



Α.Τ.Ε.Ι ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΕΛΕΓΧΟΥ ΠΤΗΣΗΣ
ΜΗ-ΕΠΑΝΔΡΩΜΕΝΟΥ ΑΕΡΟΣΚΑΦΟΥΣ

ΠΛΑΤΗΣ ΑΓΓΕΛΟΣ Α.Μ: 37996



Επιβλέποντες Καθηγητές: ΑΛΑΦΟΔΗΜΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΙΟΣ

ΠΕΙΡΑΙΑΣ - 2013

Copyright © Πλατής Άγγελος, 2013

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσεως, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν την χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται στην εργασία αυτήν εκφράζουν την άποψη του συγγραφέα και δεν πρέπει να θεωρηθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πειραιά.

Για οποιαδήποτε χρήση του εμπειριεχόμενου υλικού επικοινωνήστε με τον συγγραφέα στην διεύθυνση: agplatis@gmail.com

ΠΕΡΙΕΧΟΜΕΝΑ:

Κεφάλαιο 1

- Εισαγωγή

1.1	Η Χρησιμότητα των UAV και προβλήματα υλοποίησης.....	9
1.2	Περίληψη πτυχιακής.....	11

Κεφάλαιο 2

- Θεωρητική προσέγγιση

2.1	Εξισώσεις ενός UAV συστήματος.....	13
2.1.1	Μαθηματικό μοντέλο ενός quadcopter.....	13
2.1.2	Εξισώσεις Νεύτωνα-Euler.....	16
2.1.3	Εξισώσεις Euler-Lagrange.....	18
2.1.4	Αεροδυναμικές επιδράσεις.....	20
2.2	Εξισώσεις ελεγκτών.....	21
2.2.1	Ο ελεγκτής PID.....	21
2.2.2	Ελεγκτής ασαφούς λογικής.....	29
2.2.3	Το φίλτρο Kalman.....	32
2.3	Εξισώσεις αισθητηρίων.....	37
2.3.1	Το γυροσκόπιο.....	37
2.3.2	Το επιταχυνσιόμετρο.....	39

Κεφάλαιο 3

- Μοντελοποίηση συστημάτων στο MATLAB – Simulink

3.1	Πρώτο μοντέλο quadcopter.....	44
3.1.1	Παρουσίαση του μοντέλου.....	44

3.1.2 Ανάλυση του μοντέλου.....	47
3.1.3 Αποτελέσματα προσομοίωσης.....	53
3.2 Δεύτερο μοντέλο quadcopter.....	59
3.2.1 Παρουσίαση του μοντέλου.....	59
3.2.2 Διατάξεις των γενικών μας εισόδων.....	61
3.2.3 Μοντελοποίηση των DC μοτέρ.....	76
3.2.4 Μοντελοποίηση του quadcopter.....	82
3.2.5 Μετρήσεις μέσω αισθητηρίου.....	86
3.2.6 Φιλτράρισμα των μετρήσεων.....	91
3.2.7 Αποτελέσματα προσομοίωσης.....	98

Κεφάλαιο 4

- Γενικές παρατηρήσεις και συμπεράσματα

4.1 Παρατηρήσεις κατά την διεξαγωγή των πειραμάτων.....	104
4.1.1 Ανάπτυξη εφεδρικού μοντέλου για τα DC μοτέρ.....	104
4.1.2 Διαφορές πρώτου και δεύτερου μοντέλου.....	110
4.2 Συμπεράσματα.....	112
4.3 Μελλοντική εργασία.....	114

<u>Βιβλιογραφία.....</u>	117
--------------------------	-----

<u>Συνοπτικά στοιχεία βιβλιογραφίας.....</u>	123
--	-----

- Μοντελοποίηση.....123
- Έλεγχος.....127
- Πλοήγηση.....131

Πίνακας Εικόνων

Εικόνα 1: Παράδειγμα πιθανής αποστολής ενός quadcopter.....	10
Εικόνα 2: Η αρχικοποίηση και η δομή ενός quadcopter.....	13
Εικόνα 3: Ο ελεγκτής PID σε σύστημα.....	21
Εικόνα 4: Απόκριση συστήματος όταν $K_p = K_{op}$ και μέτρηση της περιόδου.....	24
Εικόνα 5: Διάγραμμα BODE συστήματος και εύρεση του περιθωρίου κέρδους και φάσης.....	25
Εικόνα 6: Βηματική απόκριση συστήματος ανοικτού βρόγχου.....	27
Εικόνα 7: Ακολουθία λειτουργίας ενός απλού ελεγκτή ασαφούς λογικής.....	30
Εικόνα 8: Η λειτουργία ενός φίλτρου.....	33
Εικόνα 9: Απόκριση ενός φίλτρου Kalman (πράσινα στίγματα) μετά τις προβλέψεις (μαύρα στίγματα).....	36
Εικόνα 10: Δομή ενός απλού γυροσκοπίου.....	37
Εικόνα 11: Δομή ενός επιταχυνσιόμετρου.....	39
Εικόνα 12: Το επιταχυνσιόμετρο κατά την κίνηση του προς τα αριστερά.....	40
Εικόνα 13: Το επιταχυνσιόμετρο και η αντίδρασή του όταν το ακουμπάμε στο έδαφος.....	40
Εικόνα 14: Το επιταχυνσιόμετρο με άσκηση δύναμης σε δύο από τους άξονές του.....	41
Εικόνα 15: Άξονες ενός επιταχυνσιόμετρου.....	42
Εικόνα 16: Το πρώτο μας μοντέλο.....	45
Εικόνα 17: Ρυθμίσεις για το περιβάλλον εικονικής πραγματικότητας.....	46
Εικόνα 18: Σχέδιο ενός quadcopter στο AutoCAD.....	48
Εικόνα 19: Περιβάλλον εικονικής πραγματικότητας με το σχεδιασμένο μοντέλο μας.....	48
Εικόνα 20: Το block ελέγχου των γωνιών και της θέσης του μοντέλου μας.....	49
Εικόνα 21: Η διαδικασία που εκτελείται εσωτερικά του block "Position_control".....	50
Εικόνα 22: Η διαδικασία που επιτελείται εσωτερικά του block "Euler_Angles_Control".....	50
Εικόνα 23: Η διαδικασία ελέγχου της γωνίας Roll και της γωνίας Pitch.....	52
Εικόνα 24: Εικόνα κατά την προσομοίωση του μοντέλου μας.....	53
Εικόνα 25: Κίνηση του μοντέλου μας, με κλίση προς τα μπροστά.....	55
Εικόνα 26: Κίνηση του μοντέλου μας, με κλίση προς τα πίσω.....	55
Εικόνα 27: Κίνηση του μοντέλου μας, με κλίση προς τα αριστερά.....	56
Εικόνα 28: Κίνηση του μοντέλου μας, με κλίση προς τα δεξιά.....	56
Εικόνα 29: Γραφήματα συμπεριφοράς του μοντέλου μας.....	57
Εικόνα 30: Quadcopter στο MATLAB.....	60
Εικόνα 31: Εσωτερική διάταξη ελέγχου κίνησης του quadcopter χειροκίνητα.....	61
Εικόνα 32: Εσωτερικό των block "x_υπολογισμός", "y_υπολογισμός".....	62
Εικόνα 33: Εσωτερικό του block "x/y_υπολογισμός".....	63
Εικόνα 34: Κεντρικό παράθυρο του FIS Editor.....	66

Εικόνα 35: Διαμόρφωση εισόδων – εξόδων για την εφαρμογή μας.....	67
Εικόνα 36: Δήλωση των κανόνων ελέγχου για τον ελεγκτή ασαφούς λογικής χειριστήριου.....	68
Εικόνα 37: Δήλωση των κανόνων ελέγχου για τον ελεγκτή ασαφούς λογικής αυτόματης πορείας.....	71
Εικόνα 38: Γραφικές για τον καθορισμό περιπτώσεων (πάνω) και των αποτελεσμάτων κάθε μεθόδου defuzzification (κάτω).....	75
Εικόνα 39: Σύστημα ελέγχου με PID για καθένα από τα 4 μοτέρ.....	76
Εικόνα 40: Διαφορικές εξισώσεις ενός DC μοτέρ.....	77
Εικόνα 41: Μοντελοποίηση των DC μοτέρ από τις διαφορικές του εξισώσεις.....	78
Εικόνα 42: Απόκριση του μοτέρ μας χωρίς εφαρμογή ελέγχου.....	79
Εικόνα 43: Απόκριση του μοτέρ μας με αναλογικό έλεγκτή $C(s)=100$	80
Εικόνα 44: Επιθυμητή απόκριση ενός DC μοτέρ με ελεγκτή PID.....	81
Εικόνα 45: Μοντελοποίηση ενός quadcopter μέσα από τις διαφορικές εξισώσεις του.....	82
Εικόνα 46: Το εσωτερικό του block Έλεγχος_εισόδων.....	83
Εικόνα 47: Το εσωτερικό του block Υπολογισμοί.....	83
Εικόνα 48: Το εσωτερικό του block θ	84
Εικόνα 49: Το εσωτερικό του block ϕ	84
Εικόνα 50: Το εσωτερικό του block g	85
Εικόνα 51: Το αισθητήριο MinIMU-9.....	86
Εικόνα 52: Ο μικροελεξεργαστής Arduino.....	86
Εικόνα 53: Μετρήσεις των γωνιών Euler από το αισθητήριο.....	90
Εικόνα 54: Το εσωτερικό του block Αισθητήρια καθώς επίσης και των block, Γυροσκόπιο – Επιταχυνσιόμετρο.....	91
Εικόνα 55: Μετρήσεις επιταχυνσιόμετρου.....	92
Εικόνα 56: Το εσωτερικό του block “Φίλτρο_Kalman”.....	93
Εικόνα 57: Ρυθμίσεις του φίλτρου kalman.....	93
Εικόνα 58: Το εσωτερικό του φίλτρου kalman.....	97
Εικόνα 59: Απόκριση θέσης για την περίπτωση 1.....	99
Εικόνα 60: Απόκριση θέσης για την περίπτωση 2.....	99
Εικόνα 61: Απόκριση θέσης για την περίπτωση 3.....	99
Εικόνα 62: Απόκριση θέσης για την περίπτωση 4.....	99
Εικόνα 63: Μονοπάτι εισόδου.....	100
Εικόνα 64: Απόκριση θέσης του μοντέλου μας σε αυτόματη λειτουργία.....	101
Εικόνα 65: Αποτελέσματα μετρήσεων των γωνιών μας, πριν (με άσπρο) και μετά (με μπλε) το φίλτρο kalman.....	102
Εικόνα 66: Απόκριση του μοτέρ μας χωρίς έλεγχο.....	106
Εικόνα 67: Απόκριση του μοτέρ μας με αναλογικό ελεγκτή $C(s)=100$	107
Εικόνα 68: Απόκριση του μοτέρ μας με εφαρμογή ελεγκτή PID.....	108
Εικόνα 69: Σχηματισμός του “πλέγματος” των δορυφόρων γύρω από την Γη.....	115
Εικόνα 70: Μοντέλο quadcopter με ενσωματωμένη κάμερα.....	116

Κεφάλαιο 1: Εισαγωγή

1.1 Η Χρησιμότητα των UAV και προβλήματα υλοποίησης

Ένα UAV (Unmanned Aerial Vehicle), είναι ένα αεροσκάφος χωρίς να υπάρχει άμεσος χειρισμός αυτού από κάποιον πιλότο. Οι πτήσεις τους πραγματοποιούνται είτε αυτόνομα, ελεγχόμενες μέσω υπολογιστών, οι οποίοι υπάρχουν μέσα στο αεροσκάφος, είτε μέσω ενός χειριστηρίου, το οποίο θα βρίσκεται έξω από το αεροσκάφος αυτό ή μέσα σε ένα άλλο συνοδευτικό αεροσκάφος και ο έλεγχος του θα γίνεται μέσω του πιλότου.

Υπάρχει μια τεράστια ποικιλία από UAV, διαφορετικού σχήματος, μεγέθους, διαμόρφωσης και χαρακτηριστικών. Ιστορικά, τα UAV ελέγχονταν κατά την πτήση τους από πιλότους μέσω ενός χειριστηρίου, αλλά ο αυτόνομος έλεγχος, με το πέρασμα των χρόνων, εφαρμόζεται ολοένα και περισσότερο.

Τα UAV έχουν αναπτυχθεί κατά κύριο λόγο για στρατιωτικές εφαρμογές, όμως υπάρχει μια σταδιακή αύξηση χρησιμοποίησής τους και σε αστικές εφαρμογές, όπως σε πυροσβεστικές αποστολές και σε χώρους μη-στρατιωτικής ασφάλειας, π.χ η επιτήρηση των αγωγών μιας εγκατάστασης. Τα UAV συχνά προτιμούνται για αποστολές οι οποίες είναι αδύνατο να υλοποιηθούν από επανδρωμένα αεροσκάφη. Τέλος, τα UAV, μπορούν να χρησιμοποιηθούν σε αποστολές έρευνας και διάσωσης, βοηθώντας καταλυτικά στο να βρεθούν άνθρωποι οι οποίοι έχουν χαθεί, ή έχουν παγιδευτεί μέσα στα συντρίμια κτιρίων, ή επιπλέουν στην θάλασσα, ψάχνοντας σε σημεία που για τον άνθρωπο είναι επικίνδυνα και δύσκολα προσβάσιμα.

Πέρα όμως από τα πολλά πλεονεκτήματα που μας προσφέρουν τα UAV, σαν κατασκευές, στην υλοποίηση τους πρέπει να ξεπεράσουμε κάποια προβλήματα. Το κυριότερο, από τα προβλήματά μας, είναι πως όλα τα αισθητήρια παράγουν “θορύβους” με αποτέλεσμα οι μετρήσεις τους να μην είναι τόσο ακριβείς.

Ένα εξίσου σημαντικό πρόβλημα, κατά την υλοποίηση μιας τέτοιας κατασκευής είναι πως βάζοντας ένα UAV σε έναν άγνωστο χώρο, αυτό δεν θα έχει κάποια προηγούμενη πληροφορία. Για να μπορέσει λοιπόν να φτάσει σε ένα απομακρυσμένο στόχο μέσα στον χώρο αυτό, πρέπει να αποκτήσει γνώση του χώρου και να την αποθηκεύσει με τέτοιο τρόπο, που να είναι εφικτή αργότερα η αξιοποίησή της για τον σχεδιασμό ενός βέλτιστου μονοπατιού, προς τον τελικό στόχο.



Εικόνα 1: Παράδειγμα πιθανής αποστολής ενός quadcopter

1.2 Περίληψη πτυχιακής

Κατά την διεξαγωγή της πτυχιακής αυτής έγινε χρήση βασικών γνώσεων που αφορούν στον έλεγχο ενός μοντέλου, όπως είναι ένας PID ελεγκτής. Επίσης, χρησιμοποιήθηκε η μέθοδος μοντελοποίησης, για την απόδοση των μοντέλων των μοτέρ μας από τις διαφορικές του εξισώσεις καθώς επίσης και μερική μοντελοποίηση του quadcopter μας. Εκτός από την μοντελοποίηση των μοτέρ μας, έγινε και η δημιουργία της συνάρτησης μεταφοράς του αναπτύσσοντας τις διαφορικές του εξισώσεις. Τέλος, χρησιμοποιήθηκαν προγράμματα που χρησιμοποιήσαμε κατά την διάρκεια των εξαμήνων όπως το AutoCAD και το MATLAB.

Πέρα όμως από αυτά τα βασικά στοιχεία που χρησιμοποιηθήκαν στην εργασία αυτή, έγινε μελέτη και εφαρμογή και άλλων στοιχείων. Έγινε παρουσίαση θεωρίας σε συνεργασία με ανάπτυξη κώδικα στο MATLAB για την καλύτερη κατανόηση μεθόδων ελέγχου όπως ο ελεγκτής ασαφούς λογικής, και το φίλτρο Kalman. Πρόκειται για δυο αρκετά απαιτητικές μεθόδους ελέγχου οι οποίες χρειάστηκαν αρκετή μελέτη και παρατήρηση, ώστε να φτάσουμε σε ικανοποιητικό επίπεδο κατανόησης της λειτουργίας τους, για να μπορέσουμε να υποστηρίξουμε την θεωρία τους με ένα παραδειγμα στο MATLAB.

Στο δεύτερο κεφάλαιο, θα γίνει μια παρουσίαση του θεωρητικού μοντέλου, και συγκεκριμένα των εξισώσεων που απαρτίζουν ένα τέτοιο UAV σύστημα. Επίσης, θα γίνει παρουσίαση των εξισώσεων των ελεγκτών και των αισθητηρίων που θα μελετήσουμε. Στο τρίτο κεφάλαιο, θα γίνει μια μελέτη όλων των σύγχρονων μεθόδων ελέγχου, σε συνδυασμό με σύγχρονα αισθητήρια, ενός UAV. Για να ενισχυθεί και ολοκληρωθεί η μελέτη αυτή, θα γίνει ανάπτυξη δύο μοντέλων στο προγραμματιστικό περιβάλλον του MATLAB και συγκεκριμένα στο Simulink, η οποία θα μας βοηθήσει να κατανοήσουμε καλύτερα την κάθε μέθοδο στην πράξη, καθώς επίσης και την

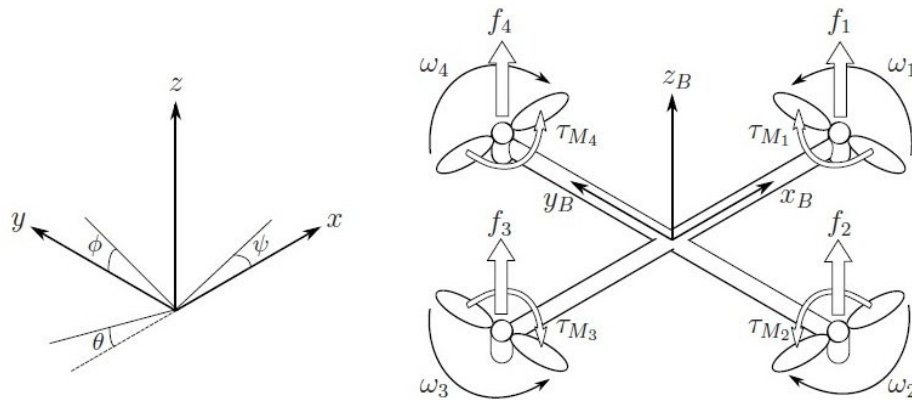
λειτουργία των αισθητηρίων, και κατ'επέκταση να επιλέξουμε την καταλληλότερη για τα δικά μας μοντέλα. Το πρώτο μας μοντέλο, δημιουργήθηκε παράλληλα με την μελέτη της θεωρίας του quadcopter προκειμένου να παρατηρήσουμε την συμπεριφορά του, ενώ αντίθετα το δεύτερο μοντέλο που αναπτύχθηκε βασίζεται στις διαφορικές εξισώσεις ενός τέτοιου συστήματος και τα περισσότερα κομμάτια του είναι δικής μας δημιουργίας. Στο τέταρτο και τελευταίο κεφάλαιο της εργασίας αυτής, θα γίνει ανάπτυξη των παρατηρήσεων-αστοχιών που υπήρξαν κατά την μοντελοποίηση. Στην συνέχεια, θα παρουσιαστούν τα αποτελέσματα που βγάλαμε από τα διάφορα μοντέλα που αναπτύχθηκαν. Τέλος, θα γίνει μια παρουσίαση των μελλοντικών μου σχεδίων, όσο αφορά την εξέλιξη και την βελτιστοποίηση της εργασίας αυτής.

Κεφάλαιο 2: Θεωρητική προσέγγιση

2.1 Εξισώσεις ενός UAV συστήματος

2.1.1 Μαθηματικό μοντέλο ενός quadcopter

Η δομή ενός quadcopter παρουσιάζεται στο παρακάτω σχέδιο, συμπεριλαμβάνοντας τις γωνιακές ταχύτητες, ροπές και δυνάμεις, οι οποίες δημιουργήθηκαν από τους 4 έλικες (με αρίθμηση από το 1 έως το 4).



Εικόνα 2: Η αρχικοποίηση και η δομή ενός quadcopter

Η γραμμική θέση ενός μη-επανδρωμένου αεροσκάφους προσδιορίζεται στον αρχικό σχηματισμό των αξόνων x , y , z με ξ . Η συμπεριφορά του, δηλαδή η γωνιακή του ταχύτητα, προσδιορίζεται στον αρχικό σχηματισμό, με τρεις Euler γωνίες η . Η γωνία Pitch θ προσδιορίζει την περιστροφή ενός μη-επανδρωμένου αεροσκάφους γύρω από τον άξονα y . Η γωνία Roll φ προσδιορίζει την περιστροφή ενός μη-επανδρωμένου αεροσκάφους γύρω από τον άξονα x και η γωνία Yaw ψ γύρω από τον άξονα z . Το διάνυσμα \mathbf{q} περιέχει τα γραμμικά και περιστροφικά διανύσματα θέσης.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \quad (1)$$

Η αρχή των αξόνων για το μοντέλο μας, είναι στο κέντρο μάζας του. Στο μοντέλο επίσης, οι γραμμικές ταχύτητες συμβολίζονται με \mathbf{V}_B και οι γωνιακές ταχύτητες με \mathbf{v} .

$$\mathbf{V}_B = \begin{bmatrix} U_{x,B} \\ U_{y,B} \\ U_{z,B} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2)$$

Ο πίνακας περιστροφής από το μοντέλο στην αρχική θέση είναι:

$$\mathbf{R} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (3)$$

Ο πίνακας περιστροφής \mathbf{R} είναι ορθογώνιος οπότε ισχύει, $\mathbf{R}^{-1} = \mathbf{R}^T$, οποίος είναι ο πίνακας περιστροφής από την αρχική θέση στο μοντέλο.

Ο πίνακας μετασχηματισμού για τις γωνιακές ταχύτητες από την αρχική θέση στο μοντέλο είναι \mathbf{W}_n , και από το μοντέλο στην αρχική θέση είναι \mathbf{W}_n^{-1} :

$$\dot{\eta} = \mathbf{W}_n^{-1} \mathbf{v}, \quad \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (4)$$

$$\mathbf{v} = \mathbf{W}_n \dot{\eta}, \quad \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \varphi & \cos \theta \sin \varphi \\ 0 & -\sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix},$$

Ο πίνακας \mathbf{W}_n είναι αναστρέψιμος αν ισχύει: $\theta \neq \frac{(2\kappa-1)\varphi}{2}, \kappa \in \mathbb{Z}$

Το quadcopter σαν κατασκευή, θεωρείται συμμετρικό, με τέσσερα άκρα ευθυγραμμισμένα με τους άξονες x, y . Για τον λόγο αυτόν λοιπόν, ο πίνακας αδράνειας του είναι διαγώνιος, με συμβολισμό I , στον οποίο: $I_x = I_y$.

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (5)$$

Η γωνιακή ταχύτητα του μοτέρ i , υποδηλώνοντας ω_i , δημιουργεί μια δύναμη f_i σύμφωνα με την κατεύθυνση του μοτέρ. Η γωνιακή ταχύτητα και επιτάχυνση του μοτέρ δημιουργεί επίσης μια ροπή τ_{Mi} γύρω από τον άξονα του μοτέρ.

$$f_i = k\omega_i^2, \tau_{Mi} = b\omega_i^2 + I_M \dot{\omega}_i \quad (6)$$

στην οποία, η σταθερά ανύψωσης είναι το k , η σταθερά υστέρησης είναι το b και η αδράνεια του μοτέρ είναι I_M . Συχνά το $\dot{\omega}_i$ θεωρείται μικρό και έτσι παραλείπεται.

Ο συνδυασμός των δυνάμεων του μοτέρ δημιουργεί μια ώθηση T , με κατεύθυνση επάνω στον άξονα z. Η ροπή τ_B αποτελείται από τις ροπές τ_φ , τ_θ και τ_ψ , οι οποίες έχουν κατεύθυνση, αντίστοιχη των γωνιών του μοντέλου.

$$T = \sum_{i=1}^4 f_i = k\sum_{i=1}^4 \omega_i^2, \quad T^B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \quad (7)$$

$$\tau_B = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l k (\omega_4^2 - \omega_2^2) \\ l k (\omega_3^2 - \omega_1^2) \\ \sum_{i=1}^4 T_{Mi} \end{bmatrix}, \quad (8)$$

στις οποίες το l είναι η απόσταση μεταξύ του μοτέρ και του κέντρου μάζας του quadcopter. Έτσι, μια κίνηση Roll επιτυγχάνεται μειώνοντας την ταχύτητα του 2^{ου} μοτέρ και αυξάνοντας την ταχύτητα του 4^{ου}, και αντιστρόφως. Ομοίως, μια κίνηση Pitch επιτυγχάνεται μειώνοντας την ταχύτητα του 1^{ου} μοτέρ και αυξάνοντας την ταχύτητα του 3^{ου}, και αντιστρόφως. Μια κίνηση Yaw επιτυγχάνεται αυξάνοντας τις γωνιακές ταχύτητες των δύο απέναντι μοτέρ και μειώνοντας την γωνιακή ταχύτητα των άλλων δύο.

2.1.2 Εξισώσεις Νεύτωνα-Euler

Το quadcopter θεωρείται μια άκαμπτη κατασκευή και έτσι οι εξισώσεις του Νεύτωνα-Euler μπορούν να χρησιμοποιηθούν για να περιγράψουν την δυναμική του. Στο μοντέλο, η δύναμη που απαιτείται για την επιτάχυνση της μάζας $m\dot{V}_B$ και της φυγόκεντρης δύναμης $\mathbf{v} \times (m\mathbf{V}_B)$ είναι ίση με την συνολική βαρύτητα $\mathbf{R}^T \mathbf{G}$ συν την συνολική ανύψωση των μοτέρ \mathbf{T}_B .

$$m\dot{V}_B + \mathbf{v} \times (m\mathbf{V}_B) = \mathbf{R}^T \mathbf{G} + \mathbf{T}_B \quad (9)$$

Στην αρχική θέση, η φυγόκεντρος είναι μηδενική. Για τον λόγο αυτό, οι δυνάμεις που συνεισφέρουν στην επιτάχυνση του quadcopter, είναι αυτή της βαρύτητας και η έκταση και κατεύθυνση της ανύψωσης του.

$$m\ddot{\xi} = \mathbf{G} + \mathbf{R}\mathbf{T}_B ,$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ \cos \theta \cos \varphi \end{bmatrix} \quad (10)$$

Στο μοντέλο, η γωνιακή επιτάχυνση της αδράνειας $\mathbf{I}\dot{\boldsymbol{\nu}}$, η κεντρομόλος δύναμη $\mathbf{v} \times (\mathbf{I}\boldsymbol{\nu})$ και οι δυνάμεις από το γυροσκόπιο $\boldsymbol{\Gamma}$, είναι ίσα με την εξωτερική ροπή $\boldsymbol{\tau}$.

$$\mathbf{I}\dot{\boldsymbol{\nu}} + \mathbf{v} \times (\mathbf{I}\boldsymbol{\nu}) + \boldsymbol{\Gamma} = \boldsymbol{\tau} ,$$

$$\dot{\boldsymbol{\nu}} = \mathbf{I}^{-1} \left(- \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} \times \begin{bmatrix} I_x \mathbf{p} \\ I_y \mathbf{q} \\ I_z \mathbf{r} \end{bmatrix} - I_r \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \boldsymbol{\omega}_\Gamma + \boldsymbol{\tau} \right) , \quad (11)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_y - I_z)qr/I_x \\ (I_z - I_x)pr/I_y \\ (I_x - I_y)pq/I_z \end{bmatrix} - I_r \begin{bmatrix} q/I_x \\ -p/I_y \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\varphi/I_x \\ \tau_\theta/I_y \\ \tau_\psi/I_z \end{bmatrix},$$

στην οποία, $\omega_\Gamma = \omega_1 - \omega_2 + \omega_3 - \omega_4$. Η γωνιακή επιτάχυνση στην αρχική θέση έλκεται από την επιτάχυνση του μοντέλου με τον πίνακα μετασχηματισμού W_n^{-1} και την παράγωγό του.

$$\ddot{\eta} = \frac{d}{dt}(W_n^{-1}\mathbf{v}) = \frac{d}{dt}(W_n^{-1})\mathbf{v} + W_n^{-1}\dot{\mathbf{v}} \quad (12)$$

$$= \begin{bmatrix} 0 & \dot{\varphi} \cos \varphi \tan \theta + \dot{\theta} \sin \varphi / \cos^2 \theta & -\dot{\varphi} \sin \varphi \cos \theta + \dot{\theta} \cos \varphi / \cos^2 \theta \\ 0 & -\dot{\varphi} \sin \varphi & -\dot{\varphi} \cos \varphi \\ 0 & \dot{\varphi} \cos \varphi / \cos \theta + \dot{\varphi} \sin \varphi \tan \theta / \cos \theta & -\dot{\varphi} \sin \varphi / \cos \theta + \dot{\theta} \cos \varphi \tan \theta / \cos \theta \end{bmatrix} \mathbf{v} + W_n^{-1}\dot{\mathbf{v}}$$

2.1.3 Εξισώσεις Euler – Lagrange

Το Lagrangian L είναι το άθροισμα της κινητικής μεταφορικής ενέργειας E_{trans} και της κινητικής περιστροφικής ενέργειας E_{rot} πλην της δυναμικής ενέργειας E_{pot} .

$$L(q, \dot{q}) = E_{trans} + E_{rot} - E_{pot} = \frac{m}{2} \dot{\xi}^T \dot{\xi} + \frac{1}{2} v^T I v - mgz \quad (13)$$

Η Euler – Lagrange εξίσωση με τις εξωτερικές δυνάμεις και ροπές είναι:

$$\begin{bmatrix} f \\ \tau \end{bmatrix} = \frac{d}{dt} \left(\frac{dL}{d\dot{q}} \right) - \frac{dL}{dq} \quad (14)$$

Τα γραμμικά και τα γωνιακά στοιχεία δεν εξαρτώνται το ένα από το άλλο και έτσι μπορούν να μελετηθούν ξεχωριστά. Η γραμμική εξωτερική δύναμη είναι η συνολική ώθηση των μοτέρ. Οι γραμμικές Euler – Lagrange εξισώσεις είναι:

$$f = RT_B = m\ddot{\xi} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (15)$$

η οποία είναι ισοδύναμη με την εξίσωση (10).

Ο πίνακας Jacobian $J(\eta)$ από v μέχρι $\dot{\eta}$ είναι:

$$J(\eta) = J = W_{\eta}^T / W_{\eta} ,$$

$$= \begin{bmatrix} I_x & 0 & -I_x \sin \theta \\ 0 & I_y \cos^2 \varphi + I_z \sin^2 \varphi & (I_y - I_z) \cos \varphi \sin \varphi \cos \theta \\ -I_x \sin \theta & (I_y - I_z) \cos \varphi \sin \varphi \cos \theta & I_x \sin^2 \theta + I_y \sin^2 \varphi \cos^2 \theta + I_z \cos^2 \varphi \cos^2 \theta \end{bmatrix} \quad (16)$$

Έτσι, η περιστροφική ενέργεια E_{rot} μπορεί να εκφραστεί στο αρχικό πλαίσιο

$$E_{rot} = \frac{1}{2} v^T I v = \frac{1}{2} \dot{\eta}^T J \dot{\eta}. \quad (17)$$

Η εξωτερική περιστροφική δύναμη είναι οι ροπές των μοτέρ. Οι γωνιακές Euler-Lagrange εξισώσεις είναι:

$$\tau = \tau_B = J\ddot{\eta} + \dot{J}\dot{\eta} - \frac{1}{2} \frac{d}{d\eta} (\dot{\eta}^T J \dot{\eta}) = J\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} \quad (18)$$

στην οποία ο πίνακας $C(\eta, \dot{\eta})$ είναι ο όρος Coriolis, ο οποίος περιέχει τους γυροσκοπικούς και κεντρομόλους όρους.

2.1.4 Αεροδυναμικές επιδράσεις

Το προηγούμενο μοντέλο είναι μια απλοποίηση πολύπλοκων δυναμικών αλληλεπιδράσεων. Για να επιβάλουμε μια πιο ρεαλιστική συμπεριφορά του quadcopter, θα συμπεριλάβουμε δυνάμεις οι οποίες παράγονται από την αντίσταση του αέρα. Έτσι, η εξίσωση (10), με τον διαγώνιο πίνακα συντελεστών που συνδυάζει τις γραμμικές ταχύτητες με την δύναμη που επιβραδύνει την κίνηση, θα γίνει:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ \cos \theta \cos \varphi \end{bmatrix} - \frac{1}{m} \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (19)$$

στην οποία οι συντελεστές A_x , A_y και A_z είναι οι συντελεστές της δύναμης για τις ταχύτητες των αντίστοιχων κατευθύνσεων στην αρχική θέση.

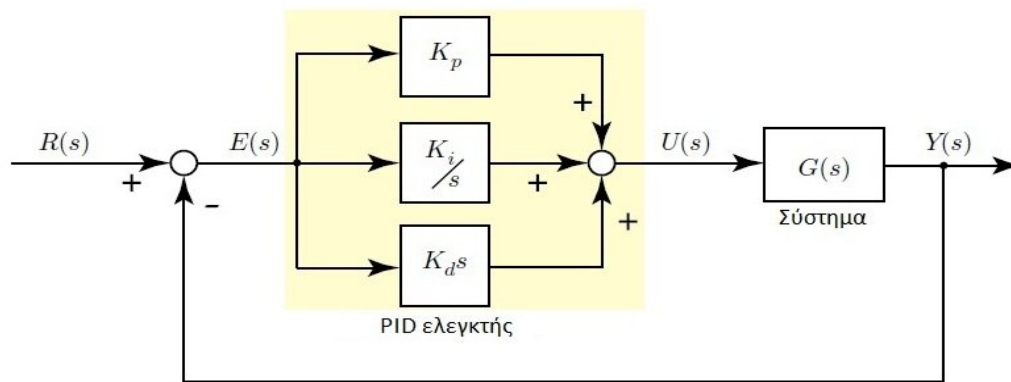
Πολλές άλλες αεροδυναμικές επιδράσεις μπορούν να συμπεριληφθούν στο μοντέλο. Για παράδειγμα, η εμπιστοσύνη της υστέρησης πάνω στην γωνία εφαρμογής, ο κυματισμός του πτερυγίου και οι διαταραχές στην ροή του αέρα. Η επιρροή των αεροδυναμικών επιδράσεων είναι πολύπλοκη και οι επιδράσεις είναι δύσκολο να μοντελοποιηθούν. Επίσης, κάποιες από τις επιδράσεις αποκτούν υπόσταση σε υψηλές ταχύτητες λειτουργίας. Έτσι, οι επιδράσεις αυτές, εξαιρούνται από το μοντέλο και χρησιμοποιείται το μοντέλο που παρουσιάστηκε νωρίτερα.

2.2 Εξισώσεις ελεγκτών

2.2.1 Ο ελεγκτής PID

– Θεωρητικό Υπόβαθρο:

Ένας PID ελεγκτής είναι ένας γενικός μηχανισμός κλειστού βρόχου, ο οποίος χρησιμοποιείται ευρέως σε βιομηχανικά συστήματα ελέγχου. Ο ελεγκτής αυτός, υπολογίζει ένα "σφάλμα", και πιο συγκεκριμένα την διαφορά μεταξύ της επιθυμητής μας εισόδου και της πραγματικής εξόδου του συστήματός μας. Ο PID επιχειρεί να ελαχιστοποιήσει το "σφάλμα" αυτό, όσο το περισσότερο δυνατόν. Το πόσο εύστοχα και γρήγορα θα καταφέρει να μηδενίσει το "σφάλμα", εξαρτάται από την ρύθμιση που θα του γίνει.



Εικόνα 3: Ο ελεγκτής PID σε σύστημα

Η εξίσωση στο πεδίο του χρόνου που περιγράφει την λειτουργία του ελεγκτή αυτού είναι η εξής:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_D \frac{d}{dt} e(t) ,$$

στην οποία:

- K_p : Αναλογικό κέρδος
- K_i : Ολοκληρωτικό κέρδος
- K_D : Διαφορικό κέρδος

Η αντίστοιχη εξίσωση στο πεδίο της συχνότητας είναι η εξής:

$$C(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s},$$

η οποία αποτελεί και την συνάρτηση μεταφοράς του.

Εμπειρικά, τα κέρδη K_p , K_I , K_D μπορούν να ερμηνευθούν χρονικά:

- το αναλογικό κέρδος (K_p) εξαρτάται από το σφάλμα του παρόντος,
- το ολοκληρωτικό κέρδος (K_I) εξαρτάται από την συσσώρευση προηγούμενων σφαλμάτων,
- και το διαφορικό κέρδος (K_D) είναι μια πρόβλεψη μελλοντικών σφαλμάτων.

Κάποια συστήματα μπορεί να μην χρειάζονται και τα τρία στοιχεία ενός PID για να αποκτήσουν την επιθυμητή απόκριση, αλλά μόνο ένα ή δύο εξ αυτών. Αυτό επιτυγχάνεται θέτοντας τις σταθερές που δεν χρειαζόμαστε, ίσες με το μηδέν. Όταν απουσιάζουν λοιπόν κάποιες από τις τρεις σταθερές ενός PID, αυτός αλλάζει όνομα, αφήνοντας έξω από αυτό το αντίστοιχο στοιχείο που του λείπει. Παραδείγματα τέτοιων περιπτώσεων, είναι ο αναλογικός ελεγκτής P ($K_I = 0$, $K_D = 0$), ο ολοκληρωτικός ελεγκτής I ($K_p = 0$, $K_D = 0$), ο διαφορικός ελεγκτής D ($K_p = 0$, $K_I = 0$), ο ελεγκτής PI ($K_D = 0$) και ο ελεγκτής PD ($K_I = 0$). Ο ελεγκτής PI είναι αρκετά συχνός στην πράξη, καθώς ο διαφορικός παράγοντας D ενός ελεγκτή PID, είναι ευαίσθητος στον θόρυβο των μετρήσεων, και λαμβάνοντας υπόψιν πως η απουσία του ολοκληρωτικού παράγοντα I, μπορεί να εμποδίσει το σύστημά μας από το να φθάσει την επιθυμητή μας απόκριση.

Στον πίνακα που ακολουθεί, θα δούμε την επίδραση που έχει κάθε σταθερά ενός PID (K_p , K_i , K_D) σε ένα σύστημα κλειστού βρόχου.

Σταθερά Ελεγκτή	Χρόνος Ανύψωσης	Υπερύψωση	Χρόνος Αποκατάστασης	Μόνιμο Σφάλμα
K_p	Μείωση	Αύξηση	Μικρή αλλαγή	Μείωση
K_i	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
K_D	Μικρή αλλαγή	Μείωση	Μείωση	Μικρή αλλαγή

Πίνακας 1: Επίδραση κάθε σταθεράς ενός PID

Να σημειωθεί πως ο παραπάνω πίνακας μπορεί να μην είναι απόλυτα ακριβής διότι οι σταθερές K_p , K_i , K_D αλληλεξαρτώνται. Στην πραγματικότητα, η αλλαγή μιας εκ των τριών σταθερών, μπορεί να επιρρεάσει την επίδραση και των άλλων δύο.

- Μέθοδοι ρύθμισης ενός ελεγκτή PID:

Έχουν αναπτυχθεί διάφορες μέθοδοι ρύθμισης ενός ελεγκτή PID. Οι πιο γνωστές μέθοδοι, είναι αυτές των Ziegler-Nichols και των Cohen-Coon, είτε το σύστημά μας είναι γνωστό είτε άγνωστο. Σε ορισμένες περιπτώσεις όμως, μπορεί να μην αρκούν αυτές οι μέθοδοι ώστε να πετύχουμε απόλυτα την επιθυμητή απόκριση στο σύστημά μας και γι' αυτό να χρειάζεστε να επέμβουμε εμείς στις ρυθμίσεις του ελεγκτή μας. Κι αν και οι δικές μας επεμβάσεις δεν οδηγήσουν στο επιθυμητό αποτέλεσμα, τότε με την βοήθεια κάποιου προγράμματος θα μπορέσουμε με αρκετά μεγάλη ακρίβεια να επιτύχουμε μια απόκριση όμοια με την επιθυμητή μας, δίνοντας επαρκή δεδομένα στο πρόγραμμά μας.

Παρακάτω, θα παρουσιάσουμε συνοπτικά την μέθοδο υπολογισμού, κατα Ziegler-Nichols, των τριών σταθερών K_p , K_i , K_D ενός ελεγκτή PID, όταν το σύστημά μας είναι γνωστό και όταν είναι άγνωστο.

Ziegler – Nichols:

α) Γνωστό σύστημα

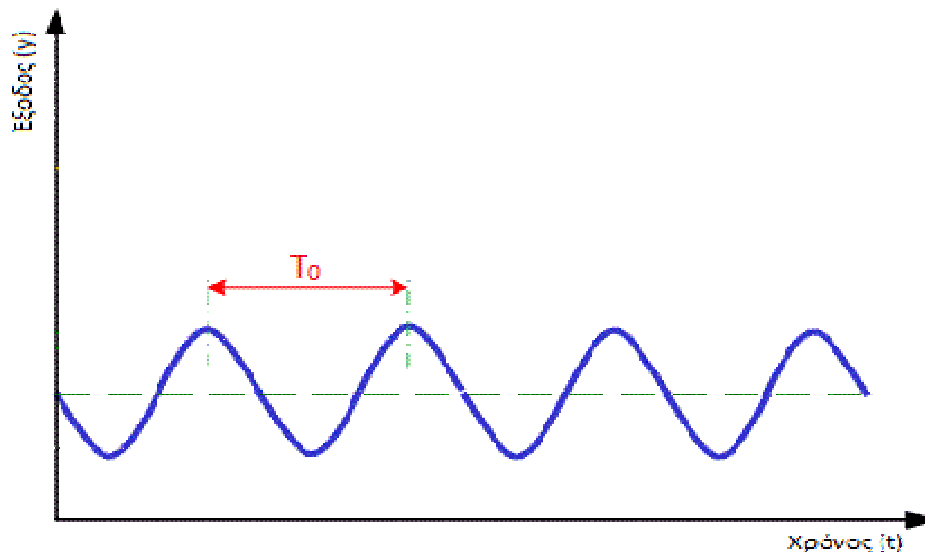
Η διαδικασία αυτή αφορά τις περιπτώσεις, στις οποίες το σύστημα $G(s)$, μας είναι γνωστό.

Έχουν αναπτυχθεί διάφοροι τρόποι υπολογισμού των παραμέτρων του ελεγκτή PID, θα αναφερθούμε όμως σε από τους τρόπους αυτούς.

- 1) Θέτουμε τους όρους διαφόρισης και ολοκλήρωσης (K_I και K_D) σε όσο το δυνατόν χαμηλότερες τιμές, και αυξάνουμε σταδιακά την τιμή του αναλογικού όρου K_P μέχρι το σύστημά μας να αποκτήσει απόκριση οριακής ευστάθειας (απλή αρμονική ταλάντωση). Η τιμή του K_P όταν παίρνουμε την απόκριση αυτή ονομάζεται και K_{op} με περίοδο ταλάντωσης T_0 η οποία υπολογίζεται από την κυματομορφή μας. Έχοντας συλλέξει τα στοιχεία αυτά, οι παράμετροι υπολογίζονται από τις παρακάτω εξισώσεις:

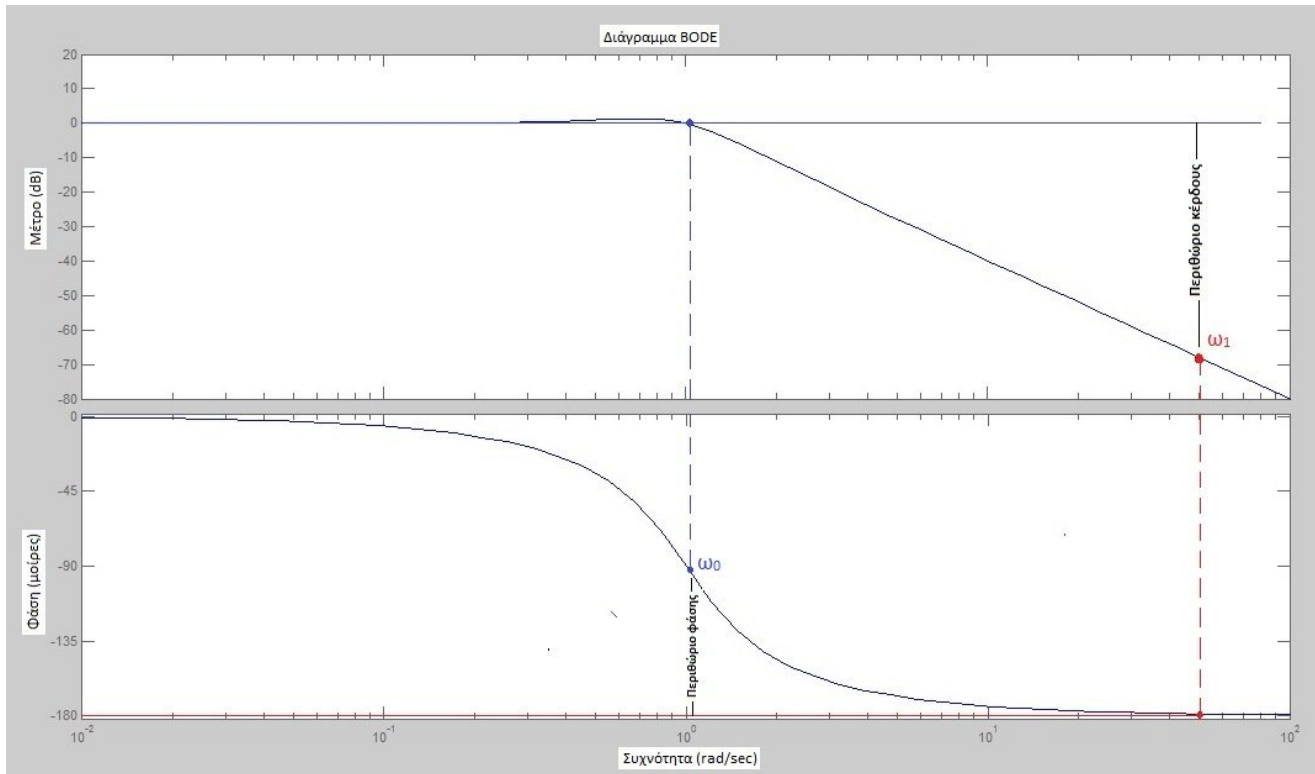
Ελεγκτής	K_P	K_I	K_D
P	$0.5K_{op}$	-	-
PI	$0.45K_{op}$	$1.2K_P/T_0$	-
PID	$0.6K_{op}$	$2K_P/T_0$	$K_P T_0/8$

Πίνακας 2: Υπολογισμός των K_P , K_I , K_D με δεδομένα τα K_{op} , T_0



Εικόνα 4: Απόκριση συστήματος όταν $K_P = K_{op}$ και μέτρηση της περιόδου

2) Εναλλακτικά, οι παράμετροι K_{op} και T_0 μπορούν να υπολογισθούν από τα διαγράμματα BODE, τα οποία περιγράφουν την απόκριση του συστήματός μας, στο πεδίο της συχνότητας (μέτρο και φάση)



Εικόνα 5: Διάγραμμα BODE συστήματος και εύρεση του περιθώριου κέρδους και φάσης

- Περιθώριο Κέρδους:

Το περιθώριο κέρδους το βρίσκουμε από την απόκριση φάσης. Πηγαίνουμε στις -180° και βλέπουμε σε ποιο σημείο τέμνει την απόκρισή μας. Προεκτείνουμε το σημείο αυτό και το μεταφέρουμε στην απόκριση του μέτρου και εκεί που θα την τέμνει, αυτό θα είναι το σημείο ω_1 . Η απόσταση του ω_1 από τα 0dB, είναι και το περιθώριο κέρδους.

- Περιθώριο Φάσης:

Το περιθώριο φάσης το βρίσκουμε από την απόκριση μέτρου. Πηγαίνουμε στα 0dB και βλέπουμε σε ποιο σημείο τέμνει την απόκρισή μας. Προεκτείνουμε το σημείο αυτό και το μεταφέρουμε στην απόκριση της φάσης και εκεί που θα την, αυτό θα είναι το σημείο ω_0 . Η απόσταση του ω_0 , από τις -180° , είναι και το περιθώριο φάσης.

Σημείωση: Για να είναι ευσταθές ένα σύστημα, πρέπει το περιθώριο κέρδους να είναι κάτω από τα 0dB και/ή το περιθώριο φάσης να είναι πάνω από τις -180° .

Υπολογισμοί:

1. $K_{κρ} = \frac{1}{A_{dB}}$

2. $T_0 = \frac{2\pi}{\omega_0}$

Στην συνέχεια υπολογίζουμε με βάση τους προηγούμενους τύπους τα K_p , K_i , K_D του ελεγκτή μας. Για να επιτύχουμε τέλεια απόκριση, οι τιμές ίσως χρειαστούν περαιτέρω ρύθμιση, αλλά παρόλα αυτά το συστήμα μας θα βρίσκεται πολύ κοντά στην επιθυμητή.

β) Άγνωστο σύστημα

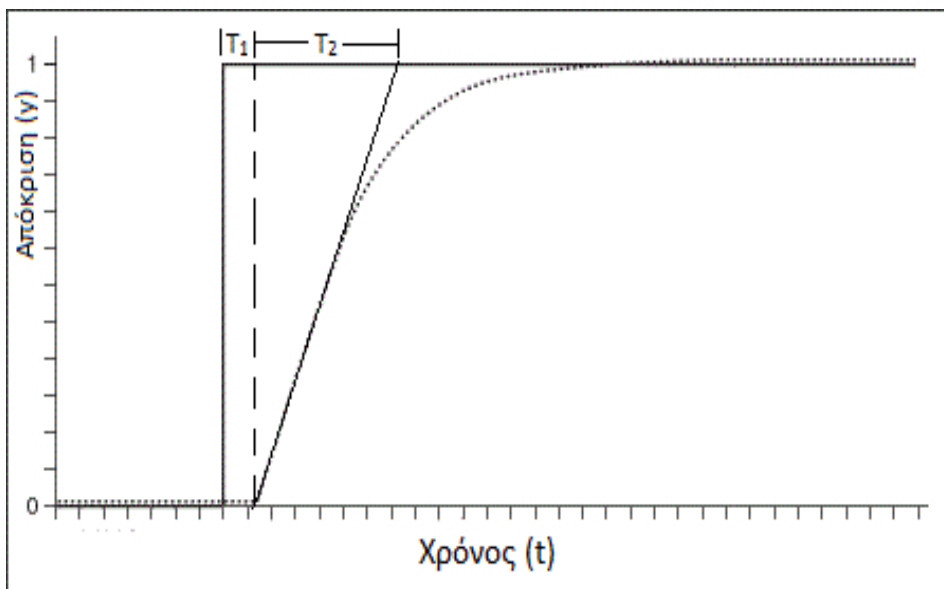
Πρώτο μας βήμα στην περίπτωση αυτή είναι να βρούμε το άγνωστο μας σύστημα $G(s)$ και στην συνέχεια να υπολογίσουμε τις παραμέτρους του PID ελεγκτή μας.

Το σύστημά μας μπορούμε να το υπολογίσουμε μέσω της απόκρισής του, στο πεδίο του χρόνου, εφαρμόζοντάς του μια βηματική μεταβολή σε ανοιχτό βρόγχο. Από την απόκριση αυτή μπορούμε να υπολογίσουμε κατά προσέγγιση τις παραμέτρους

του άγνωστου συστήματος μας. Το συνηθέστερο είναι να γίνει επιλογή ενός συστήματος πρώτου βαθμού της μορφής:

$$G(s) = \frac{K}{1+sT_2} e^{-sT_1}$$

Οι τιμές K , T_1 , T_2 βρίσκονται γραφικά από την βηματική απόκριση του συστήματος μας. Με τις τιμές των K , T_1 , T_2 μπορούμε να υπολογίσουμε προσεγγιστικά τις τιμές των παραμέτρων του ελεγκτή μας (K_p , K_i , K_D). Τέλος, με δικές μας ρυθμίσεις κόντα στις τιμές που έχουμε βρει, επιτυγχάνουμε τον έλεγχο του άγνωστου συστήματος.



Εικόνα 6: Βηματική απόκριση συστήματος ανοικτού βρόγχου

Εφόσον βρήκαμε τις τιμές K , T_1 , T_2 μπορούμε να βρούμε την συνάρτηση μεταφοράς του PID ελεγκτή μας.

$$C(s) = K_p \left(1 + \frac{K_i}{s} + K_D s \right)$$

Ελεγκτής	K_p	K_i	K_D
P	$\frac{1}{ST_1}$	-	-
PI	$\frac{0.9}{ST_1}$	$3T_1$	-
PID	$\frac{1.2}{ST_1}$	$2T_1$	$0.5T_1$

Πίνακας 3: Υπολογισμός παραμέτρων PID με γνωστές τις τιμές K , T_1 , T_2

$$\text{με } S = \frac{K}{T_2} \text{ (κλίση της απόκρισης)}$$

2.2.2 Ελεγκτής ασαφούς λογικής

Ασαφής λογική είναι ένα συγκρότημα λογικών συνθηκών ή λογικών πιθανοτήτων, η οποία έχει να κάνει με κατά προσέγγιση αιτιολογίες πάρα με σταθερές και ακριβείς. Σε σύγκριση με το δυαδικό σύστημα, στο οποίο οι τιμές μας μεταφράζονται ως ψευδείς ή αληθείς, τιμές στην ασαφή λογική οι τιμές μπορεί να έχουν αληθείς(1) και ψευδείς(0) τιμές όμως έχουν και ενδιάμεσες καταστάσεις μεταξύ των τιμών αυτών. Η ασαφής λογική αναπτύχθηκε για να χειρίζεται την έννοια της μερικής αλήθειας, στην οποία η αληθής τιμή μπορεί να βρίσκεται ανάμεσα στην απόλυτα αληθή και ψευδή τιμή. Επιπλέον, όταν χρησιμοποιούνται γλωσσικές μεταβλητές, οι ενδιάμεσες αυτές τιμές μπορούν να διαχειρίζονται από συγκεκριμένες συναρτήσεις. Το φαινόμενο αυτό είναι γνωστό ως “fuzzjective”. Η ασαφής λογική χρησιμοποιείται σε πολλούς τομείς, από την θεωρία του ελέγχου μέχρι την τεχνητή νοημοσύνη.

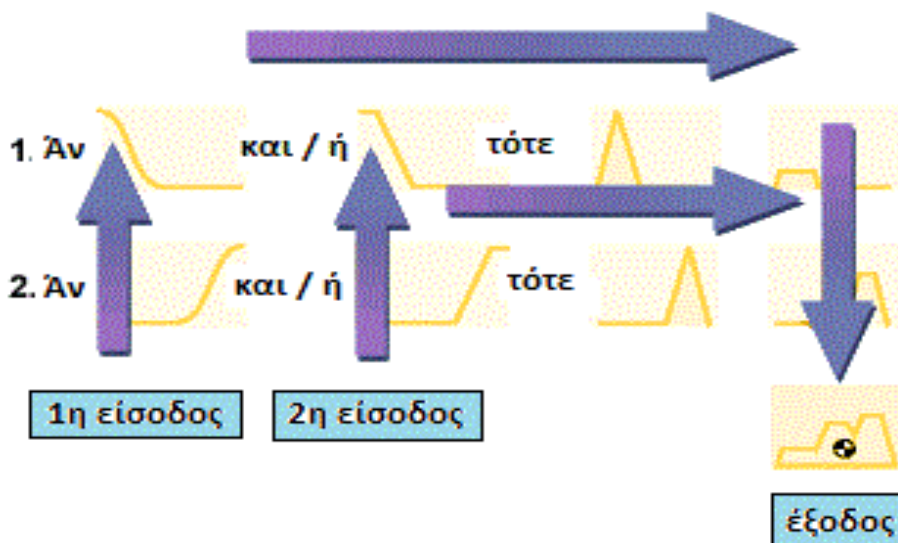
Μια βασική εφαρμογή της ασαφούς λογικής θα μπορούσε να χαρακτηρίζει τμήματα μιας συνεχής μεταβλητής. Για παράδειγμα, μια μέτρηση θερμοκρασίας για το σύστημα απεμπλοκής των τροχών θα μπορούσε να αποτελείται από αρκετές συναρτήσεις, οι οποίες θα καθορίζουν συγκεκριμένα τμήματα θερμοκρασίας για τα οποία θα χρειαστεί να γίνει έλεγχος των φρένων. Κάθε συνάρτηση από αυτές, καθορίζει ένα εύρος τιμών, μέσα στο οποίο εκτελεί μια ακολουθία. Το εύρος κάθε κυμαίνεται μεταξύ των τιμών 0 και 1. Αναλόγως με το αποτέλεσμα που θα παρθεί από την μέτρησή μας, θα κλιθεί και η ανάλογη συνάρτηση.

Καθώς οι μεταβλητές στα μαθηματικά παίρνουν ως τιμές αριθμούς, σε εφαρμογές ασαφούς λογικής, η μη-αριθμητικές “γλωσσικές μεταβλητές” χρησιμοποιούνται συχνά για να διευκολύνουν τον χαρακτηρισμό των κανόνων και των γεγονότων. Μια γλωσσική μεταβλητή όπως για παράδειγμα η ηλικία θα μπορούσε να έχει την τιμή νέος ή το αντίθετό του γέρος. Ωστόσο, η μεγάλη χρησιμότητα των μεταβλητών αυτών

είναι πως μπορούν να τροποποιηθούν μέσω γλωσσικών αντισταθμιστών οι οποίοι εφαρμόζονται σε πρωτογενείς όρους. Οι γλωσσικές μεταβλητές μπορούν να συνδεθούν με ορισμένες συναρτήσεις.

- Λειτουργία της ασαφούς λογικής:

Όσο δύσκολη κι αν μας φαίνεται η κατανόηση ενός τέτοιου ελεγκτή, η υλοποίησή του στην πραγματικότητα είναι πολύ απλή, και δίνεται με τέτοια δομή, που σου δίνει πάρα πολλές δυνατότητες. Η ασαφής λογική, βασίζεται στην γνωστή σε όλους μας συνθήκη αληθείας: Άν (συνθήκη) τότε..... Το δύσκολο κομμάτι στον ελεγκτή αυτόν, είναι η ρύθμιση των εισόδων – εξόδων μας, ώστε να πάρουμε τα επιθυμητά αποτελέσματα. Παρακάτω σας παρουσιάζουμε σχηματικά έναν τέτοιο ελεγκτή, και στην συνέχεια, μια πιθανή λειτουργία του.



Εικόνα 7: Ακολουθία λειτουργίας ενός απλού ελεγκτή ασαφούς λογικής

Ένας ελεγκτής ασαφούς λογικής θα μπορούσε να εκτελεί την παρακάτω ακολουθία εντολών:

```
IF temperature IS very cold THEN stop fan  
IF temperature IS cold THEN turn down fan  
IF temperature IS normal THEN maintain level  
IF temperature IS hot THEN speed up fan
```

Στις συνθήκες του ελεγκτή ασαφούς λογικής, δεν υπάρχει η εντολή “αλλιώς”, γιατί μέσα σε δυο διαφορετικά διαστήματα τιμών, μπορεί να υπάρξει ταύτιση. Όπως μπορούμε να δούμε πρόκειται για έναν πολύ απλό τρόπο σκέψης, ο οποίος όμως για να λειτουργήσει σωστά, θα πρέπει να του γίνει πολύ καλή ρύθμιση. Πρέπει να επιλέξουμε με όσο το δυνατόν περισσότερη ακρίβεια τις γραφικές που θέλουμε να περιγράψουν τις εισόδους και τις εξόδους μας. Υπάρχει μια μεγάλη ποικιλία γραφικών που μας προσφέρει το MATLAB. Όμως παρόλο που μπορεί να έχουμε κάνει την καλύτερη επιλογή και ρύθμιση των γραφικών μας, υπάρχει περίπτωση και πάλι να μην πάρουμε το επιθυμητό αποτέλεσμα. Το γιατί μπορεί να συμβεί κάτι τέτοιο θα το δούμε καλύτερα στο κομμάτι της υλοποίησης ενός τέτοιου ελεγκτή στο μοντέλο μας.

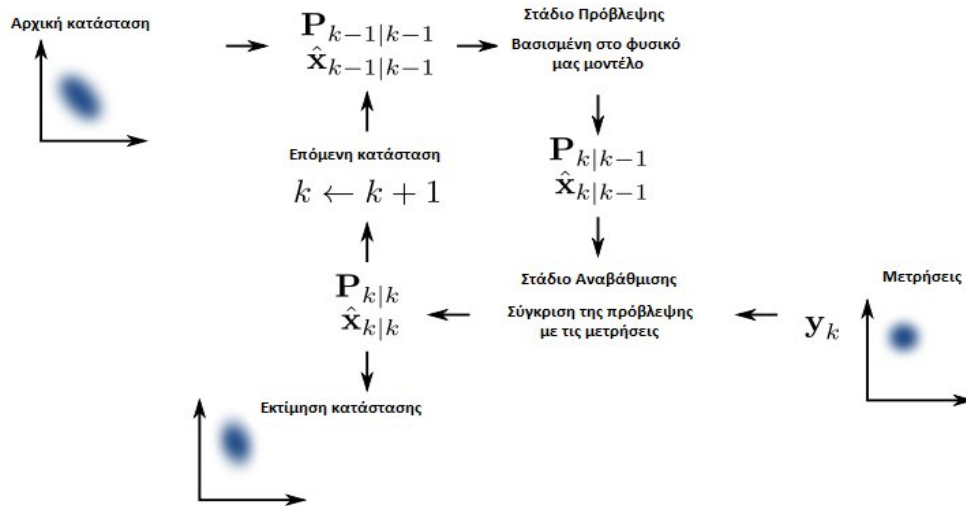
2.2.3 Το φίλτρο Kalman

Το φίλτρο Kalman, είναι ένας αλγόριθμος ο οποίος χρησιμοποιεί μια σειρά από μετρήσεις, οι οποίες παρατηρήθηκαν με το πέρας του χρόνου και περιέχουν θόρυβο και άλλες αστοχίες, για να παράγει εκτιμήσεις άγνωστων μεταβλητών οι οποίες έτειναν να είναι πιο ακριβείς από αυτές που βασίζονταν σε μια και μόνη μέτρηση. Πιο επίσημα, το φίλτρο Kalman λειτουργεί αναδρομικά σε είσοδους με θόρυβο για να παράγει μια στατιστική βέλτιστη εκτίμηση του χώρου κατάστασης.

Το φίλτρο Kalman έχει πολυάριθμες εφαρμογές στην τεχνολογία. Συχνή εφαρμογή του φίλτρου αφορά την καθοδήγηση και τον έλεγχο οχημάτων, και ιδιαίτερα αεροσκαφών και διαστημόπλοιων.

Λειτουργία του φίλτρου kalman

Ο αλγόριθμος αυτός λειτουργεί σε μια διαδικασία δυο σταδίων. Στο στάδιο της πρόβλεψης, το φίλτρο kalman παράγει εκτιμήσεις των τωρινών μεταβλητών κατάστασης, μαζί με τον θόρυβο. Μόλις εφαρμοστεί η επόμενη μέτρηση (σίγουρα διαταραγμένη με κάποιο σφάλμα, συμπεριλαμβανομένου του τυχαίου σφάλματος), οι προηγούμενες εκτιμήσεις αναβαθμίζονται με έναν σταθμισμένο μέσο όρο, ο οποίος έχει περισσότερη βαρύτητα σε πιο σίγουρες εκτιμήσεις. Εξαιτίας της αναδρομικότητας του αλγορίθμου, μπορεί να "τρέξει" σε πραγματικό χρόνο χρησιμοποιώντας μόνο τις παρούσες μετρήσεις και την προηγούμενη υπολογισμένη κατάσταση.



Εικόνα 8: Η λειτουργία ενός φίλτρου kalman

Εξισώσεις ενός φίλτρου kalman

Το φίλτρο kalman διευθύνει το γενικό πρόβλημα προσπαθώντας να κάνει πρόβλεψη της κατάστασης $x \in \mathcal{R}^n$ μιας διαδικασίας ελέγχου, η οποία διέπεται από την γραμμική εξίσωση:

$$\bar{x}_t = Ax_{t-1} + Bu_t + \varepsilon_x \quad (1)$$

με μια πρόβλεψη μέτρησης $z \in \mathcal{R}^n$:

$$\bar{z}_t = C\bar{x}_t + \varepsilon_z \quad (2)$$

με: $\varepsilon_x, \varepsilon_z$: Gaussian errors

\bar{x}_t : πρόβλεψη κατάστασης

\bar{z}_t : πρόβλεψη μέτρησης

Η εκτίμηση της κατάστασης, γίνεται από την παρακάτω εξίσωση:

$$x_{\text{εκτ.}} = \bar{x}_t + K(z_t - \bar{z}_t) \quad (3)$$

με: K : κέρδος kalman

z_t : Μέτρηση

Σημείωση: Ο όρος $K(z_t - \bar{z}_t)$ διορθώνει την πρόβλεψη μας, σε περίπτωση που η πρόβλεψη της μέτρησης έχει απόκλιση από την πραγματική μέτρηση, για να έχουμε καλύτερη εκτίμηση της κατάστασης.

Για να υπολογίσουμε τους πίνακες A, B, C θα μας βοηθήσουν οι εξισώσεις θέσης και ταχύτητας.

$$X = \begin{bmatrix} p \\ v \end{bmatrix}, \quad p_t = p_{t-1} + v_{t-1}t + \frac{1}{2}\alpha_t t^2$$

$$v_t = v_{t-1} + \alpha_t t$$

κοιτώντας την εξίσωση (1), (2):

$$\bar{x}_t = \begin{bmatrix} p_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix} \alpha_t + \varepsilon_x \quad (4)$$

$$\bar{z}_t = p_{t+1} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_t \\ v_t \end{bmatrix} + \varepsilon_z \quad (5)$$

Οπότε, με την βοήθεια των εξισώσεων θέσης και ταχύτητας από την φυσική, οι τιμές των πινάκων A, B, C είναι:

$$A = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{dt^2}{2} \\ dt \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Τα ε_x , ε_z υπολογίζονται ως εξής:

$$\varepsilon_x = E_x = \begin{bmatrix} \sigma_p^2 & \sigma_p \sigma_v \\ \sigma_v \sigma_p & \sigma_v^2 \end{bmatrix}, \quad \varepsilon_z = E_z = \sigma_x^2$$

Ο πίνακας covariance για την πρόβλεψη του σφάλματος είναι:

$$\bar{\Sigma}_t = A \Sigma_{t-1} A^T + E_x$$

Το κέρδος kalman υπολογίζεται από τον παρακάτω τύπο:

$$K = \bar{\Sigma}_t C^T (C \bar{\Sigma}_t C^T + E_z)^{-1}$$

Και τέλος ο πίνακας covariance για την εκτίμηση του σφάλματος είναι:

$$\Sigma_t = (I - KC) \bar{\Sigma}_t$$

Για την υποστήριξη της θεωρίας του φίλτρου Kalman επειδή πρόκειται για ένα αρκετά περίπλοκο “φίλτρο” αναπλύχθηκε κώδικας στο MATLAB ενός τυχαίου σήματος ώστε να δούμε στην πράξη πως λειτουργεί. Ο κώδικας που αναπτύχθηκε μαζί με τον σχολιασμό του, ακολουθεί αμέσως:

```
%Ορίζουμε τα δείγματά μας (διάρκεια και συχνότητα δειγματοληψίας)
diarkeia = 10;
dt = 0.1;

%Ορίζουμε τις εξισώσεις ανανέωσης (πίνακες Coefficient)
A = [1 dt; 0 1];
B = [(dt^2)/2; dt];
C = [1 0];

%Ορίζουμε τις κύριες μεταβλητές
u = 1.5; %Ορισμός εισόδου
Q = [0; 0]; %Αρχικοποίηση κατάστασης... έχουμε δυο συνιστώσες [θέση/ταχύτητα]
Q_estimate = Q;
thoruvos_diadikasias_mag = 0.5; %"Θόρυβος" διαδικασίας
thoruvos_metrisis_mag = 3; %"Θόρυβος" μετρήσης
Ez = thoruvos_metrisis_mag^2; %μετατροπή "θορύβου" μέτρησης σε covariance μέτρηση
Ex = (thoruvos_diadikasias_mag^2) * [(dt^4)/4 (dt^3)/2; (dt^3)/2 dt^2]; %μετατροπή
"θορύβου" διαδικασίας σε covariance κατάσταση
P = Ex; %Πίνακας Covariance

%Αρχικοποίηση μεταβλητών αποτελεσμάτων
Q_loc = []; %Πραγματική διαδρομή
vel = []; %Πραγματική ταχύτητα
Q_loc_meas = []; %Διαδρομή χωρίς φίλτρο

%Εξομοίωση διαδρομής σε σχέση με τον χρόνο
figure(2);clf
figure(1);clf
for t = 0 : dt : diarkeia

    %Παραγωγή διαδρομής
    thoruvos_diadikasias = (thoruvos_diadikasias_mag^2) * (((dt^2)/2)*randn;
dt*randn);
    Q = A * Q+ B * u+ thoruvos_diadikasias;

    %Παραγωγή διαδρομής βάσει μετρήσεων
    thoruvos_metrisis = thoruvos_metrisis_mag * randn;
    y = C * Q+ thoruvos_metrisis;
    Q_loc = [Q_loc; Q(1)];
    Q_loc_meas = [Q_loc_meas; y];
    vel = [vel; Q(2)];
    %Επαναληπτικό γράφημα για τις μετρήσεις μας
    plot(0:dt:t, Q_loc, '-r.')
    plot(0:dt:t, Q_loc_meas, '-b.')
    axis([0 10 -30 80])
    hold on
    pause
end

%Θεωρητικό γράφημα για τις μετρήσεις χωρίς kalman
plot(0:dt:t, smooth(Q_loc_meas), '-g.')
```

```

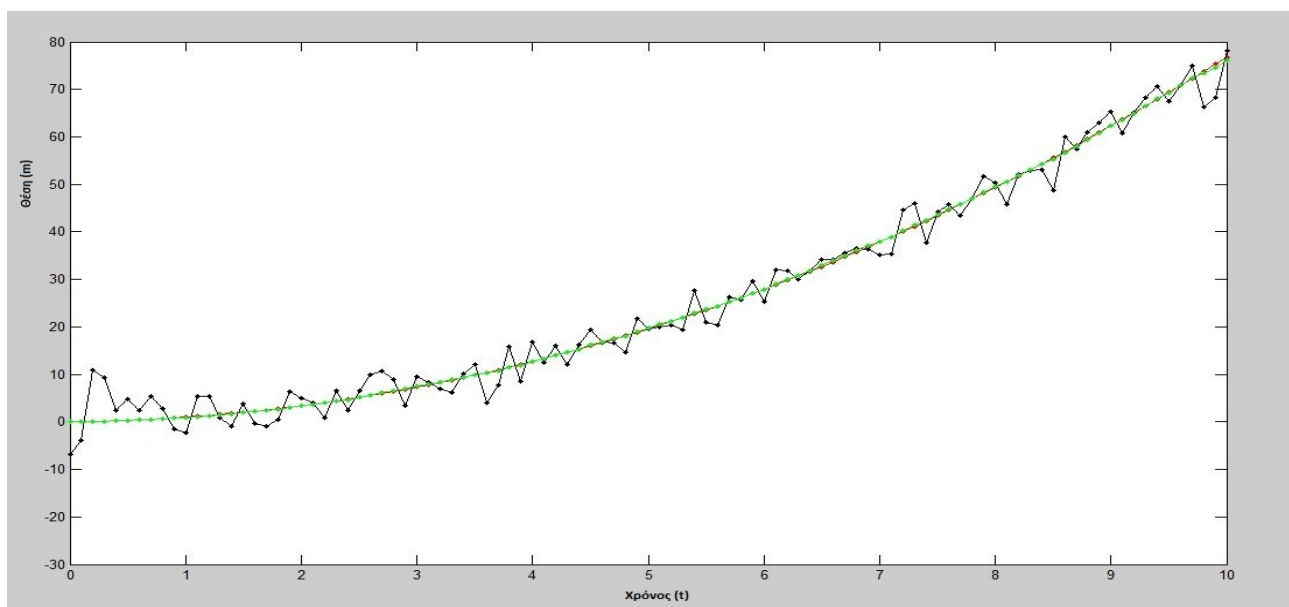
%Φιλτράρισμα με Kalman
Q_loc_estimate = [];
vel_estimate = [];
Q = [0; 0];
P_estimate = P;
P_mag_estimate = [];
predict_state = [];
predict_var = [];
for t = 1 : length(Q_loc)
    %Πρόβλεψη επόμενης κατάστασης διαδρομής
    Q_estimate = A * Q_estimate + B * u;
    predict_state = [predict_state; Q_estimate(1)];
    %Πρόβλεψη επόμενης covariance
    P = A * P * A' + Ex;
    predict_var = [predict_var; P];
    %Πρόβλεψη covariance μέτρησης
    %Κέρδος Kalman
    K = P * C' * inv(C * P * C' + Ez);
    Q_estimate = Q_estimate + K * (Q_loc_meas(t) - C * Q_estimate);
    %Ανανέωση εκτίμησης covariance
    P = (eye(2) - K * C) * P;
    %Στοιχεία για τις γραφικές μας
    Q_loc_estimate = [Q_loc_estimate; Q_estimate(1)];
    vel_estimate = [vel_estimate; Q_estimate(2)];
    P_mag_estimate = [P_mag_estimate; P(1)];
end

%Γραφικές αποτελεσμάτων
figure(2);
tt = 0 : dt : diarkeia;
plot(tt, Q_loc, '-r.', tt, Q_loc_meas, '-k.', tt, Q_loc_estimate, '-g.')
axis([0 10 -30 80])

```

Κώδικας στο MATLAB για την υλοποίηση ενός φίλτρου Kalman

Το γράφημα που παράγει ο κώδικας μας, δείχνει πως το “φίλτρο” μας, δημιουργεί μια καθαρή απόκριση παίρνοντας στην είσοδό του ένα σήμα το οποίο περιέχει σημαντικό “θόρυβο”.

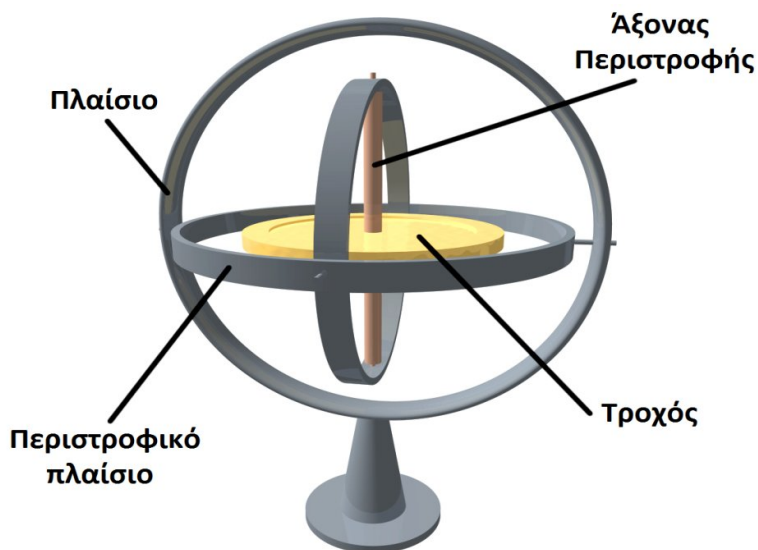


Εικόνα 9: Απόκριση ενός φίλτρου kalman (πράσινα στίγματα), μετά τις προβλέψεις (μαύρα στίγματα)

2.3 Εξισώσεις αισθητηρίων

2.3.1 Το Γυροσκόπιο

Το γυροσκόπιο είναι μια συσκευή μέτρησης και διατήρησης του προσανατολισμού, βασισμένη στις αρχές της στροφορμής. Μηχανικά, το γυροσκόπιο είναι ένας στρεφόμενος τροχός ή δίσκος, του οποίου ο άξονας περιστροφής είναι ελεύθερος κινηθεί προς οποιαδήποτε κατεύθυνση. Παρόλο που η κατεύθυνση αυτή δεν παραμένει σταθερή, αλλάζει εξαιτίας μιας εξωτερικής ροπής, πολύ μικρότερης και σε αντίθετη κατεύθυνση απ' ό τι θα άλλαζε χωρίς την μεγάλη στροφορμή που συνδέεται με τον υψηλό ρυθμό περιστροφής του τροχού και της αδράνειας. Η κατεύθυνση της συσκευής παραμένει σχεδόν σταθερή, ανεξάρτητα από την κίνηση της πλατφόρμας, επειδή βάζοντας την συσκευή σε ένα στρεφόμενο πλαίσιο, μειώνεται η εξωτερική ροπή.



Εικόνα 10: Δομή ενός απλού γυροσκοπίου

Εξισώση ενός γυροσκοπίου

Όπως κάθε μηχανικό σύστημα, η κίνηση ενός γυροσκοπίου μπορεί να γίνει πλήρως αντιληπτεί από την εφαρμογή του 2^{ου} νόμου του Νεύτωνα:

$$\vec{F} = m\vec{a}$$

σε όλα τα μέρη από τα οποία αποτελείται το γυροσκόπιο.

Είναι πολύ πιο αποτελεσματικό, να εκμεταλλευτούμαι το γεγονός ότι οι περισσότερες από τις δυνάμεις ασκούνται ανάμεσα στα μέρη του γυροσκοπίου, και έχουν ως αποτέλεσμα να κάνουν την συσκευή άκαμπτη. Η συνολική κίνηση περιγράφεται από την παρακάτω σχέση:

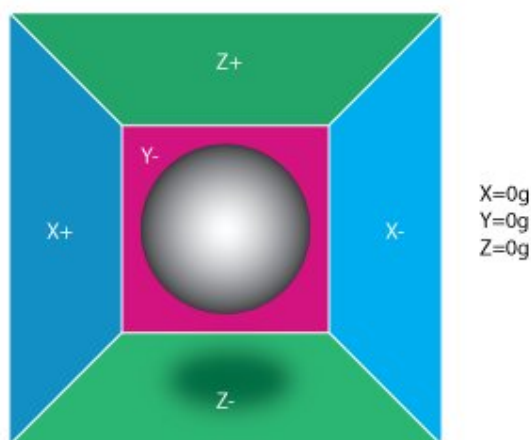
$$\vec{\tau} = \frac{d\vec{L}}{dt}$$

με: $\vec{\tau} = \vec{r} \times \vec{F}$, την ροπή που οφείλεται σε εξωτερικές δυνάμεις \vec{F} . Παρόλο που μοιάζει μια απλή εξίσωση, η ανάλυση των κινήσεων μπορεί να γίνει αρκετά πολύπλοκη.

2.3.2 Το επιταχυνσιόμετρο

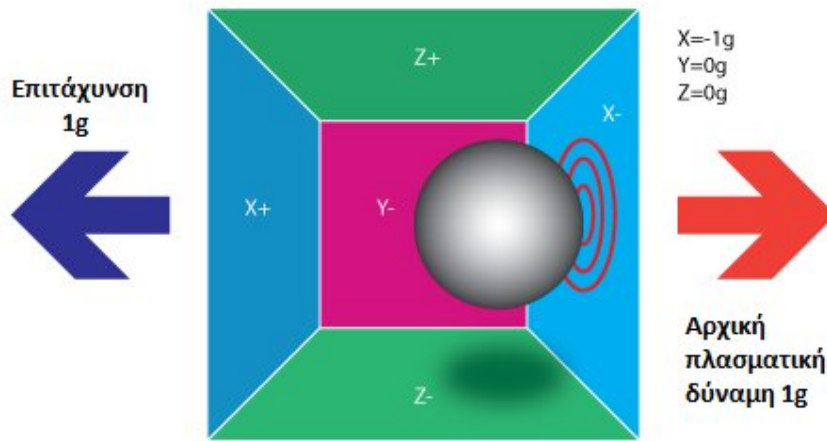
Το επιταχυνσιόμετρο είναι μια συσκευή, που όπως φανερώνει και το όνομά της, μετράει την επιτάχυνση. Η επιτάχυνση που μετράται από ένα επιταχυνσιόμετρο δεν είναι απαραίτητα ισοδύναμη με τον ρυθμό μεταβολής της ταχύτητας. Ένας άλλος τύπος επιτάχυνσης που μπορεί να μετρήσει ένα επιταχυνσιόμετρο είναι η επιτάχυνση λόγω της βαρύτητας.

Για να μπορέσουμε να κατανοήσουμε την δομή ενός επιταχυνσιομέτρου, μπορούμε να φανταστούμε έναν κύβο, ο οποίος στο εσωτερικό του, κρύβει μια σφαίρα.



Εικόνα 11: Δομή ενός επιταχυνσιομέτρου

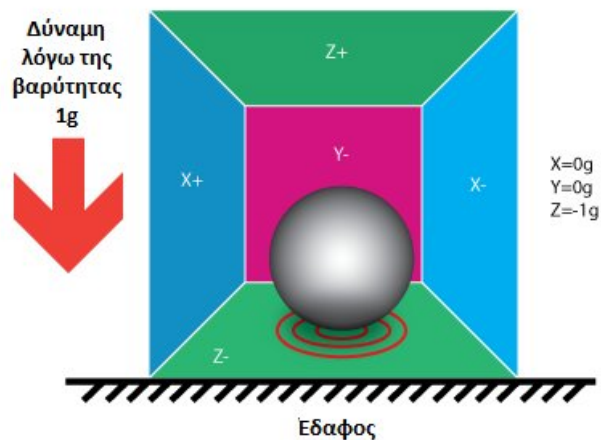
Αν αφήσουμε το επιταχυνσιόμετρο σε ηρεμία η μπάλα που βρίσκεται στο εσωτερικού του κύβου, θα βρίσκεται στο κέντρο του, και θα αιωρείται. Φανταστείτε λοιπόν, πως καθένας από τους τοίχους είναι ευαίσθητος στην πίεση. Αν εμείς, κινήσουμε ξαφνικά τον κύβο μας, αριστερά, το επιταχύνουμε με μια επιτάχυνση $g = 9.81 \text{ m/s}^2$, και στην συνέχεια η σφαίρα μας χτυπάει στον τοίχο X-. Στην συνέχεια, μετράμε την πίεση που ασκεί η μπάλα στον τοίχο και εξάγουμε μια τιμή του τύπου $-1g$ στον άξονα X.



Εικόνα 12: Το επιταχυνσιόμετρο κατά την κίνησή του προς τα αριστερά

Παρατηρούμε πως το επιταχυνσιόμετρο ουσιαστικά ανιχνεύει μια δύναμη η οποία είναι αντίθετη σε κατεύθυνση από το διάνυσμα της επιτάχυνσής μας. Αυτή η δύναμη ονομάζεται **αρχική δύναμη** ή **πλασματική δύναμη**. Ένα επιταχυνσιόμετρο, λοιπόν, μετράει την επιτάχυνση έμμεσα μέσω της δύναμης αυτής που εφαρμόζεται σε έναν από τους τοίχους (εξαρτάται και από τον τύπο του επιταχυνσιομέτρου). Η δύναμη αυτή, δημιουργείται από την επιτάχυνση, αλλά όπως θα δούμε παρακάτω δεν δημιουργείται πάντοτε λόγω αυτής.

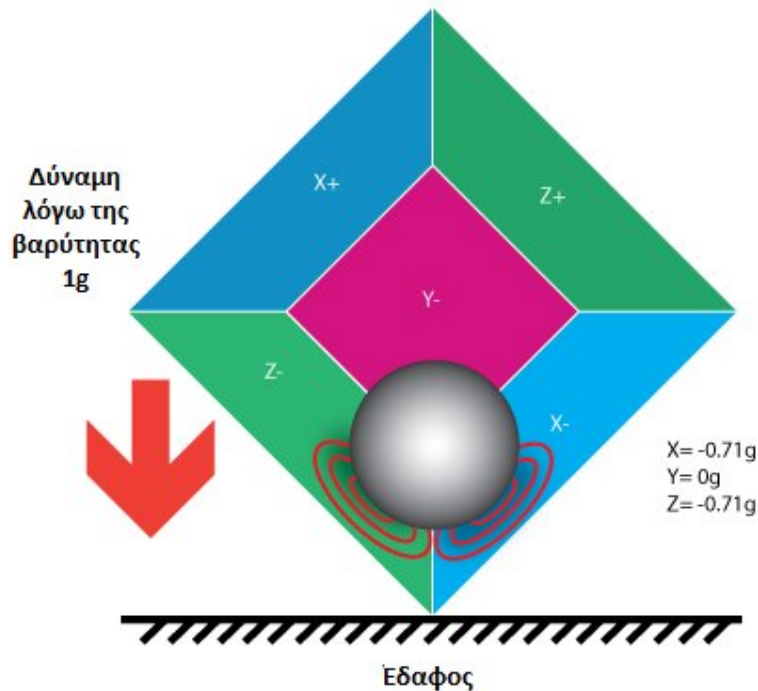
Αν ακουμπήσουμε την συσκευή μας στο έδαφος, η σφαίρα που βρίσκεται στο εσωτερικό της, θα πέσει κάτω (στον τοίχο Z-) και θα εφαρμόσει μια δύναμη 1g σε αυτόν.



Εικόνα 13: Το επιταχυνσιόμετρο και η αντίδραση του όταν το ακουμπάμε στο έδαφος

Στην περίπτωση αυτή ο κύβος μας δεν κινείται αλλά παρόλα αυτά παίρνουμε την τιμή $-1g$ στον άξονα Z. Η πίεση που εφάρμοσε η σφαίρα πάνω στον τοίχο προκλήθηκε εξαιτίας της βαρύτητας. Με το παράδειγμα αυτό, θέλουμε να δείξουμε πως ουσιαστικά το επιταχυνσιόμετρο μετράει δύναμη και **όχι** επιτάχυνση. Απλώς τυχαίνει η επιτάχυνση να δημιουργεί μια εσωτερική δύναμη στην συσκευή μας, η οποία ανιχνεύεται από τον μηχανισμό ανίχνευσης δύναμης του επιταχυνσιομέτρου.

Μέχρι στιγμής αναλύσαμε την έξοδο ενός επιταχυνσιομέτρου σε έναν μόνο άξονα. Η πραγματική τιμή ενός επιταχυνσιομέτρου τριών αξόνων, προέρχεται από το γεγονός ότι μπορεί να ανιχνεύσει αρχικές δυνάμεις και στους τρεις του άξονες.

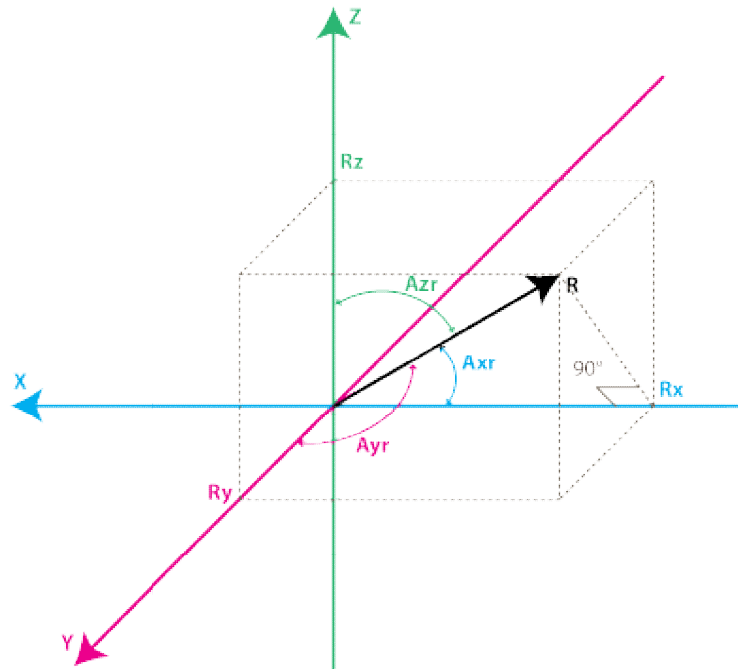


Εικόνα 14: Το επιταχυνσιόμετρο με άσκηση δύναμης σε δυο από τους άξονές του

Οι τιμές 0.71 δεν είναι αυθαίρετες, στην πραγματικότητα είναι μια προσέγγιση της τιμής $\sqrt{\frac{1}{2}}$.

Εξισώσεις ενός επιταχυνσιομέτρου

Με την βοήθεια της παρακάτω εικόνας, θα κατανοήσουμε τους υπολογισμούς.



Εικόνα 15: Άξονες ενός επιταχυνσιομέτρου

Το διάνυσμα R που φαίνεται στην εικόνα, είναι το διάνυσμα της δύναμης που μετράει το επιταχυνσιόμετρο. Τα R_x , R_y , R_z είναι οι προβολές του διανύσματος R στους άξονες X, Y, Z για τα οποία ισχύει η κάτωθι σχέση:

$$R^2 = R_x^2 + R_y^2 + R_z^2 \quad (1)$$

Όμως, οι γωνίες οι οποίες μας ενδιαφέρουν είναι οι γωνίες μεταξύ των αξόνων X, Y, Z και του διανύσματος R . Τις προσδιορίζουμε ως A_{xr} , A_{yr} , A_{zr} . Από το ορθογώνιο τρίγωνο μεταξύ των διανυσμάτων R , R_x και της κοινής τους πλευράς μπορούμε να βρούμε τις τιμές των γωνιών αυτών.

$$\cos(A_{xr}) = \frac{R_x}{R}, \quad \cos(A_{yr}) = \frac{R_y}{R}, \quad \cos(A_{zr}) = \frac{R_z}{R}$$

Από την (1):
$$R = \sqrt{R_x^2 + R_y^2 + R_z^2}$$

$$A_{xr} = \arccos\left(\frac{R_x}{R}\right), \quad A_{yr} = \arccos\left(\frac{R_y}{R}\right), \quad A_{zr} = \arccos\left(\frac{R_z}{R}\right)$$

Το επιταχυνσιόμετρο σε συνδυασμό με το γυροσκόπιο παρέχουν πιο ακριβείς εκτιμήσεις κλίσης.

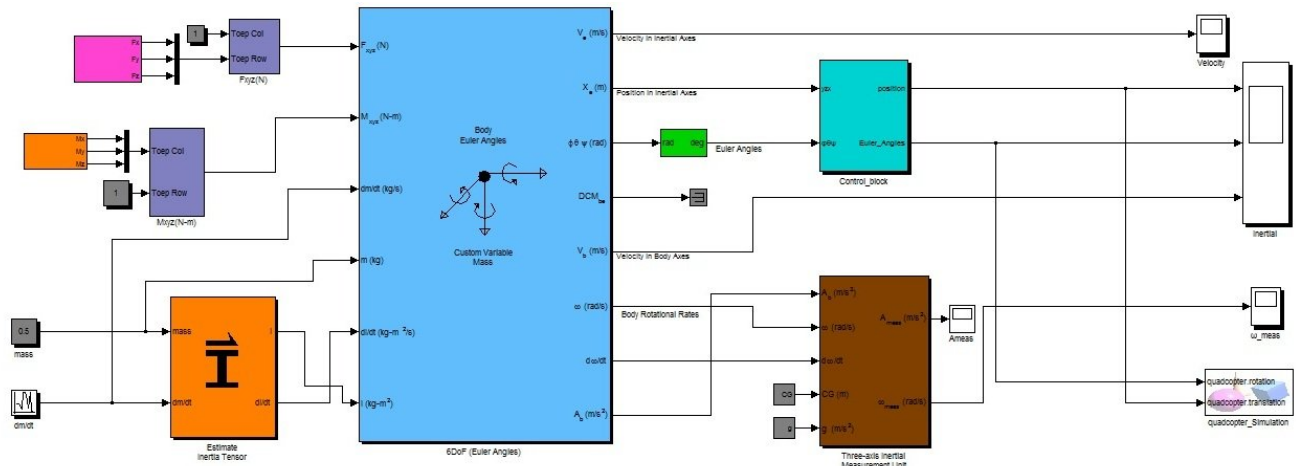
Κεφάλαιο 3: Μοντελοποίηση συστημάτων στο MATLAB – Simulink

Στο κεφάλαιο αυτό θα γίνει παρουσίαση των δύο μοντέλων που κατασκευάστηκαν στο προγραμματιστικό περιβάλλον του MATLAB – Simulink. Το πρώτο μοντέλο μας δημιουργήθηκε παράλληλα με την μελέτη της θεωρίας και χρησιμοποιήθηκαν αρκετά έτοιμα block, τα οποία μας παρείχε το περιβάλλον του Simulink προκειμένου να βλέπουμε και στην πράξη, την συμπεριφορά ενός τέτοιου συστήματος. Αντίθετα, το δεύτερο μοντέλο μας, δημιουργήθηκε με κεντρική βάση τις διαφορικές εξισώσεις που περιγράφουν ένα τέτοιο σύστημα σε συνδυασμό με δικές μας επεμβάσεις προκειμένου να το προσαρμόσουμε για την δική μας μοντελοποίηση.

3.1 Πρώτο μοντέλο quadcopter

3.1.1 Παρουσίαση του μοντέλου

Το μοντέλο αυτό αποτελείται από αρκετά έτοιμα block που MATLAB – Simulink, αλλά η υλοποίηση του μοντέλου και ο έλεγχος είναι δικής μας επιμέλειας. Το μοντέλο αυτό, όπως έχουμε αναφέρει και στην εισαγωγή του κεφαλαίου μας, δημιουργήθηκε παράλληλα με την μελέτη του τρόπου λειτουργίας ενός quadcopter και των εξισώσεων του. Αυτό έγινε για να βλέπουμε και στην πράξη την λειτουργία του, και επίσης αποκτήσαμε μια πολύ καλή πρώτη εμπειρία μοντελοποίησης προκειμένου να υλοποιήσουμε, αργότερα την δική μας, με δικά μας block. Ξεκινώντας την περιγραφή του μοντέλου αυτού, θα παρουσιάσουμε την εικόνα του μοντέλου μας συνολικά:



Εικόνα 16: Το πρώτο μας μοντέλο

Τα block που πήραμε έτοιμα από το MATLAB – Simulink και φαίνονται στην παραπάνω εικόνα, είναι το κεντρικό μας block που υλοποιεί τις εξισώσεις ενός UAV (με γαλάζιο χρώμα). Το block αυτό έχει 6 εισόδους και 8 εξόδους. Οι 6 εισοδοι είναι: 1) οι δυνάμεις που δέχεται ένα UAV στους άξονές του(x, y, z), 2) οι ροπές που αναπτύσσει ένα UAV στους άξονές του(x, y, z), 3) ο ρυθμός μεταβολής της μάζας ενός UAV, 4) την μάζα ενός UAV, 5) ο ρυθμός μεταβολής του πίνακα αδράνειας ενός UAV και 6) ο πίνακας αδράνειας ενός UAV. Οι 8 έξοδοι του block αυτού αναφέρονται στην θέση και την ταχύτητα με αναφορά την Γη (ή σημείο εκκίνησης), τις γωνίες του Euler σε rad, έναν πίνακα 3x3 που μετασχηματίζει τις συντεταγμένες του UAV μας από το σημείο εκκίνησης στο σημείο που βρίσκεται το UAV μας, την γραμμική και γωνιακή ταχύτητα του μοντέλου μας, και τέλος την γραμμική και γωνιακή επιτάχυνση του μοντέλου μας.

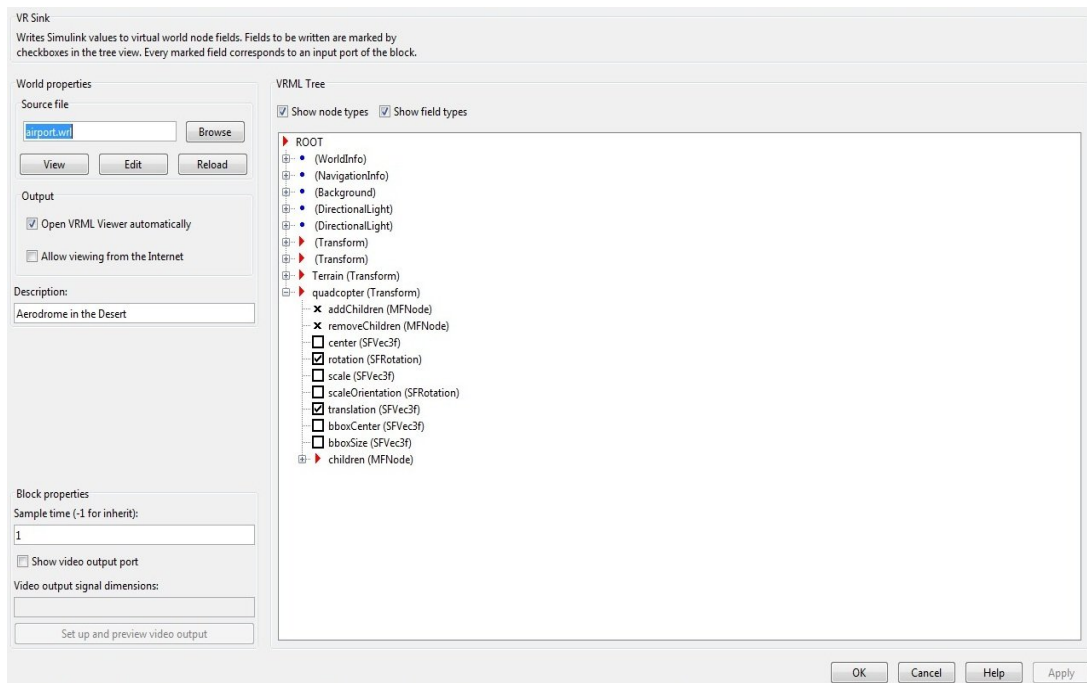
Στην παραπάνω εικόνα παρατηρούμε πως για τις εισόδους του ρυθμού μεταβολής της αδράνειας και της αδράνειας χρησιμοποιούμε ένα block. Το block αυτό είναι έτοιμο, και δίνοντάς του στην είσοδο τον ρυθμό μεταβολής της μάζας, και την μάζα, αυτό κάνει μια εκτίμηση για τον ρυθμό αδράνειας και την αδράνεια του μοντέλου

μας. Επίσης, οι εισοδοι των δυνάμεων και των ροπών υλοποιήθηκαν από συναρτήσεις τυχαίων τιμών.

Το επόμενο block που χρησιμοποιήσαμε, είναι αυτό των αισθητηρίων. Στην εικόνα μας, το block αυτό έχει καφέ χρώμα και ουσιαστικά υλοποιεί γυροσκόπιο τριών αξόνων και επιταχυνσιόμετρο τριών αξόνων. Στην είσοδό του δέχεται 3 από τις 8 εξόδους του μοντέλου μας. Την γραμμική και γωνιακή επιτάχυνση και την γωνιακή ταχύτητα του μοντέλου μας. Επίσης, ως εισόδους παίρνει την επιτάχυνση της βαρύτητας και το κέντρο μάζας του μοντέλου μας. Στις εξόδους του μας δίνει την γωνιακή ταχύτητα και επιτάχυνση που υπολογίζουν τα αισθητήρια.

Στην συνέχεια υπάρχει το block με τίτλο “control_block” το οποίο έχει δημιουργηθεί για τον έλεγχο των γωνιών του Euler και της θέσης του μοντέλου μας και η λεπτομερέστερη ανάλυση θα γίνει στην επόμενη ενότητα.

Τέλος, υπάρχει ένα block το οποίο θα μας βοηθήσει να παρατηρήσουμε και γραφικά την πορεία ενός quadcopter στον χώρο. Το block αυτό μας δίνει την δυνατότητα να δημιουργήσουμε ένα περιβάλλον εικονικής πραγματικότητας στο οποίο θα εντάξουμε το μοντέλο μας. Μας δίνεται η δυνατότητα να επιλέξουμε εμείς τις εισόδους του block αναλόγως με το τι θέλουμε να αποδώσουμε γραφικά. Στην παρακάτω εικόνα βλέπουμε την ρύθμιση που έχουμε κάνει για την περίπτωση μας.



Εικόνα 17: Ρυθμίσεις για το περιβάλλον εικονικής πραγματικότητας

3.1.2 Ανάλυση του μοντέλου

Στην ενότητα αυτή θα ασχοληθούμε με το block “control_block” και την εσωτερική του δομή, για το πως θα γίνει ο έλεγχος του μοντέλου μας. Το block αυτό δημιουργήθηκε προκειμένου να μπαίνουν καθαρά τα δεδομένα μας μέσα στο περιβάλλον της εικονικής πραγματικότητας για να παρατηρήσουμε την συμπεριφορά του quadcopter μας.

Πριν προχωρήσουμε όμως στην επεξήγηση του block μας, θα ήθελα να αναφερθεί στο σημείο αυτό πως μέσα στον χώρο εικονικής πραγματικότητας, δεν υπήρχε κάποιο έτοιμο quadcopter. Έτσι, με την βοήθεια του autoCAD σχεδιάστηκε ένα τέτοιο μοντέλο και με τον παρακάτω κώδικα που γράφτηκε στο περιβάλλον του MATLAB δημιουργήσαμε την εικονική μας πραγματικότητα.

```
stl2vrml('quadcopter.stl'); %Μετατρέπει τα αρχεία .stl σε αρχεία VRML (.wrl)

%Δημιουργούμε το σχέδιό μας σαν όχημα εικονικής πραγματικότητας
quadcopter = Aero.VirtualRealityAnimation;

%Φόρτωση του χώρου εικονικής πραγματικότητας
quadcopter.VRWorldFilename = ['C:\Users\Aggelos\Documents\MATLAB\airport.wrl'];

%Δημιουργία αντίγραφου του αρχείου μας
copyfile(quadcopter.VRWorldFilename, [tempdir, 'quadcopter.wrl'], 'f');

%Φόρτωση του σχεδίου μας
quadcopter.VRWorldFilename = [tempdir, 'quadcopter.wrl'];

%Δημιουργία και συμπλήρωση του αντικειμένου εικονικής πραγματικότητας
quadcopter.initialize();

%Εισαγωγή του σχεδίου μας στο περιβάλλον εικονικής πραγματικότητας
quadcopter.addNode('quadcopter', 'C:\Users\Aggelos\Documents\MATLAB\quadcopter.wrl');
```

Κώδικας εισαγωγής του σχεδίου μας στο MATLAB και στην συνέχεια δημιουργίας εικονικής πραγματικότητας με το σχέδιό μας εσωτερικά αυτού



Εικόνα 18: Σχέδιο ενός quadcopter στο AutoCAD

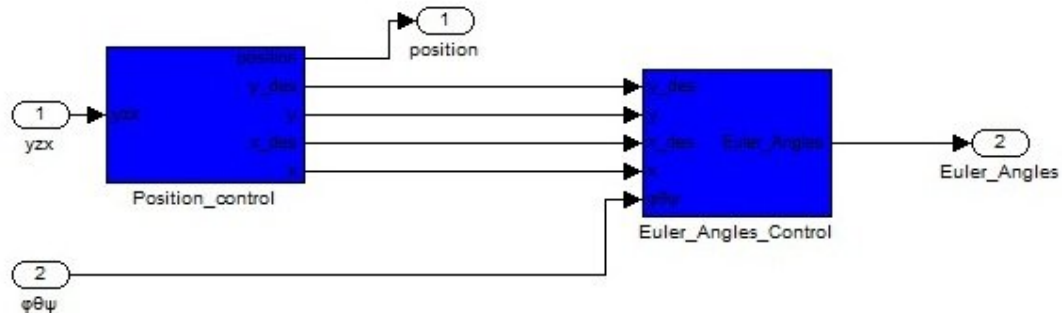
“Τρέχοντας” λοιπόν τον κώδικα αυτόν το αποτέλεσμα στο περιβάλλον εικονικής πραγματικότητας είναι το εξής:



Εικόνα 19: Περιβάλλον εικονικής πραγματικότητας με το σχεδιασμένο μοντέλο μας

Εφόσον παρουσιάστηκε το μοντέλο μας έτοιμο μέσα στο περιβάλλον εικονικής πραγματικότητας, είμαστε έτοιμοι να προχωρήσουμε στον έλεγχο αυτού, για να παρατηρήσουμε την συμπεριφορά του. Το block είναι υλοποιημένο εξ'ολοκλήρου

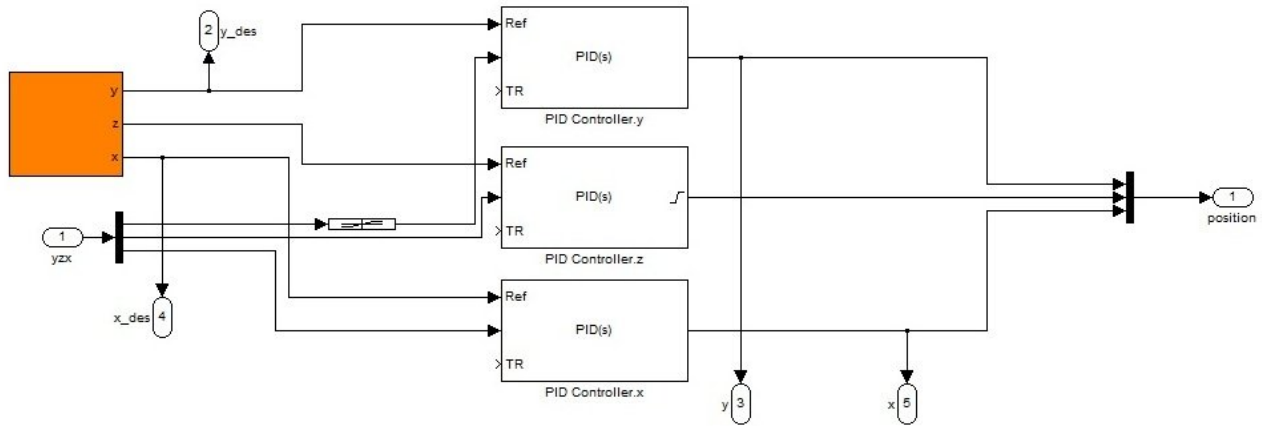
από την αρχή και χτισμένο για τον έλεγχο των γωνιών και της θέσης του μοντέλου μας στον χώρο. Παρακάτω θα δούμε το εσωτερικό του block μας.



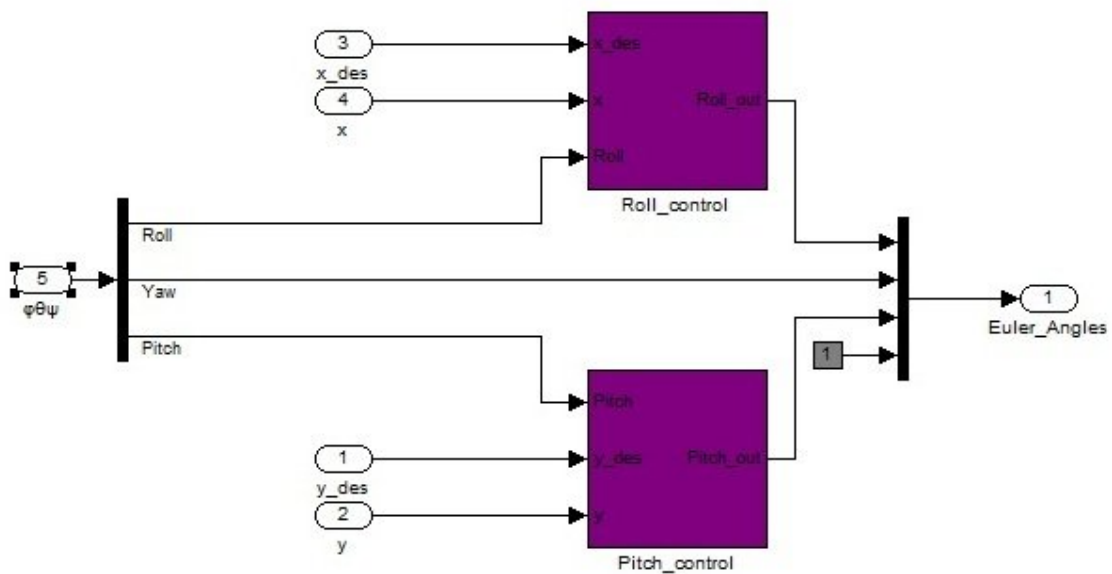
Εικόνα 20: Το block ελέγχου των γωνιών και της θέσης του μοντέλου μας

Στις εισόδους του block αυτού βλέπουμε τις τιμές των τριών συντεταγμένων μας, x , y , z και τις τιμές των γωνιών του Euler. Οι τιμές αυτές χωρίζονται σε άλλα δυο block τα οποία πραγματοποιούν έλεγχο, το καθένα για την δική του ομάδα μεταβλητών (το block με τίτλο `Position_control` δέχεται ως εισόδους τις τιμές των συντεταγμένων, ενώ το άλλο με τίτλο `Euler_Angles_Control` δέχεται ως εισόδους τις τιμές των γωνιών Euler). Όπως θα παρατηρήσατε από την εικόνα, το block με τίτλο `Position_control` έχει πέντε εξόδους, από τις οποίες οι τέσσερις γίνονται είσοδοι του block με τίτλο `Euler_Angles_Control` και συνολικά πέντε με την προσθήκη των γωνιών μας. Εννοείται πως και τα δυο block έχουν εξόδους με τις συντεταγμένες και τις γωνίες μας.

Πηγαίνοντας στο εσωτερικό των δυο αυτών block θα δούμε τις διαδικασίες ελέγχου που πραγματοποιούνται προκειμένου να δούμε το μοντέλο μας να κινείται με τον σωστό τρόπο.



Εικόνα 21: Η διαδικασία που εκτελείται εσωτερικά του block “Position_control”



Εικόνα 22: Η διαδικασία που επιτελείται εσωτερικά του block “Euler_Angles_Control”

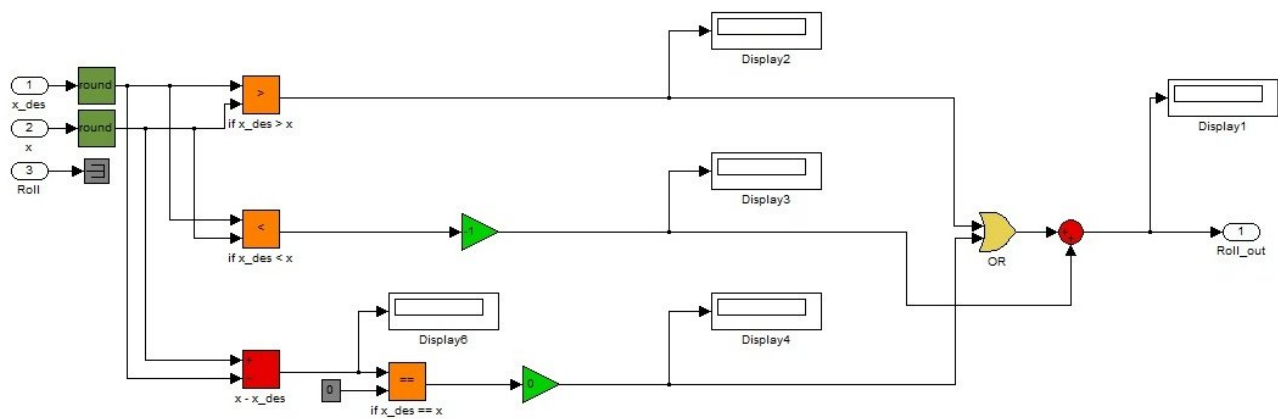
Στο block της εικόνας 21, βλέπουμε ένα block με πορτοκαλί χρώμα, μέσα στο οποίο δηλώνουμε τις τιμές των επιθυμητών συντεταγμένων μας. Έχουμε θέσει σε εφαρμογή 3 PID ελεγκτές, έναν για κάθε συντεταγμένη, οι οποίοι δέχονται στις εισόδους τους 2 τιμές. Η πρώτη τιμή προέρχεται από το μοντέλο μας και η δεύτερη τιμή προέρχεται από εμάς. Η λειτουργία ενός τέτοιου ελεγκτή έχει απότερω σκόπο, να

κάνει ίση την τιμή που έρχεται από το μοντέλο μας, με αυτήν που θα θέσουμε εμείς ως επιθυμητή. Στο εσωτερικό του PID αυτού, έχουμε θέσει τα παρακάτω κέρδη:

$$K_p = 0.4 , K_i = 1 , K_D = 0$$

Ουσιαστικά έχουμε εφαρμόσει έλεγχο PI, και οι τιμές αυτές στα κέρδη επιλέχθηκαν μέσω δοκιμών. Αλλαγές στα κέρδη, ουσιαστικά επηρεάζουν την ταχύτητα ταύτισης της τιμής που μπαίνει στον ελεγκτή με την επιθυμητή μας, την αποτελεσματικότητα επίτευξης και διατήρησης της επιθυμητής μας εισόδου. Στην δική μας ρύθμιση, “τρέχοντας” το μοντέλο μας, βλέπουμε πως αλλάζοντας μια από τις συντεταγμένες, η τιμή εισόδου αυξάνεται προοδευτικά μέχρι που φτάνει την επιθυμητή μας τιμή, και σταθεροποιείται εκεί. Μια τέτοια ρύθμιση, μας ικανοποιεί ιδιαιτέρως γιατί δεν μας απασχολεί τόσο η ταχύτητα μετάβασης στην επιθυμητή μας θέση, όσο η διατήρηση της ισορροπίας του quadcopter μας κατά την διαδρομή του προς την νέα μας θέση, και η διατήρηση της θέσης αυτής.

Στο άλλο block της εικόνας 22, βλέπουμε τις γωνίες μας να “σπάνε” σε τρεις διαφορετικές γωνίες (roll, pitch, yaw) και δέχονται μια επεξεργασία πριν ενωποιηθούν εκ νέου και οδηγηθούν στην έξοδο του block μας. Όπως θα παρατηρήσατε η μόνη από τις γωνίες που δεν δέχεται επεξεργασία είναι η γωνία Yaw του άξονα z και αυτό γιατί στην λειτουργία των αντικριστών μοτέρ (M1-M3) εφαρμόζεται φορά περιστροφής αντίθετη από αυτή των μοτέρ (M2-M4). Οπότε οι γωνίες που θα χρειαστούν έλεγχο είναι η γωνία roll και η γωνία pitch. Στην εικόνα 21 βλέπουμε δύο block με μωβ χρώμα. Το ένα αφορά την γωνία roll (και ως εισόδους παίρνει τις δυο συντεταγμένες για το x) και το άλλο την γωνία pitch (και ως εισόδους παίρνει τις δυο συντεταγμένες για το y). Στις εικόνες που θα ακολουθήσουν θα δούμε το εσωτερικό του κάθε block.



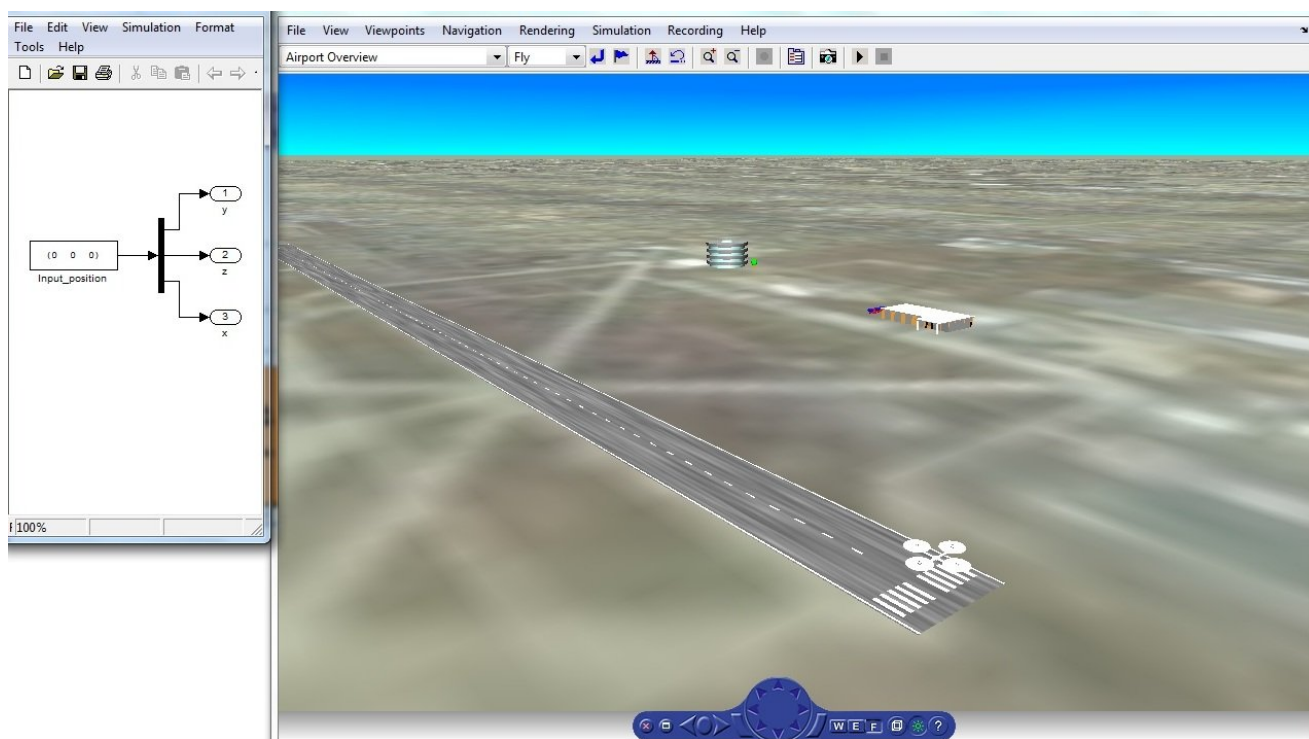
Εικόνα 23: Η διαδικασία ελέγχου της γωνίας Roll και της γωνίας Pitch

Ο έλεγχος της γωνίας στον άξονα x και στον άξονα y γίνεται με την διαδικασία που φαίνεται στην παραπάνω εικόνα. Δηλαδή τα block μας παίρνουν ως είσοδο τις τιμές των συντεταγμένων για το x και το y (επιθυμητές και πραγματικές) και στην συνέχεια επιτελείται μια διεγασρία με συνολικά τρία αποτελέσματα.

1. Στην περίπτωση που η επιθυμητή μας τιμή είναι μεγαλύτερη από την πραγματική, τότε παράγεται ένας παλμός ο οποίος περνάει από μια πύλη OR για να αποφευχθούν αστοχίες, και στην συνέχεια οδηγείτε στην έξοδο
2. Στην περίπτωση τώρα που η επιθυμητή μας τιμή είναι μικρότερη από την πραγματική, τότε παράγεται ένας αρνητικός παλμός ο οποίος περνάει προστίθεται στο αποτέλεσμα της πύλης OR.
3. Τρίτη και τελευταία περίπτωση είναι η επιθυμητή με την πραγματική τιμή να είναι ίσες που σημαίνει πως το quadcopter θα βρίσκεται στην θέση που εμείς θέλουμε. Στην περίπτωση αυτή λοιπόν, αν υπάρχει οποιοσδήποτε παλμός, αυτός μηδενίζεται προκειμένου το quadcopter να σταθεροποιηθεί στην θέση που του έχουμε δώσει, περιμένοντας έναν νέο προορισμό.

3.1.3 Αποτελέσματα προσομοίωσης

“Τρέχοντας” λοιπόν το μοντέλο μας είμαστε έτοιμοι να παρατηρήσουμε την συμπεριφορά του. Κατά την διάρκεια της προσομοίωσης βλέπουμε την παρακάτω εικόνα:



Εικόνα 24: Εικόνα κατά την προσομοίωση του μοντέλου μας

Το αριστερό μικρό παράθυρο, είναι η δήλωση της επιθυμητής κατεύθυνσης και το δεξί παράθυρο περιέχει το μοντέλο μας, στο περιβάλλον εικονικής πραγματικότητας. Δίνοντας εμείς επιθυμητές τιμές, λοιπόν, για τις συντεταγμένες μας, θα παρατηρήσουμε την πορεία του μοντέλου μας. Οι πιθανές περιπτώσεις βασικής κίνησης είναι οι εξής:

- Άξονας z: Η κίνηση του quadcopter μας στον άξονα z, ορίζει το υψόμετρο στο οποίο θα φτάσει το μοντέλο μας. Δίνοντας λοιπόν, τιμή στο z παρατηρούμε το μοντέλο μας να παίρνει ή να χάνει ύψος. Με τον σημαντικό περιορισμό του να μην μπορούμε να δώσουμε τιμή μικρότερη του μηδενός, δεδομένου ότι το μηδέν, ορίζει το έδαφος στο οποίο ακουμπά το μοντέλο μας.
- Άξονας y: Η κίνηση του quadcopter μας στον άξονα y, γίνεται με μια κλίση προς τα μπροστά ή προς τα πίσω αναλόγως προς τα που θέλουμε να το κατευθυνθούμε. Σύμφωνα με τον έλεγχο που πραγματοποιείται στην εικόνα 22, όταν εμείς δώσουμε τιμή μεγαλύτερη από αυτήν που βρίσκεται το μοντέλο μας, αυτό θα πάρει κλίση προς τα μπροστά ενώ αν του δώσουμε θέση μικρότερη από αυτήν που βρίσκεται, αυτό θα πάρει κλίση προς τα πίσω.
- Άξονας x: Η κίνηση του quadcopter μας στον άξονα x, γίνεται με μια κλίση προς τα αριστερά ή προς τα δεξιά αναλόγως προς τα που θέλουμε να το κατευθυνθούμε. Σύμφωνα με τον έλεγχο που πραγματοποιείται στην εικόνα 22, όταν εμείς δώσουμε τιμή μεγαλύτερη από αυτήν που βρίσκεται το μοντέλο μας, αυτό θα πάρει κλίση προς τα αριστερά ενώ αν του δώσουμε θέση μικρότερη από αυτήν που βρίσκεται, αυτό θα πάρει κλίση προς τα δεξιά.

Πέρα όμως από τις βασικές αυτές περιπτώσεις έχουμε και περιπτώσεις σύνθετης κίνησης. Δηλαδή κίνηση του quadcopter μας σε δυο από τους τρεις του άξονες. Χωρίζοντας λοιπόν την κίνηση στους άξονες x και y υπάρχουν οι εξής τέσσερις πιθανές περιπτώσεις σύνθετης κίνησης.

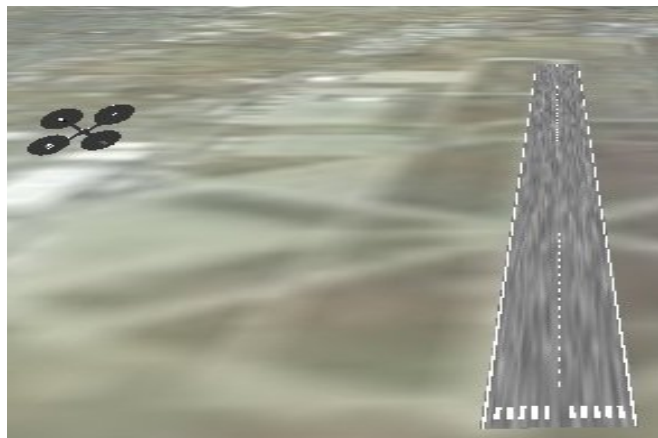
1. 1^ο τεταρτημόριο: Η κίνηση αυτή πραγματοποιείται δίνοντάς στις συντεταγμένες x, y θετικές τιμές. Με την είσοδο των δύο αυτών

συντεταγμένων τα μοτέρ που βρίσκονται από την μεριά του τεταρτημορίου αυτού, ελαττώνουν την ταχύτητά τους και αυξάνεται αυτή των άλλων δυο. Το αποτέλεσμα μιας τέτοιας περίπτωσης φαίνεται στο παρακάτω στιγμίοτυπο:



Εικόνα 25: Στιγμίοτυπο προσομοίωσης παράλληλης κίνησης στους άξονες x , y (για x , y θετικά)

2. 2^ο τεταρτημόριο: Η κίνηση αυτή πραγματοποιείται δίνοντάς στην συντεταγμένη x αρνητική τιμή και στην y θετική τιμή. Με την είσοδο των δύο αυτών συντεταγμένων τα μοτέρ που βρίσκονται από την μεριά του τεταρτημορίου αυτού, ελαττώνουν την ταχύτητά τους και αυξάνεται αυτή των άλλων δυο. Το αποτέλεσμα μιας τέτοιας περίπτωσης φαίνεται στο παρακάτω στιγμίοτυπο:



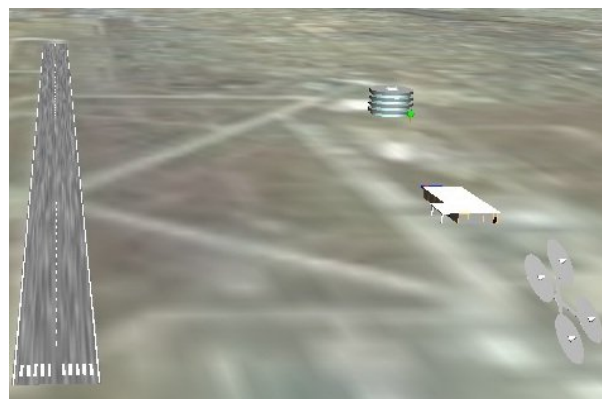
Εικόνα 26: Στιγμίοτυπο προσομοίωσης παράλληλης κίνησης στους άξονες x , y (για x αρνητικό, y θετικό)

3. 3^ο τεταρτημόριο: Η κίνηση αυτή πραγματοποιείται δίνοντάς στις συντεταγμένες x , y αρνητικές τιμές. Με την είσοδο των δύο αυτών συντεταγμένων τα μοτέρ που βρίσκονται από την μεριά του τεταρτημορίου αυτού, ελαττώνουν την ταχύτητά τους και αυξάνεται αυτή των άλλων δυο. Το αποτέλεσμα μιας τέτοιας περίπτωσης φαίνεται στο παρακάτω στιγμίοτυπο:



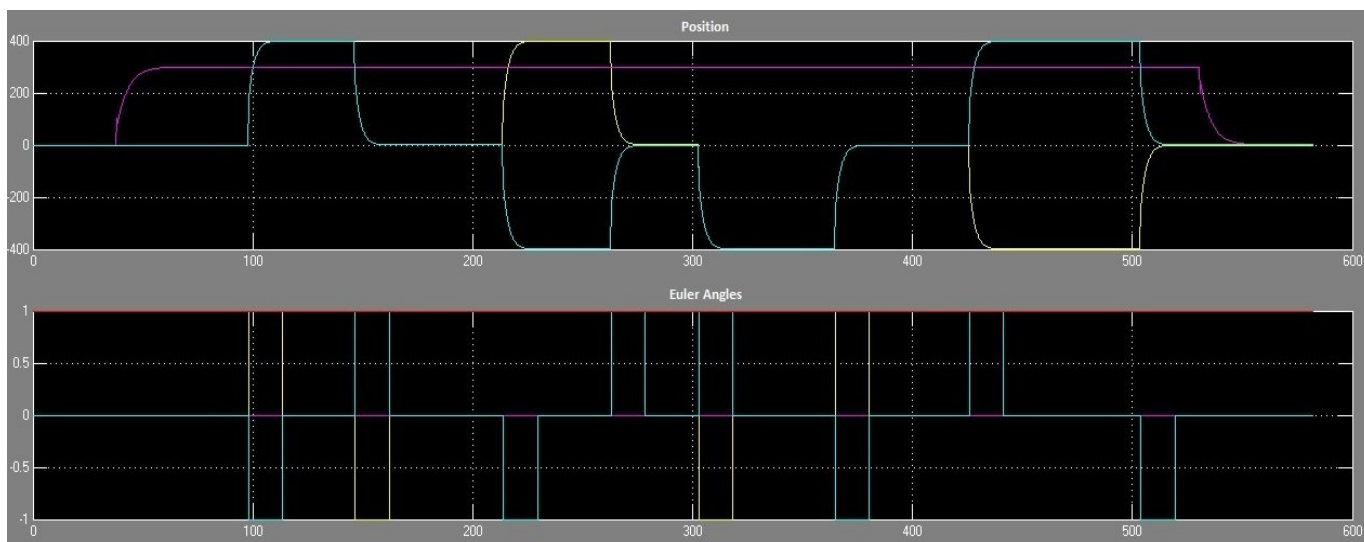
Εικόνα 27: Στιγμίοτυπο προσομοίωσης παράλληλης κίνησης στους άξονες x , y (για x , y αρνητικά)

4. 4^ο τεταρτημόριο: Η κίνηση αυτή πραγματοποιείται δίνοντάς στην συντεταγμένη x θετική τιμή και στην y αρνητική τιμή. Με την είσοδο των δύο αυτών συντεταγμένων τα μοτέρ που βρίσκονται από την μεριά του τεταρτημορίου αυτού, ελαττώνουν την ταχύτητά τους και αυξάνεται αυτή των άλλων δυο. Το αποτέλεσμα μιας τέτοιας περίπτωσης φαίνεται στο παρακάτω στιγμίοτυπο:



Εικόνα 28: προσομοίωσης παράλληλης κίνησης στους άξονες x , y (για x θετικό, y αρνητικό)

Για την υποστήριξη της λειτουργίας αυτής παίρνουμε και κάποια γραφήματα προκειμένου να δούμε και τις τιμές που παίρνουμε από κάθε κίνηση. Το γράφημα με τα δεδομένα που θέλουμε να παρατηρήσουμε ακολουθεί:



Εικόνα 29: Γραφήματα συμπεριφοράς του μοντέλου μας

Πρόκειται για τα γραφήματα των συντεταγμένων μας και των γωνιών μας. Το επάνω γράφημα αναφέρεται στις συντεταγμένες μας, και το αμέσως από κάτω στις γωνίες μας.

Στο πρώτο γράφημα παρατηρούμε πόσο ομαλά μεταβαίνει το μοντέλο μας στην επιθυμητή μας θέση, στην περίπτωση κίνησης στον άξονα x (γαλάζια γραμμή), στον άξονα y (κίτρινη γραμμή) και στον άξονα z (μωβ γραμμή). Στην προσομοίωσή μας έχουμε κάνει την εξής διαδρομή. Ανύψωση από το έδαφος, παράλληλη κίνηση στους άξονες x, y (1^ο τεταρτημόριο), επιστροφή αρχική θέση, παράλληλη κίνηση στους άξονες x, y (2^ο τεταρτημόριο), επιστροφή αρχική θέση, παράλληλη κίνηση στους άξονες x, y (3^ο τεταρτημόριο), επιστροφή αρχική θέση, παράλληλη κίνηση στους άξονες x, y (4^ο τεταρτημόριο), επιστροφή στην αρχική θέση και τέλος προσγείωση. Η διαδρομή αυτή πραγματοποιήθηκε για να διαπιστώσουμε αν το μοντέλο μας μπορεί να ανταποκριθεί σε πολύπλοκες διαδρομές. Οι αποκρίσεις σε συνδυασμό με την

προσομοίωση στην εικονική πραγματικότητα μας δείχνουν πως το μοντέλο μας αποκρίνεται αρκετά καλά. Το μόνο που ίσως μας απασχολεί είναι η αστοχία της εικονικής πραγματικότητας να αποδώσει την γωνία κλίσης του quadcopter μας, αλλά όπως έχουμε αναφέρει στην εισαγωγή της ενότητας αυτής, το μοντέλο αυτό αναπτύχθηκε για να κατανοήσουμε την λειτουργία ενός τέτοιου συστήματος.

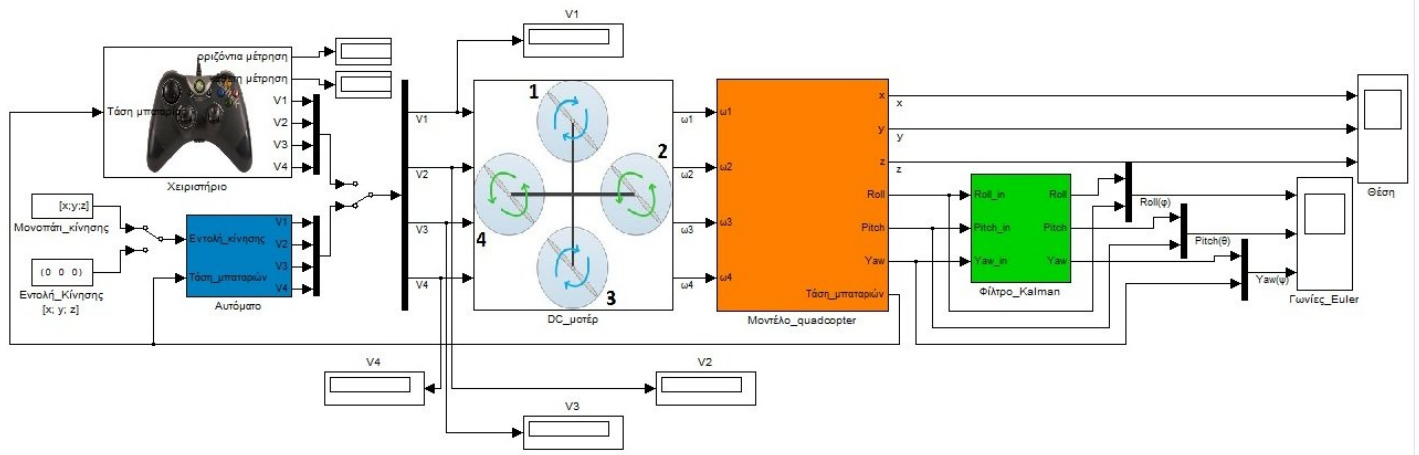
Στο δεύτερο γράφημα παρατηρούμε τους παλμούς που παίρνουμε κάθε φορά, ανάλογα με την κίνηση που θέλουμε να κάνουμε ώστε να αποκτήσει το μοντέλο μας, και την ανάλογη κλίση. Στον άξονα z , δεν υπάρχουν διαταραχές της γωνίας, γι'αυτό και είναι η τιμή του σταθερή στο μηδέν (μωβ γραμμή).

Το μοντέλο αυτό, σε μεγάλο βαθμό, κατάφερε να μας παρουσιάσει τον τρόπο λειτουργίας ενός τέτοιου συστήματος, και μας βοήθησε παράλληλα με την θεωρία να εξοικιωθούμε γρήγορα με αυτό. Στην επόμενη ενότητα όμως, θα παρουσιαστεί το δικό μας μοντέλο, το οποίο θα βασίζεται στις διαφορικές εξισώσεις τις οποίες παράγαμε εμείς οι ίδιοι.

3.2 Δεύτερο μοντέλο quadcopter

3.2.1 Παρουσίαση του μοντέλου

Το μοντέλο της παρακάτω εικόνας, είναι εμφανώς μικρότερο, από το πρώτο μας μοντέλο, αλλά παρ'όλα αυτά είναι φτιαγμένο από εμάς προκειμένου να εξυπηρετήσει τους σκοπούς της πτυχιακής αυτής. Το μοντέλο μας αποτελείται συνολικά από έξι μέρη. Το πρώτο του μέρος αποτελείται από τις εισόδους του συστήματός μας, είτε χειροκίνητα είτε αυτόματα, διανέμοντας την τάση στα μοτέρ μας, ανάλογα με την κίνηση που επιθυμούμε να πραγματοποιηθεί. Το δεύτερό του μέρος αποτελείται από την μοντελοποίηση και τον έλεγχο στροφών των τεσσάρων μοτέρ μας (η εικόνα που βρίσκεται ως “μάσκα” στο block DC_μοτέρ μας βοηθάει στην αρίθμηση των μοτέρ μας, αλλά και στο να κάνουμε σαφές, την φορά περιστροφής του καθενός). Το τρίτο και παράλληλα τέταρτό του μέρος αποτελείται από το μοντέλο του quadcopter μας και εσωτερικά αυτού βρίσκονται τα αισθητήριά μας. Επίσης εσωτερικά του block αυτού βρίσκονται οι μπαταρίες που χρησιμοποιούμε στο μοντέλο μας, καθώς επίσης και μια κάμερα η οποία μας δίνει δεδομένα για την πτήση μας τα οποία αποθηκεύονται σε αρχείο αμέσως μετά το τέλος της προσομοίωσής μας. Το πέμπτο του μέρος αποτελείται από το “φιλτράρισμα” που κάνουμε στις τιμές που λαμβάνουμε από τα αισθητήρια. Το έκτο και τελευταίο του μέρος αποτελείται από τα γραφήματα που παίρνουμε κατά την διάρκεια λειτουργίας του quadcopter μας.



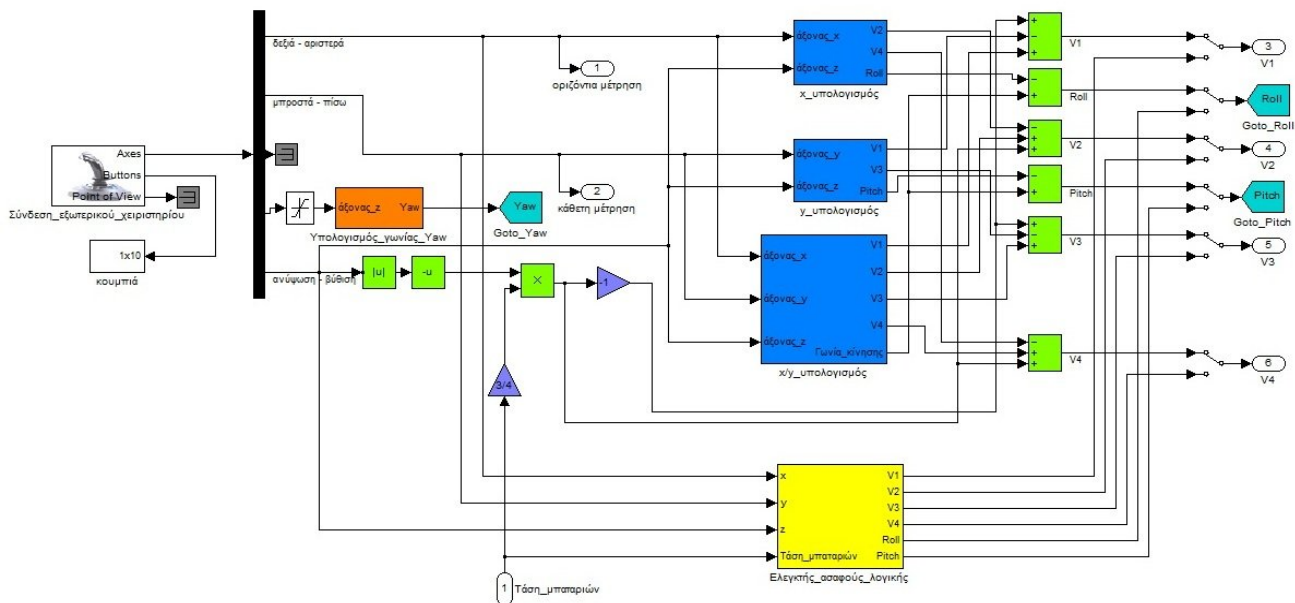
Εικόνα 30: Quadcopter στο MATLAB

Είσοδος στο μοντέλο μας (πορτοκαλί block) είναι οι γωνιακές ταχύτητες των τεσσάρων μοτέρ μας, σε RPM ($\omega_1, \omega_2, \omega_3, \omega_4$). Το πρόσημο των τάσεων στην γενική είσοδό μας έχει φυσική σημασία και συγκεκριμένα υποδηλώνει την φορά περιστροφής των μοτέρ μας. Δεξιόστροφη περιστροφή του έλικα για θετικό πρόσημο και αριστερόστροφη περιστροφή του έλικα για αρνητικό πρόσημο. Αυτό εφαρμόζεται στα αντικριστά μοτέρ μας (M1-M3 και M2-M4), και ο λόγος είναι η εξουδετέρωση της αεροδυναμικής ροπής (γωνιακή επιτάχυνση), στον άξονα z και αυτό σημαίνει άμεσα πως δεν χρειάζεται να γίνει σταθεροποίηση της γωνίας yaw. Το μοντέλο μας συνολικά έχει 7 εξόδους (x, y, z, roll, pitch, yaw, τάση_μπαταριών) από τις οποίες παίρνουμε 2 γραφικές. Μια με τις τιμές των θέσεων (x, y, z) και μια άλλη με τις γωνίες του Euler (roll, pitch, yaw). Η τάση των μπαταριών διανέμεται από την βάση του μοντέλου μας, στην γενική είσοδο, είτε του χειριστηρίου είτε της διάταξης αυτόματης πορείας.

3.2.2 Διατάξεις των γενικών μας εισόδων

- Διατάξεις χειροκίνητης εισόδου:
 - Υλοποίηση ελέγχου μέσω εφαρμογής συνθηκών:

Το block της εικόνας που ακολουθεί με τίτλο “χειριστήριο”, στο εσωτερικό του περιέχει δυο μεθόδους χειροκίνητου ελέγχου μέσω των αναλογικών μοχλών ενός χειριστηρίου.



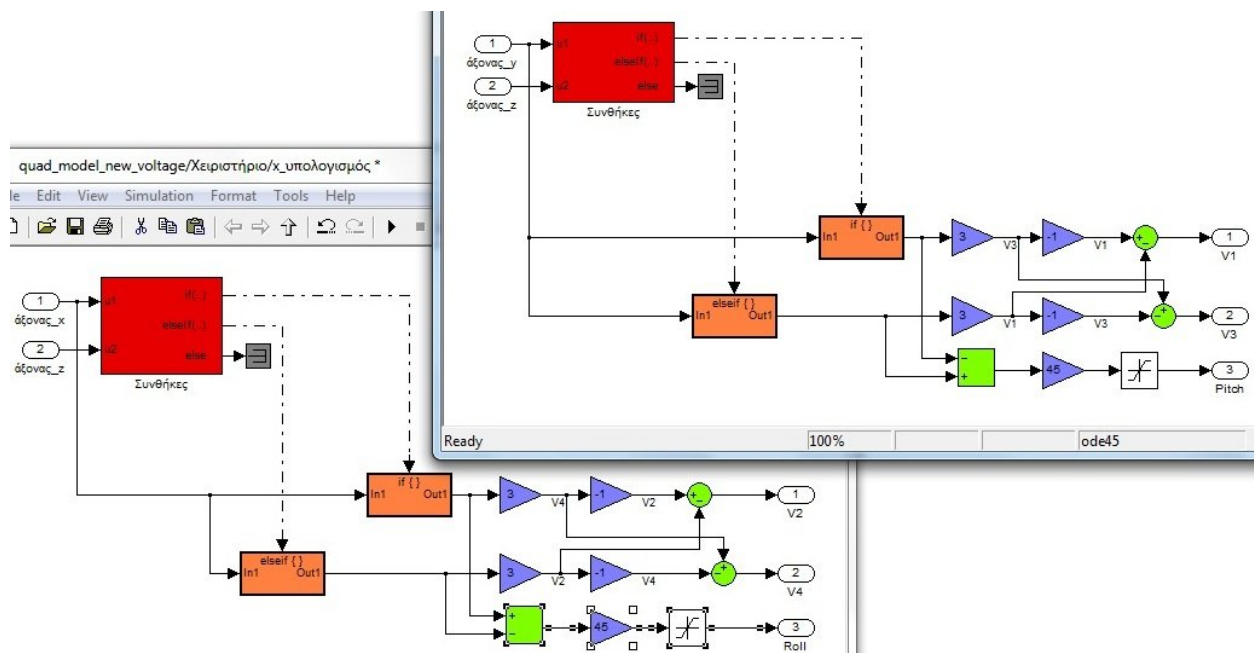
Εικόνα 31: Εσωτερική διάταξη ελέγχου κίνησης του quadcopter χειροκίνητα

Συνδέοντας το χειριστήριο στον υπολογιστή μας, παίρνουμε το ID που του δίνει ο υπολογιστής μας, και το τοποθετούμε ως είσοδο στο block μας “Σύνδεση_εξωτερικού_χειριστηρίου”. Αρχικά έγιναν δοκιμές για να δούμε πως αποκρίνεται το χειριστήριό μας στις διάφορες εντολές που του δίναμε (πάτημα κουμπιών, κίνηση των αναλογικών μοχλών). Το αποτέλεσμα των δοκιμών αυτών μας έδειξε πως πατώντας ένα κουμπί αλλάζουμε την κατάσταση του από 0 σε 1, και για τους μοχλούς, πως καθένας τους έχει 2 άξονες κίνησης με εύρος τιμών από -1 μέχρι 1. Επίσης ως είσοδο στο σύστημά μας εισάγουμε και την τάση των μπαταριών που είναι τοποθετημένες στο quadcopter μας, ώστε να γίνονται οι διάφορες μεταβολές τάσεις, όταν κάνουμε κάποιες κινήσεις στους μοχλούς του χειριστηρίου.

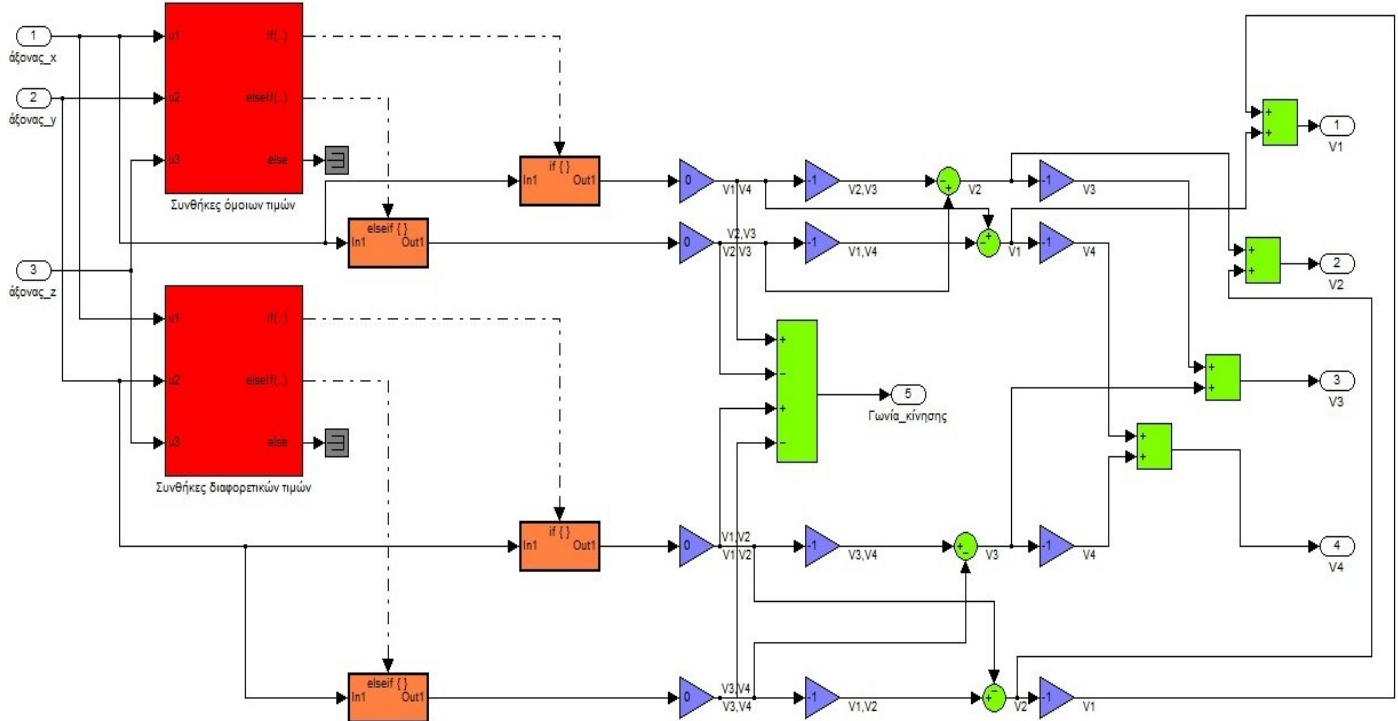
Το block μας έχει δυο πιθανές εξόδους: 1) τις τάσεις που θα μπουν ως είσοδοι στα μοτέρ μας, μετά τον υπολογισμό τους από την διάταξη που αναπτύξαμε, και 2) τις τάσεις που θα μπουν ως είσοδοι στα μοτέρ μας, μετά από τον υπολογισμό τους μέσω του ελεγκτή ασαφούς λογικής, που έχουμε ρυθμίσει.

Αφού ολοκληρώθηκε η διαδικασία αυτή, στην συνέχεια μελετήσαμε όλες τις πιθανές περιπτώσεις κίνησης του quadcopter κατά τους άξονές του, απορρίπτοντας οποιαδήποτε περίπτωση συνδυασμού με τον άξονα z. Ένα βασικό μέτρο που λήφθηκε είναι σε περίπτωση που το σύστημά μας βρίσκεται στο έδαφος, να μην δέχεται κίνηση στους άξονες x, γ αν πρώτα δεν αποκτήσει ύψος. Αναπτύχθηκαν λοιπόν δυο περιπτώσεις: Η πρώτη περίπτωση ήταν να ετοιμάσουμε με απλές εντολές if, όλες τις πιθανές περιπτώσεις κίνησης του quadcopter μας. Η δεύτερη περίπτωση πραγματοποιήθηκε με έναν ελεγκτή ασαφούς λογικής. Και οι δυο περιπτώσεις, θα αναπτυχθούν εκτενώς στην συνέχεια.

Στην εικόνα 31 βλέπουμε τρία block. Οι τίτλοι αυτών είναι: “x_υπολογισμός”, “γ_υπολογισμός”, “x/γ_υπολογισμός”. Το εσωτερικό των block αυτών εμφανίζεται στις παρακάτω εικόνες.



Εικόνα 32: Εσωτερικό των block “x_υπολογισμός”, “γ_υπολογισμός”



Εικόνα 33: Εσωτερικό του block “x/y_Υπολογισμός”

Στα δυο block της εικόνας η διαδικασία που επιτελείται είναι σχεδόν όμοια. Στο block “x_Υπολογισμός”, έχουμε δυο εισόδους. Η πρώτη είσοδος αφορά στα δεδομένα του αριστερού μοχλού του χειριστηρίου μας (πιο συγκεκριμένα για τις κινήσεις αριστερά-δεξιά) και η δεύτερη είσοδος αφορά τον άξονα z και χρησιμοποιείται κυρίως για τον βασικό περιορισμό μας, δηλαδή να μην κινείται το quadcopter, αν πρώτα δεν αποκτήσει ύψος. Οι τρεις έξοδοι του block αυτού αφορούν την αλλαγή της τάσης στα μοτέρ M2-M4 προκειμένου να πραγματοποιηθεί η κάθε κίνηση και ο υπολογισμός της γωνίας Roll. Οι συνθήκες που χρησιμοποιούμε στο κόκκινο block είναι οι εξής:

1) Αν $X > 0$ (κίνηση αριστερού μοχλού δεξιά) & $Z < -0.2$ (κίνηση δεξί μοχλού προς τα πάνω) ΤΟΤΕ:

$$V2 = -6, V4 = -12, \text{Roll} = -45^\circ.$$

2) Αν $X < 0$ (κίνηση αριστερού μοχλού αριστερά) & $Z < -0.2$ (κίνηση δεξί μοχλού προς τα πάνω) ΤΟΤΕ:

$$V2 = -12, V4 = -6, \text{Roll} = 45^\circ.$$

Ομοίως στο block “ γ _Υπολογισμός”, έχουμε την μια είσοδο που αφορά την κίνηση του αριστερού μοχλού του χειριστηρίου μας (πιο συγκεκριμένα για τις κινήσεις πάνω-κάτω) και η δεύτερη που αφορά όπως και στο προηγούμενο block μας τον βασικό περιορισμό μας. Οι τρεις έξοδοι του block αυτού, σε αντίθεση με το προηγούμενο, αφορούν την αλλαγή της τάσης στα μοτέρ M1-M3 προκειμένου να πραγματοποιηθεί η κάθε κίνηση και τον υπολογισμό της γωνίας Pitch. Οι συνθήκες που χρησιμοποιούμε στο κόκκινο block είναι οι εξής:

1) Αν $Y > 0$ (κίνηση αριστερού μοχλού δεξιά) & $Z < -0.2$ (κίνηση δεξί μοχλού προς τα πάνω) ΤΟΤΕ:

$$V1 = 12, V3 = 6, \text{Pitch} = 45^\circ.$$

2) Αν το $Y < 0$ (κίνηση αριστερού μοχλού αριστερά) & $Z < -0.2$ (κίνηση δεξί μοχλού προς τα πάνω) ΤΟΤΕ:

$$V1 = 6, V3 = 12, \text{Pitch} = -45^\circ.$$

Αυτά τα δυο block είχαν να κάνουν με τις ευθύγραμμες κινήσεις του quadcopter. Το τρίτο block που θα εξετάσουμε τώρα έχει να κάνει με την σύνθετη κίνηση του μηχανισμού μας, με παράλληλη μεταβολή στον x και y άξονα (στο χειριστήριο αυτό επιτυγχάνεται με την κίνηση του αριστερού μοχλού διαγωνίως) καθώς επίσης και τον σύνθετο υπολογισμό της γωνίας μας. Οι είσοδοι στο block αυτό είναι τρεις και αφορούν τους τρεις άξονες κινήσεως μας. Οι πιθανές διαγώνιες κινήσεις του μοχλού είναι τέσσερις. Οι δυο θα έχουν τα ίδια X, Y ενώ οι άλλες δυο θα έχουν διαφορετικά. Οπότε χωρίσαμε τις τέσσερις περιπτώσεις αυτές σε δυο συνθήκες. Μια όταν τα X, Y θα είναι ομόσημα, και μια όταν θα είναι ετερόσημα. Αντίθετα με τα προηγούμενα block, αυτό το block θα δημιουργεί μεταβολή τάσης και για τα τέσσερα μοτέρ μας και θα παράγει και δυο επιθυμητές γωνίες Roll και Pitch. Κοιτώντας το πάνω κόκκινο block της εικόνας, το οποίο αφορά την ομόσημη κίνηση του μοχλού, δημιουργήσαμε τις εξής συνθήκες:

1) Αν $X > 0$ & $Y > 0$ (κίνηση αριστερού μοχλού διαγώνια κάτω-δεξιά) & $Z < -0.2$ (κίνηση δεξί μοχλού πάνω) ΤΟΤΕ:

$$V1 = 12, V2 = -6, V3 = 6, V4 = -12, \text{Roll} = -45^\circ, \text{Pitch} = -45^\circ.$$

2) Αν $X < 0$ & $Y < 0$ (κίνηση αριστερού μοχλού διαγώνια πάνω-αριστερά) & $Z < -0.2$ (κίνηση δεξι μοχλού πάνω) τότε:

$$V1 = 6, V2 = -12, V3 = 12, V4 = -6, \text{Roll} = 45^\circ, \text{Pitch} = 45^\circ.$$

Κοιτώντας τώρα το κάτω κόκκινο block της ίδιας εικόνας, το οποίο αφορά την ετερόσημη κίνηση του μοχλού, δημιουργήσαμε τις εξής συνθήκες:

3) Αν $X < 0$ & $Y > 0$ (κίνηση αριστερού μοχλού διαγώνια κάτω-αριστερά) & $Z < -0.2$ (κίνηση δεξι μοχλού πάνω) ΤΟΤΕ:

$$V1 = 12, V2 = -12, V3 = 6, V4 = -6, \text{Roll} = 45^\circ, \text{Pitch} = -45^\circ.$$

4) Αν $X > 0$ & $Y < 0$ (κίνηση αριστερού μοχλού διαγώνια πάνω-δεξιά) & $Z < -0.2$ (κίνηση δεξι μοχλού πάνω) ΤΟΤΕ:

$$V1 = 6, V2 = -6, V3 = 12, V4 = -12, \text{Roll} = -45^\circ, \text{Pitch} = 45^\circ.$$

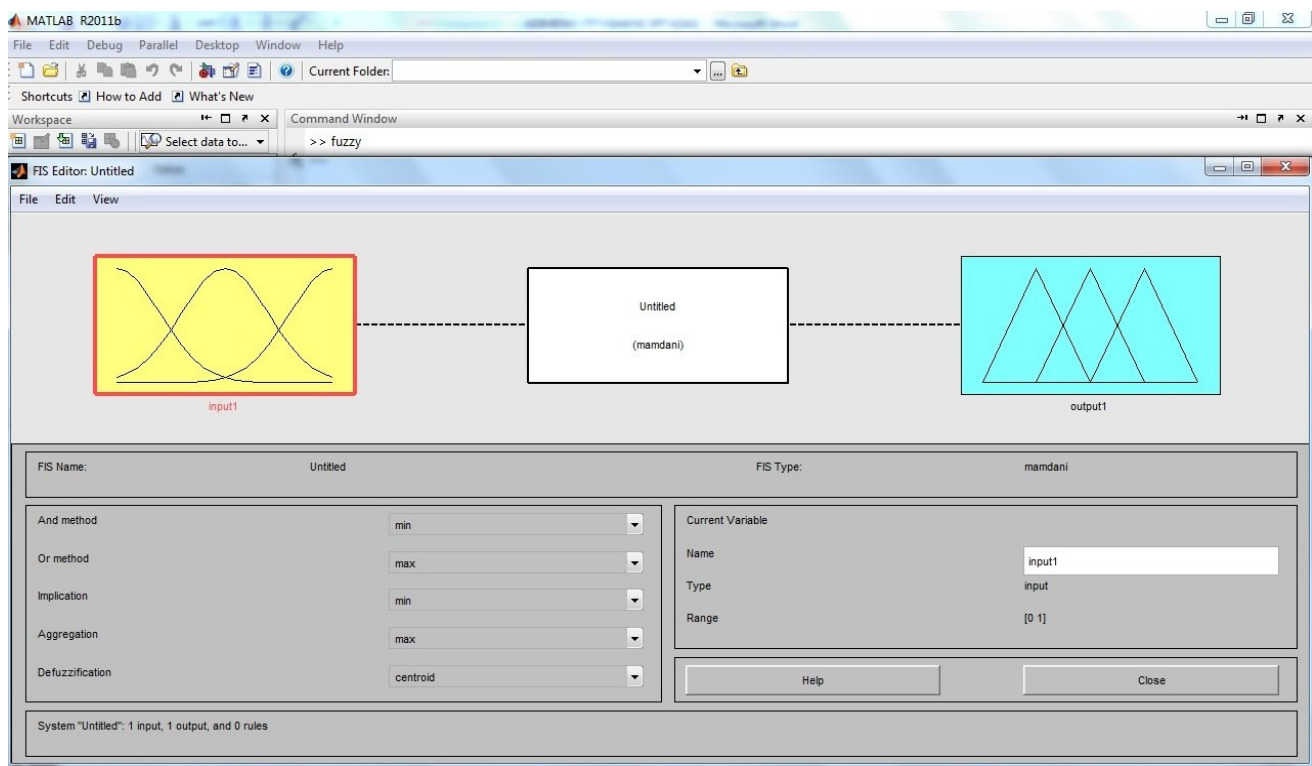
Πέρα όμως από τις συνθήκες αυτές, υπάρχει και μια ακόμα λεπτομέρεια που αφορά τον άξονα z και είναι η περίπτωση εκκίνησης ανύψωσης του quadcopter μας. Στην περίπτωση αυτή δεχόμαστε την είσοδο που αφορά την τάση των μπαταριών μας, και με τον τρόπο αυτό γίνεται η αρχική διανομή στα μοτέρ, η οποία εν συνεχεία διαφοροποιείται ανάλογα με την κίνηση του αριστερού μοχλού μας. Η γενική περίπτωση που αφορά την ανύψωση του quadcopter, διανέμει τις εξής τάσεις:

$$V1 = V3 = 9 \text{ και } V2 = V4 = -9, \text{Roll} = 0^\circ, \text{Pitch} = 0^\circ.$$

Τέλος, αθροίζουμε όλα τα σήματα που αφορούν την ίδια τάση (είναι τρία για καθεμιά από τις τέσσερις τάσεις μας) και κάνουμε δοκιμές ώστε να επιτευχθεί ο κατάλληλος συνδυασμός, που δεν δημιουργεί προβλήματα στην διανομή της τάσης μας, σύμφωνα πάντα με την μελέτη που έχουμε πραγματοποιήσει.

– Υλοποίηση και εφαρμογή ενός ελεγκτή ασαφούς λογικής:

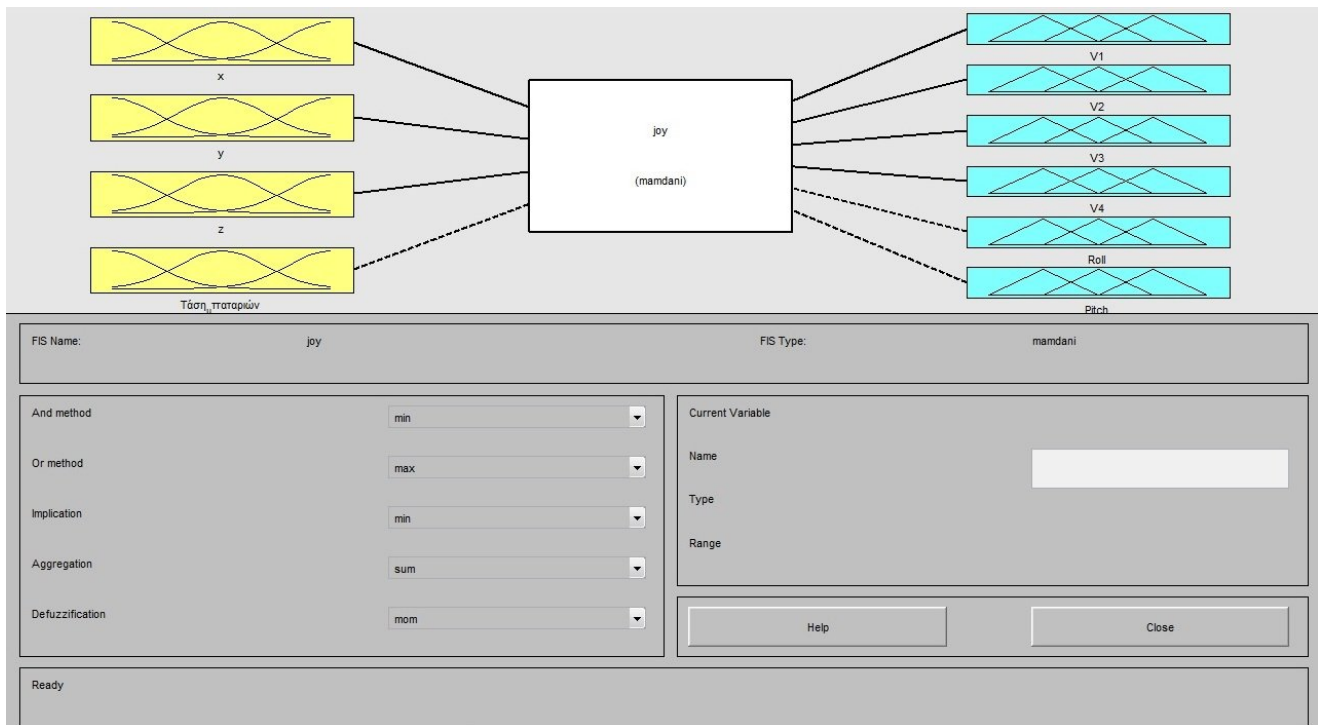
Βλέποντας την εικόνα 34 υπάρχει ένα κίτρινο block με τίτλο “Ελεγκτής ασαφούς λογικής”. Εσωτερικά αυτού υπάρχει το block που μας παρέχει το Simulink το μόνο που μας ζητείται είναι να εισάγουμε ένα όνομα αρχείου με κατάληξη “.fis”. Το δύσκολο κομμάτι του ελεγκτή αυτού όπως είπαμε και στην θεωρία είναι η ρύθμισή του. Η λύση βρίσκεται εκτός του περιβάλλοντος του Simulink. Συγκεκριμένα βρίσκεται σε μια εντολή του κεντρικού περιβάλλοντος του MATLAB. Έτσι γράφοντας στο “Command Window” του MATLAB την εντολή “fuzzy” μας εμφανίζεται το παρακάτω παράθυρο:



Εικόνα 34: Κεντρικό παράθυρο του FIS Editor

Στο παράθυρο αυτό βλέπουμε μια είσοδο, μια έξοδο, και την ενδιάμεση διαδικασία που επιτελείται. Εννοείται πως μπορούμε να εισάγουμε όσες εισόδους και εξόδους θέλουμε εμείς. Στην περίπτωση μας, θα χρειαστούμε τέσσερις εισόδους

(x,y,z,τάση_μπαταριών) και έξι εξόδους (V1,V2,V3,V4,Roll,Pitch). Η εικόνα που θα βλέπουμε εμείς, είναι η παρακάτω:



Εικόνα 35: Διαμόρφωση εισόδων – εξόδων για την εφαρμογή μας

Πατώντας διπλό-κλικ σε κάποια από τις εισόδους-εξόδους μας εμφανίζει όλα τα σήματά μας (εισόδου-εξόδου). Εκεί τα διαμορφώνουμε όπως εμείς θέλουμε ανάλογα με τις περιπτώσεις που θέλουμε να δημιουργήσουμε. Εμείς έχουμε ορίσει τις εξής γραφικές και τιμές:

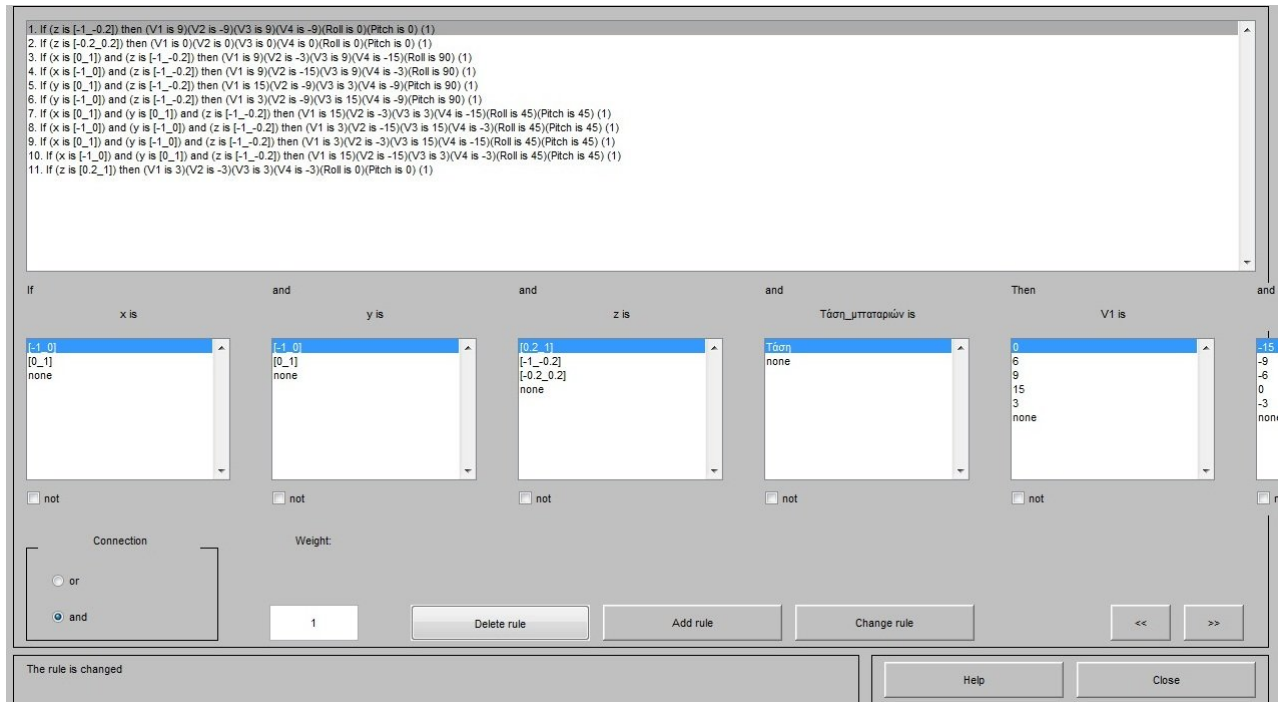
Είσοδοι:

1. για το x έχουμε ορίσει δυο πιθανά σήματα. (ένα από -1 έως 0 και το άλλο από 0 έως 1)
2. για το y έχουμε ορίσει δυο πιθανά σήματα. (ένα από -1 έως 0 και το άλλο από 0 έως 1)
3. για το z έχουμε ορίσει τρία πιθανά σήματα. (ένα από -1 έως -0.2 ένα από -0.2 έως 0.2 και το τελευταίο από 0.2 έως 1)
4. τάση από τις μπαταρίες μας ορισμένη στα 12V.

Έξοδοι:

1. για την V1 έχουμε ορίσει πέντε πιθανές τιμές (0, 3, 6, 9, 12)
2. για την V2 έχουμε ορίσει πέντε πιθανές τιμές (-12,-9,-6,-3,0)
3. για την V3 έχουμε ορίσει πέντε πιθανές τιμές (0, 3, 6, 9, 12)
4. για την V4 έχουμε ορίσει πέντε πιθανές τιμές (-12,-9,-6,-3,0)
5. για την γωνία Roll έχουμε ορίσει πέντε πιθανές τιμές (-90°, -45°, 0°, 45°, 90°)
6. για την γωνία Pitch έχουμε ορίσει πέντε πιθανές τιμές (-90°, -45°, 0°, 45°, 90°)

Με αυτές λοιπόν τις τιμές μπορούμε τώρα να δημιουργήσουμε τις περιπτώσεις που θέλουμε να ελέγχει ο ελεγκτής ασαφής λογικής. Πατώντας επάνω στο άσπρο κουτάκι της εικόνας 35, εμφανίζεται η παρακάτω εικόνα:



Εικόνα 36: Δήλωση των κανόνων ελέγχου για τον ελεγκτή ασαφούς λογικής χειριστήριου

Συνολικά δημιουργήσαμε 11 κανόνες που συμπεριλαμβάνουν ό,τι ακριβώς είχαμε πραγματοποιήσει στην πρώτη μας απόπειρα χειροκίνητου ελέγχου, κάνοντας μερικές προσθήκες που θα μας δώσουν περισσότερη ακρίβεια. Σίγουρα όπως βλέπουμε σε

σχέση με την πρώτη περίπτωση είναι πολύ πιο μαζεμένα τα δεδομένα μας, και καταλαμβάνει και μικρότερο χώρο στην μοντελοποίησή μας.

Εφόσον, δημιουργήσαμε τους κανόνες μας, στην συνέχεια προκειμένου να αποθηκεύσουμε την διαδικασία αυτή μαζί με τους κανόνες μας, το κάνουμε “export σε αρχείο” και του δίνουμε όποιο όνομα θέλουμε. Τέλος, επιστρέφουμε στο Simulink και εκεί που μας ζητάει το όνομα του αρχείου, βάζουμε το όνομα που έχουμε δώσει στο αρχείο που μόλις δημιουργήσαμε. Ο ελεγκτής ασαφούς λογικής, είναι έτοιμος προς εφαρμογή. Πάλι πρέπει να πραγματοποιήσουμε πολλές δοκιμές, με όλες τις δυνατές περιπτώσεις, που μπορεί να υπάρξουν, και αν κάπου εντοπίσουμε λάθος, είτε διορθώνουμε κάποιον κανόνα είτε δημιουργούμε έναν νέο.

Παρόλο που τα αποτελέσματα, και από άποψη απόδοσης αλλά και εξοικονόμησης σημαντικού χώρου στην μοντελοποίησή μας είναι πάρα πολύ ικανοποιητικά, δυστυχώς ο ελεγκτής αυτός έχει ένα σοβαρό ελάττωμα. Μπαίνοντας στο πρόγραμμά μας, πρέπει κάθε φορά να κάνουμε “import στο workspace” το αρχείο με την διαδικασία ελέγχου. Το σημαντικότερο ελάττωμά του όμως είναι πως ανοίγοντάς το μέσω του MATLAB πρέπει να μεταφέρονται τα δεδομένα της διαδικασίας που έχουμε ορίσει στο Simulink, κατάσταση η οποία θα μας κοστίζει πολύ σε χρόνο εξαιτίας της μεταφοράς αρκετών δεδομένων.

- Διάταξη αυτόματης πορείας:

Η διάταξη αυτόματης πορείας βρίσκεται εσωτερικά του μπλε block της εικόνας 30. Το block αυτό στο εσωτερικό του περιέχει έναν ελεγκτή ασαφούς λογικής τον οποίο έχουμε προγραμματίσει εμείς, ώστε να επιτελεί μια διαδικασία με μια απλή μεταβολή στις τιμές των αξόνων μας. Το block αυτό έχει δυο εισόδους. Η πρώτη παίρνει ως δεδομένα τις τιμές κάθε άξονα. Το εύρος τιμών που παίρνουν οι άξονές μας είναι από -1 μέχρι 1, και αυτό γιατί δεν θέλουμε να δώσουμε στο μοντέλο μας εμείς την θέση, αλλά να την παράξει το ίδιο. Τις τιμές αυτές, τις δηλώνουμε για να δώσουμε στο μοντέλο μας κατεύθυνση, προς τα που δηλαδή θα κινηθεί. Για την είσοδο της κίνησης στην διάταξη αυτόματης πορείας, έχουμε αναπτύξει δυο τρόπους. Ο ένας είναι να αλλάζουμε εμείς οι ίδιοι τις τιμές των αξόνων ενώ ο άλλος είναι να έχουμε δημιουργήσει ένα επιθυμητό μονοπάτι πριν ξεκινήσουμε την προσομοίωση και να παρακολουθήσουμε την πορεία του μοντέλου μας. Η δεύτερη είσοδος του block μας παίρνει ως δεδομένο την τάση των μπαταριών μας. Οι έξοδοι του block μας είναι επτά και αφορούν τις τέσσερις τάσεις που θα μπαίνουν τελικά ως είσοδο στα μοτέρ μας και οι τρεις επιθυμητές γωνίες κλίσης του συστήματός μας.

Ο ελεγκτής ασαφούς λογικής της διάταξης αυτής σε σχέση με τον ελεγκτή της χειροκίνητης διάταξης που έχουμε αναπτύξει, έχει και αυτός τέσσερις εισόδους (x,y,z,τάση_μπαταριών) αλλά έχει επτά εξόδους (τις τέσσερις τάσεις των μοτέρ μας, και τις τρεις γωνίες κλίσης μας). Πατώντας, λοιπόν διπλό-κλικ σε κάποια από τις εισόδους-εξόδους μας εμφανίζει όλα τα σήματά μας (εισόδου-εξόδου). Εκεί τα διαμορφώνουμε όπως εμείς θέλουμε ανάλογα με τις περιπτώσεις που θέλουμε να δημιουργήσουμε. Εμείς έχουμε ορίσει τις εξής γραφικές και τιμές:

Είσοδοι:

1. για το x έχουμε ορίσει τρία πιθανά σήματα. (ένα από -1 έως 0, το άλλο από 0 έως 1 και το τελευταίο ακριβώς στο 0)
2. για το y έχουμε ορίσει τρία πιθανά σήματα. (ένα από -1 έως 0, το άλλο από 0 έως 1 και το τελευταίο ακριβώς στο 0)
3. για το z έχουμε ορίσει τρία πιθανά σήματα. (ένα από -1 έως 0, το άλλο από 0 έως 1 και το τελευταίο ακριβώς στο 0)
4. για την τάση από τις μπαταρίες μας την οποία έχουμε ορίσει στα 12V.

Έξοδοι:

1. για την $V1$ έχουμε ορίσει πέντε πιθανές τιμές (0, 3, 6, 9, 12)
2. για την $V2$ έχουμε ορίσει πέντε πιθανές τιμές (-12,-9,-6,-3,0)
3. για την $V3$ έχουμε ορίσει πέντε πιθανές τιμές (0, 3, 6, 9, 12)
4. για την $V4$ έχουμε ορίσει πέντε πιθανές τιμές (-12,-9,-6,-3,0)
5. για την γωνία Roll έχουμε ορίσει πέντε πιθανές τιμές (-90°, -45°, 0°, 45°, 90°)
6. για την γωνία Pitch έχουμε ορίσει πέντε πιθανές τιμές (-90°, -45°, 0°, 45°, 90°)
7. για την γωνία Yaw έχουμε ορίσει μια πιθανή τιμή στις 0°.

Παρακάτω στην εικόνα 37 θα δούμε το παράθυρο με τους κανόνες που έχουμε

θέσει:

The screenshot shows a rule editor interface. At the top, there is a list of 17 rules. Below this, a visual representation of a rule is shown, consisting of several input fields and a 'Then' field. The inputs are labeled 'x is', 'y is', 'z is', 'Τάση_μπαταριών is', and 'V1 is'. Each input field has a dropdown menu with options like '-1_0', '[0_1]', '0', and 'none'. The 'Then' field has a dropdown menu with options like '0', '6', '9', '15', '3', and 'none'. Below the visual representation, there are buttons for 'not', 'or', and 'and'. The 'and' button is selected. There is also a 'Weight' field with the value '1'. At the bottom, there are buttons for 'Delete rule', 'Add rule', 'Change rule', 'Help', and 'Close'. The status bar at the bottom left says 'The rule is changed'.

Εικόνα 37: Δήλωση των κανόνων ελέγχου για τον ελεγκτή ασαφούς λογικής αυτόματης πορείας

Συνολικά δημιουργήσαμε 17 κανόνες που συμπεριλαμβάνουν ό,τι ακριβώς είχαμε μελετήσει και στην περίπτωση που χρησιμοποιούσαμε το χειριστήριο κάνοντας μια προσθήκη στις εξόδους για την γωνιά Yaw και 6 προσθήκες κανόνων διότι στην διάταξη αυτή, θα πρέπει να αποφύγουμε οποιαδήποτε περίπτωση που θα εκθέσει το μοντέλο μας σε σημαντικό κίνδυνο.

Εφόσον, δημιουργήσαμε τους κανόνες μας, στην συνέχεια προκειμένου να αποθηκεύσουμε την διαδικασία αυτή μαζί με τους κανόνες μας, το κάνουμε “export σε αρχείο” και του δίνουμε όποιο όνομα θέλουμε. Τέλος, επιστρέφουμε στο Simulink και εκεί που μας ζητάει το όνομα του αρχείου, βάζουμε το όνομα που έχουμε δώσει στο αρχείο που μόλις δημιουργήσαμε. Ο ελεγκτής ασαφούς λογικής, είναι έτοιμος προς εφαρμογή. Πάλι πρέπει να πραγματοποιήσουμε πολλές δοκιμές, με όλες τις δυνατές περιπτώσεις, που μπορεί να υπάρξουν, και αν κάπου εντοπίσουμε λάθος, είτε διορθώνουμε κάποιον κανόνα είτε δημιουργούμε έναν νέο.

Καταφέραμε με αρκετούς τρόπους, συγκεκριμένα τέσσερις, να ελέγξουμε την κίνηση του quadcopter μας με μεταβολές στην τάση εισόδου, ώστε ανάλογα με την κάθε κίνηση που επιθυμούμε να κάνει να μεταβάλλονται αναλόγως οι ταχύτητες. Στην ενότητα αυτή υλοποιήσαμε δυο ελεγκτές ασαφούς λογικής, έναν για την περίπτωση χρήσης χειριστηρίου και έναν για την αυτόματη πορεία, εφαρμόζοντας όσα μάθαμε από την θεωρία του. Το δύσκολο κομμάτι της υλοποίησης των ελεγκτών μας ήταν η ρύθμισή τους ώστε να μας δίνουν τις τιμές που θέλουμε. Στην εικόνα 35 βλέπουμε κάτω αριστερά πέντε πεδία στα οποία μπορούμε να αλλάξουμε τιμές. Τα πεδία αυτά, μαζί με τις τιμές που τους έχουμε δώσει στους δικούς μας ελεγκτές, είναι:

1. And method (Μέθοδος λογικής πράξης “και”) : min
2. Or method (Μέθοδος λογικής πράξης “ή”) : max
3. Implication (Υπαινιγμός) : min
4. Aggregation (Συσσωμάτωση) : sum

5. Defuzzification (περιφραστικά θα μπορούσε να μεταφραστεί ως αποδιασαφίνιση) : mom

Το σημαντικότερο πεδίο από τα παραπάνω είναι το τελευταίο. Το MATLAB μας δίνει πέντε επιλογές που μπορούμε να κάνουμε για το πεδίο αυτο, και είναι οι εξής:

1. Centroid
2. Bisector
3. Mom (Middle of Maximum)
4. Lom (Largest of Maximum)
5. Som (Smallest of Maximum)

Αναπτύχθηκε κώδικας στο περιβάλλον του MATLAB προκειμένου να μπορέσουμε να κατανοήσουμε την σημασία και την λειτουργία της κάθε μεθόδου “αποδιασαφίνισης” και να επιλέξουμε την κατάλληλη για εμάς. Παρακάτω θα παρουσιαστεί ο κώδικας που αναπτύχθηκε:

```
% Δημιουργία εισόδου για defuzzification
x = -10:0.1:10; %Οριοθετούμε των οριζόντιο άξονά μας
mf1 = trapmf(x, [-10 -8 -2 2]); %Δημιουργούμε το πρώτο μας σήμα, Τραπεζοειδή συνάρτηση μέλος
mf2 = trapmf(x, [-5 -3 2 4]); %Δημιουργούμε το δεύτερό μας σήμα
mf3 = trapmf(x, [2 3 8 9]); %Δημιουργούμε το τρίτο μας σήμα
mf = max(0.5*mf2, max(0.9*mf1, 0.1*mf3)); %Δημιουργία aggregate (συσσωματωμένου) σήματος με
%την μέθοδο max

%Γραφικές παραστάσεις
subplot(2,1,1)
plot(x,mf1, 'LineWidth',3, 'Color', 'r');
hold on
plot(x,mf2, 'LineWidth',3, 'Color', 'g');
hold on
plot(x,mf3, 'LineWidth',3, 'Color', 'b');
title('Σήματα καθορισμού περιπτώσεων') %Τίτλος γραφικής
subplot(2,1,2)
plot(x,mf, 'LineWidth',3) %Γραφική απεικόνιση του ενιαίου σήματός μας
set(gca, 'YLim', [-1 1], 'YTick', [0 0.5 1]); %Πραγματοποιούμε ρυθμίσεις για τον άξονα Y
title('Μέθοδοι defuzzification') %Τίτλος γραφικής
```

```

%Μέθοδος Centroid
%Η μέθοδος Centroid επιστρέφει την κεντρική περιοχή κάτω από την γραφική μας. Αν σκεφτούμε
%την γραφική μας ως πραγματικό σχήμα, το Centroid είναι το σημείο στο οποίο το σχήμα μας θα
%ισορροπούσε.

x1 = defuzz(x,mf,'centroid') %Υπολογισμός του σημείου που θα επιστρέψει το defuzzification
%που πραγματοποιήσαμε στο σήμα μας

h1 = line([x1 x1],[-0.2 1.2]); %Δημιουργία κάθετης γραμμής στο σημείο που μας επιστρέφει η
%μέθοδος μας
t1 = text(x1,-0.2,' Centroid','FontWeight','bold'); %Δημιουργία του κειμένου στο σημείο μας
set([h1 t1],'Color','k') %Ένταξη του σήματός μας στην τελική γραφική μας

%Μέθοδος Bisector
%Η μέθοδος Bisector είναι μια κάθετη γραμμή η οποία θα χωρίσει την γραφική μας σε δυο ίσα
%τμήματα. Κάποιες φορές, αλλά όχι πάντα η γραμμή της μεθόδου Bisector συμπίπτει με αυτή της
%μεθόδου Centroid.

x2 = defuzz(x,mf,'bisector') %Υπολογισμός του σημείου που θα επιστρέψει το
%defuzzification που πραγματοποιήσαμε στο σήμα μας

h2 = line([x2 x2],[-0.4 1.2]); %Δημιουργία κάθετης γραμμής στο σημείο που μας επιστρέφει η
%μέθοδος μας
t2 = text(x2,-0.4,' Bisector','FontWeight','bold'); %Δημιουργία του κειμένου στο σημείο μας
set([h2 t2],'Color','r') %Ένταξη του σήματός μας στην τελική γραφική μας

%Μέθοδοι Mom(Middle of Maximum), Som(Smallest of Maximum), Lom(Largest of Maximum). Οι
%τρεις αυτές μέθοδοι αναδεικνύουν την μέγιστη τιμή που έχει δημιουργηθεί από την
%συσσωρευμένη συνάρτησης μέλους. Στο δικό μας παράδειγμα, επειδή η μέγιστη τιμή της
%συνάρτησής μας δεν είναι σημείο, αλλά εύρος τιμών, τα σημεία αυτά είναι διαφορετικά. Αν η
%συσσωρευμένη συνάρτηση μέλος εμφανίζει μέγιστη τιμή σε σημείο, τότε και οι τρεις αυτές
%μέθοδοι, επιστρέφουν ως αποτέλεσμα το σημείο αυτό.

x3 = defuzz(x,mf,'mom') %Υπολογισμός του σημείου που θα επιστρέψει το defuzzification
%της μεθόδου mom
x4 = defuzz(x,mf,'som') %Υπολογισμός του σημείου που θα επιστρέψει το defuzzification
%της μεθόδου som
x5 = defuzz(x,mf,'lom') %Υπολογισμός του σημείου που θα επιστρέψει το defuzzification
%της μεθόδου lom

h3 = line([x3 x3],[-0.7 1.2]); %Δημιουργία κάθετης γραμμής στο σημείο που μας επιστρέφει η
%μέθοδος μας
t3 = text(x3,-0.7,' MOM','FontWeight','bold'); %Δημιουργία του κειμένου στο σημείο μας
set([h3 t3],'Color','b') %Ένταξη του σήματός μας στην τελική γραφική μας

h4 = line([x4 x4],[-0.8 1.2]); %Δημιουργία κάθετης γραμμής στο σημείο που μας επιστρέφει η
%μέθοδος μας
t4 = text(x4,-0.8,' SOM','FontWeight','bold'); %Δημιουργία του κειμένου στο σημείο μας
set([h4 t4],'Color','y') %Ένταξη του σήματός μας στην τελική γραφική μας

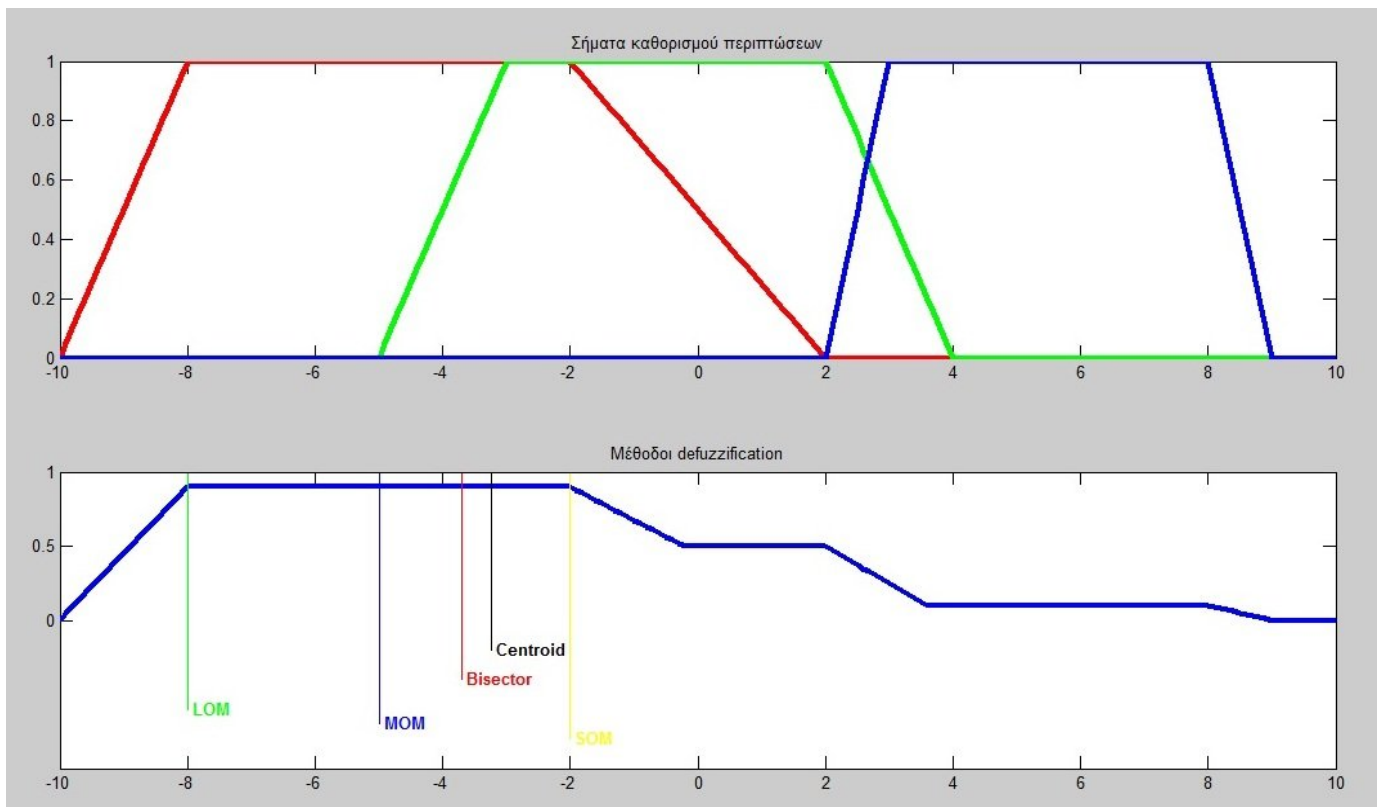
h5 = line([x5 x5],[-0.6 1.2]); %Δημιουργία κάθετης γραμμής στο σημείο που μας επιστρέφει η
%μέθοδος μας
t5 = text(x5,-0.6,' LOM','FontWeight','bold'); %Δημιουργία του κειμένου στο σημείο μας
set([h5 t5],'Color','g') %Ένταξη του σήματός μας στην τελική γραφική μας

```

Κώδικας υλοποιημένος στο MATLAB για την διαδικασία defuzzification που εκτελεί ένας ελεγκτής ασαφούς λογικής

“Τρέχοντας” τον παραπάνω κώδικα το MATLAB μας επιστρέφει τα παρακάτω σημεία και στην συνέχεια τελική γραφική μας.

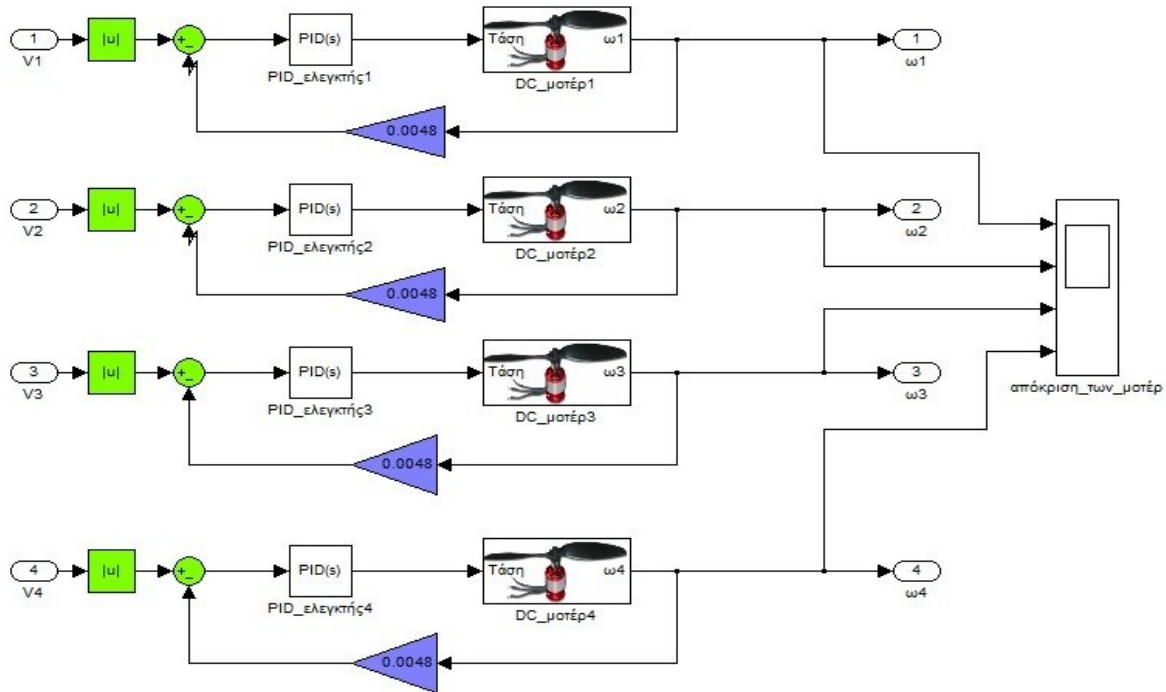
- $x_1 = -3.2281$ (σημείο μεθόδου Centroid)
- $x_2 = -3.7000$ (σημείο μεθόδου Bisector)
- $x_3 = -5$ (σημείο μεθόδου mom)
- $x_4 = -2$ (σημείο μεθόδου som)
- $x_5 = -8$ (σημείο μεθόδου lom)



Εικόνα 38: Γραφικές για τον καθορισμό περιπτώσεων (πάνω) και των αποτελεσμάτων κάθε μεθόδου defuzzification (κάτω)

3.2.3 Μοντελοποίηση των DC μοτέρ

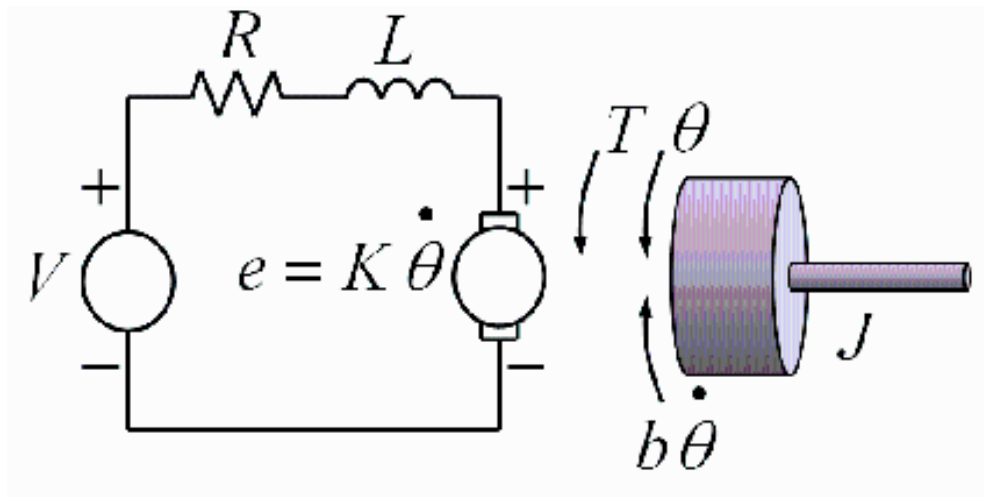
Το block με την εικόνα, ως “μάσκα” (εικόνα 30), περιέχει στο εσωτερικό του τα μοντέλα των τεσσάρων μοτέρ μας καθώς επίσης και την διαδικασία ελέγχου τους με ελεγκτές PID. Το εσωτερικό του block αυτού εμφανίζεται στην παρακάτω εικόνα:



Εικόνα 39: Σύστημα ελέγχου με PID για καθένα από τα 4 μοτέρ

Αρχικά, έγινε μοντελοποίηση των DC μοτέρ βάσει των διαφορικών εξισώσεων (μηχανικής – ηλεκτρολογικής λειτουργίας) οι οποίες είναι οι εξής:

1. $\frac{d\omega(t)}{dt} = \frac{K_t}{J} i(t) - \frac{b}{J} \omega(t)$: διαφορική εξίσωση μηχανικής λειτουργίας
2. $\frac{di(t)}{dt} = \frac{1}{L} (V_s - R i(t) - K_e \omega(t))$: διαφορική εξίσωση ηλεκτρολογικής λειτουργίας

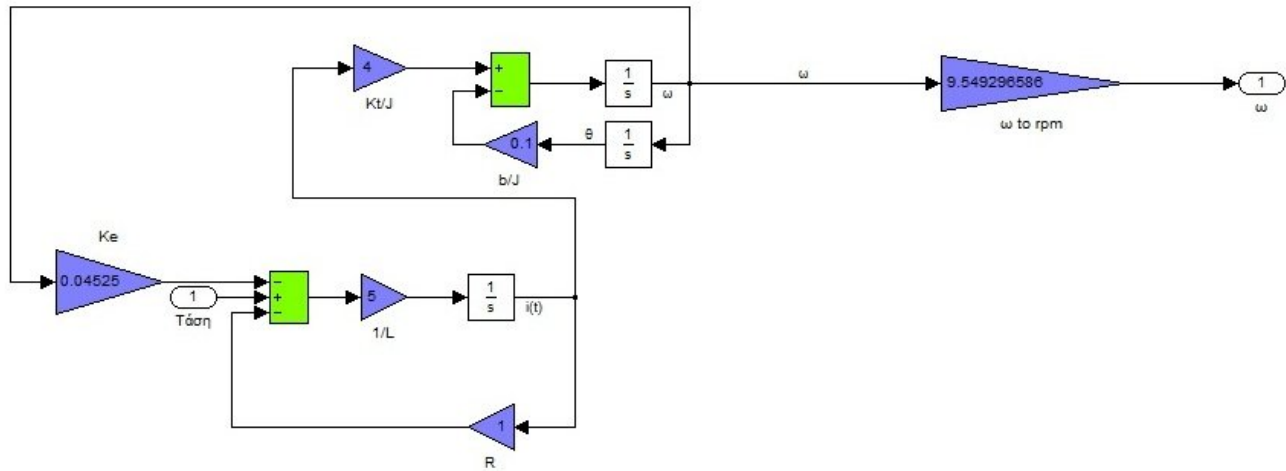


Εικόνα 40: Διαφορικές εξισώσεις ενός DC μοτέρ

Η είσοδος στο μοντέλο μας είναι η τάση από την πηγή και η έξοδός του είναι η γωνιακή ταχύτητα με την οποία περιστρέφεται. Οι τιμές στους συντελεστές των μεταβλητών επιλέχθηκαν με τέτοιο τρόπο, ώστε όταν βάζουμε στο σύστημα τη μέγιστη δυνατή τάση, τα μοτέρ μας να αποδίδουν κάποια RPM. Στην δική μας περίπτωση όταν βάζουμε είσοδο 12V, η έξοδος μας δίνει 2500RPM. Οι συντελεστές που χρησιμοποιήθηκαν στο μοντέλο μας ώστε να επιτευχθεί η εν λόγω περίπτωση είναι οι εξής:

- $K_t = 0.04$
- $J = 0.01$
- $R = 1$
- $L = 0.2$
- $K_e = 0.04525$
- $b = 0.001$

Το εσωτερικό του μοντέλου μας φαίνεται στην παρακάτω εικόνα:



Εικόνα 41: Μοντελοποίηση των DC μοτέρ από τις διαφορικές του εξισώσεις

Παρατηρώντας την εικόνα 39, βλέπουμε πως σε κάθε ανάδραση των τεσσάρων μοτέρ μας, υπάρχει μια διάταξη κέρδους, η οποία περιέχει την τιμή 0.0048 . Η χρησιμότητα του κέρδους αυτού είναι για να μεταβούμε από τα RPM ξανά σε V ώστε η σύγκριση και η νέα είσοδος μας, να είναι σε V. Η τιμή αυτή υπολογίστηκε ως εξής:

$$V_{-} = \frac{12 \omega_{\text{εξ}}}{2500}$$

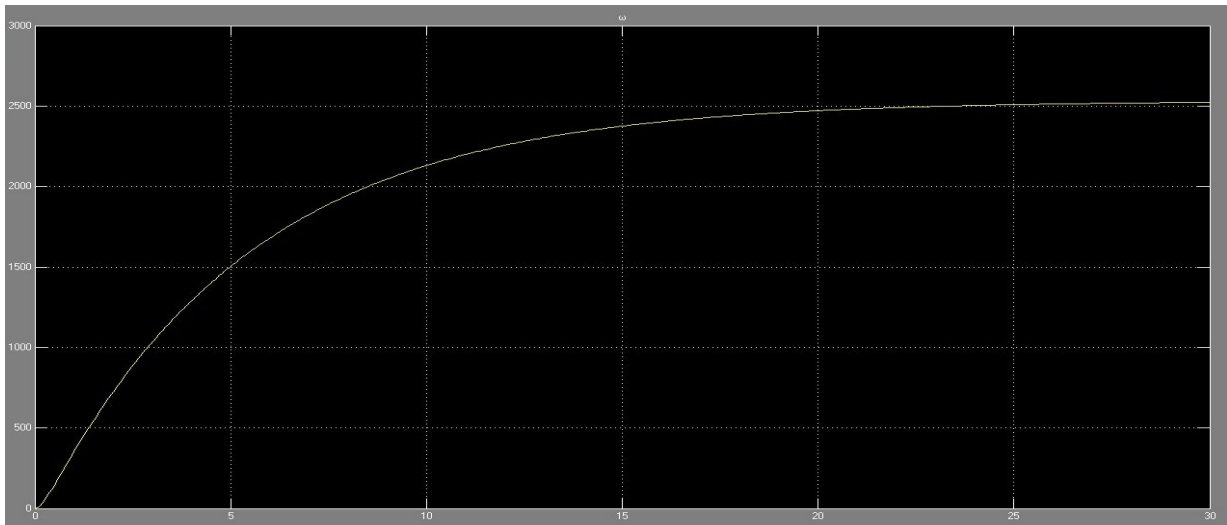
οι αριθμοί, 12 και 2500, είναι τα μέγιστα V και τα μέγιστα RPM αντιστοίχως.

Η απόκριση που επιθυμούμε να έχει το μοντέλο μας πρέπει να ικανοποιεί τις εξής τρεις συνθήκες:

1. Χρόνος αποκατάστασης μικρότερος των 2 δευτερολέπτων
2. Υπερύψωση μικρότερη από 5%
3. Μόνιμο σφάλμα μικρότερο από 1%

Ο λόγος που θέτουμε τις συνθήκες αυτές είναι πως εξαιτίας πιθανόν συχνών αλλαγών ταχύτητας, προκειμένου να υπάρχει κίνηση, θέλουμε τα μοτέρ μας να φτάνουν γρήγορα την επιθυμητή μας ταχύτητα και με όσο το δυνατόν μεγάλη ακρίβεια, με αποτέλεσμα την διατήρηση της ισορροπίας του quadcopter μας.

Δοκιμάζοντας το μοντέλο μας χωρίς έλεγχο, η απόκριση που παίρνουμε είναι η παρακάτω:



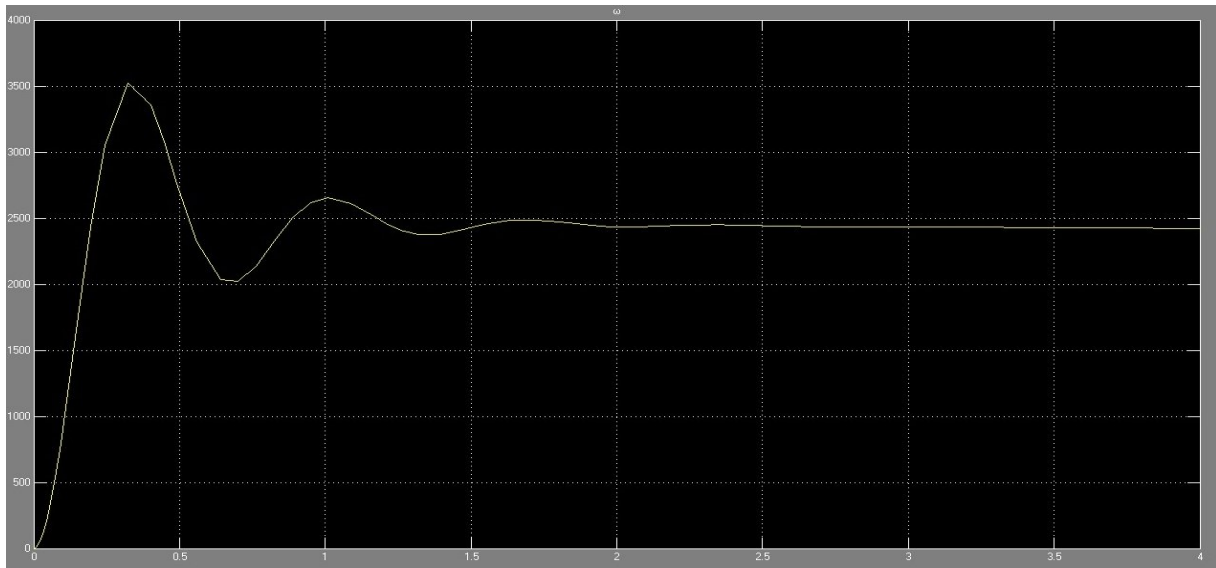
Εικόνα 42: Απόκριση του μοτέρ μας χωρίς εφαρμογή ελέγχου

Όπως μπορούμε να δούμε από την απόκριση μας, το μοτέρ μας παίρνει ως τελική τιμή την επιθυμητή ταχύτητα, αλλά έχει πολύ αργό χρόνο αποκατάστασης της τιμής αυτής. Για τον λόγο αυτό λοιπόν θα χρειαστεί να εισάγουμε ελεγκτή στο σύστημά μας, προκειμένου να βελτιώσουμε την απόκρισή του, και να ικανοποιήσουμε και τις τρεις συνθήκες μας. Επειδή δεν έχουμε κάποιο σημείο αστάθειας που θα μας βοηθούσε να υπολογίσουμε τις σταθερές K_p , K_i , K_D ενός ελεγκτή PID, θα δώσουμε τιμές στο αναλογικό κέρδος K_p και στην συνέχεια θα εξετάσουμε και πάλι την απόκριση που παίρνουμε.

Στο σημείο αυτό θυμίζουμε πως η συνάρτηση μεταφοράς ενός ελεγκτή PID έχει την παρακάτω μορφή:

$$C(s) = K_p + \frac{K_i}{s} + K_D s = \frac{K_D s^2 + K_p s + K_i}{s}$$

Μετά από δοκιμές τιμών παρατηρήθηκε πως για $K_p = 100$, το σύστημά μας είχε μια απόκριση που θα μας βοηθούσε να προχωρήσουμε και στην επιλογή τιμών για τα άλλα δυο κέρδη, αν κριθεί απαραίτητο.



Εικόνα 43: Απόκριση του μοτέρ μας με αναλογικό ελεγκτή $C(s) = 100$

Στην παραπάνω απόκριση βλέπουμε πως έχουμε καταφέρει να ικανοποιήσουμε μια από τις τρεις μας συνθήκες. Έχουμε ρίξει τον χρόνο αποκατάστασης στα 2 δευτερόλεπτα, που είναι πολύ σημαντικό. Όμως το πρόβλημα υπερύψωσης είναι αρκετά μεγάλο (40%) και επίσης έχουμε αρκετά μεγάλο μόνιμο σφάλμα. Από την θεωρία ξέρουμε πως αυξάνοντας το αναλογικό κέρδος μειώνουμε το μόνιμο σφάλμα, αλλά από την άλλη πλευρά αυξάνεται η υπερύψωση. Οπότε στην περίπτωση αυτή, δεν θα είναι αποδοτική μια τέτοια ενέργεια. Χωρίς καμία αμφιβολία λοιπόν, θα χρειαστούμε και τις άλλες 2 σταθερές του ελεγκτή μας. Προσθέτοντας την σταθερά ολοκλήρωσης K_i θα εξαλείψουμε το μόνιμο σφάλμα, ενώ με την σταθερά διαφόρισης K_D θα μειώσουμε σημαντικά την υπερύψωση.

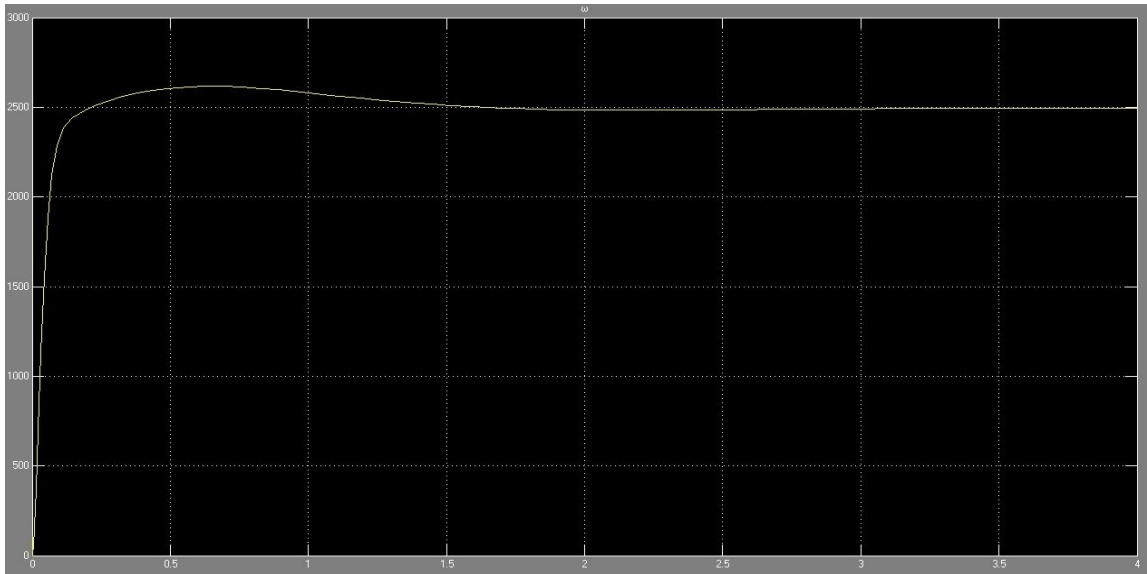
Έχοντας στο μυαλό μας τις 2 ακόμα συνθήκες μας αυξάνουμε σιγά – σιγά τις τιμές των σταθερών μας μέχρι να πετύχουμε ένα αποτέλεσμα το οποίο να μας τις ικανοποιεί. Μετά από αρκετές δοκιμές καταλήξαμε στις εξής τιμές:

$$K_p = 100, K_i = 200, K_D = 26.1$$

με συνάρτηση μεταφοράς:

$$C(s) = \frac{26.1s^2 + 100s + 200}{s}$$

Εφαρμόζοντας αυτόν τον ελεγκτή στο σύστημά μας η απόκριση που παίρνουμε είναι η εξής:



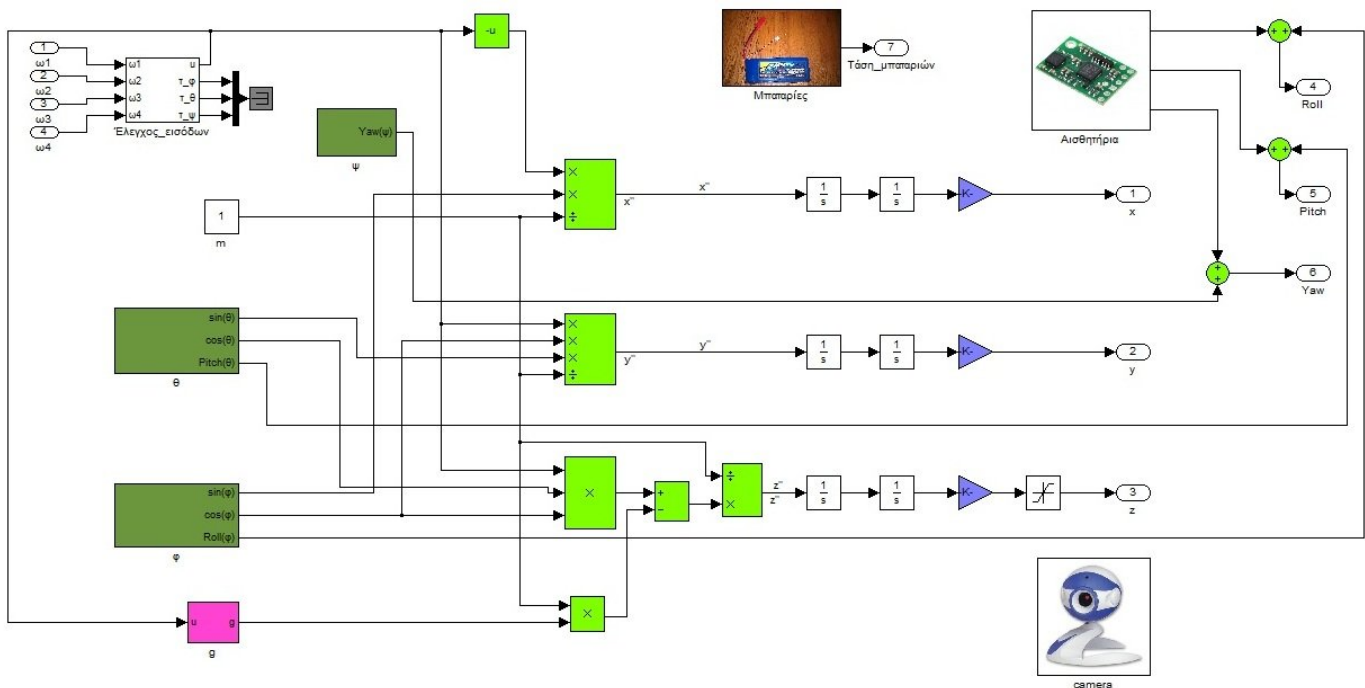
Εικόνα 44: Επιθυμητή απόκριση ενός DC μοτέρ με ελεγκτή PID

Βλέπουμε λοιπόν πως με τις τιμές αυτές στα κέρδη του ελεγκτή μας, η απόκρισή μας είναι απόλυτα ικανοποιητική. Ο χρόνος αποκατάστασης έχει μειωθεί αρκετά (1.7 δευτερόλεπτα), και αυτό είναι πολύ καλό για την απόδοση του μοντέλου μας. Επίσης, καταφέραμε να μειώσουμε αισθητά την υπερύψωσή μας στο 4.67% και να μηδενίσουμε το μόνιμο σφάλμα μας.

Ο έλεγχος των μοτέρ είναι πολύ σημαντικός για την ομαλή πτήση του quadcopter μας, γιατί ανά πάσα στιγμή πρέπει τα μοτέρ να είναι σε θέση να αλλάξουν ταχύτητα γρήγορα και αποτελεσματικά ώστε να μην υπάρξει πρόβλημα απώλειας της ισορροπίας του.

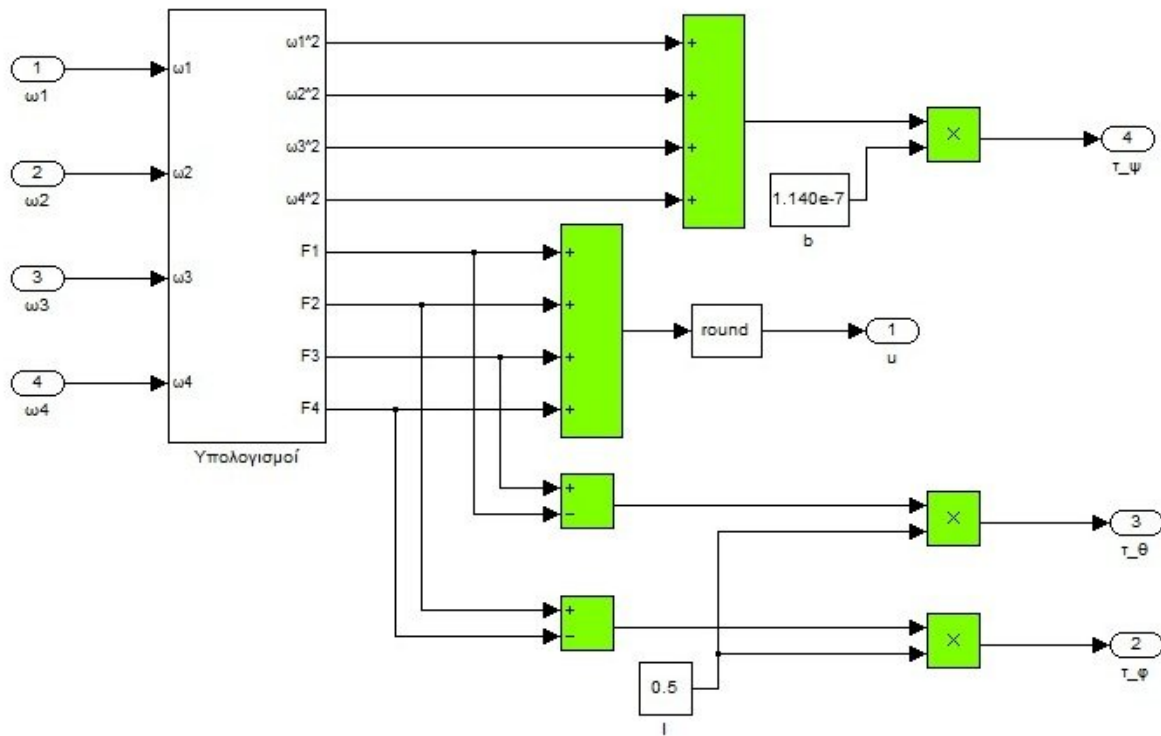
3.2.4 Μοντελοποίηση του Quadcopter

Έχοντας εξασφαλίσει την επιθυμητή απόκριση για τα DC μοτέρ μας, προχωράμε στην υλοποίηση του μοντέλου μας, βάσει των διαφορικών εξισώσεων του. Η υλοποίηση του μοντέλου μας βρίσκεται στην παρακάτω εικόνα (εσωτερικά του block με το πορτοκαλί χρώμα):



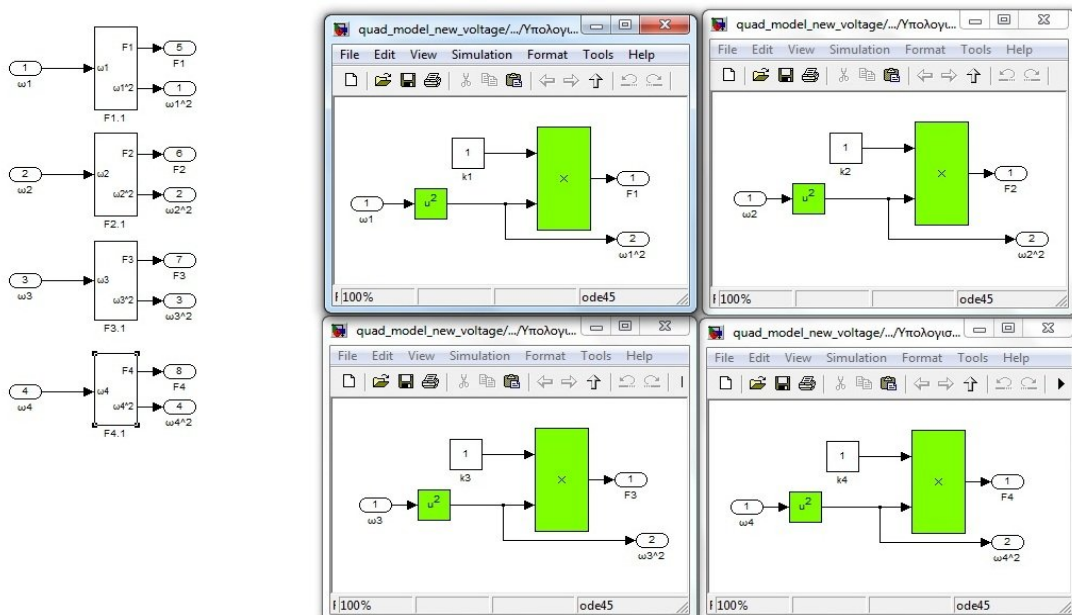
Εικόνα 45: Μοντελοποίηση ενός quadcopter μέσα από τις διαφορικές εξισώσεις του

Στο σημείο αυτό θα σταθούμε ώστε να γίνει αναλυτική επεξήγηση σχετικά με το τι ακριβώς περιέχουν κάποια από τα block (με τους τίτλους: θ , ϕ , έλεγχος_εισόδων, g). Πρώτα θα ασχοληθούμε με το block που έχει τίτλο, “Έλεγχος_εισόδων”. Στο εσωτερικό του block αυτού εκτελείται μια διαδικασία, η οποία ουσιαστικά μετατρέπει τις γωνιακές ταχύτητες των μοτέρ, σε χαρακτηριστικά της πτήσης του. Το εσωτερικό του block αυτού παρουσιάζεται στην παρακάτω εικόνα:



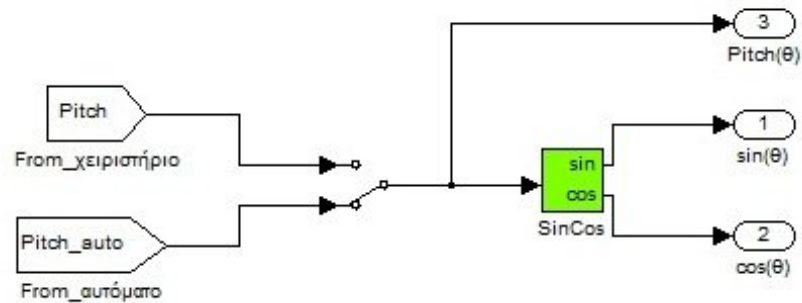
Εικόνα 46: Το εσωτερικό του block Έλεγχος_εισόδων

Στο block με τίτλο “Υπολογισμοί” (Εικόνα 46), γίνονται όλοι οι απαιτούμενοι υπολογισμοί προκειμένου να πάρουμε τις τιμές των γωνιών του Euler και της ανύψωσης του quadcopter. Στο εσωτερικό του block με τίτλο “Υπολογισμοί” βρίσκεται η διαδικασία του υπολογισμού.



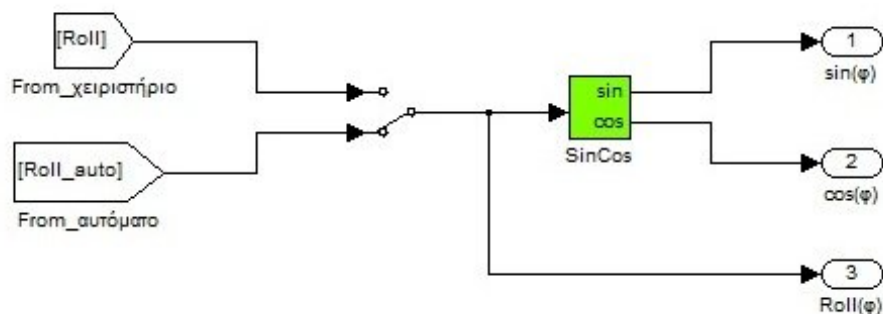
Εικόνα 47: Το εσωτερικό του block Υπολογισμοί

Το επόμενο block που μας ενδιαφέρει έχει τίτλο “θ” και αυτό γιατί στο εσωτερικό του γίνονται υπολογισμοί οι οποίοι παίρνουν στην είσοδο του block την γωνία θ , είτε έχουμε το μοντέλο μας στο χειριστήριο είτε στο αυτόματο και μας επιστρέφει το συνημίτονο και το ημίτονο της γωνίας αυτής, τα οποία τα χρησιμοποιούμε για τον υπολογισμό της θέσης του quadcopter μας.



Εικόνα 48: Το εσωτερικό του block θ

Με την ίδια ακριβώς λογική σχεδιάστηκε και το block με τίτλο “φ” το οποίο όπως και το block με τίτλο “θ”, παίρνει σαν είσοδο την γωνία ϕ , είτε έχουμε το μοντέλο μας στο χειριστήριο είτε στο αυτόματο και μας επιστρέφει το συνημίτονο και το ημίτονο της γωνίας αυτής, τα οποία τα χρησιμοποιούμε για τον υπολογισμό της θέσης του quadcopter μας.

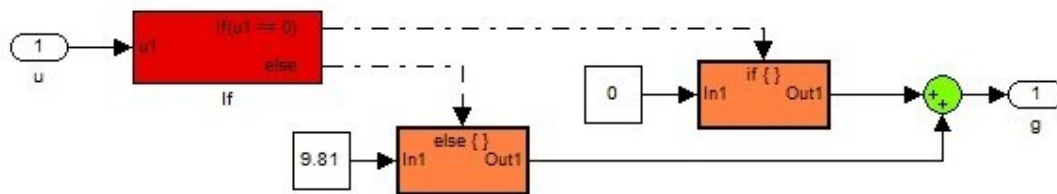


Εικόνα 49: Το εσωτερικό του block φ

Το τελευταίο block το οποίο έχει τίτλο “g” είναι βοηθητικό προκειμένου το αποτέλεσμα της προσομοίωσής μας να είναι πιο αληθοφανές, όταν αρχικά όλες οι

ταχύτητές μας στα μοτέρ είναι μηδέν, δεν πρέπει να υπάρχει κάποιο αποτέλεσμα στις εξόδους πέραν του μηδενός. Έτσι το block αυτό, διαβάζει την τιμή του u , το οποίο περιέχει και όλες τις ταχύτητες των μοτέρ στο εσωτερικό του, και αν το άθροισμά τους είναι μηδέν, τότε και το g θα είναι μηδέν, αλλιώς θα είναι 9,81.

Το εσωτερικό του block παρουσιάζεται παρακάτω:



Εικόνα 50: Το εσωτερικό του block g

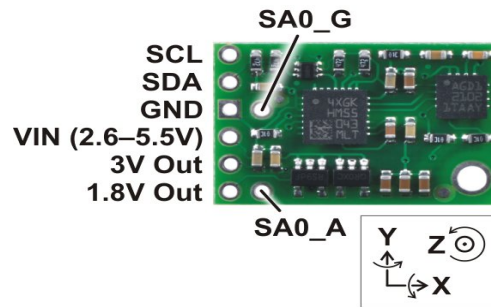
Το block μας συνολικά έχει 7 εξόδους: οι τρεις πρώτες αναφέρονται στις συντεταγμένες του quadcopter (x , y , z), οι αμέσως τρεις επόμενες αναφέρονται στις γωνίες του Euler roll(ϕ), pitch(θ), yaw(ψ) και η τελευταία μας έξοδος αναφέρεται στην τάση που έχουν οι μπαταρίες που είναι τοποθετημένες στο μοντέλο μας.

Για την λεπτομερέστερη απόκριση του συστήματός μας, ώστε να μπορέσουμε να αποκτήσουμε τον πλήρη έλεγχο του quadcopter, θα κάνουμε χρήση αισθητηρίων (γυροσκόπιο σε συνδυασμό με επιταχυνσιόμετρο) από τα οποία θα παίρνουμε μετρήσεις και θα τις καταχωρούμε στο μοντέλο που χρησιμοποιούμε ως δεδομένα επιβεβαίωσης θέσης του quadcopter (όχι στο επίπεδο, αλλά στους άξονές του). Η διαδικασία αυτή θα πραγματοποιηθεί στην επόμενη παράγραφο.

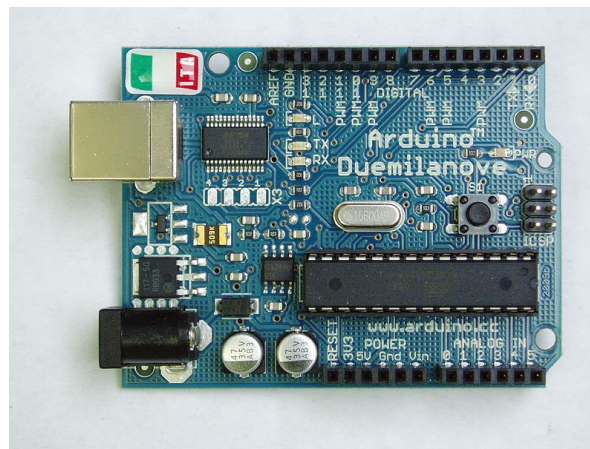
Τέλος, στην εικόνα 45, υπάρχει ένα block με τίτλο camera, στο εσωτερικό του οποίου με block που μας παρείχε το Simulink, κάναμε εισαγωγή των στοιχείων της κάμερας που ενώσαμε με τον υπολογιστή μας, ώστε να αποκτήσουμε εικόνα της πτήσης μας. Με το τέλος κάθε προσομοίωσης, το video που έχει τραβήξει η κάμερά μας αποθηκεύεται αυτόματα στον υπολογιστή μας, και μπορούμε να το αναπαράγουμε όποτε εμείς θελήσουμε.

3.2.5 Μετρήσεις μέσω αισθητηρίου

Το αισθητήριο που θα μας βοηθήσει να πάρουμε τις μετρήσεις μας είναι το “MiniMU-9” σε συνεργασία με τον μικροεπεξεργαστή arduino.



Εικόνα 51: Το αισθητήριο MiniMU-9



Εικόνα 52: Ο μικροεπεξεργαστής Arduino

Η συνδεσμολογία των 2 αυτών hardware είναι αρκετά απλή και εύκολη. Παρακάτω θα δούμε ποια pin του arduino πρέπει να συνδεσμολογίσουμε με τα pin του MiniMU-9.

5V -> VIN

GND -> GND

Analog Pin 5 -> SCL

Analog Pin 4 -> SDA

pin arduino -> pin MiniMU-9

Αφού εξασφαλίσουμε την σωστή συνδεσμολογία, είμαστε έτοιμοι να γράψουμε κώδικα για να τρέξουμε στον μικροεπεξεργαστή μας. Παρακάτω, παραθέτουμε επιλεκτικά κομμάτια κώδικα με μια σύντομη επεξήγηση.

Αναλόγως το αισθητήριο που χρησιμοποιούμε πρέπει να ορίσουμε τα όρια των σταθερών για το μαγνητόμετρο μας.

```
#define M_X_MIN -796
#define M_Y_MIN -457
#define M_Z_MIN -424
#define M_X_MAX 197
#define M_Y_MAX 535
#define M_Z_MAX 397
```

Εδώ αρχικοποιούμε έναν πίνακα του οποίου οι τιμές θα κάνουν διόρθωση στις κατευθύνσεις x, y, z.

```
int SENSOR_SIGN[9] = {1,1,1,-1,-1,-1,1,1,1}
```

Ο κύριος βρόγχος στον οποίο γίνονται όλοι οι υπολογισμοί και από τον οποίο θα τραβήξουμε ουσιαστικά με την βοήθεια του MATLAB τις μετρήσεις μας ώστε να τις εκχωρήσουμε στο μοντέλο μας είναι ο κάτωθι:

```
//Κύριος Βρόγχος
void loop()
{
  if((millis()-timer)>=20)
  {
```

```
counter++;

timer_old = timer;

timer=millis();

if (timer>timer_old)

    G_Dt = (timer-timer_old)/1000.0;

else

    G_Dt = 0;

Read_Gyro Read_Accel();

if (counter > 5)

{

    counter=0;

    Read_Compass();

    Compass_Heading();

}

// Υπολογισμοί

Matrix_update();

Normalize();

Drift_correction();

Euler_angles();

printdata();

}

}
```


Για να μπορέσουμε όμως να τραβήξουμε δεδομένα από το MinIMU-9 πρέπει να γράψουμε κώδικα στον MATLAB για να δημιουργήσουμε επικοινωνία ανάμεσα στο arduino και στο περιβάλλον του MATLAB. Στο σημείο αυτό, θα παρουσιαστεί ο κώδικας που δημιουργήθηκε στο MATLAB για να υπάρξει επικοινωνία με το arduino και στην συνέχεια απορρόφηση δεδομένων από το αισθητήριό μας.

```
s=serial('COM5','BaudRate',115200); %Χρησιμοποιούμε την COM port που είναι
                                     %συνδεδεμένο το arduino

fopen(s); %Ανοίγει την θύρα για να επιτευχθεί η επικοινωνία με το arduino
fscanf(s); %Εμφανίζεται το μήνυμα Serial.println("Pololu MinIMU-9 + Arduino
           %AHRs")

for x1=1:6
    v(x1)=fscanf(s,'%d');
end

clear v; clear x1; clear ans;

metrisi = fscanf(s,'%d'); %πρώτη μέτρηση των τριων μεταβλητών μας

for i=1:1000
    metrisi = metrisi(6:28); %"φιλτράρουμε" τις μετρήσεις μας
    m = str2num(metrisi); %το αποτέλεσμα αυτό το κάνουμε τρεις αριθμούς
    r(i) = m(1,1); %αποθήκευση τιμών για το roll
    p(i) = m(1,2); %αποθήκευση τιμών για το pitch
    y(i) = m(1,3); %αποθήκευση τιμών για το yaw
    metrisi = fscanf(s,'%d'); %παιρνουμε νέα μέτρηση
end

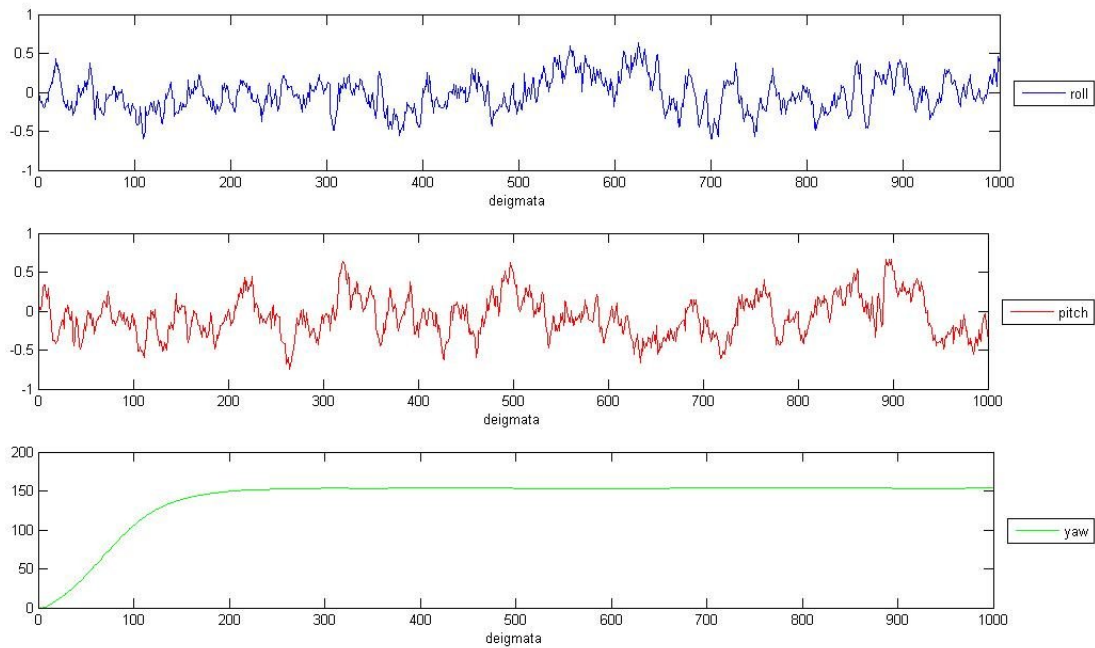
clear metrisi; clear i; clear m;

d = linspace(1,1000,1000); %Δημιουργία του οριζόντιου άξονα, δείγματα

%Γραφικές παραστάσεις των τριων μεταβλητών
subplot(3,1,1)
plot(d,r,'b')
xlabel('Δείγματα')
legend('roll','Location','EastOutside')
subplot(3,1,2)
plot(d,p,'r')
xlabel('Δείγματα')
legend('pitch','Location','EastOutside')
subplot(3,1,3)
plot(d,y,'g')
xlabel('Δείγματα')
legend('yaw','Location','EastOutside')
```

Κώδικας στο MATLAB για την απόκτηση δεδομένων μέσω του αισθητήριου

“Τρέχοντας” τον κώδικα αυτό στο περιβάλλον του MATLAB θα εμφανίζονται μετρήσεις μέχρι να ολοκληρωθεί η διαδικασία προσομοίωσης. Μόλις αυτή τελειώσει, θα εμφανιστεί ένα παραθύρο, το οποίο θα περιέχει τις τρεις γραφικές μας, μια για κάθε γωνία (roll, pitch, yaw).



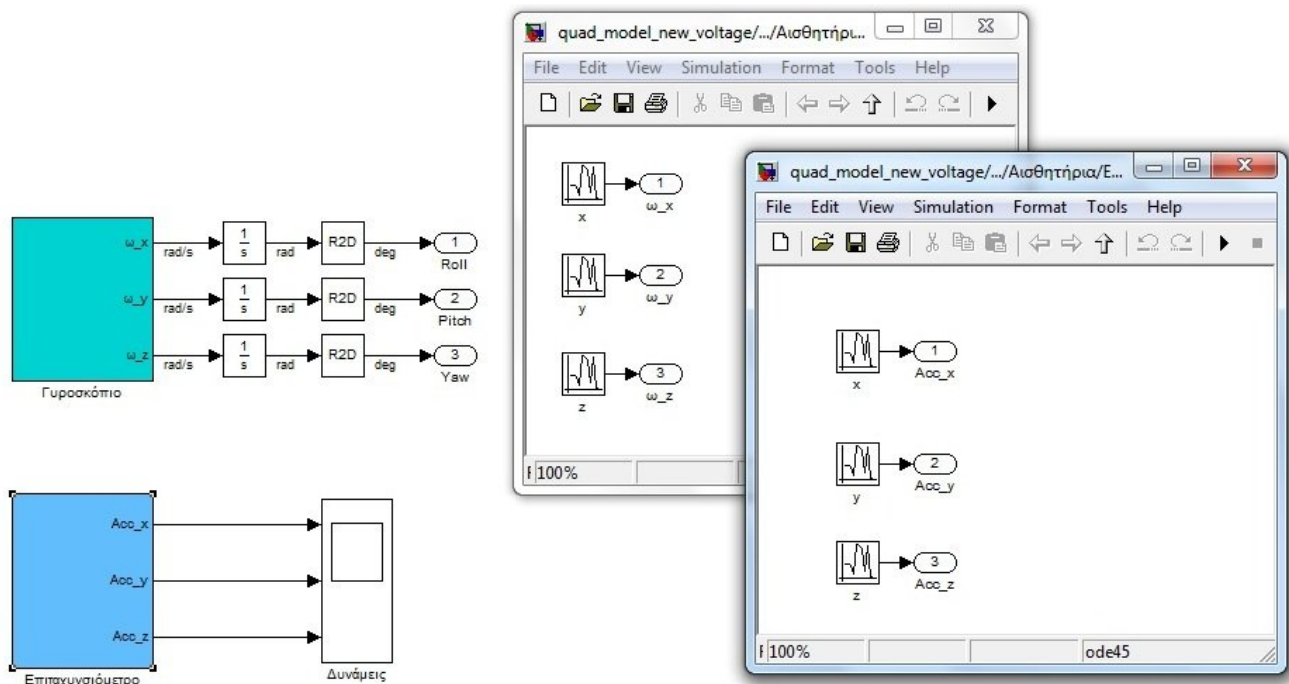
Εικόνα 53: Μετρήσεις των γωνιών Euler από το αισθητήριο

Φυσικά τις τιμές αυτές δεν μπορούμε να τις χρησιμοποιήσουμε στο μοντέλο μας γιατί είναι υπερβολικά ασταθείς, εξαιτίας του θορύβου που παράγει το ίδιο το αισθητήριο, και για τον λόγο αυτό οι μετρήσεις μας θα πρέπει να “φιλτραριστούν”.

3.2.6 Φιλτράρισμα των μετρήσεων

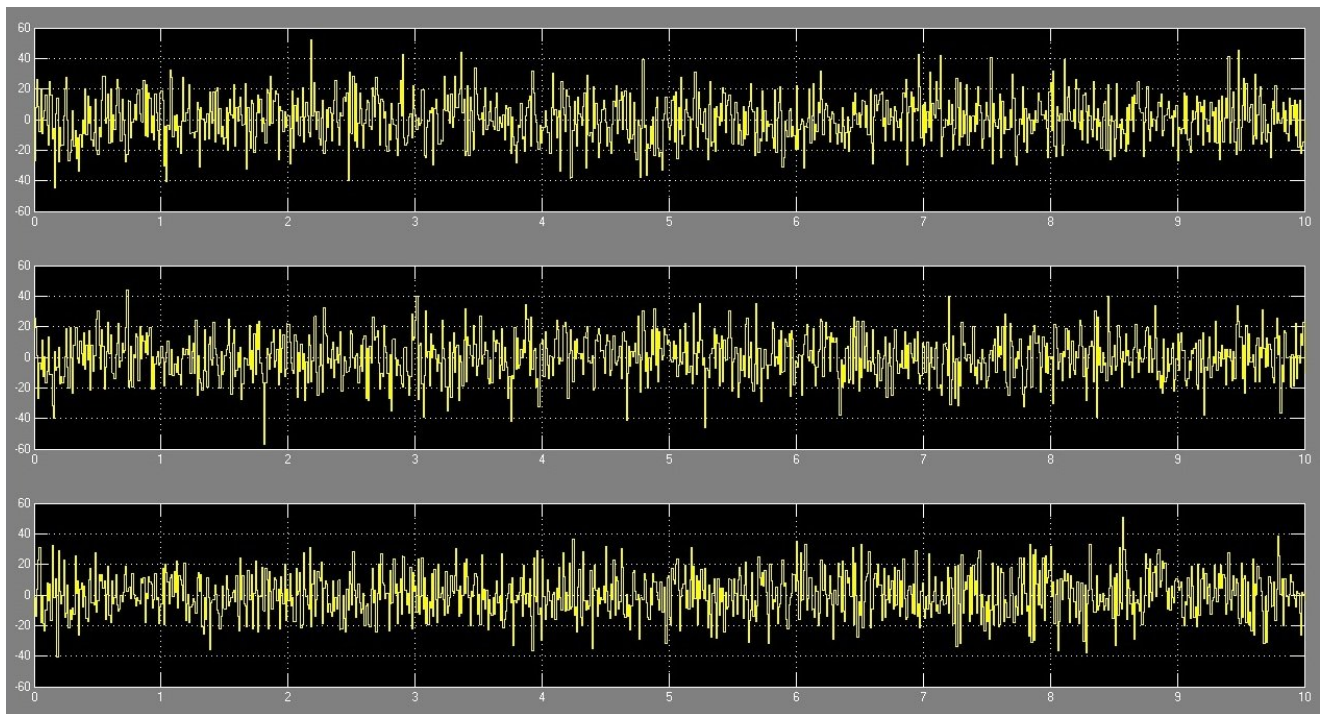
Βλέποντας την εικόνα 45(σελ.81) θα παρατηρήσατε ένα block με την εικόνα μιας μικρής πλακέτας το οποίο έχει τίτλο “Αισθητήρια”, με το οποίο δεν είχαμε ασχοληθεί μέχρι στιγμής. Στην ενότητα αυτή θα αναλύσουμε τα σήματα που παίρνουμε στην έξοδο των αισθητηρίων αυτών και με ποιον τρόπο αυτά θα “φιλτραριστούν” ώστε να έχουμε όσο το δυνατόν περισσότερη ομαλότητα στις κινήσεις του quadcopter μας.

Πίσω από το block αυτό βρίσκονται τα αισθητήριά μας (γυροσκόπιο – επιταχυνσιόμετρο) τα οποία μας δίνουν αποτελέσματα για τις γωνίες του Euler. Το αισθητήριο που επιτελεί τον υπολογισμό των γωνιών αυτών είναι ουσιαστικά το γυροσκόπιο, αλλά με την βοήθεια του επιταχυνσιόμετρου οι τιμές των γωνιών είναι πιο ακριβείς. Το εσωτερικό του block αυτού παρουσιάζετε παρακάτω:



Εικόνα 54: Το εσωτερικό του block Αισθητήρια καθώς επίσης και των block, Γυροσκόπιο – Επιταχυνσιόμετρο

Η έξοδος από το γυροσκόπιο μετατρέπεται σε μοίρες, αφού πρώτα περάσει από έναν ολοκληρωτή προκειμένου να μεταβούμε σε γωνία με δεδομένο την γωνιακή ταχύτητα. Η έξοδος του επιταχυνσιόμετρου μας δίνει ουσιαστικά μια δύναμη, ανάλογα με το που έχει κλίση του quadcopter μας. Τα αισθητήριά μας έχουν υλοποιηθεί με συναρτήσεις τυχαίων τιμών, με απότερω σκοπό, την παρατήρηση και επίτευξη διόρθωσης. Επίσης το συγκεκριμένο εγχείρημα θα αποτελέσει και μια ακραία κατάσταση διόρθωσης τιμών, η οποία θα μπορούσε να έχει αντίκρισμα και στο πραγματικό μας μοντέλο. Οι μετρήσεις που παίρνουμε από το επιταχυνσιόμετρο είναι οι εξής:

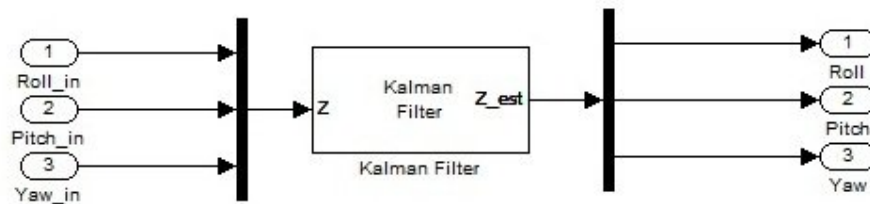


Εικόνα 55: Μετρήσεις επιταχυνσιόμετρου

Το αποτέλεσμα αυτό είναι λογικό καθώς όπως έχουμε ήδη αναφέρει πρόκειται για τυχαίες τιμές. Παρόμοιες είναι και οι μετρήσεις από το γυροσκόπιο, οι οποίες όμως θα παρουσιαστούν αργότερα μαζί με τις “φιλτραρισμένες” μετρήσεις, και να βγαλουμε τα συμπεράσματά μας.

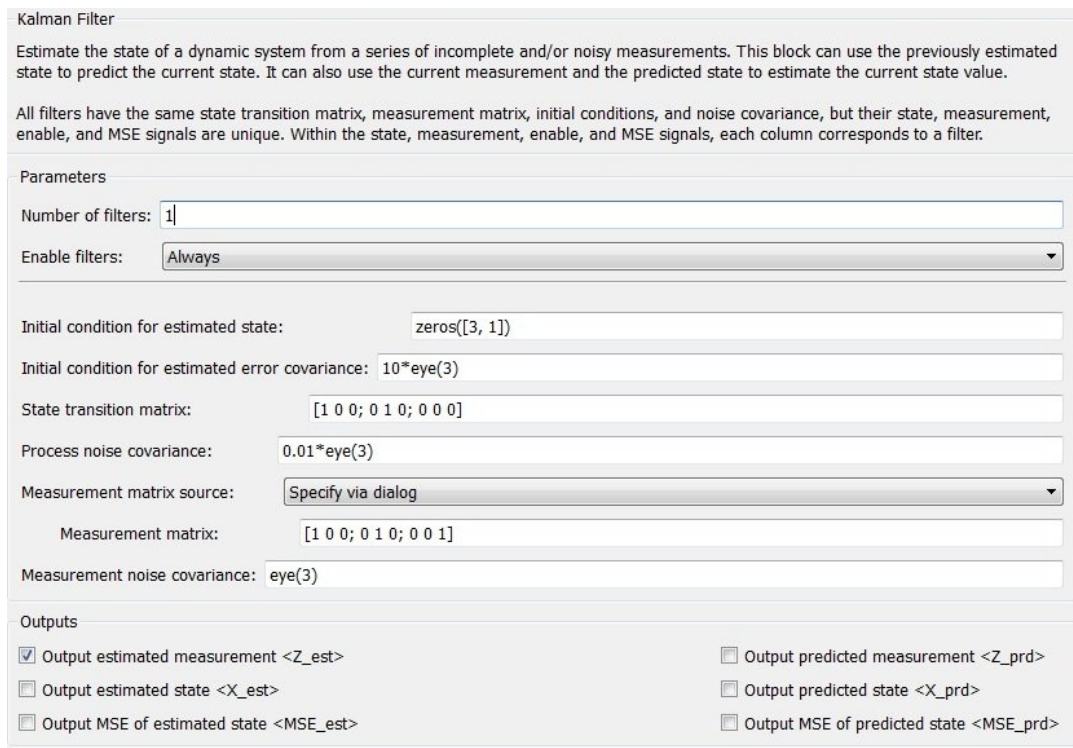
Έχοντας ως δεδομένο τέτοιου είδους μετρήσεις, μπορούμε εύκολα να καταλάβουμε πως πρέπει με κάποιον τρόπο αυτές να διορθωθούν, προκειμένου το μοντέλο μας, να έχει μια αρκετά καλή απόκριση και πορεία.

Σε αυτό εδώ το σημείο λοιπόν, θα μπει στο σύστημά μας, το φίλτρο Kalman. Πρόκειται για το πράσινο block της εικόνας 30(σελ.59). Το εσωτερικό του block με τίτλο “Φίλτρο_Kalman ” παρουσιάζεται παρακάτω:



Εικόνα 56: Το εσωτερικό του block “Φίλτρο_Kalman ”

Για να χρησιμοποιηθεί ένα μόνο φίλτρο, δημιουργούμε έναν πίνακα 3x1 με τα δεδομένα μας (τις τρεις γωνίες του Euler). Αφού περάσουν τα δεδομένα μας μέσα από το φίλτρο, τα διαχωρίζουμε εκ νέου. Στο φίλτρο μας έχουμε κάνει τις εξής ρυθμίσεις:



Εικόνα 57: Ρυθμίσεις του φίλτρου Kalman

Η παραπάνω εικόνα αποτελεί το περιβάλλον ρύθμισης ενός φίλτρου kalman. Εδώ βάζουμε στο φίλτρο μας τους απαραίτητους πίνακες ώστε να γίνει η διαδικασία βελτίωσης του σήματός μας όσο το δυνατόν πιο αποτελεσματική.

Οι πιθανές είσοδοι του block μας είναι τρεις:

1. Το σήμα μας, κατά προτίμηση σε πίνακα (στην περίπτωση μας πρόκειται για ένα πίνακα 3x1)
2. Η παράμετρος enable. Αυτή η παράμετρος καθορίζει σε περίπτωση που έχουμε και άλλα φίλτρα σε εφαρμογή, τότε πρέπει να ενεργοποιείται το καθένα. Υπάρχει και η επιλογή “always”, κατά την οποία το φίλτρο μας είναι πάντοτε ενεργό. Η παράμετρος αυτή στην δική μας περίπτωση είναι ορισμένη με την τιμή “always”. Η κατάσταση της παραμέτρου αυτής αλλάζει από το πεδίο “Enable filters”.
3. Η τρίτη και τελευταία παράμετρος είναι ο πίνακας μέτρησης (H), ο οποίος εισάγεται στο φίλτρο μας με δύο πιθανούς τρόπους. Ο πρώτος τρόπος είναι να γίνει η εισαγωγή του, εξωτερικά του φίλτρου, ενώ ο δεύτερος, ο οποίος χρησιμοποιείται στην μοντελοποίησή μας είναι να γίνει εισαγωγή του μέσα στο φίλτρο μας. Η κατάσταση της παραμέτρου αυτής αλλάζει από το πεδίο “Measurement matrix source”. Ο πίνακας μέτρησης που έχουμε δηλώσει στο φίλτρο μας είναι ο εξής:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

δηλαδή ο πίνακας αυτός αντιπροσωπεύει καθεμιά από τις μετρήσεις των συντεταγμένων μας (x, y, z).

Στο σημείο αυτό ας θυμηθούμε κάποια πράγματα από την θεωρία του φίλτρου μας ώστε να μπορέσουμε να εξηγήσουμε την δήλωση των πινάκων που έχουμε πραγματοποιήσει.

Οι γραμμικές εξισώσεις που περιγράφουν το φίλτρο μας είναι οι εξής:

$$\bar{X}_t = Ax_{t-1} + Bu_t + \varepsilon_x$$

με μια πρόβλεψη μέτρησης $z \in \mathcal{R}^n$:

$$\bar{Z}_t = C\bar{X}_t + \varepsilon_z$$

με: $\varepsilon_x, \varepsilon_z$: Gaussian errors

\bar{x}_t : πρόβλεψη κατάστασης

\bar{z}_t : πρόβλεψη μέτρησης

Στις παραπάνω εξισώσεις θα γίνουν οι εξής αλλαγές:

1. Ο όρος Bu_t θα αφαιρεθεί από την εξίσωση μας, εφόσον τα δεδομένα μας δεν περιέχουν επιτάχυνση. Οπότε η πρώτη εξίσωση θα απλοποιηθεί:

$$\bar{X}_t = Ax_{t-1} + \varepsilon_x$$

2. Ο πίνακας C , θα αντικατασταθεί από τον πίνακα H , ώστε να μην μπερδευτούμε. Άρα:

$$\bar{Z}_t = H\bar{X}_t + \varepsilon_z$$

Επίσης από την θεωρία θα χρειαστούμε τις παρακάτω εξισώσεις:

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^T + E_x$$

$$K = \bar{\Sigma}_t H^T (H\bar{\Sigma}_t H^T + E_z)^{-1}$$

$$\Sigma_t = (I - KH) \bar{\Sigma}_t$$

Έχοντας συγκεντρώσει τις παραπάνω εξισώσεις, μπορούμε να προχωρήσουμε στην επεξήγηση των πινάκων που έχουμε δηλώσει στο block μας.

Στο πεδίο “Initial condition for estimated state” βάζουμε την αρχική τιμή που θέλουμε να έχει η εκτίμησή μας. Στο φίλτρο μας, επειδή τα δεδομένα μας είναι ένας πίνακας 3×1 βάλουμε την τιμή: `zeros([3, 1])`, η οποία δημιουργεί έναν πίνακα 3×1 :

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Το επόμενο πεδίο “Initial condition for estimated error covariance” πρόκειται για το E_x που βλέπουμε στην τρίτη εξίσωσή μας. Μετά από δοκιμές για την εύρεση της κατάλληλης τιμής (πειραματικά), καταλήξαμε στην τιμή: $10 \cdot \text{eye}(3)$, η οποία δημιουργεί έναν πίνακα 3×3 :

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

Στο επόμενο κατά σειρά πεδίο “State transition matrix” πρέπει να δώσουμε τις διαστάσεις και την τιμή του πίνακα A της πρώτης μας εξίσωσης. Όπως και με τον πίνακα H , έτσι και με τον πίνακα A , δηλώσαμε για κάθε μέτρηση μια κατάσταση και έτσι ο πίνακάς μας θα έχει την παρακάτω τιμή:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Το επόμενο πεδίο “Process noise covariance” είναι πολύ καθοριστικό για τον βαθμό αποτελεσματικότητας του φίλτρου μας. Η τιμή που έχει δοθεί στο πεδίο αυτό είναι αποτέλεσμα αρκετών δοκιμών. Κατά τις δοκιμές παρατηρήθηκε πως όσο πιο μικρή τιμή βάζαμε στο πεδίο αυτό, τόσο καλύτερη γινόταν η απόδοση του φίλτρου μας. Έτσι ως τελική και ικανοποιητική τιμή ορίστηκε η: $0.01 \cdot \text{eye}(3)$, η οποία σαν αποτέλεσμα μας δίνει έναν πίνακα 3×3 :

$$\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

Το επόμενο πεδίο “Measurement matrix source” είναι ο πίνακας H , ο οποίος έχει αναλυθεί λεπτομερώς.

Τέλος, το τελευταίο πεδίο “Measurement noise covariance” λειτουργεί σε συνδυασμό με το πεδίο “Process noise covariance”, και για τον λόγο αυτό έχουμε αφήσει το πεδίο αυτό σταθερό, για να κάνουμε ό,τι βελτιώσεις θέλουμε από το πεδίο “Process noise covariance”. Η τιμή που έχουμε ορίσει στο πεδίο αυτό είναι:

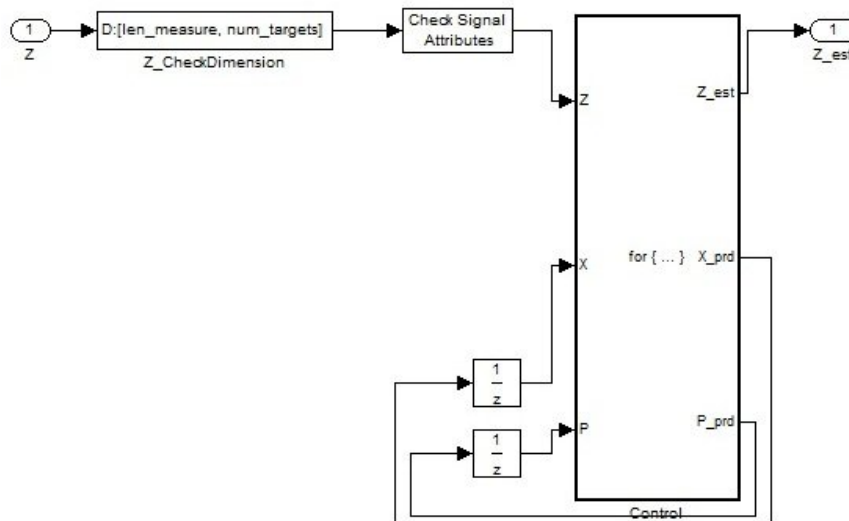
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Υπάρχουν επίσης έξι πιθανές έξοδοι για το φιλτραρισμένο σήμα μας, τις οποίες θα αναλύσουμε αμέσως.

1. Η πρώτη πιθανή έξοδος “Output estimated measurement $\langle Z_{est} \rangle$” μας επιστρέφει την εκτίμηση της μέτρησης. Στο φίλτρο μας έχουν επιλέξει αυτή την έξοδο.
2. Η δεύτερη πιθανή έξοδος “Output estimated state $\langle X_{est} \rangle$” μας επιστρέφει την εκτίμηση της κατάστασης εισόδου. Έγινε δοκιμή του αποτελέσματος και με την έξοδο αυτή, και το αποτέλεσμα ήταν το ίδιο με αυτό της προηγούμενης εξόδου.
3. Η τρίτη πιθανή έξοδος “Output MSE of estimated state $\langle MSE_{est} \rangle$” μας επιστρέφει το μέσο σφάλμα της εκτίμησης της κατάστασης εισόδου.
4. Η τέταρτη πιθανή έξοδος “Output predicted measurement $\langle Z_{prd} \rangle$” μας επιστρέφει την πρόβλεψη της μέτρησης.
5. Η πέμπτη πιθανή έξοδος “Output predicted state $\langle X_{prd} \rangle$” μας επιστρέφει την πρόβλεψη της κατάστασης εισόδου.
6. Η έκτη πιθανή έξοδος “Output MSE of predicted state $\langle MSE_{prd} \rangle$” μας επιστρέφει το μέσο σφάλμα της πρόβλεψης της κατάστασης εισόδου.

Μπορούμε να επιλέξουμε όποια από τις εξόδους θέλουμε για να δούμε το αντίστοιχο αποτέλεσμα, και επιπλέον μπορούμε να επιλέξουμε περισσότερες τις μιας εξόδου. Επίσης μια ακόμη χρήσιμη πληροφορία είναι πως όλες οι έξοδοι έχουν τις ίδιες διαστάσεις (πίνακα) με αυτές της εισόδου που έχουμε εισάγει στο φίλτρο μας.

Η διαδικασία που τρέχει πίσω από φίλτρο μας είναι η εξής:



Εικόνα 58: Το εσωτερικό του φίλτρου Kalman

3.2.7 Αποτελέσματα προσομοίωσης

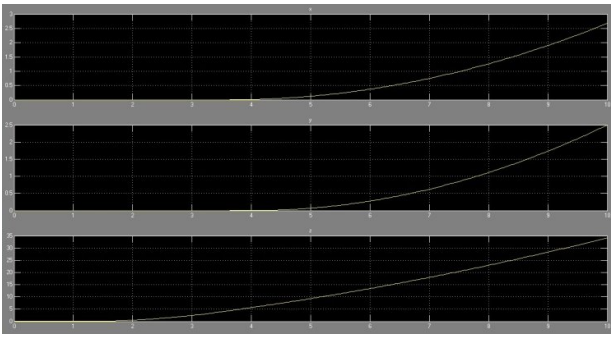
Στην παράγραφο αυτή, θα εξετάσουμε το πως αποκρίνεται το μοντέλο που αναπτύξαμε, κάνοντας εφαρμογή χειροκίνητης λειτουργίας και αυτόματης λειτουργίας. Με τις τιμές των ω_1 , ω_2 , ω_3 , ω_4 υπολογίζουμε την τιμή της ανύψωσης του quadcopter (u) και τις τιμές των γωνιών του Euler, $\text{roll}(\phi)$, $\text{pitch}(\theta)$, $\text{yaw}(\psi)$. Αρχικά “τρέξαμε” το μοντέλο για απλές περιπτώσεις όπως για παράδειγμα την περίπτωση ανύψωσης – κατάβασης του quadcopter και των ευθύγραμμων μετατοπίσεων του, μπροστά – πίσω και δεξιά – αριστερά. Εφόσον όλα τα αποτελέσματα ήταν ικανοποιητικά, επόμενό μας βήμα και βασικός μας στόχος ήταν να εξετάσουμε το μοντέλο αυτό σε περιπτώσεις παράλληλης κίνησης, για παράδειγμα κίνηση δεξιά και μπροστά. Στο σημείο αυτό η εικόνα του block DC_motors θα μας βοηθήσει σε αρκετά μεγάλο βαθμό να καταλάβουμε για ποια κίνηση εκτελείται κάθε φορά.

– Απόκριση χειροκίνητης λειτουργίας:

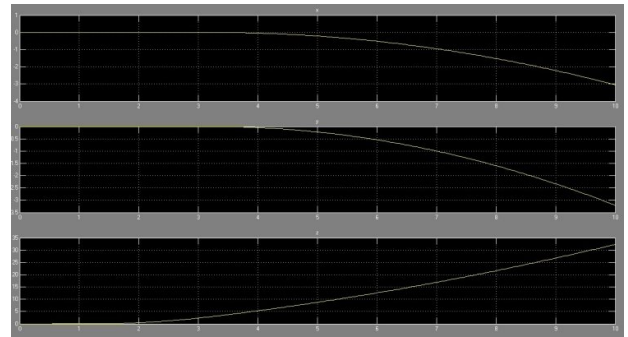
Οι περιπτώσεις που θα εξετάσουμε στην χειροκίνητη λειτουργία του μοντέλου μας είναι οι εξής τέσσερις:

1. Διαγώνια κίνηση, επάνω και δεξιά
2. Διαγώνια κίνηση, επάνω και αριστερά
3. Διαγώνια κίνηση, κάτω και αριστερά
4. Διαγώνια κίνηση, κάτω και δεξιά

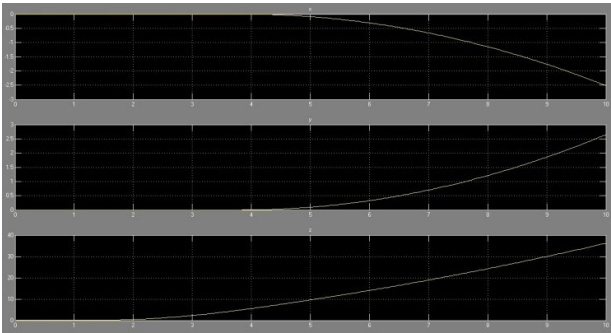
“Τρέχοντας” το μοντέλο για καθεμιά από τις παραπάνω περιπτώσεις, λαμβάνουμε μια απόκριση (για κάθε περίπτωση).



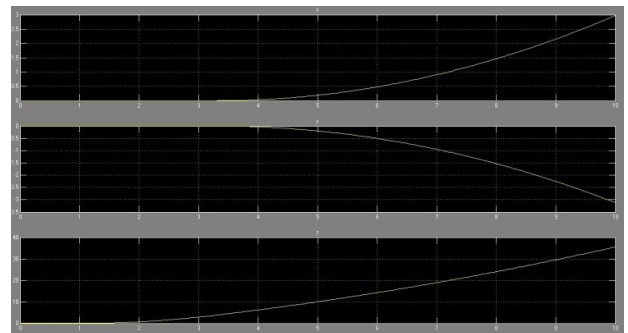
Εικόνα 59: Απόκριση θέσης για την περίπτωση 1



Εικόνα 61: Απόκριση θέσης για την περίπτωση 3



Εικόνα 60: Απόκριση θέσης για την περίπτωση 2



Εικόνα 62: Απόκριση θέσης για την περίπτωση 4

Έχοντας στο μυαλό μας την εικόνα ενός καρτεσιανού επιπέδου, μπορούμε αμέσως να καταλάβουμε πως οι αποκρίσεις που έχουμε παραπάνω για τις τέσσερις περιπτώσεις είναι απόλυτα επιθυμητές. Και με τους δυο τρόπους χειροκίνητης λειτουργίας που έχουμε αναπτύξει, το αποτέλεσμα των αποκρίσεών είναι ακριβώς το ίδιο, αποτέλεσμα το οποίο εννοείται πως επιδιώκαμε.

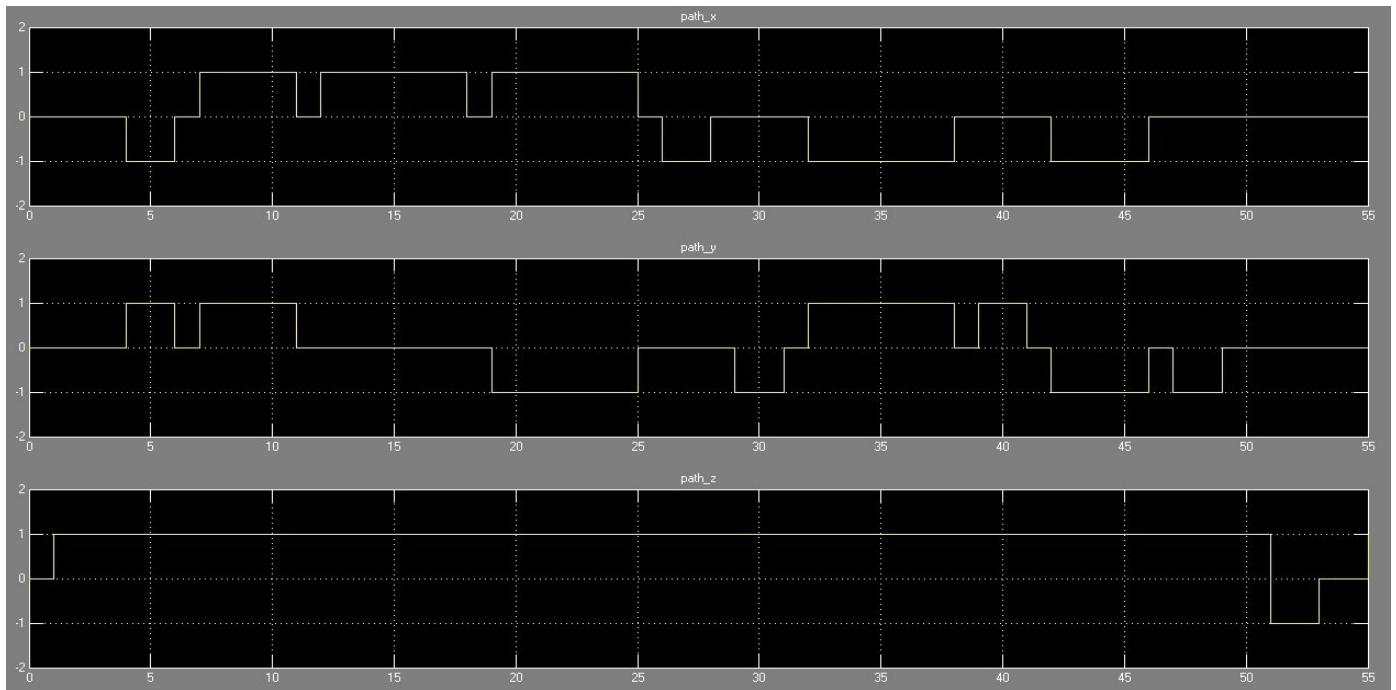
Οι αποκρίσεις των γωνιών μας σε συνδυασμό με το φίλτρο kalman ήταν εξίσου σωστές και ικανοποιητικές, θα παρουσιαστούν όμως στα αποτελέσματα αυτόματης λειτουργίας, κατά την οποία εμφανίζουν περισσότερο ενδιαφέρον.

– Απόκριση αυτόματης λειτουργίας:

Για την παρουσίαση της απόκρισης αυτής, πρέπει να σημειώσουμε πως σε αντίθεση με την προηγούμενη απόκρισή μας, χρειάστηκε να κάνουμε κάποιες προσθήκες στο μοντέλο του quadcopter μας. Πρόκειται για την προσθήκη δυο αναδράσεων μεταξύ των δυο ολοκληρωτών μας που επιστρέφουν τις συντεταγμένες

του μοντέλου μας. Αυτή η προσθήκη έγινε για να παρατηρούμε την μεταβολή θέσης του quadcopter μας.

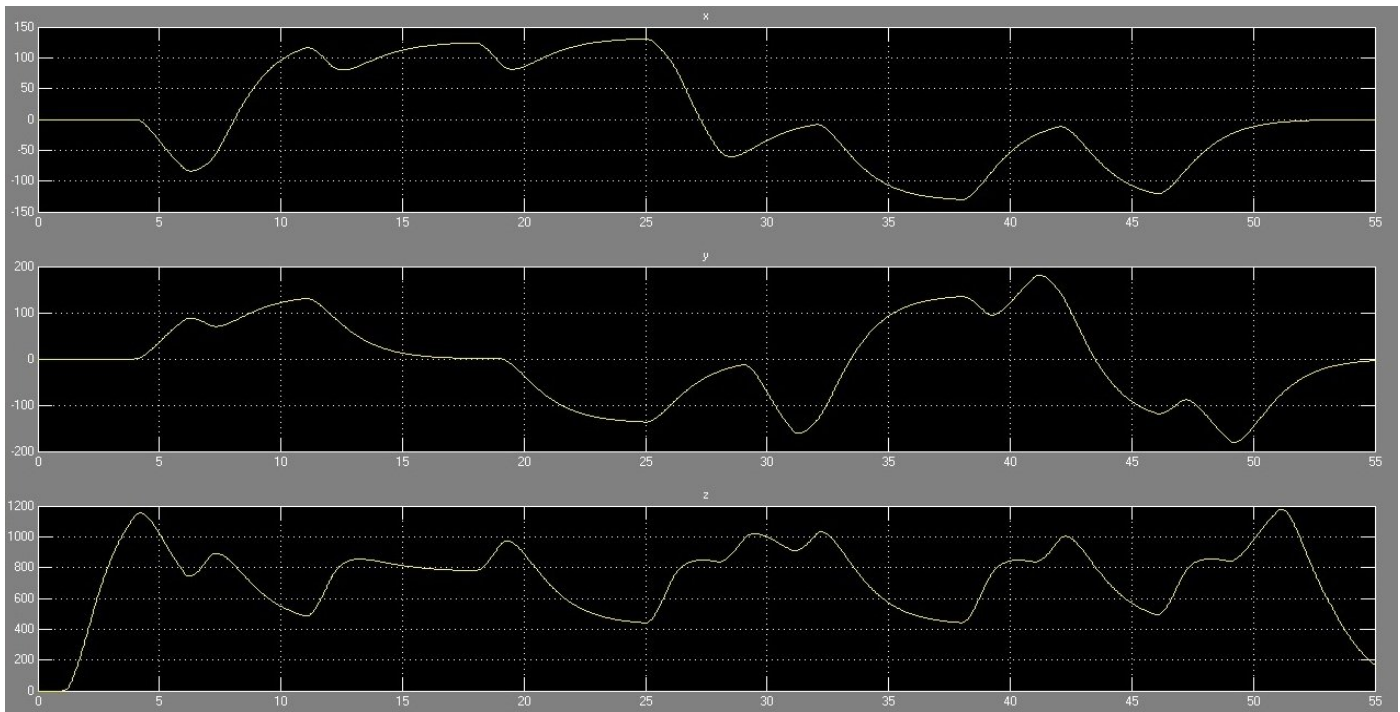
Είσοδος της αυτόματης λειτουργίας μας θα είναι ένα μονοπάτι, το οποίο έχουμε κατασκευάσει εμείς, και έχει την εξής μορφή:



Εικόνα 63: Μονοπάτι εισόδου

Όπως βλέπουμε πρόκειται για μια σειρά από παλμούς, οι οποίοι χωρίς να επέμβουμε εμείς, θα αλλάζει την διανομή τάσης σύμφωνα με τις ρυθμίσεις που έχουμε κάνει στον ελεγκτή ασαφούς λογικής, προκειμένου να κινηθεί το μοντέλο μας.

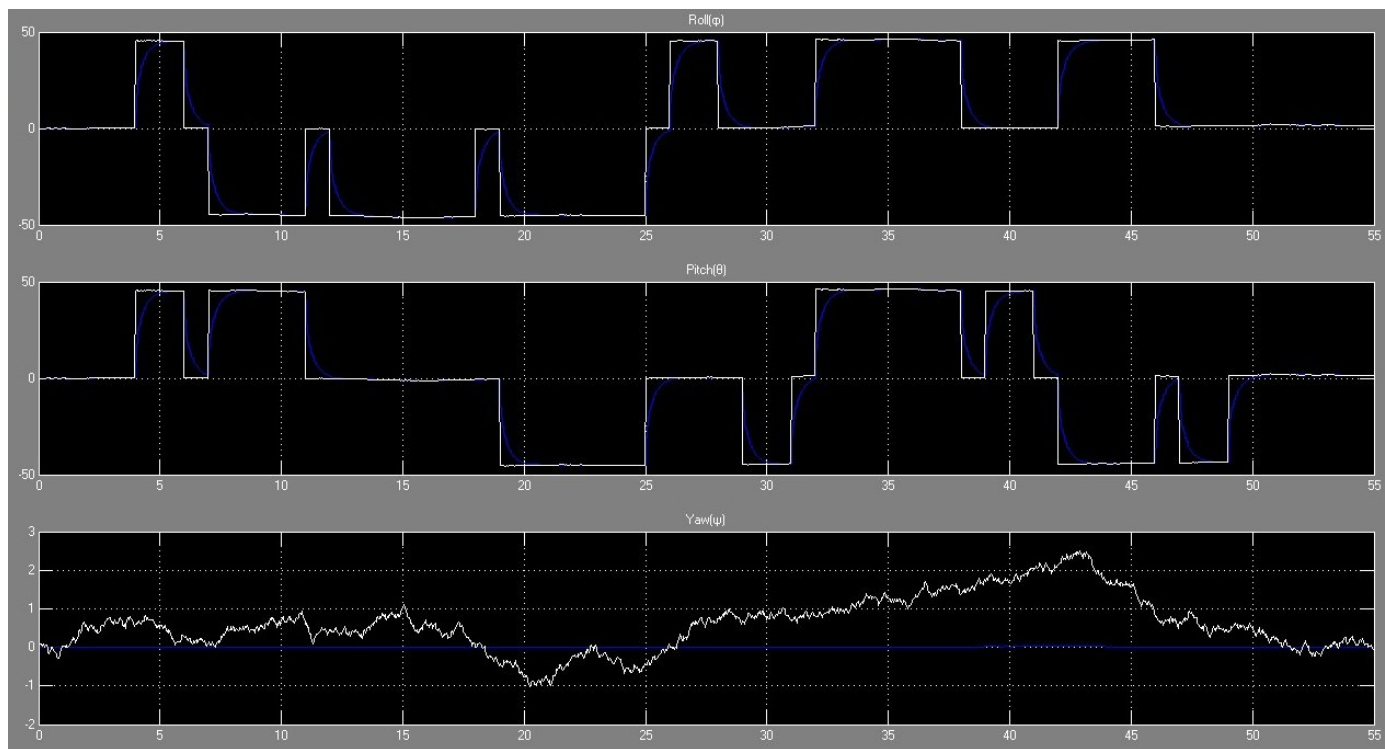
“Τρέχοντας” λοιπόν το μοντέλο μας με δεδομένο το τυχαίο μονοπάτι που έχουμε θέσει, το αποτέλεσμα που θα πάρουμε είναι το εξής:



Εικόνα 64: Απόκριση θέσης του μοντέλου μας σε αυτόματη λειτουργία

Παρόλο που η απόκριση αυτή μοιάζει να μην μας ικανοποιεί, εξαιτίας των αλλαγών στις τιμές των συντεταγμένων μας, βλέποντας την είσοδό μας κομμάτι-κομμάτι, βλέπουμε πως μέχρι να ολοκληρωθεί η απόκρισή μας, έχουμε συνολικά 9 αλλαγές διανομής τάσης στα μοτέρ, για κίνηση. Οπότε η απόκριση αυτή, είναι φυσικό να αλλάζει τιμές ώστε να βλέπουμε στην συνέχεια ποια είναι η κίνηση που ακολουθεί. Η μοναδική απόκριση που ίσως μας ανησυχεί είναι αυτή του άξονα z διότι φαίνεται πως το quadcopter μας, δεν διατηρεί το ύψος στο οποίο είχε φτάσει. Όμως πρόκειται για μια απολύτως λογική συμπεριφορά, καθώς κατά την κίνηση του quadcopter οι δυνάμεις που δημιουργούν την ανύψωση δρουν υπό γωνία με αποτέλεσμα η τιμή τους να είναι μικρότερη από αυτή της κάθετης θέσης, με αποτέλεσμα ο μηχανισμός μας, να χάνει μέρος από το ύψος στο οποίο είχε φτάσει. Αυτό το φαινόμενο, αποδίδεται πολύ καλά στην παραπάνω απόκρισή μας και το τελικό αποτέλεσμα μας αφήνει ικανοποιημένους. Μια τελευταία παρατήρηση που πρέπει να κάνουμε για την απόκρισή μας είναι πως επιστρέφει την μεταβολή των συντεταγμένων του quadcopter μας, και όχι την ακριβή θέση του.

Η απόκριση των γωνιών μας μετά την αυτόματη λειτουργία του μοντέλου μας, είναι η εξής:



Εικόνα 65: Αποτελέσματα μετρήσεων των γωνιών μας, πριν (με άσπρο) και μετά (με μπλε) το φίλτρο Kalman

Όπως βλέπουμε από την απόκρισή των γωνιών μας, οι γωνίες Roll και Pitch παρόλο που έχουν διαταραχές εξαιτίας των αισθητηρίων μας (άσπρη απόκριση), με την εφαρμογή και ρύθμιση του φίλτρου kalman το τελικό σήμα μας (μπλε απόκριση), φτάνει στην τελική μας επιθυμητή τιμή, προοδευτικά και έχοντας αποβάλει σε μεγάλο βαθμό τις διαταραχές από τα αισθητήριά μας. Όσο για την γωνία Yaw το φίλτρο μας, κάνει άριστη δουλειά και ενώ βλέπουμε πως στην είσοδο του φίλτρου μας έχουμε ένα πολύ κακό σήμα, στην έξοδο του παίρνουμε το απόλυτα επιθυμητό, το οποίο πρέπει να είναι σταθερό στις 0° , ώστε το μοντέλο μας να διατηρείται σταθερό στο ύψος που θα φτάσει.

Ως τελική παρατήρηση συνολικά για την απόκριση του μοντέλου μας κατά την αυτόματη λειτουργία του, είδαμε πως η μετάβαση από την μια συντεταγμένη στην επόμενη γίνεται με ομαλό τρόπο, παρόλο που αλλάζουμε σχετικά γρήγορα τις

συντεταγμένες μας και όσο για την απόκριση των γωνιών μας, παρόλο που οι μετρήσεις από τα αισθητήριά μας ήταν αρκετά κακές, και με αρκετό θόρυβο, είναι οφθαλμοφανές πως με το φίλτρο Kalman οι μετρήσεις μας πλέον είναι αρκετά καλύτερες και πολύ πιο σταθερές στην διακύμανσή τους, με εντυπωσιακότερη εξ'αυτών την πολύ εύστοχη διατήρηση της γωνίας Yaw κοντά στις 0°.

Κεφάλαιο 4: Γενικές παρατηρήσεις και συμπεράσματα

4.1 Παρατηρήσεις κατά την διεξαγωγή των πειραμάτων

Στο κεφάλαιο 3 έγινε η παρουσίαση της μοντελοποίησης του συστήματός μας στο MATLAB – Simulink. Πριν όμως φτάσουμε στο τελικό μοντέλο μας, σε κάποια από τα στάδια είχαν αναπτυχθεί και με διαφορετικό τρόπο, ο οποίος είχε μεν την επιθυμητή απόδοση που εμείς θα θέλαμε, αλλά από άποψη υλοποίησης και εφαρμογής σε ένα πραγματικό μοντέλο, δεν ήταν ρεαλιστικός.

4.1.1 Ανάπτυξη εφεδρικού μοντέλου για τα DC μοτέρ

Πριν εφαρμόσουμε τα μοτέρ που παρουσιάστηκαν στο κεφάλαιο 3, είχαμε αναπτύξει μοτέρ τα οποία βασίζονταν στην συνάρτηση μεταφοράς τους. Ο υπολογισμός της συνάρτησης μεταφοράς του μοντέλου μας, γίνεται από τις διαφορικές εξισώσεις του μοτέρ. Οπότε:

$$1. J \frac{d\omega(t)}{dt} + b \omega(t) = K_t i(t)$$

$$2. L \frac{di(t)}{dt} + R i(t) = V - K_e \omega(t)$$

Εφαρμόζοντας μετασχηματισμό κατά Laplace στις παραπάνω εξισώσεις παίρνουμε:

$$1. Js \omega(s) + b \omega(s) = K I(s) \Rightarrow I(s) = \frac{(Js+b)\omega(s)}{K}$$

$$2. Ls I(s) + R I(s) = V(s) - K \omega(s) \Rightarrow$$

$$(Ls + R) \frac{(Js+b)\omega(s)}{K} = V(s) - K \omega(s) \Rightarrow$$

$$(Ls + R)(Js + b) \omega(s) = K V(s) - K^2 \omega(s) \Rightarrow$$

$$[(Ls + R)(Js + b) + K^2] \omega(s) = K V(s)$$

Στο σημείο αυτό δημιουργώντας το πηλίκο $\frac{\omega(s)}{V(s)}$, βρίσκουμε την συνάρτηση μεταφοράς του μοτέρ μας. Οπότε:

$$\frac{\omega(s)}{V(s)} = \frac{K}{(Ls + R)(Js + b) + K^2}$$

Έχοντας βρει την συνάρτηση μεταφοράς μπορούμε να βάλουμε τιμές στις μεταβλητές μας για να δημιουργήσουμε το μοτέρ μας. Παρακάτω παρουσιάζεται ο κώδικας υλοποίησης και υπολογισμού της συνάρτησης μεταφοράς μας στο MATLAB.

Υλοποιώντας τις εξισώσεις αυτές στο MATLAB – Simulink πήραμε το παρακάτω μοντέλο:

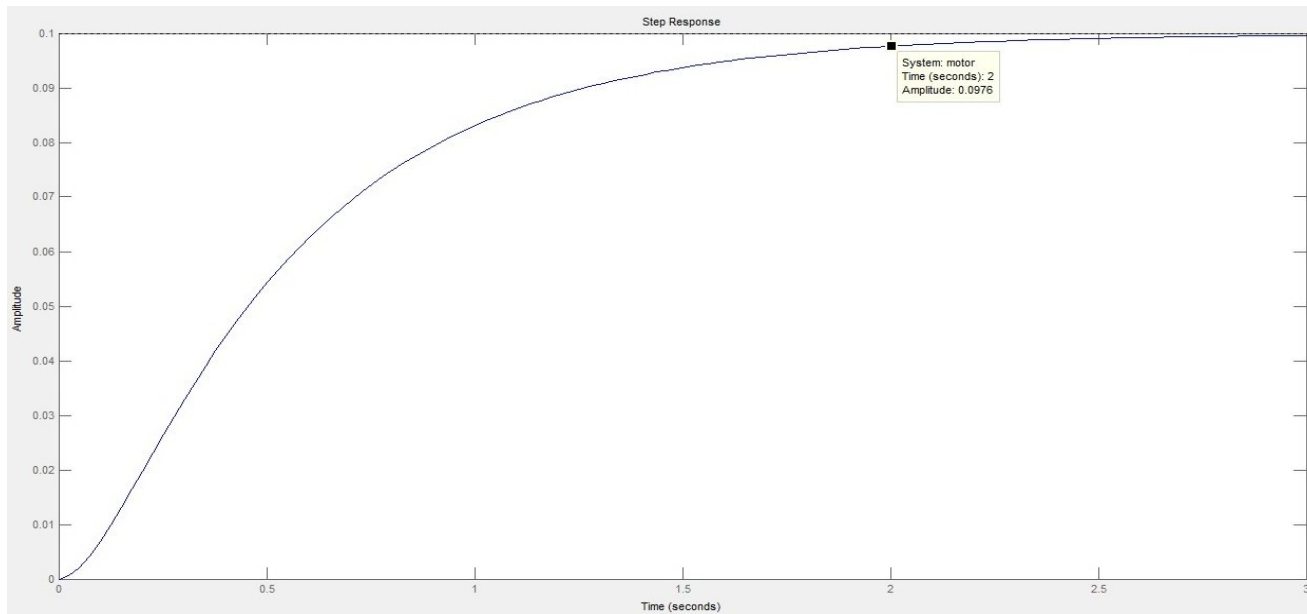
```
J = 0.01; %Αδράνεια του μοτέρ (Kg*m2)
b = 0.1; %Σταθερά τριβής (N*m*s)
K = 0.01; %Ke: Σταθερά ηλεκτρεγερτικής δύναμης (V*sec/rad)
          %Kt: Σταθερά ροπής μοτέρ (N*m/A)
R = 1; %Αντίσταση (Ω)
L = 0.5; %Επαγωγή (H)
s = tf('s'); %Ορισμός του s ως μεταβλητή
motor1 = K / ((J*s+b) * (L*s+R) + K^2) %Συνάρτηση μεταφοράς
```

Κώδικας υπολογισμού της συνάρτησης μεταφοράς του μοτέρ μας στο MATLAB

Η απόκριση που επιθυμούμε να έχει το μοντέλο μας πρέπει να ικανοποιεί τις εξής τρεις συνθήκες:

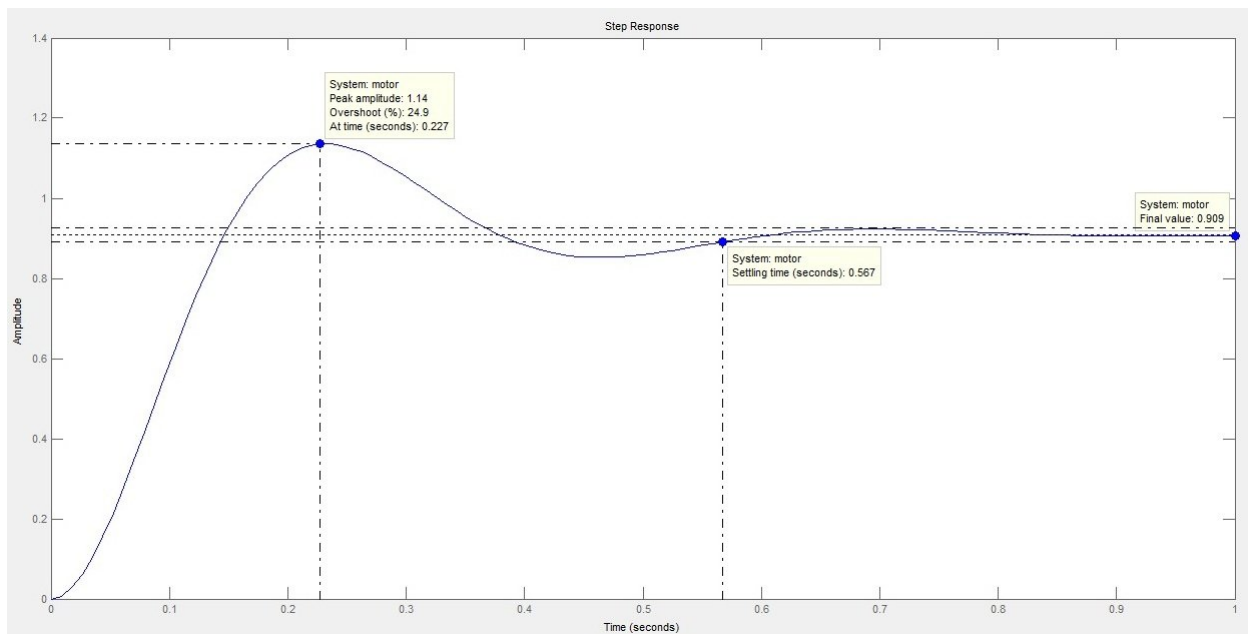
1. Χρόνος αποκατάστασης μικρότερος των 2 δευτερολέπτων
2. Υπερύψωση μικρότερη από 5%
3. Μόνιμο σφάλμα μικρότερο από 1%

Όπως και στο κεφάλαιο 3, έτσι και τώρα θα εξετάσουμε κατά πόσο το μοντέλο μας θα χρειαστεί έλεγχο. Η απόκριση που μας δίνει το μοτέρ σε βηματική είσοδο, είναι η ακόλουθη:



Εικόνα 66: Απόκριση του μοτέρ μας χωρίς έλεγχο

Παρατηρώντας την απόκρισή μας, βλέπουμε πως δεν πληρείται η μια εκ των τριών μας συνθηκών (ο χρόνος αποκατάστασης είναι μεγαλύτερος των 2 δευτερολέπτων). Επίσης, παρατηρούμε πως το σύστημά μας δείχνει να σταθεροποιείται σε μια τιμή, η οποία όμως δεν είναι η τιμή της εισόδου. Για τους δυο αυτούς λόγους, το σύστημά μας θα χρειαστεί, την εισαγωγή ενός ελεγκτή PID. Μην έχοντας κάποιο οριακό κέρδος για την απόκριση του συστήματός μας, αρχικά θα εφαρμόσουμε έναν αναλογικό ελεγκτή με κέρδος K_p και με απότερω σκοπό να ανεβάσουμε την απόκρισή μας στην τιμή της εισόδου μας και να την παρατηρήσουμε ξανά. Μετά από δοκιμές τιμών, θέτουμε το $K_p = 100$ και το σύστημά μας αποκρίνεται ως εξής:



Εικόνα 67: Απόκριση του μοτέρ μας με αναλογικό ελεγκτή $C(s) = 100$

Η απόκριση αυτή είναι κοντά στην τιμή της εισόδου μας και επίσης μας δίνει και στοιχεία τα οποία θα μας βοηθήσουν να αποφασίσουμε το πως θα κινηθούμε. Ο χρόνος αποκατάστασης έχει πέσει στην τιμή 0.567 δευτερόλεπτα, το οποίο μας ικανοποιεί την πρώτη μας συνθήκη. Όμως πλέον, δεν ικανοποιούνται οι άλλες δυο. Έχουμε μια υπερύψωση μεγαλύτερη του 5% (24.9%) και έχουμε και μόνιμο σφάλμα. Όπως και στο κεφάλαιο 3, θα χρειαστούμε και τα άλλα δυο κέρδη (ολοκληρωτικό, διαφορικό) του ελεγκτή μας προκειμένου να βελτιστοποιήσουμε την απόκριση του μοτέρ μας.

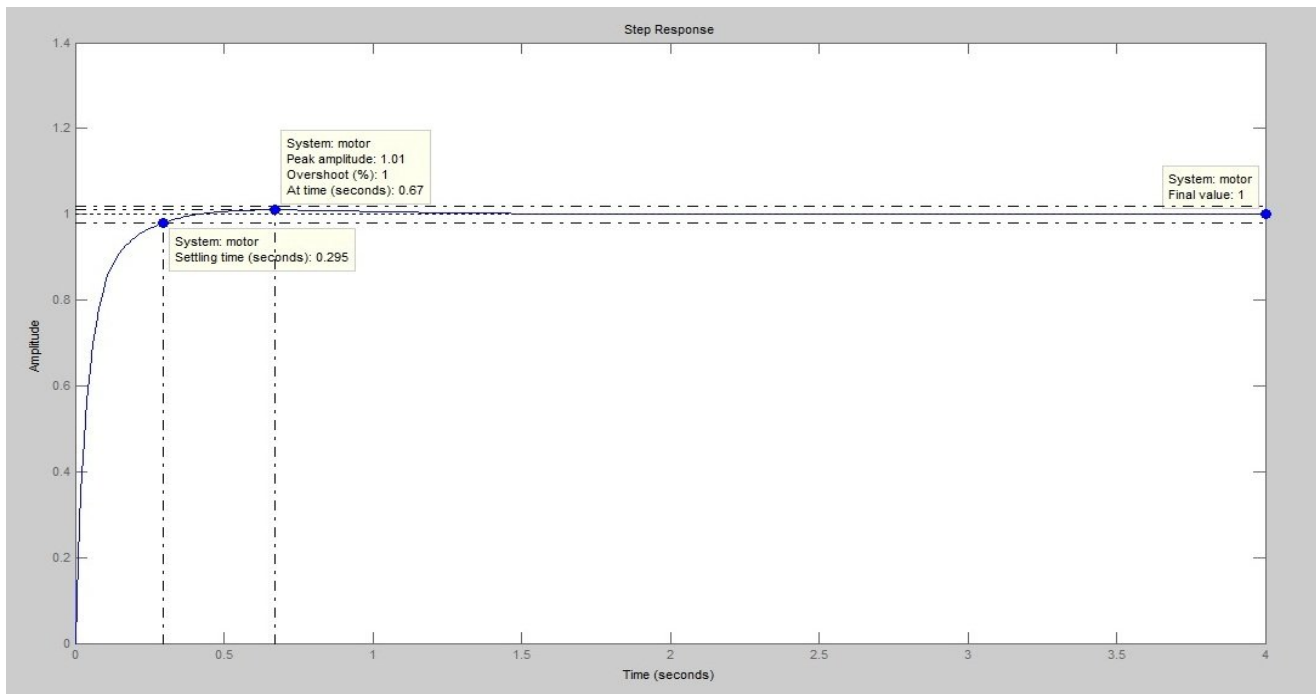
Δοκιμάζοντας αρκετές τιμές καταλήξαμε στις παρακάτω, οι οποίες θα διαμορφώσουν τον ελεγκτή μας:

$$K_p = 100, K_i = 200, K_D = 11.2$$

με συνάρτηση μεταφοράς:

$$C(s) = \frac{11.2s^2 + 100s + 200}{s}$$

Η απόκριση που παίρνουμε είναι η εξής:



Εικόνα 68: Απόκριση του μοτέρ μας με εφαρμογή ελεγκτή PID

Η απόκριση με τον ελεγκτή αυτό, είναι πάρα πολύ καλή, και ικανοποιεί και τις τρεις συνθήκες μας. Ο χρόνος αποκατάστασης έχει μειωθεί σχεδόν στο μισό από την προηγούμενη μας απόκριση (0.295 δευτερόλεπτα), η υπερύψωση είναι σε αρκετά χαμηλότερο επίπεδο από το ζητούμενο (1%), και το μόνιμο σφάλμα έχει εξαλειφθεί.

Ο συμπληρωματικός κώδικας που δημιουργήθηκε για την εισαγωγή του ελεγκτή μας στο σύστημά είναι ο παρακάτω:

```
Kp = 100;      %Τιμή για το αναλογικό κέρδος
Ki = 200;      %Τιμή για το ολοκληρωτικό κέρδος
Kd = 11.2;     %Τιμή για το διαφορικό κέρδος
C = pid(Kp,Ki,Kd); %Δημιουργία της συνάρτησης μεταφοράς του
                  %PID ελεγκτή μας
motor = feedback(C*motor1,1) %Δημιουργία του κλειστού βρόγχου
                  %και υπολογισμός συνολικής συνάρτησης μεταφοράς
step(motor,[0:0.01:4]) %Απόκριση του συστήματός μας
```

Συμπληρωματικός κώδικας για την εισαγωγή του ελεγκτή PID στο σύστημά μας

Το μοντέλο που παρουσιάστηκε εδώ σε σύγκριση με αυτό του κεφαλαίου 3, μοιάζει σαφώς καλύτερο. Με μια γρήγορη σύγκριση μπορούμε να δούμε πως η υπερύψωση του μοντέλου είναι σε χαμηλότερα επίπεδα από αυτή του κεφαλαίου 3 (1% αντί για 4.67%). Επίσης, έχει πολύ γρηγορότερο χρόνο αποκατάστασης, το οποίο εμάς μας ενδιαφέρει σε πολύ μεγάλο βαθμό. Όμως αυτό που μας οδήγησε στο να “απορρίψουμε” το μοντέλο αυτό είναι η είσοδος του. Το μοντέλο που παρουσιάστηκε στην ενότητα αυτή, σταθεροποιεί την τιμή που του έχουμε δώσει στην είσοδό του. Οπότε για να δούμε αποτέλεσμα θα πρέπει να δίνουμε ως είσοδο την γωνιακή ταχύτητα, και κάτι τέτοιο δεν έχει υπόσταση στην πραγματικότητα. Έτσι παρόλο που το μοντέλο του κεφαλαίου 3 μοιάζει οριακό ως προς τις συνθήκες μας, πλην του μόνιμου σφάλματος, προτιμήθηκε διότι στην είσοδό του δέχεται την τάση (π.χ από μια μπαταρία) και την αποδίδει σε γωνιακή ταχύτητα, το οποίο εφαρμόζεται και στην πραγματικότητα.

4.1.2 Διαφορές πρώτου και δεύτερου μοντέλου

Στην εργασία μας αναπτύχθηκαν δυο μοντέλα. Το πρώτο μοντέλο μας δημιουργήθηκε για να μπορέσουμε να παρατηρήσουμε σε ένα πρώτο στάδιο, το πως λειτουργεί και αποκρίνεται ένα τέτοιο σύστημα. Επειδή όμως το μοντέλο αυτό αναπτύχθηκε παράλληλα με την μελέτη της θεωρίας ενός τέτοιου συστήματος έγινε χρήση έτοιμων block του Simulink τα οποία συνδέσαμε μεταξύ τους προκειμένου να αποδόσουμε όσο καλύτερα γινόταν ένα τέτοιο σύστημα. Αντιθέτως, το δεύτερο μοντέλο μας αναπτύχθηκε σύμφωνα με τις διαφορικές εξισώσεις ενός τέτοιου συστήματος (μοντελοποίηση των μοτέρ και του quadcopter). Ειδικότερα στο μοντέλο αυτό δημιουργήθηκαν διάφοροι τρόποι εισαγωγής συντεταγμένων, δυο μέθοδοι εισαγωγής από χειριστήριο και δυο μέθοδοι εισαγωγής με αυτόματη διάταξη για τον έλεγχο της οποίας έχουμε εφαρμόσει έλεγχο με έναν ελεγκτή ασαφούς λογικής. Είναι σαφές λοιπόν πως τα δυο μοντέλα μας έχουν διαφορές στον τρόπο με τα οποία είναι υλοποιημένα, και στην ενότητα αυτή θα τις αναφέρουμε συνοπτικά.

- Το πρώτο μας μοντέλο εξαιτίας των έτοιμων block που χρησιμοποιήθηκαν, μας κατέστησε δύσκολη την όποια επέμβαση σε αυτό προκειμένου να επεξεργαστούμε τα δεδομένα μας όπως εμείς θα επιθυμούσαμε. Έχοντας λοιπόν στο μυαλό μας την δημιουργία ευελιξίας, το δεύτερο μας μοντέλο αναπτύχθηκε με τέτοιο τρόπο ώστε να δώσει ποικίλους τρόπους εισαγωγής των συντεταγμένων μας στο μοντέλο μας.
- Το πρώτο μας μοντέλο αντιμετώπιζε το σύστημά μας ως ένα σύνολο στοιχείων (κατασκευή-μοτέρ), το οποίο φυσικά δεν ήταν κάτι το επιθυμητό. Για τον λόγο αυτό στο δεύτερο μοντέλο μας αναπτύξαμε τα μοτέρ μας ξεχωριστά από το σύστημά μας για να ελέγξουμε σε πρώτη φάση την απόκρισή τους. Στην συνέχεια αναπτύξαμε το ίδιο το σύστημά μας, και στο

τελικό στάδιο κάναμε δοκιμές για την απόκρισή του μέσω των μεθόδων εισαγωγής συντεταγμένων που είχαμε αναπτύξει.

- Μια σημαντική διαφορά μεταξύ του πρώτου και του δεύτερου μοντέλου μας είναι πως στο δεύτερο μοντέλο κάνουμε χρήση ενός φίλτρου kalman για να μειώσουμε στο ελάχιστο τον θόρυβο που παράγουν τα αισθητήριά μας, και το αποτέλεσμα που παίρνουμε είναι αρκετά ικανοποιητικό.

Ουσιαστικά το πρώτο μοντέλο που αναπτύχθηκε δεν μας εξυπηρέτησε στην απορροή συμπερασμάτων, αλλά μας βοήθησε αρκετά στο να παρατηρήσουμε αρχικά την συμπεριφορά ενός τέτοιου συστήματος ώστε να μπορέσουμε, σαν επόμενο βήμα, να πραγματοποιήσουμε την δική μας έρευνα, αναπτύσσοντας από την αρχή, ένα δικό μας τέτοιο σύστημα.

4.2 Συμπεράσματα

Το quadcopter ως θέμα είναι πολύ απαιτητικό, και χρειάζεται αρκετή μελέτη στην αρχή ώστε να κατανοηθεί απόλυτα το αντικείμενο προκειμένου να μεταβούμε σε οποιαδήποτε δική μας εφαρμογή. Είχε αρκετά καινούρια στοιχεία ως θέμα, τα οποία είναι υψίστης σημαντικότητας ως εκπαιδευτικές εμπειρίες για την μετέπειτα πορεία μας. Συμπεράσματα σχετικά με την διαδικασία υλοποίησης της εργασίας ακολουθούν ευθύς αμέσως:

- Επιτεύχθηκε μια αρκετά ρεαλιστική ανάπτυξη ενός μοντέλου quadcopter που είναι το βασικότερο και πιο δύσκολο κομμάτι της μοντελοποίησής μας.
- Κατορθώθηκε έλεγχος των τεσσάρων μοτέρ σε τέτοιο βαθμό, ώστε να φτάνει αρκετά γρήγορα στην επιθυμητή μας ταχύτητα και να διατηρεί την τιμή αυτή για να μην υπάρχουν αστάθειες κατά την πτήση. Ο PID έλεγχος που αναπτύσσεται στην εργασία αυτή, γίνεται με παραμέτρους που θα μπορούσαν (ίσως) να εφαρμοστούν και σε ένα πραγματικό μοντέλο.
- “Φιλτραρίστηκαν” οι μετρήσεις μας από τα αισθητήριά μας σε ικανοποιητικό βαθμό, με την χρήση του φίλτρου Kalman. Ο αλγόριθμος αυτός είναι αρκετά πολύπλοκος και χρειάζεται χρόνο και αρκετά πειράματα μέχρι να εξοικιωθούμε με αυτόν. Αποτέλεσε ένα από τα απαιτητικότερα στοιχεία της εργασίας αυτής, και αποδείχθηκε πολύ αποτελεσματικό.
- Η μοντελοποίηση ενός quadcopter είναι μια πρώτη προσέγγιση ενός πραγματικού τέτοιου συστήματος, στο οποίο μπορούμε να επέμβαιουμε με περισσότερη ασφάλεια απ’ότι στο πραγματικό (γιατί κάποιο ενδεχόμενο

λάθος, θα σήμαινε αυτομάτως κάποια ενδεχόμενη πτώση). Όμως παρόλο που μπορεί το μοντέλο μας να δείχνει αρκετά ικανοποιητικό, υπάρχει πιθανότητα το ίδιο ακριβώς μοντέλο, να έχει προβλήματα σε πραγματικό χρόνο.

- Καθώς “τρέχουμε” το μοντέλο μας στο MATLAB-Simulink παρατηρήθηκε πως όταν υπάρχει ο ελεγκτής ασαφούς λογικής στο μοντέλο μας, αυτό γίνεται πολύ αργό, και ο λόγος είναι πως γίνεται μεταφορά δεδομένων από και προς το κεντρικό περιβάλλον του MATLAB με αποτέλεσμα όλη η διαδικασία αυτή να “βαρύνει” το μοντέλο μας.

4.3 Μελλοντική εργασία

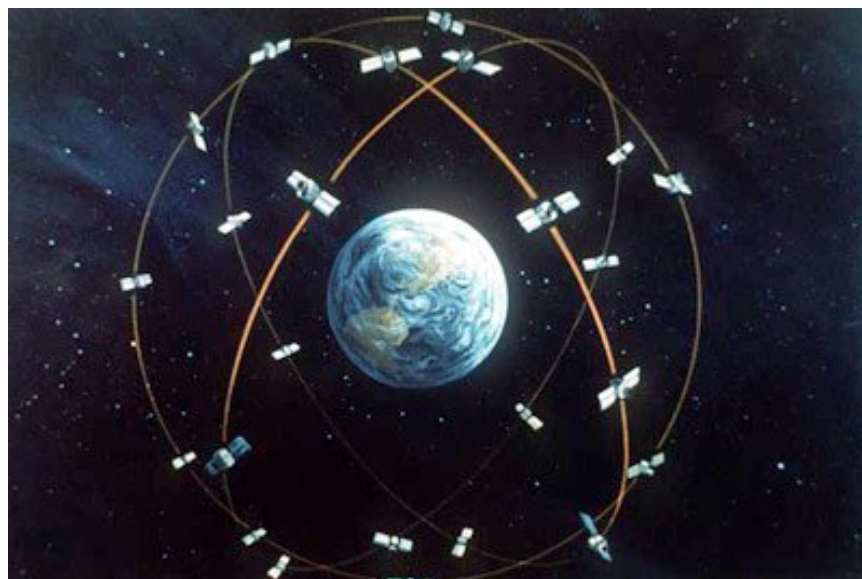
Στην εργασία αυτή χτίσαμε όλες τις απαραίτητες θεωρητικές βάσεις με την βοήθεια του MATLAB – Simulink, προκειμένου να γίνει μια προσομοίωση και παρατήρηση ενός δικού μας μοντέλου quadcopter.

Έχοντας λοιπόν συγκεντρώσει όλα αυτά τα απαραίτητα στοιχεία για τον σχεδιασμό και ακόμα περισσότερο την πιθανή συμπεριφορά ενός τέτοιου συστήματος σε πραγματικό χρόνο, ως επόμενο στάδιο και μελλοντικό στόχο, έχω την κατασκευή ενός τέτοιου πραγματικού μοντέλου και την υλοποίηση μελέτης εξ'ολοκλήρου από την αρχή για να διαπιστωθεί, αν η θεωρητική μας προσέγγιση είναι αρκετά ικανοποιητική ώστε να εφαρμοστεί σε ένα πραγματικό μοντέλο. Στον σχεδιασμό του ελεγκτή μας όμως, θα πρέπει να λάβουμε υπόψιν μας και τις επιδράσεις διάφορων διαταραχών, με σημαντικότερη αυτών την επίδραση του αέρα, με σκοπό την βέλτιστη ρύθμιση του ελεγκτή μας για να μπορεί να σταθεροποιεί τον μηχανισμό μας σε οποιαδήποτε πιθανή ρηπή ανέμου.

Πολύ σημαντικό, επίσης, είναι να γίνει και η βέλτιστη κατανομή όλου του εξοπλισμού που θα προσαρμοστεί στο quadcopter μας (μπαταρίες, αισθητήρια, μικροεπεξεργαστής, κάμερα κ.ά), ώστε να διατηρήσουμε το κέντρο μάζας της κατασκευής μας ακριβώς στο κέντρο της, το οποίο δεν είναι εύκολα εφικτό. Τα 4 μοτέρ δεν θα μας επηρεάσουν στην σωστή κατανομή, μόνο στην περίπτωση που αυτά τοποθετηθούν στην ίδια ακριβώς απόσταση από το κέντρο (άλλωστε αυτός είναι και ο σκοπός της κατασκευής).

Αφού λοιπόν καταφέρουμε να κάνουμε όσο το δυνατόν καλύτερη την διανομή του εξοπλισμού μας, σειρά θα έχει ο χειρισμός. Σε πρώτο στάδιο, θα γίνει χειρισμός του quadcopter με την βοήθεια χειριστηρίου, το οποίο είναι αρκετά συνηθισμένο πλέον ως εργασία. Αυτό που θα κάνει ξεχωριστή όμως την εργασία μας, είναι η

προσπάθεια επίτευξης χειρισμού, χωρίς να γίνεται χρήση χειριστηρίου, αλλά με χρήση μιας εντολής, η οποία θα περιέχει τις συντεταγμένες μας στον χώρο. Στην εργασία μας αυτή θα μας ήταν χρήσιμη η λειτουργία ενός GPS (Global Positioning System). Το GPS, είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα “πλέγμα” εικοσιτεσσάρων δορυφόρων της Γης, στους οποίους υπάρχουν ειδικές συσκευές, που ονομάζονται “δέκτες GPS”. Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνσή του. Επίσης, υπάρχει και η δυνατότητα χαρτογράφησης, αλλά θα πρέπει να συνδυαστεί με ειδικό λογισμικό. Με την βοήθεια ενός τέτοιου συστήματος λοιπόν, θα μπορούσαμε να εξασφαλίσουμε πλήρη αυτονομία πτήσης, καθώς επίσης και πτήση σε άγνωστα ή ανεξερεύνητα εδάφη (δυνατότητα χαρτογράφησης).



Εικόνα 69: Σχηματισμός του “πλέγματος” των δορυφόρων γύρω από την Γη

Τέλος, θα προσαρμοστεί κάμερα στο μοντέλο μας, και με την βοήθεια σύνδεσης μέσω ασύρματου internet θα συλλέγουμε όλα τα δεδομένα που θα “βλέπει” το quadcopter, προκειμένου να τα επεξεργαστούμε.



Εικόνα 70: Μοντέλο quadcopter με ενσωματωμένη κάμερα

Πρόκειται για μια αρκετά απαιτητική μελλοντική εξέλιξη της εργασίας αυτής, η οποία θα χρειαστεί πάρα πολύ χρόνο, και αρκετή μελέτη ώστε να φτάσουμε σε ένα τελικό πραγματικό μοντέλο, το οποίο περιγράφεται στην τελευταία παράγραφο.

Βιβλιογραφία:

- [1] Khantsis S. and Bourmistrova A. "Stochastic Design of a Non-Linear Controller for Autonomous UAV Recovery."
- [2] A.R. Babaei, M. M., M.H. Moradi (2009). "Classical and fuzzy-genetic autopilot design for unmanned aerial vehicles."
- [3] Abdelkader Abdessameud, A. T. (2010). "Global trajectory tracking control of VTOL-UAVs without linear velocity measurements."
- [4] Agudelo, C. M. V. a. A. (2006). "Modeling, Simulation and Rapid Prototyping of an Unmanned Mini-Helicopter."
- [5] Alexandros SOUMELIDIS, B. L., Zoltán PROHÁSZKA and Gergely REGULA, Péter BAUER (2008). "EMBEDDED COMPUTER BASED NONLINEAR VEHICLE CONTROL: A QUADROTOR HELICOPTER EXPERIMENT."
- [6] Alexandros Soumelidis, P. e. G. a. a., Gergely Regula, Bela Lantos "Control of an experimental mini quad-rotor UAV."
- [7] Alexandros Soumelidis, P. e. G. a. a., Peter Bauer, Bela Lantos, and Zoltan Prohaszka "Design of an embedded microcomputer based mini quadrotor UAV."
- [8] Ali Karimodini, H. L., Ben M. Chen, Tong Heng Lee (2010). "Hybrid formation control of the Unmanned Aerial Vehicles."
- [9] Allen, M. J. (2007). "Guidance and Control of an Autonomous Soaring UAV."
- [10] Anibal Ollero , L. s. M. (2004). "Control and perception techniques for aerial robotics."
- [11] Atheer L. Salih, M. M., Haider A. F. Mohamed and Khalaf Sallom Gaeid (2010). "Flight PID controller design for a UAV quadrotor."

- [12] Boris Vidolov, J. D. M., and Stéphane Bonnet "AURYON – A Mechatronic UAV Project Focus on Control Experimentations."
- [13] C. Nicol, C. J. B. M., A. Ramirez-Serrano (2011). "Robust adaptive control of a quadrotor helicopter."
- [14] CARLOS M. VÉLEZ S., A. A. (2006). "Rapid software prototyping for real-time simulation and control of a mini-helicopter robot."
- [15] Christel-Loic Tisse, H. D.-W., R. Andrew Hicks (2006). "An optical navigation sensor for micro aerial vehicles."
- [16] Christian Bermes, S. B., Dario Schafroth, Roland Siegwart (2011). "Design of the autonomous micro helicopter muFly."
- [17] Downey, A. B. (2007-2010). Physical Modeling in MATLAB
- [18] G. Angeletti, J. R. P. V., L. Iocchi, D. Nardi (2008). "Autonomous Indoor Hovering with a Quadrotor."
- [19] Gabriel M. Hoffmann, H. H., Steven L. Waslander, Claire J. Tomlin "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment."
- [20] Gabriel M. Hoffmann, H., Steven L. Waslander, Claire J. Tomlin (2011). "Precision flight control for a multi-vehicle quadrotor helicopter testbed."
- [21] Ganglin, W. (2009). "Key Parameters and Conceptual Configuration of Unmanned Combat Aerial Vehicle Concept."
- [22] Georgios P. Kladis, J. E., Kevin Knowles, Jimmy Lauber, Thierry-Marie Guerra (2010). "Energy conservation based fuzzy tracking for unmanned aerial vehicle missions under a priori known wind information."
- [23] Grzegorz Kopecki, J. P., Tomasz Rogalski, Pawel Rzucidło, Andrzej Tomczyk (2010). "PROPOSAL FOR NAVIGATION AND CONTROL SYSTEM FOR SMALL UAV."

- [24] Guowei Cai, B. M. C., Tong H. Lee, Miaobo Dong (2009). "Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters."
- [25] Guowei CAI, B. M. C., Tong H. LEE (2009). "An overview on development of miniature unmanned rotorcraft systems."
- [26] Guowei Cai, B. M. C., Xiangxu Dong, Tong H. Lee (2011). "Design and implementation of a robust and nonlinear flight control system for an unmanned helicopter."
- [27] Guowei Cai, L. F., Ben M. Chen , Tong H. Lee (2008). "Systematic design methodology and construction of UAV helicopters."
- [28] Hansol Park and Doo-Hyun Kim, J.-G. K., Chun-Hyon Chang (2008). "Experimental Evaluation of Unmanned Aerial Vehicle System Software Based on the TMO Model."
- [29] Hong, S. K. (2003). "Fuzzy logic based closed-loop strapdown attitude system for unmanned aerial vehicle (UAV)."
- [30] Hyondong Oh, D.-Y. W., Sung-Sik Huh, David Hyunchul Shim and Min-Jea Tahk "Experimental Framework for Controller Design of a Rotorcraft Unmanned Aerial Vehicle Using Multi-Camera System."
- [31] Iván F. Mondragón, P. C., Carol Martinez, Miguel Olivares (2011). "Omnidirectional vision applied to Unmanned Aerial Vehicles (UAVs) attitude and heading estimation."
- [32] J.F. Guerrero-Castellanos, N. M., A.Hably, S.Lesecq, J.Delamare (2011). "Bounded attitude control of rigid bodies: Real-time experimentation to a quadrotor mini-helicopter."
- [33] Jasmin Velagic, Z. V., Edin Omerdic (2001). "Adaptive fuzzy ship autopilot for track-keeping."
- [34] Javier Bilbao, A. O., Eugenio Bravo, Olatz García, Concepción Varela and Miguel Rodríguez (2008). "How design an unmanned aerial vehicle with great efficiency in the use of existing resources."

- [35] Jonathan Courbon, Y. M., Nicolas Gue´nard, Philippe Martinet (2010). "Vision-based navigation of unmanned aerial vehicles."
- [36] Jongho Shin , H. J., SewookPark , YoudanKim (2009). "Model predictive flight control using adaptive support vector regression."
- [37] Jongki Moon, J. V. R. P. (2010). "Minimum-time approach to obstacle avoidance constrained by envelope protection for autonomous UAVs."
- [38] Joseph Egbert, R. W. B. (2010). "Low-altitude road following using strap-down cameras on miniature air vehicles."
- [39] K. Dalamagkidis, K. P. V., L.A.Piegl (2008). "On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the National Airspace System."
- [40] Kostas Alexis, G. N., AnthonyTzes (2011). "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances."
- [41] L. Doitsidis, K. P. V., N. C. Tsourveloudis, M. Kontitsis (2004). "A Framework for Fuzzy Logic Based UAV Navigation and Control. "
- [42] Mario Luca Fravolini, A. F., Giampiero Campa, Marcello Rosario Napolitano, Brad Seanor (2004). "Modeling and control issues for autonomous aerial refueling for UAVs using a probe–drogue refueling system."
- [43] N.R. Gans , W. E. D., R. Lind , A. Kurdila (2009). "A hardware in the loop simulation platform for vision-based control of unmanned air vehicles."
- [44] Oliveira, L. R. R. a. N. M. F. (2010). "UAV Autopilot Controllers Test Platform Using Matlab/Simulink and X-Plane."
- [45] Oliver Purwin, R. D. A. (2010). "Performing and extending aggressive maneuvers using iterative learning control."
- [46] Paul Pounds, R. M., Peter Corke "Modelling and Control of a Quad-Rotor Robot."
- [47] Pedro Castillo, R. L. a. A. E. D. (2004). "Modelling and Control of Mini-Flying Machines. "

- [48] Pehlivanoglu, Y. V. (2011). "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV."
- [49] Raza Samar, A. R. (2010). "Autonomous terrain-following for unmanned air vehicles."
- [50] S.D. Lee, C. H. Y., K.Y.Hsiu, Y.F.Hsieh, C.Y.Tzeng, Y.Z.Kehr (2009). "Design and experiment of a small boat track-keeping autopilot."
- [51] Surmann, V. T. a. H. "Hardware architecture of a four-rotor UAV for USAR/WSAR scenarios."
- [52] Tarek Hamel, R. M., Rogelio Lozano, James Ostrowski (2002). "DYNAMIC MODELLING AND CONFIGURATION STABILIZATION FOR AN X4-FLYER."
- [53] Tsung-Ying Sun, C.-L. H., Shang-Jeng Tsai, Yu-Hsiang Yu, Chan-Cheng Liu (2011). "Intelligent flight task algorithm for unmanned aerial vehicle."
- [54] X.Q. Chen, Q. O., D. R. Wong, Y. J. Li, M. Sinclair, A. Marburg "Flight Dynamics Modelling and Experimental Validation for Unmanned Aerial Vehicles."
- [55] Yew Chai Paw, G. J. B. (2010). "Development and application of an integrated framework for small UAV flight control development."
- [56] Zhaotong Lian , A. D. (2004). "Performance prediction of an unmanned airborne vehicle multi-agent system."
- [57] HaiYang Chao, YongCan Cao, and YangQuan Chen (2010). "Autopilots for Small Unmanned Aerial Vehicles: A Survey."
- [58] P.M. Mäkilä (2004). "Kalman Filtering and Linear Quadratic Gaussian Control."
- [59] Greg Welch, Gary Bishop (2001). "An Introduction to the Kalman Filter."
- [60] Shay Reuveny, Maayan Zadik. "3D Motion tracking with Gyroscope and Accelerometer."
- [61] Teppo Luukkonen (2011). "Modelling and control of quadcopter."

- [62] Erişmiş, Mehmet Akif (2004). "MEMS ACCELEROMETERS AND GYROSCOPES FOR INERTIAL MEASUREMENT UNITS."
- [63] Rachel Kleinbauer (2004). "Kalman Filtering Implementation with Matlab."
- [64] Tomáš Jiinec (2011). "Stabilization and Control of Unmanned Quadcopter."
- [65] Lum Yue Hao Joseph (2011). "UNMANNED AERIAL VEHICLE (UAV) AUTOPILOT SYSTEM."
- [66] Holly Moore (2011). "MATLAB® for Engineers."
- [67] C. Balas (2007). "Modelling and linear control of a quadrotoer"

Συνοπτικά στοιχεία βιβλιογραφίας:

- Μοντελοποίηση:

1. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΛΕΓΧΟΣ ΕΝΟΣ QUADROTOR

Μέχρι σήμερα, τα περισσότερα quad-rotor εναέρια ρομπότ έχουν βασιστεί στα πετούμενα παιχνίδια. Αν και τέτοια συστήματα μπορούν να χρησιμοποιηθούν ως πρωτότυπα, δεν είναι επαρκώς ισχυρές για να χρησιμεύσει ως πειραματική ρομποτική πλατφόρμα. Έχουν αναπτύξει το X-4 Flyer, ένα quad-rotor ρομπότ χρησιμοποιώντας ένα ειδικά προσαρμοσμένο αμάξωμα και με τα αεροηλεκτρικά στοιχεία με κινητήρες που είναι στον αέρα και μπαταρίες, για να είναι μια υψηλά αξιόπιστη πειραματική πλατφόρμα. Τα οχήματα χρησιμοποιούν συντονισμένες δυναμικές εγκαταστάσεις με ένα ενσωματωμένο ελεγκτή συμπεριφοράς, για την σταθεροποίηση της πτήσης.

Στην πορεία του paper μας αναφέρει τις ήδη υπάρχοντες πλατφόρμες για quadrotor, καθώς επίσης και οι στόχοι της τρέχουσας ανάπτυξης. Μετά μας παρουσιάζει τα hardware στοιχεία και την δομή του X-4. Συγκεκριμένα αναφέρει, για το αμάξωμα, για το σύστημα οδήγησης, για τον έλεγχο, για τις εντολές και την τηλεμετρίας. Στην συνέχεια, γίνεται ανάλυση της δυναμικής του quadrotor μέσα από μαθηματικούς τύπους και σχεδιαγράμματα. Μετά γίνεται ανάλυση για την παραμετροποίηση του μοντέλου και την σταθερότητα του, δίνοντας μαθηματικούς τύπους και εξισώσεις, και γραφικές παραστάσεις. Ακολούθως, γίνεται η επιλογή του κατάλληλου ελεγκτή και προσομοιώνεται το μοντέλο, δίνοντας και πάλι μαθηματικές ιδιότητες καθώς επίσης και το block διάγραμμα του όλου συστήματος. Τέλος, γίνεται η εφαρμογή των όσων έχουν αναφερθεί, αλλά και η καταγραφή της επίδοσης του quadrotor. Η ομάδα αυτή κατάφερε, να αναπτύξει μια μεγαλύτερη πλατφόρμα για το quadrotor σε σχέση με αυτή που τυπικά χρησιμοποιούσαν στις τρέχουσες ρομποτικές έρευνες. Η ανάλυση των ιπτάμενων δυναμικών συμπεριφορών μας επιτρέπουν να προσαρμόσουμε τον μηχανικό σχεδιασμό για τον καλύτερο έλεγχο της ευαισθησίας και την απόρριψη των διαταραχών. Σχεδίασαν έναν ελεγκτή για να σταθεροποιήσει την κυρίαρχη αποσυνδεδεμένη κλίση και τους τρόπους περιστροφής, και χρησιμοποίησε ένα μοντέλο διαταραχής των εισόδων, για την εκτίμηση της απόδοσης των εγκαταστάσεων. Βρήκαν πως ο αντισταθμιστής, ρυθμίζει επιτυχώς την συμπεριφορά και σε χαμηλές ταχύτητες του ρότορα.

2. ΔΥΝΑΜΙΚΟ ΜΟΝΤΕΛΟ ΚΑΙ ΔΙΑΜΟΡΦΩΣΗ ΣΤΑΘΕΡΟΠΟΙΗΣΗΣ ΓΙΑ ΕΝΑ X-4 FLYER

Σε αυτό το paper, προτείνεται ένα μοντέλο για τις δυναμικές των τεσσάρων ροτόρων κάθετη απογείωση και προσγείωση του οχήματος, γνωστό σαν X-4 flyer. Το μοντέλο περιλαμβάνει την άτρακτο και την δυναμική του κινητήρα καθώς και τις αεροδυναμικές και τις γυροσκοπικές συνέπειες, λόγω των ροτόρων. Η διορθωμένη κλίση, οι άκαμπτοι ρότορες, και η κίνηση των τεσσάρων κινητήρων οδηγούν σε σημαντικές διαφορές στο προτεινόμενο μοντέλο σε σχέση με τα άλλα VTOL συστήματα που έχουν μελετηθεί πρόσφατα στην λογοτεχνία. Ο σχεδιασμός του ελέγχου που προτείνεται είναι βασισμένος στο να διαχωρίζει τη δυναμική του άκαμπτου σώματος (αεροσκάφος) από την δυναμική του κινητήρα, αναπτύσσοντας διαχωριζόμενο έλεγχο συνάρτησεων Lyapunov για τον συνδυασμό συστημάτων και τότε οριοθετείται το σφάλμα της διαταραχής που οφείλεται στην αλληλεπίδραση, για να αποκτήσει ισχυρή πρακτικά σταθερότητα του όλου συστήματος.

Στην πορεία της παρουσίασης γίνεται ανάλυση του μοντέλου δίνοντας σχήμα με του μοντέλου μαζί με τις δυνάμεις που ασκούνται στους κινητήρες του κάθε έλικα. Στην συνέχεια παρουσιάζεται η μεθοδολογία σχεδιασμού του ελεγκτή κάνοντας μαθηματική ανάλυση προκειμένου να βρούμε τις παραμέτρους του ζητούμενού μας ελεγκτή. Μια πιο πλήρης συζήτηση του δυναμικού του προτεινόμενου σχεδιασμού ελέγχου είναι πέρα από το επίπεδο εφαρμογής σε αυτό το paper. Απλή αλλά κουραστικοί υπολογισμοί δείχνουν ότι επιλέγοντας K_u μεγαλύτερο από το δοσμένο όριο στον θεωρητικό ισχυρισμό, εξασφαλίζει πως τα δυο τελευταία στοιχεία της κύριας διαγωνίου του πίνακα A, είναι ξεκάθαρα θετικά. Η απόδειξη ακολουθεί εφαρμόζοντας άμεσα την μέθοδο του Lyapunov.

3. ΣΧΕΔΙΑΣΜΟΣ ΜΙΚΡΟΥ QUAD-ROTOR UAV ΜΕ ΒΑΣΗ ΕΝΑΝ ΕΝΣΩΜΑΤΩΜΕΝΟ ΜΙΚΡΟΕΛΕΓΚΤΗ

Σε αυτό το paper περιγράφεται ο σχεδιασμός και η υλοποίηση ενός μικρού quad-rotor UAV που ξεκίνησε στο System and Control Laboratory στο Computer and Automation Research institute της Ουγγρικής ακαδημίας Επιστήμης σε συνεργασία με τα τμήματα του Τεχνολογικού και Οικονομικού Πανεπιστημίου της Βουδαπέστης. Το μικρό quad-rotor UAV προορίζεται να χρησιμοποιηθεί σε διάφορους τομείς όπως στην εναέρια επιτήρηση με κάμερα, στον έλεγχο της κυκλοφορίας, σε περιβαλλοντικές μετρήσεις, κλπ. Το paper εστιάζει στην εκτέλεση του ενσωματωμένου μικροελεγκτή ενός μικρού UAV, περιγράφει τα στοιχεία της εκτέλεσης, τα εργαλεία που χρησιμοποιήθηκαν για την κατασκευή του

μαθηματικού μοντέλου, καθώς επίσης επιτυγχάνει μια σύντομη περιγραφή του σχεδιασμού του ελέγχου.

Στο paper αυτό παρουσιάζεται το πως μπορούμε να κατασκευάσουμε ένα UAV, και με δυο εναλλακτικούς τρόπους μας δείχνει πως συνδέονται οι τέσσερις κινητήρες. Μετά με τον ίδιο τρόπο όπως και στο προηγούμενο paper, γίνεται ο αεροδυναμικός σχεδιασμός, η ταυτότητα και ο μοντελισμός του UAV δίνοντάς μας και την δομή της συσκευής μετρήσεων. Τέλος, μας αναφέρει με λίγα λόγια πως γίνεται ο έλεγχος ενός UAV δίνοντας μερικές αποκρίσεις κάποιων πιθανών ελεγκτών. Η ομάδα αυτού του paper κατάφερε να σχεδιάσει και να υλοποιήσει ένα μικρό quad-rotor UAV χρησιμοποιώντας έναν ενσωματωμένο μικροελεγκτή. Η ανάπτυξη των UAV προορίζεται να εφαρμοστεί σε εναέρια επιτήρηση και σε περιβαλλοντικές μετρήσεις.

4. ΜΟΝΤΕΛΟΠΟΙΗΣΗ, ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΤΑΧΕΙΑ ΠΡΟΤΥΠΟΠΟΙΗΣΗ ΕΝΟΣ UAV MINI-ΕΛΙΚΟΠΤΕΡΟΥ

Σε αυτό το paper παρουσιάζεται ο σχεδιασμός, η προσομοίωση, και η εκτέλεση σε πραγματικό χρόνο του συστήματος ελέγχου για ένα αυτόνομο mini-ελικόπτερο, χρησιμοποιώντας software εργαλεία ταχείας προτυποποίησης. Τα εργαλεία αυτά περιλαμβάνουν το μοντέλο προσομοίωσης ενός mini-ελικόπτερου, ένα φίλτρο πλοήγησης, αρκετούς ελεγκτές, ένα εργαλείο γραμμικοποίησης του μοντέλου, και παραμέτρους και λειτουργίες ευρεστικής ταυτοποίησης. Το software περιβάλλον ταχείας προτυποποίησης (RSPE) προορίζεται στο να μειώσει τον χρόνο και το κόστος ανάπτυξης, δίνοντας στον σχεδιαστή περισσότερο χρόνο να εστιάσει στις μαθηματικές και αλγοριθμικές μεθόδους.

Στην πορεία του paper γίνεται μια περιγραφή του RSPE, λέγοντάς μας πως "έκτισε" το software αυτό μέσω του Simulink και με ποια βήματα περιέχει το software αυτό. Στην συνέχεια γίνεται η μαθηματική περιγραφή του μοντέλου ενός mini-ελικόπτερου και αμέσως μετά από αυτό μας αναλύονται περισσότερο τα modules του Colibri software, τα οποία είναι: 1) το module προσομοίωσης, 2) το module πλοήγησης, 3) το module ελέγχου, 4) το module της γραμμικοποίησης, 5) το module του επίγειου σταθμού ελέγχου και 6) το module της ταυτοποίησης του συστήματος, δίνοντας τις συνδεσμολογίες ή απεικονίσεις αυτών μέσω του Simulink. Τέλος γίνεται μια hardware-in-the-loop προσομοίωση δείχνοντάς μας τις συνδεσμολογίες των hardware στοιχείων ενός mini-ελικόπτερου, και βγάζοντας κάποια συμπεράσματα μέσω αυτής της προσομοίωσης. Η ομάδα υποστηρίζει πως με το RSPE,

είναι πιθανό να δοκιμαστούν νέες μέθοδοι (ελέγχου, παρακολούθησης, επεξεργασία σήματος, εκτίμησης, ταυτοποίησης κ.ά) με έναν εύκολο και ευέλικτο τρόπο. Το περιβάλλον ολοκληρώνει γενικά εργαλεία που είναι βασισμένα στις εργαλειοθήκες του Matlab, με την πιθανότητα πρόσθεσης νέων λειτουργιών σύμφωνα με τις νέες απαιτήσεις.

5. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΓΡΑΜΜΙΚΟΣ ΕΛΕΓΧΟΣ ΕΝΟΣ QUADROTOR

Στο paper αυτό γίνεται μια πλήρης παρουσία μια εργασίας παρόμοιας με την αναφερόμενη πτυχιακή μας. Αναλύεται σε μεγάλο επίπεδο οι εξισώσεις ενός τέτοιου συστήματος και αρκετές μέθοδοι ελέγχου για ένα τέτοιο σύστημα. Η εργασία αυτή έχει πραγματοποιηθεί σε επίπεδο προσομοίωσης στο προγραμματιστικό περιβάλλον του MATLAB-Simulink στο οποίο πραγματοποιείται εκτενείς ανάπτυξη ενός τέτοιου συστήματος.

Ειδικότερα, έγιναν δοκιμές για διάφορες τιμές προκειμένου να πραγματοποιηθεί έλεγχος του μοντέλου και της κατάλληλης διανομής τάσης στα μοτέρ. Χρησιμοποιήθηκαν τρεις τεχνικές στην συγκεκριμένη εργασία. Οι δυο πρώτες είχαν να κάνουν με εφαρμογή ενός ελεγκτή PID, και η μόνη διαφορά τους ήταν σχετικά με τις εξόδους που θα επέστρεφε στην ανάδραση, ενώ στην τρίτη τεχνική γινόταν εφαρμογή ενός ελεγκτή PID και ενός γραμμικού δευτεροβάθμιου ρυθμιστή. Τα αποτελέσματα που παρουσιάζονται δεν ήταν όλα αποδεκτά και συγκεκριμένα μόνο η τρίτη τεχνική παρείχε ένα επιθυμητό αποτέλεσμα. Το σημαντικότερο κομμάτι της πτυχιακής αυτής, το οποίο στέφθηκε με επιτυχία ήταν πως το μοντέλο ακολουθούσε επαρκώς ένα μονοπάτι που του δινόταν στην είσοδό του.

- **Έλεγχος:**

1. ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΟΛΟΚΛΗΡΩΜΕΝΟΥ ΠΛΑΙΣΙΟΥ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΠΤΗΣΗΣ ΕΝΟΣ ΜΙΚΡΟΥ UAV

Ο στόχος αυτού του paper είναι να παρουσιάσει μια συστηματική προσέγγιση για την ανάπτυξη του ολοκληρωμένου πλαισίου ελέγχου πτήσης που συνδυάζει την θεωρητική σχεδίαση εργαλείων και πειραματικών διαδικασιών ώστε οι μηχανισμοί ελέγχου να μπορούν εύκολα να συνθεθούν, να εφαρμοστούν και να δοκιμαστούν οι ελεγκτές πτήσης στα μικρά UAV συστήματα σε ένα ασφαλέστερο, από πλευράς κόστους και αποτελεσματικού χρόνου, τρόπο. Ένα COTS (commercially off the shelf) ελεγχόμενο μέσω ραδιοφώνου αεροσκάφους, οργανομένο με αεροηλεκτρικά συστήματα πτήσης χρησιμοποιείται σαν πλατφόρμα δοκιμών για να παρουσιάσει την ανάπτυξη του ολοκληρωμένου ελέγχου πτήσης και πλαίσιο δοκιμών. Μη γραμμικός μοντελισμός της δυναμικής πτήσης του UAV γίνεται χρησιμοποιώντας την αρχή της πρώτης θεωρίας του περιβάλλοντος του Matlab/Simulink με πειράματα που διεξάγονται για να καθορίσουν τις παραμέτρους του φυσικού μοντέλου. Η αναγνώριση του υπό δοκιμής συστήματος πτήσης διεξήχθη για να αναπτύξει και να επαληθεύσει την εξέλιξη του μοντέλου. Οι παραμετρικές αβεβαιότητες που αντλούνται από τα διεξαγμένα πειράματα μοντελοποιούνται σε ένα μη γραμμικό μοντέλο προσομοίωσης. Ένα απλοποιημένο αβέβαιο κάθετα γραμμικό μοντέλο χρησιμοποιείται για να συνθεθεί ο κάθετος άξονας του ελεγκτή της γωνίας κύλισης. Ο σχεδιασμένος ελεγκτής δοκιμάζεται λογισμικά και επεξεργάζεται σε βρόχο σε ολοκληρωμένο περιβάλλον δοκιμής. Το ολοκληρωμένο πλαίσιο παρέχει μια στοιχειώδη και συστηματική προσέγγιση για να δοκιμαστεί ο συνδυασμένος ελεγκτής πρώτου τοποθετηθεί πάνω στο UAV για την πραγματική δοκιμαστική πτήση.

Στην συνέχεια του υπόλοιπου paper γίνεται περιγραφή του μοντέλου προσομοίωσης του μικρού UAV, η πλατφόρμα του, το μοντέλο προώθησης, τα χαρακτηριστικά της προπέλας, το μοντέλο αδράνειας, το αεροδυναμικό μοντέλο (εξισώσεις δυνάμεων, ροπών και κινηματικές), ταυτοποίηση των δοκιμασμένων παραμέτρων πτήσης, και ο αβέβαιος μοντελισμός, στον οποίο δίνεται και ένα χαρακτηριστικό παράδειγμα. Μετά ακολουθεί η αρχιτεκτονική, η σύνθεση και η εφαρμογή του ελέγχου πτήσης δίνοντας διαγράμματα που σχετικά με τις εξισώσεις του συστήματος και τέλος γίνεται μια παρουσίαση του ολοκληρωμένου πλαισίου. Ακολουθεί ένα πείραμα, δείχνοντάς μας πως γίνεται η

συνδεσμολογία του όλου συστήματος για να γίνει η προσομοίωση και παρουσιάζονται μερικές αποκρίσεις του συστήματος όσο ν'αφορά την γωνία στρέψης. Η ομάδα αυτή κατάφερε, η επίδοση του συντεθειμένου ελεγκτή χρησιμοποιώντας το ανεπτυγμένο μοντέλο πετυχαίνει τους στόχους επίδοσης του σχεδιασμένου ελέγχου, το μοντέλο και το αβέβαιο μοντέλο που αναπτύχθηκαν έχουν επαρκή πιστότητα για τους σκοπούς σύνθεσης και ανάλυσης του ελεγκτή, και το πλαίσιο που χρησιμοποιήθηκε για την σύνθεση και την επικύρωση είναι εφικτό και πετυχαίνει τους επιδιωκόμενους στόχους.

2. ΥΒΡΙΔΙΚΟΣ ΕΛΕΓΧΟΣ ΤΟΥ ΣΧΗΜΑΤΙΣΜΟΥ ΕΝΟΣ UAV

Σε αυτό το paper προτείνεται μια αρχιτεκτονική υβριδικού επιτηρητή ελέγχου για το επίπεδο σχεδιασμού του μονοπατιού των UAV ελικοπτέρων, που εμπλέκονται σε ένα σενάριο σχηματισμού, ηγέτη-ακόλουθου. Ο υβριδικός μοντελισμός και έλεγχος, είναι ένα πολύ ισχυρό πλαίσιο που πιάνει και τη διακριτή, και τη συνεχή δυναμική ενός συστήματος, ταυτόχρονα και συλλογικά. Παρέχει μια περιληπτική ανάλυση για τις αλληλεπιδράσεις μεταξύ του διακριτού μέρους και της συνεχούς εξέλιξης του συστήματος. Υπάρχουν κάποιες προσπάθειες να γίνει σύλληψη του σχηματισμού ελέγχου μέσα σ'ένα υβριδικό πλαίσιο. Ωστόσο, έχουν ως επί το πλείστον τελειώσει με το υβριδικό μοντελισμό προκειμένου να κάνουν υβριδική ανάλυση του συστήματος. Ο λόγος είναι ότι παρά την ισχύ της θεωρίας του υβριδικού μοντελισμού, τα εργαλεία υβριδικής ανάλυσης είναι τυπικά δύσκολο να εφαρμοστούν στο σύστημα. Αντιθέτως, η προτεινόμενη προσέγγιση αυτού του paper, παρέχει ένα προσιτό πλαίσιο υβριδικής σύνθεσης για τον έλεγχο του σχηματισμού. Μέσα σε αυτό το πλαίσιο, έχουν εισάγει μια νέα μέθοδο αφαίρεσης, βασισμένη στους πολικούς διαχωρισμούς του χώρου κατάστασης. Επιπλέον, έχουνε χρησιμοποιήσει πολυσυσχετιζόμενων ιδιοτήτων συναρτήσεων κατά την διάρκεια του χωρισμένου χώρου για την κατασκευή ενός υβριδικού μοντέλου που θα μπορεί να συλληφθεί από την πεπερασμένη διακριτή περίπτωση μοντέλου συστήματος (DES). Χρησιμοποιώντας την τεχνική αυτή, μειώσουν το γνήσιο υβριδικό σύστημα με μηχανή πεπερασμένης κατάστασης που μπορεί να χειριστεί αποτελεσματικά από καθιερωμένες θεωρίες του εποπτικού ελέγχου DES. Μέσα στο πλαίσιο εποπτικού ελέγχου DES, μπορεί να σχεδιαστεί ο ελεγκτής για την επίτευξη του σχηματισμού, για την διατήρηση του, και για την αποφυγή συγκρούσεων.

Στην συνέχεια του paper γίνεται μια περιγραφή των προβλήματων που υπάρχουν πάνω στην εφαρμογή του υβριδικού μοντέλου. Μετά γίνεται μια ανάλυση των διανυσματικών πεδίων πάνω από τους πολιικούς διαχωρισμούς στον χώρο κατάστασης κάνοντας μαθηματική ανάλυση και δίνοντας γραφικές παραστάσεις. Γίνονται αναφορές για δυο είδη ελέγχου, έναν ελεγκτή για αναλοιώτη περιοχή και έναν ελεγκτή για την έξοδο από μια άκρη και στην συνέχεια παρουσιάζεται η κατασκευή του ελεγκτή με μαθητική ανάλυση και διάφορα σχήματα σχετικά. Τέλος, ακολουθεί μια πολύ λεπτομερή περιγραφή της δουλειάς της ομάδας αυτής με αποκρίσεις του συστήματος και πολλά άλλα. Η ομάδα αυτή πρότεινε μια νέα προσέγγιση υβριδικού ελέγχου επιτήρησης για το πρόβλημα σχηματισμού ηγέτη-ακόλουθου. Η προσέγγιση ήταν βασισμένη στους πολιικούς διαχωρισμούς του χώρου κατάστασης. Η διαδικασία σχεδιασμού του ελεγκτή ήταν κλιμακώμενη, δηλαδή σχεδιάστηκε ο ελεγκτής για τον έλεγχο του σχηματισμού, και ο ελεγκτής για την αποφυγή συγκρούσεων, ξεχωριστά, και μετά τους συνδύασαν χρησιμοποιώντας παράλληλη σύνδεση, για να πετύχουν τον όλο ελεγκτή που θα ικανοποιεί τις γενικές προδιαγραφές.

3. ΠΕΙΡΑΜΑΤΙΚΟ ΠΛΑΙΣΙΟ ΣΧΕΔΙΑΣΜΟΥ ΕΛΕΓΚΤΗ ΓΙΑ UAV ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΣΥΣΤΗΜΑΤΑ ΜΕ ΠΟΛΛΕΣ ΚΑΜΕΡΕΣ

Σε αυτό το paper, προτείνεται ένα πειραματικό πλαίσιο που εκτιμά τα χαρακτηριστικά των rotorcraft UAV για την σχεδίαση και την επικύρωση ενός συστήματος ελέγχου εφαρμόζοντας την εσωτερική ιδέα των πλατφόρμων δοκιμών χρησιμοποιώντας ένα σύστημα όρασης. Η εξέλιξη του συστήματος ελέγχου του rotorcraft UAV ξεκινά με μη γραμμική μοντελοποίηση του οχήματος που ακολουθείται από την σχεδίαση του ελεγκτή χρησιμοποιώντας μια αριθμητική προσομοίωση και ολοκληρώνεται με μια δοκιμαστική πτήση. Το πειραματικό αυτό πλαίσιο πραγματοποιεί την διαδικασία εξέλιξης συστηματικά για την κατασκευή του quad-rotor UAV χρησιμοποιώντας τους 3 βαθμούς ελευθερίας του πτητικού μήλου σαν ρύθμιση για τον έλεγχο της συμπεριφοράς και συστήματος με πολλές κάμερες. Το προτεινόμενο πλαίσιο μπορεί να εφαρμοστεί σε ποικίλους τύπους rotorcraft UAV με χαμηλό κόστος σε μικρό χρονικό διάστημα. Επιπλέον, εξαιτίας του ελέγχου της θέσης και της ταχύτητας εκτελείται μετά από επαρκείς επιδόσεις του ελέγχου της συμπεριφοράς, ο οποίος είναι ο κεντρικός βρόχος του συστήματος ελέγχου, επιβεβαιωμένο, ο σχεδιασμός του ελεγκτή μπορεί να δοκιμαστεί αποτελεσματικά και με ασφάλεια.

Στην πορεία του paper παρουσιάζεται το πειραματικό πλαίσιο της εργασίας, δηλαδή το quad-rotor UAV πως είναι η μορφή του, από τι χαρακτηρίζεται η δομή του, δίνοντας σχήματα σχετικά. Επίσης αναφέρονται οι βαθμοί ελευθερίας του μύλου και το σύστημα πολλαπλών καμερών. Στην συνέχεια γίνεται ανάλυση του μοντελισμού του quad-rotor UAV και ο σχεδιασμός του ελεγκτή, δίνοντας μαθηματικές περιγραφές, το πως λειτουργούν οι κινητήρες προκειμένου να επιτευχθεί η κίνηση, block-διαγράμματα, καθώς και ένα block-διάγραμμα από Simulink και τέλος γίνεται μια περιγραφή των αποτελεσμάτων του πειράματος. Η ομάδα αυτή κατάφερε να σχεδιάσει έναν ελεγκτή για το UAV μέσα από το πειραματικό πλαίσιο χρησιμοποιώντας σύστημα με πολλές κάμερες. Το προτεινόμενο αυτό πλαίσιο παρέχει πλήρη πλοηγικές πληροφορίες 6 βαθμών ελευθερίας με υψηλή ακρίβεια και επικοινωνία μεταξύ του σκάφους και του επίγειου υπολογιστικού συστήματος για την πραγματοποίηση μιας αποτελεσματικής διαδικασίας σχεδιασμού του ελεγκτή.

4. ΕΝΣΩΜΑΤΩΜΕΝΗ ΒΑΣΗ ΥΠΟΛΟΓΙΣΤΗ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΜΗ ΓΡΑΜΜΙΚΟΥ ΟΧΗΜΑΤΟΣ

Σε αυτό το paper περιγράφεται ο σχεδιασμός και η αρχική υλοποίηση του ελέγχου σε μια πειραματική εγκατάσταση ενός μικρού quad-rotor UAV που προορίζεται για έλεγχο σε εσωτερικούς χώρους. Η υλοποίηση είναι ένα σημαντικό βήμα στην διαδικασία εξέλιξης μιας πιο προηγμένης υλοποίησης που είναι κατάλληλη για εφαρμογές, στοχεύοντας στην διασάφηση στις αρχές του ελέγχου, στην απόκτηση εμπειρίας για τις λύσεις του έργου ελέγχου, και η παροχή δεξιοτήτων για την εξέλιξη των περαιτέρω υλοποιήσεων. Στα επόμενα κεφάλαια, αρχικά περιγράφεται η κατασκευή ενός πειραματικού quad-rotor UAV, και τέλος δίνονται τα θέματα εκτέλεσης και ένα παράδειγμα προσομοίωσης.

Συγκεκριμένα στην πορεία του paper παρουσιάζεται το πως μπορούμε να κατασκευάσουμε ένα UAV και το πως πρέπει, σύμφωνα με αυτούς, να γίνει η συνδεσμολογία των τεσσάρων κινητήρων, δίνοντάς και σχήμα με την συνδεσμολογία αυτή και το πως θα φαίνεται στο τέλος το quad-rotor UAV. Στην συνέχεια γίνεται αεροδυναμική μέτρηση με μια συνδεσμολογία του ρότορα για να γίνει ρύθμιση της μέτρησης. Μετά αναφέρεται η ιεραρχία του ελέγχου και τέλος ο έλεγχος ενός quad-rotor UAV. Η ομάδα αυτή παρουσίασε τον σχεδιασμό και την υλοποίηση του ελέγχου σε ένα πείραμα σε εσωτερικούς χώρους.

5. ΕΛΕΓΧΟΣ ΠΕΙΡΑΜΑΤΙΚΟΥ QUAD-ROTOR UAV

Σε αυτό το paper γίνεται περιγραφεί του σχεδιασμού και της υλοποίησης του ελέγχου σε μια πειραματική ρύθμιση ενός quad-rotor UAV που προορίζεται για χρησιμοποίηση σε ελέγχους εσωτερικού χώρου. Αυτή η υλοποίηση είναι ένα σημαντικό βήμα στην διαδικασία εξέλιξης για μια πιο ανεπτυγμένη υλοποίηση που θα είναι κατάλληλη για εφαρμογές, στοχεύει στην διευκρίνιση των αρχών ελέγχου, στην απόκτηση εμπειρίας για τις λύσεις του έργου ελέγχου, και η παροχή δεξιοτήτων για την εξέλιξη των περαιτέρω υλοποιήσεων.

Στην πορεία του paper αυτού γίνεται η περιγραφεί της κατασκευής του UAV λέγοντάς μας τι hardware υλικό χρησιμοποιήθηκε ολοκληρώνοντας ουσιαστικά αυτό που είχε κάνει η ομάδα και στα 2 προηγούμενα paper. Μετά γίνεται η ιεραρχία του ελέγχου, στην οποία ουσιαστικά μας περιγράφει τους ελέγχους και με τι σειρά θα εφαρμόζονται αυτοί. Και τέλος γίνεται ο έλεγχος του UAV και σε αυτό το κομμάτι περιγράφονται δυο μέθοδοι, η μέθοδος backstepping και παρουσιάζεται και η δομή της, η μέθοδος εκτίμησης κατάστασης και παρουσιάζεται και η δομή της, και μετά ακολουθεί η προσομοίωση. Η ομάδα αυτή στο paper αυτό περιγράφει ένα ιεραρχικό σχήμα ελέγχου σε ένα ενσωματωμένο μοντέλο το οποίο είναι χτυσμένο στην ιδέα του backstepping, και εφαρμόζεται σε έναν ηλεκτρικό κινητήρα-οδηγό στο hardware του quad-rotor UAV που είναι εξοπλισμένο με έναν υπολογιστή στο "σκάφος", αισθητήρα αδρανειακής μονάδας, καθώς επίσης και εγκαταστάσεις που είναι κατάλληλες να συμμετέχουν σε σύστημα εντοπισμού θέσης σε έναν εσωτερικό χώρο, και σε δίκτυο ασύρματης ψηφιακής επικοινωνίας.

- **Πλοήγηση:**

1. ΠΡΟΣΑΡΜΟΣΤΙΚΟΣ ΑΥΤΟΜΑΤΟΣ ΠΙΛΟΤΟΣ ΓΙΑ ΤΗΝ ΔΙΑΤΗΡΗΣΗ ΠΟΡΕΙΑΣ

Η προσέγγιση που γίνεται σε αυτό το paper προσπαθεί να ξεπεράσει την ανάγκη για τα μαθηματικά μοντέλα των πλοίων χρησιμοποιώντας απροσδιόριστο λογικό αυτόματο πιλότο, που είναι επαυξημένος από την δυνατότητα προσαρμογής των συντελεστών κλιμάκωσής της. Ο προτεινόμενος προσαρμοστικός αυτόματος πιλότος εμφανίζεται ως μια βιώσιμη πρακτική εναλλακτική λύση για την παράκτια ιστιοπλοΐα, όπου η διατήρηση της πορείας είναι ζωτικής σημασίας σε όλες τις περιστάσεις. Η κύρια εστίαση αυτού του paper είναι η ανάπτυξη ενός προσαρμοστικού αυτόματου πιλότου για την διατήρηση της πορείας, που παρέχει μια on-line τροποποίηση από τους συντελεστές κλίμακας για κάθε μεταβλητή. Ο στόχος είναι να παρουσιάσει με πως η προσαρμοστική έννοια μπορεί να εφαρμοστεί, για να βελτιώσει τις απροσδιόριστες λογικές τεχνικές όταν ενεργούν σύνθετες διαταραχές (θαλάσσια ρεύματα και κύματα) πάνω στο πλοίο.

Σε αυτό το paper γίνεται ουσιαστικά μια **ανάλυση** του συστήματος με εξισώσεις και σε ένα block διάγραμμα, και στο τέλος γίνεται μια **σύγκριση** του συστήματος αυτού με ένα πρότυπο σύστημα (το Sugeno) και για να γίνει η σύγκριση αυτή χρησιμοποιούμε εξομοίωση. Τα αποτελέσματα της σύγκρισης αυτής μας δείχνουν πως ο απροσδιόριστος αυτόματος πιλότος έδειξε πολύ καλύτερος σε επίδοση σε σχέση με τον αυτόματο πιλότο Sugeno. Όποτε με την σύγκριση αυτή βλέπουμε πως η ομάδα αυτή κατάφερε να δημιουργήσει έναν αυτόματο πιλότο με καλύτερες επιδόσεις από αυτόν του Sugeno.

2. ΟΠΤΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ ΠΛΟΗΓΗΣΗΣ ΓΙΑ ΤΑ MICRO-ΕΝΑΕΡΙΑ ΟΧΗΜΑΤΑ

Αυτό το paper ασχολείται με τις νέες τεχνολογίες οπτικής ανίχνευσης για την σταθεροποίηση και την καθοδήγηση των μη επανδρωμένων micro-εναέριων οχημάτων (micro-UAVs). Σε αυτό το paper, η γενική ιδέα της κοινής βελτίωσης της οπτικής, ανίχνευσης, και επεξεργασία πτυχών του συστήματος απεικόνισης χρησιμοποιείται για να αντιμετωπίσει τις αυστηρές απαιτήσεις του επίπεδου συστήματος.

Η ομάδα αυτή παρουσίασε μια ιδέα σχεδιασμού ενός πλακιδίου οπτικού αισθητήρα πλοήγησης. Έκανε πειράματα σε εικονικό περιβάλλον, κατασκευάζοντας ένα εικονικό αισθητήριο για micro-εναέρια οχήματα και πήρε διάφορες απεικονίσεις εσωτερικών χώρων.

Η πτητική συμπεριφορά που είναι εμπνευσμένη από έντομα στα micro-UAVs (δηλαδή αποφυγή εμποδίων, φυσική σταθεροποίηση, αναζήτησης τροφής) μπορεί να επιτευχθεί από νέες τεχνολογίες οπτικής ανίχνευσης. Επίσης κατέδειξαν πως ένας αισθητήρας, σαν αυτόν που παρουσίασαν, δίνει την δυνατότητα να συλλάβει εικόνες με εστίαση 60%, του πλήρη χώρου με πολύ μικρή παραμόρφωση.

3. ΠΤΗΣΗ ΧΑΜΗΛΟΥ ΥΨΟΜΕΤΡΟΥ ΑΚΟΛΟΥΘΩΝΤΑΣ ΕΝΑΝ ΔΡΟΜΟ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ

ΚΑΜΕΡΑ

Σε αυτό το paper, διευθετείται το πρόβλημα του εντοπισμού μιας οπτικά διακριτά συνδεδεμένης δομής στο έδαφος χρησιμοποιώντας ένα πολύ μικρό εναέριο όχημα εξοπλισμένο με κάμερα με μάντα προς τα κάτω. Ενδεχόμενες δομές συμπεριλαμβανομένων δρόμων, αγωγών, και περιμετρικά από δασικές πυρκαγιές. Γένικα στο paper αυτό η ομάδα θα εστιάσει την προσοχή της στην ακολούθηση του δρόμου.

Στην πορεία του paper η ομάδα αυτή κάνει μια παρουσίαση της αρχιτεκτονικής που θα χρησιμοποιηθεί στην εργασία αυτή δείχνοντάς μας ένα σχετικό block διάγραμμα. Στην συνέχεια γίνεται μια μαθηματική και θεωρητική προσέγγιση της ταξινόμησης του δρόμου και η ανάλυση της κατεύθυνσης του εναέριου οχήματος. Μετά μας αναλύουν πως το όχημα αυτό ακολουθεί έναν δρόμο, και τις ειόνες που αυτό μπορεί να παίρνει, και τι γίνεται σε περίπτωση που αυτό αποκλείνει από την πορεία του, όπως επίσης μας εξηγεί και την περίπτωση που ο δρόμος κάνει καμπύλη, ή υπάρχει στροφή 180 μοιρών. Η ομάδα κατάφερε να δείξει πως ο βρόχος ακολουθίας του δρόμου δούλεψε και στην προσομοίωση και στο κανονικό τέστ πτήσης. Ο δρόμος εξάχθηκε επιτυχώς από την εικόνα χρησιμοποιώντας τεχνικές στατιστικής ταξινόμησης στο χώρο HSV χρωμάτων. Ο βρόχος καθοδήγησης μπορεί να τροποποιηθεί για να ακολουθεί άλλα οπτικά διακριτά όρια. Για παράδειγμα, με τις φωτιές των δασών, ο ταξινομητής του δρόμου μπορεί να προσαρμοστεί να ακολουθεί ζεστές/κρύες περιμέτρους αντί για δρόμους. Σε αυτές τις άλλες περιπτώσεις, οι περιορισμοί που έχουν δοθεί σε αυτό το paper, έχουν άμεση εφαρμογή.

4. ΠΛΟΗΓΗΣΗ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΟΡΑΣΗ ΓΙΑ ΤΑ UAV

Σε αυτό το paper, παρουσιάζεται μια στρατηγική πλοηγού βασισμένου στην όραση για την κάθετη απογείωση και προσγείωση ενός μη επανδρωμένου εναέριου οχήματος χρησιμοποιώντας μια απλή ενσωματωμένη κάμερα παρατηρώντας τα φυσικά ορόσημα. Σε αυτή την προτεινόμενη προσέγγιση, οι εικόνες του περιβάλλοντος πρώτα, δειγματίζονται,

αποθηκεύονται και οργανώνονται σαν ένα σύνολο ταξινομημένων βασικών εικόνων (εικονικό μονοπάτι), η οποία παρέχει μια εικονική μνήμη του περιβάλλοντος. Το έργο πλοήγησης του ρομπότ, τότε ορίζεται σαν μια αλληλουχία υποσύνολων μονοπατιών (επωνομαζόμενη εικονική διαδρομή) συνδέοντας την τρέχουσα παρατηρούμενη εικόνα με μια εικόνα-στόχο, που ανήκει στην εικονική μνήμη. Το UAV ελέγχεται για να φθάσει κάθε εικόνα από την εικονική διαδρομή χρησιμοποιώντας έναν νόμο ελέγχου βασισμένο στην όραση που προσαρμόζεται στο δυναμικό μοντέλο του και χωρίς να προγραμματίζεται με σαφήνεια κάποια τρόχια. Αυτό το πλαίσιο είναι ευρέως τεκμηριωμένο από πειράματα με ένα X-4 flyer εξοπλισμένο με μια κατοπτρική κάμερα.

Στην πορεία του paper, γίνεται μια παρουσίαση του πως λειτουργεί η εικονική μνήμη, δίνοντας και ένα σχήμα επιλογής "εικόνας", ουσιαστικά διαδρομής, καθώς επίσης και η διαδρομή ενός UAV μέσα σε κτίριο. Μετά παρουσιάζεται ο εντοπισμός μιας εικόνας μέσα από ένα ευρύ φάσμα επιλογών στην μνήμη. Στην συνέχεια αναφέρεται στο πως το UAV ακολουθεί μια διαδρομή, κάνοντας μοντελοποίηση του οχήματος, μετά παρουσιάζεται ο στόχος του ελέγχου και μετά η σχεδίαση του ελέγχου. Ακολούθως αναφέρεται σε ένα μεγάλο κομμάτι της εργασίας στην εκτίμηση της κατάστασης από το γενικό μοντέλο της κάμερας, κάνοντας μια λεπτομερή μαθηματική ανάλυση, και στην συνέχεια δείχνοντάς μας πως το UAV βάζει σε μια σειρά της εικόνες που λαμβάνει. Η ομάδα αυτή μετά τα αποτελέσματα του πειράματος έφτασε στο συμπέρασμα πως ένα ολοκληρωμένο πλαίσιο για αυτόνομη πλοήγηση για ένα UAV, η οποία ενεργοποιεί ένα όχημα να ακολουθεί ένα εικονικό μονοπάτι, λαμβάνεται κατά την διάρκεια του μαθησιακού στάδιου χρησιμοποιώντας μια απλή κάμερα και φυσικά ορόσημα έχουν προταθεί. Το περιβάλλον του ρομπότ παρουσιάζεται σαν ένα διάγραμμα εικονικών μονοπατιών, που ονομάζεται εικονική μνήμη από την οποία μια εικονική διαδρομή που συνδέει τις αρχικές και τις τελικές εικόνες που μπορούν να αποσυμπιεστούν. Το όχημα μπορεί τότε να οδηγεί κατά μήκος της εικονικής διαδρομής χάρις σε έναν νόμο ελέγχου βασισμένο στην όραση, η οποία λαμβάνει υπόψη της το δυναμικό μοντέλο του ρομπότ. Επιπλέον, η κατάσταση του ρομπότ, που απαιτείται για τον υπολογισμό του νόμου ελέγχου, εκτιμάται χρησιμοποιώντας ένα γενικό μοντέλο κάμερας βασισμένο στην προοπτική, στην κατοπτρική καθώς και σε μια μεγάλη κατηγορία από κατοπτρικές κάμερες. Τα πειράματα με το X-4 flyer εξοπλισμένο με μια κατοπτρική κάμερα έδειξε την εγκυρότητα της προτεινόμενης προσέγγισης.

5. ΠΡΟΤΑΣΗ ΠΛΟΗΓΗΣΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΟΣ ΕΛΕΓΧΟΥ ΓΙΑ ΜΙΚΡΟ UAV

Σε αυτό το paper η ομάδα αυτή παρουσιάζει το project του συστήματος ελέγχου του UAV συνειδητοποιήθηκε στο Department of Avionics and Control Systems of Rzeszów University of Technology. Το project βασίζεται σε προηγούμενες εμπειρίες. Στο paper αυτό εμφανίζεται η γενική δομή του onboard συστήματος ελέγχου καθώς επίσης και η δομή του σταθμού ελέγχου. Αυτά περιγράφονται οι προτεινόμενες διαδικασίες ελέγχου και πλοήγησης καθώς επίσης και ο ανθρώπινος παράγοντας, εντοπισμός αστοχίας και αναδιαμόρφωση του συστήματος.

Στην πορεία της εργασίας γίνεται η γενική περιγραφή του συστήματος ελέγχου, δηλαδή το onboard σύστημα ελέγχου και τον σταθμό ελέγχου στο έδαφος δίνοντας μας και δυο block διαγράμματα, ένα για το κάθε σύστημα. Μετά μας παρουσιάζουν τα σημαντικά προβλήματα στην έρευνα του έργου αυτού, συνοπτικά είναι η ταυτότητα του μαθηματικού μοντέλου και οι βελτιώσεις, οι διαδικασίες ελέγχου και πλοήγησης, ο ανθρώπινος παράγοντας στο σύστημα ελέγχου σε ένα UAV, και τέλος η αποτυχία εντοπισμού και το σύστημα αναδιάρθρωσης. Η ομάδα αυτή έφτασε στο συμπέρασμα πως η προτεινόμενη προσέγγιση στον τομέα των συστημάτων ελέγχου δίνει την πιθανότητα της ανάλυσης της επιρροής του ανθρώπινου παράγοντα σε ένα UAV. Υπάρχουν κάποιες προτάσεις αυτόματων πιλότων για ένα μικρό UAV με κλειστή δομή. Χρήση ενός CAN bus με CANAerospace πρωτόκολλο κάνει το σύστημα ανοιχτό και γενικό. Μεγάλα ερευνητικά προβλήματα εμφανίζονται όχι μόνο σε UAV αλλά επίσης και σε επανδρωμένα αεροσκάφη. Ο διαδηλωτής της τεχνολογίας και η χαμηλή σε κόστος δοκιμής κατά την πτήση και επίσης παρέχει νέες πιθανότητες για την εκπαίδευση και εργαστηριακά μαθήματα κατά την πτήση, τα οποία είναι σημαντικά για το πανεπιστήμιο.

