ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Δ.Ι.Π.Α.Ε ΘΕΣΣΑΛΟΝΙΚΗΣ

ΒΕΛΤΙΣΤΟΣ ΨΗΦΙΑΚΟΣ ΕΛΕΓΧΟΣ ΤΕΤΡΑΚΟΠΤΕΡΟΥ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΝΤΖΕΜΠΕΗΣ ΧΡΥΣΟΣΤΟΜΟΣ

ΑΜ: 153438

Επιβλέπων καθηγητής

Χρήστος Υφούλης (Αναπληρωτής καθηγητής)

Περίληψη

Η συγγραφή της παρακάτω διπλωματικής εργασίας υλοποιείται στα πλαίσια του πτυχίου του τμήματος Μηχανικών παραγωγής και διοίκησης (Τμήμα Μηχανικών Αυτοματισμού) . Αρχικά, πραγματοποιείται περιήγηση στα γενικά χαρακτηριστικά καθώς και στα ιστορικά στοιχεία ενός μη επανδρωμένου αεροσκάφους, με το τετρακόπτερο να αποτελεί το κύριο μέρος της εργασίας. Έπειτα, κατασκευάζεται το μη γραμμικό μαθηματικό μοντέλο μελετώντας την κινηματική και δυναμική του τετρακοπτέρου όπου στην συνέχεια πραγματοποιείτε η γραμμικοποιήση του μοντέλου μέσω Matlab , αναλύεται ο βέλτιστος έλεγχος και η εφαρμογή των βέλτιστων ελεγκτών(LQR & LQG). Στο τελευταίο κεφάλαιο, παρουσιάζονται οι προσομοιώσεις για διάφορες καταστάσεις της κίνησης του τετρακοπτέρου, τα 3D animations που δημιουργήθηκαν για την ρεαλιστική αποτύπωση της τροχιάς του, αλλά και ανάλυση της επίδρασης των ελεγκτών. Τέλος, παρατίθεται ο κώδικας που δημιουργήθηκε και τα συμπεράσματα που προέκυψαν.

Abstract

This paper was written as part of my undergraduate studies in Industrial Engineering and Management (Department of Automation Engineering) of International Hellenic University. To begin with, there is and introduction focusing on the general and historical evidence on UAVs; including the quadcopter, that is the main topic of the study. The creation of the non-linear mathematical model will follow, profiting from the study of the kinematic and dynamic analysis of the quadcopter. The next chapter deals with the linearization of the model in Matlab and the analysis of the optimum control and the optimum controllers (LQR & LQG). Furthermore, the simulations for different situations and the 3D animations used for the realistic representation of the quadcopter's track will be presented, as well as the analysis of the controllers' impact. In the end, the created code will be presented with the aim of reaching conclusions.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη i

Abstract ii

1.ΕΙΣΑΓΩΓΗ 1

2.ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΕΡΟΣΚΑΦΗ (UAVs) 2

2.1 Ορισμός και γενικά χαρακτηριστικά 2

2.2 Κατηγορίες UAV και επιμέρους εφαρμογές 2

2.3 Ιστορικά στοιχεία 6

3.ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΕΤΡΑΚΟΠΤΕΡΟΥ (QUADCOPTER) 8

3.1 Περιγραφή λειτουργίας 8

3.2 Γωνίες Euler 10

3.3 Δυναμικό μοντέλο 12

3.3.1 Κινηματική και δυναμική ανάλυση 12

3.3.2 Δυνάμεις-Ροπές 15

3.4 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ 16

3.4.1 Μη γραμμικό μοντέλο 16

3.4.2 Μη γραμμικό μοντέλο-Simulink 17

3.4.2 Γραμμικοποιήση 20

3.4.3 Γραμμικοποιήση του μοντέλου στο Simulink 21

3.4.4 Γραμμικό μοντέλο 26

4.ΕΛΕΓΧΟΣ ΤΕΤΡΑΚΟΠΤΕΡΟΥ 29

4.1 Βέλτιστος έλεγχος 29

4.2 Γραμμικός τετραγωνικός ρυθμιστής (LQR) 30

4.2.1 Προσομοίωση LQR στο Simulink 32

4.3 Παρατηρητής κατάστασης-Αρχή διαχωρισμού 33

4.4 Φίλτρο Kalman 34

4.5 Γραμμικός τετραγωνικός γκαουσιανός έλεγχος (LQG) 36

4.2.1 Προσομοίωση LQG στο Simulink 40

5. ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΚΑΙ ANIMATIONS (MATLAB-SIMULINK) 45

5.2 LQR-Tuning (Q&R) 47

5.1 LQR- Έλεγχος ύψους αιώρησης (Hovering) 50

5.2 LQR- Έλεγχος γωνίας περιστροφής (roll ΄φ΄) 56

5.3 LQG-Εξομοίωση με αρχικές συνθήκες 60
5.4 LQG- Έλεγχος ύψους αιώρησης (Hovering) 63
5.8 LQR-LQG Εφαρμογή ελέγχου στα μη-γραμμικά μοντέλα 68
5.8.1 Βηματική είσοδος 69
5.8.2 Waypoints path 72
5.9 Εμπόδια 3D Animations 74
6.ΚΩΔΙΚΑΣ ΚΑΙ GIFS 85
7. ΣΥΜΠΕΡΑΣΜΑΤΑ 86
ΒΙΒΛΙΟΓΡΑΦΙΑ 87

1.ΕΙΣΑΓΩΓΗ

Η τεχνολογική πρόοδος του 21^{ου} αιώνα, είχε ως αποτέλεσμα μεταξύ άλλων σημαντικές αλλαγές στον τομέα των μη επανδρωμένων αεροσκαφών UAV (Unmanned Aerial Vehicles) . Έχει παρατηρηθεί ότι συνεχώς προκύπτουν βελτιώσεις των ήδη υπάρχοντών εφαρμογών, αλλά και νέες , που αποσκοπούν όχι μόνο στην αποτελεσματικότητα των UAV αλλά και στην ασφάλεια της χρήσης αυτών. Τα μη επανδρωμένα αεροσκάφη χρησιμοποιούνται ευρέως σε μια πληθώρα εφαρμογών, όπου η παρέμβαση του ανθρώπου είναι αδύνατη , περιττή , και μερικές φορές πολύ επικίνδυνη. Μερικές από αυτές τις εφαρμογές είναι η εναέρια χαρτογράφηση και αποτύπωση του εδάφους, η επιθεώρηση πετρελαίου και φυσικού αερίου, αλλά και γενικά σε διάφορους κλάδους ορυκτών πόρων, οι μεταφορές φαρμάκων, εμβολίων, οι μετρητικές εφαρμογές γενικά, οι 3D μοντελοποιήσεις και εφαρμογές όρασης υπολογιστών , η διαχείριση δασικών εκτάσεων κ.α. Σημαντικό ρόλο παίζει ακόμα σε προγράμματα άμυνας και στρατηγικής σε αρκετές χώρες του κόσμου (Εικόνα 1.0).

Εικόνα 1.0: Αμερικανικά UAVS

Πηγή:https://en.wikipedia.org/wiki/UAVs_in_the_U.S._military#/media/File:Group_photo_of_aerial_demonstrators_at_the_2005_Naval_Unmanned_Aerial_Vehicle_Air_Demo_(cropped).jpg

2. ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΕΡΟΣΚΑΦΗ (UAVs)

2.1 Ορισμός και γενικά χαρακτηριστικά

Τα μη επανδρωμένα αεροσκάφη, ευρέως γνωστά ως UAVs (Unmanned Aerial Vehicles) ή με τον αγγλικό όρο "drones", είναι ιπτάμενα οχήματα που πραγματοποιούν πτήσεις χωρίς κυβερνήτη/χειριστή. Οι πτήσεις αυτές μπορούν να είναι αυτόνομες, προγραμματισμένες ή να ελέγχονται σε πραγματικό χρόνο από τηλεχειριστήριο. Τα UAVs κινούνται πάνω από την επιφάνεια της γης, κυρίως στον αέρα, καθώς και πάνω ή κάτω από την θάλασσα. Στην όψη μοιάζουν με μικρά αεροπλάνα ή ελικόπτερα και χαρακτηρίζονται από τους ειδικούς κινητήρες και έλικες από τους οποίους είναι κατασκευασμένα.

2.2 Κατηγορίες UAV και επιμέρους εφαρμογές

Η κατηγοριοποίηση των μη επανδρωμένων εναέριων οχημάτων προκύπτει από κριτήρια όπως: α. το μέγεθος, β. την μηχανολογική κατασκευή, δηλαδή τον αριθμό των ροτόρων και γ. τις τεχνικές πτήσης και ανύψωσης .

Α. Το μέγεθος

Βάσει του πρώτου κριτηρίου, τα μεγέθη των drones διακρίνονται σε nano, mini, μεσαία και μεγάλα. Αξίζει να σημειωθεί ότι με τον όρο μέγεθος εννοούμε το σύνολο του πλαισίου και των κινητήρων, πλην των πτερυγίων.

- Όσον αφορά στα nano οχήματα, αυτά αποτελούν το μικρότερο δυνατό μέγεθος με τις συνήθεις τιμές να βρίσκονται στα 2 - 100 χιλιοστά. Σε εξαιρετικές περιπτώσεις έχουν κατασκευαστεί UAVs ακόμα μικρότερου μεγέθους. Η χρησιμότητά τους έγκειται στον χώρο της ψυχαγωγίας και αφορά κυρίως αρχαρίους που επιθυμούν να πραγματοποιήσουν πτήσεις στα πλαίσια προσωπικής δημιουργικής απασχόλησης.
- Η υποκατηγορία των mini UAVs ξεκινάει από τα 100 χιλιοστά και μπορεί να φτάσει έως τα 350. Σε τέτοιου τύπου μη επανδρωμένα αεροσκάφη είναι συνηθισμένο να ενσωματώνονται κάμερες ή ραντάρ. Η χρήση τους είναι παρόμοια με την προηγούμενη κατηγορία, δηλαδή ιδανικά για μη έμπειρους χρήστες.
- Τα drones μεγέθους 350 – 600 χιλιοστών ανήκουν στην μεσαία κατηγορία. Σε αυτά βρίσκονται ενσωματωμένα, εκτός από κάμερες, κάθε είδους εξαρτήματα, απαραίτητα για την εκάστοτε

λειτουργία. Απευθύνονται σε εξειδικευμένους χρήστες, καθώς έχουν τη δυνατότητα να δια νύουν μεγάλες αποστάσεις και σε υψηλότερες ταχύτητες.

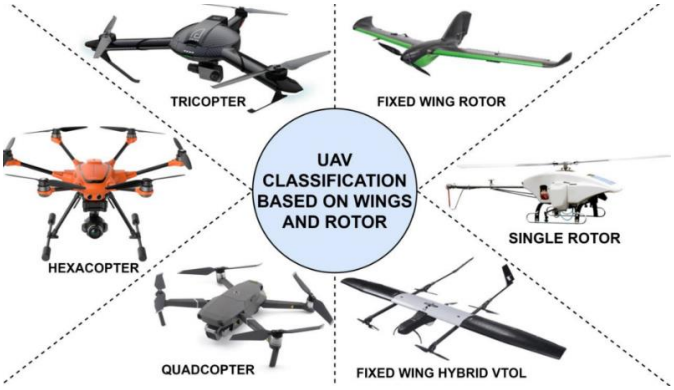
- Τα μεγέθη των μεγάλων UAVs κυμαίνονται από 0,6 μέτρα και πάνω. Είναι ειδικά κατασκευασμένα προκειμένου να μεταφέρουν φορτία έως 30 κιλά. Η δυνατότητα αυτή οφείλεται στα ανθεκτικά κατασκευασμένα πλαίσια, καθώς και τους κινητήρες. Το μεγάλο τους μέγεθος και η αντοχή σε βάρος τα καθιστούν ιδανικά για μεγάλη διάρκεια πτήσεων και την προσέγγιση δύσκολων περιοχών.

Β. Μηχανολογική κατασκευή

Μια ακόμη κατηγοριοποίηση των μη επανδρωμένων εναέριων οχημάτων γίνεται με βάση τον αριθμό των ροτόρων. Σε αυτό το σημείο αξίζει να αναφερθεί το γεγονός ότι οι περιστρεφόμενοι έλικες των UAVs είναι αυτοί που τα καθιστούν ικανά για πτήση. Ανάλογα με τον αριθμό των ροτόρων που διαθέτουν, κατατάσσονται σε μονοκόπτερα, τρικόπτερα, τετρακόπτερα, εξακόπτερα, οκτακόπτερα κ.ό.κ.

- Μονοκόπτερα ονομάζονται τα μη επανδρωμένα οχήματα τα οποία αποτελούντα από ένα ρότορα, ένα ενιαίο πτερύγιο που φέρει στην άκρη του μοτέρ. Αποτέλεσμα αυτής της μηχανολογικής κατασκευής, σε ό,τι αφορά τον τομέα της πτήσης, είναι η περιστροφή του γύρω από το κέντρο μάζας του. Δεν είναι ιδιαίτερα διαδεδομένα, παρόλα αυτά συμφέρουν από άποψη μεγέθους και μεταφοράς.
- Τα drones με τρεις ρότορες είναι ευρέως γνωστά ως τρικόπτερα. Τα τρικόπτερα είναι κατασκευασμένα με τέτοιο τρόπο, ώστε η τριγωνική διάταξη των ροτόρων να προσφέρει τη δυνατότητα ισορροπίας, ακόμα και για πολλή ώρα στο ίδιο σημείο, ενώ η πτήση τους ορίζεται κατά κανόνα από την μία έλικα.
- Τα τετρακόπτερα, που αποτελούν και τα κύριο θέμα της παρούσας έρευνας, όπως δηλώνει και το όνομά τους, αποτελούνται από τέσσερις ρότορες. Πρόκειται για την πιο γνωστή κατηγορία UAV, καθώς επιλέγεται από τους χρήστες για μια σειρά παραγόντων. Αρχικά, η κατασκευή τους προσφέρει σταθερότητα και σιγουριά, χάρη στους άρτια τοποθετημένους ρότορες. Επιπλέον, αυτός ο αριθμός τα καθιστά ασφαλή σε περίπτωση σφάλματος σε κάποιον από τους κινητήρες, καθώς όπως είδαμε και παραπάνω, τα drones μπορούν να διατηρήσουν πτήση και με λιγότερους. Τελευταίο παράγοντα αλλά εξίσου σημαντικό αποτελεί το κόστος. Αν και διαθέτει περισσότερα εξαρτήματα από τις άλλες κατηγορίες, οι τιμές τους βρίσκονται σε λογικά επίπεδα, γεγονός που τα καθιστά τόσο διαδεδομένα στο ευρύ κοινό.

- Η επόμενη κατηγορία μη επανδρωμένων αεροσκαφών με έξι ρότορες εντάσσεται στη σφαίρα των «πολυκόπτερων», μαζί με τα οκτακόπτερα κ.ό.κ. Τα εξακόπτερα χαρακτηρίζονται από μεγαλύτερη ασφάλεια σχεδιασμού σε σχέση με τα προαναφερθέντα είδη, στο ενδεχόμενο παρουσίασης βλάβης. Έχουν τη δυνατότητα διάνυσης μεγάλων αποστάσεων και επιπλέον αναπτύσσουν μεγαλύτερες ταχύτητες. Όσο περισσότεροι είναι οι ρότορες, τόσο αυξάνεται η ικανότητα προσέγγισης των εκάστοτε επιθυμητών σημείων με ακρίβεια.



Εικόνα 2.0: Ταξινόμηση

Πηγή:https://www.researchgate.net/publication/344592031_A_Comprehensive_Review_of_Unmanned_Aerial_Vehicle_Attacks_and_Neutralization_Techniques/citations

Γ. Τεχνικές Πτήσης Και Ανύψωσης

Το τελευταίο κριτήριο βάσει του οποίου μπορεί κανείς να κατηγοριοποιήσει τα drones είναι οι τεχνικές που εφαρμόζονται στις πτήσεις – ανυψώσεις τους. Συγκεκριμένα, δύο είναι τα βασικά είδη: α. σταθερής, β. περιστροφικής πτέρυγας.

Τα αεροσκάφη σταθερής πτέρυγας (Εικόνα 2.1) εύκολα διακρίνονται από τα υπόλοιπα drones, χάρη της χαρακτηριστικής τους όψης. Ειδικότερα, μοιάζουν με τα κοινά αεροπλάνα, καθώς διαθέτουν πτέρυγες σταθερά ενσωματωμένες στον κύριο κορμό του αεροσκάφους, γεγονός που τα ονοματίζει άλλωστε. Αυτού του τύπου η αεροδυναμική κατασκευή τα καθιστά σαφώς πιο ικανά στην ανύψωση και συνακόλουθα στη διατήρηση του επιθυμητού ύψους. Επιπλέον, τα οχήματα σταθερής πτέρυγας δίνουν στον χειριστή την ικανότητα να τα χειρίζεται πιο εύκολα και να πραγματοποιεί ελιγμούς. Σε

- ➥ ότι αφορά τις προγραμματισμένες πτήσεις τους, αυτά είναι ιδανικά για τη μεταφορά βάρους και τη διανομή πακέτων σε δυσπρόσιτες περιοχές.
- ➥ Τα UAVs που διαθέτουν έναν κύριο περιστρεφόμενο έλικα ονομάζονται αεροσκάφη περιστροφικής πτέρυγας (Εικόνα 2.2). Η δομή τους είναι παρόμοια με του παραδοσιακού ελικοπτέρου και το μέγεθός τους σαφώς πολύ μικρότερο. Είναι ιδανικά για πλοήγηση σε χορογραφικά οριοθετημένες περιοχές, όχι πολύ μεγάλες, καθώς ένα από τα βασικά

χαρακτηριστικά του είναι η ικανότητα διατήρησης της ίδιας ακριβώς θέσης πάνω από ένα σημείο. Το επονομαζόμενο hover είναι κατάλληλο όχι μόνο για χειροκίνητο χειρισμό από χειριστή, αλλά και για προγραμματισμένες πτήσεις. Τέτοιου είδους οχήματα συχνά φέρουν εξαρτήματα, όπως κάμερες, και χρησιμοποιούνται για λειτουργίες τύπου αεροφωτογραφιών, βιντεοσκόπησης κ.ό.κ.

Εικόνα 2.1: RQ-170 Sentinel μη επανδρωμένο αεροσκάφος σταθερής πτέρυγας.

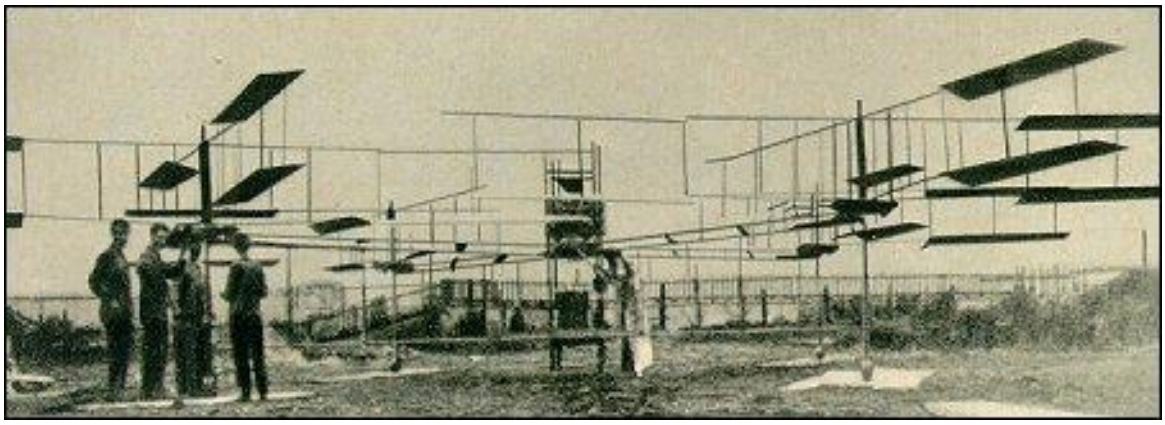
Πηγή: https://www.nicepng.com/ourpic/u2w7q8u2o0i1u2i1_lockheed-martin-rq-170-sentinel/

Εικόνα 2.2: YAMAHA RMAX μη επανδρωμένο αεροσκάφος περιστροφικής πτέρυγας

Πηγή:https://www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax

2.3 Ιστορικά στοιχεία

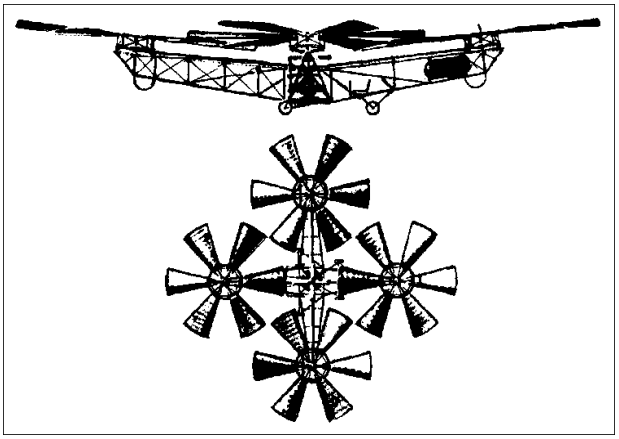
Η πρώτη προσπάθεια κατασκευής εναέριου οχήματος χρονολογείται στη δεκαετία του 1900. Πιο συγκεκριμένα, το 1907 οι αδελφοί Jacques και Louis Brequest από τη Γαλλία, επιχείρησαν τη δημιουργία περιστρεφόμενων ελίκων με το λεγόμενο “Gyroplane No. 1” ανεπιτυχώς, καθώς ο σχεδιασμός του ήταν ασταθής.



Εικόνα 2.3 : Gyroplane No. 1

Πηγή: http://www.aviastar.org/helicopters_eng/breguet_gyro.php

Το 1916, το μη επανδρωμένο αεροσκάφος ονόματι “Ruston Proctor Aerial Target” αποτέλεσε το πρώτο όχημα που επωφελήθηκε της χρήσης των τεχνικών ραδιοελέγχου. Στον απόηχό του, πραγματοποίησε πτήση το “Hewitt – Sperry Automatic Airplane” με απώτερο σκοπό την υλοποίηση της ιδέας μιας εναέριας τορπίλης. Λίγα χρόνια αργότερα, το 1924 οι Dr. George de Bothezat και Ivan Jerome πέτυχαν τον σχεδιασμό και την υλοποίηση του πρώτου τετρακόπτερου. Διέθετε από έξι πτερύγια στο σύνολο τεσσάρων ροτόρων. Το εν λόγω αεροσκάφος είχε επιτυχημένη κάθετη απογείωση και πτήση έπειτα από επανειλημμένες προσπάθειες με μέγιστο ύψος τα 5 μέτρα. Ωστόσο, η προσγείωσή του ήταν προβληματική, γεγονός που οφειλόταν στον μηχανικό σχεδιασμό και την αδύναμη για τα δεδομένα μηχανή.



Εικόνα 2.4 : Dr. George de Bothezat και Ivan Jerome Quadrotor

Πηγή: http://www.aviastar.org/helicopters_eng/bothezat.php

Στις δεκαετίες που ακολούθησαν, ολοένα και περισσότεροι μηχανικοί ασχολήθηκαν με τα αεροσκάφη. Κυρίως στα χρόνια μετά τον Β' Παγκόσμιο Πόλεμο, αξιοσημείωτη ήταν η κατασκευή του Convertawings Model, ενός ελικοπτέρου με τέσσερις έλικες και δύο μηχανές. Ο σχεδιασμός του θεωρήθηκε επιτυχής και ξεκίνησε να χρησιμοποιείται τη δεκαετία του 1950.

Εικόνα 2.5 : Convertawings Mode

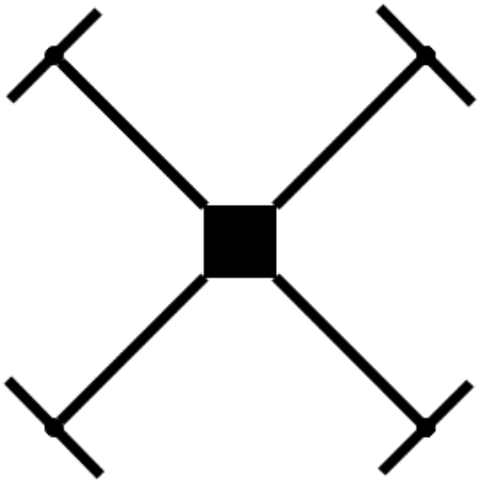
Πηγή: https://www.cradleofaviation.org/history/exhibits/exhibit-galleries/the_jet_age/convertawings_model_a_quadrotor.html

Τα τελευταία χρόνια, τόσο ο επιστημονικός – ακαδημαϊκός, όσο και ο απλός κόσμος μονοπωλείται από το ενδιαφέρον των τετρακοπτέρων και των drones σε ερευνητικό, κατασκευαστικό και ψυχαγωγικό επίπεδο.

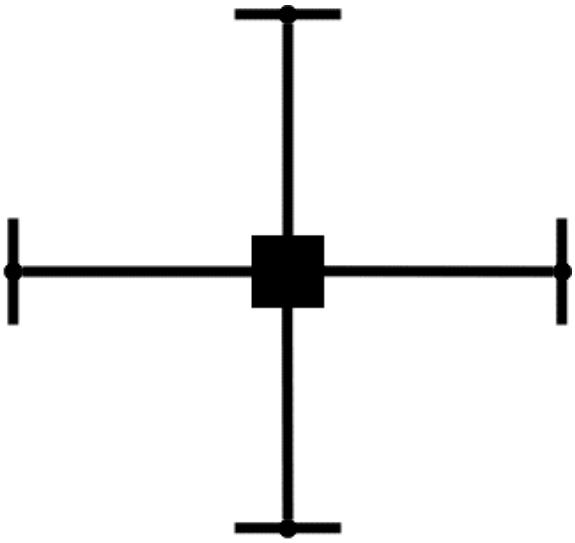
3. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΕΤΡΑΚΟΠΤΕΡΟΥ (QUADCOPTER)

3.1 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ

Οι σχηματισμοί που ακολουθούν τα τετρακόπτερα, είναι κατά βάση δύο. Προκύπτουν από την κατεύθυνση της εκάστοτε πτήσης : σχηματισμός 'x' (Εικόνα 3.1) και σχηματισμός '+' (Εικόνα 3.2). Οι διαφορές φαίνονται στα σχήματα.

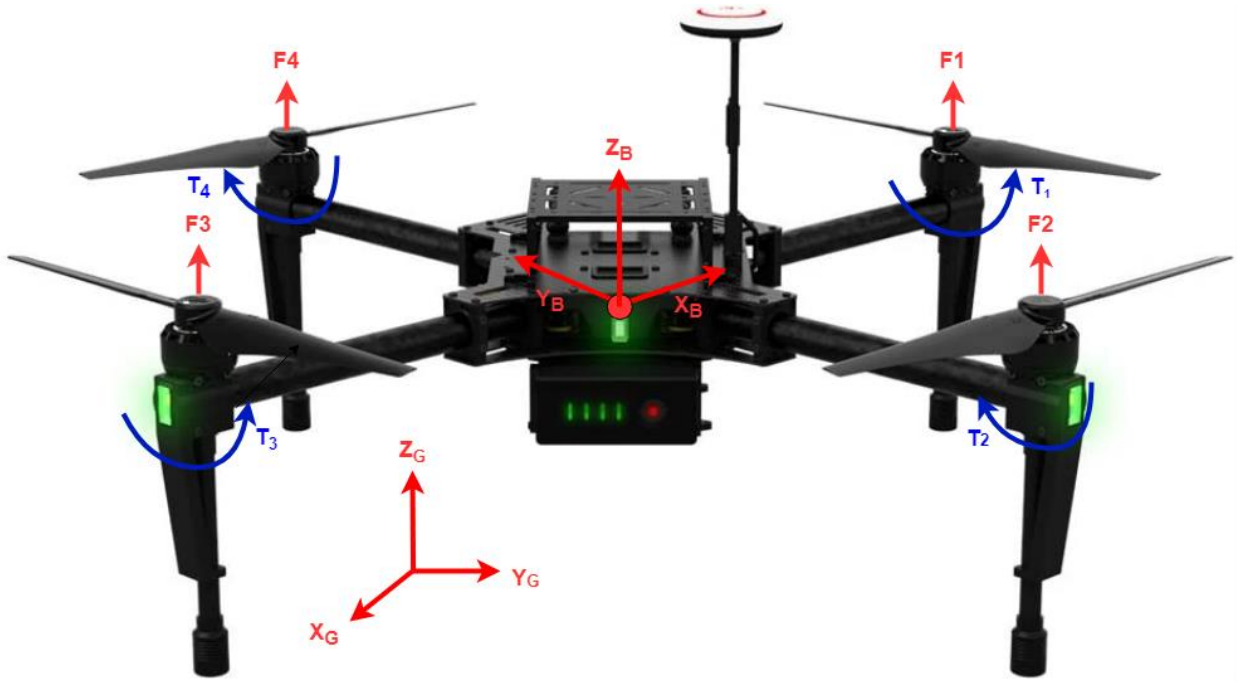


Εικόνα 3.1: Σχηματισμός 'x'



Εικόνα 3.2: Σχηματισμός '+'

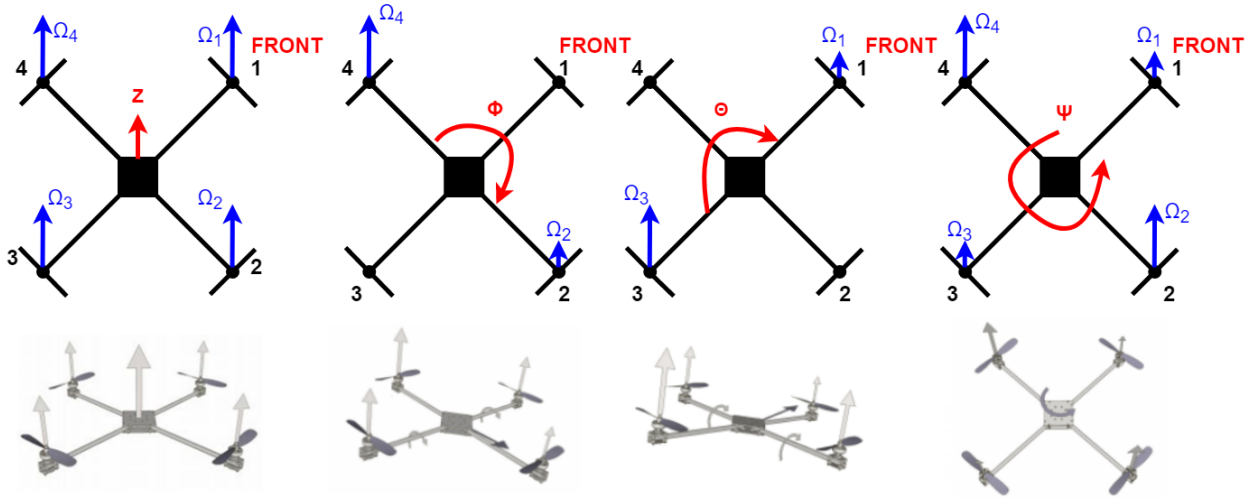
Γενικά, τα τετρακόπτερα είναι ικανά για τρισδιάστατη μεταφορική κίνηση , δηλαδή κίνηση στο χώρο όπου χρησιμοποιούν τους τέσσερις έλικες τους για την παραγωγή ώθησης και τρισδιάστατη περιστροφική κίνηση , δηλαδή περιστροφή στο χώρο. Αυτή η μεταφορική κίνηση γίνεται ως προς κάποιο σταθερό σύστημα αναφοράς $G(X_G,Y_G,Z_G)$ (Inertial frame) (Εικόνα 3.3) που ορίζεται από το έδαφος με το Z_G να είναι αντίθετης κατεύθυνσης από αυτή της βαρύτητας. Ακόμα, ορίζεται το κινούμενο σύστημα αναφοράς $B(X_B,Y_B,Z_B)$ (Body frame) (Εικόνα 3.3) με βάση τον προσανατολισμό του τετρακοπτέρου, έτσι ώστε οι άξονες X_B,Y_B, να συμπίπτουν με το σώμα της κατασκευής .



Εικόνα 3.3: Διάταξη πτήσης

Στην εικόνα 3.3 απεικονίζονται οι δυνάμεις (ώθηση-thrust) και οι ροπές αντίδρασης (drag) που αναπτύσσονται στους έλικες . Οι κινήσεις που μπορεί να πραγματοποιήσει ένα τετρακόπτερο μεταβάλλοντας τις ταχύτητες των ελίκων είναι τέσσερις και συγκεκριμένα μία μεταφορική ως προς τον άξονα Z_B και τρείς περιστροφικές ως προς τους άξονες X_B,Y_B,Z_B. Αναλυτικότερα, αν θέλουμε να πετύχουμε αιώρηση (hover) χρειάζεται οι έλικες να παράγουν αρκετή ώση ώστε να αντισταθμίζεται η βαρύτητα (Εικόνα 3.4.1). Αυξάνοντας , την ταχύτητα τους ομοιόμορφα (ίση αύξηση ταχύτητας για κάθε έλικα) από το συγκεκριμένο σημείο, η ώση αυξάνεται και το τετρακόπτερο κινείται κάθετα σε σχέση με τον προσανατολισμό του προς τα πάνω (άξονας z)(Εικόνα 3.4.1). Έτσι με αυτόν τον τρόπο πετυχαίνουμε την απογείωση και αντιστοίχως μειώνοντας την ταχύτητα με την οποία γυρνάνε οι έλικες, το drone χάνει ύψος και μπορεί να πραγματοποιηθεί η προσγείωση. Από την άλλη μεριά , η περιστροφική κίνηση , έχει να κάνει με τις γωνίες γύρω από τους άξονες X_B,Y_B,Z_B, όπου ονομάζονται roll ,pitch ,yaw αντίστοιχα και πρόκειται για τις γωνίες Euler όπως θα δούμε αναλυτικά και παρακάτω. Εφόσον , η ταχύτητα περιστροφής των τεσσάρων κινητήρων είναι ανεξάρτητη τότε μπορούμε να ελέγξουμε τις γωνίες αυτές. Έτσι αν θέλουμε περιστροφική κίνηση γύρω από τον άξονα X_B αρκεί η αύξηση της γωνιακής ταχύτητας Ω4 και αντίστοιχα μείωση της ταχύτητας της συμπληρωματική της Ω2 ώστε η ροπή να είναι μεγαλύτερη και η γωνιακή κίνηση πιο γρήγορη (roll) (Εικόνα 3.4.2). Αν θέλουμε περιστροφική κίνηση ως προς τον άξονα Y_B, τότε αρκεί η αύξηση των γωνιακής ταχύτητας Ω3 και μείωση της συμπληρωματική της Ω1 (pitch) (Εικόνα 3.4.3) .Τέλος, αυξάνοντας την ταχύτητα των γωνιακών ταχυτήτων Ω4 και Ω2 και

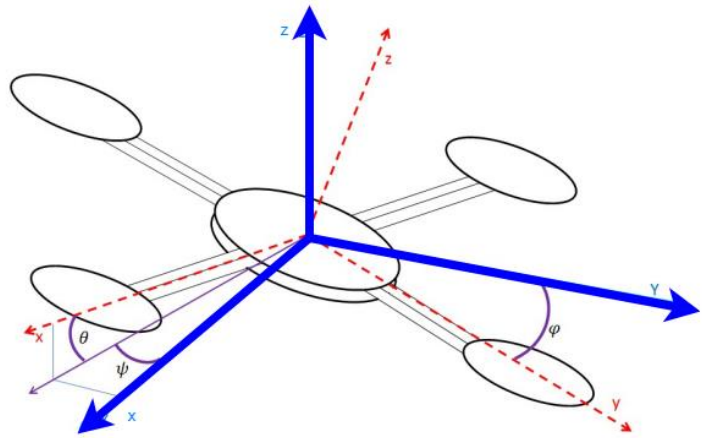
μειώνοντας Ω1 και Ω3 έχουμε περιστροφική κίνηση γύρων από τον άξονα Z_B (yaw) και αντίστοιχα αν αυξήσουμε και μειώσουμε εναλλάξ έχουμε αντίθετη περιστροφή. (Εικόνα 3.4.4).



Εικόνα 3.4.1: Thrust Εικόνα 3.4.2: Roll Εικόνα 3.4.3: Pitch Εικόνα 3.4.4: Yaw

3.2 ΓΩΝΙΕΣ EULER

Γενικά, όταν θέλουμε να περιγράψουμε τον προσανατολισμό της θέσης ενός αντικειμένου στον τρισδιάστατο χώρο (ειδικά για αεροσκάφη) χρησιμοποιούμε τις γωνίες Euler καθώς χρειάζεται να συσχετίσουμε το αδρανειακό σύστημα αναφοράς ''intertial frame'' (X_G,Y_G,Z_G) που ορίζεται από το έδαφος με το Z_G να είναι αντίθετης κατεύθυνσης από αυτό της βαρύτητας με το σύστημα αναφοράς ως προς το ιπτάμενο όχημα 'body frame' για τις τρείς ξεχωριστές περιστροφές στους άξονες. Οι γωνίες Euler αποτελούνται από τις γωνίες ϕ (Roll), θ (Pitch) ,ψ (Yaw). Το Φ αφορά το roll, που είναι η γωνία περιστροφής του quadcopter γύρω από τον άξονα X_B. Το Θ αφορά το pitch, δηλαδή την γωνία περιστροφής του γύρω από τον άξονα Y_B. Τέλος, το Ψ αφορά το yaw, δηλαδή την γωνία περιστροφής γύρω από τον άξονα Z_B.



Εικόνα 3.5: Γωνίες Euler

Αρχικά, έχουμε ορίσει όπως είπαμε τα δύο συστήματα , το αδρανειακό σύστημα G (X_G,Y_G,Z_G) και ένα σύστημα σταθερά δεμένο στο κέντρο μάζας του τετρακόπτερου B (X_B,Y_B,Z_B). Αν θεωρήσουμε το σετ γωνιών Euler XYZ με αντίστοιχες γωνίες ϕ,θ,ψ και πίνακες περιστροφής $R_x(\phi)$, $R_y(\theta)$, $R_z(\psi)$ τότε μελετώντας μονάχα την περιστροφή του συστήματος B η τελική του θέση επιτυγχάνεται με 3 διαδοχικές περιστροφές γύρω από τους αντίστοιχους άξονες :

$$R_x(\varphi)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\Phi) \\ 0 & sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{1.0}$$

$$R_y(\vartheta)=\begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & -sin(\Phi) \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{1.1}$$

$$R_z(\psi)=\begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

Συνδυάζοντας τα παραπάνω για το σετ γωνιών ϕ,θ,ψ αντίστοιχα XYZ προκύπτει ο συνολικός πίνακας περιστροφής R που περιγράφει την περιστροφή από το Body frame στο Inertia Frame.

$$R=R_{XYZ}(\varphi,\vartheta,\psi) = R_z(\psi)R_y(\vartheta)R_x(\varphi) \tag{1.3}$$

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

(1.4)

Ο πίνακας που προέκυψε θα χρησιμοποιηθεί στο επόμενο κεφάλαιο για την κινηματική ανάλυση του τετρακοπτέρου.

3.3 ΔΥΝΑΜΙΚΟ ΜΟΝΤΕΛΟ

3.3.1 ΚΙΝΗΜΑΤΙΚΗ ΚΑΙ ΔΥΝΑΜΙΚΗ ΑΝΑΛΥΣΗ

Στην παρούσα υποενότητα θα αναλυθεί το δυναμικό μοντέλο και οι εξισώσεις κίνησης του τετρακοπτέρου με σκοπό την κατάληξη στις μη γραμμικές εξισώσεις και τη γραμμικοποίηση αυτών στη συνέχεια μέσω Matlab-Simulink. Αποτέλεσμα των παραπάνω θα είναι η δημιουργία του γραμμικού μοντέλου(state space model). Βασική αρχή είναι ο ορισμός του τετρακοπτέρου ως ένα συμμετρικό και αμετάβλητο σώμα, προκειμένου να καταστεί δυνατή η κινηματική ανάλυση για 3D Rigid Body Dynamics. Ως εκ τούτου, θεωρούμαι τις καταστάσεις που περιγράφουν τη μεταφορική θέση του drone $[x\ y\ z]^T$ ως προς το σταθερό σύστημα G (Εικόνα 3.3) θα θεωρηθούν οι $[\varphi\ \vartheta\ \psi]^T$, οι οποίες περιγράφουν την περιστροφική θέση ως προς το ίδιο σταθερό σύστημα. Οι καταστάσεις που περιγράφουν την γραμμική και τη γωνιακή ταχύτητα ως προς το μεταβλητό σύστημα αναφοράς B είναι οι $\boldsymbol{\upsilon}_B = [u\ v\ w]^T \in \mathbb{R}^3$ και $\boldsymbol{\omega}_B = [p\ q\ r]^T \in \mathbb{R}^3$. Με τη χρήση του πίνακα περιστροφής *R* προκύπτει:

$$\upsilon = R\,\upsilon_B$$

(1.5)

Όπου $\boldsymbol{\upsilon} = [\dot{x}\ \dot{y}\ \dot{z}]^T \in \mathbb{R}3$ η ταχύτητα του τετρακοπτέρου εκφρασμένη στο σύστημα G.

Για να εκφράσουμε την γωνιακή ταχύτητα στο πλαίσιο G, μέσω της γωνιακής στο πλαίσιο B ισχύει :

$$\omega = T\,\omega_B$$

(1.6)

με

$$T=\begin{bmatrix} 1 & sin(\varphi)tan(\theta) & sin(\varphi)tan(\theta) \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & \frac{sin(\varphi)}{cos(\theta)} & \frac{cos(\varphi)}{cos(\theta)} \end{bmatrix}$$

(1.7)

Όπου το T προκύπτει από την αρχική σχέση που εκφράζει την γωνιακή ταχύτητα αντιστρέφοντας τον πίνακα Jacobi όπως θα δούμε παρακάτω αναλυτικά. Αρχική, σχέση γωνιακής ταχύτητας:

$$\omega_B = J\,\omega$$

(1.8)

με $\omega=[\ \dot{\varphi}\ \dot{\theta}\ \dot{\psi}\ ] \in \mathbb{R}^3$ η γωνιακή ταχύτητα στο σύστημα G και J ο πίνακας Jacobi που αντιστοιχεί στον πίνακα περιστροφής από το σταθερό πλαίσιο G στο μεταβλητό πλαίσιο B ο οποίος είναι $R^{-1} = R^T = R_X(-\varphi)R_Y(-\theta)R_Z(-\psi)$ και δίνεται:

$$\mathrm{J} = [\hat{x}_1 \quad R_x(-\varphi)\,\hat{y}_1 \quad R_x(-\varphi)\,R_Y(-\theta)\,\hat{z}_1\ ]$$

(1.9)

με $\hat{x}_1 = [1\ 0\ 0]^T$ $\quad \hat{y}_1 = [0\ 1\ 0]^T$ $\quad \hat{z}_1 = [0\ 0\ 1]$ τα μοναδιαία διανύσματα του πλαισίου G. Έτσι αντιστρέφοντας τον πίνακα J και θέτοντας τον με T προκύπτει η σχέση 1.6.

Έτσι με αυτόν τον τρόπο συνδέουμε τις γωνίες Euler με το διάνυσμα της γωνιακής ταχύτητας ω_B στο body frame.

Άρα συνοψίζοντας :

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\varphi)tan(\theta) & sin(\varphi)tan(\theta) \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & \frac{sin(\varphi)}{cos(\theta)} & \frac{cos(\varphi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

(2.0)

Γίνεται αντιληπτό αν παρατηρήσουμε τον πίνακα, ότι για $\theta=\pm 90^{o}$, η παραπάνω σχέση δεν ορίζεται. Υπάρχουν δύο τρόποι για να αντιμετωπίσουμε αυτό το πρόβλημα οι οποιοί είναι οι εξής :

- Με την επιλογή άλλο σετ γωνιών Euler
- Με την χρήση quaternions

Στη συνέχεια , αναλύονται οι διαφορικές εξισώσεις για την μεταφορική και περιστροφική κίνηση. Όσο αφορά την μεταφορική κίνηση εκφρασμένη στο πλαίσιο G (earth frame) ισχύει ο 2ος Νόμος του Νεύτωνα

$$F_G = m\dot{\upsilon}$$

(2.1)

Όπου m η συνολική μάζα του τετρακοπτέρου , F_G η δύναμη που ασκείται στο τετρακόπτερο εκφρασμένη στο πλαίσιο G και $\dot{\upsilon} = [\dot{x}\ \dot{y}\ \dot{z}]^T \in \mathbb{R}3$ η επιτάγχυνση του τετρακοπτέρου εκφρασμένη στο πλαίσιο G. Για την F_G στο μοντέλο μας ισχυεί :

$$F_G = -mge_z + R_B^G f_t^B$$

(2.2)

Όπου, $e_z = [0\ 0\ 1]^T$, f_t στο σύστημα B , είναι η δύναμη που παράγεται κάθετα κατά την ώθηση και R_B^G είναι ο πίνακας που μετασχειματίζει την ώθηση απο το σύστημα B στο G. Έτσι αν αντικαταστήσουμε στην σχέση 3.1 και λύσουμε ως προς $\dot{\upsilon}$ θα προκύψει :

$$\dot{\upsilon} = -\mathrm{g}e_z + R\frac{ft}{m}$$

(2.3)

Η διαφορική εξίσωση η οποία περιγράφει την περιστροφική κίνηση εκφρασμένη στο πλαίσιο G δίνεται απο την εξίσωση του Euler (ροπές που ασκούνται στο τετρακόπτερο) και είναι :

$$I\dot{\omega}_B + \omega_B \times (I\omega_B) = \tau$$

(2.4)

Όπου $\tau = [\ \tau_\varphi\ \tau_\theta\ \tau_\psi\ ]^T$ η συνολική ροπή που ασκείται στο τετρακόπτερο (body frame) και I η μήτρα αδράνειας του τετρακοπτέρου .

Για το I ισχυεί :

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yy} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}$$

(2.5)

Αλλά λόγω του συμμετρικού σχήματος του τετρακοπτέρου ως προς και τους 3 άξονες προκύπτει ότι τα μη διαγώνια στοιχεία του πίνακα I είναι μηδενικά δηλαδή :

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

(2.6)

και άρα

$$I^{-1} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}$$

(2.7)

Απο την 3.2 αν λύσουμε ως προς $\dot{\omega_B}$ τότε προκύπτει η παρακάτω σχέση :

$$\dot{\omega_B} = I^{-1}\big(\tau - \omega_B \times (I\omega_B)\big)$$

(2.8)

Έτσι , παραγωγίζοντας τις σχέσεις 1.5 , 1.6 και κάνοντας παραδοχή μικρών γωνιών $\varphi \cong 0$ και $\theta \cong 0$ προκειμένου να εφαρμοστεί ο βέλτιστος έλεγχος θα προκύψει το μοντέλο του τετρακοπτέρου με τις μεταβλητές κατάστασεις και τις εισόδους :

$$\dot{\upsilon} = -ge_z + Rm^{-1}ft$$

$$\omega = I^{-1}\big(\tau \dot{-} \omega \times (I\omega)\big)$$

(2.9)

3.3.2 ΔΥΝΑΜΕΙΣ-ΡΟΠΕΣ

Για να πραγματοποιηθεί ο έλεγχος ενός τετρακοπτέρου θα πρέπει να ελεγχθούν οι μεταβλητές που έχουμε ως είσοδο στο σύστημα και να συσχετιστούν με την ταχύτητα των ελίκών του τετρακοπτέρου.Ως είσοδο στο σύστημα έχουμε την ώθηση (thrust) , και τις ροπές τ_ϕ, τ_θ, τ_ψ . Έτσι όπως έχουμε ήδη αναφέρει στην ενότητα της περιγραφής λειτουργίας του τετρακοπτέρου, οι ροπές στους άξονες εξαρτώνται από τις γωνιακές ταχύτητες των ελικών όπου ορίζουμε το διάνυσμα τους $\Omega=[\Omega_1\ \Omega_2\ \Omega_3\ \Omega_4]^T$ (εικόνα 3.3). Πιο συγκεκριμένα , η ροπή τ_ϕ γύρω απο τον άξονα x εξαρτάται απο τις γωνιακές ταχύτητες Ω_2,Ω_4 , η ροπή τ_θ γύρω απο τον άξονα y εξαρτάται από τις γωνιακές ταχύτητες Ω_1,Ω_3 ,ενώ η ώση και η ροπή τ_ψ είναι αποτέλεσμα του συνόλου των ταχυτήτων. Σύμφωνα με την αεροδυναμική θεωρεία προκύπτει ότι οι δυνάμεις και οι ροπές εισόδου είναι ανάλογες του τετραγώνου των γωνιακών ταχυτήτων των ελίκων. Αρα :

$$\begin{bmatrix} f_t \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}$$

(3.0)

Δηλαδή μετά απο πράξεις :

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_\varphi = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_\theta = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_\psi = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases}$$

(3.1)

Όπου l η απόσταση από το κέντρο του τετρακοπτέρου μέχρι το κέντρο του κινητήρα, b αεροδυναμική σταθερά ώθησης (thrust) και d αεροδυναμική σταθερά οπισθέλκουσας(drag).

Έτσι γνωρίζοντας , κάθε στιγμή τις δυνάμεις και τις ροπές που θέλουμε να εφαρμοστούν στην είσοδο του συστήματός μας , μπορούμε να ελέγχουμε και να προσαρμόζουμε τις γωνιακές ταχύτητες των ελίκων με τέτοιο τρόπο ώστε να παράγονται οι δυνάμεις αυτές.

3.4 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ

3.4.1 ΜΗ ΓΡΑΜΜΙΚΟ ΜΟΝΤΕΛΟ

Παρακάτω περιγράφεται το μη-γραμμικό μαθηματικό μοντέλο το οποίο προκύπτει απο τις εξισώσεις 2.9

$$
\begin{cases}
\ddot{x} = (\cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi))\dfrac{f_t}{m} \\
\ddot{y} = (\cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\cos(\varphi)\dfrac{f_t}{m} \\
\ddot{z} = \cos(\varphi)\cos(\Theta)\dfrac{f_t}{m} - g \\
\ddot{\varphi} = \dfrac{1}{I_{xx}}\tau_\varphi + \dfrac{I_{yy} - I_{zz}}{I_x}\dot{\theta}\dot{\psi} \\
\ddot{\theta} = \dfrac{1}{I_{yy}}\tau_\theta + \dfrac{I_{zz} - I_{xx}}{I_y}\dot{\varphi}\dot{\psi} \\
\ddot{\psi} = \dfrac{1}{I_{zz}}\tau_\psi + \dfrac{I_{xx} - I_{yy}}{I_z}\dot{\theta}\dot{\varphi}
\end{cases}
$$

(3.2)

Στη συνέχεια οι εξισώσεις εκφράζονται σε μεταβλητές κατάστασης προκειμένου να γραμμικοποιηθούν

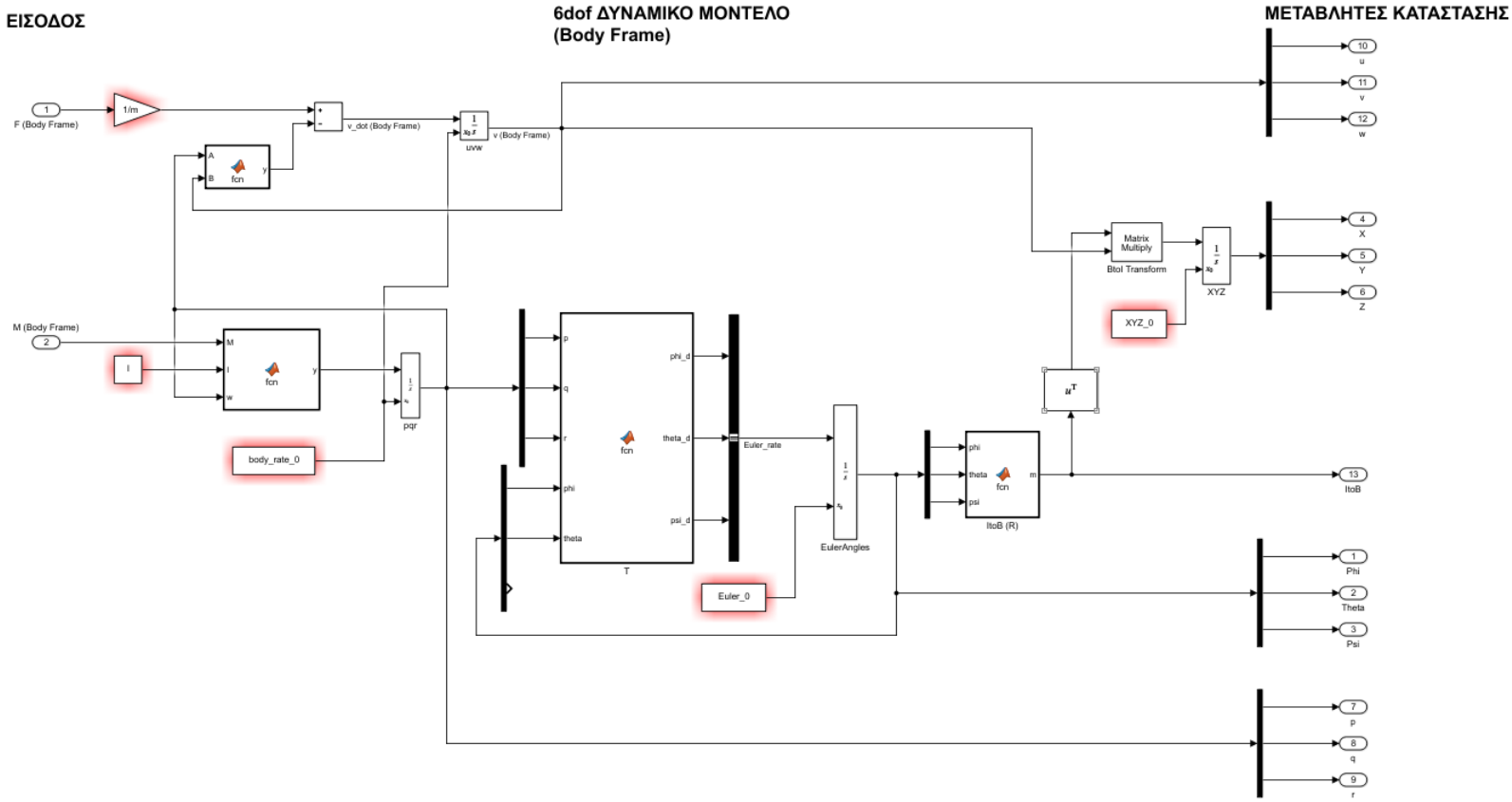
$$
\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} \varphi \\ \theta \\ \psi \\ x \\ y \\ z \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}
$$

(3.3)

Όπου έχουμε φ,ϑ,ψ γωνίες Euler , x,y,z θέση του τετρακοπτέρου στους άξονες , $\dot{\varphi},\dot{\theta},\dot{\psi}$ γωνιακές ταχύτητες και $\dot{x},\dot{y},\dot{z}$ οι γραμμικές ταχύτητες.

3.4.2 Μη γραμμικό μοντέλο-Simulink

Στο υποκεφάλαιο αυτό θα γίνει μία ανάλυση με την βοήθεια του Matlab-Simulink όλου του μη-γραμμικού μαθηματικού μοντέλου προκειμένου να εξηγηθεί στη συνέχεια η διαδικασία της γραμμικοποιήσης του. Έτσι αρχικά, στην εικόνα 3.6 διακρίνουμε τις δυναμικές εξισώσεις (body frame) με την μορφή blocks όπου είναι εύκολο να εντοπίσουμε τις μεταβλητές κατάστασης αλλά και τις εισόδους.

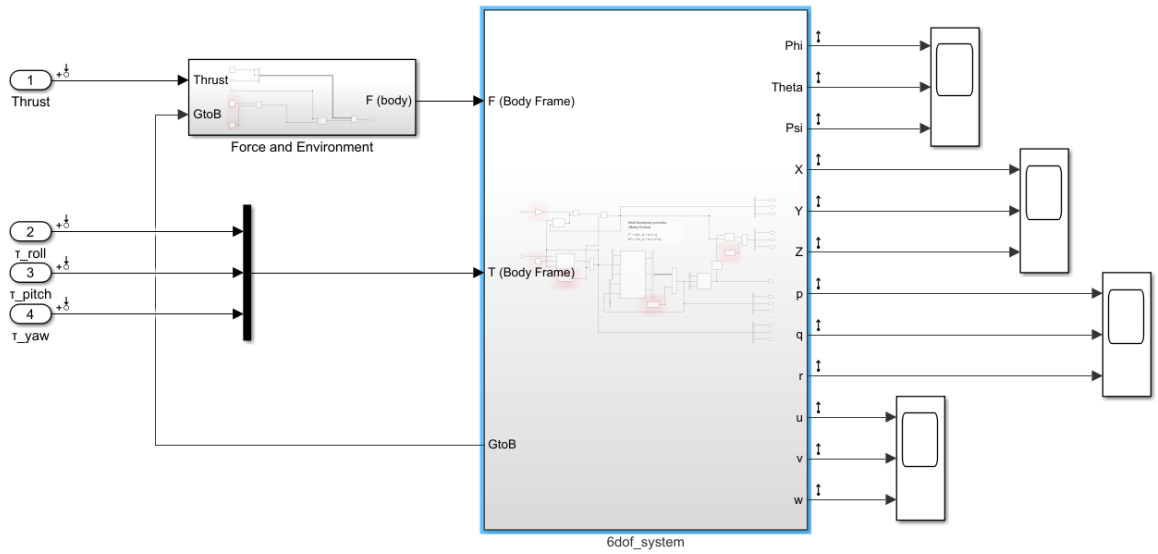


Εικόνα 3.6: Δυναμικές εξισώσεις (Simulink)

Όπως έχει προαναφερθεί , η είσοδος αποτελείται από :

- Ώθηση (thrust)
- Ροπή τ_φ (roll)
- Ροπή τ_θ (pitch)
- Ροπή τ_ψ (yaw)

Και η έξοδος από τις μεταβλητές κατάστασης (3.3) , επομένως διαμορφώνεται στο Simulink το μη-γραμμικό μοντέλο



Εικόνα 3.7 : Μη γραμμικό μοντέλο

Το T (body frame) που δηλώνει ροπή αποτελείται από τις ροπές στους άξονες και περιγράφεται βλέποντας και την εικόνα 3.6 ,από την εξίσωση που έχει ήδη αναφερθεί

$$I\dot{\omega}_B + \omega_B \times (I\omega_B) = \tau$$

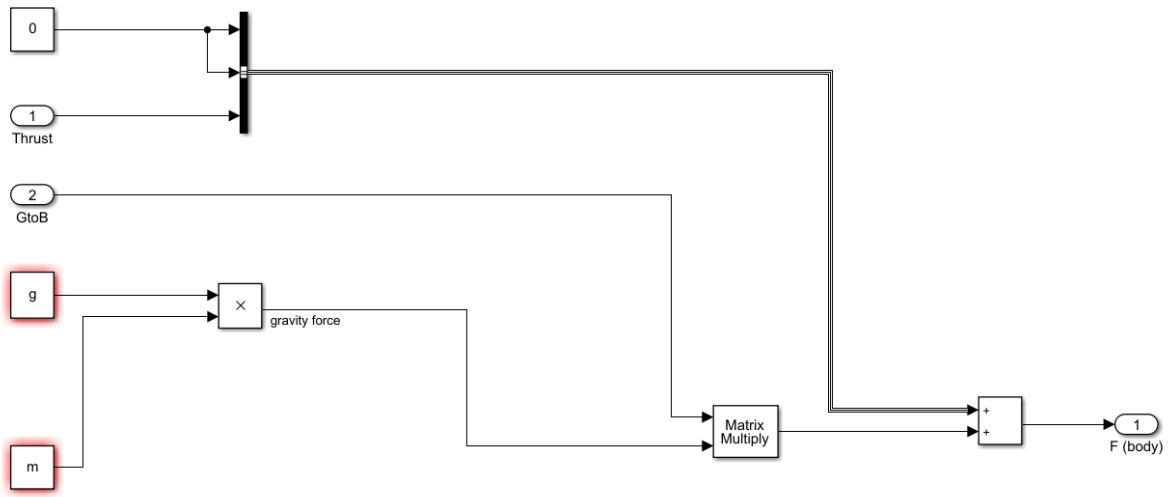
(3.4)

Αντίστοιχα, το block 'Force and Environment' αποτελείται από τη ώση (thrust) ,τη μάζα m , την βαρύτητα g και το πίνακα R_B^G (GtoB). Έτσι η δύναμη F (body frame) υπολογίζεται και είναι βασισμένη στον 2ο νόμο του Νεύτωνα όπως είδαμε αναλυτικά στο προηγούμενο κεφάλαιο (3.1)

$$F_G = m\dot{\upsilon}$$

(3.5)

Παρακάτω φαίνεται αναλυτικά το block ‘Force and Environment’

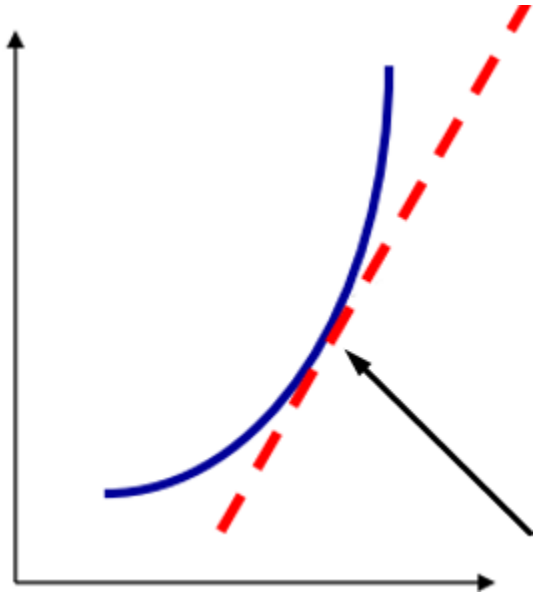


Εικόνα 3.8 : Force and Environment

Έτσι ολοκληρώνοντας , καθώς έχουμε το μη γραμμικό μοντέλο ,προκειμένου να εφαρμοστεί ο έλεγχος θα πρέπει το σύστημα μας να γίνει γραμμικό. Έτσι θα πραγματοποιηθεί γραμμικοποίηση του μοντέλου μέσω Matlab όπως θα δούμε παρακάτω αναλυτικά.

3.4.2 ΓΡΑΜΜΙΚΟΠΟΙΗΣΗ

Γενικά , πολλά είναι τα συστήματα τα οποία είναι μη-γραμμικά για ευρείες μεταβολές των παραμέτρων τους. Σκοπός μας στην παρούσα εργασία είναι να απλοποιήσουμε τις περίπλοκες εξισώσεις και να βρούμε το ισοδύναμο γραμμικό σύστημα κοντά σε ένα σημείο λειτουργίας (Εικόνα 3.9).



Εικόνα 3.9: Γραμμικοποιήση γύρω από το σημείο λειτουργίας

Η γραμμική προσέγγιση είναι συνήθως ,αλλά όχι πάντα αποδεκτή καθώς ο αυτόματος έλεγχος επιδιώκει να διατηρήσει τις ρυθμιζόμενες μεταβλητές κοντά σε κάποιες επιθυμητές τιμές. Απαιτείται η διόρθωση της προσέγγισης αν υπάρξει απομάκρυνση της διεργασίας από το σημείο της γραμμικοποίησης. Μερικοί ακόμα τρόποι γραμμικοποίησης οι οποίοι όμως δεν θα αναλυθούν στην παρούσα εργασία είναι :

1. Γραμμικοποίηση μέσω ανάδρασης-ανατροφοδότηση (feedback linearization)
2. Γραμμικοποίηση Εισόδου-Κατάστασης
3. Γραμμικοποίηση Εισόδου-Εξόδου (βλ. Slotine J. J., 1991, 207-216)

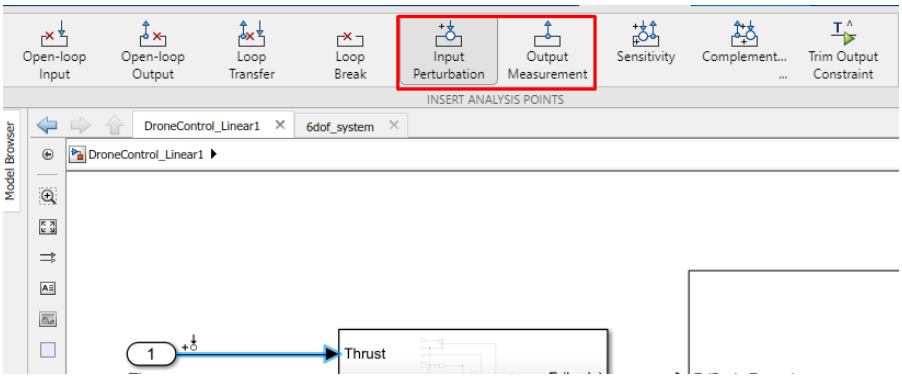
3.4.3 ΓΡΑΜΜΙΚΟΠΟΙΗΣΗ ΤΟΥ ΜΟΝΤΕΛΟΥ ΣΤΟ SIMULINK

Για αρχή ορίζουμε στο Matlab τη μάζα του τετρακοπτέρου, το χρόνο ,την επιτάχυνση της βαρύτητας, τον πίνακα I και τις αρχικές καταστάσεις με τις εξής γραμμές κώδικα :

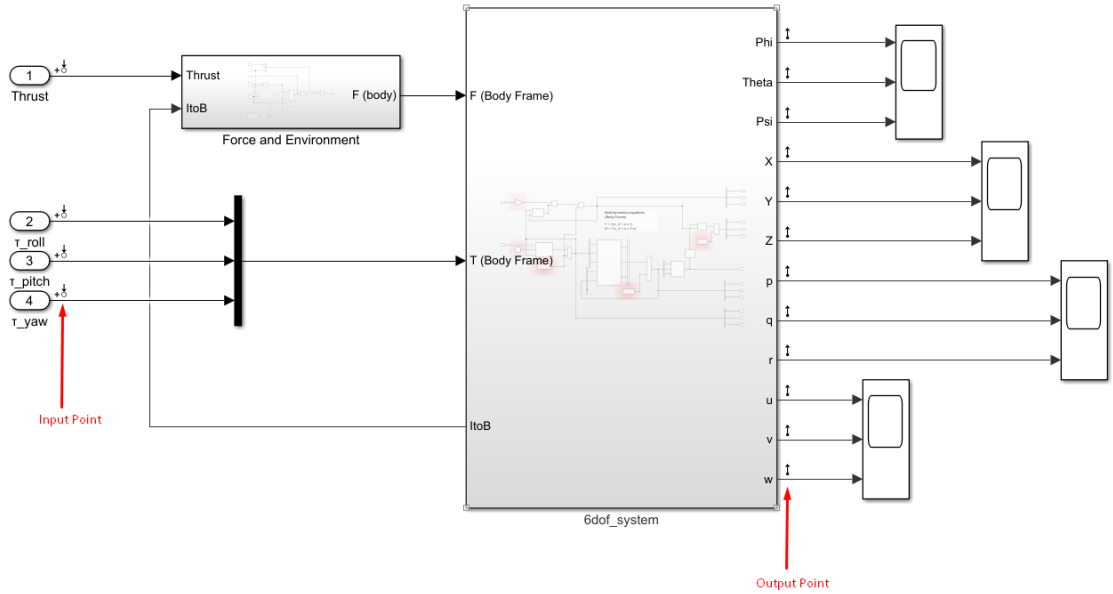
```
% Setup Drone
m = 0.2;
I = [[0.1,0,0];[0,0.1,0];[0,0,0.15]];

% sample time
ts = 0.01;

% Initial States
Euler_0 = [0;0;0];
XYZ_0 = [0;0;0];
body_rate_0 = [0;0;0]; % initial pqr

% Environment (North-East-Down coordinate)
g = [0;0;9.8];
```

Εικόνα 3.10: Κώδικας για αρχικοποιήσεις

οπότε στη συνέχεια καθορίζουμε στο μοντέλο στο Simulink τα input-outputs points που θέλουμε για να πραγματοποιηθεί η γραμμικοποιήση. Επιλέγοντας τα σήματα ελέγχου και πηγαίνοντας στο Linearization tab ρυθμίζουμε ως Input perturbation την επιλεγμένη είσοδο και ως Output Measurement τις μετρήσιμες εξόδους (μεταβλητές κατάστασεις) όπως φαίνεται και στην εικόνα 3.11-3.12.



Εικόνα 3.11: Επιλογή signals



Εικόνα 3.12: Input point – Output point

Στη συνέχεια , για να πάρουμε το γραμμικό μοντέλο θα πρέπει να βρούμε σημεία λειτουργίας. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε την εντολή *findop* του Matlab . Την εντολή αυτή θα έπρεπε να την τοποθετούσαμε σε μία διαδικασία επανάληψης (loop) με αποτέλεσμα να καλούμε συνέχεια την εντολή κάνοντας compile το Simulink model για να βρούμε τα σημεία λειτουργίας, για να το αποφύγουμε αυτό θα καθορίσουμε σημείο λειτουργίας : Z position .

$$([x_3] = [\psi] = 0) \tag{3.6}$$

```
 Operating point for the Model DroneControl_Linear1.
 (Time-Varying Components Evaluated at time t=0)

States:
----------
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0
      x: 0

Inputs:
----------
      u: -1.96
      u: 0
      u: 0
      u: 0
```

Εικόνα 3.13: Σημείο λειτουργίας

Στη συνέχεια με την εντολή *getlinio* παίρνουμε από το Simulink model τα Linear analysis points (input-output) έτσι ώστε να καθορίσουμε στην συνέχεια το τμήμα του μοντέλου που θα γραμμικοποιηθεί δημιουργώντας έναν πίνακα στο workspace με την εντολή *linio*. Τέλος , πραγματοποιείται η γραμμικοποιήση στο σημείο λειτουργίας με την εντολή *linearize* του Matlab . Παρακάτω φαίνεται ο κώδικας :

```
%  Get Linear Model from Simulink model.
 model = 'DroneControl_Linear1';

 opspec = operspec(model);

% Specify operating states: Z position
 opspec = addoutputspec(opspec,'DroneControl_Linear1/6dof_system',3);
 opspec.Outputs(1).Known = true;
 opspec.Outputs(1).y = 0;

% Find Operating point
op1 = findop(model,opspec);

% Get IO from simulink
 io = getlinio(model);
```

Εικόνα 3.14: Κώδικας για γραμμικοποίηση

```
% Set Inputs
 io(1) = linio('DroneControl_Linear1/Thrust',1,'input');
 io(2) = linio('DroneControl_Linear1/M_roll',1,'input');
 io(3) = linio('DroneControl_Linear1/M_pitch',1,'input');
 io(4) = linio('DroneControl_Linear1/M_yaw',1,'input');

 io(5) = linio('DroneControl_Linear1/6dof_system',1,'output')
 io(6) = linio('DroneControl_Linear1/6dof_system',2,'output');
 io(7) = linio('DroneControl_Linear1/6dof_system',3,'output');
 io(8) = linio('DroneControl_Linear1/6dof_system',4,'output');
 io(9) = linio('DroneControl_Linear1/6dof_system',5,'output');
 io(10) = linio('DroneControl_Linear1/6dof_system',6,'output');
 io(11) = linio('DroneControl_Linear1/6dof_system',7,'output');
 io(12) = linio('DroneControl_Linear1/6dof_system',8,'output');
 io(13) = linio('DroneControl_Linear1/6dof_system',9,'output');
 io(14) = linio('DroneControl_Linear1/6dof_system',10,'output');
 io(15) = linio('DroneControl_Linear1/6dof_system',11,'output');
 io(16) = linio('DroneControl_Linear1/6dof_system',12,'output');

% Linearize
sys = linearize(model,opl,io);
```

Εικόνα 3.15: Κώδικας για γραμμικοποίηση

Το αποτέλεσμα της γραμμικοποίησης φαίνεται παρακάτω (state space-γραμμικό μοντέλο):

A	12x12 double
B	12x4 double
C	12x12 double
D	12x4 double
E	[ ]
Scaled	0
StateName	12x1 cell
StateUnit	12x1 cell
InternalDelay	[ ]
InputDelay	[0;0;0;0]
OutputDelay	12x1 double
Ts	0
TimeUnit	'seconds'
InputName	4x1 cell
InputUnit	4x1 cell
InputGroup	1x1 struct
OutputName	12x1 cell
OutputUnit	12x1 cell
OutputGroup	1x1 struct
Notes	0x1 string
UserData	[ ]
Name	''
SamplingGrid	1x1 struct

Εικόνα 3.16: State space

3.4.4 ΓΡΑΜΜΙΚΟ ΜΟΝΤΕΛΟ

Το σύστημα μας , αναλύεται με την μεθοδολογία του χώρου κατάστασης (state-space) όπου το σύστημα περιγράφεται από ένα σύνολο εξισώσεων διαφορών πρώτης τάξης , που περιγράφουν τις μεταβλητές κατάστασης. Ορισμένα πλεονεκτήματα περιγραφής ενός συστήματος με την μέθοδο των εξισώσεων κατάστασης είναι :

1. Μπορούμε εύκολα να προγραμματίσουμε και να προσομοιώσουμε το σύστημα σε ηλεκτρονικό υπολογιστή αφού μιλάμε για γραμμικό σύστημα εξισώσεων.
2. Δίνεται η δυνατότητα περιγραφής μη γραμμικών, και χρονικά μεταβαλλόμενων συστημάτων, κάτι το οποίο δεν μπορεί να γίνει με χρήση της συνάρτησης μεταφοράς.
3. Γίνεται ευκολότερη η λύση προβλημάτων ελέγχου όπως η ευστάθεια και ο βέλτιστος έλεγχος που μας ενδιαφέρει.
4. Δίνεται η δυνατότητα περιγραφής της κατάστασης του συνολικού συστήματος κάθε χρονική στιγμή, σε αντίθεση με τη συνάρτηση μεταφοράς.

Επομένως , μετά την γραμμικοποιήση που πραγματοποιήθηκε, το σύστημα ελέγχου παρουσιάζεται στην εικόνα 3.17:

Εικόνα 3.17: Σύστημα ελέγχου συναρτήσει των x και u.

Και εκφράζεται στο χώρο κατάστασης (state-space) ως εξής :

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}$$
$$y = Cx + Du$$

(3.7)

Με $A \in \mathbb{R}^{12\times12}$ και $B \in \mathbb{R}^{12\times4}$ και $C \in \mathbb{R}^{12\times12}$ και $D \in \mathbb{R}^{12\times4}$.

Όπου αναλυτικότερα :

- x: Μεταβλητές κατάστασης
- y: Έξοδοι
- u: Έισοδοι
- A: Πίνακας του συστήματος στον χώρο κατάστασης
- B: Πίνακας εισόδου
- C: Πίνακας εξόδου
- D: Πίνακας που συνδέει την είσοδο με την έξοδο παρακάμπτωντας τις μεταβλητές κατάστασης.

Παρακάτω φαίνονται τα αποτελέσματα των πινάκων μετά την γραμμικοποιήση (Εικόνα 3.16) :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(3.8)

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \end{bmatrix} \tag{3.9}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.0}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

4.ΕΛΕΓΧΟΣ ΤΕΤΡΑΚΟΠΤΕΡΟΥ

4.1 ΒΕΛΤΙΣΤΟΣ ΕΛΕΓΧΟΣ

Ένα από τα πιο αξιοσημείωτα πεδία δραστηριότητας των μηχανικών ελέγχου αποτελεί η σχεδίαση βέλτιστων συστημάτων ελέγχου. Η πρώτη εισαγωγή του βέλτιστου ελέγχου εμφανίζεται από το 1950 σε ΗΠΑ και Πρώην Σοβιετική Ένωση σε μία εποχή που είχε ανάγκη για έρευνα τόσο σε στρατιωτικές όσο και σε διαστημικές εφαρμογές. Όπως δηλώνεται και από το όνομά του, ο έλεγχος επιδιώκει την καλύτερη δυνατή απόδοση του εκάστοτε συστήματος. Ως στόχο έχει να προσδιορίσει τα σήματα εισόδου τα οποία θα εξαναγκάσουν μία διαδικασία να ικανοποιήσει φυσικούς περιορισμούς και παράλληλα να ελαχιστοποιήσει (μεγιστοποιήσει) κάποια κριτήρια απόδοσης. Σε αντίθεση με τον κλασικό έλεγχο, η επιθυμητή συμπεριφορά είναι η ελαχιστοποίηση του σφάλματος παντού και όχι μόνο σε συγκεκριμένα χρονικά σημεία. Πιο συγκεκριμένα, η συμπεριφορά του εκάστοτε συστήματος είναι δυνατό να αποτυπωθεί με τα ολοκληρώματα των μέτρων απόδοσης. Ένα από τα πλεονεκτήματα του εν λόγω ελέγχου είναι η δυνατότητα εφαρμογής σε συστήματα με πολλές εισόδους και εξόδους (MIMO).Χαρακτηριστικό παράδειγμα είναι το ολοκλήρωμα του τετραγωνικού σφάλματος(ISE) το οποίο αποτελεί το λεγόμενο γραμμικό τετραγωνικό πρόβλημα (Linear Quadratic-LQ). Με συνάρτηση κόστους που προκύπτει από τις συναρτήσεις Lyapunov

$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)\, dt \tag{4.2}$$

Με $Q \geq 0$, $R > 0$ είναι συμμετρικοί, θετικά ορισμένοι πίνακες κατάλληλων διαστάσεων. Τα συστήματα που ρυθμίζονται έτσι ώστε να ελαχιστοποιείται κάποιος δείκτης απόδοσης συχνά ονομάζονται βέλτιστα συστήματα ελέγχου (optimal control systems).

4.2 ΓΡΑΜΜΙΚΟΣ ΤΕΤΡΑΓΩΝΙΚΟΣ ΡΥΘΜΙΣΤΗΣ (LQR)

Έχοντας ερμηνεύσει το μοντέλο μας στον χώρο κατάστασης (state space) , ορίζουμε το γενικό δείκτη απόδοσης άπειρου χρονικού ορίζοντα:

$$J = \int_0^\infty (x^T Q x(t) + u^T R u(t) dt$$

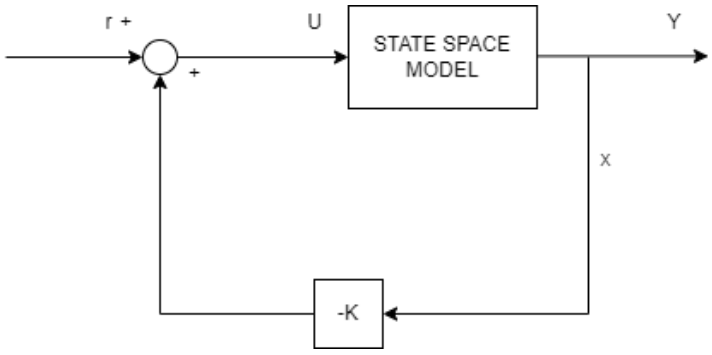
(4.3)

Οι πίνακες Q,R αποτελούν συντελεστές στάθμισης-βαρύτητας οι οποίοι είναι επιλογή του μηχανικού για να πετύχει το βέλτιστο αποτέλεσμα και τις ανάγκες του συστήματος. Οι πίνακες έχουν συνήθως διαγώνια μορφή όπου κλασσική επιλογή αποτελεί: $Q = C^T C$, $R = \lambda I$, $\lambda > 0$. Συγκεκριμένα, ο πίνακας Q που είναι θετικά ημι-ορισμένος αφορά το βάρος του διανύσματος κατάστασης (state cost) και ο πίνακας R που είναι θετικά ορισμένος αφορά το βάρος της ενέργειας που δαπανά το σύστημα για τον έλεγχο του τετρακοπτέρου(control cost). Το J είναι στην ουσία μια μορφή συμβιβασμού μεταξύ της σπουδαιότητας του διανύσματος κατάστασης και της ενέργειας. Έτσι , ο βέλτιστος ελεγκτής που ελαχιστοποιεί το J προκύπτει με ανατροφοδότηση κατάστασης και δίνεται από τη σχέση

$$u(t) = -Kx(t)$$

(4.4)

Η διαφορά από την τοποθέτηση πόλων είναι ότι σε αυτή την περίπτωση βρίσκουμε το βέλτιστο K επιλέγοντας τις βέλτιστες καταστάσεις για το σύστημά μας.



Εικόνα 4.0: Ανατροφοδότηση κατάστασης

Όπου το κέρδος του ελεγκτή δίνεται από

$$K = R^{-1}B^{T}P$$

(4.5)

ο πίνακας P είναι συμμετρική θετικά ημιορισμένη λύση της αλγεβρικής εξίσωσης Riccati

$$A^{T}P + PA - PBR^{-1}B^{T}P + Q = 0$$

(4.6)

η οποία μπορεί να λυθεί εύκολα μέσω Matlab.

Έτσι αν αντικαταστήσουμε το K στην εξίσωση $\dot{x} = Ax + Bu$ το κλειστό σύστημα θα διαμορφωθεί

$$\dot{x} = (A - BK)x$$

(4.7)

Στο Matlab, η εύρεση του πίνακα K γίνεται πολύ εύκολα μέσω της συνάρτησης

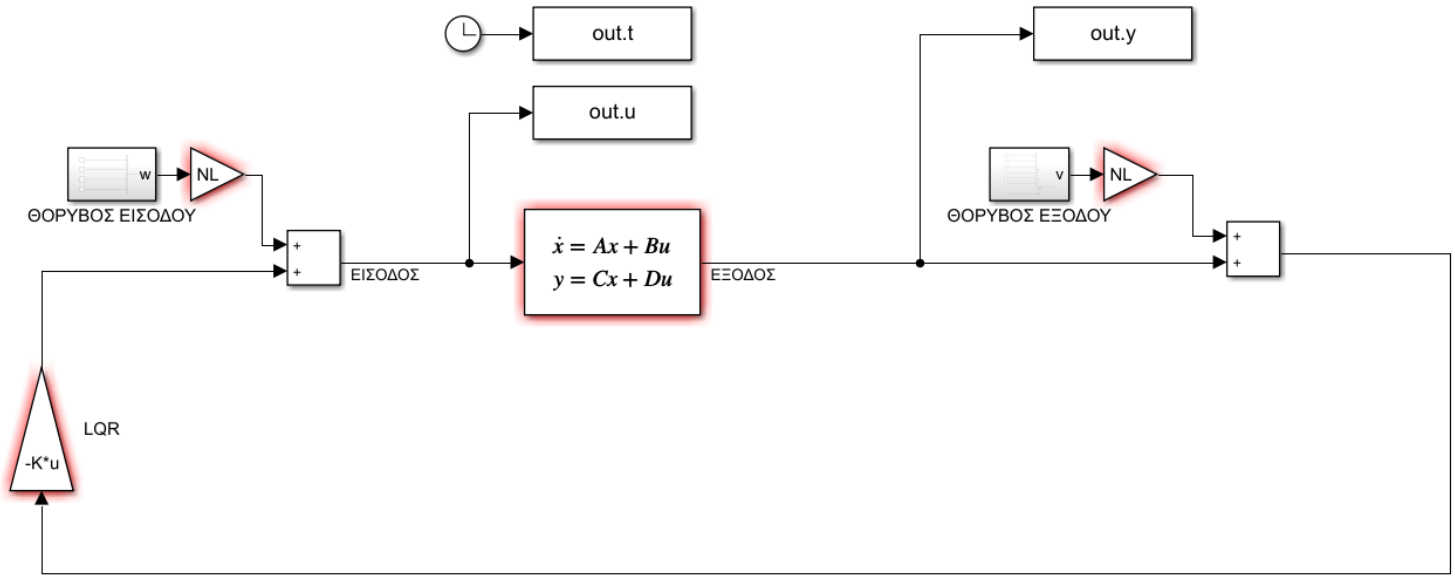
$$K = lqr(A, B, Q, R)$$

(4.8)

Η συγκεκριμένη εξίσωση λύνει έμμεσα την αλγευρική εξίσωση Riccati , αφού επιστρέφει το K και καθιστά τον έλεγχο ανατροφοδότησης κατάστασης (Εικόνα 4.0).

4.2.1 ΠΡΟΣΟΜΟΙΩΣΗ LQR ΣΤΟ SIMULINK

Παρακάτω, παρουσιάζεται το διάγραμμα που δημιουργήθηκε στο Matlab-Simulink του γραμμικού τετραγωνικού ρυθμιστή (LQR) όπου έγιναν προσομοιώσεις και δοκιμές όπως θα δούμε στο κεφάλαιο 5.



Εικόνα 4.1 : Διάγραμμα Simulink-Matlab

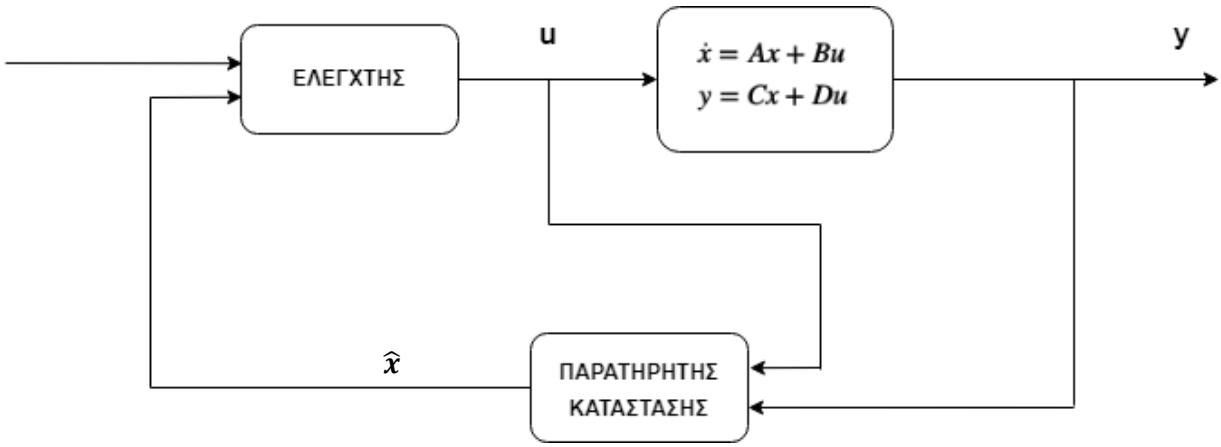
Εδώ , ο θόρυβος εισόδου και εξόδου δεν παίζει κανέναν ρόλο αφού το NL , το ορίζουμε στον κώδικα μας ως μηδέν επομένως δεν μπαίνει στην συνθήκη που αφορά τον θόρυβο. Με άλλα λόγια είναι σαν να μην υπάρχει στην συγκεκριμένη περίπτωση απλά τροποποιήθηκε για ευκολία .

4.3 ΠΑΡΑΤΗΡΗΤΗΣ ΚΑΤΑΣΤΑΣΗΣ-ΑΡΧΗ ΔΙΑΧΩΡΙΣΜΟΥ

Ένα από τα σημαντικότερα προβλήματα κατά την σχεδίαση του LQR ελεγκτή είναι ότι οι μεταβλητές κατάστασης πρέπει να είναι εφικτό να μετρηθούν όλες. Αυτό, προφανώς και είναι σχεδόν αδύνατο σε πραγματικές συνθήκες, επομένως μία ρεαλιστική θεώρηση της εξίσωσης εξόδου είναι η ,

$$\mathrm{y}(t) = Cx(t), C \neq I \tag{4.9}$$

Οπότε στην περίπτωση αυτή έχουμε την αρχή του διαχωρισμού όπου αν το διάνυσμα κατάστασης δεν είναι μετρήσιμο στο σύνολο του , τότε ως είσοδος στον μηχανισμό ελέγχου μπορεί να χρησιμοποιηθεί μία εκτίμηση $\hat{x}$ της κατάστασης, διαχωρίζοντας έτσι το συνολικό πρόβλημα σε δύο : του ελέγχου και της εκτίμησης (παρατηρητής κατάστασης).



Εικόνα 4.2 : Παρατηρητής κατάστασης

Ένα πρόβλημα το οποίο προκύπτει είναι αν η εκτίμηση $\hat{x}$ θα συγκλίνει προς την πραγματική κατάσταση. Μία λύση στο πρόβλημα είναι ο παρατηρητής να χρησιμοποιεί τους γνωστούς πίνακες A,B για να μιμηθεί το ελεγχόμενο σύστημα.

4.4 ΦΙΛΤΡΟ KALMAN

Όπως αναφέρθηκε και παραπάνω, πολλές φορές στην πράξη δεν είναι μετρήσιμες όλες οι μεταβλητές κατάστασης για διάφορους λόγους, όπως περιβαλλοντικές συνθήκες, μη αξιόπιστα όργανα μέτρησης και άλλα, τα οποία χαρακτηρίζονται ως διαταραχές που ονομάζουμε «θόρυβο».

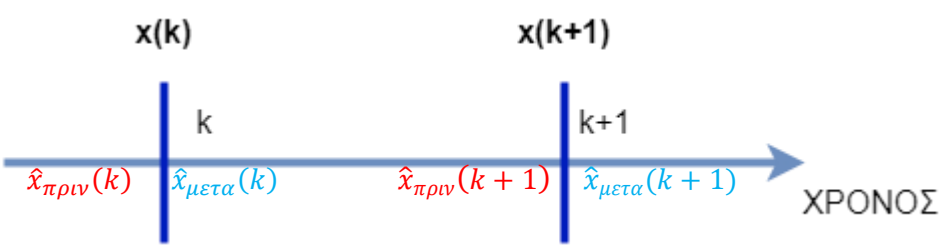
Ως φίλτρο Kalman ορίζεται ο βέλτιστος παρατηρητής δηλαδή μία σχεδιασμένη μέθοδος πρόβλεψης. Αυτό το φίλτρο χρησιμοποιείται για απαλλαγή από αυτούς της μη επιθυμητούς θορύβους. Σε θεωρητικό επίπεδο είναι ένας αλγόριθμος που ονομάζεται φίλτρο, αφού έχει ως στόχο να φιλτράρει το θόρυβο από το σύστημα και να εξομαλύνει τα εκάστοτε αποτελέσματα.

Το φίλτρο χρησιμοποιεί τα χαρακτηριστικά του θορύβου δηλαδή τη μέση τιμή και τη διακύμανση, αποβάλλοντας τον στη συνέχεια από το σύστημα. Για κάθε χρονική στιγμή k , η πραγματική τιμή της μεταβλητής είναι x(k). Η μετρούμενη τιμή του αισθητήριου οργάνου είναι $x_{\pi\rho\iota\nu}$(k), η οποία λόγω θορύβου δεν ταυτίζεται με την πραγματική τιμή, άρα προκύπτει σφάλμα

$$e_{\pi\rho\iota\nu}(k) = \left|x(k) - x_{\pi\rho\iota\nu}(k)\right| \tag{5.0}$$

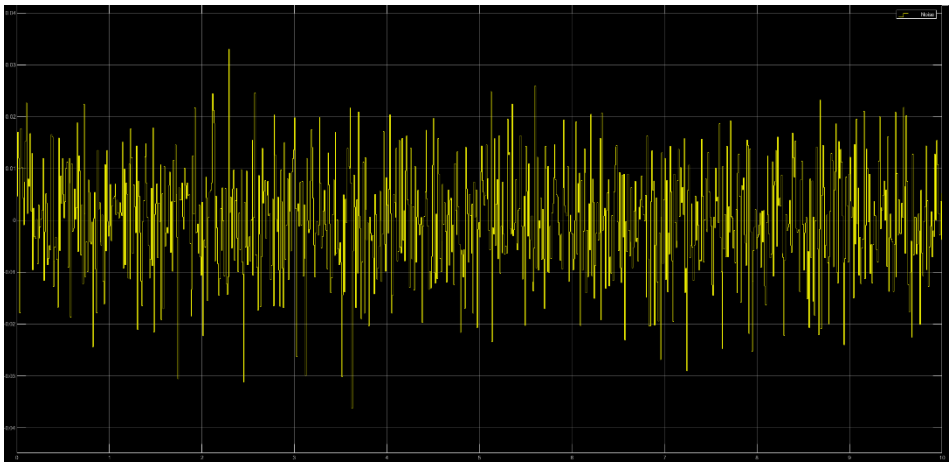
Χρησιμοποιώντας το φίλτρο, προκύπτει μια βελτιωμένη εκτίμηση $x_{\mu\varepsilon\tau\alpha}\,(k)$, πιο κοντά στην πραγματική τιμή, δηλαδή

$$e_{\mu\varepsilon\tau\acute{\alpha}}(k) = \left|x(k) - x_{\mu\varepsilon\tau\acute{\alpha}}(k)\right| < e_{\pi\rho\iota\nu}(k) \tag{5.1}$$



Εικόνα 4.3 : Εκτιμήσεις σε χρόνο k,k+1

Παρακάτω αναλύεται η έννοια θόρυβος ώστε να γίνει πιο ξεκάθαρο το πρόβλημα. Αποτελεί μία στοχαστική διαδικασία που κατά κανόνα ακολουθεί την κανονική κατανομή (Gaussian). Στην παρούσα περίπτωση χρησιμοποιούμε λευκό θόρυβο, το πιο χαρακτηριστικό είδος θορύβου στις μηχανικές εφαρμογές. Όσον αφορά στον θόρυβο, αξίζει να σημειωθεί πως για να εφαρμοστεί το φίλτρο απαιτείται η μέση τιμή η οποία πρέπει να ισούται με το μηδέν και η διακύμανση να είναι πεπερασμένη. Στην εικόνα 4.4 παρουσιάζεται ένα παράδειγμα σήματος λευκού θορύβου.



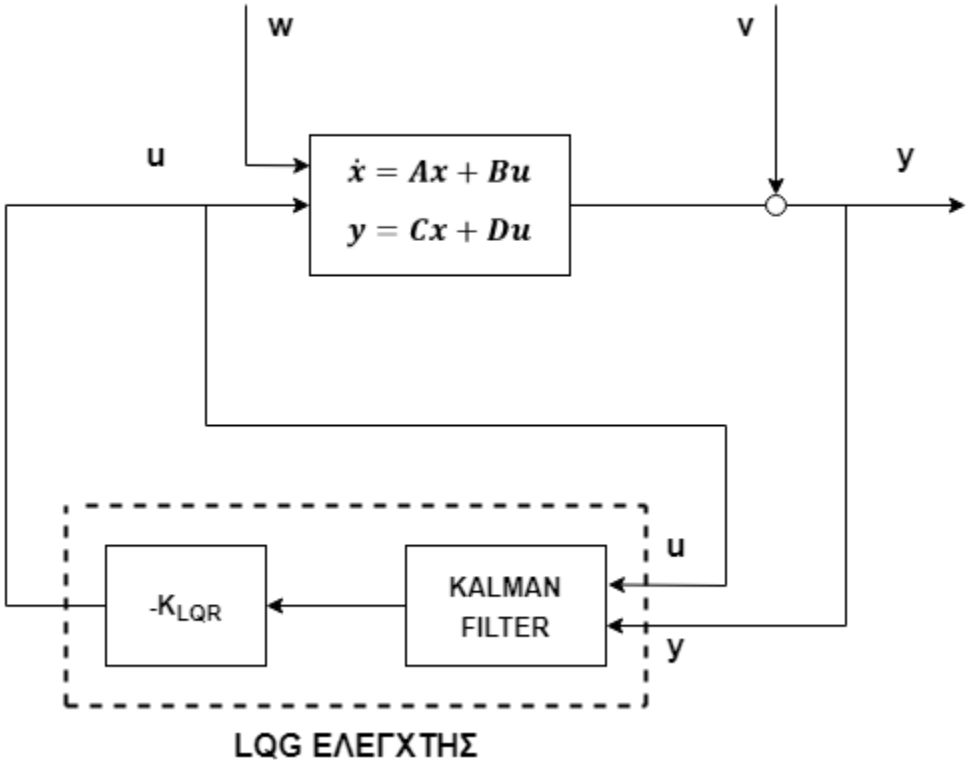
Εικόνα 4.4 : Σήμα Λευκού θορύβου

Σε βάθος χρόνου , μετά από έρευνες διαπιστώθηκε ότι η πλειοψηφία των θορύβων ακολουθούσαν Gaussian κατανομή με τα παραπάνω χαρακτηριστικά . Όλα όσα αναφέρθηκαν συντελούν στο συμπέρασμα πως ο λευκός θόρυβος είναι μία βέλτιστη μοντελοποίηση των ρεαλιστικών διαταραχών.

4.5 ΓΡΑΜΜΙΚΟΣ ΤΕΤΡΑΓΩΝΙΚΟΣ ΓΚΑΟΥΣΙΑΝΟΣ ΕΛΕΓΧΟΣ (LQG)

Ο νόμος ελέγχου που συνδυάζει το γραμμικό τετραγωνικό ρυθμιστή LQR (optimal linear quadratic regulator) με το φίλτρο Kalman ονομάζεται συνήθως γραμμικός τετραγωνικός γκαουσιανός έλεγχος (LQG) . Σε περιπτώσεις όπου η σχεδίαση βασίζεται σε ένα μαθηματικό μοντέλο στο χώρο κατάστασης , ο LQG αποτελεί ένα απλό και βασικό εργαλείο στο μηχανικό έλεγχο. Ειδικότερα, το γραμμικό σύστημα εξισώσεων περιγραφής των προβλημάτων γραμμικού τετραγωνικού γκαουσιανού ελέγχου έχει επίσης την ιδιότητα της παρουσίας του λευκού γκαουσιανού θορύβου. Συνακόλουθα , ο εν λόγω ελεγκτής είναι ένα αξιόπιστο μέσο σε περιπτώσεις θορύβου. Στόχος του προβλήματος είναι η ελαχιστοποίηση του τετραγωνικού κόστους μέσω ενός βέλτιστου νόμου ανάδρασης της εξόδου. Όσον αφορά στη σχεδίαση, αυτή παραμένει απλή για οποιοδήποτε MIMO σύστημα όπως και το δικό μας.

Παρακάτω παρουσιάζεται το μπλοκ διάγραμμα του γραμμικού τετραγωνικού γκαουσιανού ελέγχου



Εικόνα 4.5 : LQG Ελεγκτής

Όπου έχουμε το γραμμικό σύστημα εξισώσεων

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$

$$y(t) = Cx(t) + Du(t) + v(t)$$

(5.2)

Όπου

- x: Οι μεταβλητές κατάστασης
- u: Το διάνυσμα εισόδων
- y: Το διάνυσμα εξόδων
- w: Γκαουσιανός θόρυβος εισόδου
- v: Γκαουσιανός θόρυβος εξόδου

Η συνάρτηση κόστους είναι η ακόλουθη

$$J = E[x^T(t)Fx(t) + \int_0^T [x^T(t)Q_n(t)x(t) + u^T(t)R_n(t)u(t)]dt]$$

(5.4)

Τα σήματα w(t) και v(t) αποτελούν στοχαστικές διαδικασίες με κανονική (Gauss) κατανομή, μηδενική μέση τιμή και συμμεταβλητότητα (covariance) .

Άρα ισχύει

$$E[w(t)] = 0$$

Επίσης ισχυεί (5.5)

$$E[w(i)w^T(j)] = 0, i \neq j$$

(5.6)

Δηλαδή είναι ασυσχέτιστα, και ο πίνακας συνδιασποράς είναι

$$E[w(k)w^T(k)] = Q_N$$

(5.7)

Στην εξίσωση 6.0 ο πίνακας Q_n δηλώνει το πλάτος των διαταραχών του συστήματος και ο πίνακας R_n δηλώνει το πλάτος του θορύβου των μετρήσεων. Θα είναι της μορφής

$$Q_N = BB^T$$

(5.8)

Και

$$R_N = \lambda I$$

(5.9)

Έτσι, ο βέλτιστος νόμος ελέγχου: $u(k) = -Kx(k)$ αντικαταστείτε στην περίπτωση αυτή με τον ακόλουθο : $u(k) = -K\hat{x}(k)$. Το $\hat{x}(k)$ δηλώνει μια εκτίμηση της κατάστασης και ισχύει

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B(t)u(t) + L(t)[y(t) - C(t)\hat{x}(t)]$$

(6.0)

με $\hat{x}(0) = E[x(0)]$.

Ο πίνακας L αποτελεί το κέρδος Kalman (LQE) το οποίο δίνεται από τη σχέση

$$L(t) = P(t)C^T(t)R_N^{-1}(t)$$

(6.1)

και μπορούμε να το διακρίνουμε στην σχεδίαση στο Simulink στην εικόνα 4.6.

Εδώ αξίζει να σημειωθεί πως αν έχουμε σχετικά μεγάλες τιμές για το w τότε συνεπάγεται ότι θα έχουμε μεγάλα κέρδη L , δίνοντας έτσι μεγαλύτερη βαρύτητα στην έξοδο y. Αντίθετα, αν έχουμε μεγαλύτερες τιμές για το v τότε θα έχουμε μικρά κέρδη L , άρα έχουμε ως αποτέλεσμα να έχουμε μεγαλύτερη βαρύτητα στην είσοδο u. Το συμπέρασμα που παίρνουμε είναι πως μεγάλες τιμές του v σημαίνει μεγαλύτερη εμπιστοσύνη στο μοντέλο (ακριβές μοντέλο) με αποτέλεσμα πιο αργή δυναμική συμπεριφορά (ανοικτού βρόχου) , αλλά και καλύτερη αντιμετώπιση του θορύβου της εξόδου.

Στη εξίσωση 6.1 ο πίνακας P θα είναι θετικά ορισμένη λύση της αλγεβρικής εξίσωσης Riccati η οποία τρέχει προς τα εμπρός στο χρόνο :

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) - P(t)C^T(t)R_N^{-1}(t)C(t)P(t) + Q_N \tag{6.2}$$

με $P(0) = E[x(0)x^T(0)]$

Επίσης, ορίζουμε το κέρδος ανατροφοδότησης

$$K(t) = R^{-1}(t)B^T(t)S(t) \tag{6.3}$$

Η λύση S(t) θα προκύψει από την λύση της διαφορικής εξίσωσης Riccati που τρέχει προς τα πίσω στο χρόνο δηλαδή

$$A^T(t)S(t) + S(t)A(t) + \dot{S}(t) - S(t)B(t)R^{-1}(t)B^T(t)S(t) + Q(t) = 0 \tag{6.4}$$

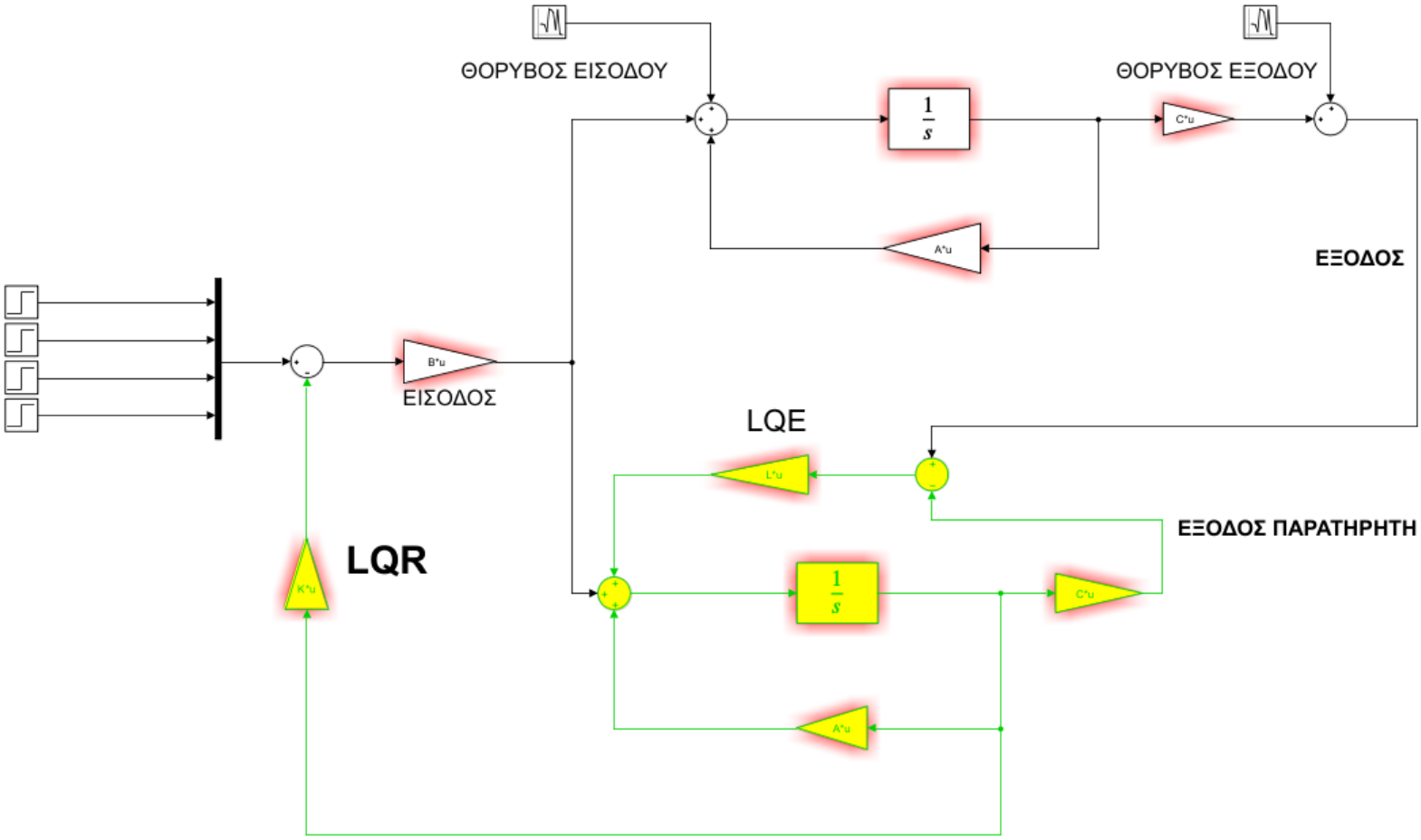
Όπου λύνοντας ως προς S(t), καταλήγουμε

$$-\dot{S}(t) = A^T(t)S(t) + S(t)A(t) - S(t)B(t)R^{-1}(t)B^T(t)S(t) + Q(t) \tag{6.5}$$

Και καταλήγοντας έχουμε το κέρδος ανατροφοδότησης K(t).

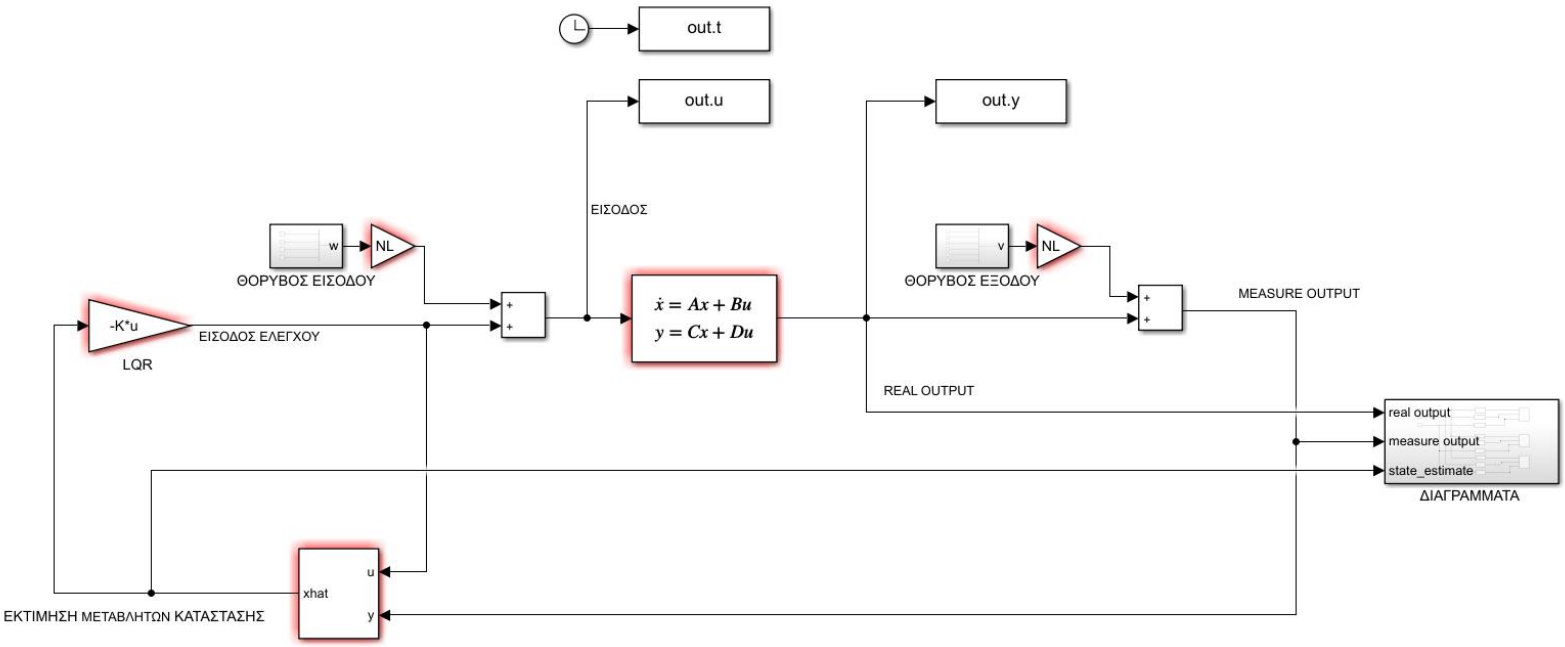
4.2.1 ΠΡΟΣΟΜΟΙΩΣΗ LQG ΣΤΟ SIMULINK

Παρακάτω βλέπουμε την αναλυτική ανάλυση του γραμμικού τετραγωνικού ρυθμιστή που έγινε στο Simulink , βλέποντας με πράσινο χρώμα την ανάδραση που παίρνουμε από τον βέλτιστο παρατηρητή (Kalman filter-LQE) και το γραμμικό τετραγωνικό ρυθμιστή (LQR) , την ανάδραση που έχουμε από την κατάσταση. Γενικά, για τις δοκιμές αλλά και το animation για το LQG δεν χρησιμοποιήθηκε η σχεδίαση που βλέπουμε στην εικόνα 4.6 αλλά η σχεδίαση στην εικόνα 4.7.



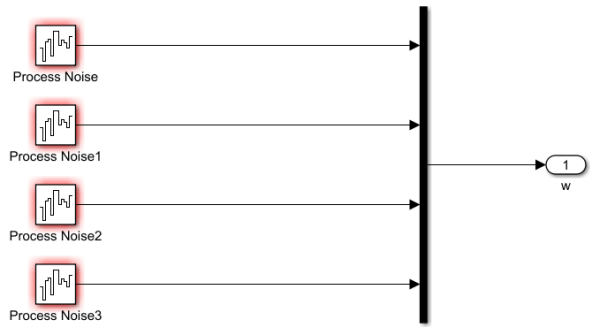
Εικόνα 4.6 : Προσομοίωση στο Simulink με βηματική είσοδο

Η σχεδίαση η οποία χρησιμοποιήθηκε για δοκιμές στο Simulink φαίνεται παρακάτω

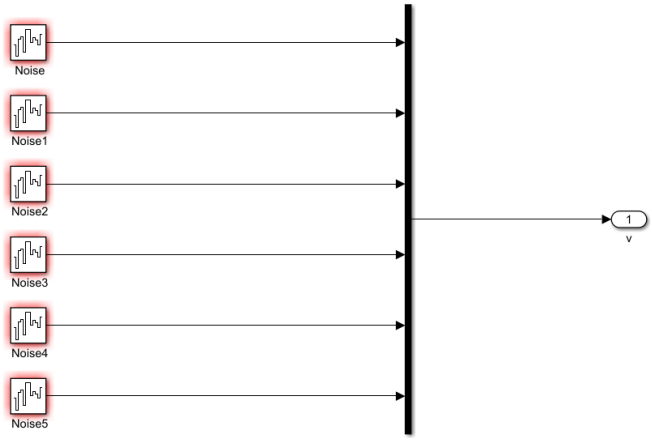


Εικόνα 4.7 : Προσομοίωση στο Simulink

Στην περίπτωση του LQG ελεγκτή το NL το ορίζουμε ως ένα , επομένως στον κώδικα μας συμπεριλαμβάνεται και ο θόρυβος εισόδου-εξόδου που είναι και το ζητούμενο έτσι ώστε να δούμε πως ανταποκρίνεται το σύστημα. Τα blocks του θορύβου αποτελούνται από λευκό θόρυβο όπως έχουμε αναφέρει, όπου στην εικόνα 4.8 έχουμε τέσσερις λευκούς θορύβους εφόσον έχουμε τέσσερις εισόδους και στην εικόνα 4.9 έξι αφού έχουμε έξι μετρήσιμες μεταβλητές κατάστασης (έξοδος).

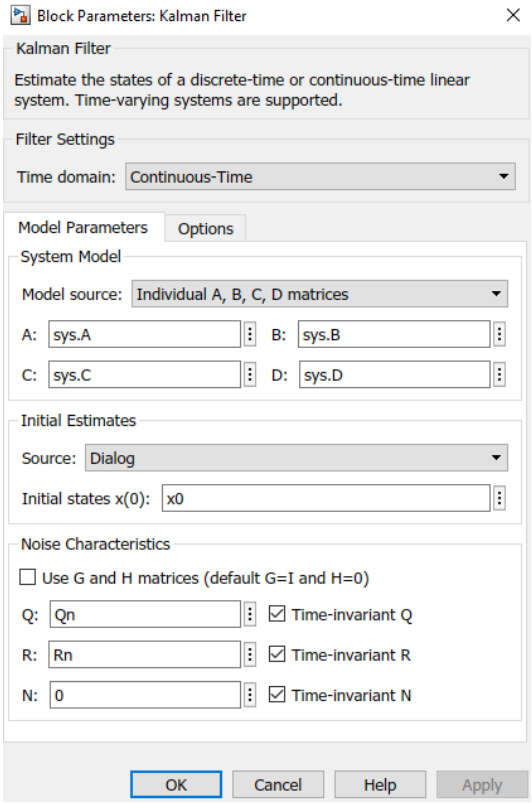


Εικόνα 4.8 : Θόρυβος εισόδου



Εικόνα 4.9 : Θόρυβος εξόδου

Βλέπουμε, στην εικόνα 4.10 ότι χρησιμοποιήθηκε το έτοιμο Kalman Filter block από την βιβλιοθήκη του Simulink στο οποίο τοποθετήσαμε τους πίνακες

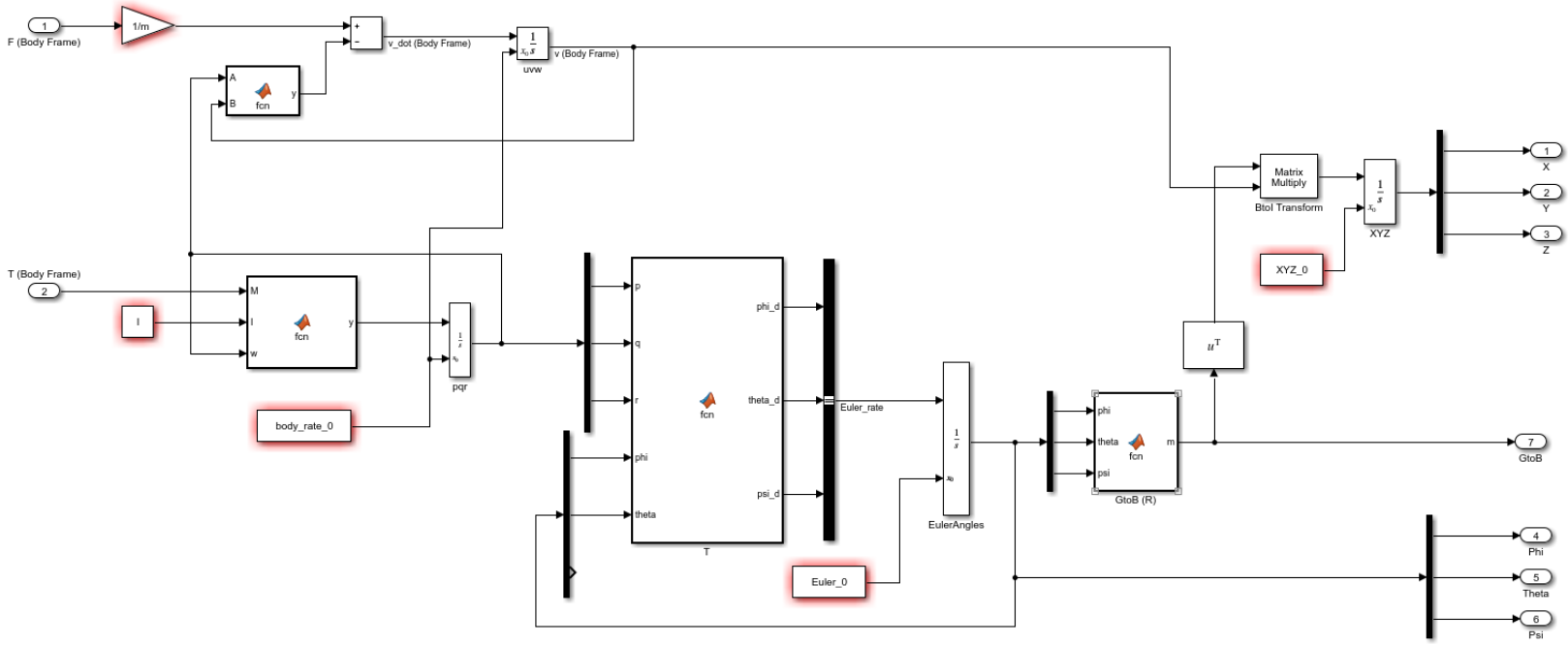


Εικόνα 4.10 : Kalman Filter block

Οι πίνακες A,B,C,D είναι οι γνωστοί πίνακες του γραμμικού μοντέλου μας , και οι πίνακες Qn,Rn είναι οι πίνακες πλάτους που είδαμε παραπάνω. (εξισώσεις 5.8-5.9)

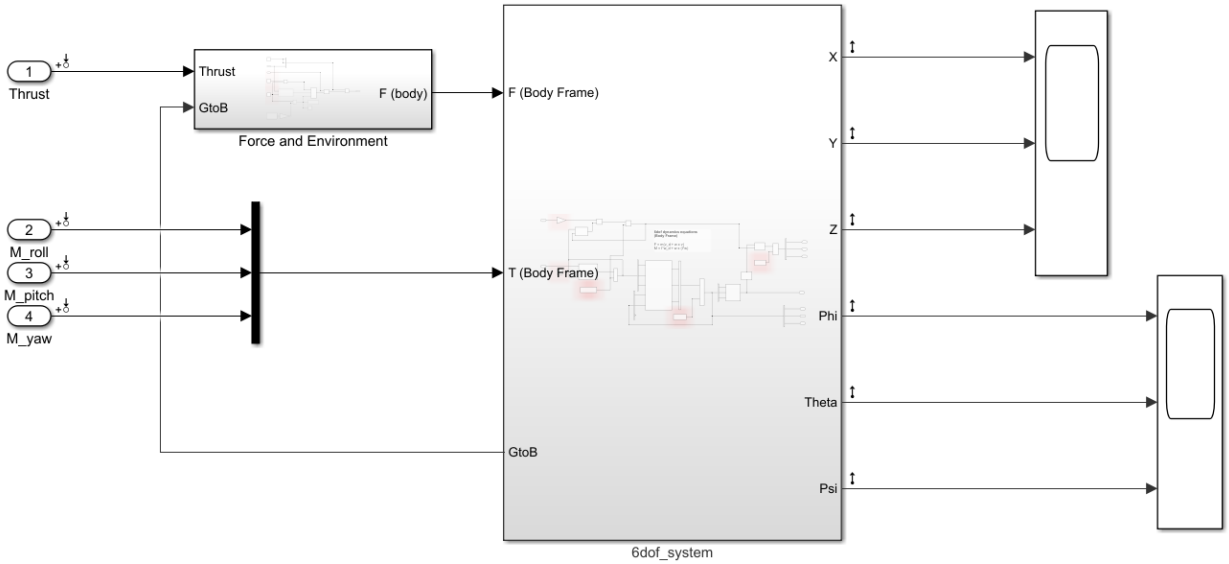
Να τονίσουμε πως για τις ανάγκες τις εργασίας , το γραμμικό μοντέλο το οποίο χρησιμοποιήθηκε διαφέρει από το μοντέλο που χρησιμοποιήθηκε στο LQR καθώς εδώ δεν είμαστε σε θέση όπως έχουμε ήδη προαναφέρει να μπορούμε να μετρήσουμε και να γνωρίζουμε όλες τις μεταβλητές κατάστασης λόγω διαταραχών (φθορά αισθητηρίων ή περιβαλλοντικές συνθήκες ή απόσταση μέτρησης) και έτσι η γραμμικοποιήση πραγματοποιήθηκε στο μοντέλο στην εικόνα 4.12.

Οι δυναμικές εξισώσεις που τροποποιήθηκαν για να προκύψει το μη γραμμικό μοντέλο με τις λιγότερες εξόδους φαίνεται παρακάτω



Εικόνα 4.11 : Δυναμικές εξισώσεις

Το μη γραμμικό μοντέλο που χρησιμοποιήθηκε για την εφαρμογή του LQG ελεγκτή



Εικόνα 4.12 : Μη γραμμικό μοντέλο

Όπου έχουμε τώρα τις μεταβλητές κατάστασης

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \psi \end{bmatrix}$$

(6.6)

Μετά την γραμμικοποιήση του μη γραμμικού μοντέλου μέσω Matlab όπως είδαμε αναλυτικά στο κεφάλαιο 3.4.3 προκύπτουν οι πίνακες

$A \in \mathbb{R}^{12\times12}$ και $B \in \mathbb{R}^{12\times4}$ και $C \in \mathbb{R}^{6\times12}$ και $D \in \mathbb{R}^{6\times4}$

(6.7)

Όπου οι πίνακες C και D είναι διαφορετικών διαστάσεων από τους πίνακες για τον ελεγκτή LQR καθώς έχουμε έξι μεταβλητές κατάστασης στην έξοδο σε αντίθεση με δώδεκα που είχαμε για το LQR.

5. ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΚΑΙ ANIMATIONS (MATLAB-SIMULINK)

Στο κεφάλαιο αυτό , θα δούμε τις διάφορες προσομοιώσεις αλλά και animations που πραγματοποιήθηκαν ώστε να δούμε πως ανταποκρίνεται ο κάθε ελεγκτής σε διάφορες συνθήκες. Αρχικά θα δούμε τον LQR για το συγκεκριμένο γραμμικό μοντέλο που προέκυψε μετά την γραμμικοποιήση όπως είδαμε, αλλά και τον LQG ελεγκτή με το διαφορετικό γραμμικό μοντέλο από αυτό του LQR. Έπειτα θα εφαρμόσουμε τους δύο ελεγκτές απευθείας στα μη γραμμικα μοντέλα για να δούμε τις αποκρίσεις αλλά και να εφαρμόσουμε ένα path μέσω waypoints που θα εξηγηθεί αναλυτικά παρακάτω στο κεφάλαιο. Επίσης, αναπτύχθηκε κώδικας ώστε να έχουμε πιο ρεαλιστικό περιβάλλον βάζοντας κάποια 3D εμπόδια στο χώρο , δημιουργώντας έτσι διαφορετικά plots . Επιλέγοντας τα κατάλληλα waypoints το τετρακόπτερο καταφέρνει να αποφύγει τα εμπόδια και να προσγειωθεί με επιτυχία όπως θα δούμε στα animations.

Επίσης δημιουργήθηκαν διαγράμματα που αφορούν τα X,Y,Z στο χώρο αλλά και τις εισόδους (ώθηση (thrust), ροπή (τ_φ), ροπή (τ_θ), ροπή (τ_ψ)). Ακόμα , πραγματοποιήθηκε και μία εκτίμηση του κόστους ισχύος της μπαταρίας χρησιμοποιώντας μία παρόμοια συνάρτηση με αυτή του Lyapunov που χρησιμοποιείται στη συνάρτηση κόστος του LQR. Έτσι είμαστε σε θέση να εκτιμήσουμε την ενέργεια (E) όπου η τιμή της υπολογίζεται από την γραμμική και γωνιακή επιτάχυνση. Η ενέργεια παρουσιάζεται σε σχέση με το χρόνο για κάθε γράφημα ξεχωριστά όπως θα δούμε. Παρακάτω θα γίνει μία εξήγηση της συνάρτησης. Έτσι ορίζουμε μία εκτίμηση του πίνακα $P_{o\lambda}(Power)$ ως το τετράγωνο των εισόδων όπως φαίνεται παρακάτω

$$P_{o\lambda} = U_1^2 + U_2^2 + U_3^2 + U_4^2 \tag{6.8}$$

Όπου

U_1^2 : Πίνακας Ώθησης (Thrust)

U_2^2 : Πίνακας Ροπής (τ_φ)

U_3^2 : Πίνακας Ροπής (τ_θ)

U_4^2 : Πίνακας Ροπής (τ_ψ)

Υπολογίζοντας την ενέργεια

$$E = P' \times t_s \tag{6.9}$$

με t_s=0.01 sec

και

$$P' = \sum_{t=0}^{t} P_{o\lambda_t} \tag{7.0}$$

Το P' είναι το άθροισμα της ισχύος του πίνακα $P_{o\lambda}$ για κάθε χρονική στιγμή του χρόνου προσομοίωσης t.

Με αυτό τον τρόπο πραγματοποιήθηκε μία εκτίμηση της δαπάνης της ισχύος σε σχέση με το χρόνο της προσομοίωσης .

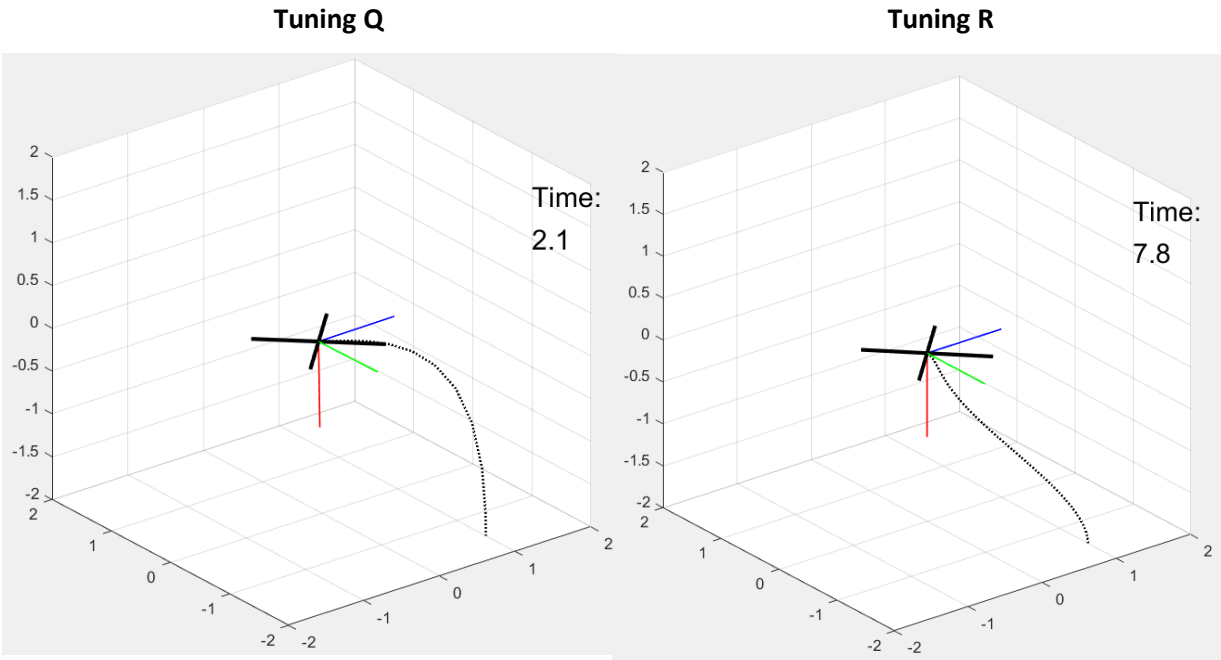
Τέλος, απαραίτητη προϋπόθεση για την σωστή λειτουργία του κώδικα και ως εκ τούτου των προσομοιώσεων και των animations είναι η αρχικοποίηση των παρακάτω παραμέτρων στην αρχή του κώδικα όπως φαίνεται στον πίνακα 1.0.

Ορισμός	Σύμβολο	Τιμή	Μονάδα μέτρησης
Επίταγχυνση της Βαρύτητας	g	9.81	m/s^2
Μάζα	m	0.2	m
Χρόνος δείγματος	ts	0.01	s
Ροπή αδράνειας ως προς τον άξονα x	I_{xx}	0.1	$Kg.m^2$
Ροπή αδράνειας ως προς τον άξονα y	I_{yy}	0.1	$Kg.m^2$
Ροπή αδράνειας ως προς τον άξονα z	I_{zz}	0.15	$Kg.m^2$

Πίνακας 1.0 : Παράμετροι για την προσομοίωση

5.2 LQR-Tuning (Q&R)

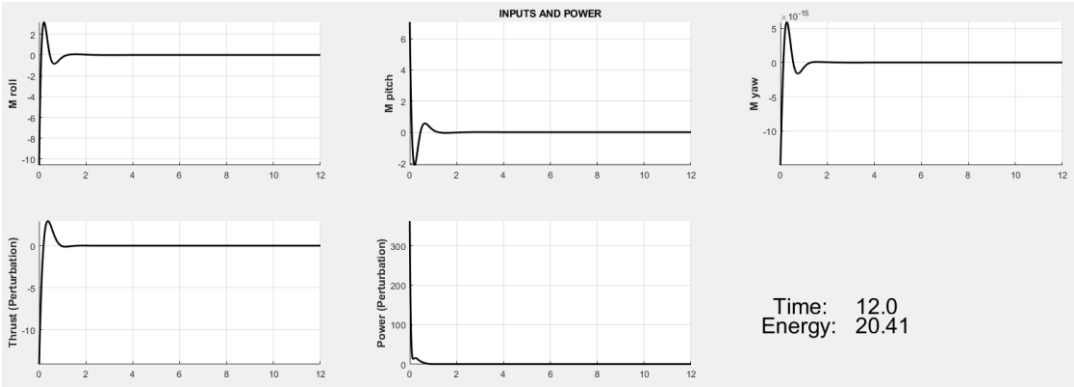
Ο ελεγκτής έχει ως στόχο την επίτευξη της ισορροπίας ανάμεσα στην απόδοση του τετρακοπτέρου και την ενέργεια που καταναλώνει όπως περιγράφηκε αναλυτικά στο κεφάλαιο 4.2. Ανάλογα με την επιλογή των βαρών που θα επιλέξουμε θα έχουμε διαφορετική απόκριση του τετρακοπτέρου. Η αύξηση του κόστους απόδοσης (Q) θα οδηγήσει σε ένα τετρακόπτερο ταχύτερης απόκρισης ενώ αντίθετα, η αύξηση του κόστους ελέγχου (R) θα οδηγήσει σε αργότερη απόκριση αλλά θα έχουμε εξοικονόμηση ενέργειας. Έτσι , παρακάτω παρατηρούνται δύο animations , όπου στη μία περίπτωση επιλέχτηκαν υψηλές τιμές για τον πίνακα Q και στην άλλη υψηλές τιμές για τον πίνακα R. Επίσης επιλέχτηκαν αρχικές συνθήκες X=1 , Y=-1.5, Z=2, με το τετρακόπτερο να καταλήγει στο σημείο ισορροπίας (0,0,0).



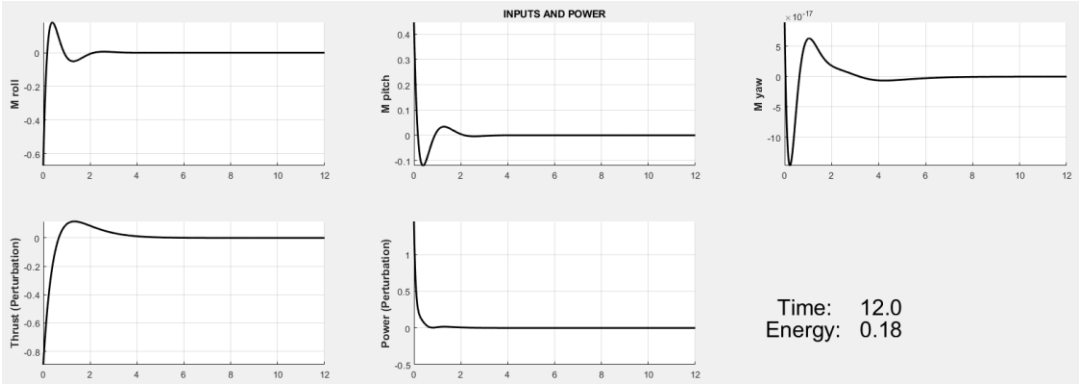
Εικόνα 5.0 : Tuning Q&R

Ο χρόνος εξομοιώσης και για τις δύο περιπτώσεις είναι 12 sec. Όπως παρατηρείται και απο τα animations στην πρώτη περίπτωση (Tuning Q) έχουμε πιο απότομη και πιο γρήγορη καμπύλη καθώς το τετρακόπτερο φτάνει στο σημείο ισορροπίας (0,0,0) σε t=2.1s ενω στην δεύτερη (Tuning R) έχουμε πιο ομαλή και πιο αργή αφού φτάνει στο σημείο σε t=7.8s.

Παρακάτω , παρουσιάζονται τα γραφήματα των εισόδων αλλά και η ενέργεια που καταναλώνεται , όπου συγκεκριμένα στην πρώτη περίπτωση που έχουμε υψηλή απόδοση έχουμε energy=20.41, ενώ στην δεύτερη περίπτωση έχουμε energy=0.18.

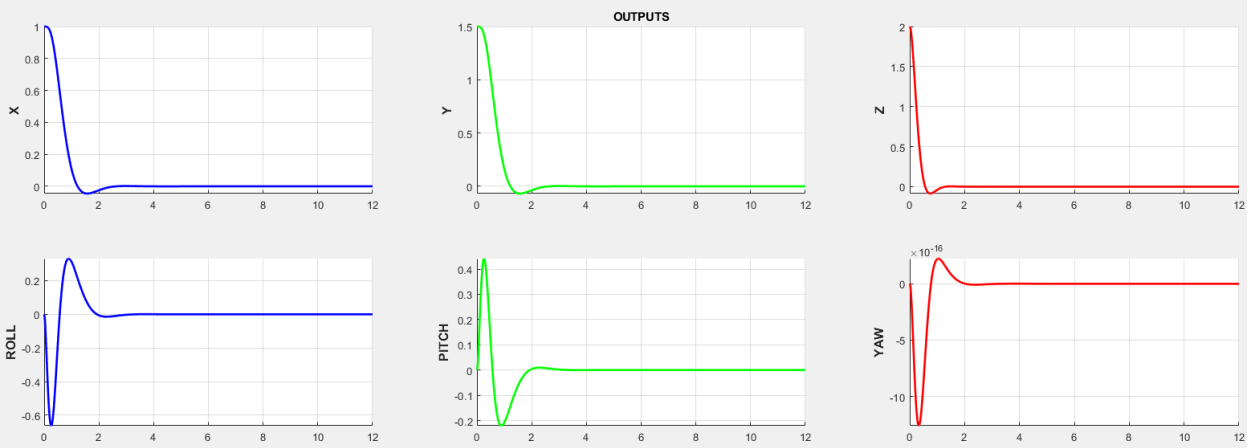


Εικόνα 5.1 : Tuning Q

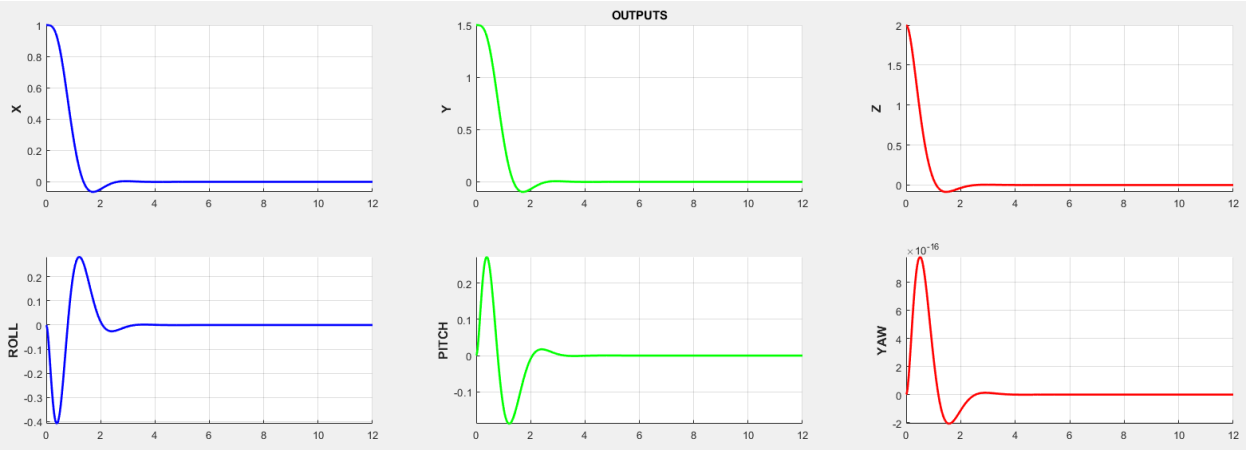


Εικόνα 5.2 : Tuning R

Παρακάτω φαίνονται οι αποκρίσεις των εξόδων όπου φαίνεται πόσο γρήγορη είναι η απόκριση του τετρακοπτέρου στο διάγραμμα Z αλλά και στο Yaw στην εικόνα 5.3 σε σχέση με την εικόνα 5.4.



Εικόνα 5.3 : Tuning Q



Εικόνα 5.4 : Tuning R

Συμπεραίνουμε, πως αλλάζοντας απλά τα βάρη για τους πίνακες Q και R , η διαδρομή που ακολουθεί το τετρακόπτερο αλλά και η απόκριση του αλλάζει ανάλογα με την επιλογή , χωρίς την παρουσία PID. Έτσι παρακάτω θα δούμε τον έλεγχο ύψους αιώρησης του τετρακοπτέρου αλλά και έλεγχο γωνίας περιστροφής (roll).

5.1 LQR-Έλεγχος ύψους αιώρησης (Hovering)

Η πρώτη υλοποίηση εξομοίωσης είναι η προσπάθεια επίτευξης ελέγχου ύψους αιώρησης (hover) , δηλαδή η σταθεροποίηση του τετρακοπτέρου σε σταθερό ύψος πάνω από ένα συγκεκριμένο σημείο μετά την απογείωση του. Αυτό επιτυγχάνεται με τρεις τρόπος :

1. Με τη χρήση βηματικής εισόδου , μηδενίζοντας τις ροπές και διατηρώντας μόνο την ώθηση (thrust) όπως φαίνεται στην εικόνα 5.6.
2. Χρησιμοποιώντας το block desired state το οποίο δημιουργήθηκε για τις ανάγκες τις εργασίας ,το οποίο αναλύεται παρακάτω.
3. Μηδενίζοντας όλες τις αρχικές σύνθηκες x0(initial conditions) εκτός από το z.

Έτσι, στην εξομοίωση ορίζονται μηδενικές αρχικές συνθήκες που σημαίνει πως το τετρακόπτερο θα ισορροπεί στο κέντρο του γραφήματος και δεν θα έχει κάποια μεταβολή στο χώρο. Ισχύει

$$x_0 = \begin{bmatrix} x_{0(1)} \\ x_{0(2)} \\ x_{0(3)} \\ x_{0(4)} \\ x_{0(5)} \\ x_{0(6)} \\ x_{0(7)} \\ x_{0(8)} \\ x_{0(9)} \\ x_{0(10)} \\ x_{0(11)} \\ x_{0(12)} \end{bmatrix} = \begin{bmatrix} \varphi \\ \theta \\ \psi \\ x \\ y \\ z \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.1}$$

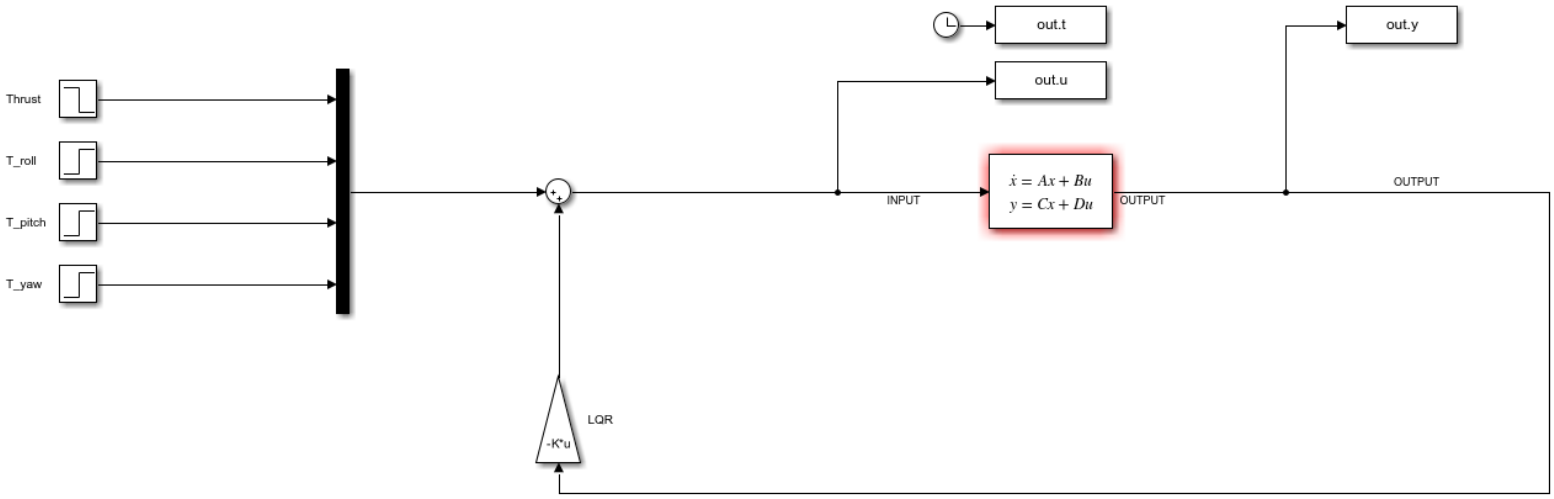
Έπειτα γίνεται η επιλογή των πινάκων Q και R για την συγκεκριμένη εξομοίωση , όπου αρχικά επιλέγονται ως μοναδιαίοι πίνακες ώστε να δούμε την απόκριση τους. Έτσι, με τον τρόπο αυτό δεν δίνουμε κάποιο βάρος στο διάνυσμα κατάστασης αλλά ούτε και στην ενέργεια. Μετά την εντολή LQR που χρησιμοποιούμε στο Matlab προκύπτουν οι πόλοι του συστήματος.

```
P =

  -1.0001 + 0.0000i
  -1.0001 + 0.0000i
  -1.0117 + 0.0000i
  -1.0215 + 0.0000i
  -2.1594 + 2.2669i
  -2.1594 - 2.2669i
  -2.1594 + 2.2669i
  -2.1594 - 2.2669i
  -4.8945 + 0.0000i
  -6.5895 + 0.0000i
  -9.9976 + 0.0000i
  -9.9976 + 0.0000i
```

Εικόνα: 5.5 : Πόλοι του συστήματος

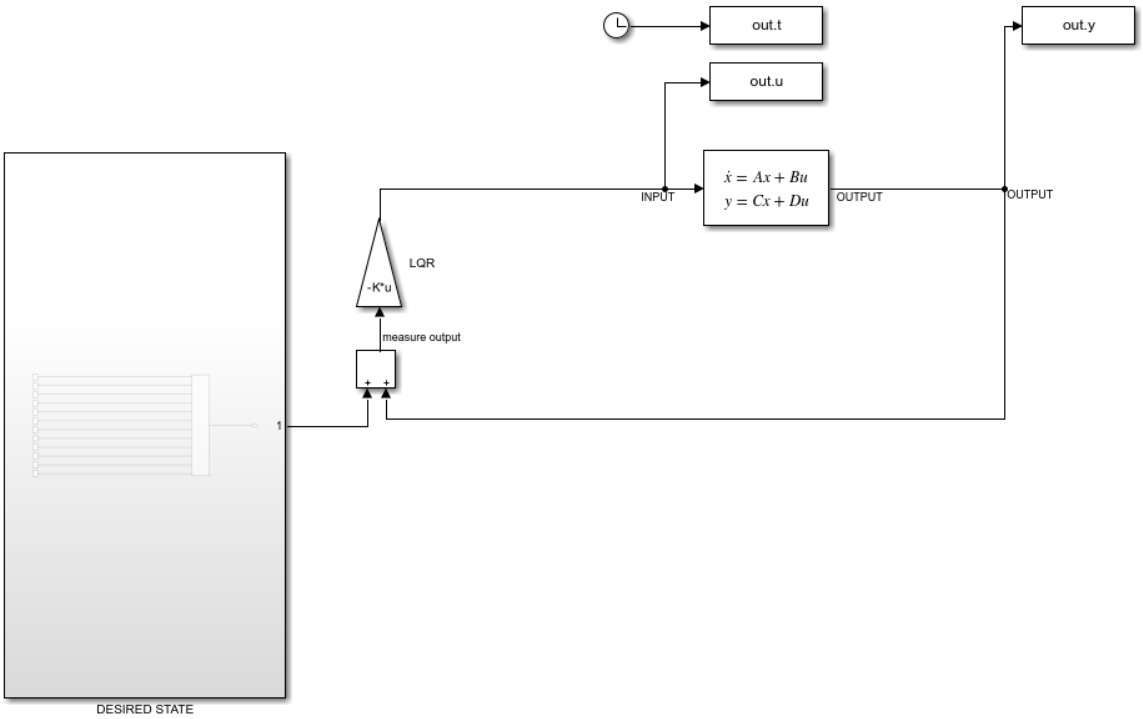
Αξίζει να σχολιαστεί μετά την εφαρμογή του ελεγκτή , πως το σύστημα κλειστού βρόγχου βρίσκεται σε ευστάθεια. Βλέποντας της πόλους του συστήματος μπορούμε να επιβεβαιωθούμε περί αυτού καθώς το πραγματικό μέρος των μηγαδικών πόλων βρίσκεται στο αριστερό μιγαδικό ημιεπίπεδο. Επίσης, παρατηρούνται οι μεγάλες τιμές του πραγματικού μέρους των τεσσάρων πόλων όπου μας προσδίδουν ταχύτητα στην απόσβεση του μοντέλου.

Παρακάτω φαίνεται ο πρώτος τρόπος με την βηματική είσοδο για την επίτευξη του ελέγχου ύψους αιώρησης.

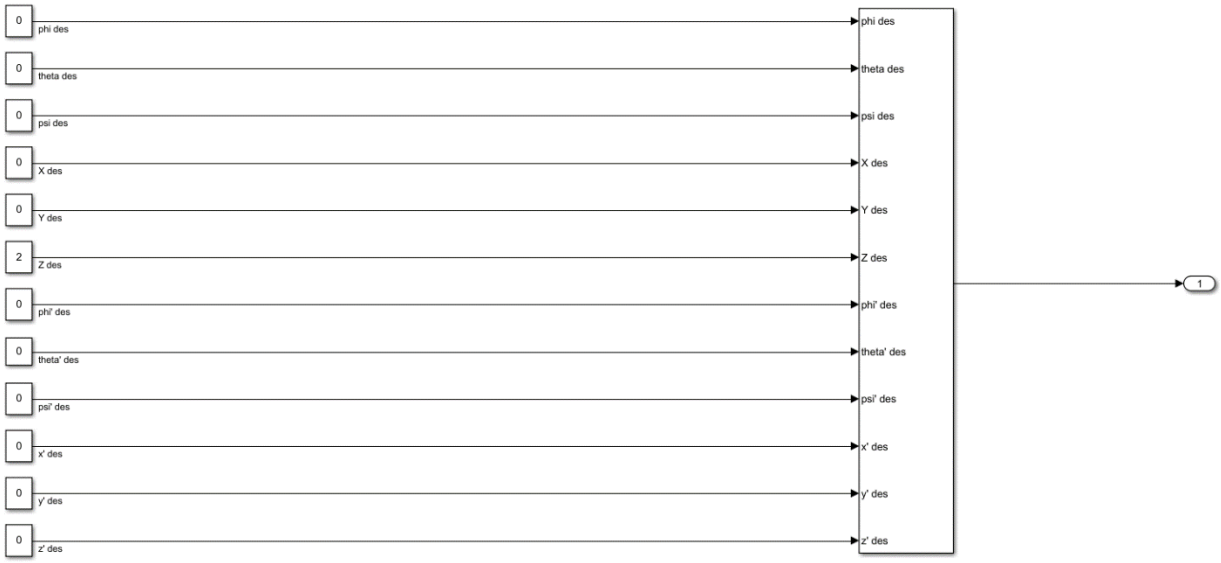


Εικόνα 5.6 : Βηματική είσοδος

Παρακάτω φαίνεται ο δεύτερος τρόπος που δημιουργήθηκε για να εισάγουμε την επιθυμητή κατάσταση.



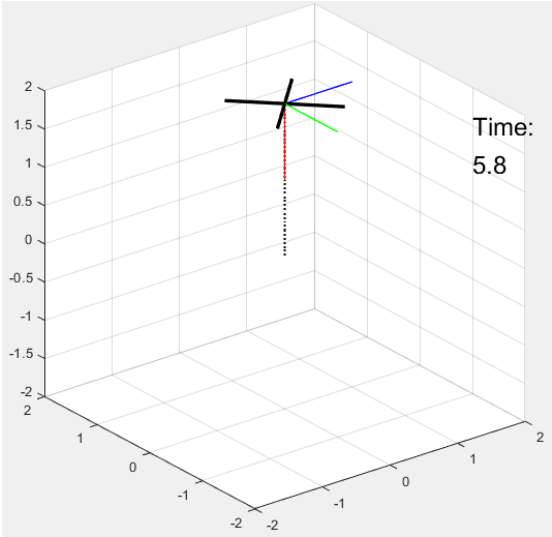
Εικόνα 5.7: LQR με Desired State



Εικόνα 5.8: Μπλοκ Desired State

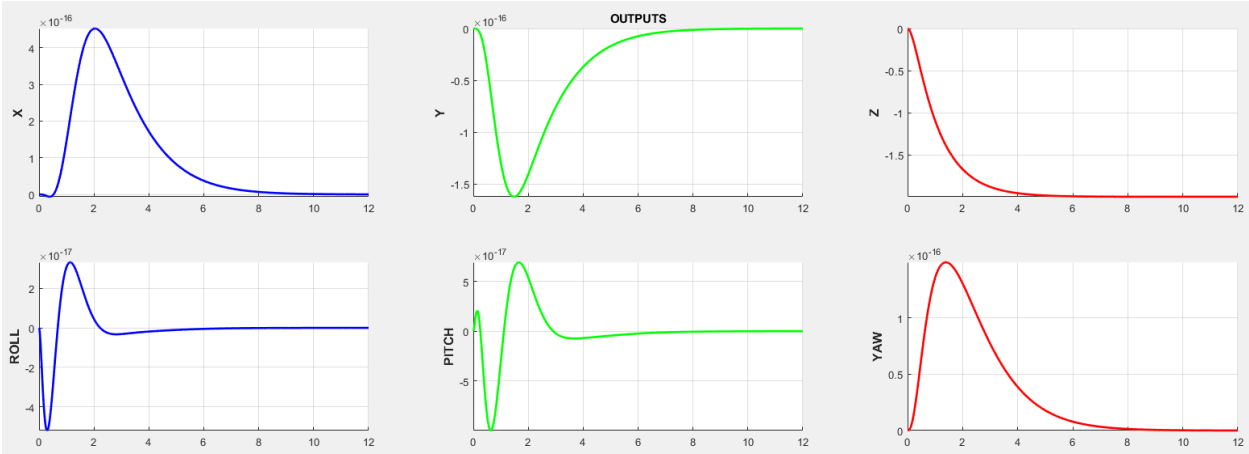
Ο δεύτερος τρόπος ο οποίος χρησιμοποιήθηκε έχει ακριβώς την ίδια απόκριση με τον τρίτο τρόπο. Η διαφορά των δύο, βρίσκεται στο γεγονός ότι στον δεύτερο τρόπο εισάγουμε την επιθυμητή κατάσταση, δηλαδή μπορούμε να εισάγουμε την κίνηση που μπορεί να κάνει το τετρακόπτερο μέσω του Simulink μπλοκ που βλέπουμε παραπάνω (εικόνα 5.7) αλλάζοντας τις τιμές (εικόνα 5.8). Αντίθετα στον τρίτο τρόπο εισάγουμε την αρχική κατάσταση που έχει το τετρακόπτερο όπου στη συνέχεια ισορροπεί στο μηδέν.

Για τις εξομοιώσεις χρησιμοποιήθηκε το μπλοκ με την επιθυμητή κατάσταση , έτσι ορίστηκε το z=2 και πραγματοποιήθηκε εξομοίωση διάρκειας 12 δευτερολέπτων. Παρακάτω φαίνεται το animation για τον έλεγχο ύψους αιώρησης

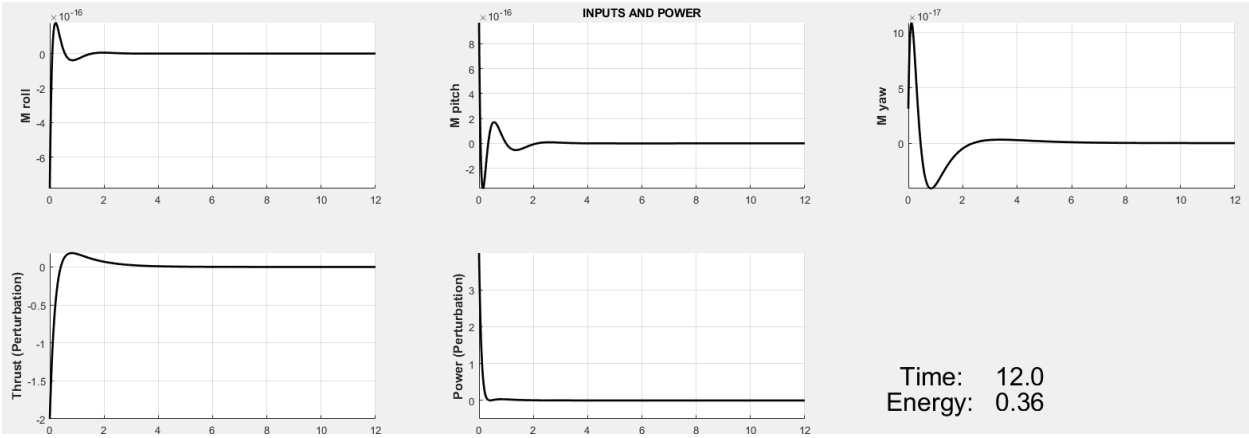


Εικόνα 5.9: Animation ελέγχου ύψους αιώρησης

Παρατηρείται από το animation πως το τετρακόπτερο απογειώνεται από το σημείο (0,0,0) και φτάνει στο σημείο (0,0,2) όπου ισορροπεί σε χρόνο t=5.8s. Ο χρόνος για να φτάσει στο σημείο ισορροπίας είναι σχετικά αργός για την επιλογή των πινάκων Q και R που κάναμε. Παρακάτω φαίνονται και οι αποκρίσεις αρχικά για την έξοδο (X,Y,Z,roll,pitch,yaw) και έπειτα για την είσοδο (thrust,T_roll,T_pitch,T_yaw) εκτιμώντας παράλληλα και την ενέργεια.

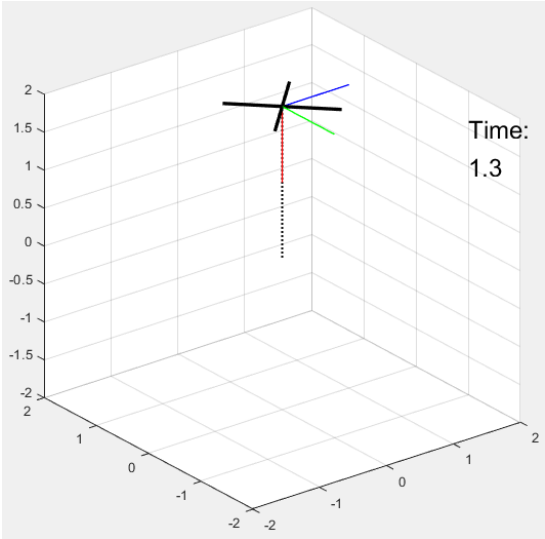


Εικόνα 5.10: Αποκρίσεις εξόδου



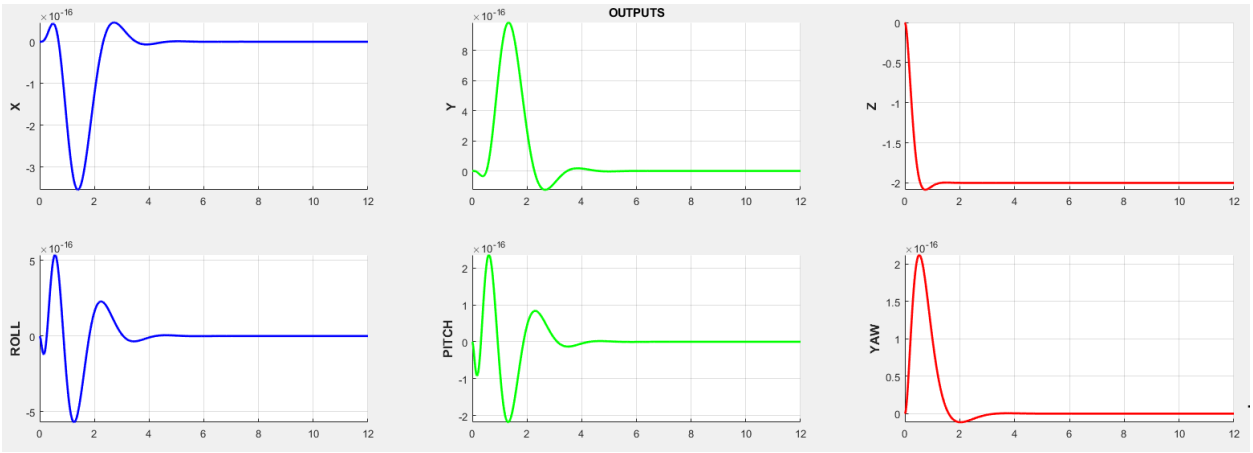
Εικόνα 5.11: Αποκρίσεις εισόδου και ενέργεια

Παρατηρούμαι πως η ενέργεια είναι αρκετά μικρή για την επιλογή που κάναμε. Να τονίσουμε εδώ πως το scale στις εισόδους αλλάζει. Στη συνέχεια θα γίνει αλλαγή των βαρών , και θα επιλέξουμε τέτοιο ώστε να έχουμε αύξηση του κόστους απόδοσης Q , οπότε θα βάλουμε μία αρκετά μεγάλη τιμή στο z , παραδείγματος χάρη Q(6)=50. Παρακάτω φαίνονται οι αποκρίσεις για την επιλογή που κάναμε.



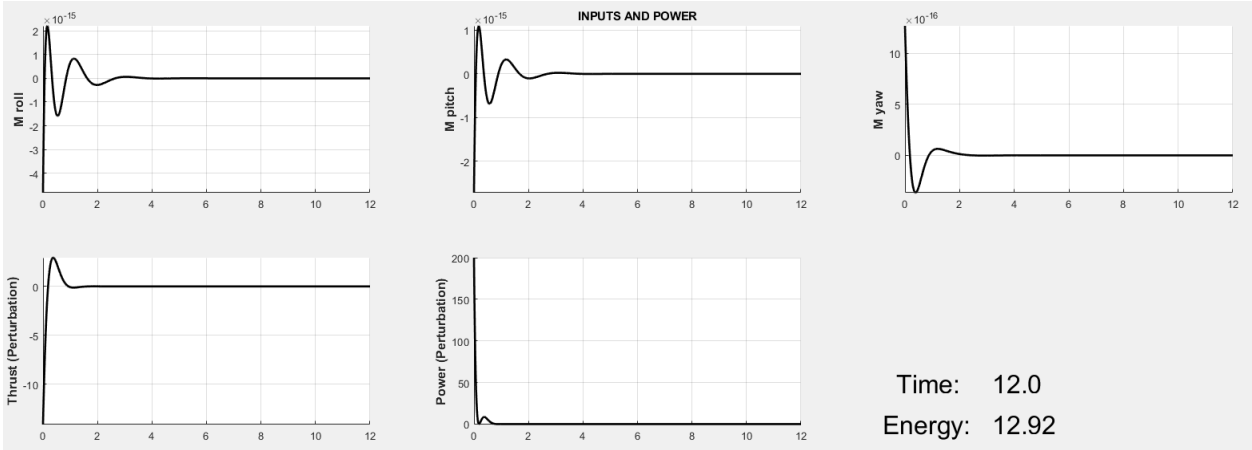
Εικόνα 5.12: Αύξηση του κόστους απόδοσης

Βλέπουμε , με την επιλογή που κάναμε μία πάρα πολύ γρήγορη απόκριση καθώς το τετρακόπτερο σε t=1.3s βρίσκεται στο επιθυμητό σημείο ισορροπίας.



Εικόνα 5.13: Αποκρίσεις εξόδου

Μπορούμε να εντοπίσουμε τις μεταβολές που έχουν οι καμπύλες, σε σχέση με την εικόνα 5.10, και επίσης πόσο γρήγορα φτάνει στο μηδέν στο άξονα Z στην εικόνα 5.13 σε σχέση με την εικόνα 5.10.



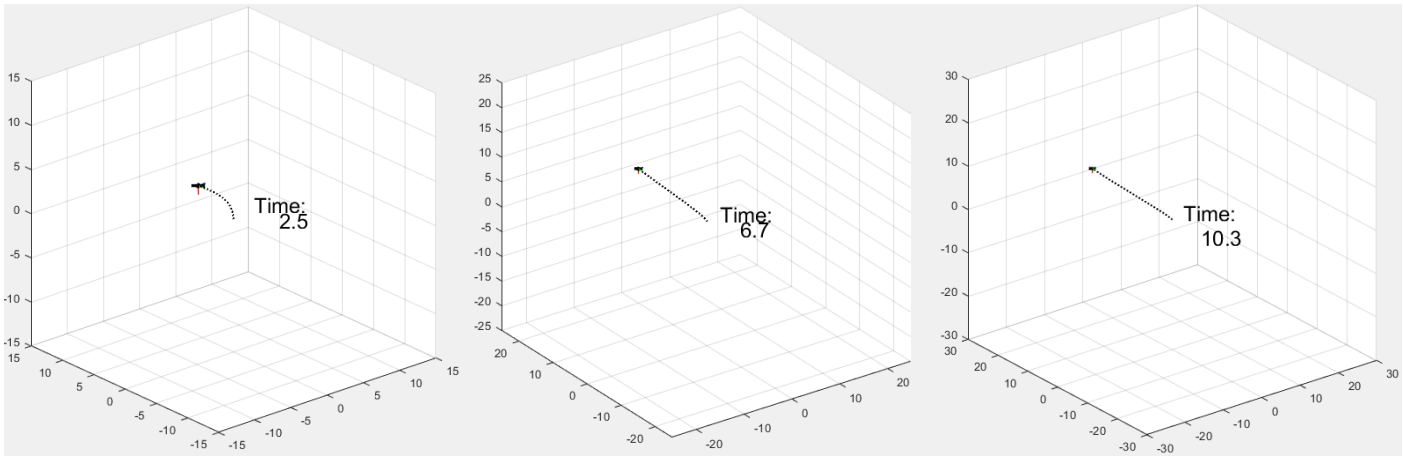
Εικόνα 5.14: Αποκρίσεις εισόδου και ενέργεια

Βλέπουμε όμως πως η ενέργεια που απαιτείται είναι πολύ μεγαλύτερη για να πετύχουμε αυτήν την απόκριση και την διαφορά που έχουμε στα διαγράμματα σε σχέση με τις ταλαντώσεις που έχουμε στις εισόδους μας.

5.2 LQR-Έλεγχος γωνίας περιστροφής (roll ΄φ΄)

Η συγκεκριμένη εξομοίωση αφορά τον έλεγχο γωνίας περιστροφή στο roll , δηλαδή αφορά την γωνία περιστροφής γύρω από τον άξονα X_B (Διάταξη πτήσης ,Εικόνα 3.3) . Συγκεκριμένα , θα χρησιμοποιήσουμε την ίδια τακτική που ακολουθήσαμε και στον έλεγχο ύψους αιώρησης , μόνο που εδώ θα προσθέσουμε τιμή και στο 'φ' στο μπλοκ Desired State (Εικόνα 5.8) που δείξαμε στο προηγούμενο κεφάλαιο όπου θα βάλουμε την τιμή 3. Χρησιμοποιώντας αρχικά τους κλασσικούς πίνακες για το Q και το R (περίπτωση 2), έπειτα πολύ μεγάλες τιμές για το δείκτη απόδοσης (περίπτωση 1) και στη συνέχεια, πολύ μεγάλες για το κόστος ελέγχου (περίπτωση 3), έχουμε τις εξής διαδρομές

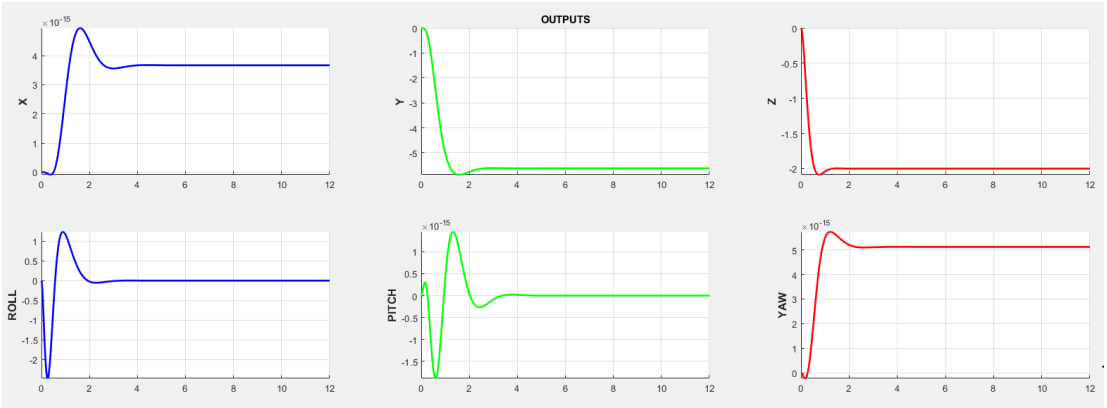
Περίπτωση 1 **Περίπτωση 2** **Περίπτωση 3**



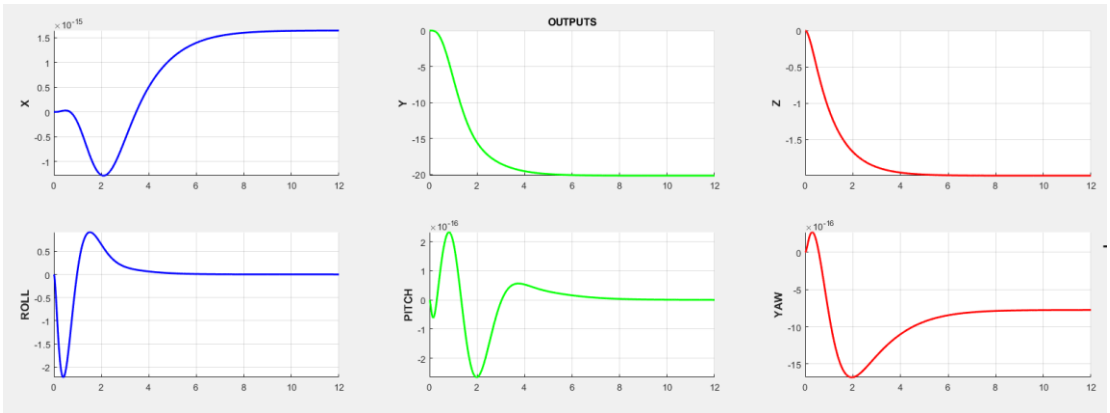
Εικόνα 5.15: Διαδρομή τετρακοπτέρου για roll ‘φ’

Παρατηρούμε πως η διαδρομή που ακολουθάει το τετρακόπτερο στην περίπτωση με μεγάλο δείκτη απόδοσης είναι πολύ καλύτερη σε σχέση με την διαδρομή που ακολουθάει το τετρακόπτερο στις άλλες δύο περίπτωσεις όπου απομακρύνεται πάρα πολύ καθώς αν δούμε έχουμε και μεγαλύτερα scale στο view του άξονα y. Επίσης και η ταχύτητα που ολοκληρώνει τη διαδρομή είναι πολύ καλύτερη.

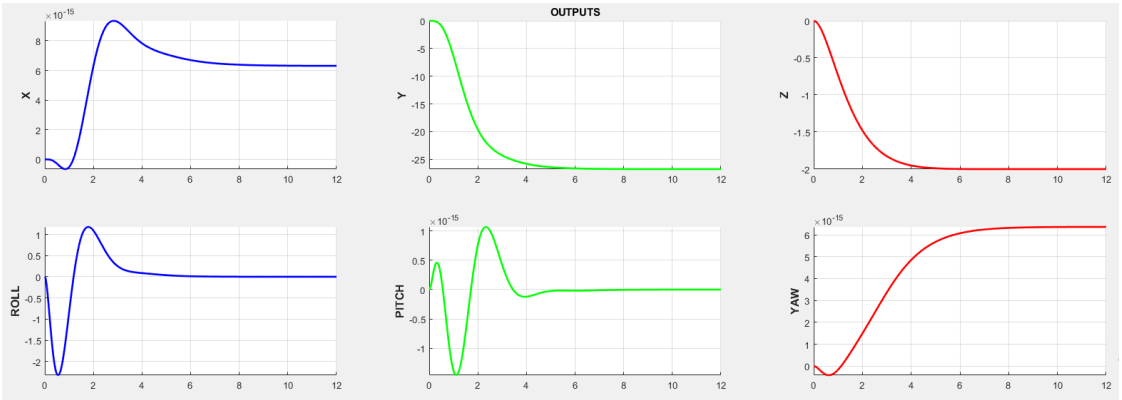
Παρακάτω φαίνονται τα διαγράμματα εξόδων



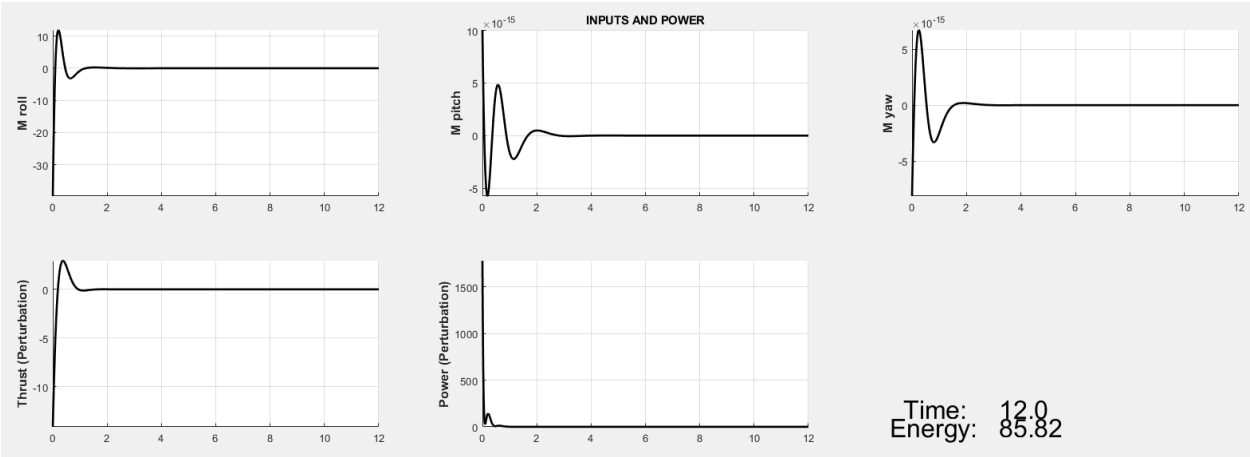
Εικόνα 5.16: Περίπτωση 1



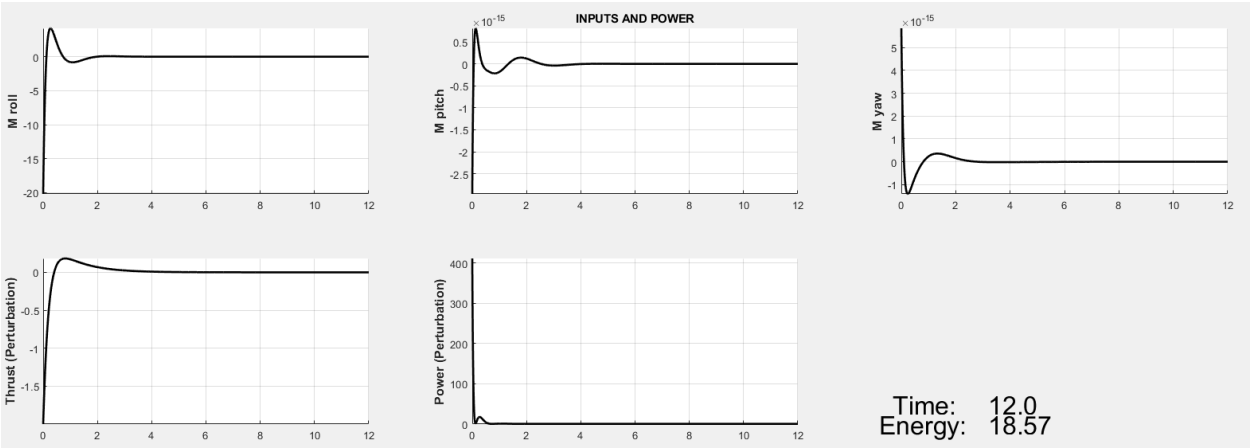
Εικόνα 5.17: Περίπτωση 2



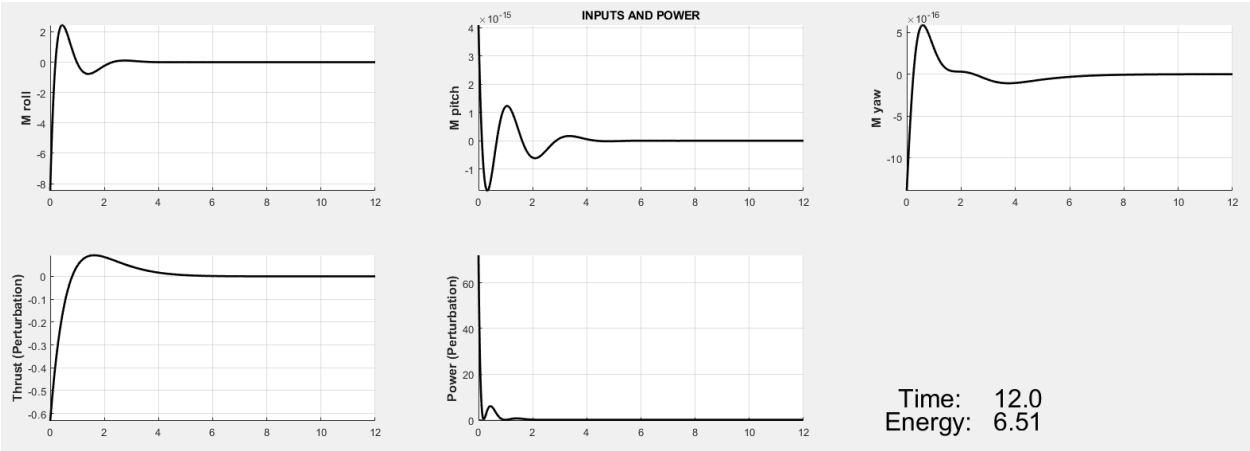
Εικόνα 5.18: Περίπτωση 3



Εικόνα 5.19: Περίπτωση 1



Εικόνα 5.20: Περίπτωση 2



Εικόνα 5.21: Περίπτωση 3

Είναι εμφανής από τα διαγράμματα οι διαφορές των τριών περιπτώσεων. Αν παρατηρήσουμε τα διαγράμματα των εξόδων , στην πρώτη περίπτωση έχουμε πολύ γρήγορη απόκριση και καλύτερη ευστάθεια , από τις άλλες δύο περιπτώσεις. Το πρόβλημα είναι ότι η εκτίμηση ενέργειας είναι πάρα πολύ μεγάλη (Εικόνα 5.19) σε σχέση με τις άλλες δύο περιπτώσεις.

5.3 LQG-Εξομοίωση με αρχικές συνθήκες

Όπως έχει ήδη προαναφερθεί στο κεφάλαιο 4.5 , το γραμμικό μοντέλο το οποίο χρησιμοποιήθηκε για τις εξομοιώσεις είναι διαφορετικό από αυτό του LQR καθώς έχουμε λιγότερες μεταβλητές κατάστασης(αβέβαιο μοντέλο). Επίσης , λαμβάνουμε υπόψη τον θόρυβο των μετρήσεων κάτι το οποίο δεν λαμβάνεται στις εξομοιώσεις του LQR καθώς θεωρούμε ιδανικές συνθήκες. Αρχικά, θα γίνει εξομοίωση για χρόνο t=12s χρησιμοποιώντας τους κλασσικούς πίνακες για το LQR (μοναδιαίους) και βάζοντας για το πλάτος του θορύβου των μετρήσεων και το πλάτος των διαταραχών του συστήματος τις τιμές

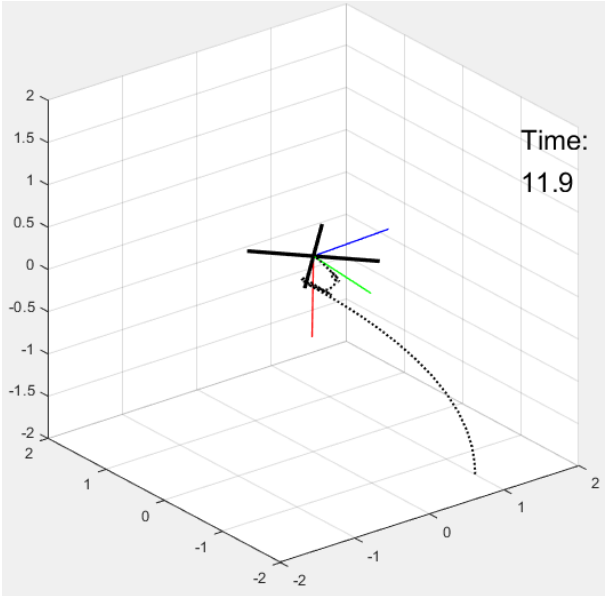
$$\mathrm{R}_n = 0.001 \tag{7.2}$$

Και

$$Q_n = 0.001 \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

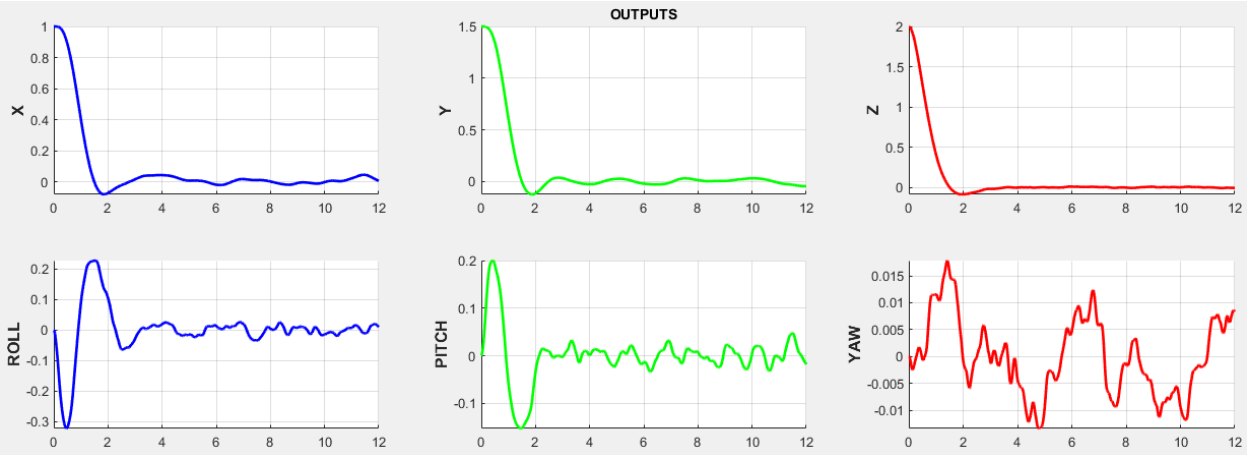
Παρατηρείται από τον πίνακα Q_n ότι το πλάτους διαταραχών εφαρμόζεται στις μεταβλητές κατάστασης που δεν μπορούμε να μετρήσουμε όπου είναι οι $\dot{\varphi}, \dot{\theta}, \dot{\psi}, \dot{x},\ \dot{y}, \dot{z}$ όπου θα γίνει εκτίμηση αυτών μαζί με των μετρήσιμων με το Kalman filter block.

Πραγματοποιώντας την εξομοίωση με αρχικές συνθήκες (X=1,Y=1.5,Z=2) όπως στο κεφάλαιο 5.2 και τις τιμές των θορύβων εισόδου-εξόδου w=0.001 και v=0.001 φαίνεται το παρακάτω animation

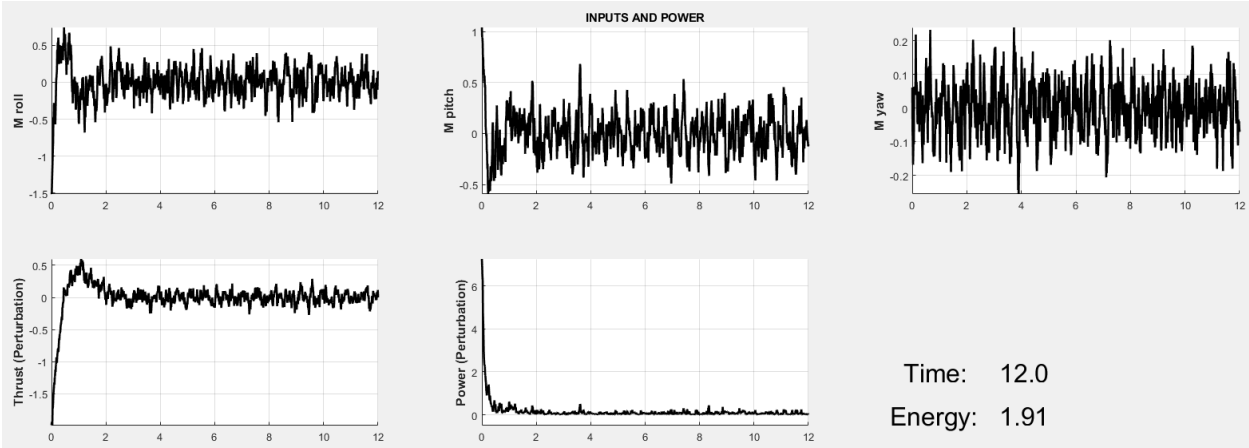


Εικόνα 5.22: Εξομοίωση για αρχικές συνθήκες

Παρατηρείται ότι στο συγκεκριμένο animation το τετρακόπτερο αντιπροσωπεύει πιο πολύ την πραγματικότητα. Φαίνεται ότι δεν μπορεί να ισορροπήσει, αυτό μπορούμε να το δούμε και από τα διαγράμματα που δημιουργήθηκαν.



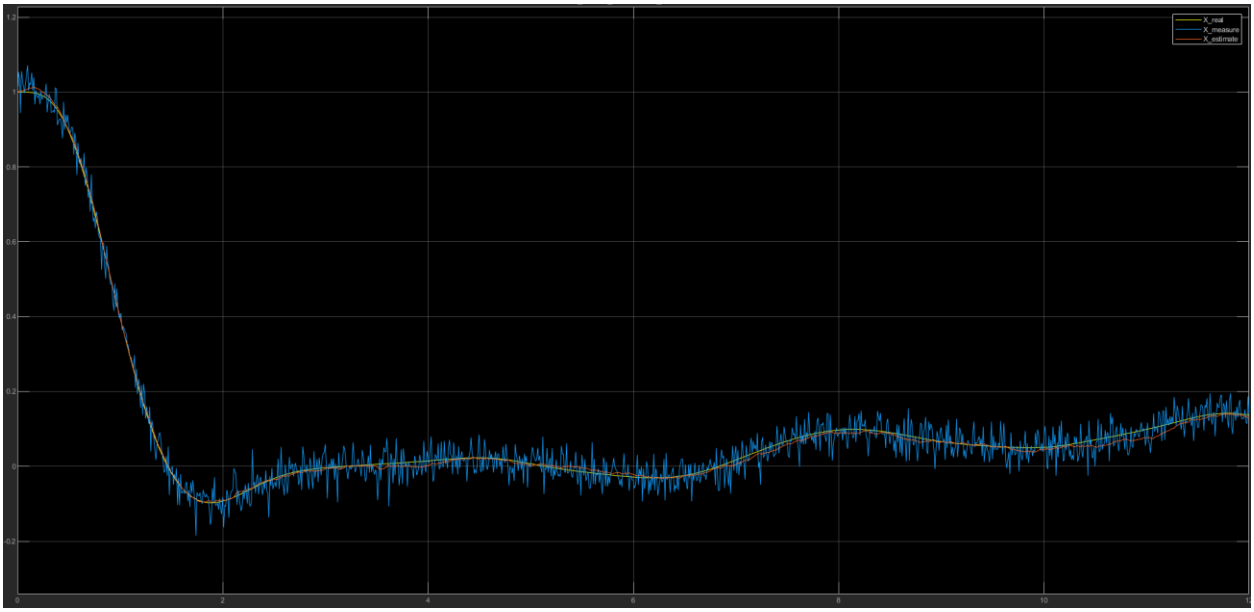
Εικόνα 5.23: Διάγραμμα εξόδων



Εικόνα 5.24: Διάγραμμα εισόδων και εκτίμηση ενέργειας

Παρατηρείται και στα διαγράμματα για το X,Y,Z ότι το τετρακόπτερο αδυνατή να ισορροπήσει στο σημείο (0,0,0) και παρουσιάζει μία ελαφρώς αστάθεια αν δούμε και τα διαγράμματα Roll,Pitch,Yaw. Επίσης στην εικόνα 3.7.9 φαίνεται η επίδραση του θορύβου στην είσοδο.

Παρακάτω δημιουργήθηκε ένα scope μπλοκ για να γίνει κατανοητό και να συγκριθεί η επίδραση του Kalman Filter. Θα γίνει σύγκριση του X πραγματικού με του X μετρήσιμου και του X εκτιμήσιμου ($\hat{x}$).

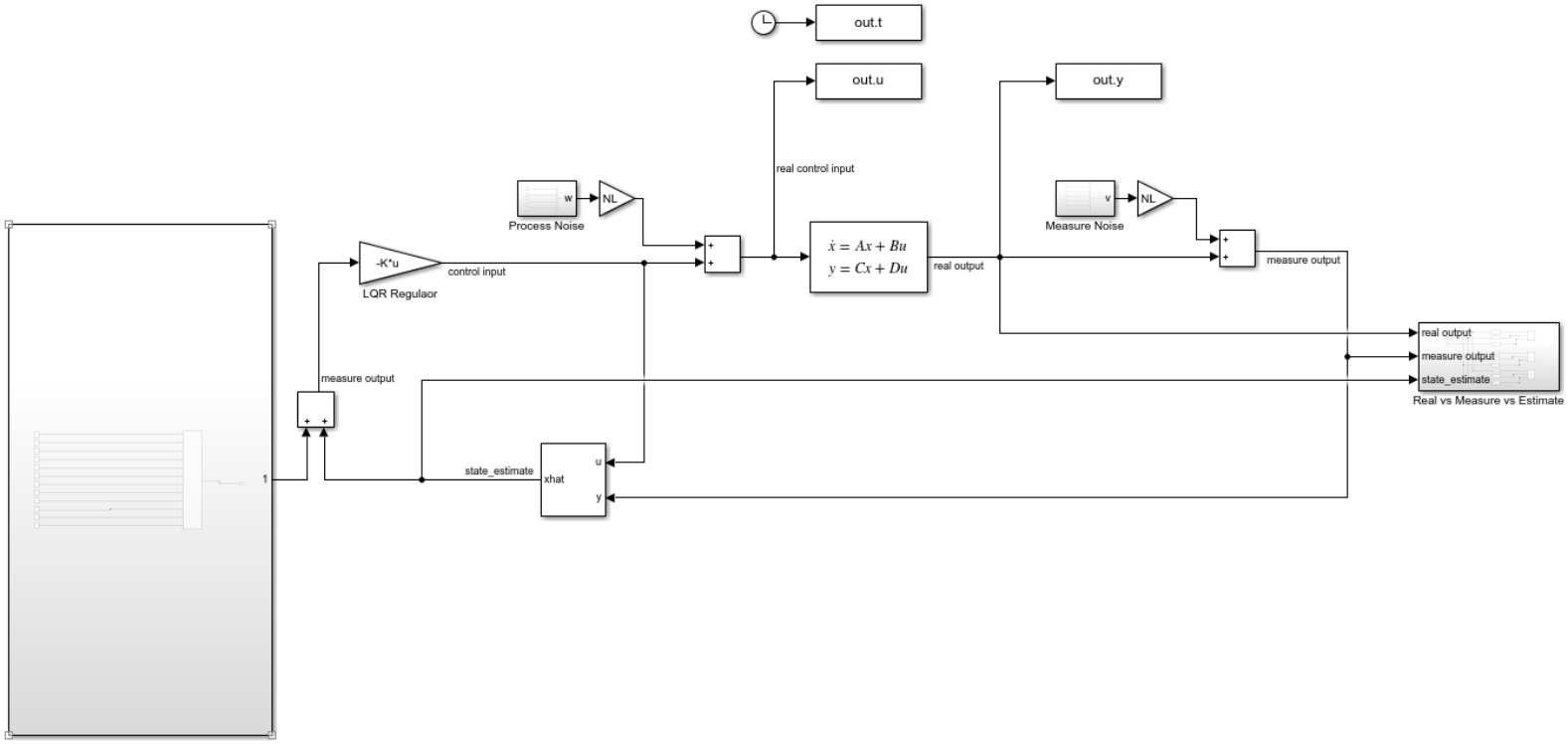


Εικόνα 5.25: Scope X

Το πραγματικό X είναι αυτό με το κίτρινο χρώμα , αυτό που μετράμε είναι η γραμμή με το μπλε χρώμα ενώ η εκτίμηση είναι η γραμμή με το πορτοκαλί χρώμα. Έτσι βλέπουμε ότι με το Kalman Filter γίνεται μία πολύ καλή εκτίμηση του x , κόντα στην πραγματική τιμή που θέλουμε. Το ίδιο συμβαίνει και για το Y,Z.

5.4 LQG-Έλεγχος ύψους αιώρησης (Hovering)

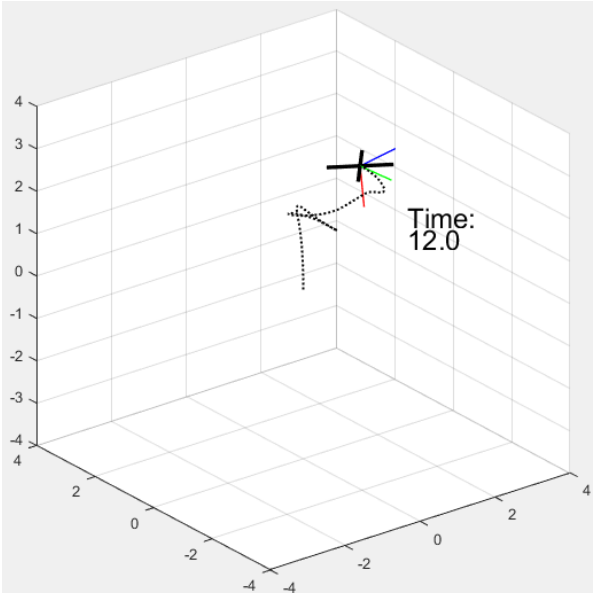
Θα γίνει έλεγχος ύψους αιώρησης με παρόμοιο τρόπο με το LQR ώστε να δούμε πως ανταποκρίνεται το τετρακόπτερο σε συνθήκες θορύβου, και με όχι τόσο έμπιστο μοντέλο. Στην περίπτωση της εξομοίωσης αυτής θα επιλέξουμε λίγο μεγαλύτερή τιμή το θόρυβο στην έξοδο. Θα επιλέξουμε v=0.1. Στη συνέχεια φαίνεται η τροποποίηση του μπλοκ διαγράμματος στο Simulink για να πραγματοποιήσουμε το desired state όπως στο LQR.



Εικόνα 5.26: LQG-Desired State block

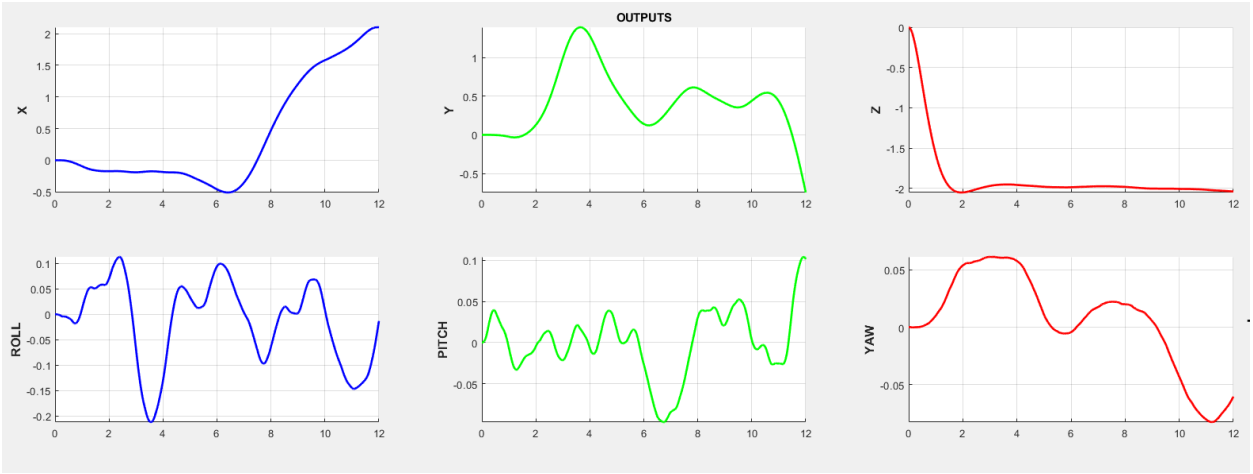
Η εξομοίωση λειτουργεί ακριβώς με τον ίδιο τρόπο που εξηγήσαμε στο κεφάλαιο 5.1 για το desired state μπλοκ.

Έτσι επιλέγοντας z=2 για την απογείωση του τετρακοπτέρου και μηδενικές αρχικές συνθήκες , προκύπτει η διαδρομή που ακολουθεί



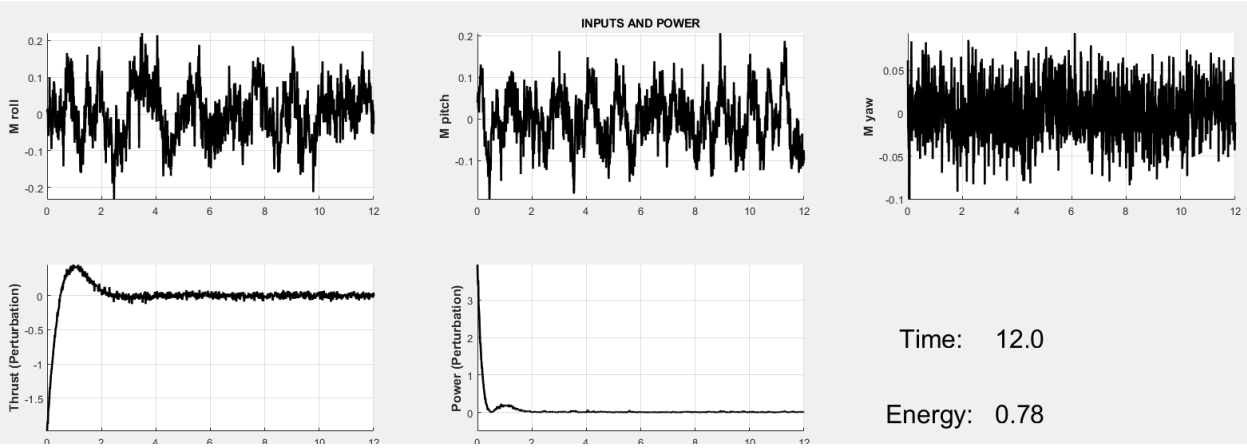
Εικόνα 5.27: Animation μεγάλος θόρυβος εξόδου

Παρατηρείται , η πολύ κακή,αργή και εκτός πορείας διαδρομή του τετρακοπτέρου καθώς δεν μπορεί να ισορροπήσει στο σημείο (0,0,2). Παρακάτω φαίνονται και τα διαγράμματα



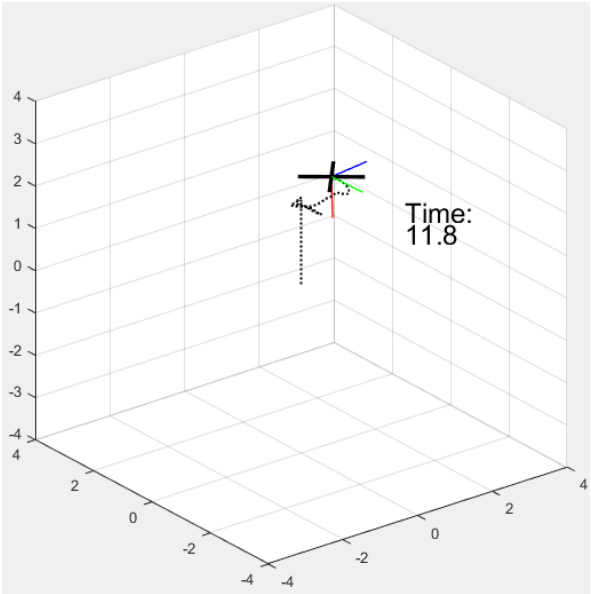
Εικόνα 5.28: Διάγραμμα εξόδου

Αυτό επαληθεύεται και από τα διαγράμματα παραπάνω όπου βλέπουμε πόσο ασταθής είναι το τετρακόπτερο στα διαγράμματα X,Y,Roll,Pitch,Yaw.



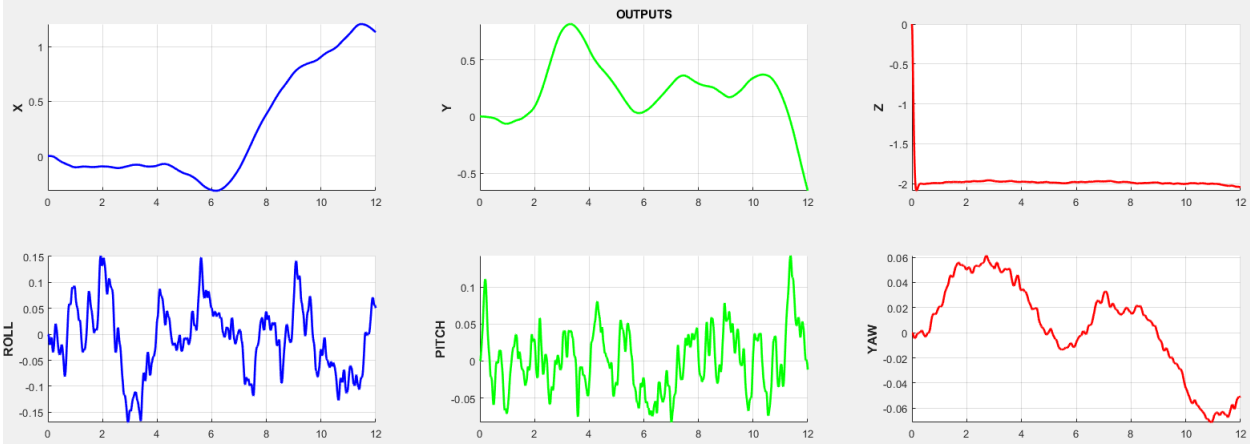
Εικόνα 5.29: Διάγραμμα εισόδου και εκτίμηση ενέργειας

Βλέπουμε ότι η ενέργεια που εκτιμάται δεν είναι μεγάλη και πράγμα που δείχνει και την αργή συμπεριφορά του τετρακοπτέρου στα διαγράμματα. Έτσι παρακάτω θα επιλέξουμε να έχουμε την βέλτιστη γρήγορη απόκριση ρυθμίζοντας το Q.

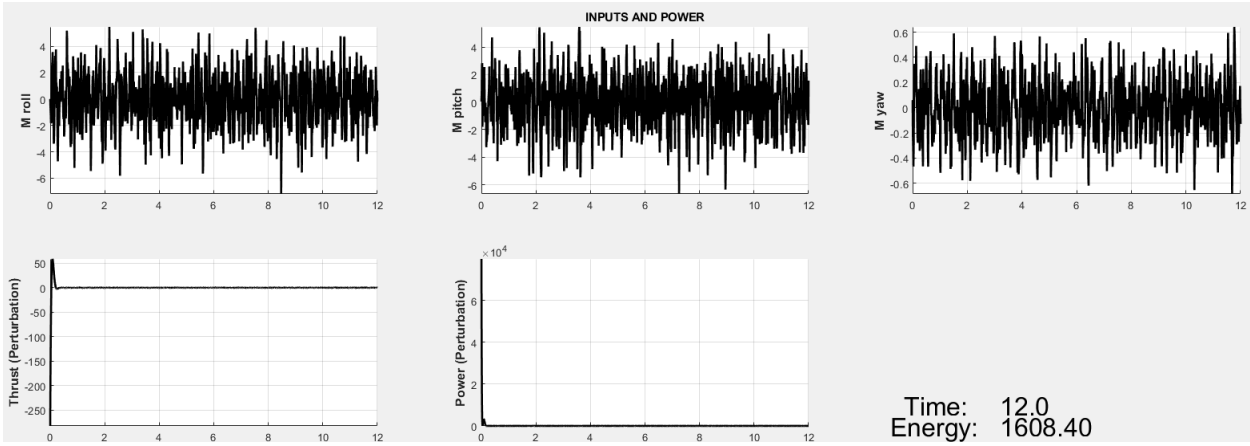


Εικόνα 5.30: Animation επιλογή μεγάλου Q

Παρατηρείται μία πολύ γρήγορη απόκριση όπου φαίνεται και από τα διαγράμματα παρακάτω.

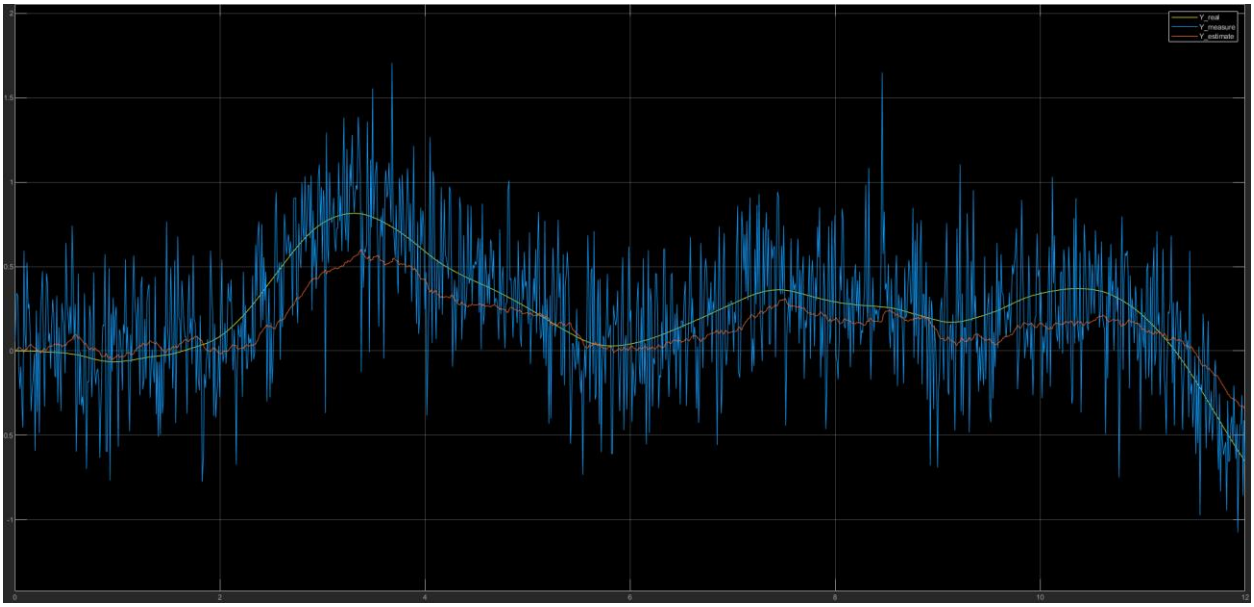


Εικόνα 5.31: Διάγραμμα εξόδου



Εικόνα 5.32: Διάγραμμα εισόδου

Αυτή η πολύ γρήγορη απόκριση έχει οδηγήσει σε πάρα πολλές ταλαντώσεις του τετρακοπτέρου στις γωνίες περιστροφής όπως φαίνεται (Roll,Pitch,Yaw) και μία πάρα πολύ μεγάλη εκτίμηση ενέργειας.

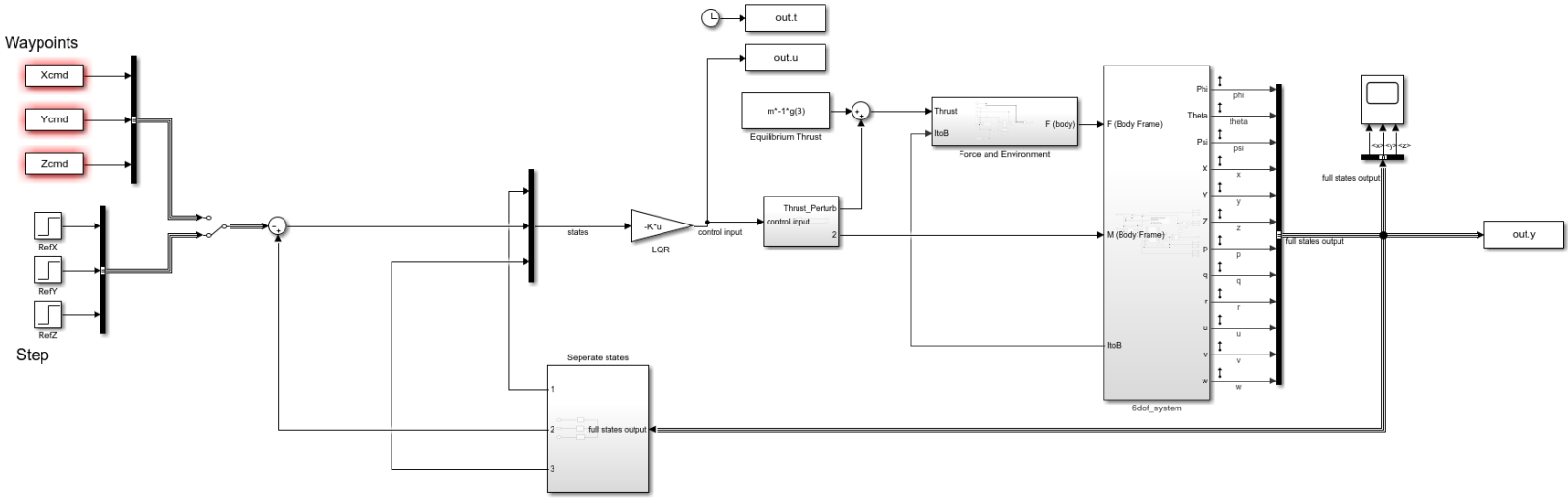


Εικόνα 5.33: Scope Y

Εδώ η εκτίμηση που πραγματοποιείτε (πορτοκαλί γραμμή) στη θέση του τετρακοπτέρου στον άξονα Y φαίνεται πιο ορατά καθώς έχουμε μεγαλύτερο πλάτος θορύβου και βλέπουμε πόσο κοντά είναι η εκτίμηση στην πραγματική έξοδο. Παρόμοια και για τους άξονες X και Y.

5.8 LQR-LQG Εφαρμογή ελέγχου στα μη-γραμμικά μοντέλα

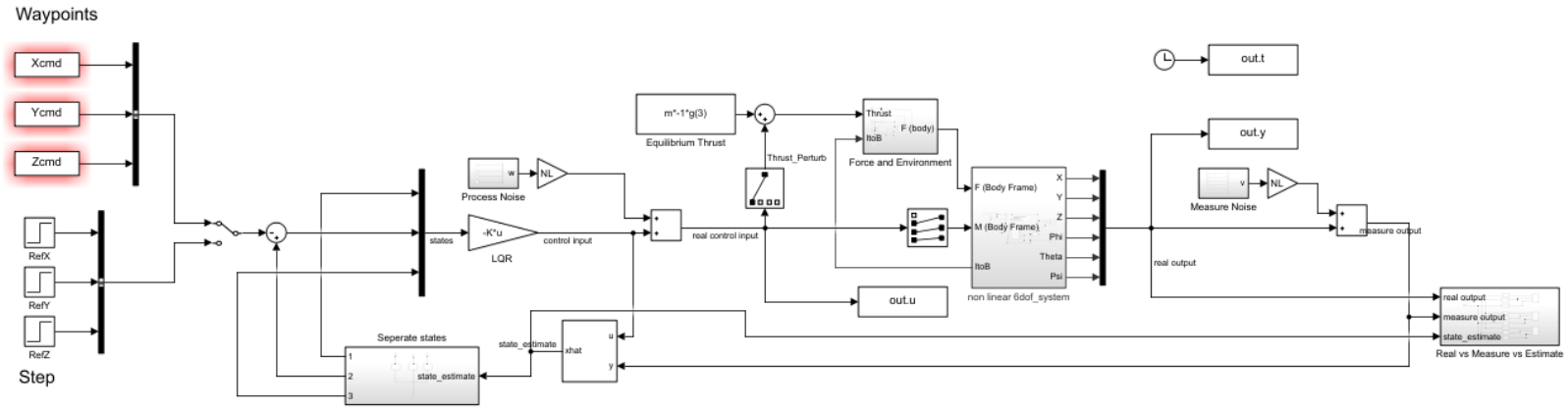
Έπειτα από την σχεδίαση των ελεγκτών LQR αλλά και LQG στα γραμμικά μοντέλα όπως αναλύσαμε παραπάνω , δημιουργήθηκε στο Simulink ένα μπλοκ όπου εφαρμόζουμε τον έλεγχο απευθείας στο μη γραμμικό μοντέλο για να δούμε πως ανταποκρίνεται το σύστημα. Να τονίσουμε εδώ πως ο ελεγκτής LQR σχεδιάστηκε και δοκιμάστηκε στο ‘’ιδανικό’’ γραμμικό μοντέλο όπως είδαμε. Παρακάτω παρουσιάζεται αρχικά ο ελεγκτής LQR στην εφαρμογή του στο μη-γραμμικό μοντέλο.



Εικόνα 5.34:Εφαρμογή LQR στο μη-γραμμικό μοντέλο

Τοποθετούμε τον ελεγκτή K όπου υπολογίζεται από την εντολή LQR του Matlab μπροστά από το μη-γραμμικό μοντέλο. Εισάγουμε ως είσοδο τα X,Y,Z που θέλουμε να ακολουθήσει το τετρακόπτερο είτε με την μορφή βηματικής εισόδου είτε με την μορφή Waypoints όπου στη συνέχεια ο ελεγκτής K θα λάβει τα σφάλματα των 12 μεταβλητών κατάστασης και θα εξάγει τα τέσσερα σήματα ελέγχου για να μηδενίσει αυτά τα σφάλματα.

Με την ίδια λογική εφαρμόστηκε ο ελεγκτής LQG , όπου το Simulink μπλοκ φαίνεται παρακάτω.

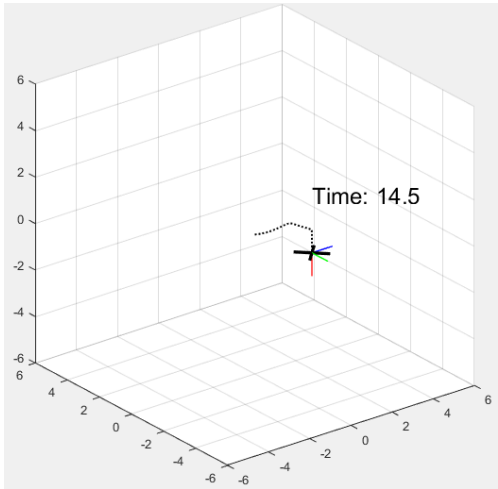


Εικόνα 5.35:Εφαρμογή LQG στο μη-γραμμικό μοντέλο

5.8.1 Βηματική είσοδος

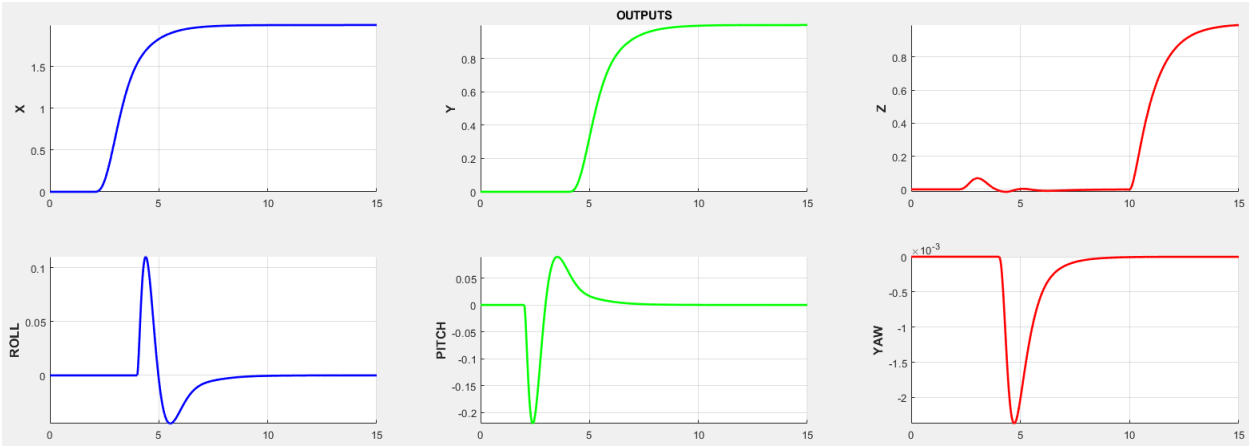
Αρχικά , διατηρώντας τις παραμέτρους για το LQR τους κλασσικούς πίνακες Q και R, προσθέσαμε τυχαίες βηματικές ως είσοδο στο X,Y,Z για να δούμε την απόκριση που θα πάρουμε. Η διαδρομή που ακολουθάει το τετρακόπτερο για τις παρακάτω εισόδους φαίνεται στο animation για χρόνο προσομοίωσης t=15sec.

- Βηματική στο X τη χρονική στιγμή t=2 sec με πλάτος 2
- Βηματική στο Y τη χρονική στιγμή t=4 sec με πλάτος 1
- Βηματική στο Z τη χρονική στιγμή t=10 sec με πλάτος 1

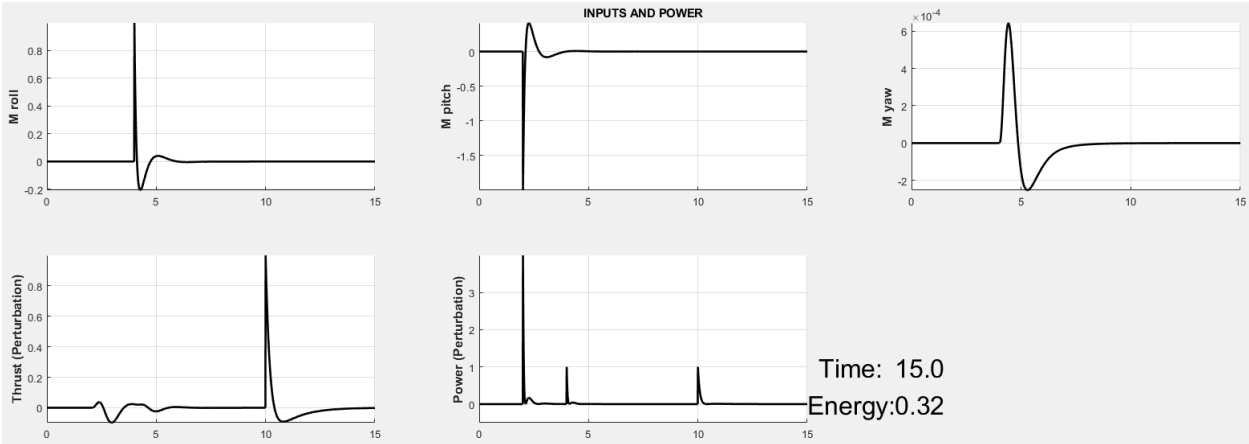


Εικόνα 5.36:LQR-Animation Βηματική

Παρακάτω φαίνονται τα διαγράμματα



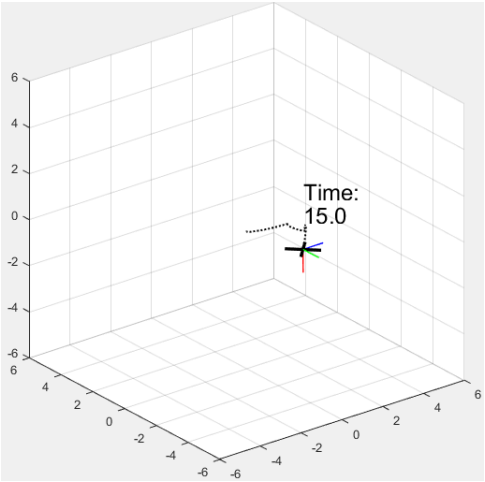
Εικόνα 5.37: Έξοδοι-Είσοδος Βηματική



Εικόνα 5.38:Είσοδοι-Είσοδος Βηματική

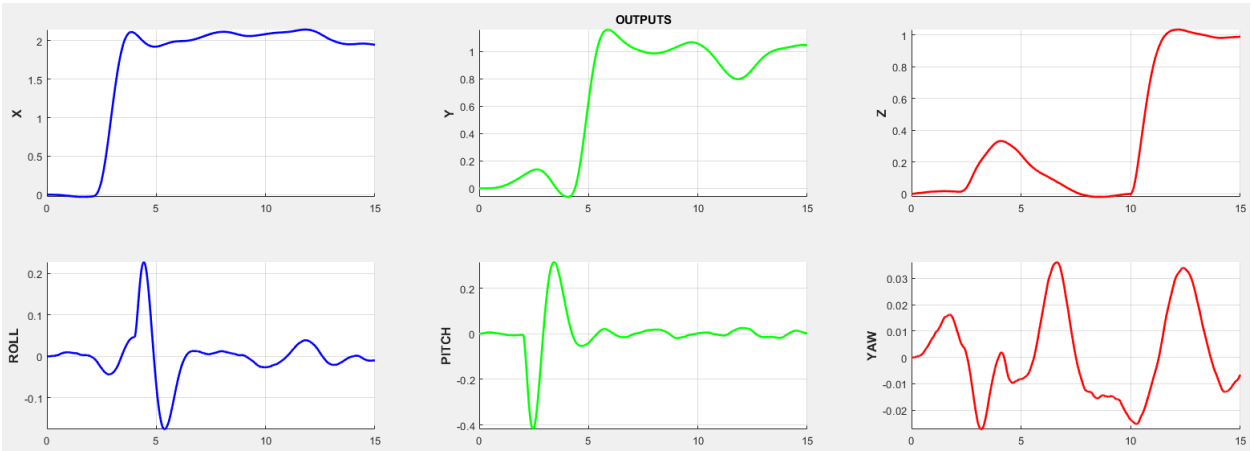
Παραπάνω βλέπουμε τις αποκρίσεις για το τετρακόπτερο με είσοδο βηματική .

Στην συνέχεια θα γίνει η ίδια εξομοίωση για το LQG με v=0.0001 και w=0.001.

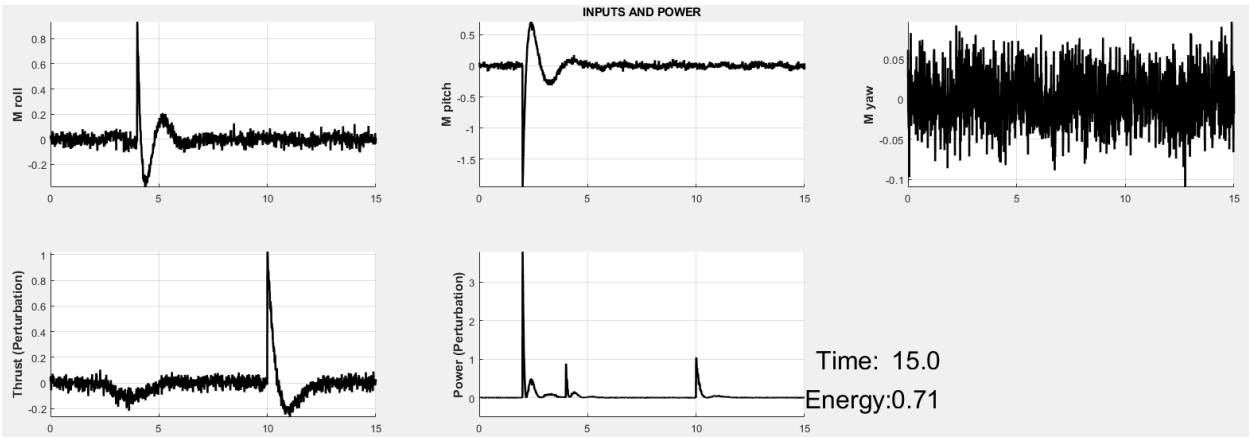


Εικόνα 5.39: LQG-Animation Βηματική

Παρατηρείται πως το LQG αντιπροσωπεύει πιο πολύ την πραγματικότητα από το LQR. Παρακάτω φαίνονται και τα διαγράμματα



Εικόνα 5.40:Έξοδοι-Είσοδος Βηματική



Εικόνα 5.41:Είσοδοι-Είσοδος Βηματική

5.8.2 Waypoints path

Αναπτύχθηκε κώδικας , όπου μπορούμε να καθορίσουμε την τροχιά που θα ακολουθήσει το τετρακόπτερο μέσω waypoints που εισάγονται στο πρόγραμμα από ένα mat-file. Εισάγουμε τις τιμές X,Y,Z που θα έχει το τετρακόπτερο τον εκάστοτε χρόνο t στο mat-file η ακόμα και μέσα από excel. Στην συνέχεια εισάγονται ως ορίσματα στον κώδικα. Οι πρώτες δύο γραμμές του mat-file ορίζουν την απογείωση του τετρακοπτέρου ενώ τις υπόλοιπες τις επιλέγουμε με βάση την τροχιά που θέλουμε να ακολουθήσει.

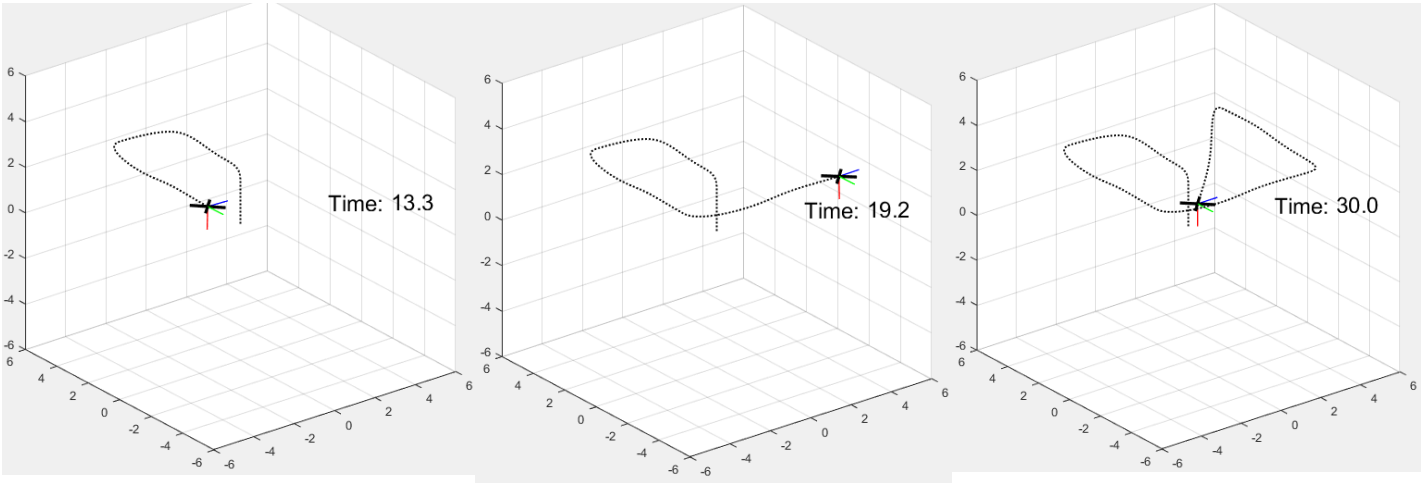
	1	2	3	4
1	0	0	0	1
2	0	0	-3	2
3	0	-4	-3	3
4	-4	-4	-3	7
5	-4	1	-3	9
6	-4	4	-3	13
7	1	4	-3	15
8	4	4	-3	19
9	4	-1	-3	21
10	4	-4	-3	25
11	2	-1	-2	26
12	0	0	0	30
13	0	0	0	32

Εικόνα 5.42: Waypoints X,Y,Z,t

Οι πρώτες τρείς στήλες είναι τα X,Y,Z ενώ η τέταρτη είναι ο χρόνος προσομοίωσης , άρα για χρόνο προσομοιώσεις ts=13sec το τετρακόπτερο θα βρίσκεται στη θέση X=-4 , Y=4 ,Z=-3. Στη συνέχεια θα γίνει προσομοίωση του συστήματος με στιγμιότυπα το συγκεκριμένο χρόνο για να φανεί η θέση που παίρνει το τετρακόπτερο. Θα γίνει με τον ελεγκτή LQR για τους κλασσικούς πίνακες Q και R.

Παρακάτω φαίνεται η τροχιά του τετρακοπτέρου για τον πίνακα της εικόνας 5.42 όπου έχουμε για t_1=13sec , t_2=19sec, t_3=30sec.

T_1 **T_2** **T_3**



Εικόνα 5.43: Τροχιά τετρακοπτέρου

Παρατηρείται από τα animations πως η τροχιά του τετρακοπτέρου για τις χρονικές στιγμές T_1,T_2,T_3 επαληθεύει την τροχιά που έχουμε στην εικόνα 5.42.

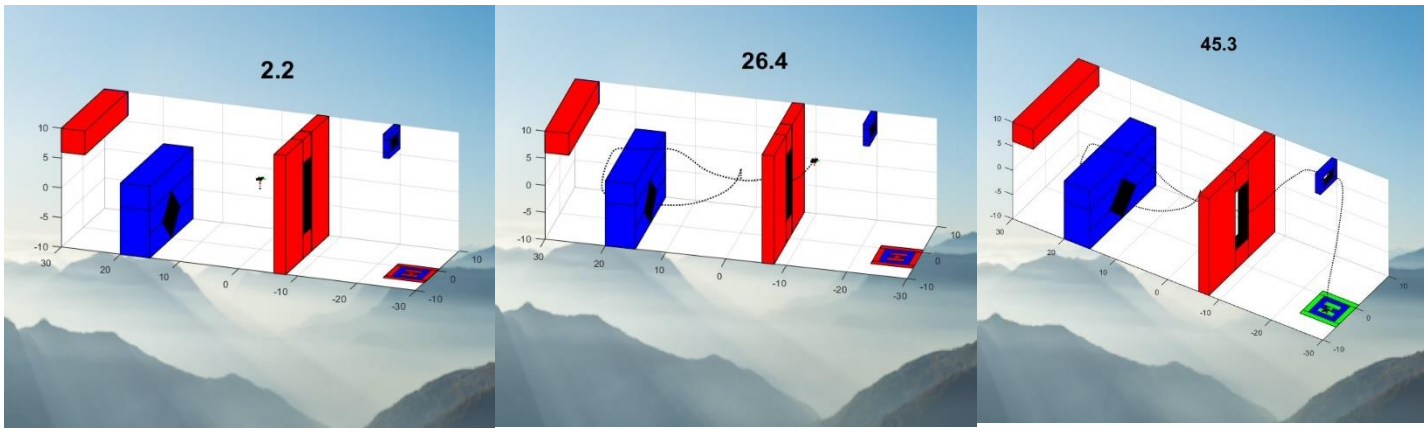
5.9 Εμπόδια 3D Animations

Στο κεφάλαιο αυτό , πραγματοποιείτε η δημιουργία πιο ρεαλιστικών animations ώστε να μπορέσουμε να δοκιμάσουμε τους δύο ελεγκτές. Το τετρακόπτερο και στις δύο περιπτώσεις θα ακολουθεί τη βέλτιστη τροχιά μέσω Waypoints . Τα πρώτα δύο plots έχουν να κάνουν με την τοποθέτηση εμποδίων στο χώρο, ώστε το τετρακόπτερο να πετύχει την αποφυγή τους ενώ στο τρίτο plot, γίνεται αναγνώριση σε πόσο χρόνο το τετρακόπτερο καταφέρνει να φτάσει στο κάθε σημείο με εναλλαγή χρώματος του σημείου. Αναπτύχθηκε κώδικας , με την βοήθεια της εντολής fill3 του Matlab, όπου δημιουργήθηκαν πίνακες για τα X,Y,Z που επιθυμούμε και μπορούμε μέσω αυτής της εντολής να σχεδιάσουμε πολύγωνα. Παράλληλα τοποθετήθηκαν σε ορισμένα και κάποιες φωτογραφίες για τα backgrounds ώστε να έχουμε πιο καλή απόδοση της πραγματικότητας. Στην συνέχεια πραγματοποιήθηκαν τα animation στο μη-γραμμικό μοντέλο για τον LQR και τον LQG ελεγκτή για το κάθε animation ξεχωριστά.

- <u>Animation- Αποφυγή εμποδίων περνώντας από μέσα</u>

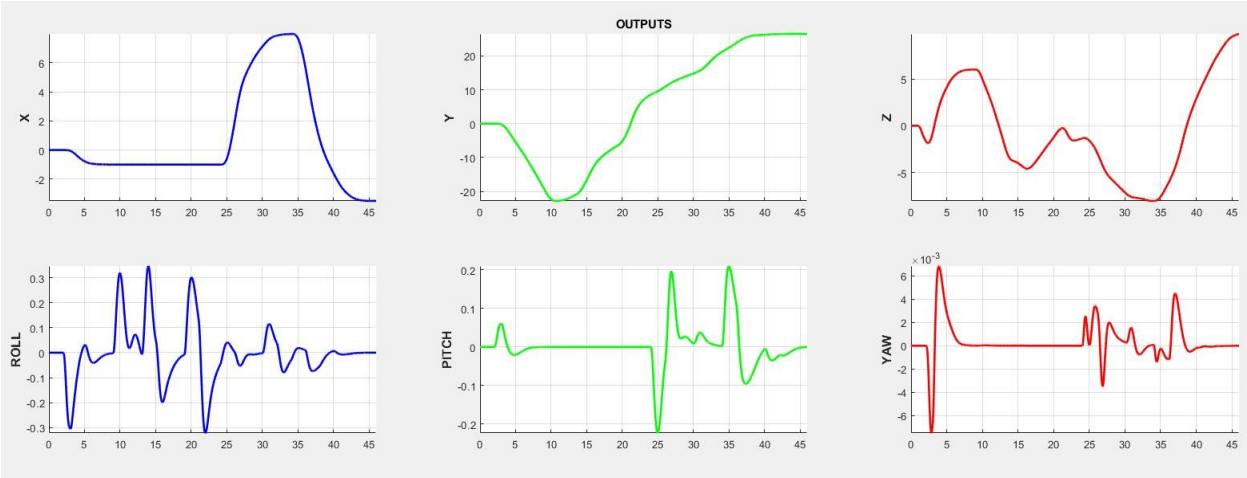
Η εξομοίωση πραγματοποιήθηκε πρώτα για τον LQR ελεγκτή.

t=2.4 **t=26.4** **t=45.3**

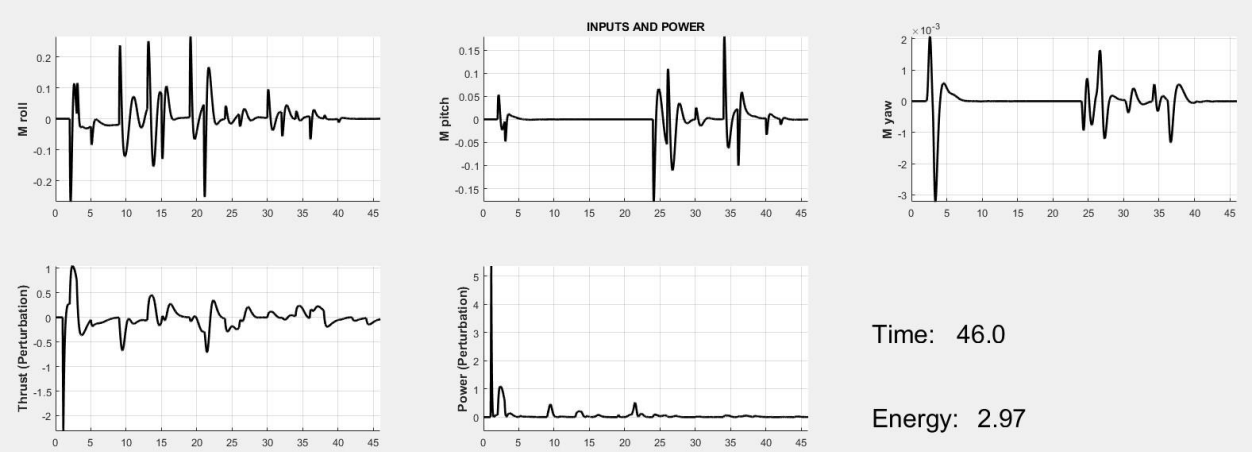


Εικόνα 5.44: Animation LQR-εμπόδια

Στην εικόνα 5.44 βλέπουμε την τροχιά που ακολουθάει το τετρακόπτερο για τα συγκεκριμένα εμπόδια που έχουν τοποθετηθεί στο plot. Παρατηρούμε ότι για χρόνο 2.4 το τετρακόπτερο απογειώνεται από την θέση (0,0,0) ενώ σε χρόνο t=26.4 έχει περάσει μεγάλο τετράγωνο με την τρύπα στην μέση και κατευθύνεται για το τελευταίο σημείο. Τέλος, καταλήγει στην προσγείωση , όπου όταν φτάνει πρασινίζει η ένδειξη προσγείωσης. Παρακάτω παρουσιάζονται και τα διαγράμματα εξόδου-εισόδου αλλά και η εκτίμηση ενέργειας.

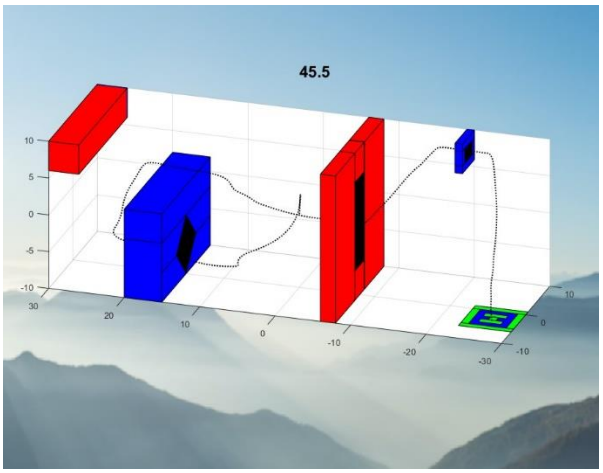


Εικόνα 5.45: Διάγραμμα εξόδου

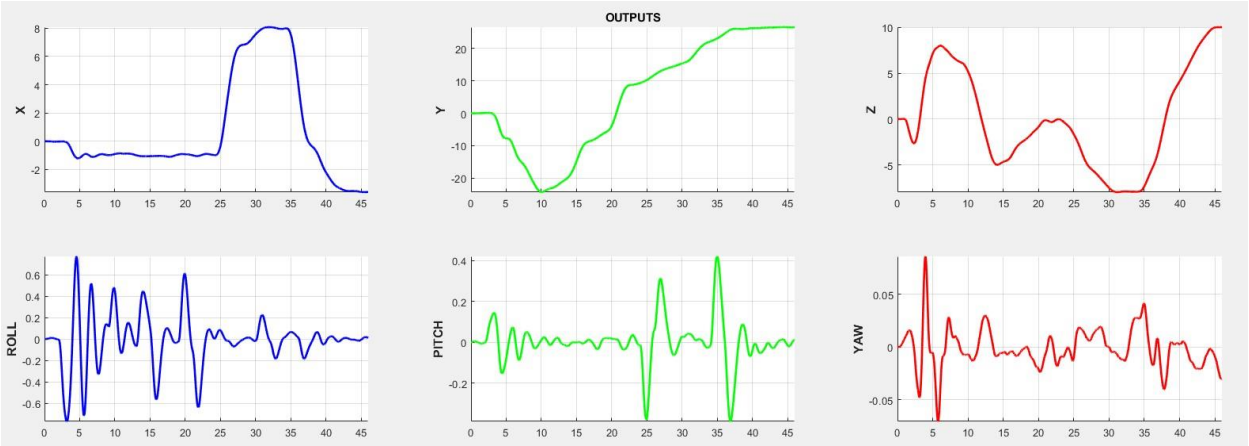


Εικόνα 5.46: Διάγραμμα εισόδου-εκτίμηση ενέργειας

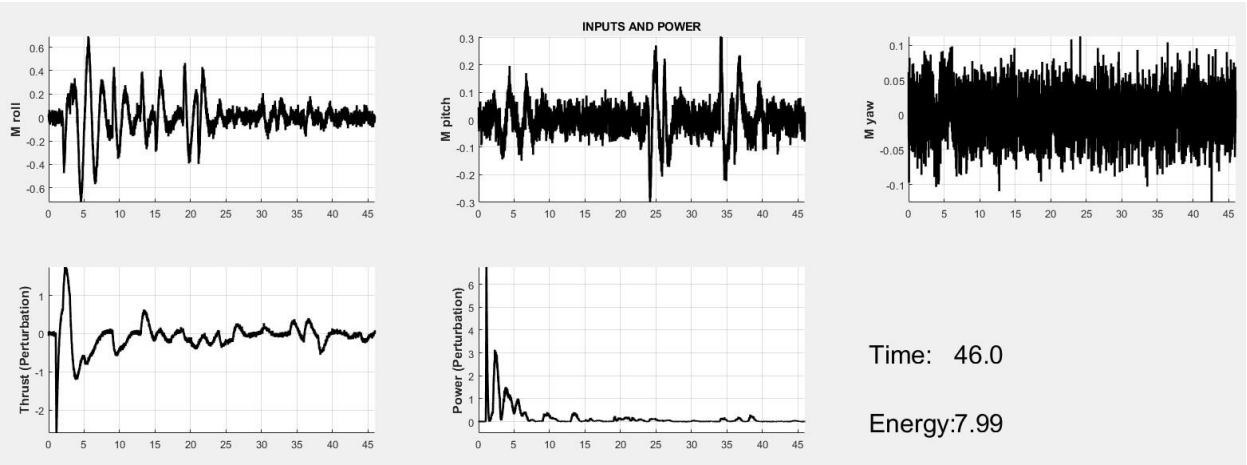
Η εξομοίωση θα πραγματοποιηθεί για τον LQG ελεγχτή



Εικόνα 5.47: Animation LQG-εμπόδια



Εικόνα 5.48: Διάγραμμα εξόδου



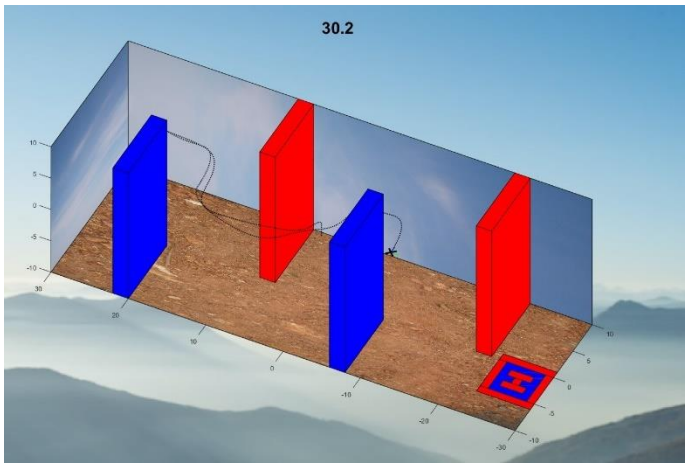
Εικόνα 5.49: Διάγραμμα εισόδου-εκτίμηση ενέργειας

Παρατηρείται πως ενώ η τροχιά που ακολουθάει το τετρακόπτερο είναι η ίδια , στην περίπτωση του LQG, ο θόρυβος επηρεάζει τη τροχιά του τετρακοπτέρου και η ενέργεια που απαιτείται είναι μεγαλύτερη. Το συμπεραίνουμε και από τα διαγράμματα που φαίνονται παραπάνω.

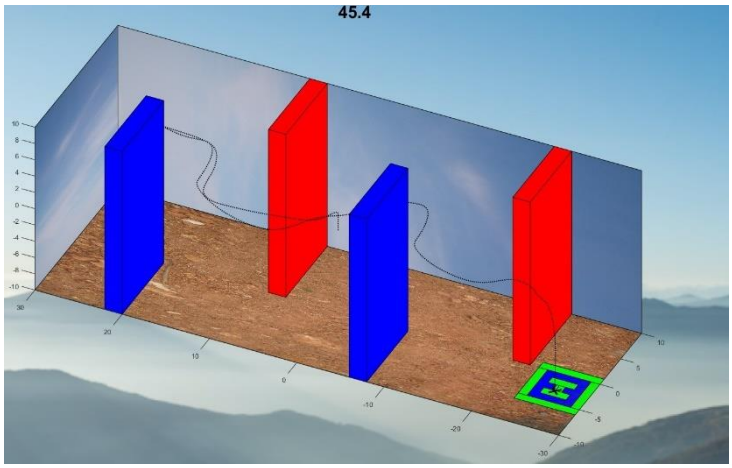
- <u>Animation- Αποφυγή Εμποδίων δεξιά-αριστερά</u>

Στο συγκεκριμένο animation τοποθετούνται εμπόδια δεξιά-αριστερά , και το τετρακόπτερο ακολουθάει την διαδρομή που χρειάζεται για να φτάσει στο σημείο προσγείωσης. Παρατηρείται στο animations πως έχουν προστεθεί backgrounds για να γίνει πιο ρεαλιστικό το plot.

Η εξομοίωση πραγματοποιήθηκε πρώτα για τον LQR ελεγκτή.

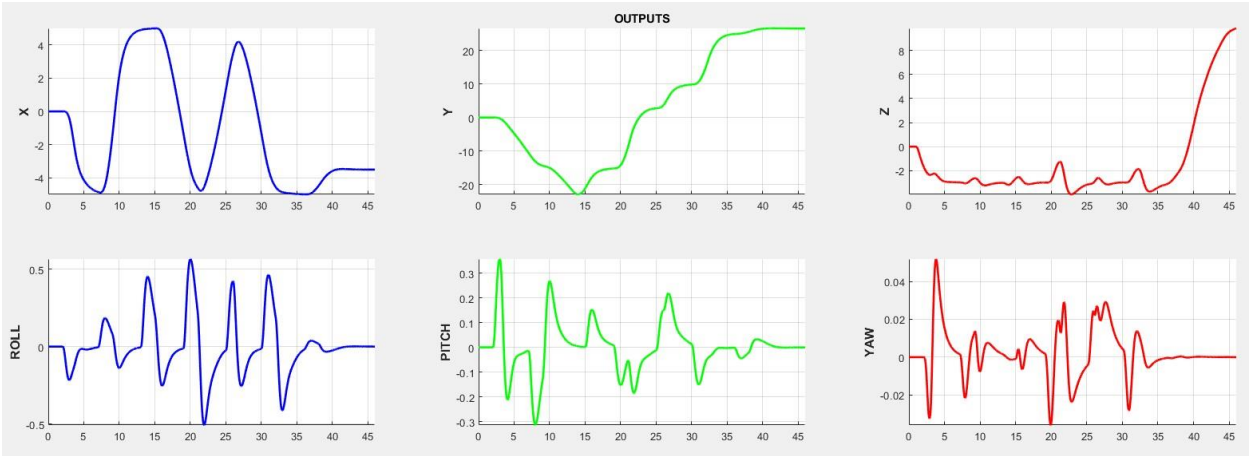


Εικόνα 5.50: Τροχιά τετρακοπτέρου -LQR

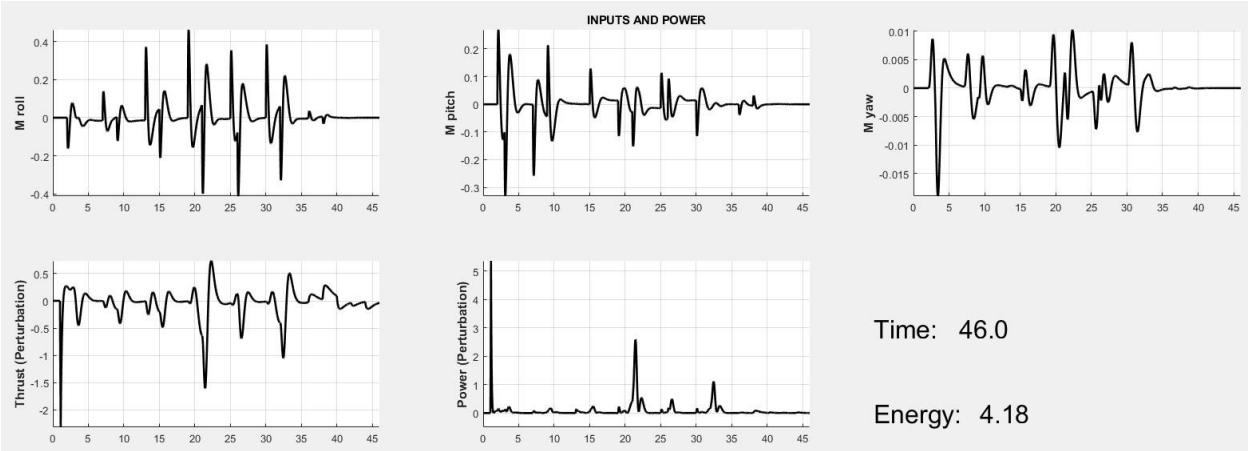


Εικόνα 5.51: Προσγείωση τετρακοπτέρου

Βλέπουμε πως εδώ συγκεκριμένα το τετρακόπτερο αναγκάζεται να επιχειρήσει πολλές μεταβολές στο roll και στον άξονα x για να φτάσει στο σημείο προσγείωσης

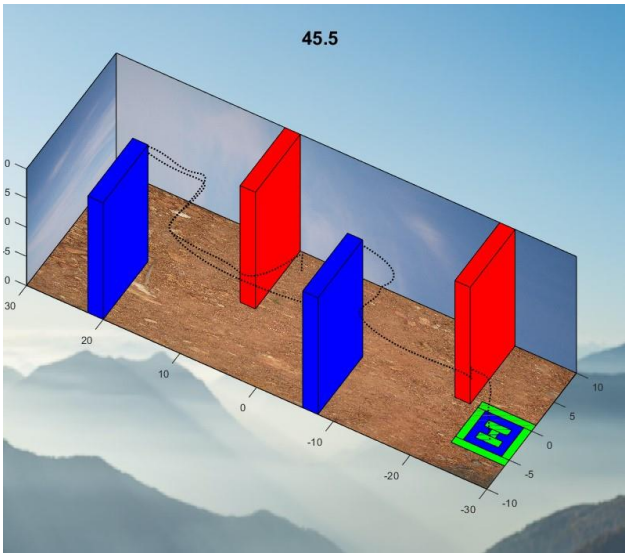


Εικόνα 5.52: Διάγραμμα εξόδου

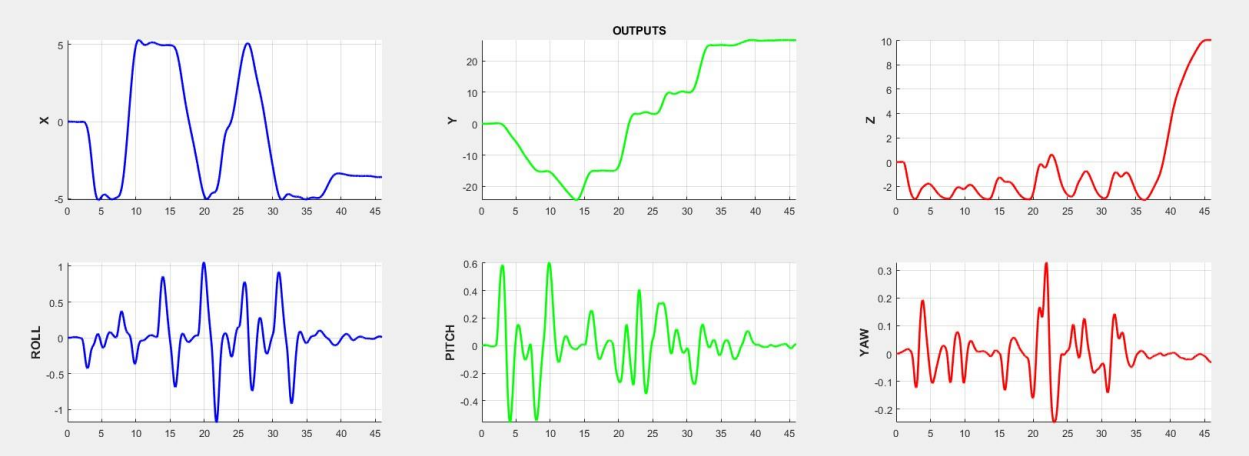


Εικόνα 5.53: Διάγραμμα εισόδου και εκτίμηση ενέργειας

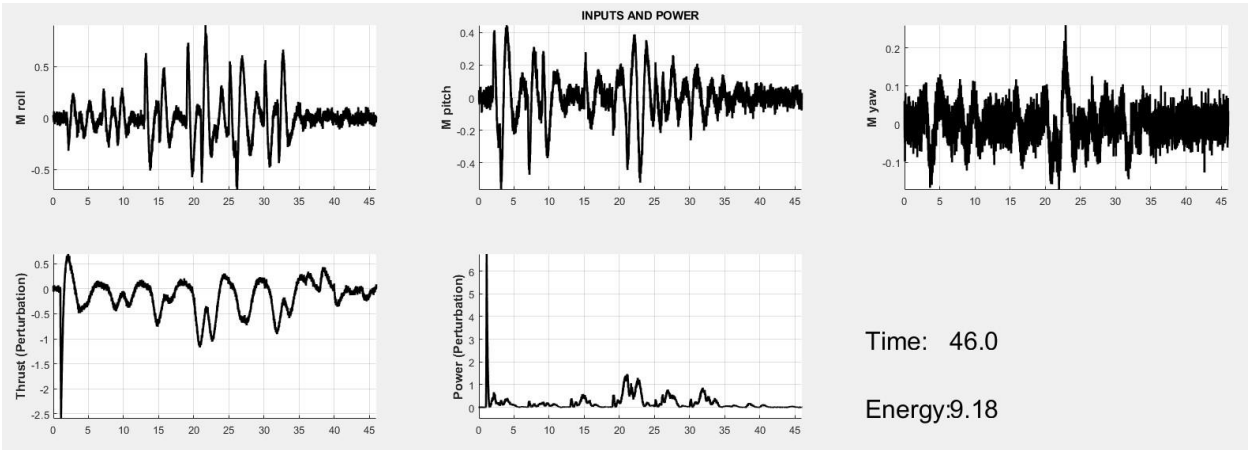
Η εξομοίωση θα πραγματοποιηθεί για τον LQG ελεγχτή



Εικόνα 5.54: Τροχιά τετρακοπτέρου-LQG



Εικόνα 5.55: Διάγραμμα εξόδου

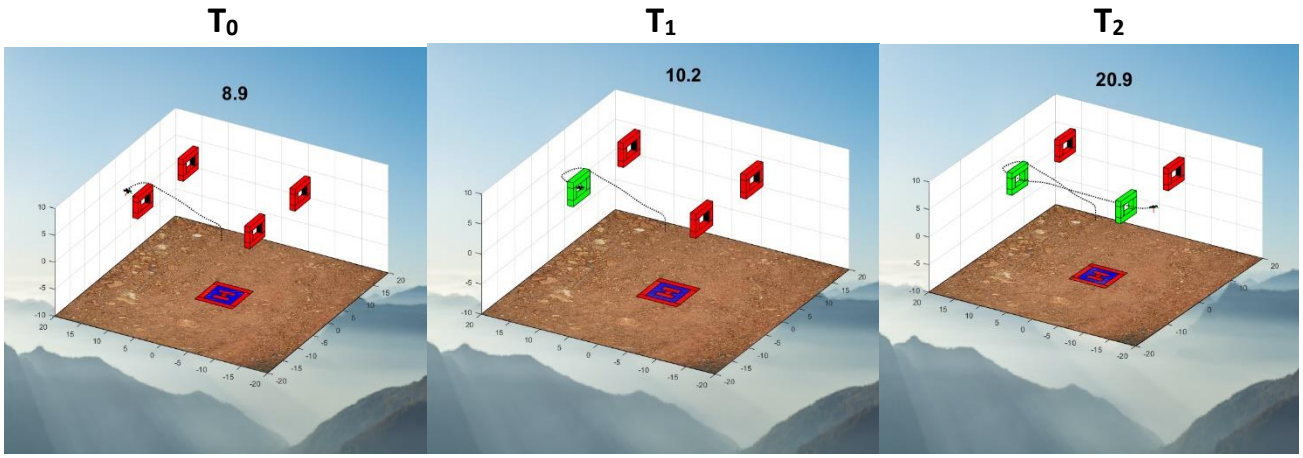


Εικόνα 5.56: Διάγραμμα εισόδου και εκτίμηση ενέργειας

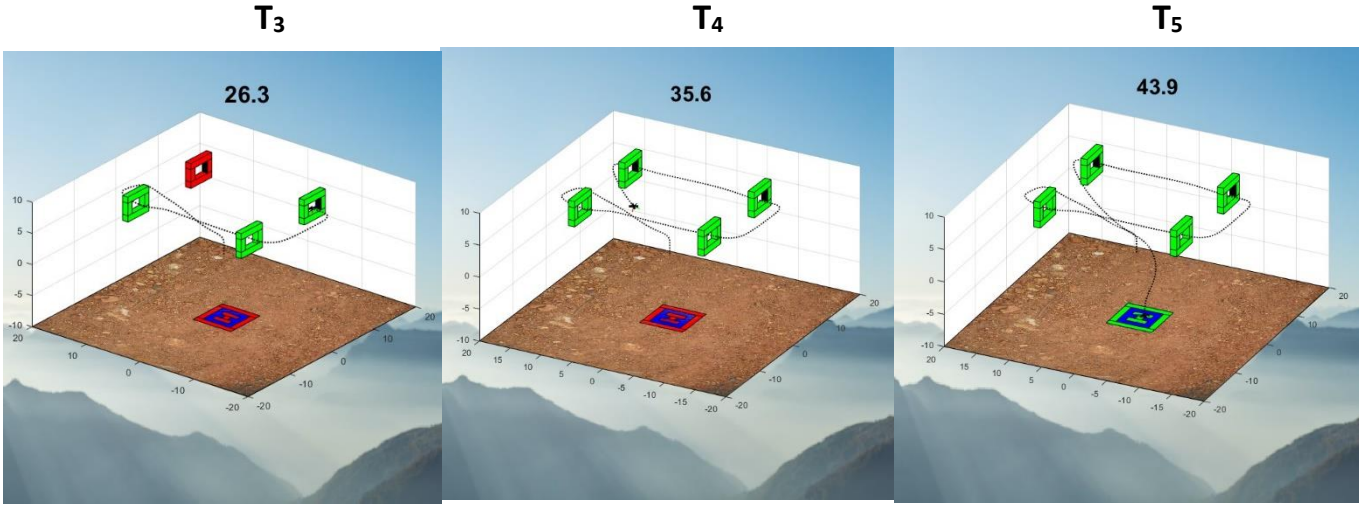
Παρατηρείται πως η επίδραση θορύβου αλλά και η αβεβαιότητα του μοντέλου απαιτεί τουλάχιστον διπλάσια ενέργεια.

- <u>Animation- Υπολογισμός χρόνο άφιξης σε σημείο</u>

Το τελευταίο animation , έχει να κάνει με το χρόνο άφιξης του τετρακοπτέρου σε συγκεκριμένο σημείο. Έτσι υπολογίζει το χρόνο όπου θα περάσει από το σημείο που έχει οριστεί και τότε αναγνωρίζει το τετρακόπτερο και κάνει εναλλαγή χρώματος. Παρακάτω φαίνεται για χρόνο που έχει οριστεί στα t_1=10sec, t_2=20, t_3=26,t_4=35,t_5=43.

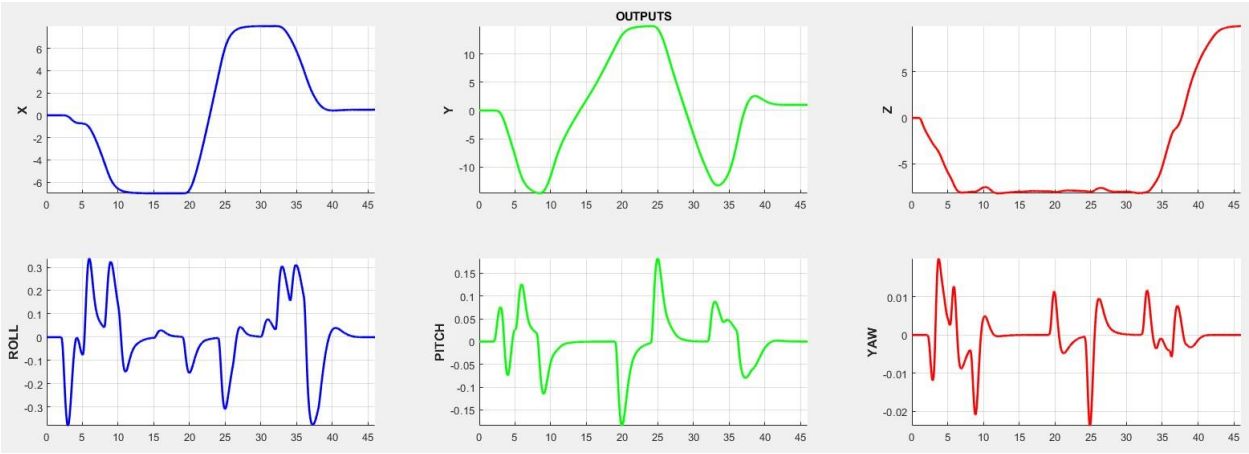


Εικόνα 5.57: Animation υπολογισμός χρόνος άφιξης

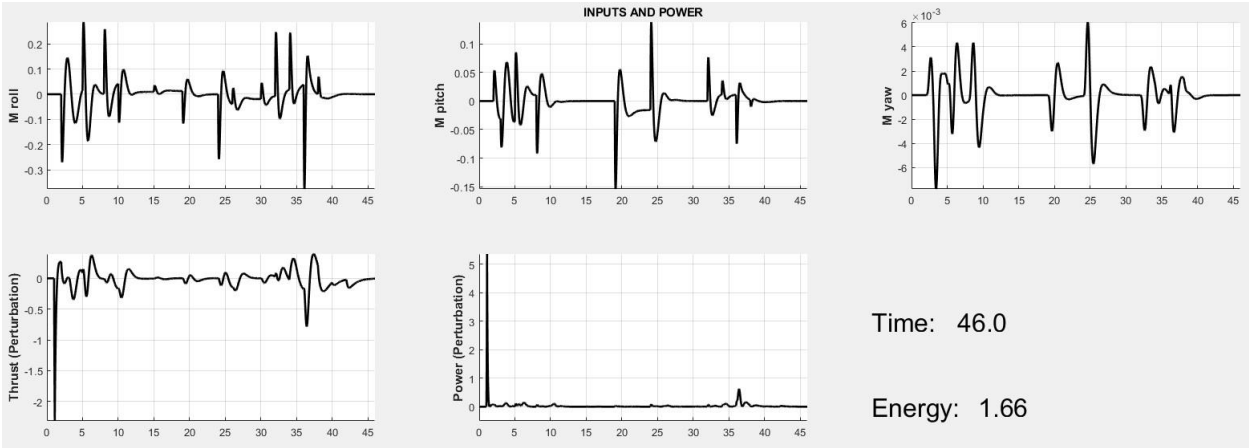


Εικόνα 5.58: Animation υπολογισμός χρόνος άφιξης

Παρακάτω φαίνονται και τα διαγράμματα



Εικόνα 5.59: Διάγραμμα εξόδου

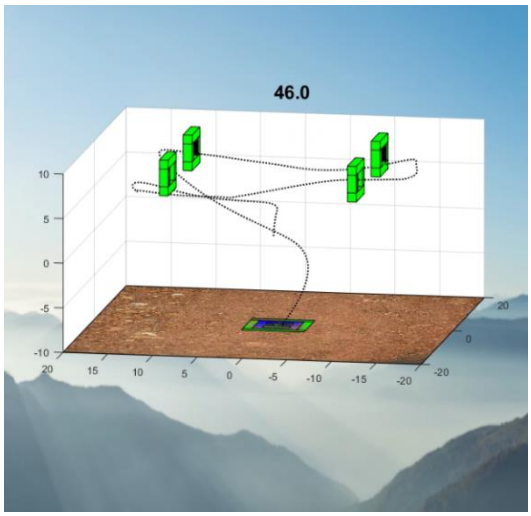


Εικόνα 5.60: Διάγραμμα ειόδου-εκτίμηση ενέργειας

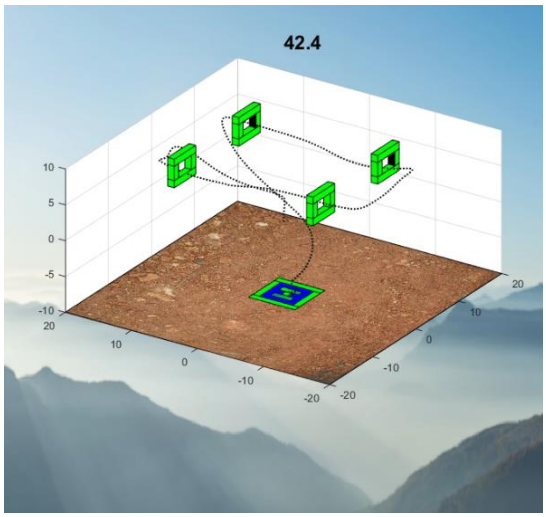
Επίσης ,παραπάνω δοκιμάστηκε να πετύχουμε μέγιστο κόστος απόδοσης , αλλά η απόκριση ήταν ναι μεν γρήγορη αλλά η διαδρομή ήταν αρκετά ασταθής.

Η εξομοίωση θα πραγματοποιηθεί για τον LQG ελεγκτή

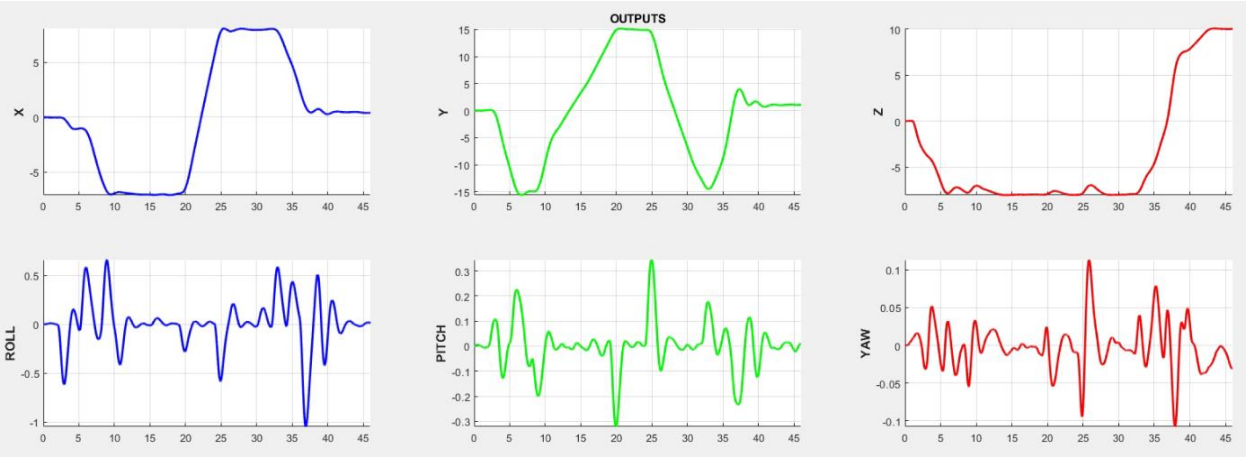
Πραγματοποιήθηκε εφαρμόζοντας διαφορετικό θόρυβο στις μετρήσεις κάθε φορά για να δούμε την απόκριση του τετρακοπτέρου.



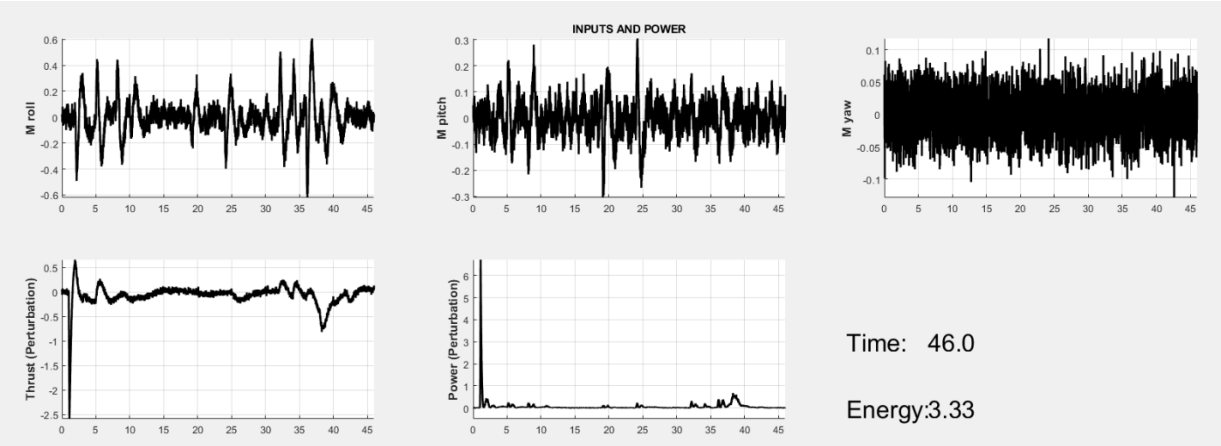
Εικόνα 5.61: LQG-θόρυβος μετρήσεων 0.0001



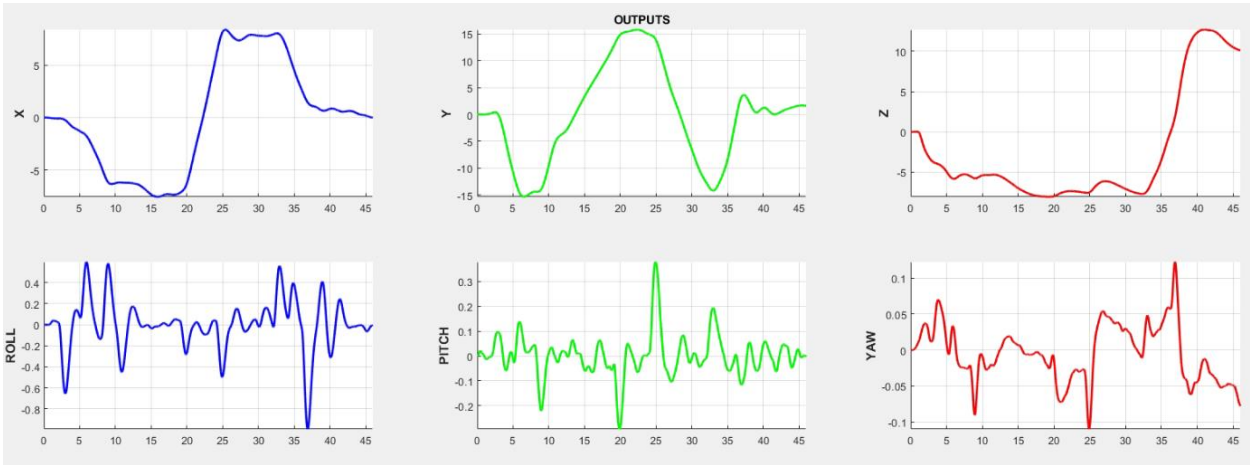
Εικόνα 5.62: LQG-θόρυβος μετρήσεων 0.01



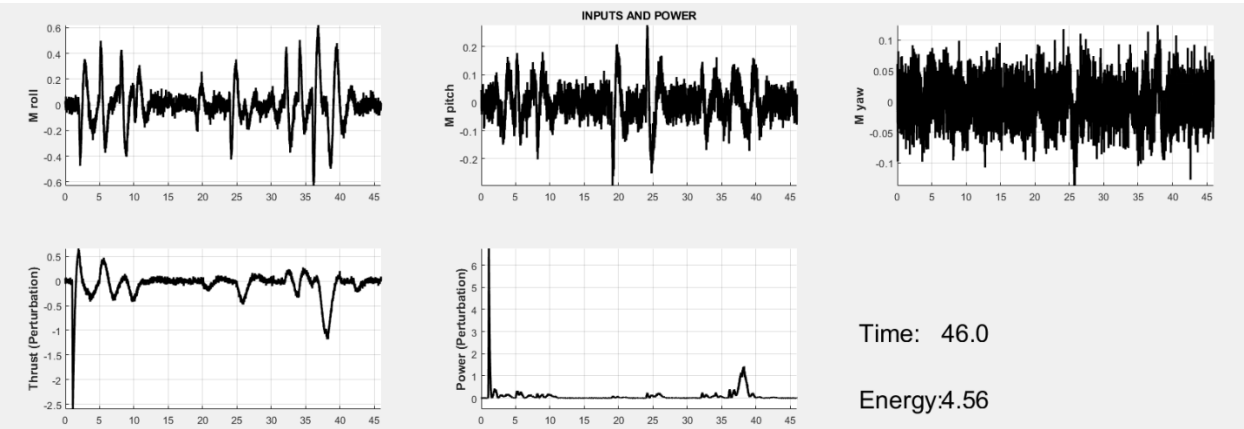
Εικόνα 5.63: Διάγραμμα εξόδου με w=0.0001



Εικόνα 5.64: Διάγραμμα εξόδου με w=0.0001



Εικόνα 5.65: Διάγραμμα εξόδου με w=0.01



Εικόνα 5.66: Διάγραμμα εισόδου με w=0.01

Βλέπουμε πως όταν αυξάνεται ο θόρυβος έχουμε και πιο μεγάλη αστάθεια όπως φαίνεται απο τις καμπύλες τις εξόδου αλλά απαιτείται και μεγαλύτερη ενέργεια για να ανταποκριθεί το σύστημα.

6. ΚΩΔΙΚΑΣ ΚΑΙ GIFS

Παρακάτω φαίνεται το link για τον κώδικα που δημιουργήθηκε αλλα και τα Simulink models που χρησιμοποιήθηκαν για την προσομοίωση. Επίσης μέσα στο github παρουσιάζονται και μερικά gifs για να δούμε τα animations την κίνηση που πραγματοποιεί το τετρακόπτερο.

Link : **https://github.com/chrysostomos997/Quadcopter.git**

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια της παρούσας εργασίας, γίνεται παράθεση κάποιων συμπερασμάτων τόσο σε θεωρητικό υπόβαθρο όσο και σε συμπεράσματα που προκύπτουν με βάση τα συστήματα που εξετάστηκαν και τους ελεγκτές (LQR & LQG).

Αρχικά, προκύπτει ένα μαθηματικό μοντέλο (κινηματική και δυναμική) που περιγράφεται ορθά μέσα από τις εξισώσεις του νόμου του Νεύτωνα και τις εξισώσεις των γωνιών Euler για την περιστροφική και μεταφορική κίνηση. Έπειτα από γραμμικοποιήση των δύο διαφορετικών μη-γραμμικών μοντέλων μέσω Simulink , καθώς για το LQG η γραμμικοποίηση έγινε με λιγότερες εξόδους (αβεβαιότητα μοντέλου), εφαρμόστηκαν οι ελεγκτές στα γραμμικά μοντέλα και πραγματοποιήθηκαν προσομοιώσεις και 3D animations. Στη συνέχεια ο έλεγχος εφαρμόστηκε απευθείας στα μη-γραμμικά μοντέλα και δημιουργήθηκαν ρεαλιστικά animations δημιουργώντας χάρτες (plots) εμποδίων με επιλογή τροχιάς του τετρακοπτέρου. Στις προσομοιώσεις , παρατηρήθηκαν ορισμένα αποτελέσματα όπου τα συνοψίζουμε παρακάτω :

- Αλλάζοντας τα βάρη του κόστους απόδοσης (Q) και κόστους ελέγχου (R) στον ελεγκτή LQR , πραγματοποιήθηκε έλεγχος και βελτιστοποίηση της κατανάλωσης ενέργειας αλλά και τις απόδοσης.
- Παρατηρήθηκε επίσης πόσο σημαντική και ικανοποιητική είναι η εφαρμογή του φίλτρου Kalman στον ελεγκτή LQG όταν έχουμε παρουσία θορύβου στις μετρήσεις ή αβέβαιο μοντέλο .
- Ένα από τα μειονεκτήματα του LQR είναι ότι απαιτεί πολύ πληροφορία (‘ιδανικό μοντέλο’).
- Ο έλεγχος των εισόδων στην τεχνική LQG είναι ο ίδιος αν γνωρίζαμε τις μεταβλητές όλων των αγνώστων (εκτιμήσεις καταστάσεων).
- Η χρήση του LQR δίνει πολύ καλύτερη απόκριση (‘ιδανικές συνθήκες’)
- Ο ελεγκτής LQG ανταποκρίνεται πολύ καλύτερα στην πραγματικότητα (παρουσία θορύβου, αβεβαιότητα μοντέλου)
- Πραγματοποιήθηκε και επιτεύχθηκε ευστάθεια για έλεγχο ύψους αιώρησης και γωνίας περιστροφής.

Η παρούσα εργασία , ως έρευνα πάνω στο αντικείμενο ενδιαφέροντος που είναι το τετρακόπτερο εμπλούτισε τις γνώσεις μου και με βοήθησε στην αναζήτηση πηγών και μεθόδων ελέγχου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Draganfly, Inc. 2009. [online] A Short History of Unmanned Aerial Vehicles (UAVs), by Cam Tetrault

[2] Z. Tahir, M. Jamil, S. A. Liaqat, L. Mubarak, Waleed Tahir and Syed Omer Gilani, Design and Development of Optimal Control System for Quad Copter UAV, Indian Journal of Science and Technology, Vol 9(25), 10.17485/ijst/2016/v9i25/96611, July 2016.

[3] F. Sabatino, "Quadrotor control: modeling, nonlinearcontrol design, and simulation", 2015.

[4] Saraf, Prathamesh & Gupta, Manan & Parimi, Alivelu. (2020). A Comparative Study Between a Classical and Optimal Controller for a Quadrotor. 1-6. 10.1109/INDICON49873.2020.9342485.

[5] Muhammad Usman, “Quadcopter Modelling and Control with MATLAB/Simulink Implementation

[6] R. Dorf, Modern Control Systems.

[7] Faraz Ahmad*, Pushpendra Kumar , Anamika Bhandari,Pravin P.Patil ,Simulation of the quadcopter Dynamics with LQR based Control, Department of Mechanical Engineering Graphic Era University Dehradun-248002,India

[8] S. Bouabdallah, “Design and Control of Quadrotors with Application to Autonomous flying”, EPFL, Lausanne, PhD Thesis 2007

[9] Hajiyev, Chingiz. (2013). LQR Controller with Kalman Estimator Applied to UAV Longitudinal Dynamics. Positioning. 04. 36-41. 10.4236/pos.2013.41005.

[10] R. Beard, «Quadrotor Dynamics and Control Rev 0.1,» February 19, 2008.

[11] Joao P. Hespanha ,’’ Undergraduate lecture notes on LQG/LQR controller design’’, 2001

[12] Mathworks, Simulink Control Design , User’s Guide , R2015b

[13] Πουλιέζος Α. (2014), Σημειώσεις μεταπτυχιακού μαθήματος Βέλτιστος έλεγχος Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης

[14] Moysis, Lazaros. (2018). Kalman filtering , The Prime Magazine

[15] Καραμπετάκης, Βέλτιστος έλεγχος συστημάτων, Αθήνα: Ζήτη, 2009

[16] Κωνσταντίνα Γεωργάκη , ‘’Βέλτιστος Έλεγχος και Φίλτρο Kalman’’ , 2020

[17] Richard Dorf, Bishop Robert. ‘Σύγχρονα Συστήματα Αυτόματου Ελέγχου’ , Εκδόσεις Τζιόλα,2017

[18] Χρήστος Υφούλης, Σημειώσεις μαθήματος Τεχικές Βελτιστοποίησης , ‘Βέλτιστοι ελεγκτές τύπου LQR’,μέρος Α-Β , ΧΕ 21-22

[19] Χρήστος Υφούλης, Σημειώσεις μαθήματος Τεχικές Βελτιστοποίησης ,’Παρατηρητές (Observers) ‘ ,μέρος Α-Β , ΧΕ 21-22

Links

www.mathworks.com

https://youtu.be/H4_hFazBGxU

https://ocw.mit.edu/courses/aeronautics-and-astronautics

https://youtu.be/BVnnPGoF1Iw

http://www.kostasalexis.com/

https://youtu.be/wEevt2a4SKI

https://www.youtube.com/watch?v=3VliM0yysYg&t=639s&ab_channel=Chun-WeiKong