτη

Υπολογίζεται το κεντρικό σημείο κάθε τμήματος της πίστας. Μετά, με έναν δείκτη j , εισάγονται συμπληρωματικά σημεία στο κέντρο κάθε ευθείας. Οι στροφές αντιμετωπίζονται ως κυκλικά τμήματα (καθώς το αμάξι κάνει κυκλική κίνηση κατά την στροφή). Υπολογίζεται το αντίστροφο της ακτίνας κάθε στροφής (καμπυλότητα) με το πρόσημο που αναλογεί σε κάθε τύπο στροφής.

```

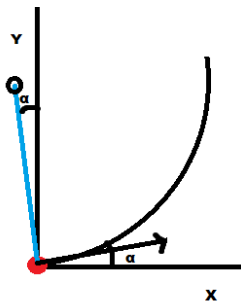
222      %% Πλέγμα (Meshing)
223
224      % νέος πίνακας θέσεων
225      if floor(L)<L % έλεγχος για ένθετο σημείο στο τέλος
226          x = [(0:mesh:floor(L))';L];
227      else
228          x = (0:mesh:floor(L))';
229      end
230      % βήμα πλέγματος
231      dx = diff(x);
232      dx = [dx;dx(end)]; tr.dx = dx;
233      % αριθμός σημείων πλέγματος
234      n = length(x); tr.n = n;
235      % πίνακας καμπυλότητας
236      r = interp1(xx,r,x,'pchip','extrap');
237      tr.r = r;
238

```

Εικόνα 23: Δημιουργία mesh πίστας

Στην παραπάνω εικόνα, ως πρώτο βήμα, ελέγχεται εάν το σύνολο των σημείων στον πίνακα είναι ίσο με την μεταβλητή L (στρογγυλοποιημένο προς την κοντινότερη μονάδα) και δημιουργείται πίνακας x ανάμεσα στο αρχικό και τελικό σημείο, με βήμα

την μεταβλητή mesh. Το πλέγμα σημείων της πίστας δημιουργείται παίρνοντας την διαφορά κάθε σημείου και αποθηκεύοντας στον πίνακα dx. Εισάγεται ένα συμπληρωματικό σημείο στο τέλος του πίνακα, που έχει την ίδια τιμή με το τελευταίο του στοιχείο. Στην μεταβλητή r αποθηκεύεται το αντίστροφο των ακτινών κάθε στροφής, καθώς θα χρησιμοποιηθεί για να βρεθούν οι κορυφές (apex) των στροφών. Στο επόμενο κομμάτι κώδικα, θα δημιουργηθεί δισδιάστατος χάρτης με όλα τα σημεία του χάρτη, τα οποία θα διαγραμμίζουν την χάραξη της πίστας. Κατά την στροφή, το όχημα κινείται κυκλικά γύρω από ένα σημείο το οποίο βρίσκεται κάθετα σε απόσταση ίση με την μεταβλητή R του κώδικα. Γεωμετρικά, η γωνία α που σχηματίζει η κατεύθυνση του οχήματος με κάποια νοητή γραμμή X είναι ίση με την γωνία που σχηματίζεται από την ακτίνα του οχήματος και μία δεύτερη νοητή γραμμή Y, κάθετη στην γραμμή X.



Εικόνα 24: Σχηματική απεικόνιση των γωνιών

```

239 %% Δημιουργία χάρτη
240
241 % αρχικοποίηση πινάκων συντεταγμένων
242 X = zeros(n,1);
243 Y = zeros(n,1);
244 % γωνίες τμημάτων στροφών
245 angle_seg = rad2deg(dx.*r);
246 % γωνίες κατεύθυνσης σε κάθε τμήμα
247 angle_head = cumsum(angle_seg);
248 dh = [mod(angle_head(end),sign(angle_head(end))*360);angle_head(end)-sign(angle_head(end))*360];
249 [~,idx] = min(abs(dh));
250 dh = dh(idx);
251 angle_head = angle_head-x/L*dh;
252 angle_seg = [angle_head(1);diff(angle_head)];
253 angle_head = angle_head-angle_head(1);
254 % δημιουργία χάρτη
255 for i=2:n
256     % προηγούμενο σημείο
257     p = [X(i-1);Y(i-1);0];
258     % επόμενο σημείο
259     xyz = rotz(angle_head(i-1))*[dx(i-1);0;0]+p;
260     % αποθήκευση συντεταγμένων επόμενου σημείου
261     X(i) = xyz(1);
262     Y(i) = xyz(2);
263 end
264

```

Εικόνα 25: Δημιουργία χάρτη

Με τα σημεία που είχαν εισαχθεί στις γραμμές 160-200, και εφόσον ο πίνακας r είναι το αντίστροφο της ακτίνας των αντίστοιχων σημείων, ο πίνακας $angle_seg$ είναι το πολλαπλάσιο του βήματος κάθε σημείο (πίνακας dx) με τον πίνακα r . Καθώς πολλαπλασιάζεται κάθε στοιχείο του ενός πίνακα με του άλλου, χρησιμοποιείται τελεία πριν τον αστερίσκο. Το αποτέλεσμα είναι ένας πίνακας με τις γωνίες κατεύθυνσης του οχήματος. Όταν το όχημα κοιτάει προς μια κατεύθυνση, για να αλλαχθεί η κατεύθυνση αυτή πρέπει να μεταβεί μέσα από όλες τις ενδιάμεσες κατευθύνσεις. Ο πίνακας $angle_head$ περιγράφει αυτή την κίνηση, με την εντολή `cumsum`, η οποία δημιουργεί έναν πίνακα ίσων διαστάσεων, προσθέτοντας σε κάθε σειρά όλα τα προϋπάρχοντα στοιχεία του πίνακα. Σε πίστες που η γραμμή εκκίνησης είναι και η γραμμή τερματισμού, η τελική γωνία πρέπει να είναι 360 μοίρες. Για αυτόν τον λόγο, μεταβλητή dh υπολογίζεται σε μορφή πίνακα 2×1 : Είναι το υπόλοιπο της ακέραιας διαίρεσης του τελευταίου στοιχείου του πίνακα $angle_head$, με το 360 (το οποίο έχει το ίδιο πρόσημο με το τελευταίο στοιχείο του ίδιου πίνακα). Στην κάτω στήλη

υπολογίζεται η διαφορά των ίδιων όρων. Στην μεταβλητή `idx` δίνεται ο απόλυτα μικρότερος από τους δύο όρους. Με την περισπωμένη στην πρώτη θέση της αγκύλης ο κώδικας αγνοεί την αποθηκευμένη τιμή του πίνακα και απλά επιστρέφει τη θέση της, η οποία δίνεται στην μεταβλητή `idx`. Με την εντολή `dh(idx)` δίνεται στον όρο `dh` η τιμή που χρειάζεται. Υπολογίζεται η `angle_head` και αφαιρείται ο πρώτος όρος. Μέσα στον βρόγχο `for`, υπολογίζεται κάθε σημείο `X` και `Y` της πίστας περιστρέφοντας γύρω από τον άξονα `z` με την εντολή `rotz`. Για να μην εμφανίσει σφάλμα ο κώδικας, προστέθηκε ένα `0` στην Τρίτη διάσταση του πίνακα το οποίο παραμένει έτσι και δεν χρησιμοποιείται πουθενά αλλού. Η πίστα θεωρείται τελείως δισδιάστατη, δηλαδή τελείως επίπεδη. Ο πίνακας περιστροφής μας δίνεται από το `documentation` του `MATLAB`:

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Εικόνα 26: Πίνακας περιστροφής

Οι τιμές αποθηκεύονται στον πίνακα `X` και στον πίνακα `Y` αντιστοίχως.

Ως τελικό κομμάτι είναι η εύρεση των σημείων που μας ενδιαφέρουν στον γύρο, δηλαδή οι κορυφές των στροφών. Ως κορυφή της στροφής αναφέρεται το πιο στενό κομμάτι της στροφής, στο οποίο ο οδηγός πρέπει να έχει την μικρότερη ταχύτητα. Οι κορυφές των στροφών, είναι τα τοπικά ακρότατα της καμπυλότητας του χάρτη. Στην συνάρτηση `simulate`, θα υπολογιστούν ξεχωριστά οι ταχύτητες για τις κορυφές από όλα τα υπόλοιπα σημεία.

Συμπληρωματικά, κατά την εισαγωγή δεδομένων μπορεί να έχει γίνει κάποιο λάθος με αποτέλεσμα το τελευταίο σημείο του χάρτη να μην συμπέφτει επάνω στο πρώτο κομμάτι του, δηλαδή η πίστα να μην είναι «κλειστή». Για να κλείσει η πίστα, αρκεί να

διαιρεθεί ο πίνακας x με το πολλαπλάσιο της μεταβλητής L και της διαφοράς του πρώτου με του τελευταίου σημείου της πίστας, και στις δύο διαστάσεις. Η λειτουργία αυτή δεν αλλάζει το ολικό μήκος της πίστας.

```

265      %% Κορυφές στροφών
266
267      % εύρεση κορυφών
268      [~,apex] = findpeaks(abs(r));
269      % διόρθωση τύπου στροφών
270      r_apex = r(apex);
271
272      %% Επεξεργασία χάρτη
273
274      % κλείσιμο χάρτη
275      % πίνακες διόρθωσης
276      DX = x/L*(X(1)-X(end));
277      DY = x/L*(Y(1)-Y(end));
278      % προσθήκη διόρθωσης
279      X = X+DX; tr.X = X;
280      Y = Y+DY; tr.Y = Y;
281      disp('Δημιουργία πίστας επιτυχής.')
282

```

Εικόνα 27: Εύρεση κορυφών και διόρθωση πίστας

Στο τέλος του κώδικα, ο χρήστης ενημερώνεται για την εκτέλεση των λειτουργιών με το μήνυμα «Δημιουργία πίστας επιτυχής».

Πριν την εκτέλεση της προσομοίωσης, ο χρήστης επίσης ενημερώνεται για την αποθήκευση όλων των μεταβλητών που θα χρησιμοποιηθούν στα αντίστοιχα structures «veh» και «tr».

```

282
283      %% Αποθήκευση των structure οχήματος και πίστας
284      veh.TrackFilename = TrackFilename;
285      veh.fz_total = fz_total; veh.ax_tyre_max_acc = ax_tyre_max_acc; veh.ax_tyre_max_dec = ax_tyre_max_dec; veh.ay_max = ay_max; veh.Aero_Dr = Aero_Dr; veh.Roll_Dr = Roll_Dr;
286      veh.wx = wx; veh.wy = wy; veh.wz = wz; veh.ax_drag = ax_drag; veh.v_max = v_max; veh.Aero_Df = Aero_Df; veh.factor_drive = factor_drive; veh.factor_aero = factor_aero;
287      veh.driven_wheels = driven_wheels; veh.ax_power_limit = max(ax_power_limit); veh.RollRes = RollRes;
288      disp('Η αποθήκευση των structure οχήματος και πίστας ήταν επιτυχής.')
289

```

Εικόνα 28: Αποθήκευση structures

5.2.4 Η συνάρτηση simulate

```

290      %% Προσομοίωση
291
292      [sim] = simulate(tr,veh);
293
294      disp('Τα αποτελέσματα είναι έτοιμα.')
295

```

Εικόνα 29: Κάλεσμα συνάρτησης simulate

Στην γραμμή 292, καλείται η συνάρτηση simulate. Είσοδος της συνάρτησης είναι τα

structures «tr» και «veh», και έξοδος είναι το structure «sim» το οποίο περιέχει τα αποτελέσματα της προσομοίωσης. Αφού περατωθεί η λειτουργία της συνάρτησης, ο χρήστης ενημερώνεται με το κατάλληλο μήνυμα «Τα αποτελέσματα είναι έτοιμα». Στο MATLAB, οι συναρτήσεις ενός κώδικα γράφονται στις τελευταίες γραμμές του κώδικα. Η συνάρτηση αρχίζει στην γραμμή 403.

```

402 %%
403 function[sim] = simulate(tr,veh)
404
405 % υπολογισμός καμπύλης μέγιστης ταχύτητας (υποθέτουμε πλήρως πλευρική κατάσταση)
406 v_max = single(zeros(tr.n,1));
407 tps_v_max = single(zeros(tr.n,1));
408 for i=1:tr.n
409     [v_max(i),tps_v_max(i)] = max_speed(veh,tr,i);
410 end
411
412 % εύρεση κορυφών
413 [v_apex,apex] = findpeaks(-v_max); % η εντολή λειτουργεί για τα μέγιστα, οπότε αντιστρέφουμε τις τιμές
414 v_apex = -v_apex; % αντιστρέφουμε για να έχουμε θετικές τιμές
415 apex = [1;apex]; % εισάγουμε τον πρώτο όρο ως κορυφή
416 v_apex = [1;v_apex]; % εισάγουμε αρχική ταχύτητα 1 m/s (λόγω διάταξης γραμμής εκκίνησης)
417
418 % ελέγχουμε εαν δεν υπάρχουν κορυφές και προσθέτουμε μία εάν ισχυρεί η συνθήκη
419 if isempty(apex)
420     [v_apex,apex] = min(v_max);
421 end
422 % ταξινομούμε τις κορυφές για βελτιστοποίηση του αλγορίθμου
423 apex_table = sortrows([v_apex,apex],1);
424 v_apex = apex_table(:,1);
425 apex = apex_table(:,2);
426 % υπολογίζουμε τη θέση του γκαζιού στην κορυφή
427 tps_apex = tps_v_max(apex);
428

```

Εικόνα 30: Αρχή συνάρτησης simulate

Η συνάρτηση simulate αρχικά καλεί άλλη μια συνάρτηση, την max_speed, με είσοδο τους μηδενικούς πίνακες v_max και tps_v_max. Ο v_max θα περιέχει την μέγιστη ταχύτητα σε κάθε σημείο της πίστας, και ο tps_v_max το μέγιστο ποσοστό γκαζιού στο αντίστοιχο σημείο. Η μοναδική χρήση του υπολογισμού του γκαζιού σε αυτή την εργασία είναι για τον υπολογισμό της κατανάλωσης του ηλεκτροκινητήρα, σε αργότερο σημείο. Αφού συμπληρωθεί ο πίνακας v_max από την συνάρτηση, τα τοπικά ελάχιστα του πίνακα υπολογίζονται με την εντολή findpeaks. Εισάγεται πριν τον πίνακα γιατί η εντολή findpeaks υπολογίζει τα τοπικά μέγιστα, οπότε τα τοπικά μέγιστα του αντίθετου πίνακα είναι τα τοπικά μέγιστα του κανονικού. Η μεταβλητή v_apex είναι τύπου double και περιέχει την ταχύτητα στην κορυφή (apex) σε m/s, ενώ η

μεταβλητή *apex* περιέχει το σημείο της κορυφής. Εισάγεται μια τιμή 1 στην αρχή του πίνακα *apex*, με αντίστοιχη αρχική ταχύτητα του γύρου 1 m/s. Αυτή η επιλογή έχει δύο λόγους: Εάν επιλεγθεί αρχική ταχύτητα 0, κατά τον υπολογισμό του χρόνου το μηδέν μπαίνει σε παρονομαστή και απειρίζεται το αποτέλεσμα. Δεύτερον, δεν πρόκειται να υπάρξει αρχική ταχύτητα 0 σε κανένα δυναμικό αγώνισμα, καθώς πάντα το όχημα ξεκινάει από μία δευτερεύουσα γραμμή πίσω από την γραμμή εκκίνησης – τερματισμού. Για να εξοικονομηθεί χρόνος στην εκτέλεση, γίνεται *sort* (ταξινόμηση) στους πίνακες. Αποθηκεύεται επίσης το ποσοστό γκαζιού (από 0 έως 100) στον πίνακα *tps_apex*.

Ακολουθεί ο κύριος βρόγχος της συνάρτησης. Αρχικοποιούνται τα δεδομένα που θα υπολογιστούν.

```

429     %% προσομοίωση
430
431     % αρχικοποίηση πινάκων
432     N = ((length(apex))); % αριθμός κορυφών στροφής
433     flag = false(tr.n,2); % ελέγχουμε με flag εάν η ταχύτητα έχει εκτιμηθεί σωστά
434     % η 1η διάσταση του πίνακα είναι ο αριθμός των σημείων στο πλέγμα της πίστας
435     % η 2η διάσταση είναι ο αριθμός των στροφών
436     % η 3η διάσταση του πίνακα είναι για τις 2 περιπτώσεις κίνησης (επιτάχυνση ή επιβράδυνση)
437     v = inf*ones(tr.n,N,2);
438     ax = zeros(tr.n,N,2);
439     ay = zeros(tr.n,N,2);
440     tps = zeros(tr.n,N,2);

```

Εικόνα 31: Αρχικοποίηση δεδομένων

Συγκεκριμένα, χρησιμοποιείται ο αριθμός των στροφών, ένα *flag* για να ελεγχθεί εάν έχει εκτιμηθεί σωστά η ταχύτητα, και μηδενικοί πίνακες για τα αποτελέσματα. Ο πίνακας *v* απειρίζεται σκόπιμα.

```

441 % εκτέλεση προσομοίωσης
442 for i=1:N % αριθμός στροφής
443     for k=(1:2) % αριθμός περίπτωσης
444         switch k
445             case 1 % περίπτωση 1 - επιτάχυνση
446                 mode = 1;
447                 k_rest = 2;
448             case 2 % περίπτωση 2 - επιβράδυνση
449                 mode = -1;
450                 k_rest = 1;
451         end
452         % εισάγουμε τις υπόλοιπες στροφές για έλεγχο αργότερα με τη συνάρτηση other_points
453         [i_rest] = other_points(i,N);
454         if isempty(i_rest)
455             i_rest = i;
456         end
457         % νούμερο στροφής
458         j = uint32(apex(i));
459         % αποθηκεύουμε την ταχύτητα, την πλευρική επιταχ. και το γκάζι από τις προλυμένες στροφές
460         v(j,i,k) = v_apex(i);
461         ay(j,i,k) = v_apex(i)^2*tr.r(j);
462         tps(j, :,1) = tps_apex(i)*ones(1,N);
463         tps(j, :,2) = tps_apex(i)*ones(1,N);
464         % θέτουμε το flag θετικό
465         flag(j,k) = true;
466         % παίρνουμε το νούμερο του επόμενου σημείου με τη συνάρτηση next_point
467         [j_next] = next_point(j,tr.n,mode);
468         % υποθέτουμε ότι έχουμε ίση ταχύτητα ακριβώς μετά την στροφή
469         v(j_next,i,k) = v(j,i,k);
470         % προχωράμε στο επόμενο σημείο
471         [j_next,j] = next_point(j,tr.n,mode);
472         while 1
473             % υπολογισμός ταχύτητας, επιταχύνσεων και γκαζιού με την συνάρτηση vehicle_model
474             [v(j_next,i,k),ax(j,i,k),ay(j,i,k),tps(j,i,k),overshoot] = vehicle_model(veh,tr,v(j,i,k),v_max(j_next),j,mode) ;
475             % ελέγχουμε αν υπερβήκαμε τα όρια του οχήματος με την λογική μεταβλητή overshoot
476             if overshoot
477                 break
478             end
479             % ελέγχουμε εάν έχουμε ξαναλύσει το ίδιο σημείο σε άλλη στροφή
480             if flag(j,k) || flag(j,k_rest)
481                 if max(v(j_next,i,k)>v(j_next,i_rest,k)) || max(v(j_next,i,k)>v(j_next,i_rest,k_rest))
482                     break
483                 end
484             end
485             % προχωράμε στο επόμενο σημείο
486             [j_next,j] = next_point(j,tr.n,mode);
487             % ελέγχουμε αν ολοκληρώθηκε ο γύρος
488             if j==apex(i) % φτάσαμε στην ίδια στροφή
489                 break
490             end
491             if j==1 % φτάσαμε στην εκκίνηση
492                 break
493             end
494         end
495     end
496 end

```

Εικόνα 32: Κύριος βρόγχος συνάρτησης

Υπάρχουν δύο εμφωλευμένοι βρόγχοι: Για κάθε στροφή (μεταβλητή N) υπολογίζονται 2 περιπτώσεις: Επιτάχυνση και επιβράδυνση. Η επιτάχυνση παίρνει την τιμή 1 στην μεταβλητή mode, ενώ η επιβράδυνση την τιμή -1. Καλείται η συνάρτηση other_points, η οποία δίνει έξοδο την μεταβλητή i_rest. Ύστερα, μεταθέεται το νούμερο της στροφής στον δείκτη j, και αποθηκεύονται οι αντίστοιχες ταχύτητες, επιταχύνσεις και γκάζι από την προηγούμενη συνάρτηση. Το flag παίρνει την τιμή true. Με την

συνάρτηση `next_point`, αποθηκεύεται το σημείο της επόμενης στροφής μαζί με τον δείκτη `j` και το `mode`. Η ταχύτητα αμέσως μετά το σημείο `apex` της στροφής θεωρείται ίδια με την ταχύτητα επάνω στο σημείο, και ύστερα καλείται πάλι η συνάρτηση `other_points` για τον υπολογισμό του δείκτη `j_next`. Γίνεται χρήση της συνάρτησης `vehicle_model` με είσοδο τα structures «tr», «veh», τον πίνακα `v`, τον πίνακα `v_max`, τον δείκτη `j` και την μεταβλητή `mode`. Ως έξοδος βγαίνουν ο πίνακας `v`, ο πίνακας `ax,ay`, ο πίνακας `tps` (throttle position sensor, ή αλλιώς το ποσοστό του γκαζιού) και η μεταβλητή `overshoot`. Όπως θα φανεί και αργότερα, η μεταβλητή `overshoot` γίνεται 1 όταν το αυτοκίνητο δεν έχει τις δυνατότητες να ανταπεξέλθει σε μία στροφή. Εάν η μεταβλητή είναι 1, σπάει ο βρόγχος.

Στην γραμμή 481, με τον λογικό τελεστή `||` ελέγχεται εάν έχει υπολογιστεί το ίδιο ακριβώς σημείο σε άλλη περίπτωση, για εξοικονόμηση χρόνου. Με κάλεσμα της συνάρτησης `next_point` ο κώδικας προχωράει στο επόμενο σημείο. Με χρήση του δείκτη `j` στον πίνακα `apex` ελέγχεται εάν τελειώσει ο γύρος, με επιστροφή στο ίδιο σημείο που άρχισε.

Η επεξεργασία και αποθήκευση των αποτελεσμάτων γίνεται μέσα στην συνάρτηση `simulate`. Υπάρχουν 4 εμφωλευμένες συναρτήσεις μέσα στην συνάρτηση `simulate` όμως, που πρέπει να μελετηθούν ξεχωριστά. Οι συναρτήσεις αυτές, όπως φάνηκε και από τον κώδικα, είναι η `max_speed`, η `vehicle_model`, η `next_point` και η `other_points`.

5.2.5 Η συνάρτηση max_speed

```
540 %%  
541 function [v,tps] = max_speed(veh,tr,p)  
542  
543 % αρχικοποίηση  
544 g = 9.81 ;  
545 r = tr.r(p) ;  
546 driven_wheels = veh.driven_wheels ;  
547 M = veh.M ; % μάζα  
548 Wz = M * g ; % βάρος στον άξονα z  
549 Wx = M*g ; % βάρος στον άξονα x  
550 Wy = M*g ; % βάρος στον άξονα y  
551 Mu_y = veh.Mu_y ; % εισαγωγή μεταβλητών από το structure του οχήματος  
552 Mu_x = veh.Mu_x ;  
---
```

Εικόνα 33: Αρχικοποίηση συνάρτησης

Γίνεται εισαγωγή των απαιτούμενων μεταβλητών από τα structure «veh» και «tr».

```
554 % υπολογισμός μέγιστης ταχύτητας  
555 if r==0 % ευθεία (περιορισμένη από τον κινητήρα ή την οπισθέλκουσα)  
556     % ελέγχουμε για το όριο ταχύτητας του κινητήρα  
557     v = veh.v_max ;  
558     tps = 1 ; % γκάζι στο 100%  
559 else % στροφή (περιορισμένη από τον κινητήρα, την οπισθέλκουσα ή το όχημα)  
560     % αρχικός υπολογισμός ταχύτητας  
561     D = -1/2*veh.AirDens*veh.Lift*veh.FrontArea ; % θέτουμε μεταβλητή αρνητικής άντωσης  
562     v = sqrt((Mu_y*((M/4)*g)-(Wz/4))+Wy)/(M*r+Mu_y*(D/8)); % λύνουμε ως προς v με ισορροπία των δυνάμεων  
563     % ελέγχουμε για το όριο ταχύτητας του κινητήρα  
564     v = min([v,veh.v_max]);
```

Εικόνα 34: Υπολογισμός μέγιστης ταχύτητας

Στον βρόγχο if της εικόνας υπολογίζεται η μέγιστη ταχύτητα που μπορεί να έχει το όχημα σε όλα τα σημεία της πίστας. Αν το σημείο είναι ευθεία, η μέγιστη ταχύτητα που μπορεί να έχει είναι η τελική ταχύτητα του οχήματος, και το γκάζι στο 100. Σε στροφή, χρειάζεται να υπολογισθεί η ταχύτητα του οχήματος με εξισώσεις κίνησης.

Γίνεται η υπόθεση ότι το όχημα βρίσκεται σε ισορροπία, δηλαδή οι δέχεται δυνάμεις στον άξονα z που έχουν το ίδιο μέτρο με την κεντρομόλο δύναμη της στροφής, και κάθετη διεύθυνση. Από την φυσική είναι γνωστό ότι αν οι δύο αυτές δυνάμεις είναι κάθετες και έχουν ίσο μέτρο, η συνισταμένη τους θα είναι μηδέν. Η ισορροπία αυτή περιγράφεται από την εξίσωση:

$$F_{y_t} - F_{y_m} = Mv^2r \quad (2.1)$$

Όπου F_{y_t} η πλευρική δύναμη του ελαστικού, F_{y_m} η πλευρική δύναμη της μάζας του οχήματος και Mu^2r ο τύπος της κεντρομόλου δύναμης του οχήματος, όπου M η μάζα

του οχήματος, v η ταχύτητα και r η ακτίνα της στροφής. Εάν το μέτρο από κάποιο από αυτά τα σκέλη της εξίσωσης είναι μεγαλύτερο από το άλλο, το αυτοκίνητο θα βγει εκτός πορείας. Η F_{y_m} είναι η μάζα του οχήματος πολλαπλασιασμένη με 9,81, ενώ η F_{y_t} περιγράφεται από την παρακάτω εξίσωση:

$$\mu y \left(\frac{M}{4} g - \frac{F_{z_{ολικό}}}{4} \right) \quad (2.2)$$

όπου μy ο πλευρικός συντελεστής τριβής των ελαστικών, M η μάζα του οχήματος, g η επιτάχυνση της βαρύτητας, και $F_{z_{ολικό}}$ η ολική δύναμη στον z άξονα, δηλαδή το βάρος του οχήματος στον z άξονα (η μάζα του οχήματος επί 9,91) συν ο συντελεστής D (η αρνητική άντωση χωρίς την ταχύτητα στην εξίσωση). Αντικαθιστώντας στην εξίσωση 2.1 βγαίνει:

$$\mu y \left(\frac{M}{4} - \frac{Wz + D}{4} \right) - Wy = Mv^2 r \quad (2.3)$$

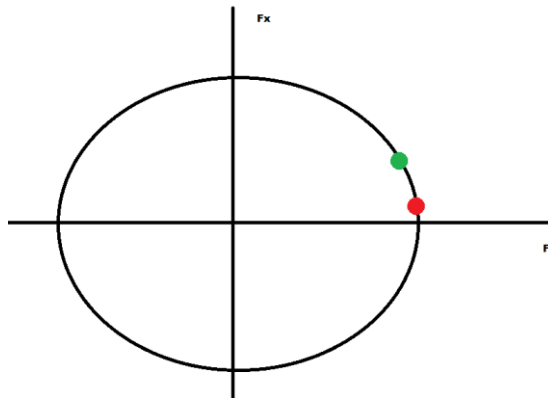
όπου μy ο πλευρικός συντελεστής τριβής των ελαστικών, M η μάζα του οχήματος, g η επιτάχυνση της βαρύτητας, Wy και Wz τα βάρη στον y και z άξονα αντίστοιχα, και D ο συντελεστής αρνητικής άντωσης. Λύνοντας ως προς v παράγεται η εξίσωση:

$$v = \sqrt{\frac{\mu y \left(\frac{M}{4} - \frac{Wz}{4} \right) + Wy}{M r + \mu y \frac{D}{8}}} \quad (2.4)$$

Στην γραμμή 564, ελέγχεται αν το αποτέλεσμα ξεπερνάει το όριο ταχύτητας του κινητήρα. Εάν ισχύει η συνθήκη επιλέγεται η μέγιστη τιμή που επιτρέπεται από τον κινητήρα.

Παρόλο που η πρώτη προσέγγιση είναι ικανοποιητική, χρειάζεται μείωση της ταχύτητας. Η λύση αυτή χρησιμοποιεί την μέγιστη πλευρική επιτάχυνση που μπορεί να προσφέρει το ελαστικό. Ως αποτέλεσμα, το ελαστικό δεν έχει αρκετή πρόσφυση για να αντισταθμίσει την αντίσταση του αέρα, η οποία τείνει να επιβραδύνει το όχημα. Αφού το ελαστικό χρειάζεται περισσότερη επιτάχυνση στον άξονα x , πρέπει να μετριαστεί η

επιτάχυνση στον άξονα y, για να παραμείνει επάνω στην έλλειψη που σχηματίζεται από τις μέγιστες δυνάμεις που μπορούν να υποστούν.



Εικόνα 35: Όρια πρόσφυσης ελαστικού

Στην υποθετική έλλειψη που σχηματίζεται στο σχεδιάγραμμα, το κόκκινο σημείο είναι η αρχική πλευρική δύναμη που υπολογίστηκε. Εάν τα ελαστικά δεν μπορούν να προσφέρουν πρόσφυση στον άξονα x, πρέπει να επιλεγθεί σε άλλο σημείο επάνω στην έλλειψη. Για να υπάρχει η βέλτιστη ταχύτητα σε κάθε σημείο της πίστας, ο αλγόριθμος πάντα υπολογίζει σημεία επάνω στις καμπύλες. Το πράσινο σημείο προσφέρει αρκετή πρόσφυση για να αντισταθμίσει την οπισθέλκουσα αλλά λιγότερη ταχύτητα επάνω στην στροφή. Στο παρακάτω απόσπασμα κώδικα υπολογίζονται οι σχετικές δυνάμεις χρησιμοποιώντας εξισώσεις που έχουν αναπτυχθεί ήδη. Όσο η λογική μεταβλητή `adjust_speed` είναι 1, υπολογίζονται η αντίσταση του αέρα, η αρνητική άντωση και η αντίσταση κύλισης. Η μέγιστη πλευρική επιτάχυνση των ελαστικών ορίζεται ως:

$$a_y = \frac{\text{signτης1}}{M} (\mu_y(W_z + D_f)) \quad (2.5)$$

όπου a_y η μέγιστη πλευρική επιτάχυνση του ελαστικού, sign το πρόσημο της μεταβλητής r , M η μάζα του οχήματος, μ_y ο πλευρικός συντελεστής τριβής, W_z το βάρος του οχήματος στον άξονα z και D_f η τιμή της αρνητικής άντωσης.

```

567         while adjust_speed
568             % αεροδυναμικά φορτία
569             Aero_Df = 1/2*veh.AirDens*veh.Lift*veh.FrontArea*v^2 ;
570             Aero_Dr = 1/2*veh.AirDens*veh.Drag*veh.FrontArea*v^2 ;
571             % αντίσταση κύλισης
572             Roll_Dr = veh.RollRes*(-Aero_Df+Wz) ;
573             % φορτία στους κινητήριους τροχούς
574             Wz = M*g;
575             % μέγιστη πλευρική επιτάχυνση διαθέσιμη από τα ελαστικά
576             ay_max = sign(r)/M*(Muy*(Wz-Aero_Df));
577             % επιβράδυνση λόγω της οπισθέλκουσας
578             ax_drag = (Aero_Dr+Roll_Dr+Wz)/M;
579             % απαιτούμενη πλευρική επιτάχυνση για να στρίψουμε στην ταχύτητα που θέλουμε
580             ay_needed = v^2*r ; % τύπος κεντρομόλου επιταχύνσεως στην κυκλική κίνηση
581             % υπολογισμός επιταχύνσεων και γκαζιού
582             if ax_drag<=0 % θέλουμε παραπάνω γκάζι για να αντισταθίσουμε την αντίσταση του αέρα (οπισθέλκουσα)
583                 ax_tyre_max_acc = 1/M*(Mux)*nd*driven_wheels; % υπολογίζουμε την μέγιστη διαμήκης επιτάχυνση που μπορούν να παρέχουν τα ελαστικά
584                 % διαθέσιμη διαμήκης επιτάχυνση στο ay που χρειαζόμαστε
585                 ax_acc = veh.ax_tyre_max_acc*sqrt(1-(ay_needed/ay_max)^2); % καμπύλη επιτάχυνσης
586                 % διαθέσιμη ay εάν τα ελαστικά δεν μπορούν να αντισταθίσουν την αντίσταση του αέρα
587                 ay = ay_max*sqrt(1-(ax_drag/ax_tyre_max_acc)^2);
588                 % ποσοστό γκαζιού
589                 scale = min([-ax_drag,ax_acc])/veh.ax_power_limit;
590                 trs = max([min([1,scale]),0]); % το γκάζι πρέπει να είναι θετικό ποσοστό
591             else % χρειαζόμαστε φρένο γιατί δεν έχουμε αρκετή πλευρική επιτάχυνση
592                 ax_tyre_max_dec = -1/M*Mux*(Wz-Aero_Df);
593                 % διαθέσιμη ay εάν τα ελαστικά δεν μπορούν να αντισταθίσουν την αντίσταση του αέρα
594                 ay = (ay_max*sqrt(1-(ax_drag/ax_tyre_max_dec)^2)); % καμπύλη επιτάχυνσης
595                 trs = 0 ; % αφήνουμε το γκάζι εάν πατάμε φρένο
596             end
597             % ελέγχουμε εάν τα ελαστικά μπορούν να παρέχουν την απαιτούμενη ay μαζί με την ax
598             if ay/ay_needed<1 % δεν έχουμε αρκετή πρόσφυση
599                 v = sqrt(abs(ay/r));
600             else % έχουμε αρκετή πρόσφυση
601                 adjust_speed = false ;
602             end
603         end
604     end
605 end

```

Εικόνα 36: Διόρθωση ταχύτητας

Η αντίσταση του αέρα υπολογίζεται όπως αναπτύχθηκε και πριν. Η πλευρική επιτάχυνση που απαιτείται υπολογίζεται χρησιμοποιώντας τον τύπο της κεντρομόλου επιτάχυνσης στην γραμμή 580. Εάν η συνθήκη του βρόγχου είναι αληθής, τότε η επιτάχυνση που απαιτείται είναι:

$$ax = ax_max \sqrt{1 - \left(\frac{drag}{ax_tyre}\right)^2} \quad (2.6)$$

όπου ax η ζητούμενη επιτάχυνση, ax_max η μέγιστη επιτάχυνση, drag η επιβράδυνση που οφείλεται στην αντίσταση του αέρα, και ax_tyre η μέγιστη επιτάχυνση που προσφέρουν τα ελαστικά. Συμπληρωματικά, υπολογίζεται το γκάζι με χρήση της μεταβλητής scale, η οποία είναι η μικρότερη τιμή ανάμεσα στην επιτάχυνση και την επιβράδυνση της οπισθέλκουσας, δια το όριο επιτάχυνσης του οχήματος. Στην δεύτερη περίπτωση του βρόγχου, το αμάξι πρέπει να επιβραδύνει για να απελευθερωθεί αρκετή πρόσφυση στα ελαστικά ώστε να υπάρχει αρκετή πλευρική επιτάχυνση. Κάνοντας χρήση της εξίσωσης 1.10 με παρόμοιο τρόπο υπολογίζεται η πλευρική επιτάχυνση που

εξυπηρετεί το ζητούμενο σκόπο. Στο γκάτζι δίνεται η τιμή μηδέν. Τελικά, ελέγχεται εάν τα ελαστικά μπορούν να προσφέρουν την ζητούμενη πλευρική επιτάχυνση, κλείνει ο βρόγχος και η μεταβλητή `adjust_speed` θέτεται ως `false`.

5.2.6 Η συνάρτηση `vehicle_model`

Η συνάρτηση `vehicle_model` αρχικά αποθηκεύει τα απαιτούμενα δεδομένα.

```

607     %%
608     function [v_next,ax,ay,tps,overshoot] = vehicle_model(veh,tr,v,v_max_next,j,mode)
609
610     % αρχικοποίηση
611     overshoot = false ;
612     % δεδομένα οχήματος και πίστας που χρειάζονται στους υπολογισμούς
613     dx = tr.dx(j);
614     r = tr.r(j);
615     Mux = veh.Mux;
616     wx = veh.wx;
617     M = veh.M;
618     Muy = veh.Muy;
619     Wz = veh.Wz;
620     Wy = veh.Wy;
621     AirDens = veh.AirDens;
622     Drag = veh.Drag;
623     Lift = veh.Lift;
624     FrontArea = veh.FrontArea;
625     if mode==1
626         factor_drive = veh.factor_drive ;
627         factor_aero = veh.factor_aero ;
628         driven_wheels = veh.driven_wheels ;
629     else
630         factor_drive = 1 ;
631         factor_aero = 1 ;
632         driven_wheels = 4 ;
633     end
634     Aero_Dr = 1/2*AirDens*Drag*FrontArea*v^2 ;
635     Aero_Df = 1/2*AirDens*Lift*FrontArea*v^2 ;
636     Roll_Dr = veh.RollRes*(-Aero_Df+Wz);
637     Wd = (factor_drive*Wz+(-factor_aero*Aero_Df))/driven_wheels;

```

Εικόνα 38: Αρχικοποίηση δεδομένων

Αρχικά υπολογίζεται η επιτάχυνση για να μην υπάρξει υπερβολική ταχύτητα στο επόμενο σημείο. “Υστερα, υπολογίζονται οι απαιτούμενες επιταχύνσεις στους άξονες x και y .

```

639     ax_max = mode*(v_max_next^2-v^2)/dx; % επιτάχυνση για να μην έχουμε υπερβολική ταχύτητα στο επόμενο σημείο
640     ax_drag = (Aero_Dr+Roll_Dr+wx)/M; % επιβράδυνση λόγω της αντίστασης του αέρα
641     ax_needed = ax_max-ax_drag; % υπολογισμός απαιτούμενης ax μαζί με οριοθέτηση ax_max
642     ay = v^2*r; % απαιτούμενη πλευρική επιτάχυνση

```

Εικόνα 39: Υπολογισμός επιταχύνσεων

Στον επόμενο βρόγχο, ανάλογα με το σημείο της πίστας, υπολογίζονται οι μέγιστες

πλευρικές επιταχύνσεις. Η μέγιστη πλευρική επιτάχυνση υπολογίζεται ως:

$$\frac{\text{sign}(a_y)1}{M} \mu y (Wz - Df) + Wy \quad (2.7)$$

όπου sign το πρόσημο της επιτάχυνσης, μy ο συντελεστής πλευρικής τριβής, M η μάζα του οχήματος, Wz , Wy τα βάρη στους άξονες y και z αντίστοιχα και Df η τιμή της αεροδυναμικής αρνητικής άντωσης.

```

644 % δυνάμεις ελαστικών
645 % συντελεστής καμπύλης τριβής ελαστικών
646 if sign(ay)~=0 % σε στροφή
647     % μέγιστη πλευρική επιτάχυνση διαθέσιμη από τα ελαστικά
648     ay_max = (sign(ay)*1/M*mu_y*(Wz-Aero_Df)+Wy) ;
649     % μέγιστη πλευρική επιτάχυνση διαθέσιμη από τα ελαστικά
650     if abs(ay/ay_max)>1 % ελέγχουμε για υπερβολική ταχύτητα (για να μην εμφανιστούν μιγαδικοί στα αποτελέσματα)
651         ellipse_multi = 0 ;
652     else
653         ellipse_multi = sqrt(1-(ay/ay_max)^2); % για να μείνουμε επάνω στην καμπύλη
654     end
655 else % ευθεία
656     ellipse_multi = 1 ;
657 end

```

Εικόνα 40: Εύρεση δεκτών τιμών επιτάχυνσης

Γίνεται χρήση της ίδιας μεθόδου με την εξίσωση 2.6 για να παραμείνει το όχημα στα δεδομένα όρια πρόσφυσης. Στον παρακάτω βρόγχο υπολογίζονται οι επιταχύνσεις και το ποσοστό γκαζιού που χρειάζεται για να παραμείνει το όχημα στα όρια, λόγω της επιβράδυνσης από την αντίσταση του αέρα. Όλες οι εξισώσεις έχουν εμφανιστεί ξανά στον κώδικα και χρησιμοποιείται πανομοιότυπη μεθοδολογία. Στην γραμμή 683, υπολογίζεται η επόμενη ταχύτητα του οχήματος, με βάση το βήμα dx της πίστας και την επιτάχυνση a_x που έχει υπολογιστεί. Εάν βρίσκεται σε ευθεία το όχημα, το γκάζι παίρνει αυτόματα τιμή 100, εάν δεν είναι ήδη.

```

659     if ax_needed>=0 % χρειάζεται γκάζι
660         % μέγιστη ax διαθέσιμη από τα κινήτρια ελαστικά
661         ax_tyre_max_acc = 1/M*Mux*wd*driven_wheels;
662         ax_tyre_max = ax_tyre_max_acc*driven_wheels *wd;
663         % μέγιστη ax διαθέσιμη από τα ελαστικά περιορισμένη από την καμπύλη
664         ax_tyre = ax_tyre_max*ellipse_multi;
665         % όριο δύναμης κινήτρια
666         ax_power_limit = 1/M*(interp1(veh.vehicle_speed,veh.fx_engine,v,'linear',0));
667         % τιμή γκαζιού
668         scale = min([ax_tyre,ax_needed]/ax_power_limit) ;
669         tps = max([min([1,scale]),0]);
670         % τελικός υπολογισμός ax
671         ax_com = tps*ax_power_limit;
672     else % χρειάζεται φρένο
673         ax_tyre_max_dec = -1/M*(Mux)*(Wz-Aero_Df);
674         % μέγιστη ax διαθέσιμη από όλα ελαστικά
675         ax_tyre = ax_tyre_max_dec*ellipse_multi;
676         tps = 0 ; % γκάζι = 0
677         % τελικός υπολογισμός ax
678         ax_com = -min(-[ax_tyre,ax_needed]);
679     end
680     % συνολική ax
681     ax = ax_com+ax_drag ;
682     % επόμενη ταχύτητα
683     v_next = sqrt(v^2+2*mode*ax*tr.dx(j));
684     % θέτουμε την τιμή του γκαζιού στο 100 στις ευθείες
685     if tps>0 && v/veh.v_max>=0.999
686         tps = 1 ;
687     end

```

Εικόνα 41: Υπολογισμός γκαζιού και διόρθωση επιταχύνσεων

Ο τελευταίος βρόγχος της συνάρτησης ελέγχει για υπερβολική ταχύτητα. Εάν τηρηθεί η συνθήκη, η μεταβλητή overshoot γίνεται ορθή (true), μηδενίζονται όλες οι τιμές εξόδου της συνάρτησης και ο αλγόριθμος βγαίνει από τον βρόγχο της συνάρτησης.

```

688     % έλεγχος για υπερβολική ταχύτητα
689     if v_next/v_max_next>1
690         % θέτουμε την μεταβλητή ως true
691         overshoot = true;
692         % θέτουμε τις τιμές
693         v_next = inf;
694         ax = 0;
695         ay = 0;
696         tps = -1;
697         return
698     end
699 end

```

Εικόνα 42: Συνθήκη υπερβολικής ταχύτητας

5.2.7 Η συνάρτηση next_point

```
%%  
function [j_next,j] = next_point(j,j_max,mode)  
switch mode  
case 1 % επιτάχυνση  
if j==j_max-1  
j = j_max;  
j_next = 1;  
elseif j==j_max  
j = 1;  
j_next = j+1;  
else  
j = j+1;  
j_next = j+1;  
end  
  
case -1 % επιβράδυνση  
if j==2  
j = 1;  
j_next = j_max;  
elseif j==1  
j = j_max;  
j_next = j-1;  
else  
j = j-1;  
j_next = j-1;  
end  
  
end  
end
```

Εικόνα 43: Η συνάρτηση next_point

Η συνάρτηση έχει ως είσοδο τον δείκτη j , j_max και την μεταβλητή $mode$. Έξοδοι είναι η j_next και j . Η συνάρτηση εισέρχεται σε έναν βρόγχο switch με 2 περιπτώσεις. Στην περίπτωση 1, την περίπτωση της επιτάχυνσης, εάν ο δείκτης j είναι μικρότερος από τον δείκτη j_max κατά ένα, τότε αλλάζει τιμή και γίνεται ο j_max , και ο j γίνεται 1. Αυτή η συνθήκη εκπληρώνεται όταν ο δείκτης j φτάνει στο τέλος του γύρω κατά την επιτάχυνση. Κατά την περίπτωση -1, εάν ο δείκτης j είναι ίσος με 2, ο j_next είναι ίσος με τον j_max , και ο j γίνεται 1. Εάν ο j είναι ήδη ένα, γίνεται j_max και ο j_next γίνεται $j-1$. Εάν δεν ισχύει καμία από τις δύο συνθήκες, αφαιρείται 1 από τον j και ο j_next γίνεται ο j .

5.2.8 Η συνάρτηση other_points

```
729 %%  
730 function [i_rest] = other_points(i,i_max)  
731     i_rest = (1:i_max)';  
732     i_rest(i) = [];  
733 end
```

Εικόνα 11: Η συνάρτηση other_points

Η συνάρτηση έχει ως είσοδο την μεταβλητή i και την i_max . Έξοδος είναι η i_rest . Ως i_rest θέτει έναν πίνακα μιας στήλης που ξεκινάει από το 1 μέχρι το i_max . Η παρούσα θέση i στον πίνακα i_rest διαγράφεται με τον τελεστή `[]`.

5.2.9 Κώδικας πίνακα αποτελεσμάτων

Η συνέχεια της συνάρτησης simulate ασχολείται με την επεξεργασία και αποθήκευση αποτελεσμάτων.

```
498 % αρχικοποίηση αποτελεσμάτων  
499 V = zeros(tr.n,1);  
500 AX = zeros(tr.n,1);  
501 AY = zeros(tr.n,1);  
502 TPS = zeros(tr.n,1);  
503  
504 % επιλογή λύσης  
505 for i=1:tr.n  
506     IDX = length(v(i,:),1));  
507     [V(i),idx] = min([v(i,:),1),v(i,:),2]); % η σειρά του k στο v(i,:,K) μέσα στο min() πρέπει να είναι ίδια με την σειρά περιπτώσεων έτσι ώστε να μην αγνοήσουμε τις σωστές τιμές  
508     if idx<=IDX % λύθηκε στην περίπτωση 1 (επιτάχυνση)  
509         AX(i) = ax(i,idx,1);  
510         AY(i) = ay(i,idx,1);  
511         TPS(i) = trs(i,idx,1);  
512     else % λύθηκε στην περίπτωση 2 (επιβράδυνση)  
513         AX(i) = ax(i,idx-IDX,2);  
514         AY(i) = ay(i,idx-IDX,2);  
515         TPS(i) = trs(i,idx-IDX,2);  
516     end  
517 end  
518 end
```

Εικόνα 45: Επιλογή και αποθήκευση αποτελεσμάτων

Εφόσον υπάρχουν δύο ομάδες αποτελεσμάτων (μια για κάθε mode) επιλέγονται τα αποτελέσματα στον παρακάτω βρόγχο.

```
% επιλογή λύσης  
for i=1:tr.n  
    IDX = length(v(i,:),1);  
    [V(i),idx] = min([v(i,:),1),v(i,:),2]); % η σειρά τ  
    if idx<=IDX % λύθηκε στην περίπτωση 1 (επιτάχυνση)  
        AX(i) = ax(i,idx,1);  
        AY(i) = ay(i,idx,1);  
        TPS(i) = trs(i,idx,1);  
    else % λύθηκε στην περίπτωση 2 (επιβράδυνση)  
        AX(i) = ax(i,idx-IDX,2);  
        AY(i) = ay(i,idx-IDX,2);  
        TPS(i) = trs(i,idx-IDX,2);  
    end  
end
```

Εικόνα 46: Επιλογή σωστών αποτελεσμάτων

Στον παραπάνω βρόγχο, με τον δείκτη idx επιλέγεται η μικρότερη ταχύτητα και

αποθηκεύονται τα δεδομένα που αντιστοιχούν σε αυτή. Στο τελευταίο κομμάτι της συνάρτησης, υπολογίζεται ο χρόνος, η κατανάλωση ενέργειας και αποθηκεύονται η ταχύτητα, οι επιταχύνσεις στο structure «sim». Επίσης, λόγω της διάταξης της γραμμής εκκίνησης δίνεται πάλι αρχική ταχύτητα 1 m/s, ένα το αρχείο πίστας είναι το acceleration.

```

519 % υπολογισμός χρόνου
520 if strcmpi(veh.Trackfilename,'Acceleration.xlsx') == 1
521     V(1) = 1; % διόρθωση αρχικής ταχύτητας στο Acceleration
522 end
523 sim.V = V; % αποθήκευση στο structure sim
524 time = cumsum(tr.dx./V); % ο χρόνος είναι το σύνολο του διαφορικού της θέσης προς την ταχύτητα (t = s/v)
525 sim.time = time; % αποθήκευση στο structure sim
526 laptime = time(end);
527 sim.laptime = laptime; % αποθήκευση στο structure sim
528 % κατανάλωση ενέργειας
529 engine_power = TPS.*interp1(veh.vehicle_speed,veh.engine_power,V,'linear','extrap'); % υπολογισμός κατανάλωσης με γραμμική παρεμβολή στην καμπύλη ταχύτητας και τυχόν με συντελεστή
530 energy_cons = cumsum(engine_power.*tr.dx/veh.Eff)/360000; % μετατροπή σε κιλοβατώρες
531 energy_cons_total = energy_cons(end);
532 sim.energy_cons = energy_cons; % αποθήκευση στο structure sim
533 sim.energy_cons_total = energy_cons_total; % αποθήκευση στο structure sim
534 % γκάζι και επιταχύνσεις
535 sim.TPS = TPS; % αποθήκευση στο structure sim
536 sim.AX = AX; % αποθήκευση στο structure sim
537 sim.AY = AY; % αποθήκευση στο structure sim

```

Εικόνα 47: Υπολογισμός χρόνου και κατανάλωσης ενέργειας

Συγκεκριμένα, ο χρόνος υπολογίζεται διαιρώντας κάθε στοιχείο της dx με κάθε στοιχείο του πίνακα V και προσθέτοντας συνολικά όλα τα αθροίσματα με την εντολή cumsum (cumulative sum). Εάν σε οποιοδήποτε σημείο η V γίνει μηδέν, αφού είναι σε παρονομαστή το αποτέλεσμα θα γίνει άπειρο. Η κατανάλωση ενέργειας υπολογίζεται με γραμμική παρεμβολή στην καμπύλη που δημιουργείται από τους πίνακες vehicle_speed και engine_power, στις συντεταγμένες κάθε στοιχείου του πίνακα V. Για την πραγματική κατανάλωση, διαιρείται με τον συντελεστή απόδοσης και μετατρέπεται σε κιλοβατώρες. Συμπληρωματικά με τα βασικά αποτελέσματα, στις γραμμές 606 έως 621 υπολογίζονται δυναμικά αποτελέσματα που αφορούν τα φορτία του οχήματος κατά την διάρκεια του γύρου. Γίνεται χρήση της εξίσωσης της αεροδυναμικής άντωσης για να υπολογιστεί η τιμή της για κάθε τιμή της V, η οποία είναι η ταχύτητα του οχήματος σε κάθε σημείο του γύρου. Αφαιρώντας την τιμή αυτή στην μάζα του οχήματος (επειδή η αρνητική άντωση έχει αρνητικό πρόσημο) υπολογίζεται το συνολικό φορτίο του οχήματος στο z άξονα (μεταβλητή Total_Load). Στην παρούσα εργασία, γίνεται η

υπόθεση ότι το σημείο που ασκούνται οι αεροδυναμικές φορτίσεις (κέντρο πίεσης η αλλιώς center of pressure) συμπίπτει με το κέντρο βαρύτητας του οχήματος.

```

606 DF = (1/2)*veh.AirDens*veh.Lift*veh.FrontArea.*sim.V;
607 Total_Load = (veh.M * 9.81) - DF;
608 F_WheelRate = veh.FStiff*veh.FrontMR^2 * cosd(veh.FAngle);
609 R_WheelRate = veh.RStiff*veh.RearMR^2 * cosd(veh.RAngle);
610 TotalStiff = (F_WheelRate * 2) + (R_WheelRate * 2);
611 dRh = (DF .* sqrt((veh.WB/2)^2 + (veh.TR/2)^2));
612 dRh = dRh./TotalStiff;
613 sim.dRh = dRh;
614 RideHeight = veh.RideHgt + dRh;
615 sim.RideHeight = RideHeight * 1000;
616 Front_Axle_tq = veh.WtDist * (Total_Load * sqrt((veh.WB/2)^2 + (veh.TR/2)^2));
617 Rear_Axle_tq = (1 - veh.WtDist) * (Total_Load * sqrt((veh.WB/2)^2 + (veh.TR/2)^2));
618 FAz = Front_Axle_tq; sim.FAz = FAz;
619 RAz = Rear_Axle_tq; sim.RAz = RAz;
620 FAy = ((AY * veh.M) * veh.WtDist)/2 ; sim.FAy = FAy;
621 RAy = ((AY * veh.M) * (1- veh.WtDist))/2 ; sim.RAy = RAy;
622
623 end

```

Υπολογίζεται την μεταβλητή Wheel Rate για τους μπροστά και πίσω τροχούς. Με τον όρο Wheel Rate (σκληρότητα τροχού) εννοείται η δύναμη που χρειάζεται για να μετατοπιστεί ένα χιλιοστό ο τροχός. Γίνεται η υπόθεση ότι οι τροχοί έχουν σταθερή αυτή την σκληρότητα καθ' όλη την διάρκεια της διαδρομής τους. Υπολογίζεται με την παρακάτω εξίσωση:

$$k MR^2 \cos \alpha \quad (2.8)$$

όπου k η σκληρότητα του ελατηρίου, MR ο λόγος κίνησης του τροχού (μετατόπιση ελατηρίου προς μετατόπιση τροχού) και α η γωνία του ελατηρίου της ανάρτησης με το έδαφος. Γίνεται χρήση της συνάρτησης cosd για να ληφθεί υπόψιν η μεταβλητή FAngle και RAngle σε μοίρες. Γίνεται υπόθεση ότι και τα 4 ελατήρια των αναρτήσεων του οχήματος είναι παράλληλα, οπότε η συνισταμένη τους είναι το άθροισμα τους. Το όχημα για να είναι συμβατό με τους κανονισμούς πρέπει να έχει ύψος 30 χιλιοστά με τον οδηγό μέσα, οπότε αρκεί να υπολογιστεί η τιμή της αεροδυναμικής αρνητικής άντωσης για να βρεθεί η μεταβολή του ύψους κατά τον γύρο. Η μεταβολή υπολογίζεται ως:

$$Df \sqrt{\frac{WB^2}{2} + \frac{TR^2}{2}} \quad (2.9)$$

Όπου Df η τιμή της αρνητικής άντωσης και WB, TR το μεταξόνιο και το μετατρόχιο αντίστοιχα. Ουσιαστικά, υπολογίζεται η ροπή που ασκείται από την αεροδυναμική άντωση σε κάποιον τροχό (όλοι οι τροχοί ισαπέχουν από το κέντρο βαρύτητας). Η μεταβολή αυτή αποθηκεύεται στην μεταβλητή dRh και διαιρείται με την συνολική σκληρότητα του οχήματος. Το αποτέλεσμα είναι η μεταβολή της απόστασης από το έδαφος για κάθε σημείο της πίστας, σε έναν πίνακα ιδίων διαστάσεων με την V.

Προσθέτοντας την αρχική τιμή της απόστασης σε κάθε στοιχείο του πίνακα επιφέρει το αποτέλεσμα που χρειάζεται, το οποίο πολλαπλασιάζεται επί 1000 για να είναι σε χιλιοστά. Υπενθυμίζεται ότι το μοντέλο του οχήματος είναι σταθερής κατάστασης, και το όχημα βρίσκεται σε ισορροπία σε κάθε κατάσταση. Ως αποτέλεσμα, δεν λαμβάνονται υπόψιν αλλαγές στην κατανομή βάρους, πλευρικά είτε δια μήκους.

Με παρόμοιο τρόπο με την μεταβλητή dRh υπολογίζονται και οι ροπές στον μπροστά και τον πίσω άξονα με μοναδική διαφορά ότι λαμβάνεται υπόψιν η κατανομή βάρους στο όχημα. Οι ροπές αυτές ασκούνται στον άξονα z και αποθηκεύονται στις μεταβλητές FAz και RAz αντίστοιχα. Οι δυνάμεις στον y άξονα στους μπροστά και τους πίσω άξονες του οχήματος είναι το πηλίκο της συνολικής κεντρομόλου επιτάχυνσης του οχήματος με την μάζα του οχήματος (θεμελιώδης νόμος της μηχανικής) επί τον συντελεστή κατανομής βάρους. Οι δυνάμεις αποθηκεύονται στις μεταβλητές FAy και RAy για τον μπροστά και πίσω άξονα αντίστοιχα.

Το πρόγραμμα εξέρχεται από την συνάρτηση στην γραμμή 623 και επιστρέφει στην γραμμή 303. Χρησιμοποιώντας τις εντολές figure και subplot, φτιάχνονται δύο παράθυρα με όλα τα αποτελέσματα του κώδικα.

```

296 % Αποτελέσματα
297
298 disp('Διαγράμμιση αποτελεσμάτων...')
299
300 % figure
301 set(0,'units','pixels');
302 SS = get(0,'screensize');
303 H = 900;
304 W = 1600;
305 Xpos = floor((SS(3)-W)/2);
306 Ypos = floor((SS(4)-H)/2);
307 f = figure('Name','Results','Position',[Xpos,Ypos,W,H]);
308 % σειρές και στήλες figure
309 rows = 8;
310 cols = 2;
311 if strcmpi(TrackFilename,'Acceleration.xlsx') == 1
312     acceleration = true;
313 else
314     acceleration = false;
315 end
316 % κινητήρας
317 subplot(rows,cols,1)
318 hold on
319 title('Χαρακτηριστικά ηλεκτροκινητήρα')
320 xlabel('Στροφές κινητήρα (rpm)')
321 yyaxis left
322 plot(rpm_curve,torque_curve)
323 ylabel('Ροπή κινητήρα (Nm)')
324 grid on
325 xlim([rpm_curve(1),rpm_curve(end)])
326 yyaxis right
327 plot(rpm_curve,en_power_curve/745.7)
328 ylabel('Ισχύς κινητήρα (Hp)')
329
330 % πλευρική επιτάχυνση
331 subplot(rows,cols,3)
332 hold on
333 title('Πλευρική επιτάχυνση (Ay)')
334 xlabel('Απόσταση (m)')
335 plot(sim.AY)
336 ylabel('Πλευρική επιτάχυνση (m/s^2)')
337 grid on
338
339 % διαμήκης επιτάχυνση
340 subplot(rows,cols,5)
341 hold on
342 title('Διαμήκης επιτάχυνση (Ax)')
343 xlabel('Απόσταση (m)')
344 plot(sim.AX)
345 ylabel('Διαμήκης επιτάχυνση (m/s^2)')
346 grid on
347
348 % διάγραμμα g-g-v
349 subplot(rows,cols,[2,4,6,8])
350 hold on
351 title('Διάγραμμα G-G-V')
352 surf(GGV(:,1:2),GGV(:,1),GGV(:,3),'EdgeAlpha',0.3,'FaceAlpha',0.8)
353 grid on
354 xlabel('Πλευρική επιτάχυνση (m/s^2)')
355 ylabel('Διαμήκης επιτάχυνση (m/s^2)')
356 zlabel('Ταχύτητα (m/s)')
357 view(185,5)
358 set(gca,'DataAspectRatio',[1 1 0.8])
359 cbar = colorbar;
360 set(get(cbar,'Title'),'String','Ταχύτητα (m/s)')
361 scatter3(sim.AY,sim.AX,sim.V,'ro','filled','MarkerEdgeColor',[0,0,0])
362
363 % χάρτης
364 subplot(rows,cols,[9,15])
365 hold on
366 grid on
367 axis equal
368 axis tight
369 xlabel('x [m]')
370 ylabel('y [m]')
371

```

```

372 set(get(cbar,'Title'),'String','Ταχύτητα (m/s)')
373 scatter3(sim.AY,sim.AX,sim.V,'ro','filled','MarkerEdgeColor',[0,0,0])
374
375 % χάρτης
376 subplot(rows,cols,[9,15])
377 hold on
378 grid on
379 axis equal
380 axis tight
381 xlabel('x [m]')
382 ylabel('y [m]')
383 scatter(X,Y,5,sim.V*3.6)
384 cbar = colorbar;
385 set(get(cbar,'Title'),'String','Ταχύτητα (km/h)')
386 axis equal
387
388 % ταχύτητα
389 subplot(rows,cols,7)
390 title('Ταχύτητα')
391 hold on
392 grid on
393 plot(sim.V*3.6)
394 xlabel('Απόσταση (m)')
395 ylabel('Ταχύτητα (km/h)')
396
397 % αποτελέσματα
398 subplot(rows,cols,[12,14])
399 axis off
400 if acceleration == true
401     text(0, 0.8, "Ο χρόνος για τα 75 μέτρα είναι "+(sim.time(75/mesh) - sim.time(1))+ ' δευτερόλεπτα.', 'FontSize',20)
402     sim.energy_cons = cumsum(sim.energy_cons(2:75/mesh));
403     sim.energy_cons = sim.energy_cons(end);
404     text(0, 0.4, "Η κατανάλωση ενέργειας είναι "+sim.energy_cons+ ' kWh.', 'FontSize',20)
405 else
406     text(0, 0.8, "Ο χρόνος πέρασματος είναι "+sim.laptime+ ' δευτερόλεπτα.', 'FontSize',20)
407     text(0, 0.4, "Η κατανάλωση ενέργειας είναι "+sim.energy_cons_total+ ' kWh.', 'FontSize',20)
408 end
409
410 % figure 2
411 set(0,'units','pixels');
412 SS = get(0,'screensize');
413 H = 900;
414 W = 1600;
415 Xpos = floor((SS(3)-W)/2);
416 Ypos = floor((SS(4)-H)/2);
417 g = figure('Name','Dynamic Results','Position',[Xpos,Ypos,W,H]);
418 % σειρές και στήλες figure
419 rows = 5;

```



```

422 % ύψος οχήματος
423 subplot(rows,cols,1)
424 hold on
425 title('Ύψος οχήματος (mm)')
426 xlabel('Απόσταση (m)')
427 plot(sim.RideHeight)
428 ylabel('Ύψος (mm)')
429 grid on
430
431 % δυνάμεις μπροστά άξονα στον άξονα z
432 subplot(rows,cols,2)
433 hold on
434 title('Δυνάμεις Μπροστά Άξονα (N) (z άξονας)')
435 xlabel('Απόσταση (m)')
436 plot(sim.FAz)
437 ylabel('Δύναμη(N)')
438 grid on
439
440 % δυνάμεις πίσω άξονα στον άξονα z
441 subplot(rows,cols,3)
442 hold on
443 title('Δυνάμεις Πίσω Άξονα (N) (z άξονας)')
444 xlabel('Απόσταση (m)')
445 plot(sim.RAz)
446 ylabel('Δύναμη(N)')
447 grid on
448
449 % δυνάμεις μπροστά άξονα στον άξονα y
450 subplot(rows,cols,4)
451 hold on
452 title('Δυνάμεις Μπροστά Άξονα (N) (y άξονας)')
453 xlabel('Απόσταση (m)')
454 plot(sim.FAy)
455 ylabel('Δύναμη(N)')
456 grid on
457 % δυνάμεις πίσω άξονα στον άξονα y
458 subplot(rows,cols,5)
459 hold on
460 title('Δυνάμεις Πίσω Άξονα (N) (y άξονας)')
461 xlabel('Απόσταση (m)')
462 plot(sim.RAy)
463 ylabel('Δύναμη(N)')
464 grid on
465 disp('Άνοιγμα αποτελεσμάτων...')
466

```

Εικόνα 48: Εμφάνιση αποτελεσμάτων

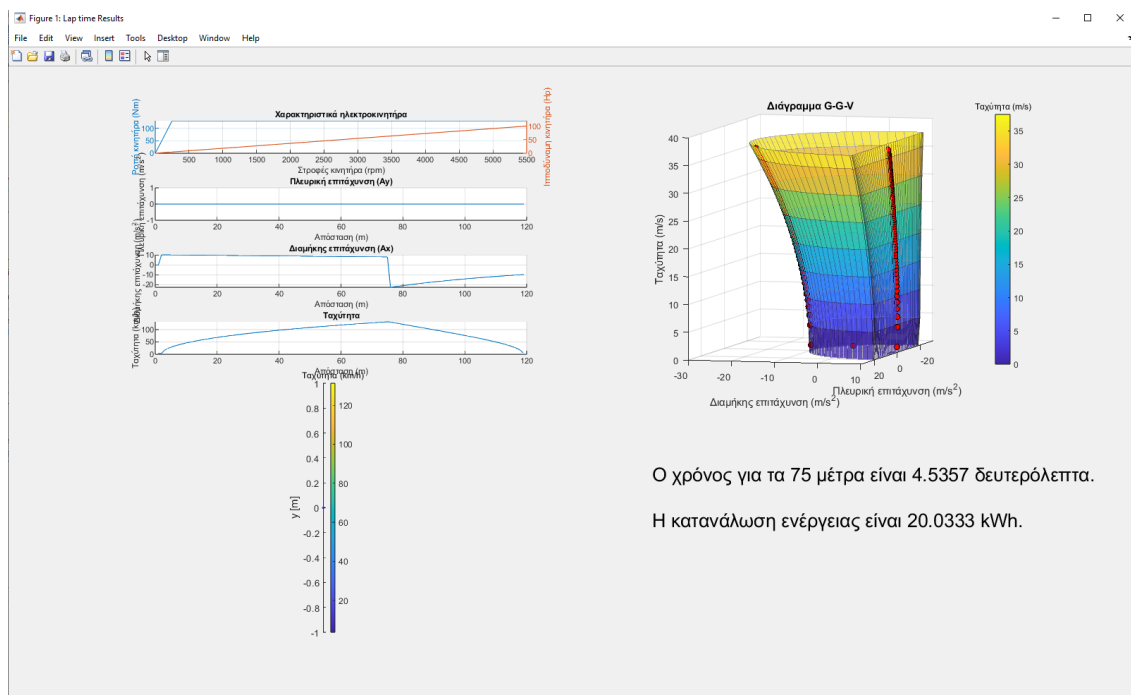
Διαγραμμίζονται οι διαμήκης και πλευρικές επιταχύνσεις, η ταχύτητα, τα χαρακτηριστικά του κινητήρα, το διάγραμμα G-G-V, ο χάρτης της πίστας και αναγράφεται ο χρόνος και η κατανάλωση ενέργειας του οχήματος. Δίνεται προσοχή στο αγώνισμα του Acceleration έτσι ώστε να εμφανίζονται αποτελέσματα μόνο για τα πρώτα 75 μέτρα. Στο δεύτερο figure, εμφανίζονται τα εξής διαγράμματα: Ύψος οχήματος, δυνάμεις στον μπροστά και πίσω άξονα στον άξονα z, και δυνάμεις στον μπροστά και πίσω άξονα στον άξονα y.

Κεφάλαιο 6

Αποτελέσματα προσομοίωσης

6.1 Πίνακες αποτελεσμάτων

Για την προσομοίωση το όχημα που περιγράφεται στις εικόνες 12 και 13 είναι αυτό που προσομοιώθηκε και mesh (πλέγμα πίστας) 1m. Χρησιμοποιώντας 4 φύλλα Excel που αντιστοιχούν σε πίστες των δυναμικών αγωνισμάτων, εμφανίζονται τα παρακάτω αποτελέσματα:

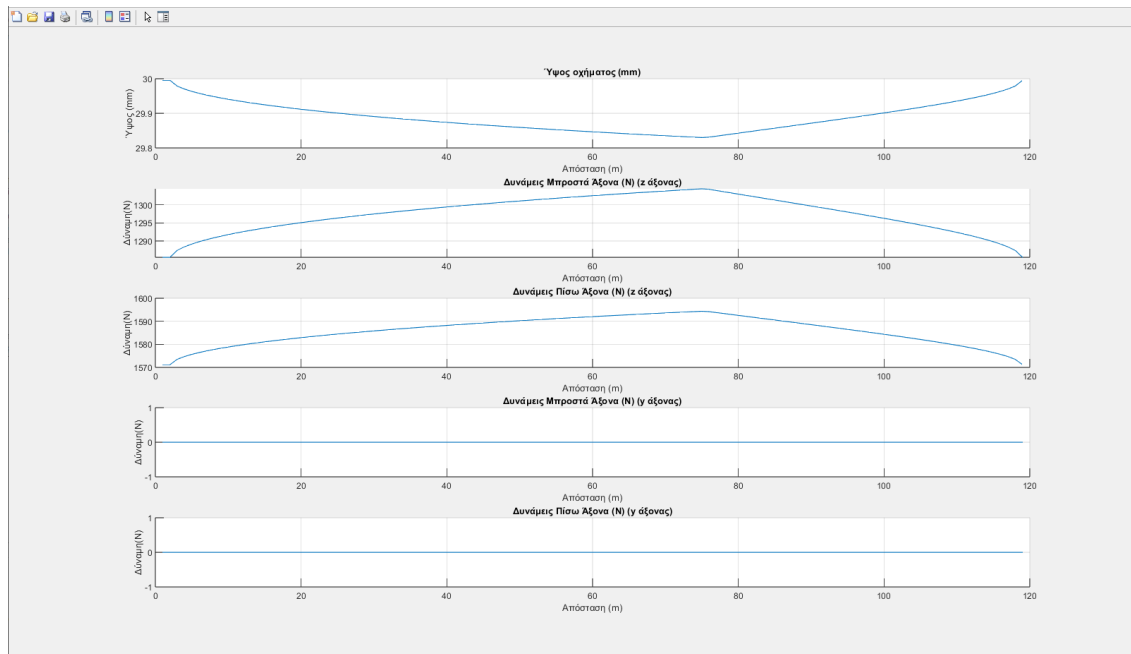


Εικόνα 49.1: Αποτελέσματα Acceleration

Αρχικά, από τα χαρακτηριστικά κινητήρα φαίνεται ότι η ροπή παραμένει σταθερή κατά το μεγαλύτερο μέρος της λειτουργίας του κινητήρα, όπως είναι αναμενόμενο από έναν ηλεκτροκινητήρα. Η πορτοκαλί καμπύλη έχει ακρότατο στα 100 άλογα, στις 5500 στροφές. Το διάγραμμα αυτό μένει σταθερό σε όλα τα αποτελέσματα, καθώς χρησιμοποιήθηκε το ίδιο φύλλο Excel.

Το διάγραμμα πλευρικής επιτάχυνσης είναι σταθερά στο μηδέν, καθώς το όχημα απλά επιταχύνει σε μία ευθεία. Η διαμήκης επιτάχυνση αυξάνεται απότομα, μέχρι που φτάνει

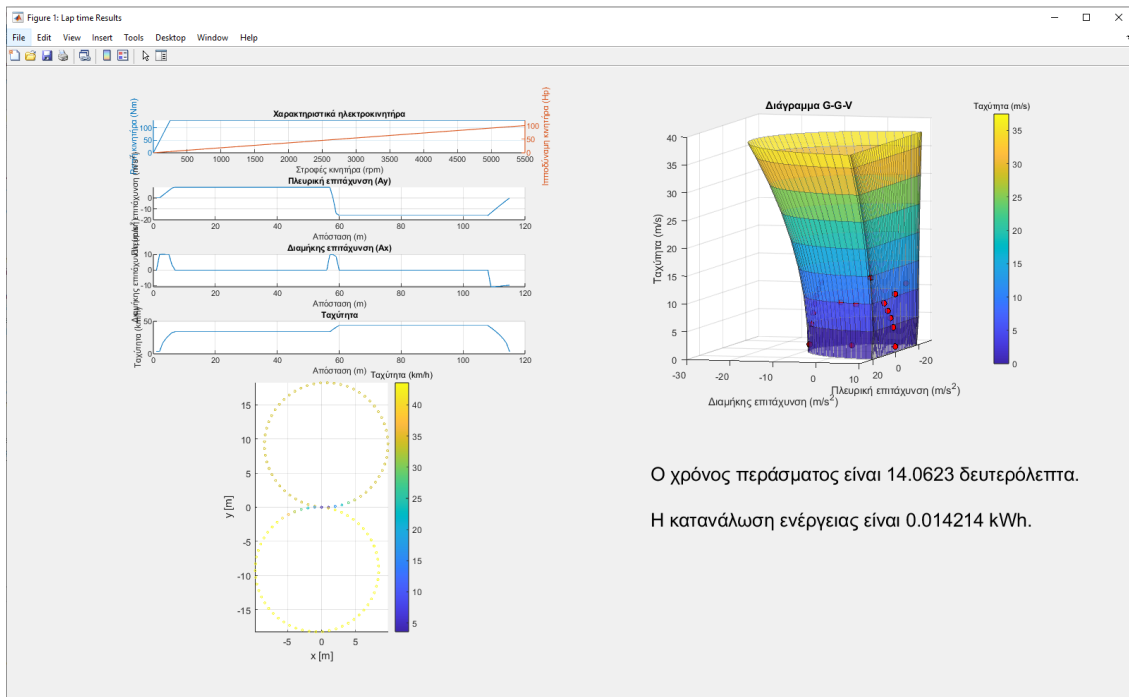
το όριο της και παραμένει σταθερή γύρω στα 8m/s^2 . Όταν το όχημα φτάσει στα 75 μέτρα, δηλαδή την γραμμή τερματισμού, φρενάρει απότομα όποτε η επιτάχυνση μειώνεται δραστικά στα -23m/s^2 . Η επιβράδυνση αυτή δεν είναι σταθερή και μειώνεται με γρήγορο ρυθμό. Η ταχύτητα φτάνει στα 130 χιλιόμετρα την ώρα στο 75° μέτρο, πριν μειωθεί δραστικά λόγω του φρεναρίσματος. Ο κώδικας εμφανίζει μια γραμμή 75 μέτρων στον χάρτη, ο οποίος δεν έχει πάχος οπότε ο χάρτης δεν εμφανίζει τίποτα. Το διάγραμμα G-G-V απεικονίζει τις δυνατότητες του οχήματος. Οι κόκκινες τελείες είναι οι τιμές που επιτεύχθηκαν κατά την προσομοίωση. Οι δύο άξονες στο επίπεδο είναι η πλευρική και διαμήκης επιτάχυνση σε m/s^2 , ενώ ο κάθετος άξονας είναι η ταχύτητα του οχήματος, σε m/s . Ο χρόνος των 4,535 δευτερολέπτων θα σχολιαστεί σε αργότερο κεφάλαιο. Η κατανάλωση των 20kWh είναι υπερβολικά υψηλή, αλλά είναι μόνο για 4 δευτερόλεπτα και εμφανίζεται επειδή το όχημα πηγαίνει με 100% γκάζι καθ' όλη τη διάρκεια που καταγράφεται η ενέργεια.



Εικόνα 49.2: Αποτελέσματα Acceleration

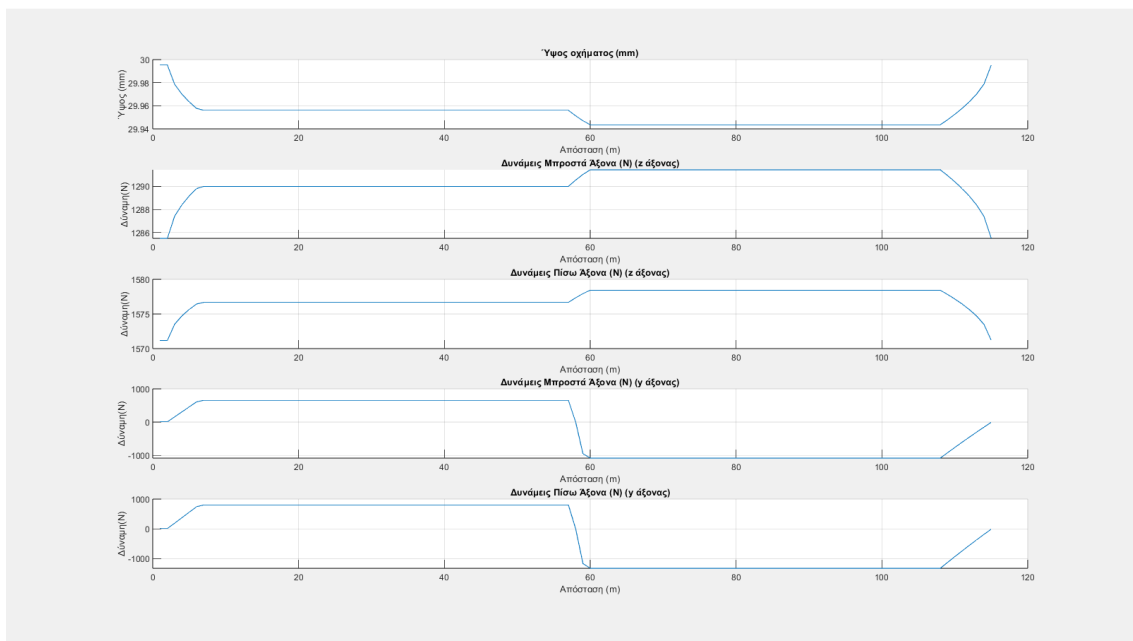
Για τα δυναμικά αποτελέσματα, υπάρχει μείωση της απόστασης από το έδαφος κατά

την επιτάχυνση, κάτι που είναι αναμενόμενο. Επίσης παρατηρείται ανάλογη αύξηση της δύναμης στον άξονα z μπροστά και πίσω, καθώς αυξάνεται η αρνητική άντωση. Οι δυνάμεις στον άξονα y είναι μηδενικές και παραμένουν έτσι, καθώς το όχημα κινείται σε ευθεία.



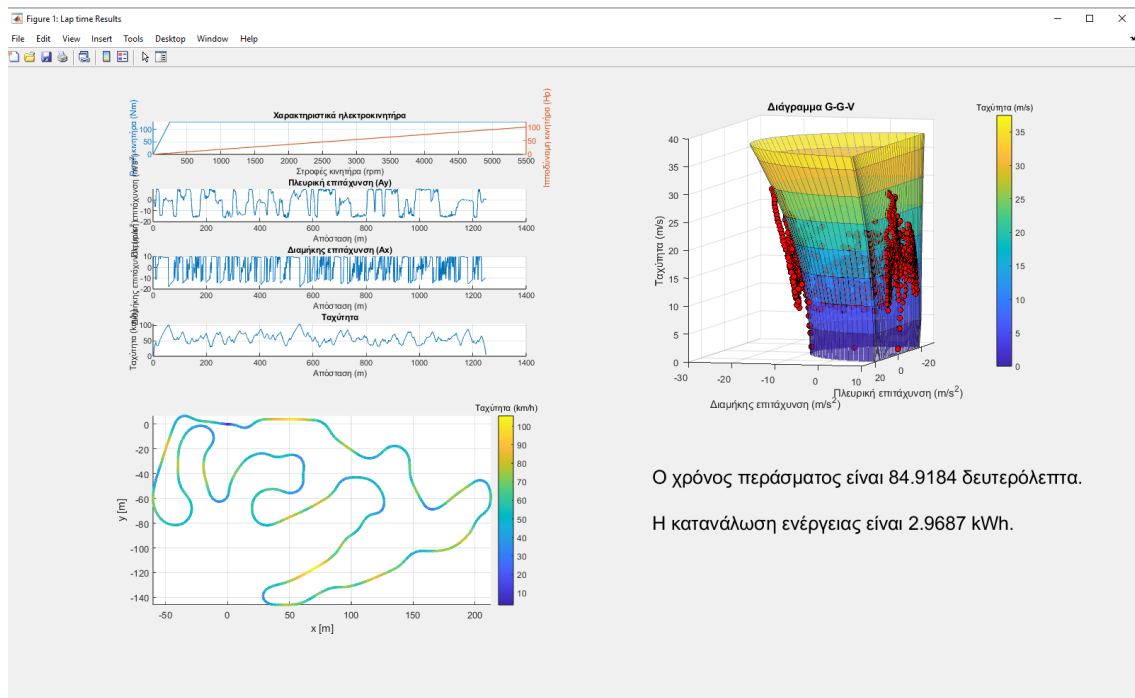
Εικόνα 50.1: Αποτελέσματα Skidpad

Στο Skidpad παρατηρείται ότι το όχημα διανύει τον δεύτερο κύκλο με μεγαλύτερη ταχύτητα από τον πρώτο. Η πλευρική επιτάχυνση αλλάζει πρόσημο όταν αλλάζει και η κατεύθυνση της στροφής, ενώ η διαμήκης μένει σταθερή καθώς μένει σταθερή η ταχύτητα της στροφής. Ο πρώτος κύκλος θα διανυθεί με ταχύτητα 33,9 χιλιόμετρα την ώρα σε 6,088 δευτερόλεπτα. Αφού το όχημα διατηρεί ταχύτητα 43 χιλιόμετρα την ώρα κατά τον δεύτερο κύκλο, και ο δεύτερος κύκλος έχει περίμετρο 57,33 μέτρα, υπολογίζεται ότι θα τον διανύσει σε 4,744 δευτερόλεπτα. Όπως και στο προηγούμενο αποτέλεσμα, με κόκκινες κουκίδες εμφανίζονται οι τιμές στο διάγραμμα G-G-V, που επιτεύχθηκαν κατά την προσομοίωση.



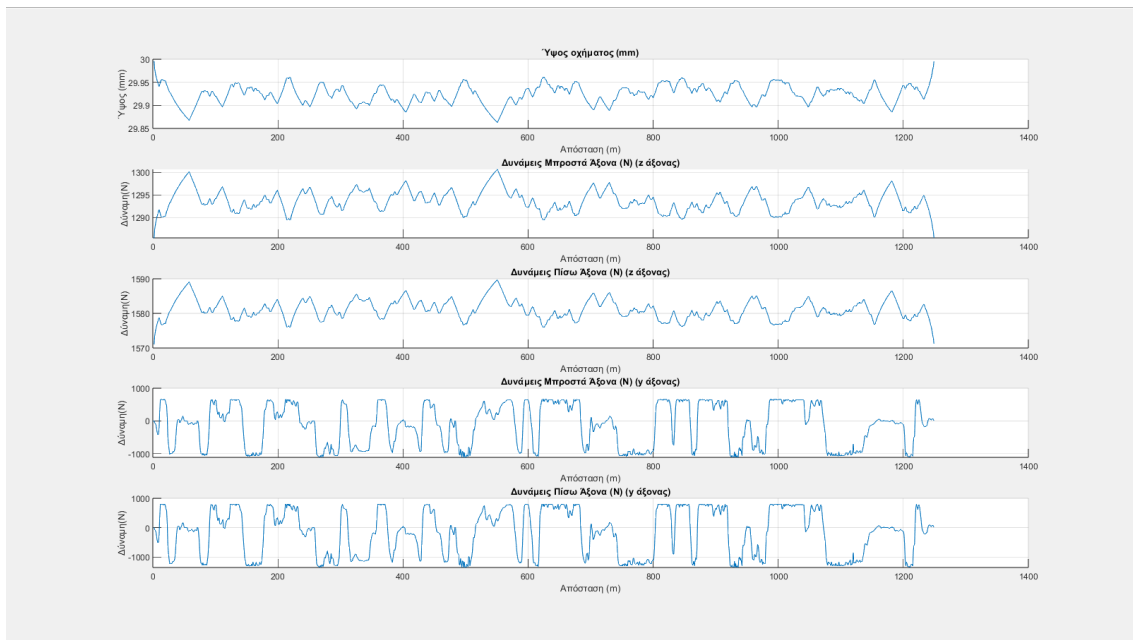
Εικόνα 50.2: Αποτελέσματα Skidpad

Υπάρχει μία μείωση στο ύψος του οχήματος καθώς ανεβαίνει η ταχύτητα. Ανάλογη πορεία με την ταχύτητα ακολουθούν και οι φορτίσεις στον άξονα z. Στον άξονα y, οι δυνάμεις εναλλάσσονται ανάλογα με την κατεύθυνση της στροφής (θετικό πρόσημο για τις δεξιές και αρνητικό για τις αριστερές). Καθώς το όχημα διανύει με μεγαλύτερη ταχύτητα την δεύτερη στροφή, το μέτρο της δύναμης είναι μεγαλύτερο. Στο Autocross φαίνονται οι πραγματικές δυνατότητες του αλγορίθμου:



Εικόνα 51.1: Αποτελέσματα Autocross

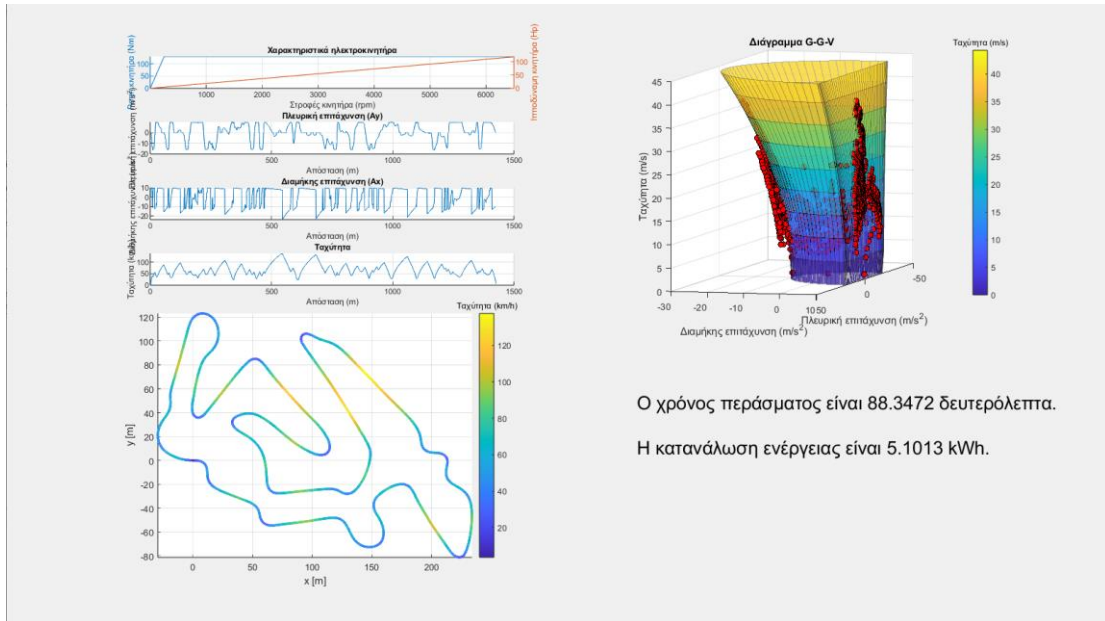
Οι πλευρικές και διαμήκης επιταχύνσεις ταιριάζουν σε μεγάλο βαθμό με την αναμενόμενη συμπεριφορά του οχήματος στην πίστα. Ο χάρτης της πίστας, όπως και στο Skidpad, είναι σκιαγραφημένος με την ταχύτητα του οχήματος στο αντίστοιχο σημείο. Το διάγραμμα G-G-V έχει εκατοντάδες σημεία με κόκκινες κουκίδες, που περιγράφουν τις επιταχύνσεις του οχήματος ανάλογα με την ταχύτητα του. Οι κουκίδες εφάπτονται της πολύχρωμης καμπύλης, που σημαίνει ότι το όχημα αποδίδει στα όρια που του θέσαμε. Ο χρόνος είναι στα 84,918 δευτερόλεπτα. Η κατανάλωση, ενώ είναι αντιπροσωπευτική πραγματικών οχημάτων Formula Student, δεν καταμετράται στο Efficiency, οπότε δε λαμβάνεται ακόμα υπόψιν.



Εικόνα 51.2: Αποτελέσματα Autocross

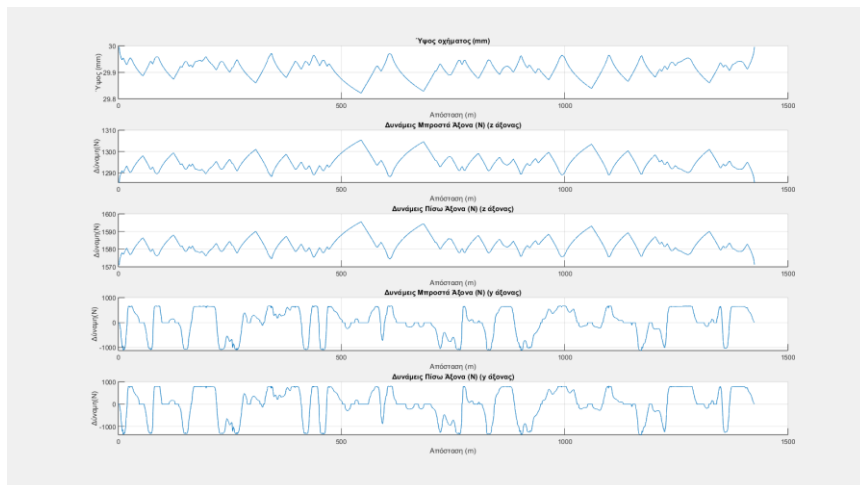
Η απόσταση του οχήματος αυξομειώνεται ανάλογα με την ταχύτητα, με ελάχιστη τιμή τα 29,86 χιλιοστά από το έδαφος. Οι κάθετες δυνάμεις φτάνουν τα 1300N στον μπροστά άξονα και τα 1590N στον πίσω άξονα. Οι πλευρικές αλλάζουν πρόσημο ανάλογα με την κατεύθυνση της στροφής (θετικό πρόσημο για τις δεξιές και αρνητικό για τις αριστερές) με μέγιστο μέτρο τα -1104N στον μπροστά και -1335N στον πίσω άξονα (η πίστα έχει πιο απαιτητικές αριστερές στροφές).

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα του Endurance:



Εικόνα 52.1: Αποτελέσματα Endurance

Ο χρόνος των 88,347 δευτερολέπτων, σε 18 γύρους μεταφράζεται σε χρόνο 29 λεπτών και 26 δευτερολέπτων (1591, δευτερόλεπτα). Η κατανάλωση 5,1kWh στο Endurance συναντάται και στις πηγές της εργασίας, και είναι αντιπροσωπευτική των ταχυτήτων που αναπτύσσονται στον γύρο. Κατά την διάρκεια του γύρου, το όχημα φτάνει τα 136 χιλιόμετρα την ώρα, όπως φαίνεται και από το σχετικό διάγραμμα.



Εικόνα 52.2: Αποτελέσματα Endurance

Σύμφωνα με τα διαγράμματα, το ελάχιστο ύψος του οχήματος είναι στα 29,82 χιλιοστά. Η μεγαλύτερη κάθετη δύναμη στον μπροστά άξονα είναι στα 1304N, ενώ

στον πίσω στα 1595N. Η μεγαλύτερη πλευρική δύναμη στον μπροστά άξονα είναι στα -1100N, ενώ στον πίσω στα -1367N, στο 902° μέτρο της πίστας.

6.2 Σύγκριση των αποτελεσμάτων με αποτελέσματα διαγωνισμών

Σύμφωνα με τα αποτελέσματα του FSG για τον διαγωνισμό του 2023, τα οποία είναι αναρτημένα στην ιστοσελίδα του διαγωνισμού, το όχημα θα έβγαινε στις παρακάτω θέσεις:

1. 32° στο Acceleration, με χρόνο 4,535 δευτερόλεπτα με έπαθλο 11,12 πόντους
2. 28° στο Skidpad, με μέσο χρόνο 4,606 δευτερόλεπτα με το έπαθλο 18,91 πόντων

Οι πίστες Autocross και Endurance είναι του 2012, οπότε θα χρειαστεί να υπολογιστούν οι βαθμολογίες με τους χρόνους από εκείνη την χρονιά, χρησιμοποιώντας το τωρινό σύστημα υπολογισμού βαθμών:

3. 8° στο Autocross με χρόνο 84,918 δευτερόλεπτα και έπαθλο 76,05 πόντους
4. 6° στο Endurance με χρόνο 1591,866 δευτερόλεπτα και έπαθλο 170,76 πόντους
5. 3° στο Efficiency με κατανάλωση 5,1013kWh και έπαθλο 39,75 πόντους.

Καθώς το 2012 οι μέγιστοι πόντοι για το Efficiency ήταν 100 αντί για 75, το έπαθλο των 39,75 πόντων έχει υποστεί πολλαπλασιασμό με το 0,75, από 49,68 που θα ήταν κανονικά. Συνολικά από τα δυναμικά αγωνίσματα το όχημα θα βραβευόταν 324,85 πόντους. Χωρίς τους βαθμούς από τα στατικά αγωνίσματα, το όχημα θα ήταν ήδη στην 21^η θέση του διαγωνισμού του 2023, στην κατηγορία των ηλεκτρικών οχημάτων.

Κεφάλαιο 7

Συμπεράσματα και περιορισμοί

Κατά την διάρκεια της εργασίας, αναπτύχθηκε ένα μοντέλο οχήματος και πίστας με τη χρήση του MATLAB και διενεργήθηκαν οι κατάλληλες προσομοιώσεις που επιτυγχάνουν τον στόχο που αναφέρθηκε. Έγινε αναφορά στα δεδομένα και τους τύπους που διέπουν την συμπεριφορά του οχήματος και στη συνέχεια επεξηγήθηκε ο κώδικας που χρησιμοποιήθηκε για την περάτωση της εργασίας. Εν κατακλείδι, η προσομοίωση του οχήματος λειτουργεί χωρίς πρόβλημα και αποδίδει αποτελέσματα που αντικατοπτρίζουν την πραγματικότητα, χωρίς την ανάγκη για δημιουργία οποιουδήποτε φυσικού μοντέλου, διαδικασία εξίσου χρονοβόρα όσο και δαπανοφόρα.

Όσον αφορά στα αποτελέσματα της προσομοίωσης, οι χρόνοι στα δυναμικά αγωνίσματα είναι ανταγωνιστικοί με άλλες ομάδες της κατηγορίας των ηλεκτροκίνητων, με αποτέλεσμα την συγκέντρωση πολλών πόντων. Τα δυναμικά αποτελέσματα δεν δείχνουν υπερβολική μεταβολή στο ύψος του οχήματος κατά την οδήγηση. Οι μέγιστες δυνάμεις που αναπτύσσονται στο όχημα κατά την οδήγηση είναι (σε απόλυτα μεγέθη):

	Άξονας z	Άξονας y
Μπροστά Άξονας	1304 N	1100 N
Πίσω Άξονας	1590 N	1367 N

Ο παραπάνω πίνακας μπορεί να χρησιμοποιηθεί για την βελτιστοποίηση της αεροδυναμικής του οχήματος, σχεδίασης της γεωμετρίας αναρτήσεως και επιλογής ελατηρίων του οχήματος. Δεν πρέπει να παραληφθεί η σημαντικότητα των ελαστικών, καθώς τα αποτελέσματα της προσομοίωσης μπορούν να χρησιμοποιηθούν σε συνδυασμό με λογισμικό επεξεργασίας δεδομένων των ελαστικών (tire data treatment), γεγονός που είναι μεγάλο πλεονέκτημα στην κατανόηση της συμπεριφοράς του οχήματος.

Βασικός περιορισμός της εργασίας είναι η απλότητα του μοντέλου: Οι δυνατότητες του μοντέλου είναι περιορισμένες και χρειάζεται επέκταση του μοντέλου (προσθήκη παραπάνω μεταβλητών, μοντελοποίηση συστήματος διεύθυνσης και πέδησης φρένων). Συμπληρωματικά, όπως αρμόζει σε κάθε προσομοίωση, η επαλήθευση των αποτελεσμάτων είτε με ηλεκτρονικά είτε με φυσικά μέσα είναι απαραίτητο μελλοντικό βήμα.

Κατάλογος αναφορών

[1] FSG: Concept –

<https://www.formulastudent.de/about/concept/>

[2] FSG: Rules & Documents –

<https://www.formulastudent.de/fsg/rules/>

[3] Analysis of the Performance and Overall Efficiency of a Formula Student Electric Car, L. Martolucci, M. Giannini –

<https://www.scirp.org/journal/paperinformation?paperid=114951>

[4] Εισαγωγή στην MATLAB, Γ. Γεωργίου, Χ. Ξενοφώντος ISBN 978-9963-644-57-5, σελίδα 1 –

<https://web.archive.org/web/20101214080310/http://www2.ucy.ac.cy/~georgios/bookfiles/MATLABbook.pdf>

[5] Formula Student East – Flickr –

<https://www.flickr.com/photos/fseast/53131781203/in/album-72177720310621000/>

<https://www.flickr.com/photos/fseast/53131770598/in/album-72177720310638454/>

<https://www.flickr.com/photos/fseast/53130692907/in/album-72177720310638454/>

[6] Ηλεκτροκινητήρας Emrax 228, φυλλάδιο πληροφοριών –

<https://emrax.com/e-motors/emrax-228/#1482059435741-232ed37a-accc>

[7] Design & Development of an Aerodynamic Package for an FSAE Race Car, I.

Οξύζογλου (2017), σελ. 116 –

https://www.researchgate.net/publication/320556659_Design_Development_of_an_Aerodynamic_Package_for_an_FSAE_Race_Car

[8] Selection of Optimal Tire and Design Optimisation of Steering System for a

Formula Student Race Car through Tire Data Treatment, V. S. Swamy, (2020), σελ. 6 –

<https://iopscience.iop.org/article/10.1088/1742-6596/1478/1/012032/pdf>

[9] Fundamentals of Vehicle Dynamics, Thomas D. Gillespie, σελ. 117 – ISBN 1-56091-199-9

[10] Design of an Electric Drive Transmission for a Formula Student Race Car, White G., Doyle D., Cunningham G. (2019) σελ. 4 πίνακας 4 –

https://pure.qub.ac.uk/files/171248998/DO_an_Electric_Drive_Transmission_for_a_FS_Race_Car_FINAL_MANUSCRIPT_SUBMISSION.pdf

[11] Michael Halkiopoulos (2020). OpenLAP Lap Time Simulator –

<https://github.com/mc12027/OpenLAP-Lap-Time-Simulator/releases/tag/V01.00>),
GitHub.

[12] Performance Vehicle Dynamics, J. Balkwill, σελ. 82 εξίσωση 4.2

[13] Fundamentals of Vehicle Dynamics T.D. Gillespie, σελ.103 εξίσωση 4.7

[14] Tire and Vehicle Dynamics, H. Pacejka, σελ. 5

[15] Motorcycle Handling and Chassis Design, T. Foale σελ. 2-29

[16] Race Car Vehicle Dynamics, Milliken, σελ. 596

[17] Αποτελέσματα FSG 2023 -

<https://www.formulastudent.de/fsg/results/2023/#c5177>

[18] FSG points calculator –

<https://fsg-scores.philenius.de/>