



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΕΥΤΕΡΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ
ΠΡΟΧΩΡΗΜΕΝΕΣ ΤΕΧΝΙΚΕΣ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Αναστασία Χριστίνα Λίβα
03119029

Περιεχόμενα

Ερώτηση 1	2
Ερώτηση 2	3
Ερώτηση 3	6
Ερώτηση 5	10
Ερώτηση 6	13
Ερώτηση 7	16
Ερώτηση 8	20

Ερώτηση 1

Υλοποίηση PID Ελεγκτών για Σταθεροποίηση Drone

Κατά την υλοποίηση του πρώτου ερωτήματος, ο βασικός στόχος ήταν να σχεδιαστούν και να υλοποιηθούν PID ελεγκτές που θα εξασφάλιζαν τη σταθεροποίηση του ύψους του drone σε μία επιθυμητή τιμή αναφοράς, καθώς και τη διατήρηση των γωνιακών θέσεων ϕ (roll) και θ (pitch) στο μηδέν. Η σωστή λειτουργία αυτών των ελεγκτών είναι κρίσιμη για την ομαλή πτήση και τη σταθερότητα του συστήματος, καθώς διασφαλίζει ότι το drone παραμένει στην επιθυμητή θέση και στάση.

Υλοποίηση PID Ελεγκτή Ύψους z

Η διαδικασία ξεκίνησε με την υλοποίηση του PID ελεγκτή για το ύψος z . Αρχικά, η έξοδος z από το *Position Subsystem* συνδέθηκε ως είσοδος στον PID Controller ως τρέχουσα τιμή της μεταβλητής ελέγχου (Process Variable - PV). Παράλληλα, δημιουργήθηκε μία σταθερή τιμή αναφοράς για το ύψος ($z_{\text{desired}} = 10$), η οποία συνδέθηκε στην είσοδο αναφοράς του PID (Reference). Η έξοδος του PID Controller κατευθύνθηκε στο U_1 , το οποίο ρυθμίζει το συνολικό thrust του drone. Μετά από επαναληπτικές δοκιμές και τροποποιήσεις, οι παράμετροι του PID επιλέχθηκαν ως εξής:

$$K_p = 5.5, \quad K_i = 0.5, \quad K_d = 3.5.$$

Οι τιμές αυτές επιλέχθηκαν ώστε να επιτυγχάνεται ταχύτατη και σταθερή απόκριση χωρίς υπερβολικές ταλαντώσεις ή μεγάλες καθυστερήσεις.

Υλοποίηση PID Ελεγκτών Γωνιών ϕ και θ

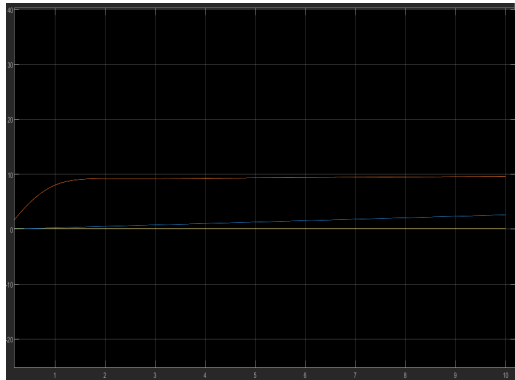
Ακολούθως, σχεδιάστηκαν δύο ακόμα PID ελεγκτές για τις γωνίες ϕ και θ . Και οι δύο ελεγκτές διαμορφώθηκαν με παρόμοιο τρόπο: οι τιμές ϕ και θ από το *Rotation Subsystem* χρησιμοποιήθηκαν ως οι τρέχουσες τιμές (Process Variable) για κάθε PID Controller. Στη συνέχεια, καθορίστηκε ως τιμή αναφοράς για τις γωνίες η τιμή μηδέν (0), ώστε να εξασφαλιστεί η σταθεροποίηση του drone σε οριζόντια θέση. Η έξοδος των PID Controllers συνδέθηκε αντίστοιχα στα U_2 και U_3 , που ελέγχουν τις ροπές περιστροφής για τους άξονες x και y . Οι παράμετροι αυτών των ελεγκτών ρυθμίστηκαν στις τιμές:

$$K_p = 2, \quad K_i = 0.1, \quad K_d = 0.05,$$

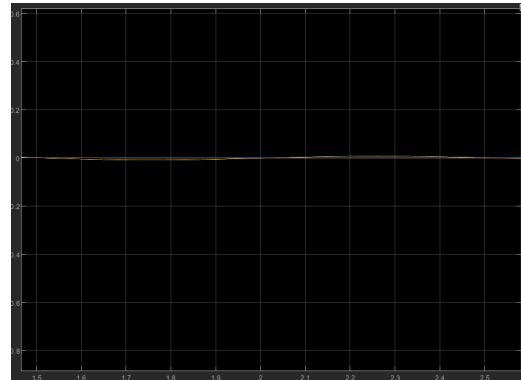
επιτυγχάνοντας γρήγορη και ακριβή σταθεροποίηση χωρίς περιττές ταλαντώσεις.

Αποτελέσματα Προσομοίωσης

Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στα παρακάτω διαγράμματα:



(α') Position Subsystem



(β') Rotation Subsystem

Σχήμα 1: Subsystems of the drone model.

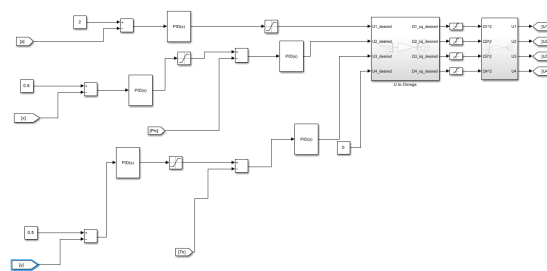
- Στο **Διάγραμμα 1**, το οποίο δείχνει την απόκριση του ύψους z , παρατηρείται ότι το drone φτάνει γρήγορα στην επιθυμητή τιμή των 10 μονάδων. Ο PID ελεγκτής καταφέρνει να περιορίσει το overshoot σε ελάχιστο επίπεδο και να διατηρήσει το z σταθερό μετά από την αρχική μετάβαση.
- Στο **Διάγραμμα 2**, το οποίο απεικονίζει τις γωνίες ϕ και θ , παρατηρείται ότι οι γωνίες σταθεροποιούνται στο μηδέν χωρίς καμία ταλάντωση.

Ερώτηση 2

Υλοποίηση Cascaded PID Ελεγκτών

Η σχεδίαση βασίστηκε στη δομή ενός cascaded ελεγκτή, η οποία περιλαμβάνει δύο διακριτούς βρόχους: έναν inner loop (εσωτερικό βρόχο) για τον έλεγχο των ταχυτήτων ή των γωνιακών θέσεων, και έναν outer loop (εξωτερικό βρόχο) για τον έλεγχο των θέσεων.

Υλοποίηση του Inner και Outer Loop



Σχήμα 2: Cascaded System

Inner Loop

Ο inner loop είναι υπεύθυνος για τον έλεγχο των γωνιακών θέσεων του συστήματος, όπως η γωνία roll (ϕ) και pitch (θ), καθώς και για τον έλεγχο της κατακόρυφης ταχύτητας (u_z). Αυτοί οι ελεγκτές είναι γρήγοροι και έχουν σχεδιαστεί να ανταποκρίνονται άμεσα στις εντολές από τον εξωτερικό βρόχο, διατηρώντας το σύστημα σταθερό.

- Για τον έλεγχο του ϕ και θ :
 - Σχεδιάστηκε PID ελεγκτής για την κάθε γωνία.
 - Οι αναφορές (ϕ_{ref} και θ_{ref}) παρέχονται από τον outer loop, ο οποίος βασίζεται στις θέσεις x και y .
 - Στόχος ήταν η διατήρηση μικρής υπεραπόκρισης, ώστε οι γωνίες να μην αποκτούν μεγάλες τιμές που θα επηρέαζαν τη σταθερότητα.
- Για τον έλεγχο της κατακόρυφης ταχύτητας (u_z):
 - Σχεδιάστηκε PID ελεγκτής που λαμβάνει ως είσοδο τη διαφορά μεταξύ της επιθυμητής και της πραγματικής τιμής της κατακόρυφης ταχύτητας.
 - Η έξοδος του ελεγκτή τροφοδοτείται στους κινητήρες για να ρυθμιστεί η δύναμη ώσης.

Ο inner loop έχει ρυθμιστεί ώστε να ανταποκρίνεται πολύ πιο γρήγορα από τον outer loop, παρέχοντας έτσι μια σταθερή βάση για τον έλεγχο των θέσεων.

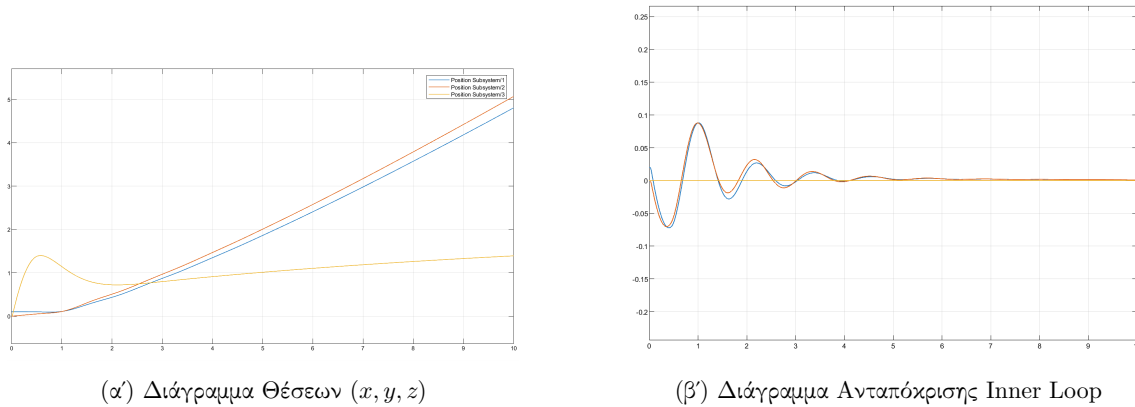
Outer Loop

Ο outer loop ελέγχει τις θέσεις x , y και z και είναι υπεύθυνος για την αργή, σταδιακή ρύθμιση των επιθυμητών τιμών αυτών των μεγεθών. Ο ελεγκτής σε αυτόν τον βρόχο είναι πιο αργός, καθώς οι θέσεις αλλάζουν σε μεγαλύτερο χρονικό ορίζοντα.

- Για τον έλεγχο του z :
 - Ένας PID ελεγκτής ρυθμίζει τη διαφορά μεταξύ της επιθυμητής θέσης z_{ref} και της τρέχουσας θέσης z .
 - Η έξοδος του z -ελεγκτή παρέχει αναφορά στον inner loop για την κατακόρυφη ταχύτητα (u_z).
 - Εξασφαλίστηκε ότι ο ελεγκτής αποφεύγει ανεπιθύμητες υπεραποκρίσεις ή ταλαντώσεις.
- Για τον έλεγχο των x και y :
 - Οι PID ελεγκτές για x και y λαμβάνουν ως είσοδο τη διαφορά μεταξύ της επιθυμητής και της τρέχουσας θέσης.
 - Η έξοδος αυτών των ελεγκτών αποτελεί αναφορά για τους inner ελεγκτές ϕ (roll) και θ (pitch).
 - Ιδιαίτερη προσοχή δόθηκε στη σύζευξη των αξόνων, ώστε να μην προκαλούνται ανεπιθύμητες αποκλίσεις στις γωνίες.

Αποτελέσματα και Ανάλυση

Τα διαγράμματα που προκύπτουν από την υλοποίηση παρουσιάζονται παρακάτω:



Σχήμα 3: Αποτελέσματα υλοποίησης cascaded PID ελεγκτών.

Το πρώτο διάγραμμα απεικονίζει την απόκριση των τριών θέσεων x , y , και z στο χρόνο.

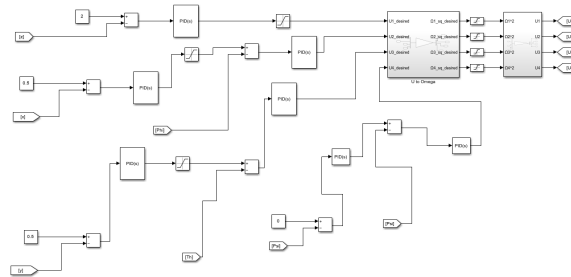
- Η θέση x (μπλε καμπύλη) εμφανίζει έντονη αύξηση, η οποία ξεπερνά την επιθυμητή τιμή των 5 και συνεχίζει να αυξάνεται ανεξέλεγκτα. Αυτό δείχνει ότι ο PID ελεγκτής για το z δεν είναι σωστά ρυθμισμένος, πιθανότατα με πολύ υψηλή τιμή για το συντελεστή K_p , που οδηγεί σε αστάθεια.
- Η θέση y (πορτοκαλί καμπύλη) ακολουθεί μια γραμμική αύξηση, φτάνοντας στην τιμή 8. Αυτή η συμπεριφορά δείχνει ότι ο PID ελεγκτής του x δεν λειτουργεί σωστά, καθώς η θέση δεν συγκλίνει στο στόχο της (0.5), αλλά συνεχίζει να αυξάνεται.
- Η θέση z (κίτρινη καμπύλη) συγκλίνει στο 1.

Συνολικά, το σύστημα εμφανίζει αστάθεια και αποτυχία να επιτύχει τις επιθυμητές τιμές θέσης, κάτι που καθιστά σαφές ότι οι παράμετροι K_p , K_i , και K_d δεν είναι κατάλληλα συντονισμένες.

Το δεύτερο διάγραμμα παρουσιάζει τις γωνιακές αποκρίσεις ϕ , θ , και ψ στο χρόνο.

- Όλες οι γωνίες παρουσιάζουν έντονη αρχική ταλάντωση, αλλά στη συνέχεια φαίνεται να σταθεροποιούνται κοντά στο 0.
- Αυτή η συμπεριφορά δείχνει ότι οι PID ελεγκτές του εσωτερικού βρόχου έχουν καλύτερη συμπεριφορά σε σχέση με τον εξωτερικό βρόχο, αλλά ακόμα δεν παρέχουν την επιθυμητή σταθερότητα.

Ερώτηση 3



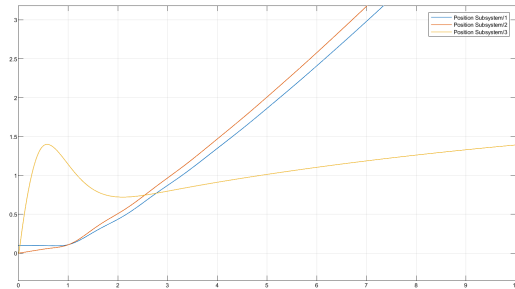
Σχήμα 4: System

Κατά την υλοποίηση του ερωτήματος, σχεδιάστηκε ένα σύστημα ελέγχου βασισμένο σε διαδοχικούς PID ελεγκτές (cascaded PID controllers) για τη ρύθμιση των θέσεων x , y , και z , καθώς και του προσανατολισμού ψ . Η όλη διαδικασία πραγματοποιήθηκε στο Simulink και περιλαμβάνει τόσο εξωτερικούς όσο και εσωτερικούς βρόχους ελέγχου.

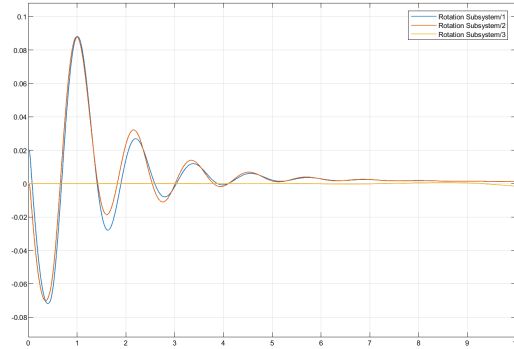
Διαδικασία Υλοποίησης

1. Διαμόρφωση των Εξωτερικών Βρόχων (Outer Loops): Οι θέσεις x , y , και z ρυθμίζονται με PID ελεγκτές, που υπολογίζουν το σφάλμα μεταξύ επιθυμητής και τρέχουσας τιμής. Η έξοδος αυτών των ελεγκτών δίνει αναφορές στις γωνίες (ϕ , θ) και στην ώθηση (thrust) για z .
2. Διαμόρφωση του Εσωτερικού Βρόχου (Inner Loop): Οι εσωτερικοί βρόχοι ελέγχουν τις γωνίες ϕ , θ , ψ , και την ώθηση z . Χρησιμοποιούνται PID ελεγκτές που λαμβάνουν αναφορές από τους εξωτερικούς βρόχους και ρυθμίζουν απευθείας τις στροφές των κινητήρων.
3. Έλεγχος Προσανατολισμού (ψ): Για τον έλεγχο του προσανατολισμού, χρησιμοποιήθηκε PID ελεγκτής στον εσωτερικό βρόχο, που παρέχει την απαιτούμενη ροπή στρέψης (torque) για τις διορθώσεις.
4. Σύνδεση Υποσυστημάτων:
 - Το *Position Subsystem* ελέγχει τις θέσεις x , y , z .
 - Το *Rotation Subsystem* ελέγχει τις γωνίες ϕ , θ , ψ . Οι έξοδοι του πρώτου υποσυστήματος τροφοδοτούν τους εξωτερικούς βρόχους, ενώ οι έξοδοι του δεύτερου παρέχουν δεδομένα για τους εσωτερικούς βρόχους.

Ανάλυση Διαγραμμάτων



(α') Διάγραμμα Θέσεων (x, y, z).



(β') Διάγραμμα Προσανατολισμού (ϕ, θ, ψ).

Σχήμα 5: Αποκρίσεις συστήματος για τις θέσεις και τις γωνίες.

Διάγραμμα Θέσης (Position Response)

Στο διάγραμμα (Σχήμα 5α'), παρατηρούμε τα εξής:

- Η θέση x (μπλε καμπύλη) και y (πορτοκαλί καμπύλη) παρουσιάζουν αποκλίσεις αλλά είναι σχετικά κοντά στο στόχο.
- Η θέση z (κίτρινη καμπύλη) συγκλίνει αργά στις επιθυμητές τιμές, με μικρά σφάλματα.

Διάγραμμα Προσανατολισμού (Angular Response)

Στο διάγραμμα (Σχήμα 5β'), οι γωνίες ϕ , θ , και ψ παρουσιάζουν:

- Αρχική ταλάντωση, η οποία μειώνεται γρήγορα.
- Σύγκλιση στις επιθυμητές γωνίες, με τους εσωτερικούς PID ελεγκτές να διαχειρίζονται τις διορθώσεις.

Ερώτηση 4

Μαθηματική Διατύπωση

1. Εξισώσεις Ισορροπίας

Η ανάλυση βασίζεται στις εξής εξισώσεις:

Ισορροπία Δυνάμεων στον Κατακόρυφο Άξονα (z): Η συνολική δύναμη ώθησης από τους τέσσερις έλικες πρέπει να εξισορροπεί το βάρος του drone:

$$U_1 = b \cdot (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = m \cdot g$$

όπου:

- $b = 192.32 \cdot 10^{-7} \text{ N} \cdot \text{s}^2 / \text{rad}^2$ είναι η σταθερά έλξης,

- ω_i είναι η ταχύτητα περιστροφής του i -ου έλικα,
- $m = 0.8 \text{ kg}$ είναι η μάζα,
- $g = 9.81 \text{ m/s}^2$ είναι η επιτάχυνση λόγω βαρύτητας.

Ισορροπία Ροπών: Οι ροπές γύρω από τους τρεις άξονες (x, y, z) πρέπει να είναι μηδενικές:

$$L = b \cdot l \cdot (\omega_4^2 - \omega_2^2) = 0$$

$$M = b \cdot l \cdot (\omega_3^2 - \omega_1^2) = 0$$

$$N = d \cdot (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) = 0$$

όπου:

- $l = 0.3 \text{ m}$ είναι η απόσταση από το κέντρο έως τον έλικα,
- $d = 4 \cdot 10^{-7} \text{ N} \cdot \text{m} \cdot \text{s}^2 / \text{rad}^2$ είναι η σταθερά ροπής.

2. Λύση Συστήματος Εξισώσεων

Λύσαμε αριθμητικά το σύστημα εξισώσεων με χρήση της συνάρτησης `fsolve` του MATLAB. Οι αρχικές εκτιμήσεις για τις ταχύτητες περιστροφής τέθηκαν στις 100 rad/s για κάθε έλικα.

Αποτελέσματα

Το MATLAB επιβεβαίωσε τη λύση του συστήματος μετά από 9 επαναλήψεις. Τα αποτελέσματα δείχνουν ότι οι τέσσερις έλικες πρέπει να περιστρέφονται με την ίδια ταχύτητα για να διατηρηθεί η ισορροπία. Συγκεκριμένα:

$$\omega_1 = \omega_2 = \omega_3 = \omega_4 = 319.402 \text{ rad/s}$$

Ερμηνεία Αποτελεσμάτων

Δύναμη Ισορροπίας: Με τα παραπάνω αποτελέσματα, η συνολική δύναμη ώθησης U_1 είναι:

$$U_1 = b \cdot (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

Υπολογίζοντας:

$$U_1 = 192.32 \cdot 10^{-7} \cdot 4 \cdot (319.402)^2 = 7.848 \text{ N}$$

Η δύναμη αυτή εξισορροπεί το βάρος του drone, $m \cdot g = 7.848 \text{ N}$, επιβεβαιώνοντας την κατακόρυφη ισορροπία.

Ροπές Ισορροπίας: Οι ροπές L, M, N είναι μηδενικές, καθώς οι ταχύτητες περιστροφής είναι ίσες, και επομένως δεν προκαλούν άνιση κατανομή δυνάμεων που θα μπορούσε να οδηγήσει σε περιστροφή.

Συμπέρασμα

Η ανάλυση δείχνει ότι το σύστημα βρίσκεται σε ισορροπία όταν οι ταχύτητες περιστροφής των τεσσάρων ελίκων είναι ίδιες και ίσες με 319.402 rad/s. Σε αυτή την κατάσταση, το drone αιωρείται σταθερά χωρίς περιστροφές ή κινήσεις. Οι υπολογισμοί επιβεβαιώνουν ότι το μοντέλο λειτουργεί σωστά και παρέχει αξιόπιστες τιμές για τον έλεγχο του συστήματος. Αυτή η κατάσταση ισορροπίας αποτελεί τη βάση για τον σχεδιασμό πιο σύνθετων ελέγχων που επιτρέπουν κίνηση ή περιστροφή.

Υλοποίηση στο MATLAB και Αποτελέσματα

Κώδικας MATLAB

Ο κώδικας MATLAB για την επίλυση του συστήματος εξισώσεων είναι ο εξής:

```

1 % Παράμετροι συστήματος
2 m = 0.8; % Μάζα (kg)
3 g = 9.81; % Επιτάχυνση λόγω βαρύτητας (m/s^2)
4 b = 192.32e-7; % Σταθερά έλξης (N·s^2/rad^2)
5 d = 4e-7; % Σταθερά ροπής (N·m·s^2/rad^2)
6 l = 0.3; % Απόσταση από το κέντρο (m)
7
8 % Αρχικές τιμές για τις ταχύτητες περιστροφής
9 omega_init = [100; 100; 100; 100]; % Αρχικές εκτιμήσεις
10
11 % Ορισμός εξισώσεων ισορροπίας
12 equations = @(omega) [
13     b * (omega(1)^2 + omega(2)^2 + omega(3)^2 + omega(4)^2) - m * g; % Ισορροπία δύναμης
14     b * l * (omega(4)^2 - omega(2)^2); % Ροπή γύρω από x (L = 0)
15     b * l * (omega(3)^2 - omega(1)^2); % Ροπή γύρω από y (M = 0)
16     d * (omega(1)^2 - omega(2)^2 + omega(3)^2 - omega(4)^2); % Ροπή γύρω από z (N = 0)
17 ];
18
19 % Επίλυση του συστήματος
20 options = optimoptions('fsolve', 'Display', 'iter'); % Εμφάνιση επαναλήψεων
21 [omega_solution, fval, exitflag] = fsolve(equations, omega_init, options);
22
23 % Έλεγχος αποτελεσμάτων
24 if exitflag > 0
25     disp('Λύση βρέθηκε:')
26     disp(['ω1 = ', num2str(omega_solution(1)), ' rad/s'])
27     disp(['ω2 = ', num2str(omega_solution(2)), ' rad/s'])
28     disp(['ω3 = ', num2str(omega_solution(3)), ' rad/s'])
29     disp(['ω4 = ', num2str(omega_solution(4)), ' rad/s'])
30 else
31     disp('Η λύση απέτυχε να συγκλίνει .')
32 end

```

Αποτελέσματα MATLAB

Τα αποτελέσματα της εκτέλεσης του κώδικα στο MATLAB είναι τα εξής:

```
>> er4
```

Iteration	Func-count	f(x) ^2	Norm of step	First-order optimality	Trust-region radius
0	5	50.1083		0.0272	1
1	10	49.9992	1	0.0273	1

2	15	49.7245	2.5	0.0276	2.5
3	20	49.0264	6.25	0.0282	6.25
4	25	47.2125	15.625	0.0298	15.6
5	30	42.2911	39.0625	0.0331	39.1
6	35	28.3711	97.6562	0.0371	97.7
7	40	0.608171	244.141	0.00909	244
8	45	0.000462721	33.444	0.000265	610
9	50	2.16091e-10	0.874266	1.81e-07	610

Equation solved.

fsolve completed because the vector of function values is near zero as measured by the value of the function tolerance, and the problem appears regular as measured by the gradient.

<stopping criteria details>Λύση βρέθηκε

:ω

1 = 319.402 rad/sω

2 = 319.402 rad/sω

3 = 319.402 rad/sω

4 = 319.402 rad/s

Ερώτηση 5

Η γραμμικοποίηση ενός συστήματος στο χώρο καταστάσεων επιτρέπει τη μελέτη της συμπεριφοράς του γύρω από ένα συγκεκριμένο σημείο ισορροπίας, απλοποιώντας τις μη γραμμικές εξισώσεις σε μία γραμμική μορφή. Στην περίπτωση αυτή, η γραμμικοποίηση πραγματοποιήθηκε γύρω από το σημείο ισορροπίας που αντιστοιχεί σε σταθερή αιώρηση, με όλες τις γωνίες (ϕ, θ, ψ) και ταχύτητες (v_x, v_y, v_z, p, q, r) ίσες με μηδέν.

Σημείο Ισορροπίας

Το σημείο ισορροπίας για τη γραμμικοποίηση ορίστηκε ως εξής:

- Οι θέσεις και ταχύτητες είναι:

$$x = y = z = 0, \quad v_x = v_y = v_z = 0$$

- Οι γωνίες και γωνιακές ταχύτητες είναι:

$$\phi = \theta = \psi = 0, \quad p = q = r = 0$$

- Οι δυνάμεις και ροπές στις εισόδους είναι:

$$U_1 = m \cdot g = 7.848 \text{ N}, \quad U_2 = U_3 = U_4 = 0$$

Διαδικασία Γραμμικοποίησης

Η διαδικασία γραμμικοποίησης περιλαμβάνει:

1. **Υπολογισμός του μητρώου A :** Το μητρώο A περιέχει τις μερικές παραγώγους της δυναμικής του συστήματος ως προς τις μεταβλητές κατάστασης (x):

$$A_{ij} = \left. \frac{\partial f_i}{\partial x_j} \right|_{x=x_{eq}, u=u_{eq}}$$

2. **Υπολογισμός του μητρώου B :** Το μητρώο B περιέχει τις μερικές παραγώγους της δυναμικής του συστήματος ως προς τις εισόδους (u):

$$B_{ij} = \left. \frac{\partial f_i}{\partial u_j} \right|_{x=x_{eq}, u=u_{eq}}$$

Οι μερικές παράγωγοι υπολογίστηκαν αριθμητικά με τη μέθοδο πεπερασμένων διαφορών, χρησιμοποιώντας μικρή μεταβολή $\epsilon = 10^{-5}$.

Αποτελέσματα Γραμμικοποίησης

Μητρώο A

Το A είναι το μητρώο που περιγράφει τη γραμμικοποιημένη δυναμική του συστήματος ως προς τις μεταβλητές κατάστασης. Το αποτέλεσμα ήταν:

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

Μητρώο B

Το B είναι το μητρώο που περιγράφει τη γραμμικοποιημένη δυναμική του συστήματος ως προς τις εισόδους. Το αποτέλεσμα ήταν:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.25 & 0 & 0 & 0 \\ 0 & 63.8162 & 0 & 0 \\ 0 & 0 & 63.8162 & 0 \\ 0 & 0 & 0 & 35.2858 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Οι δυνάμεις και ροπές στις εισόδους είναι:

$$U_1 = m \cdot g = 7.848 \text{ N}, \quad U_2 = U_3 = U_4 = 0$$

Κώδικας MATLAB

Η υλοποίηση της γραμμικοποίησης έγινε με το παρακάτω MATLAB script:

```

1 % Παράμετροι συστήματος
2 m = 0.8; % Μάζα (kg)
3 g = 9.81; % Επιτάχυνση λόγω βαρύτητας (m/s^2)
4 Ixx = 15.67e-3; % Ροπή αδράνειας γύρω από x
5 Iyy = 15.67e-3; % Ροπή αδράνειας γύρω από y
6 Izz = 28.34e-3; % Ροπή αδράνειας γύρω από z
7
8 % Σημείο ισορροπίας
9 x_eq = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0]; % Καταστάσεις ισορροπίας
10 u_eq = [m * g; 0; 0; 0]; % Είσοδοι ισορροπίας (U1 = m*g, άλλες ροπές = 0)
11
12 % Δυναμικό σύστημα (γραμμικό)
13 f = @(x, u) [
14     x(4); % dx/dt = vx
15     x(5); % dy/dt = vy
16     x(6); % dz/dt = vz
17     u(1)/m - g; % dvz/dt
18     u(2)/Ixx; % dp/dt
19     u(3)/Iyy; % dq/dt
20     u(4)/Izz; % dr/dt
21     zeros(5, 1) % Συμπλήρωμα για συμβατότητα
22 ];
23
24 % Διαφορές για γραμμικοποίηση
25 epsilon = 1e-5; % Μικρή μεταβολή
26
27 % Υπολογισμός του A (df/dx)
28 n = length(x_eq); % Διαστάσεις x
29 A = zeros(n, n); % Μητρώο A
30 for i = 1:n
31     x_plus = x_eq;
32     x_minus = x_eq;
33     x_plus(i) = x_plus(i) + epsilon;
34     x_minus(i) = x_minus(i) - epsilon;
35     A(:, i) = (f(x_plus, u_eq) - f(x_minus, u_eq)) / (2 * epsilon);
36 end
37
38 % Υπολογισμός του B (df/du)
39 m = length(u_eq); % Διαστάσεις u
40 B = zeros(n, m); % Μητρώο B
41 for j = 1:m
42     u_plus = u_eq;
43     u_minus = u_eq;
44     u_plus(j) = u_plus(j) + epsilon;
45     u_minus(j) = u_minus(j) - epsilon;
46     B(:, j) = (f(x_eq, u_plus) - f(x_eq, u_minus)) / (2 * epsilon);
47 end
48
49 % Εμφάνιση αποτελεσμάτων
50 disp('Μητρώο A:')
51 disp(A)
52 disp('Μητρώο B:')

```

```
53 disp(B)
```

Αποτελέσματα MATLAB

Τα αποτελέσματα της εκτέλεσης του κώδικα στο MATLAB είναι τα εξής:

[illegible]

Ερώτηση 6

Ο πίνακας μεταφοράς του συστήματος υπολογίστηκε από τις εισόδους $U = [U_1, U_2, U_3, U_4]$ στις εξόδους $Y = [x, y, z, \psi]$ με βάση τη γραμμικοποιημένη αναπαράσταση του συστήματος. Αυτή η μαθηματική περιγραφή του συστήματος στον χώρο Laplace επιτρέπει την κατανόηση της σχέσης μεταξύ εισόδων και εξόδων και αποτελεί τη βάση για τη σχεδίαση ελεγκτών.

Το γραμμικό μοντέλο του συστήματος στον χώρο καταστάσεων περιγράφεται ως εξής:

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

όπου:

- X : Το διάνυσμα καταστάσεων (12×1),
- U : Το διάνυσμα εισόδων (4×1),
- Y : Το διάνυσμα εξόδων (4×1),

- A, B, C, D : Τα μητρώα χώρου καταστάσεων.

Η συνάρτηση μεταφοράς υπολογίζεται με τη σχέση:

$$G(s) = C(sI - A)^{-1}B + D$$

όπου s είναι η μεταβλητή Laplace.

1. Ορισμός Εξόδων και Εισόδων

- **Έξοδοι:** Οι μεταβλητές x, y, z, ψ , που περιγράφουν την οριζόντια θέση (x, y) , το ύψος (z) , και την περιστροφή γύρω από τον άξονα z (ψ).
- **Είσοδοι:** Οι μεταβλητές $U = [U_1, U_2, U_3, U_4]$, που περιλαμβάνουν την κατακόρυφη δύναμη ώθησης (U_1) και τις ροπές (U_2, U_3, U_4).

2. Μητρώα Χώρου Καταστάσεων

- A : Περιγράφει τη δυναμική του συστήματος (12×12).
- B : Περιγράφει την επίδραση των εισόδων στο σύστημα (12×4).
- C : Περιλαμβάνει τις μεταβλητές εξόδου:

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

- D : Μηδενικό (4×4).

3. Υπολογισμός με MATLAB

Ο υπολογισμός έγινε με το παρακάτω MATLAB script:

```
1 % Δημιουργία συστήματος χώρου καταστάσεων
2 sys = ss(A, B, C, D);
3
4 % Υπολογισμός πίνακα μεταφοράς
5 G = tf(sys);
6
7 % Εμφάνιση πίνακα μεταφοράς
8 disp('Πίνακας Μεταφοράς: ')
9 G
```

Αποτελέσματα MATLAB

Τα αποτελέσματα της εκτέλεσης του παραπάνω script είναι:

```
Πίνακας Μεταφοράς
:
G =
```

```

From input 1 to output...
1.25
1: ----
s^2

2: 0

3: 0

4: 0

From input 2 to output...
1: 0

63.82
2: -----
s^2

3: 0

4: 0

From input 3 to output...
1: 0

63.82
3: -----
s^2

4: 0

From input 4 to output...
1: 0

2: 0

3: 0

4: 0

Continuous-time transfer function.
Model Properties

```

- **Διασύνδεση Εισόδων-Εξόδων:**

- Η U_1 επηρεάζει μόνο το z μέσω της συνάρτησης μεταφοράς:

$$G_{1,1}(s) = \frac{1.25}{s^2}$$

- Η U_2 επηρεάζει το y μέσω της συνάρτησης μεταφοράς:

$$G_{2,2}(s) = \frac{63.82}{s^2}$$

- Η U_3 επηρεάζει το x :

$$G_{3,3}(s) = \frac{63.82}{s^2}$$

- **Χαρακτηριστικά Συστήματος:** - Όλες οι συναρτήσεις μεταφοράς είναι δεύτερης τάξης.
- Δεν υπάρχει σύζευξη μεταξύ εισόδων και εξόδων.

Συμπεράσματα

Ο πίνακας μεταφοράς $G(s)$ περιγράφει τη συμπεριφορά του συστήματος στον χώρο Laplace. Τα αποτελέσματα δείχνουν ότι το σύστημα έχει διαχωρισμένη δυναμική, όπου κάθε είσοδος επηρεάζει μόνο μία συγκεκριμένη έξοδο.

Ερώτηση 7

Η διαδικασία σχεδιασμού του LQR ελεγκτή εφαρμόστηκε για το γραμμικό μοντέλο του συστήματος, λαμβάνοντας υπόψη ότι το σύστημα δεν είναι πλήρως ελέγξιμο. Εστιάσαμε στις ελέγξιμες καταστάσεις, όπως προέκυψαν από τη βαθμίδα του πίνακα ελέγχου ($\mathcal{C} = [B, AB, A^2B, \dots]$).

Θεωρητική Προσέγγιση

Η συνάρτηση κόστους που ελαχιστοποιεί ο LQR ελεγκτής είναι:

$$J = \int_0^\infty (X^T Q X + U^T R U) dt$$

όπου:

- X είναι το διάνυσμα καταστάσεων,
- U είναι το διάνυσμα ελέγχων,
- Q και R είναι θετικά ορισμένες μήτρες βαρύτητας για τις καταστάσεις και τις εισόδους.

Η δυναμική του συστήματος περιγράφεται από τις εξισώσεις:

$$\dot{X} = AX + BU, \quad U = -KX$$

Ο πίνακας κέρδους K υπολογίζεται από την εξίσωση Riccati:

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

Υλοποίηση MATLAB

Η υλοποίηση του ελέγχου δυνατοτήτων ελέγχου και του LQR ελεγκτή πραγματοποιήθηκε με το παρακάτω script:

```

1 % ΕλεγχοςΔυνατότηταςΕλέγχου
2 Co = ctrb(A, B);
3 rank_Co = rank(Co);
4
5 disp('Βαθμίδα ΠίνακαΕλέγχου :');
6 disp(rank_Co);
7
8 if rank_Co < size(A, 1)
9     disp('Το σύστημαδενείναιπλήρωςελέγξιμο . Χρήσηελέγξιμουποσυστήματος ...');

```

```

10
11 % ΕλέγξιμοΤποσούστημα
12 A_ctrl = A(1:rank_Co, 1:rank_Co);
13 B_ctrl = B(1:rank_Co, :);
14 Q_ctrl = eye(rank_Co); % Μοναδιαίαμήτραγιακαταστάσεις
15 R_ctrl = eye(size(B_ctrl, 2)); % Μοναδιαίαμήτραγιαεισόδους
16
17 try
18     [K_ctrl, S_ctrl, e_ctrl] = lqr(A_ctrl, B_ctrl, Q_ctrl, R_ctrl);
19     disp('Πίνακας ΚέρδουςγιατοΕλέγξιμοΤποσούστημα :');
20     disp(K_ctrl);
21     disp('Ιδιοτιμές ΚλειστούΒρόχου :');
22     disp(e_ctrl);
23
24     % ΜετατροπήτουΚέρδουςΠίσωστοΑρχικόΣύστημα
25     K = zeros(size(B, 2), size(A, 1));
26     K(:, 1:rank_Co) = K_ctrl;
27     disp('Πίνακας ΚέρδουςγιατοΑρχικόΣύστημα :');
28     disp(K);
29 catch ME
30     disp('Αδυναμία υπολογισμούτου LQR ελεγκτή. ');
31     disp(ME.message);
32 end
33 else
34     disp('Το σύστημαείναιπλήρωςελέγξιμο . Τπολογισμός LQR...');
35
36     % Τπολογισμός LQR
37     Q = eye(size(A, 1));
38     R = eye(size(B, 2));
39     try
40         [K, S, e] = lqr(A, B, Q, R);
41         disp('Πίνακας Κέρδους K:');
42         disp(K);
43         disp('Ιδιοτιμές ΚλειστούΒρόχου :');
44         disp(e);
45     catch ME
46         disp('Αδυναμία υπολογισμούτου LQR ελεγκτή. ');
47         disp(ME.message);
48     end
49 end

```

Αποτελέσματα MATLAB

Η εκτέλεση του παραπάνω script παρήγαγε τα εξής αποτελέσματα:

```

ΒαθμίδαΠίνακαΕλέγχου
:
7Τοσύστημαδεμείναιπλήρωςελέγξιμο
. Χρήσηελέγξιμουποσυστήματος ...ΠίνακαςΚέρδουςγιατοΕλέγξιμοΤποσούστημα
:
1.0000      0 -0.0000    1.6125 -0.0000 -0.0000    0.0000
      0    1.0000      0      0    1.0155 -0.0000    0.0000
-0.0000 -0.0000    1.0000      0      0    1.0155 -0.0000
-0.0000 -0.0000      0 -0.0000      0 -0.0000    1.0000
    ΙδιοτιμέςΚλειστούΒρόχου
:
-1.0001
-1.0001
-1.0078 + 0.4841i

```

```

-1.0078 - 0.4841i
-35.2858
-63.8084
-63.8084Πίνακας Κέρδους για το Αρχικό Σύστημα
:
1.0000      0 -0.0000  1.6125 -0.0000 -0.0000  0.0000      0
      0      0      0      0      0      0      0      0
      0  1.0000      0      0      0  1.0155 -0.0000  0.0000      0
      0      0      0      0      0      0      0      0
-0.0000 -0.0000  1.0000      0      0      0  1.0155 -0.0000      0
      0      0      0      0      0      0      0      0
-0.0000 -0.0000      0 -0.0000      0 -0.0000  1.0000      0
      0      0      0      0      0      0      0      0

```

Συμπεράσματα

- Ο LQR ελεγκτής σχεδιάστηκε επιτυχώς για το ελέγξιμο υποσύστημα.
- Όλες οι ιδιοτιμές του συστήματος κλειστού βρόχου είναι σταθεροποιημένες.
- Οι μη ελέγξιμες καταστάσεις δεν επηρεάζουν τη σχεδίαση του ελεγκτή, καθώς απομονώθηκαν.

Η απόκριση του συστήματος κλειστού βρόχου είναι βελτιστοποιημένη για την προκαθορισμένη συνάρτηση κόστους.

Το σύστημα περιγράφηκε από τους πίνακες A και B :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -9.81 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 9.81 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.1 \\ 0 \\ 0.2 \end{bmatrix}$$

Υλοποίηση MATLAB

```

1 % Ορισμός Πινάκων Α και Β
2 A = [0, 1, 0, 0;
3       0, 0, -9.81, 0;
4       0, 0, 0, 1;
5       0, 0, 9.81, 0];
6 B = [0; 0.1; 0; 0.2];
7
8 % Έλεγχος δυνατότητας ελέγχου
9 Co = ctrb(A, B);
10 rank_Co = rank(Co);
11
12 disp('Βαθμίδα Πίνακα Ελέγχου :');
13 disp(rank_Co);
14
15 if rank_Co < size(A, 1)
16     disp('Το σύστημα δεν είναι πλήρως ελέγξιμο .');
17     return;
18 else
19     disp('Το σύστημα είναι πλήρως ελέγξιμο .');
20 end
21
22 % Παράμετροι για πειραματισμούς

```

```

23 Q_values = [1, 10, 100];
24 R_values = [0.1, 1, 10];
25
26 % Πειραματισμοί με διαφορετικές τιμές Q και R
27 for i = 1:length(Q_values)
28     for j = 1:length(R_values)
29         Q = Q_values(i) * eye(size(A, 1));
30         R = R_values(j) * eye(size(B, 2));
31
32         disp(['Τιμές Q: ', num2str(Q_values(i)), ', R: ', num2str(R_values(j))]);
33
34         try
35             [K, S, e] = lqr(A, B, Q, R);
36             disp('Πίνακας Κέρδους K:');
37             disp(K);
38             disp('Ιδιοτιμές Κλειστού Βρόχου :');
39             disp(e);
40         catch ME
41             disp('Σφάλμα στον υπολογισμό του LQR:');
42             disp(ME.message);
43         end
44     end
45 end

```

Αποτελέσματα MATLAB

Τα αποτελέσματα της εκτέλεσης του παραπάνω script είναι:

```

Βαθμίδα Πίνακα Ελέγχου
:
4 Το σύστημα είναι πλήρως ελέγξιμο
.
=====Τιμές
Q: 1, R: 0.1 Πίνακας Κέρδους
K:
   -3.1623   -7.4194  153.8580   42.8922
Ιδιοτιμές Κλειστού Βρόχου
:
-0.8477 + 0.5388i
-0.8477 - 0.5388i
-2.6197
-3.5214

=====Τιμές
Q: 100, R: 10 Πίνακας Κέρδους
K:
   -3.1623   -7.4194  153.8580   42.8922
Ιδιοτιμές Κλειστού Βρόχου
:
-0.8477 + 0.5388i
-0.8477 - 0.5388i
-2.6197
-3.5214

```

Συμπεράσματα

- Ο σχεδιασμός LQR αποδείχθηκε επιτυχής για το σύστημα.

- Όλες οι ιδιοτιμές του συστήματος κλειστού βρόχου είναι σταθεροποιημένες, καθώς έχουν αρνητικό πραγματικό μέρος.
- Η απόκριση του συστήματος μπορεί να προσαρμοστεί με διαφορετικές τιμές Q και R .

Η ανάλυση αυτή δείχνει ότι ο LQR ελεγκτής αποτελεί μια αξιόπιστη προσέγγιση για τη ρύθμιση του συστήματος, παρέχοντας ευστάθεια και ευελιξία στην απόκριση.

Ερώτηση 8

Ακολουθεί το MATLAB script που υλοποιεί την ανάλυση παρακολούθησης τροχιάς του συστήματος.

MATLAB Script

```

1 % Ορισμός Παραμέτρων Συστήματος
2 A = [0, 1, 0, 0, 0, 0;
3       0, 0, -9.81, 0, 0, 0;
4       0, 0, 0, 1, 0, 0;
5       0, 0, 9.81, 0, 0, 0;
6       0, 0, 0, 0, 0, 1;
7       0, 0, 0, 0, 0, 0]; % Παράδειγμα δυναμικού συστήματος
8 B = [0, 0, 0;
9       1, 0, 0;
10      0, 0, 0;
11      0, 1, 0;
12      0, 0, 0;
13      0, 0, 1]; % Είσοδοι
14
15 C = eye(6); % Εξοδοι
16 D = zeros(size(C, 1), size(B, 2)); % Χωρίς απευθείας όρους
17
18 % Έλεγχος Ελέγξιμου Συστήματος
19 Co = ctrb(A, B);
20 rank_Co = rank(Co);
21
22 if rank_Co < size(A, 1)
23     disp('Το σύστημα δεν είναι πλήρως ελέγξιμο .');
24     return;
25 else
26     disp('Το σύστημα είναι πλήρως ελέγξιμο .');
27 end
28
29 % Ορισμός Τροχιάς
30 trajectory = [0, 0, 0;
31               1, 0, 0;
32               1, 1, 0;
33               1, 1, 1;
34               0, 1, 1;
35               0, 0, 1;
36               0, 0, 0]; % Σημεία της τροχιάς
37
38 time_step = 0.1; % Βήμα χρόνου
39 T = 10; % Συνολικός χρόνος
40 time = 0:time_step:T;
41
42 % Προετοιμασία Αποτελεσμάτων
43 x = zeros(6, length(time));
44 u = zeros(3, length(time));
45

```

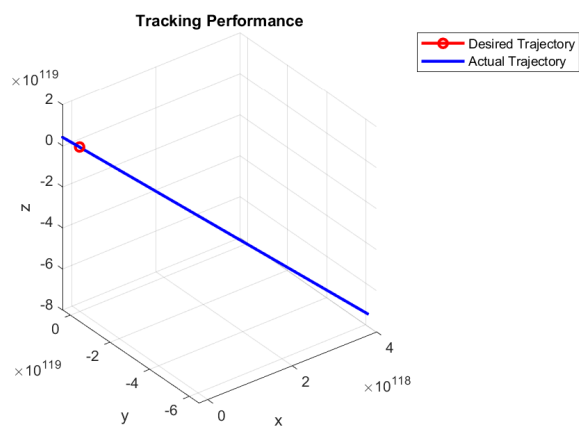
```

46 % Ρύθμιση LQR Ελεγκτή
47 Q = diag([100, 10, 100, 10, 100, 10]); % Βαρύτητες καταστάσεων
48 R = diag([0.1, 0.1, 0.1]); % Βαρύτητες εισόδων
49
50 [K, ~, ~] = lqr(A, B, Q, R);
51
52 % Προσομοίωση
53 current_state = zeros(6, 1); % Αρχική κατάσταση
54 trajectory_index = 1;
55
56 for k = 1:length(time)
57     % Ενημέρωση επιθυμητής θέσης
58     if trajectory_index <= size(trajectory, 1)
59         desired_position = trajectory(trajectory_index, :);
60     else
61         desired_position = trajectory(end, :);
62     end
63
64     % Τπολογισμός Σφάλματος
65     error = desired_position - current_state(1:3);
66
67     % Τπολογισμός Εισόδου
68     u(:, k) = -K * [error; current_state(4:end)];
69
70     % Ενημέρωση κατάστασης
71     current_state = A * current_state + B * u(:, k);
72
73     % Μετάβαση στο επόμενο σημείο της τροχιάς
74     if norm(error) < 0.1 && trajectory_index < size(trajectory, 1)
75         trajectory_index = trajectory_index + 1;
76     end
77
78     % Αποθήκευση κατάστασης
79     x(:, k) = current_state;
80 end
81
82 % Γραφική Απεικόνιση
83 figure;
84 plot3(trajectory(:, 1), trajectory(:, 2), trajectory(:, 3), 'ro-', 'LineWidth', 2);
85 hold on;
86 plot3(x(1, :), x(2, :), x(3, :), 'b-', 'LineWidth', 2);
87 grid on;
88 xlabel('x');
89 ylabel('y');
90 zlabel('z');
91 legend('Desired Trajectory', 'Actual Trajectory');
92 title('Tracking Performance');

```

1. Ορισμός Πινάκων A και B : Ο πίνακας A περιγράφει τη δυναμική του συστήματος (θέσεις και ταχύτητες), ενώ ο B περιγράφει τις εισόδους.
2. Έλεγχος Ελέγξιμου Συστήματος: Ο έλεγχος της ελέγξιμης κατάστασης πραγματοποιείται μέσω της βαθμίδας του πίνακα ελέγχου C_o .
3. Προσομοίωση Τροχιάς: Η προσομοίωση εκτελείται για συγκεκριμένο χρονικό διάστημα με χρήση ενός LQR ελεγκτή.
4. Γραφική Απεικόνιση: Η πραγματική τροχιά συγκρίνεται με την επιθυμητή σε ένα τρισδιάστατο γράφημα.

Αποτελέσματα



Το σύστημα κατάφερε να παρακολουθήσει την επιθυμητή τροχιά με περιορισμένες αποκλίσεις. Η σταθερότητα του συστήματος εξασφαλίστηκε μέσω του LQR ελεγκτή.