



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΕΦΑΡΜΟΓΩΝ ΩΣ ΕΡΓΑΛΕΙΟ ΕΚΜΑΘΗΣΗΣ ΤΗΣ ΜΟΝΤΕΡΝΑΣ ΘΕΩΡΙΑΣ ΕΛΕΓΧΟΥ

Διπλωματική Εργασία: Χασάνδρα Ελβίρα-Χριστίνα
Επιβλέπων Καθηγητής: Ιψάκης Δημήτριος

Χανιά, 2024

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική αφορά την ανάπτυξη εκπαιδευτικού υλικού που περιλαμβάνει ασκήσεις και υπολογιστικές εφαρμογές με βάση την Μοντέρνα Θεωρία Ελέγχου. Συγκεκριμένα γίνεται ανάλυση παραδειγμάτων τα οποία μπορούν να επιλυθούν με την χρήση κώδικα στο περιβάλλον της MATLAB. Στο πρώτο κεφάλαιο, παρουσιάζεται αναλυτικά ο σχεδιασμός ελεγκτών με ανάδραση, καθώς και η αναπαράσταση των σημάτων εισόδου/εξόδου και σφάλματος σε μορφή διαγραμμάτων. Συμπληρωματικά, το δεύτερο κεφάλαιο ασχολείται με κάποιες ειδικές περιπτώσεις συστημάτων ελέγχου όπως, Συστήματα Ελέγχου Πρόδρασης, Συστήματα Ελέγχου Πρόδρασης/Ανάδρασης και Συστήματα Συστοιχίας καθώς και τη σύγκριση αυτών. Έπειτα, στα επόμενα δύο κεφάλαια αναλύονται τα Μοντέλα Χώρου Κατάστασης καθώς και η γραμμικοποίηση και επίλυση συστημάτων με πολλές εισόδους και εξόδους. Συνεχίζοντας, στο πέμπτο κεφάλαιο γίνεται παρουσίαση κάποιων βασικών ιδιοτήτων των πολυμεταβλητών συστημάτων όπως η ελεγκσιμότητα και η παρατηρησιμότητα, που αποτελούν ιδιότητες των μοντέλων χώρου κατάστασης. Στο έκτο κεφάλαιο, γίνεται σχεδιασμός ελεγκτών με γραμμική ανάδραση καταστάσεων σε συστήματα κλειστού βρόγχου και συστήματα πολλών εισόδων και πολλών εξόδων (ΠΕΠΕ). Τέλος, στο έβδομο κεφάλαιο παρουσιάζονται παραδείγματα επίλυσης παρατηρητών κατάστασης καθώς και παραδείγματα παρατηρητών κατάστασης με παρουσία ελεγκτών.

ABSTRACT

This thesis concerns the development of educational material that includes exercises and computational applications based on Modern Control Theory. Specifically, it analyzes examples that can be solved using code in the MATLAB environment. The first chapter presents in detail, the design of feedback controllers, along with the representation of input/output signals and error in diagram form. Additionally, the second chapter deals with some special cases of control systems, such as Pre-Action (Feedforward) Control Systems, Pre-Action (Feedforward)/Feedback Control Systems, and Cascade Systems, as well as their comparison. Then, the next two chapters analyze State Space Models and System Linearization in multi-input - multi-output (MIMO) systems. Continuing, the fifth chapter presents fundamental properties of multivariable systems, including controllability and observability, which are properties of state space models. In the sixth chapter, the design of controllers with linear state feedback in closed-loop systems and multi-input - multi-output systems are presented. Finally, the seventh chapter, presents examples of solving state observers, as well as examples of state observers with the presence of controllers.

Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ	5
ΚΕΦΑΛΑΙΟ 1: Σχεδιασμός ή Σύνθεση Ελεγκτών με Ανάδραση	7
1.1 Σχεδιασμός Ελεγκτή με ανάδραση.....	7
Παράδειγμα 1.1.1: Σχεδιασμός Ελεγκτή ανατροφοδότησης (1 ^{ης} τάξης).....	7
Παράδειγμα 1.1.2: Σχεδιασμός Ελεγκτή ανατροφοδότησης (2 ^{ης} τάξης).....	12
Παράδειγμα 1.1.3: Σχεδιασμός Ελεγκτή ανατροφοδότησης (αρνητικό μηδενικό).....	15
Παράδειγμα 1.1.4: Σχεδιασμός Ελεγκτή (θετικό μηδενικό)	18
Παράδειγμα 1.1.5: Σχεδιασμός Ελεγκτή ανατροφοδότησης (νεκρός χρόνος)	21
Παράδειγμα 1.1.6: Σχεδιασμός Ελεγκτή ανατροφοδότησης (n-τάξης).....	24
ΚΕΦΑΛΑΙΟ 2: Ειδικές Περιπτώσεις Συστημάτων Ελέγχου.....	27
2.1 Έλεγχος Πρόδρασης	27
Παράδειγμα 2.1.1: Σχεδιασμός Ελεγκτή πρόδρασης.....	27
2.2 Έλεγχος Πρόδρασης/Ανάδρασης	33
Παράδειγμα 2.2.1: Σχεδιασμός Ελεγκτή πρόδρασης/ανάδρασης	33
2.3 Έλεγχος Συστοιχίας	38
Παράδειγμα 2.3.1: Σχεδιασμός και σύγκριση συστημάτων ελέγχου ανάδρασης και συστοιχίας.....	38
ΚΕΦΑΛΑΙΟ 3: Μοντέλα Χώρου Κατάστασης και Γραμμικοποίηση Συστημάτων	49
Παράδειγμα 3.1.1: Δοχεία σε σειρά (Σημείο ισορροπίας 0)	49
Παράδειγμα 3.1.2: Δοχεία σε σειρά (Σημείο ισορροπίας $\neq 0$)	52
Παράδειγμα 3.1.3: Ανάρτηση βαρέως οχήματος	55
ΚΕΦΑΛΑΙΟ 4: Μοντέλα Χώρου Κατάστασης και Επίλυση Συστημάτων	59
4.1 Επίλυση στο πεδίο του χρόνου	59
Παράδειγμα 4.1.1: Εύρεση εκθετικού πίνακα e^{At} (Για ομογενείς διαφορικές εξισώσεις).....	59
Παράδειγμα 4.1.2: Επίλυση στο πεδίο του χρόνου (Για ομογενείς διαφορικές εξισώσεις)	60
Παράδειγμα 4.1.3: Επίλυση στο πεδίο του χρόνου (Για μη-ομογενείς διαφορικές εξισώσεις) ...	61
4.2 Επίλυση στο πεδίο Laplace	63
Παράδειγμα 4.2.1: Επίλυση στο πεδίο Laplace (Για ομογενείς διαφορικές εξισώσεις).....	63
Παράδειγμα 4.2.2: Επίλυση στο πεδίο Laplace (Για μη-ομογενείς διαφορικές εξισώσεις)	64
4.3 Επίλυση με τη χρήση ιδιοτιμών/ιδιοδιανυσμάτων.....	66
Παράδειγμα 4.3.1: Επίλυση με τη χρήση ιδιοτιμών/ιδιοδιανυσμάτων	66
4.4 Προσδιορισμός συνάρτησης μεταφοράς και αποκρίσεων y	67
Παράδειγμα 4.4.1: Προσδιορισμός $y(t)$ εάν $u(t)=1$ (χωρίς διαταραχές)	67
Παράδειγμα 4.4.2: Προσδιορισμός $y(t)$ εάν $u(t)=1$ (παρουσία διαταραχών).....	70
ΚΕΦΑΛΑΙΟ 5: Ιδιότητες συστημάτων πολλών εισόδων –πολλών εξόδων	74
5.1 Ευστάθεια Πολυμεταβλητών Συστημάτων	74
Παράδειγμα 5.1.1: Προσδιορισμός ευστάθειας πολυμετάβλητου συστήματος.....	74

5.2	Πόλοι/Μηδενικά Πολυμεταβλητού Συστήματος	76
	Παράδειγμα 5.2.1: Υπολογισμός πόλων/μηδενικών πολυμετάβλητου συστήματος	76
5.3	Γραμμικοί Μετασχηματισμοί.....	78
	Παράδειγμα 5.3.1: Προσδιορισμός συνάρτησης μεταφοράς δύο συστημάτων (αρχικό και μετασχηματισμένο)	78
5.4	Ελεγχιμότητα/ Παρατηρησιμότητα	82
	Παράδειγμα 5.4.1: Έλεγχος ελεγχιμότητας και παρατηρησιμότητας συστημάτων	83
5.5	Συνάρτηση Μεταφοράς & Μοντέλο Χώρου Κατάστασης	84
	Παράδειγμα 5.5.1: Προσδιορισμός Συνάρτησης Μεταφοράς.....	84
ΚΕΦΑΛΑΙΟ 6: Σχεδιασμός Ελεγκτών/Γραμμική Ανάδραση Καταστάσεων		87
6.1	Γραμμική Ανάδραση Καταστάσεων.....	87
	Παράδειγμα 6.1.1: Προσδιορισμός ανύσματος κατάστασης «Κ».....	87
6.2	Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης	88
	Παράδειγμα 6.2.1: Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης ($v=0$)	89
	Παράδειγμα 6.2.2: Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης ($v \neq 0$)	92
	Παράδειγμα 6.2.3: Προσδιορισμός ανύσματος κατάστασης «Κ» με χρήση LQR ελεγκτή	95
6.3	Γραμμική Ανάδραση Καταστάσεων (ΠΕΠΕ).....	96
	Παράδειγμα 6.3.1: Προσδιορισμός ανύσματος κατάστασης «Κ» για συστήματα ΠΕΠΕ	96
	Παράδειγμα 6.3.2: Προσδιορισμός ανύσματος κατάστασης «Κ» για συστήματα ΠΕΠΕ με χρήση LQR ελεγκτή	101
ΚΕΦΑΛΑΙΟ 7: Παρατηρητές Κατάστασης		103
	Παράδειγμα 7.1.1: Προσδιορισμός ανύσματος L (παρατηρητής)	104
	Παράδειγμα 7.1.2: Πολλαπλά ερωτήματα χωρίς διαταραχή	107
	Παράδειγμα 7.1.3: Πολλαπλά ερωτήματα με διαταραχή.....	116
ΣΥΜΠΕΡΑΣΜΑΤΑ		126
ΠΑΡΑΡΤΗΜΑ		128
	ΚΕΦΑΛΑΙΟ 1:.....	128
	ΚΕΦΑΛΑΙΟ 2:.....	134
	ΚΕΦΑΛΑΙΟ 3:.....	140
	ΚΕΦΑΛΑΙΟ 4:.....	144
	ΚΕΦΑΛΑΙΟ 5:.....	149
	ΚΕΦΑΛΑΙΟ 6:.....	155
	ΚΕΦΑΛΑΙΟ 7:.....	164

ΕΙΣΑΓΩΓΗ

Το περιβάλλον της MATLAB

Η MATLAB είναι μια προγραμματιστική πλατφόρμα σχεδιασμένη ειδικά για μηχανικούς και επιστήμονες διάφορων κλάδων. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η MATLAB, η οποία χρησιμοποιεί πολλές εντολές της C++ όπως την if, την while και την switch. Είναι μια γλώσσα βασισμένη σε πίνακες που επιτρέπει μια πιο φυσική έκφραση υπολογιστικών μαθηματικών.

Συχνά χρησιμοποιείται για την επίλυση σύνθετων μαθηματικών προβλημάτων, την ανάλυση δεδομένων και την γραφική αναπαράστασή τους, την ανάπτυξη αλγορίθμων και τη δημιουργία διαφόρων μοντέλων και εφαρμογών σε διάφορες θεματικές ενότητες (στατιστική ανάλυση, βελτιστοποίηση, νευρωνικά δίκτυα, «συμβολικά» μαθηματικά κ.α.).

Βασικά στοιχεία του MATLAB

Δημιουργία m-File

Τα m-files διακρίνονται σε:

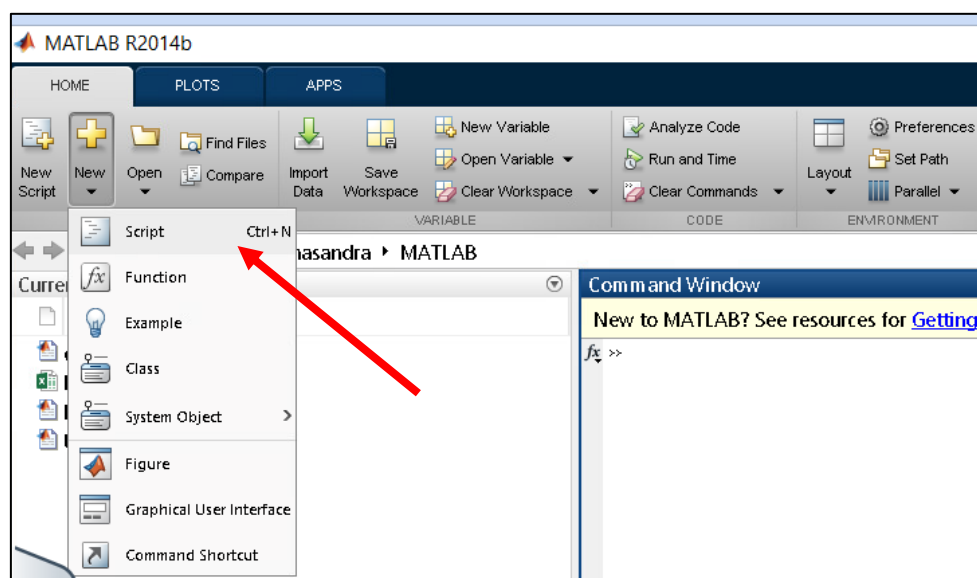
- **Αρχεία script (ή Αρχεία εντολών):** Εκτελούν μια ακολουθία εντολών σε μεταβλητές που έχουν ορισθεί στο χώρο της εργασίας
- **Αρχεία συναρτήσεων:** Περιλαμβάνουν τον ορισμό μιας συνάρτησης και δέχονται ορίσματα εισόδου και επιστρέφουν μεταβλητές εξόδου

Στην παρούσα εργασία γίνεται χρήση αρχείων script.

Για την δημιουργία αυτών, αφού συνδεθούμε στο περιβάλλον του MATLAB, επιλέγουμε το εικονίδιο



που εμφανίζεται πάνω αριστερά στο Toolstrip και πατάμε Script ή πληκτρολογούμε κατευθείαν από το πληκτρολόγιο “Ctrl + N”.

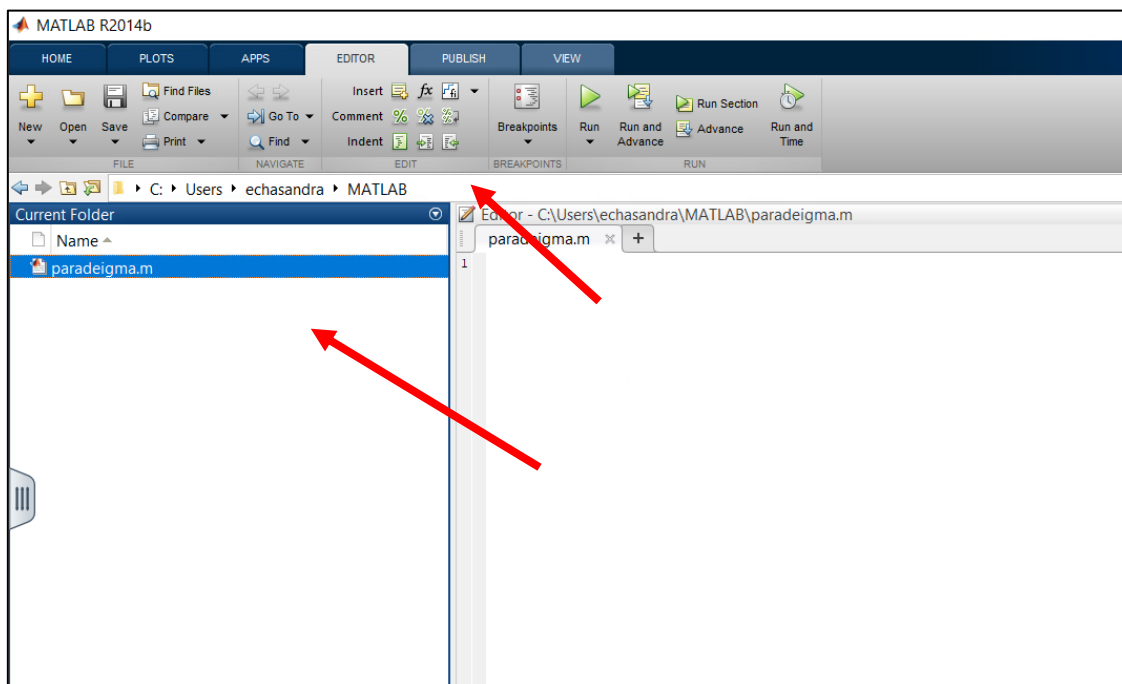


Εικόνα 1.1, Κεντρική σελίδα της πλατφόρμας MATLAB



Στην συνέχεια, μόλις ανοίξει το m.file θα πρέπει να πατήσουμε απευθείας save () πάνω αριστερά στο *Toolstrip* έτσι ώστε να δώσουμε το όνομα του αρχείου που επιθυμούμε. Είναι σημαντικό τα ονόματα αυτών των αρχείων να είναι πάντοτε με λατινικούς χαρακτήρες (π.χ. askisi1.m). Επιπλέον θα πρέπει η διαδρομή καθώς και ο φάκελος στον οποίο θα αποθηκεύσουμε το αρχείο να είναι και αυτά με λατινικούς χαρακτήρες.

Τέλος θα πρέπει η διαδρομή προορισμού που αναγράφεται στο *Toolbar* να είναι ίδια με αυτή που φαίνεται αριστερά στη περιοχή *Current Folder*, διαφορετικά δεν θα εκτελεστεί ο κώδικας και θα βγάλει κάποιο σχετικό μήνυμα λάθους ή συμβουλής.



Εικόνα 1.2 Κεντρική σελίδα της πλατφόρμας MATLAB

ΚΕΦΑΛΑΙΟ 1: Σχεδιασμός ή Σύνθεση Ελεγκτών με Ανάδραση

Σε αυτό το 1^ο κεφάλαιο θα παρουσιαστούν παραδείγματα σχεδιασμού ή σύνθεσης ελεγκτών με βάση το πρότυπο μοντέλο της συνάρτησης μεταφοράς διεργασίας $Gp(s)$ σε συστήματα ελάχιστης και μη-ελάχιστης φάσης [$Gp^-(s)$ και $Gp^+(s)$].

Συστήματα μη-ελάχιστης φάσης [$Gp^+(s)$] ορίζονται τα συστήματα τα οποία έχουν νεκρό χρόνο ή θετικά μηδενικά (ρίζες αριθμητή).

Στόχος είναι η δημιουργία ενός ελεγκτή $Gc(s)$ ο οποίος θα βοηθήσει ώστε να υπάρχουν ταυτόχρονα στο σύστημα μας τα εξής:

1. Ευστάθεια Κλειστού Βρόχου
2. Μηδενικό σφάλμα
3. Επιθυμητή δυναμική απόδοση

1.1 Σχεδιασμός Ελεγκτή με ανάδραση

Παρακάτω θα παρουσιαστούν και θα αναλυθούν παραδείγματα σχεδιασμού ελεγκτή με βάση τη μεθοδολογία Morari και Zafiriou που έχει παρουσιαστεί στην θεωρία (ή εμφανίζεται στα αγγλικά ως Internal model control), και θα αναπαρασταθούν γραφικά τα σήματα εισόδου και εξόδου καθώς και το σφάλμα, συναρτήσει του χρόνου.

Παράδειγμα 1.1.1: Σχεδιασμός Ελεγκτή ανατροφοδότησης (1^{ης} τάξης)

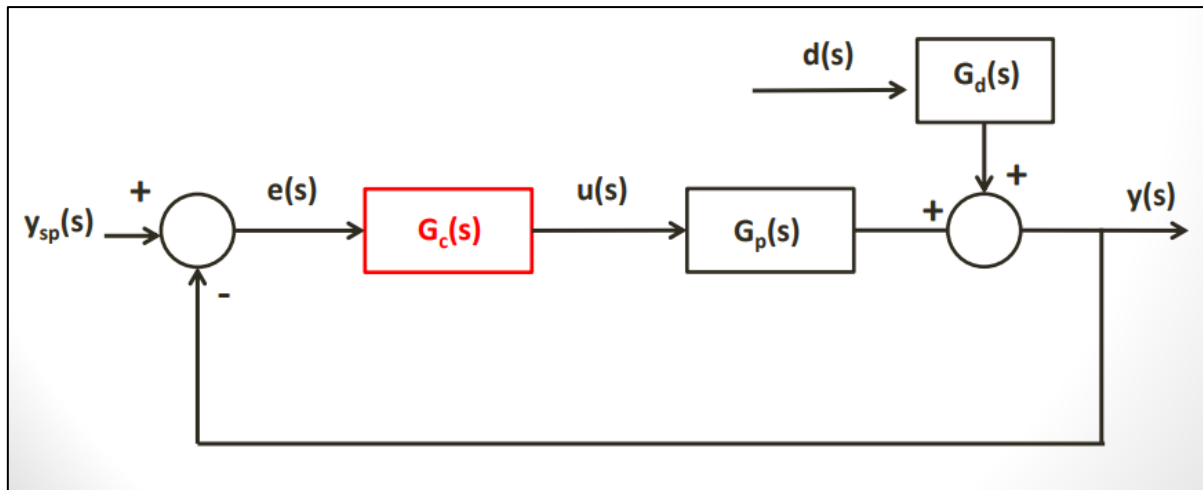
Να σχεδιάσετε έναν κατάλληλο ελεγκτή ανατροφοδότησης με βάση την μεθοδολογία Morari and Zafiriou εάν γνωρίζετε ότι $Gp(s)=K/(ts+1)$ και $Gd(s)=1$.

Από την θεωρία γνωρίζουμε ότι:

$$Gc(s) = \frac{1}{Gp^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - Gp^+(s)}$$

$$y(s) = \frac{1}{(\lambda s + 1)^r} Gp^+(s) \cdot ysp(s) + \left[1 - \frac{1}{(\lambda s + 1)^r} Gp^+(s) \right] d(s)$$

όπου r = βαθμός πολυωνύμου παρονομαστή – βαθμός πολυωνύμου αριθμητή της $Gp(s)$ και λ : μικρή χρονική σταθερά που καθορίζεται από εμάς.



Εικόνα 1.3 Σύστημα ελέγχου ανατροφοδότησης

Επίλυση: Αρχικά υπολογίζουμε στο χαρτί μας βάσει της θεωρίας τις συναρτήσεις ελάχιστης $Gp^-(s)$ και μη ελάχιστης φάσης $Gp^+(s)$.

$$Gp(s) = Gp^-(s) \cdot Gp^+(s)$$

όπου

$$Gp^+(s) = 1$$

$$Gp^-(s) = Gp(s)$$

και $r = 1 - 0 = 1$.

Με αντικατάσταση στην $G_c(s)$ προκύπτει:

$$G_c(s) = \frac{\tau}{K\lambda} \cdot \left(1 + \frac{1}{\tau s}\right)$$

όπου K =σταθερά ενίσχυσης και τ =χρονική σταθερά

ΣΗΜΕΙΩΣΗ: Ορίζοντας, $Kc=\tau/K\lambda$ και $\tau=\tau_I$ ο ελεγκτής μετατρέπεται σε έναν PI ελεγκτή που έχει αναλυθεί στο μάθημα «Συστήματα Ελέγχου Ι». Επομένως εμείς θα επιλέξουμε την τιμή του λ μόνο ως άγνωστη παράμετρο.

Όταν θέλουμε να εργαστούμε στο περιβάλλον της MATLAB θα πρέπει να πληκτρολογήσουμε τις εντολές **clear** και **clc** προκειμένου να αποφύγουμε τυχόν προβλήματα κατά την εκτέλεση του κώδικα, σε περίπτωση που υπάρχουν εντολές στο ιστορικό από προηγούμενα προβλήματα. Στην συνέχεια χρησιμοποιείται η εντολή **syms** για να δηλώσουμε ότι θα χρησιμοποιηθούν συμβολικές μεταβλητές. Θεωρώντας συμβολικές μεταβλητές κάνουμε ακριβείς πράξεις χρησιμοποιώντας την ακριβή άλγεβρα και αριθμητική που θα χρησιμοποιούσαμε κάνοντας αυτές τις πράξεις στο χαρτί. Επομένως δεν χρειάζεται να δοθούν τιμές σε αυτές τις μεταβλητές.

```
clc
clear

syms s
```

Στη συνέχεια, δηλώνουμε τις σταθερές τιμές K, τ και λ καθώς και τις συναρτήσεις $Gp(s)$ και $Gc(s)$ όπως ορίστηκαν παραπάνω.

ΣΗΜΕΙΩΣΗ: Στο σημείο αυτό θα μπορούσαμε να δηλώσουμε τις $Gp^-(s)$ και $Gp^+(s)$ που έχουμε υπολογίσει έτσι ώστε η $Gc(s)$ του ελεγκτή να προσδιοριστεί ως μια γενική μορφή των $Gp^-(s)$ και $Gp^+(s)$, χωρίς να την υπολογίσουμε στο χαρτί. Δηλαδή:

```
K=5; t=2; lamda=1;
Gp=K/(t*s+1);
Gc=(t/(K*lamda))*(1+(1/(t*s)));
K=5; t=2; lamda=0.1;
```

Έπειτα, δηλώνεται η συνάρτηση μεταφοράς της διαταραχής και το σήμα αναφοράς (Y_{sp}) και διαταραχής (D). Μπορούμε να επιλέξουμε διάφορες μεταβολές σε αυτά τα σήματα.

```
Gd=1;

Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```

Στο επόμενο βήμα προσδιορίζουμε τις σχέσεις κλειστού βρόχου $Y(s)$ και $y(t)$, το σφάλμα E και το σήμα εισόδου U που μεταβάλλεται από τον ελεγκτή μας και δηλώνουμε την διάρκεια του χρόνου:

$$Y = \frac{Gp \cdot Gc}{1 + Gp \cdot Gc} Y_{sp}(s) + \frac{Gd}{1 + Gp \cdot Gc} D(s)$$

$$E = Y_{sp}(s) - Y(s)$$

$$U = Gc(s) \cdot E(s)$$

```
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);

Total_Time=30;
t=0:0.1:Total_Time;

y=subs(y,t);
y=double(y);

u=subs(u,t);
u=double(u);

error=subs(error,t);
error=double(error);
```

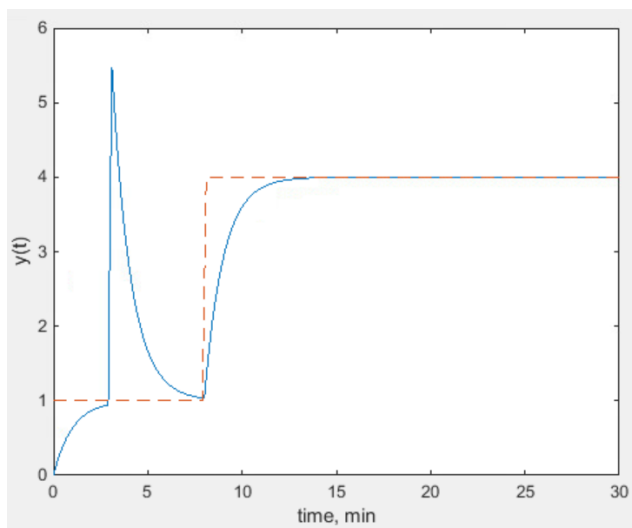
```
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
```

Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

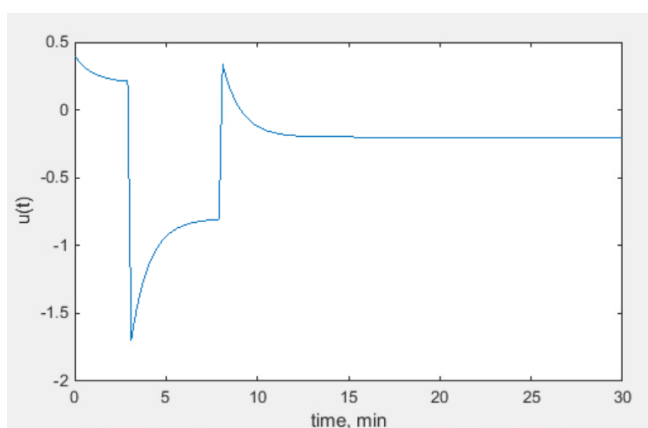
```
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

ΣΗΜΕΙΩΣΗ: Τα όρια των αξόνων X και Y μπορούμε να τα μεταβάλλουμε όπως θέλουμε με την εντολή axis (π.χ figure(1);plot(t,y,t,ysp,'--'); **axis ([0 100 0 100]);** xlabel('time, min');ylabel('y(t)'))

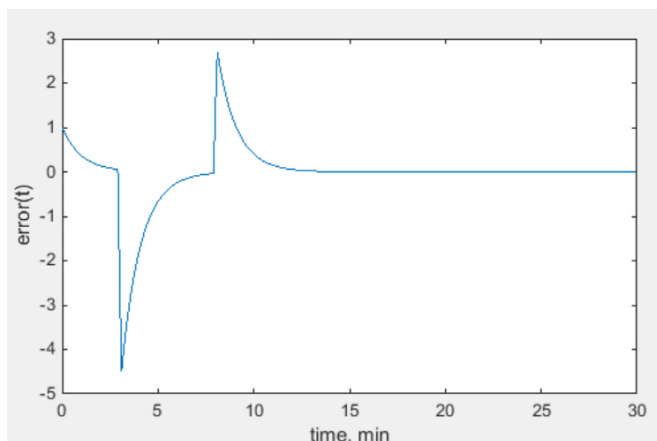
Μετά την εκτέλεση του κώδικα έχουμε τα εξής αποτελέσματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Πληκτρολογώντας στο Command Window y' (ή u' , y_{sp}' κτλ.) μας εμφανίζει ως διάνυσμα στήλης τα αποτελέσματα της $y(t)$.

```
>> y'
```

```
ans =
```

```
0
0.0952
0.1813
0.2592
0.3297
0.3935
0.4512
0.5034
0.5507
0.5934
0.6321
0.6671
0.6988
0.7275
0.7534
```

Στο συγκεκριμένο παράδειγμα εμφανίζεται ένα διάνυσμα (301x1) με τις πρώτες δεκαπέντε γραμμές (αποτελέσματα) όπως εμφανίζονται στην εικόνα δεξιά.

Τέλος για να υπολογίσουμε το τελικό σφάλμα πληκτρολογούμε τις εξής εντολές:

```
>> [x1,y1]=size(y);
>> Error=y-sp(y1)-y(y1)
```

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/1](#).

Εξάσκηση:

Βάσει του παραπάνω παραδείγματος:

1. Σχεδιάστε ένα κατάλληλο ελεγκτή ανατροφοδότησης χρησιμοποιώντας διαφορετικές Y_{sp} και D .
2. Σχεδιάστε ένα κατάλληλο ελεγκτή ανατροφοδότησης ορίζοντας διαφορετικές τιμές για τα K , τ και λ .

Παράδειγμα 1.1.2: Σχεδιασμός Ελεγκτή ανατροφοδότησης (2^{ης} τάξης)

Να σχεδιάσετε έναν κατάλληλο ελεγκτή ανατροφοδότησης με βάση την μεθοδολογία Morari και Zafiriou εάν γνωρίζεται ότι $Gp(s)=K/(t^2 s^2+2\zeta ts+1)$ και $Gd(s)=1$, με $K=5$, $\zeta=3/4$, $t=2$ και $\lambda=1$.

Επίλυση: Η άσκηση έχει ίδια λύση με το Παράδειγμα 1 που παρουσιάστηκε παραπάνω, με τη διαφορά ότι η $Gp(s)$ έχει άλλη μορφή.

Όπως στο προηγούμενο παράδειγμα έτσι κι εδώ χρησιμοποιούμε τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε ξανά με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα, δηλώνουμε τις σταθερές K , ζ , t και λ και προσδιορίζουμε τον ελεγκτή μας. Σε περίπτωση που επιλέξουμε να γράψουμε την $Gc(s)$ σαν μια γενική μορφή των $Gp^-(s)$ και $Gp^+(s)$, θα πρέπει να τις ορίσουμε σε αυτό το σημείο, καθώς και τη σταθερά r ($r=2-0=2$).

$$Gp(s) = Gp^-(s) \cdot Gp^+(s)$$

με $Gp^+(s) = 1$ και

$$Gc(s) = \frac{1}{Gp^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - Gp^+(s)}$$

1^{ος} τρόπος

```
%K=5;t=2;z=3/4;lamda=1;r=2;

%Gp=K/((t^2)*(s^2)+2*z*t*s+1);
%Gp_plus=1;
%Gp_minus=Gp;

%Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
%Gd=1;
```

Διαφορετικά με αντικατάσταση των $Gp^-(s)$ και $Gp^+(s)$ προκύπτει:

$$Gc(s) = \frac{\zeta\tau}{K\lambda} \left(1 + \frac{1}{2\zeta\tau s} + \frac{\tau}{2\zeta}s\right) \frac{1}{\left(\frac{\lambda}{2}s + 1\right)}$$

Πληκτρολογούμε τις παρακάτω εντολές για τον προσδιορισμό του ελεγκτή μας:

2^{ος} τρόπος

```
K=5;t=2;z=3/4;lamda=1;

Gp=K/((t^2)*(s^2)+2*z*t*s+1);
Gc=(z*t/(K*lamda))*(1+(1/(2*z*t*s)))+(t/(2*z))*s*(1/((lamda/2)*s+1));
```

Δηλώνουμε την συνάρτηση μεταφοράς της διαταραχής G_d , και τις εισόδους Y_{sp} (σήμα αναφοράς) και D (διαταραχή).

```
Gd=1;  
  
Ysp=(1/s)*(1+3*exp(-8*s));  
D=(5/s)*exp(-3*s);
```

Στη συνέχεια, δηλώνουμε τις γνωστές σχέσεις για το σήμα εξόδου Y , για το σήμα εισόδου στη διεργασία U και το σφάλμα E (Error). Με την εντολή `ilaplace` (αντίστροφος μετασχηματισμός Laplace, βρίσκουμε τις συναρτήσεις $y(t)$, $u(t)$ και $error(t)$.

```
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D  
Error=Ysp-Y;  
U=Gc*Error;  
  
y=ilaplace(Y);  
u=ilaplace(U);  
error=ilaplace(Error);
```

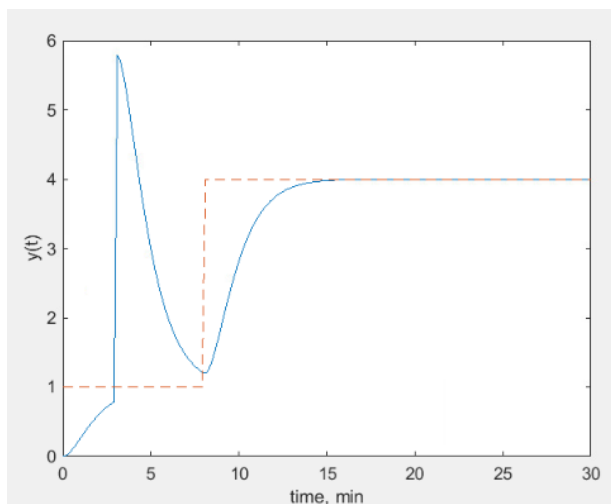
Στο σημείο αυτό θα πρέπει να δηλώσουμε την διάρκεια του χρόνου. Επιπλέον χρησιμοποιούμε τις γνωστές εντολές `subs` και `double` για να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή διαγραμμάτων.

```
Total_Time=30;  
t=0:0.1:Total_Time;  
  
y=subs(y,t);  
y=double(y);  
  
u=subs(u,t);  
u=double(u);  
  
error=subs(error,t);  
error=double(error);  
  
ysp1=ilaplace(Ysp);  
ysp1=subs(ysp1,t);  
ysp1=double(ysp1);
```

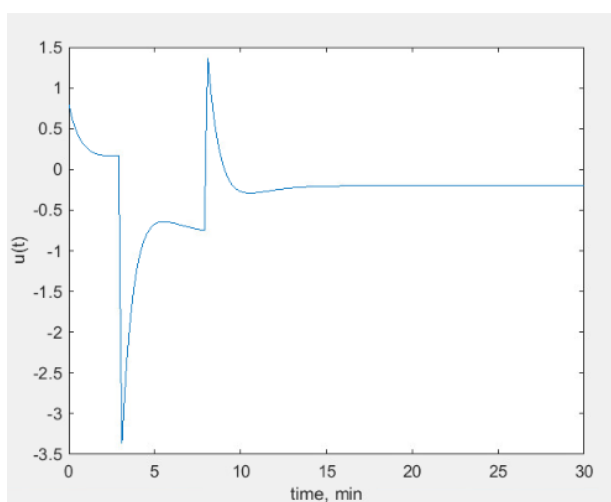
Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```
ysp=ysp1;  
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')  
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')  
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

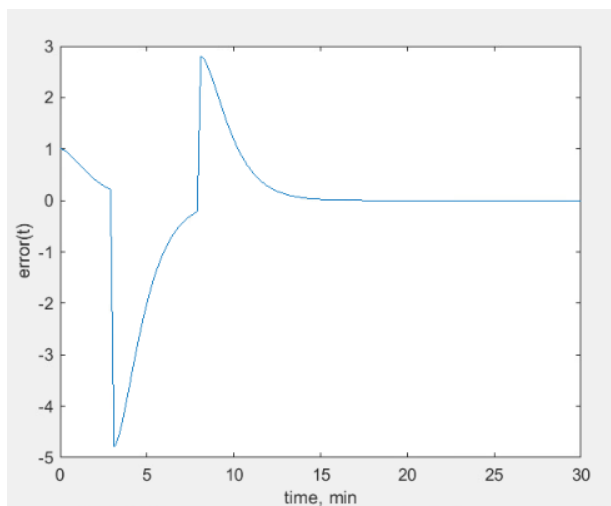
Μετά την εκτέλεση του κώδικα έχουμε τα εξής αποτελέσματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/2](#).

Παράδειγμα 1.1.3: Σχεδιασμός Ελεγκτή ανατροφοδότησης (αρνητικό μηδενικό)

Να σχεδιάσετε έναν κατάλληλο ελεγκτή ανατροφοδότησης με βάση την μεθοδολογία Morari και Zafiriou εάν γνωρίζεται ότι $Gp(s)=\dots$ και $Gd(s)=1$, με $K=5$, $\tau_a=3$, $\tau_1=2$, $\tau_2=5$ και $\lambda=1$.

Επίλυση: Η άσκηση έχει παρόμοια λύση με τα προηγούμενα παραδείγματα, με τη διαφορά ότι εδώ, έχουμε την ύπαρξη αρνητικού μηδενικού (ρίζα στον αριθμητή της συνάρτησης μεταφοράς $Gp(s)$). Επομένως η $Gp(s)$ είναι διαφορετική όπως φαίνεται παρακάτω:

$$Gp(s) = \frac{K(\tau_a s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)} = Gp^-(s) \cdot Gp^+(s)$$

με $Gp^+(s) = 1$, $Gp^-(s) = Gp(s)$ και $r = 2 - 1 = 1$.

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα δηλώνουμε τις σταθερές K , τ_a , τ_1 , τ_2 , r και λ , τη συνάρτηση μεταφοράς $Gp(s)$, τις συναρτήσεις ελάχιστης $Gp^-(s)$ και μη ελάχιστης φάσης $Gp^+(s)$, καθώς και την $Gc(s)$ του ελεγκτή μας ως μια γενική μορφή των $Gp^-(s)$ και $Gp^+(s)$.

$$Gc(s) = \frac{1}{Gp^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - Gp^+(s)}$$

```
K=5;ta=3;t1=2;t2=5;r=1;lamda=1;
Gp=K*(ta*s+1)/((t1*s+1)*(t2*s+1));

Gp_plus=1;
Gp_minus=Gp;

Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
```

Ορίζουμε την συνάρτηση μεταφοράς της διαταραχής Gd , και τις εισόδους Ysp (σήμα αναφοράς) και D (διαταραχή).

```
Gd=1;

Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```


Στη συνέχεια, δηλώνουμε τις γνωστές σχέσεις για το σήμα εξόδου Y , για το σήμα εισόδου στη διεργασία U και το σφάλμα E (Error). Με την εντολή `ilaplace` (αντίστροφος μετασχηματισμός Laplace, βρίσκουμε τις συναρτήσεις $y(t)$, $u(t)$ και $error(t)$.

```
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;

y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
```

Συνεχίζοντας, προσδιορίζουμε τη διάρκεια του χρόνου και χρησιμοποιούμε τις γνωστές εντολές `subs` και `double`, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή διαγραμμάτων.

```
Total_Time=30;
t=0:0.1:Total_Time;

y=subs(y,t);
y=double(y);

u=subs(u,t);
u=double(u);

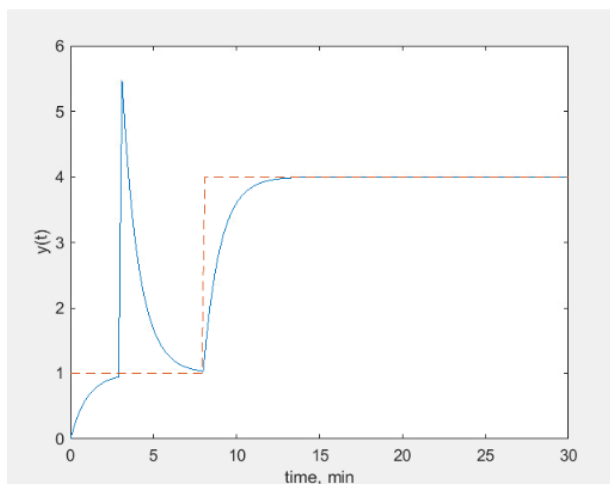
error=subs(error,t);
error=double(error);

ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
```

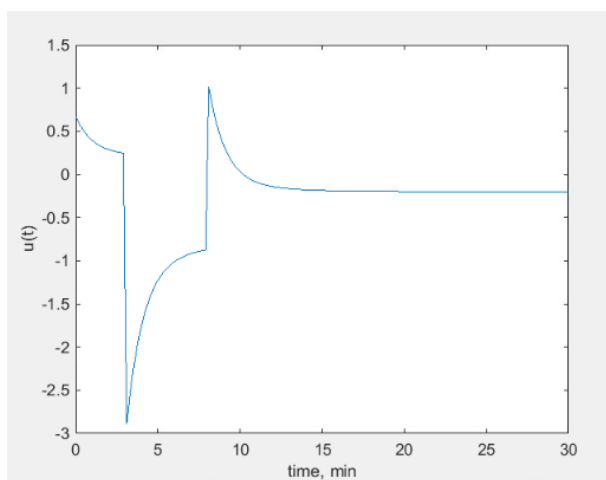
Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

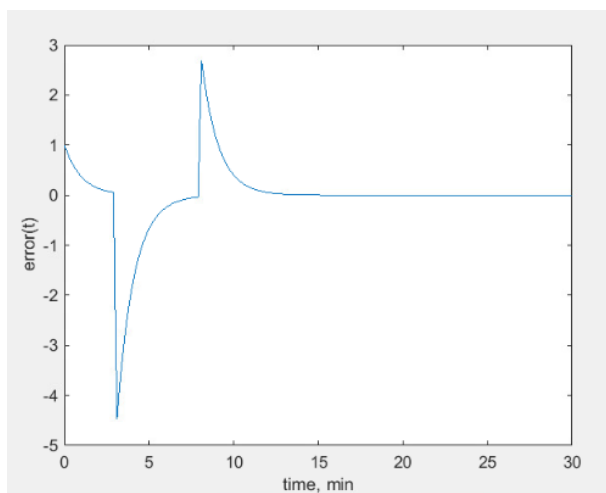
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/3](#).

Παράδειγμα 1.1.4: Σχεδιασμός Ελεγκτή (θετικό μηδενικό)

Να σχεδιάσετε έναν κατάλληλο ελεγκτή με βάση την μεθοδολογία Morari και Zafiriou εάν γνωρίζετε ότι $Gp(s)=\dots$ και $Gd(s)=1$ $\tau_a=3$, $\tau_1=2$, $\tau_2=5$, $K=5$, $\lambda=1$.

Επίλυση: Στο παράδειγμα αυτό, η συνάρτηση μεταφοράς $Gp(s)$ είναι συνδυασμός ελάχιστης και μη ελάχιστης φάσης (ύπαρξη θετικών μηδενικών (ρίζες)). Η $Gp(s)$ έχει την εξής μορφή:

$$Gp(s) = \frac{K(-\tau_a s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)} = Gp^-(s) \cdot Gp^+(s)$$

Όπου,

$$Gp^-(s) = \frac{K(\tau_a s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

$$Gp^+(s) = \frac{(1 - \tau_a s)}{(1 + \tau_a s)}$$

και $r = 2 - 1 = 1$.

Η $Gp^+(s)$ ορίστηκε κατά αυτόν τον τρόπο καθώς αποτελεί το γινόμενο των συναρτήσεων μεταφοράς που είναι μη ελάχιστης φάσης και ισχύει $|Gp^+(s=i\omega)|=1$.

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα δηλώνουμε τις σταθερές K , τ_a , τ_1 , τ_2 , r και λ , τη συνάρτηση μεταφοράς $Gp(s)$, τις συναρτήσεις ελάχιστης $Gp^-(s)$ και μη ελάχιστης φάσης $Gp^+(s)$, καθώς και την $Gc(s)$ του ελεγκτή μας ως μια γενική μορφή των $Gp^-(s)$ και $Gp^+(s)$.

$$Gc(s) = \frac{1}{Gp^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - Gp^+(s)}$$

```
K=5;ta=3;t1=2;t2=5;r=1;lamda=1;

Gp=K*(-ta*s+1)/((t1*s+1)*(t2*s+1));
Gp_plus=(1-ta*s)/(1+ta*s);
Gp_minus=K*(ta*s+1)/((t1*s+1)*(t2*s+1));

Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
```

Ορίζουμε την συνάρτηση μεταφοράς της διαταραχής G_d , και τις εισόδους Y_{sp} (σήμα αναφοράς) και D (διαταραχή).

```
Gd=1;  
  
Ysp=(1/s)*(1+3*exp(-8*s));  
D=(5/s)*exp(-3*s);
```

Στη συνέχεια, δηλώνουμε τις γνωστές σχέσεις για το σήμα εξόδου Y , για το σήμα εισόδου στη διεργασία U και το σφάλμα E (Error). Με την εντολή `ilaplace` (αντίστροφος μετασχηματισμός Laplace, αναλύθηκε στα «Συστήματα Ελέγχου Ι») βρίσκουμε τις συναρτήσεις $y(t)$, $u(t)$ και $error(t)$.

```
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D  
Error=Ysp-Y;  
U=Gc*Error;  
  
y=ilaplace(Y);  
u=ilaplace(U);  
error=ilaplace(Error);
```

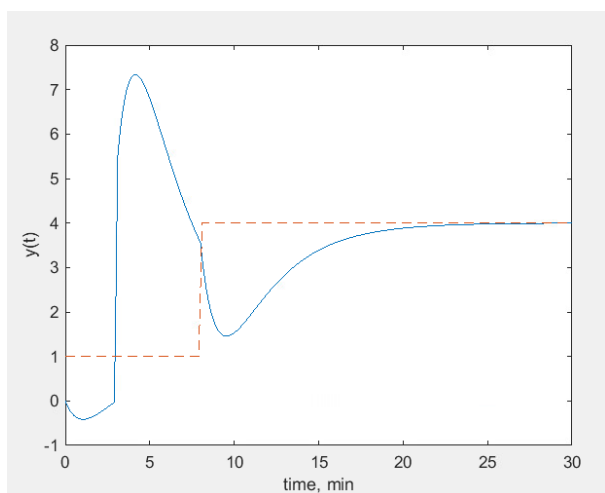
Συνεχίζοντας, προσδιορίζουμε τη διάρκεια του χρόνου και χρησιμοποιούμε τις γνωστές εντολές `subs` και `double`, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή διαγραμμάτων.

```
Total_Time=30;  
t=0:0.1:Total_Time;  
  
y=subs(y,t);  
y=double(y);  
  
u=subs(u,t);  
u=double(u);  
  
error=subs(error,t);  
error=double(error);  
  
ysp1=ilaplace(Ysp);  
ysp1=subs(ysp1,t);  
ysp1=double(ysp1);
```

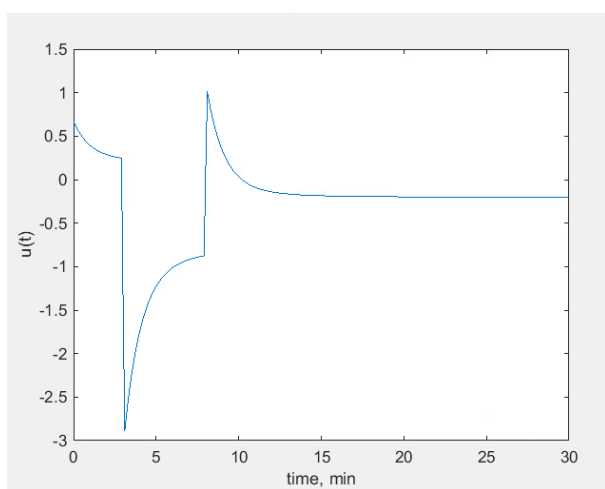
Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```
ysp=ysp1;  
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')  
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')  
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

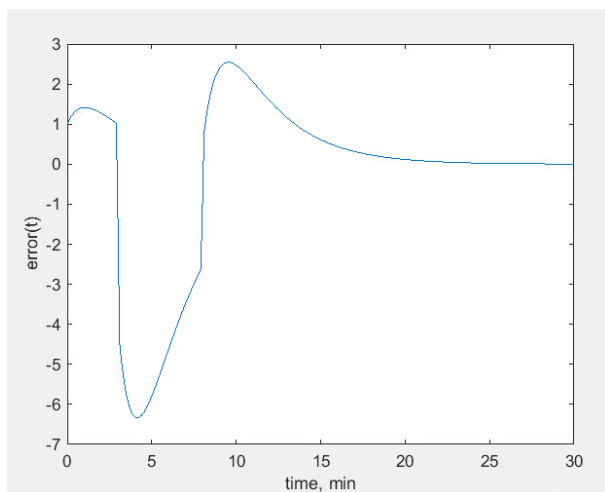
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/4](#).

Παράδειγμα 1.1.5: Σχεδιασμός Ελεγκτή ανατροφοδότησης (νεκρός χρόνος)

Να σχεδιάσετε έναν κατάλληλο ελεγκτή με βάση την μεθοδολογία Morari and Zafiriou εάν γνωρίζετε ότι $G_p(s)=\dots$ και $G_d(s)=1$ $K=5$, $\tau=3$, $\theta=2$, $\lambda=1$; Χρησιμοποιήστε προσέγγιση Pade.

Επίλυση: Η μορφή της συνάρτησης μεταφοράς $G_p(s)$ λόγω ύπαρξης νεκρού χρόνου είναι και σε αυτό το παράδειγμα συνδυασμός ελάχιστης και μη ελάχιστης φάσης.

$$G_p(s) = \frac{K}{(\tau s + 1)} \cdot e^{-\theta s} = G_p^-(s) \cdot G_p^+(s)$$

Όπου,

$$G_p^-(s) = \frac{K}{(\tau s + 1)}$$

$$G_p^+(s) = e^{-\theta s}$$

και $r = 1 - 0 = 1$.

Ξεκινάμε με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα δηλώνουμε τις σταθερές K , τ , θ , r και λ , τη συνάρτηση μεταφοράς $G_p(s)$, τις συναρτήσεις ελάχιστης $G_p^-(s)$ και μη ελάχιστης φάσης $G_p^+(s)$, καθώς και την $G_c(s)$ του ελεγκτή μας ως μια γενική μορφή των $G_p^-(s)$ και $G_p^+(s)$.

$$G_c(s) = \frac{1}{G_p^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - G_p^+(s)}$$

```
K=5;tau=3;theta=2;r=1;lamda=1;

Gp=(K/(tau*s+1))*(1-theta*0.5*s)/(1+theta*0.5*s);
Gp_plus=(1-theta*0.5*s)/(1+theta*0.5*s);
Gp_minus=K/(tau*s+1);
Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
```

Ορίζουμε την συνάρτηση μεταφοράς της διαταραχής G_d , και τις εισόδους Y_{sp} (σήμα αναφοράς) και D (διαταραχή).

```
Gd=1;

Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```

Στη συνέχεια, δηλώνουμε τις γνωστές σχέσεις για το σήμα εξόδου Y , για το σήμα εισόδου στη διεργασία U και το σφάλμα E (Error). Με την εντολή `ilaplace` (αντίστροφος μετασχηματισμός Laplace, αναλύθηκε στα «Συστήματα Ελέγχου Ι») βρίσκουμε τις συναρτήσεις $y(t)$, $u(t)$ και $error(t)$.

```

Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;

y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);

```

Συνεχίζοντας, προσδιορίζουμε τη διάρκεια του χρόνου και χρησιμοποιούμε τις γνωστές εντολές subs και double, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή διαγραμμάτων.

```

Total_Time=30;
t=0:0.1:Total_Time;

y=subs(y,t);
y=double(y);

u=subs(u,t);
u=double(u);

error=subs(error,t);
error=double(error);

ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);

```

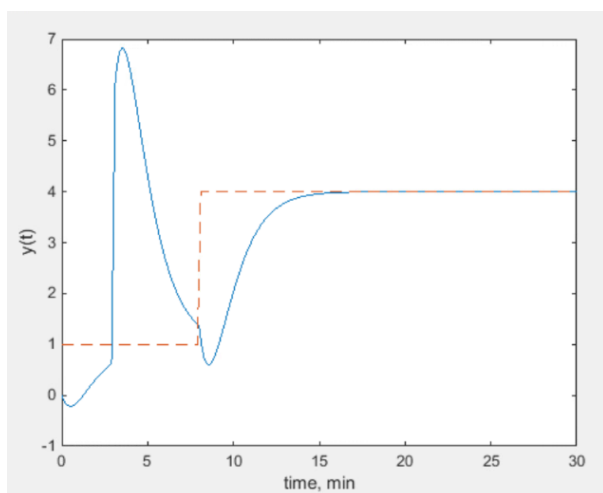
Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης y(t) και των σημάτων u και error καθώς και την απόκριση της ysp(t):

```

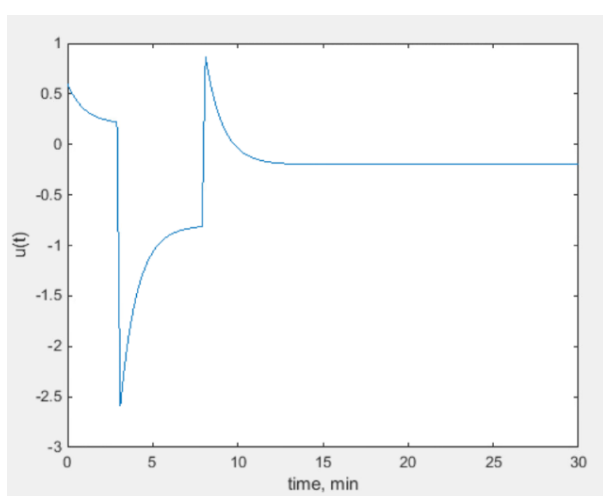
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')

```

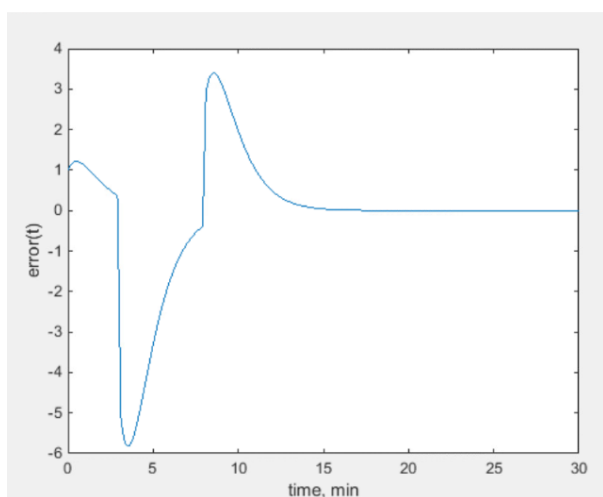
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/5](#).

Παράδειγμα 1.1.6: Σχεδιασμός Ελεγκτή ανατροφοδότησης (n-τάξης)

Να σχεδιάσετε έναν κατάλληλο ελεγκτή με βάση την μεθοδολογία Morari και Zafiriou εάν γνωρίζετε ότι $Gp(s)=\dots$ και $Gd(s)=1$ $K=5$, $t_1=2$, $t_2=3$, $t_3=5$, $\lambda=1$.

Επίλυση: Θα εργαστούμε όπως και στα προηγούμενα παραδείγματα αλλά με διαφορετική $Gp(s)$ (n-τάξεως με $r=3$):

$$Gp(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)(\tau_3 s + 1)} = Gp^-(s) \cdot Gp^+(s)$$

Όπου, $Gp^+(s) = Gp(s)$, $Gp^-(s) = 1$ και $r = 3 - 0 = 3$

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα δηλώνουμε τις σταθερές K , t_1 , t_2 , t_3 , r και λ , τη συνάρτηση μεταφοράς $Gp(s)$, τις συναρτήσεις ελάχιστης $Gp^-(s)$ και μη ελάχιστης φάσης $Gp^+(s)$, καθώς και την $Gc(s)$ του ελεγκτή μας ως μια γενική μορφή των $Gp^-(s)$ και $Gp^+(s)$.

$$Gc(s) = \frac{1}{Gp^-(s)} \cdot \frac{1}{(\lambda s + 1)^r - Gp^+(s)}$$

```
K=5;t1=2;t2=3;t3=5;r=3;lamda=1;

Gp=K/((t1*s+1)*(t2*s+1)*(t3*s+1));
Gp_plus=1;
Gp_minus=Gp;

Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus))
```

Ορίζουμε την συνάρτηση μεταφοράς της διαταραχής Gd , και τις εισόδους Ysp (σήμα αναφοράς) και D (διαταραχή).

```
Gd=1;

Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```

Στη συνέχεια, δηλώνουμε τις γνωστές σχέσεις για το σήμα εξόδου Y , για το σήμα εισόδου στη διεργασία U και το σφάλμα E (Error). Με την εντολή `ilaplace` (αντίστροφος μετασχηματισμός Laplace, βρίσκουμε τις συναρτήσεις $y(t)$, $u(t)$ και $error(t)$.

```

Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;

y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);

```

Συνεχίζοντας, προσδιορίζουμε τη διάρκεια του χρόνου και χρησιμοποιούμε τις γνωστές εντολές subs και double, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή διαγραμμάτων.

```

Total_Time=30;
t=0:0.1:Total_Time;

y=subs(y,t);
y=double(y);

u=subs(u,t);
u=double(u);

error=subs(error,t);
error=double(error);

ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);

```

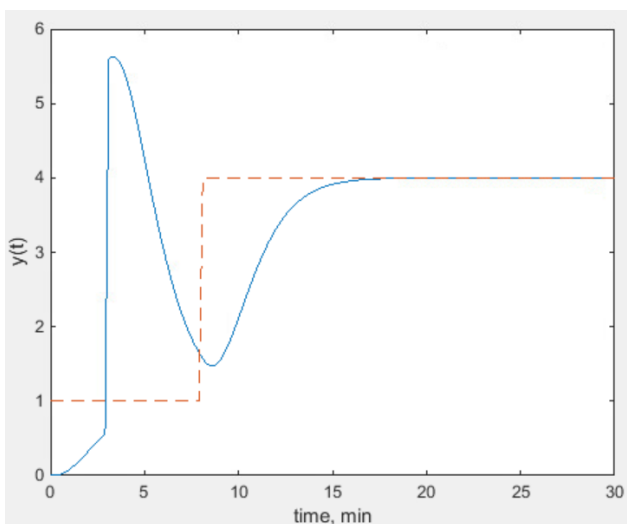
Τέλος προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```

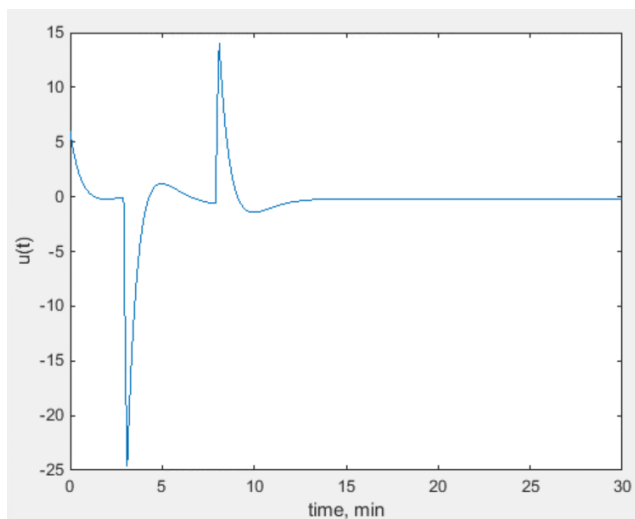
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u,'--');xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error,'--');xlabel('time, min');ylabel('error(t)')

```

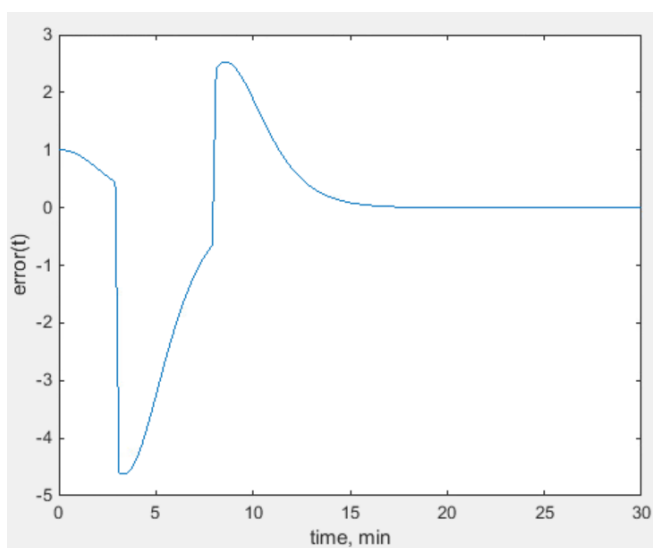
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

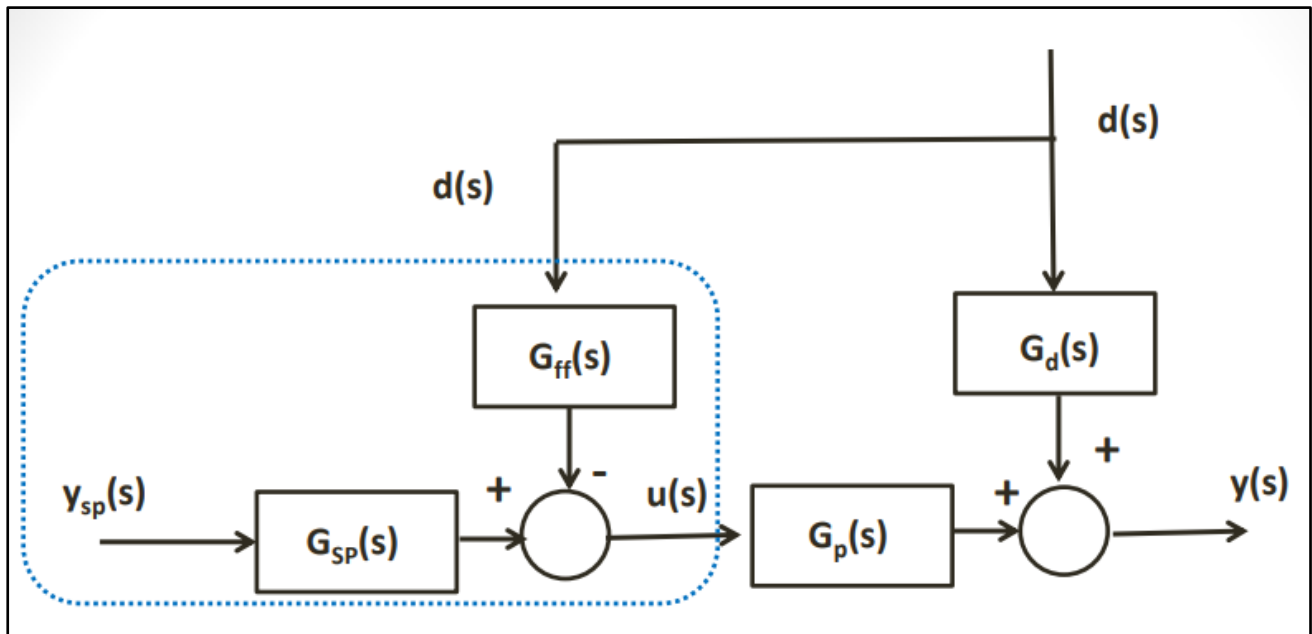
Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [1.1/6](#):

ΚΕΦΑΛΑΙΟ 2: Ειδικές Περιπτώσεις Συστημάτων Ελέγχου

Στο 2^ο Κεφάλαιο θα παρουσιασθούν παραδείγματα και εφαρμογές για ειδικές περιπτώσεις Συστημάτων Ελέγχου. Συγκεκριμένα θα επιλυθούν παραδείγματα σχεδιασμού ελεγκτών πρόδρασης, πρόδρασης/ανάδρασης και συστοιχίας στο περιβάλλον της πλατφόρμας MATLAB

2.1 Έλεγχος Πρόδρασης

Κατά τον έλεγχο πρόδρασης εισάγονται δύο νέες συναρτήσεις μεταφοράς οι $G_{sp}(s)$ και $G_{ff}(s)$, όπως αναλύθηκαν στην θεωρία. Ειδικότερα, με τον ελεγκτή πρόδρασης $G_{ff}(s)$ αξιοποιούμε την διαταραχή $d(s)$ και η συνάρτηση $G_{sp}(s)$ αποτελεί τον αντισταθμιστή πρόδρασης.



Εικόνα 2.1, Έλεγχος πρόδρασης

Παρακάτω θα αναλυθούν παραδείγματα σχεδιασμού ελεγκτή πρόδρασης και θα αναπαρασταθούν γραφικά τα σήματα εισόδου και εξόδου καθώς και το σφάλμα, συναρτήσει του χρόνου.

Παράδειγμα 2.1.1: Σχεδιασμός Ελεγκτή πρόδρασης

Να σχεδιάσετε έναν κατάλληλο ελεγκτή πρόδρασης (G_{ff} , G_{sp}) εάν γνωρίζετε ότι $G_p(s)=K/(ts+1)$ και $G_d(s)=1/(ts+1)$ με $K=5$, $t=2$ και $\lambda=1$.

Από την θεωρία γνωρίζουμε ότι:

$$y(s) = G_p(s) \cdot G_{sp}(s) \cdot y_{sp}(s) + [G_d(s) - G_p(s) \cdot G_{ff}(s)] \cdot d(s),$$

$$G_{sp,min}(s) = \frac{1}{G_p^-(s)}$$

και

$$G_{ff,min}(s) = \frac{s}{G_p^-(s)} \left\{ \frac{G_d(s)}{s \cdot G_p^+(s)} \right\}^*$$

Επίλυση: Αρχικά υπολογίζουμε στο χαρτί μας βάσει της θεωρίας τις συναρτήσεις ελάχιστης $Gp^-(s)$ και μη ελάχιστης φάσης $Gp^+(s)$ καθώς και τις συναρτήσεις $G_{sp}(s)$ και $G_{ff}(s)$ χρησιμοποιώντας τις παρακάτω σχέσεις:

$$Gp(s) = Gp^-(s) \cdot Gp^+(s)$$

όπου

$$Gp^+(s)(s) = 1$$

$$Gp^-(s) = Gp(s) = \frac{K}{\tau s + 1}$$

Επομένως,

$$G_{sp,min}(s) = \frac{\tau s + 1}{K} \rightarrow \frac{\tau s + 1}{K(\lambda s + 1)}$$

και

$$G_{ff,min}(s) = \frac{1}{K}$$

ΣΗΜΕΙΩΣΗ: Η εισαγωγή της σταθερά λ οφείλεται στο γεγονός ο βαθμός του αριθμητή στη $G_{sp,min}(s)$ είναι μεγαλύτερος από τον βαθμό του παρονομαστή, πράγμα που καθιστά τον ελεγκτή μας μη-πραγματοποιήσιμο.

Όπως στο προηγούμενο κεφάλαιο έτσι κι εδώ χρησιμοποιούμε τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε ξανά με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Στη συνέχεια, δηλώνουμε τις σταθερές K , τ και λ , την συνάρτηση μεταφοράς της διεργασίας Gp , της διαταραχής Gd , την συνάρτηση μεταφοράς του ελεγκτή πρόδρασης Gff και τον αντισταθμιστή πρόδρασης Gsp , όπως υπολογίσαμε παραπάνω.

```
K=5;tau=2;lamda=1;

Gp=K/(tau*s+1);
Gd=1/(tau*s+1);
Gsp=(tau*s+1)/(K*(lamda*s+1));
Gff=1/K;
```

Ορίζουμε τις εισόδους Ysp (σήμα αναφοράς) και D (διαταραχή). Μπορούμε να επιλέξουμε διάφορες μεταβολές σε αυτά τα σήματα.

```
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```

Στο επόμενο βήμα προσδιορίζουμε τις σχέσεις κλειστού βρόχου $Y(s)$ και $y(t)$, το σφάλμα E και το σήμα εισόδου U που μεταβάλλεται από τον ελεγκτή μας και δηλώνουμε την διάρκεια του χρόνου:

$$Y = G_p(s) \cdot G_{sp}(s) \cdot Y_{sp}(s) + [G_d(s) - G_p(s) \cdot G_{ff}(s)] \cdot D(s),$$

$$E = Y_{sp}(s) - Y(s)$$

και

$$U = G_{sp}(s) \cdot Y_{sp}(s) - G_{ff}(s) \cdot D(s)$$

```
Y=(Gp*Gsp)*Ysp+(Gd-Gp*Gff)*D
U=Gsp*Ysp-Gff*D;
Error=Ysp-Y;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);

Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
```

Έπειτα, προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $y_{sp}(t)$:

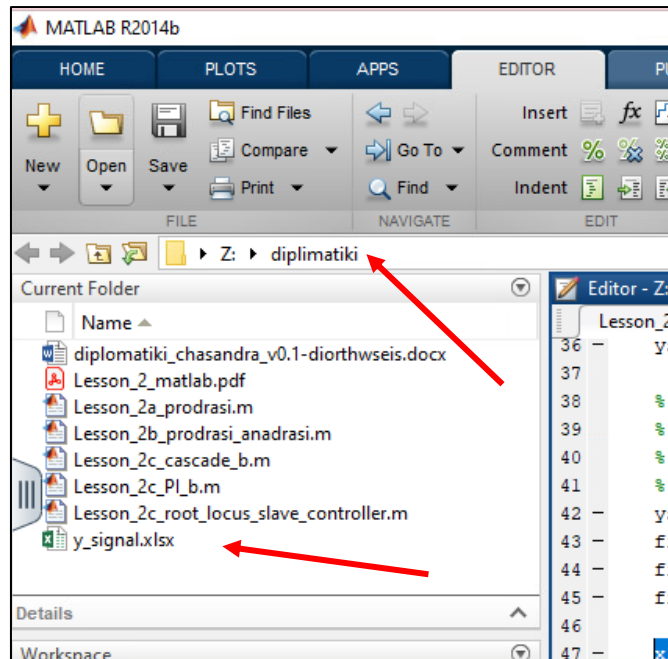
```
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

ΣΗΜΕΙΩΣΗ: Τα όρια των αξόνων X και Y μπορούμε να τα μεταβάλλουμε όπως θέλουμε με την εντολή `axis` (π.χ `figure(1);plot(t,y,t,ysp,'--');` **`axis ([0 100 0 100]);`** `xlabel('time, min');` `ylabel('y(t)')`)

Στο τέλος κάθε κώδικα πληκτρολογώντας την εντολή `xlswrite`, έχουμε την δυνατότητα να δημιουργήσουμε ένα αρχείο excel με τις τιμές των μεταβλητών μας.

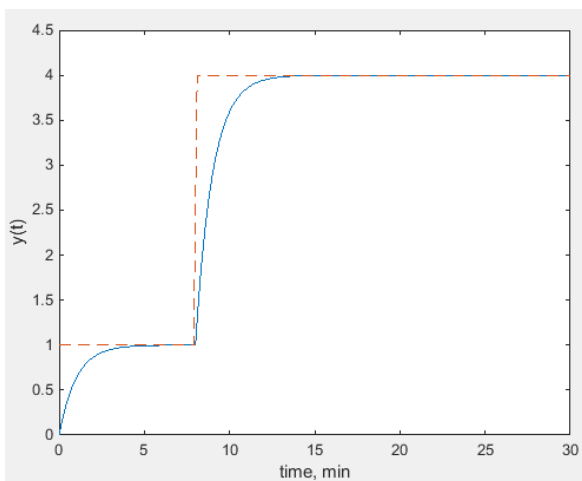
```
xlswrite('y_signal.xlsx',[t' y' ysp'])
```

Στο φάκελο που βρίσκεται και ο κώδικας του παραδείγματος δημιουργείται αυτόματα το αρχείο excel.

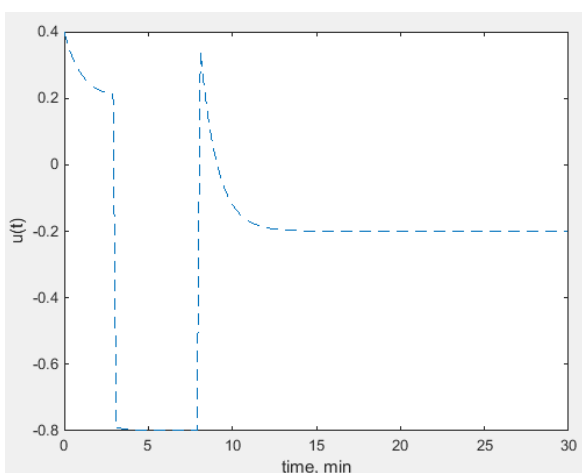


Εικόνα 2.2, Φάκελος αποθήκευσης αρχείου excel

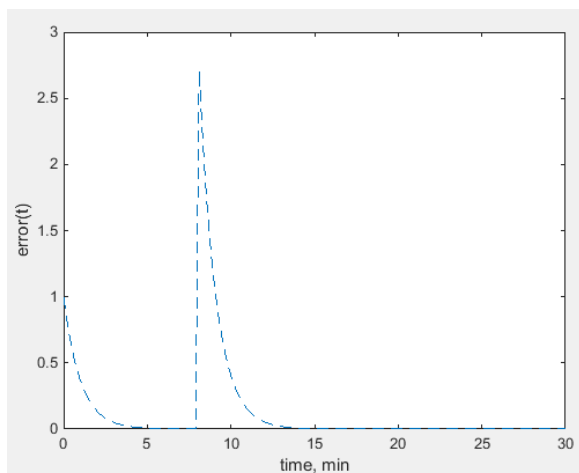
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.1/1](#).

Εξάσκηση:

Να σχεδιάσετε έναν κατάλληλο ελεγκτή πρόδρασης (G_{ff} , G_{sp}) εάν γνωρίζετε ότι:

$$G_p(s) = \frac{1-s}{(s+2) \cdot (s+3)}$$

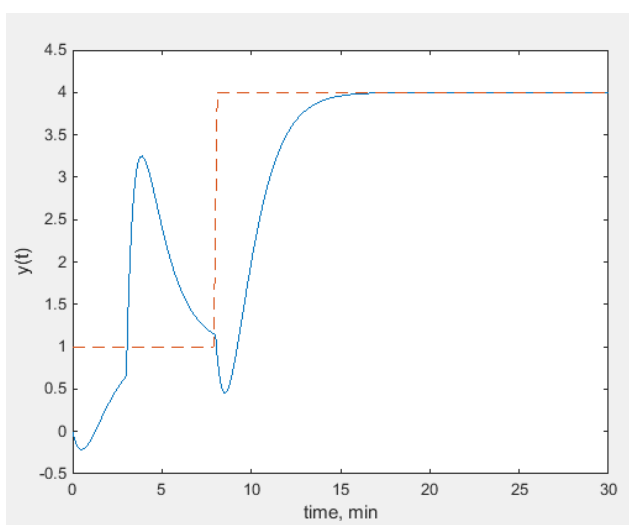
και

$$G_d(s) = \frac{1}{s+2}$$

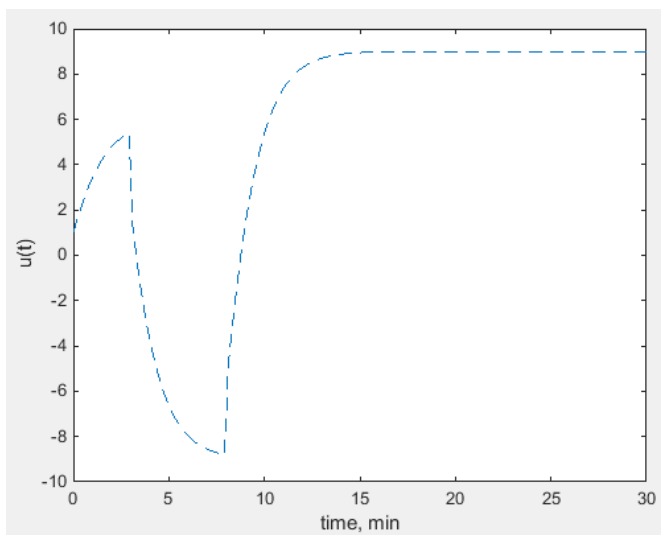
με $y_{sp}(s) = (1/s) \cdot (1 + 3 \cdot \exp(-8 \cdot s))$ και $d(s) = (5/s) \cdot \exp(-3s)$.

Επίλυση: Θα πρέπει να αντικαταστήσουμε στον κώδικα που παρουσιάστηκε προηγουμένως τις συναρτήσεις G_p και G_d καθώς και να υπολογιστούν εκ νέου οι G_{sp} και G_{ff} .

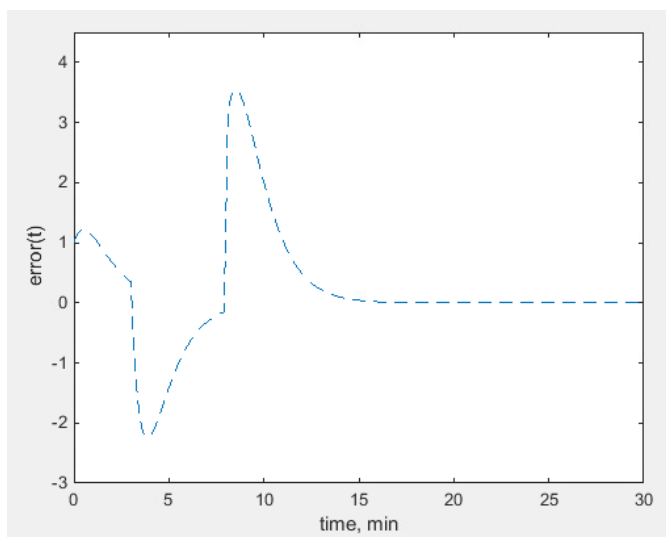
Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω αποτελέσματα:



Σήμα εξόδου Y



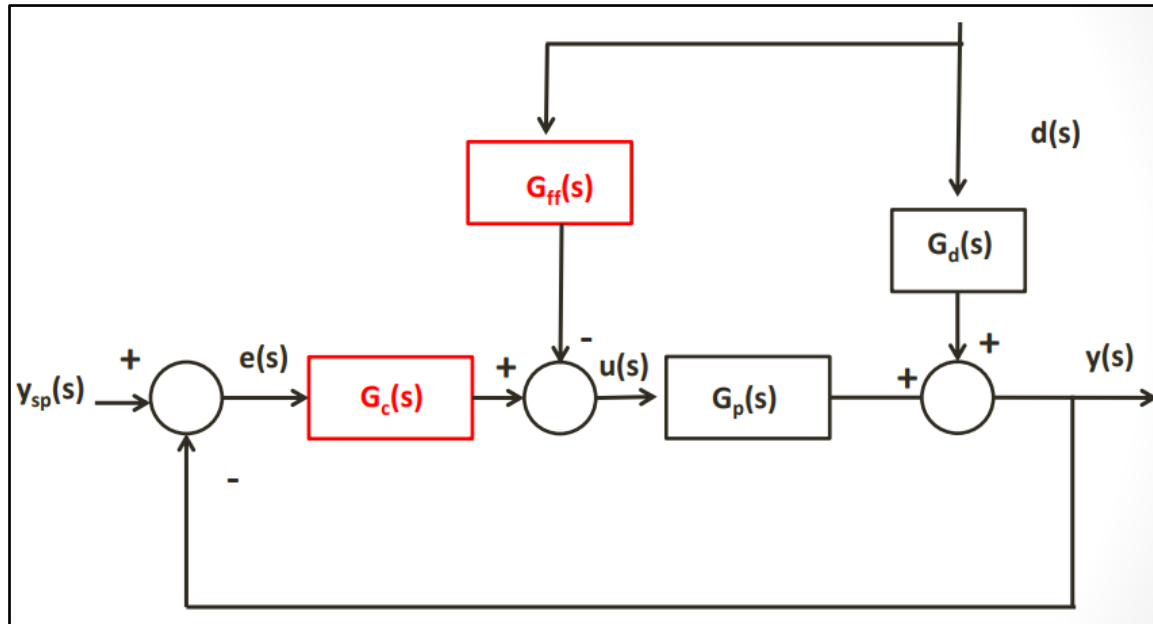
Σήμα εισόδου διεργασίας U



Σήμα σφάλματος E

2.2 Έλεγχος Πρόδρασης/Ανάδρασης

Στον έλεγχο πρόδρασης/ανάδρασης εισάγεται και πάλι η έννοια του ελεγκτή ανάδρασης $G_c(s)$.



Εικόνα 2.3, Έλεγχος Πρόδρασης/Ανάδρασης

Παρακάτω θα αναλυθούν παραδείγματα σχεδιασμού ελεγκτή πρόδρασης/ανάδρασης και θα αναπαρασταθούν γραφικά τα σήματα εισόδου και εξόδου καθώς και το σφάλμα, συναρτήσει του χρόνου.

Παράδειγμα 2.2.1: Σχεδιασμός Ελεγκτή πρόδρασης/ανάδρασης

Να σχεδιάσετε έναν κατάλληλο ελεγκτή πρόδρασης/ανάδρασης (G_c , G_{ff}) εάν γνωρίζετε ότι:

$$G_p(s) = \frac{1-s}{(s+2) \cdot (s+3)} \quad \& \quad G_d(s) = \frac{1}{s+2}$$

Επίλυση: Αρχικά υπολογίζουμε στο χαρτί μας βάσει της θεωρίας τις συναρτήσεις ελάχιστης $G_p^-(s)$ και μη ελάχιστης φάσης $G_p^+(s)$ και έπειτα τις συναρτήσεις $G_c(s)$ και $G_{ff}(s)$ χρησιμοποιώντας τις παρακάτω σχέσεις:

Από θεωρία γνωρίζουμε:

$$y(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \cdot y_{sp}(s) + \frac{G_d(s) - G_p(s) \cdot G_{ff}(s)}{1 + G_c(s)G_p(s)} \cdot d(s)$$

$$G_{c,min}(s) = \frac{1}{G_p^-(s)} \cdot \frac{1}{1 - G_p^+(s)}$$

και

$$G_{ff,min}(s) = \left[\frac{1}{G_p^-(s)} \cdot \frac{1}{1 - G_p^+(s)} \right] \cdot \left[s \cdot \left\{ \frac{G_d(s)}{s \cdot G_p^+(s)} \right\}^* - G_d(s) \right]$$

Καθώς και

$$Gp(s) = Gp^-(s) \cdot Gp^+(s)$$

όπου

$$Gp^-(s) = \frac{1+s}{(s+2) \cdot (s+3)}$$

$$Gp^+(s) = \frac{1-s}{1+s}$$

Επομένως κάνοντας τις αντικαταστάσεις και τις πράξεις καταλήγουμε στα παρακάτω:

$$G_{c,min}(s) = \frac{(s+2) \cdot (s+3)}{2s} \rightarrow \frac{(s+2) \cdot (s+3)}{2s \cdot (\lambda_1 s + 1)}$$

και

$$G_{ff,min}(s) = \frac{1}{3} \cdot (s+3) \rightarrow \frac{\frac{1}{3} \cdot (s+3)}{(\lambda_2 s + 1)}$$

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`.

```
clc
clear

syms s
```

Έπειτα δηλώνουμε τις συναρτήσεις $Gp(s)$, $Gd(s)$ τις σταθερές λ_1 και λ_2 καθώς και τις συναρτήσεις μεταφοράς του ελεγκτή πρόδρασης $Gff(s)$ και ανάδρασης $Gc(s)$, όπως τους υπολογίσαμε προηγουμένως.

ΣΗΜΕΙΩΣΗ: Στις χρονικές σταθερές λ_1 και λ_2 μπορούμε να δοκιμάσουμε διάφορες τιμές ώστε να παρατηρήσουμε τις μεταβολές στην απόκριση. Στο συγκεκριμένο παράδειγμα θεωρούμε $\lambda_1 = \lambda_2 = 0.1$.

```
%SXEDIASMOS ELEGKTI PRODRASIS/ANADRASIS GIA SYSTIMA 2is TAXIS

Gp=(1-s)/((s+2)*(s+3));
Gd=1/(s+2);
lamda1=0.1;lamda2=0.1;
Gc=((s+2)*(s+3))/((2*s)*(lamda1*s+1));
Gff=((1/3)*(s+3))/(lamda2*s+1);
```

Ορίζουμε τις εισόδους Ysp (σήμα αναφοράς) και D (διαταραχή). Μπορούμε να επιλέξουμε διάφορες μεταβολές σε αυτά τα σήματα.

```
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
```

Στο επόμενο βήμα προσδιορίζουμε τις σχέσεις κλειστού βρόχου $Y(s)$ και $y(t)$, το σφάλμα E και το σήμα εισόδου U που μεταβάλλεται από τον ελεγκτή μας και δηλώνουμε την διάρκεια του χρόνου:

$$Y = \frac{G_s(s) \cdot G_p(s)}{1 + G_c(s) \cdot G_p(s)} \cdot Y_{sp}(s) + \frac{G_d(s) - G_p(s) \cdot G_{ff}(s)}{1 + G_c(s) \cdot G_p(s)} \cdot D(s)$$

$$E = Y_{sp}(s) - Y(s)$$

και

$$U = G_c(s) \cdot E(s) - G_{ff}(s) \cdot D(s)$$

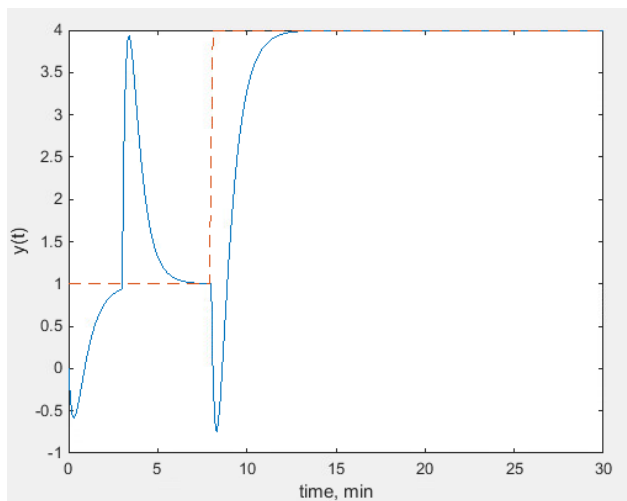
```
Y=((Gc*Gp)/(1+Gc*Gp))*Ysp+((Gd-Gp*Gff)/(1+Gc*Gp))*D
Error=Ysp-Y;
U=Gc*Error-Gff*D;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);

Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
```

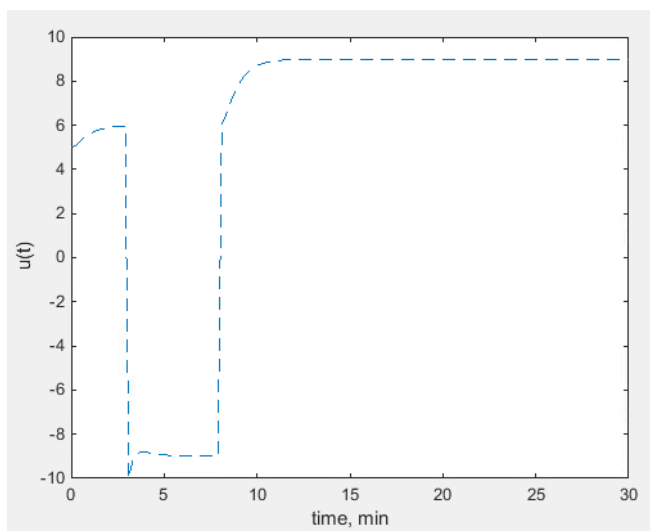
Έπειτα, προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

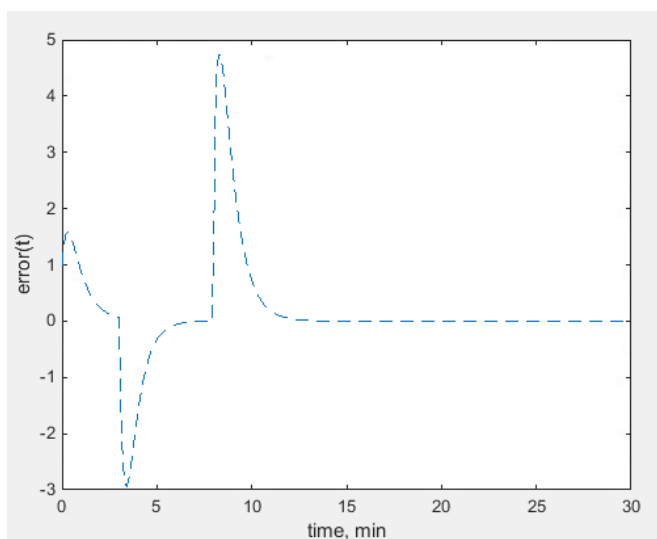
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα για $\lambda=0.1$:



Σήμα εξόδου Y

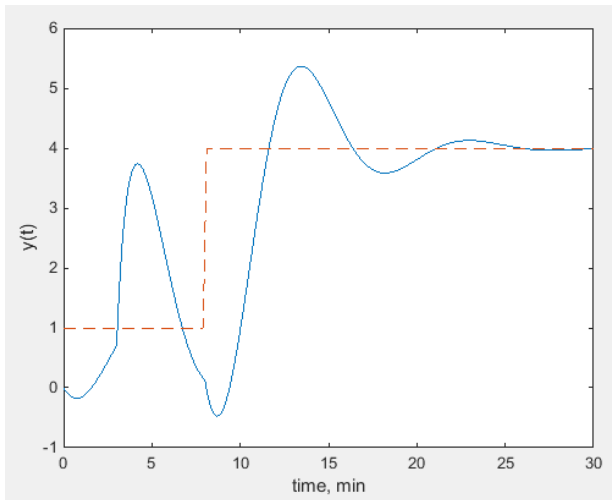


Σήμα εισόδου διεργασίας U

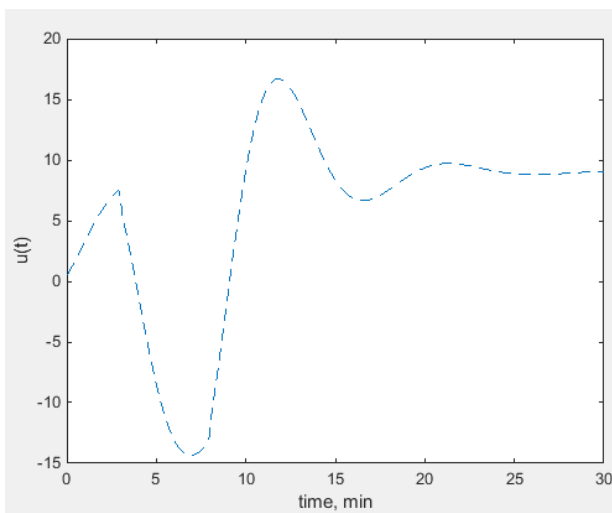


Σήμα σφάλματος E

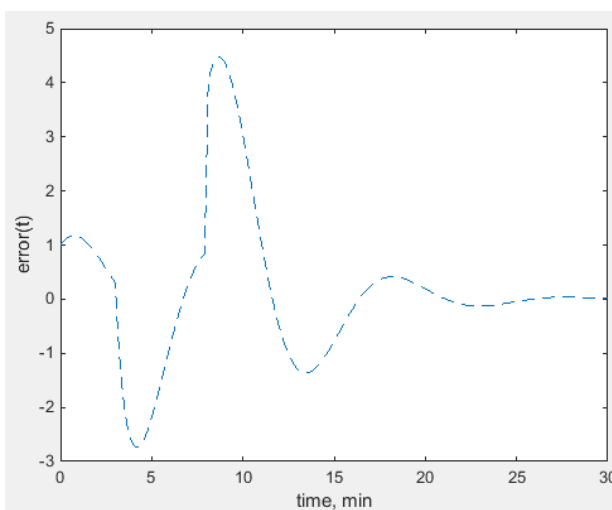
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα για $\lambda=1$:



Σήμα εξόδου Y



Σήμα εισόδου διεργασίας U

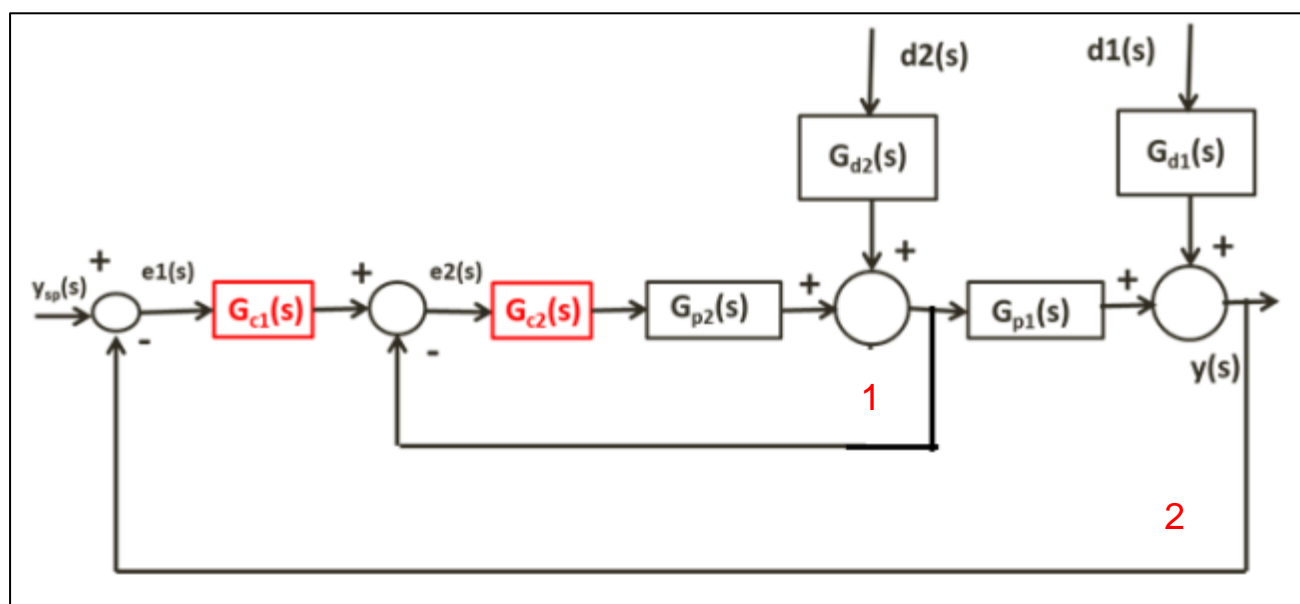


Σήμα σφάλματος E

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.2/1](#).

2.3 Έλεγχος Συστοιχίας

Ο έλεγχος συστοιχίας/αλληλουχίας (cascade control) χρησιμοποιείται συνήθως όταν υπάρχουν πολλαπλές διαταραχές και εισάγεται η έννοια του δευτερεύοντα / βοηθητικού ελεγκτή.



Εικόνα 2.4, Σημείο 1:Βοηθητικός βρόχος - Σημείο 2: Πρωτεύων / Κύριος βρόχος

Στη συνέχεια θα συγκρίνουμε ένα συμβατικό σύστημα ελέγχου ανάδρασης με ένα σύστημα ελέγχου συστοιχίας καθώς επίσης θα αναπαραστήσουμε γραφικά τα σήματα εισόδου, εξόδου και το σφάλμα, συναρτήσει του χρόνου και για τα δύο συστήματα.

Παράδειγμα 2.3.1: Σχεδιασμός και σύγκριση συστημάτων ελέγχου ανάδρασης και συστοιχίας

Να σχεδιάσετε και να συγκρίνετε ένα συμβατικό σύστημα ελέγχου ανάδρασης ($G_c(s)$) με ένα σύστημα ελέγχου συστοιχίας ($G_{c1}(s)$, $G_{c2}(s)$) εάν γνωρίζετε ότι:

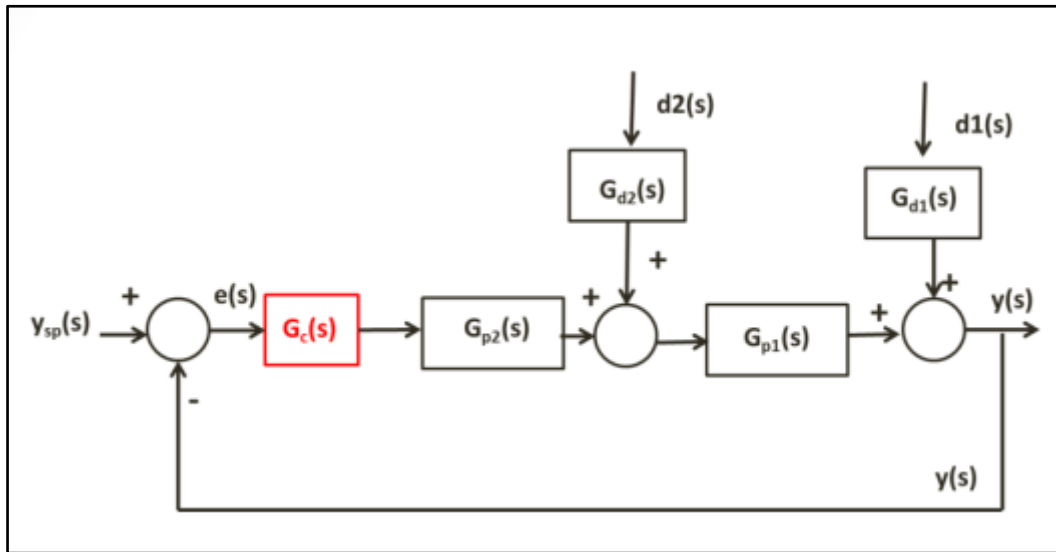
$$G_{p1}(s) = \frac{5s + 1}{(2s + 1) \cdot (4s + 1)} = G_{d1}(s)$$

$$G_{p2}(s) = \frac{1}{(0.2s^2 + 0.5s + 1)} = G_{d2}(s)$$

Επίλυση: Αρχικά θα πρέπει να σχεδιάσουμε τις συναρτήσεις μεταφοράς ($G_c(s)$, $G_{c1}(s)$, $G_{c2}(s)$) των ελεγκτών και στις δύο περιπτώσεις συστημάτων (ανάδρασης, συστοιχίας).

1^η Φάση επίλυσης

Συμβατικό Σύστημα Ελέγχου Ανάδρασης (PID ελεγκτής)



Εικόνα 2.4, Σύστημα ελέγχου ανάδρασης με ύπαρξη δύο διαταραχών

Από θεωρία στα Συστήματα Ελέγχου I γνωρίζουμε ότι:

$$G_c(s) = K_C \cdot \left(1 + \frac{1}{\tau_I s} + \tau_D s \right)$$

Με την χρήση Γεωμετρικού Τόπου των ριζών/πόλων θα βρούμε τα κρίσιμα σημεία K_{cr} και P_{cr} για να μπορέσουμε να υπολογίσουμε τις σταθερές K_c , τ_I και τ_D με τη ρύθμιση «Ziegler-Nichols».

Ξεκινάμε, με τις γνωστές εντολές *clc* και *clear all* με τις οποίες «καθαρίζουμε» το ιστορικό από τυχόν προηγούμενες εισαγωγές. Με την εντολή *syms*, δηλώνουμε ότι θα εργαστούμε με συμβολικές μεταβλητές, επομένως, δεν χρειάζεται να δώσουμε τιμές σε αυτές τις μεταβλητές. Στη συνέχεια, ορίζουμε τις συναρτήσεις μεταφοράς $G_{p1}(s)$, $G_{p2}(s)$ και $G_{d1}(s)$, $G_{d2}(s)$ και την συνάρτηση μεταφοράς του ελεγκτή $G_c(s)$ για την οποία ισχύει: $G_c(s) = K_c = 1$.

```
clc
clear

syms s

Kc=1;
Gp1= (5*s+1) / ((2*s+1) * (4*s+1)) ;

Gd1=Gp1;
Gp2=1/ (0.2*s^2+0.5*s+1) ;
Gd2=Gp2;
```

Στη συνέχεια, ορίζουμε το γινόμενο G_{o1} που απαιτείται για την εύρεση του γεωμετρικού τόπου των πόλων, σύμφωνα με την ακόλουθη σχέση

$$G_{o1} = K_c \cdot G_{p1}(s) \cdot G_{p2}(s)$$

Έπειτα, προσδιορίζουμε τον αριθμητή (num) και τον παρονομαστή (den) του παραπάνω γινομένου, με τη χρήση της εντολής *numden*. Με την εντολή *sym2poly*, θα τα μετατρέψουμε από συμβολικές μεταβλητές σε πραγματικά πολυώνυμα.

```
Gol=Kc*Gp1*Gp2;

[num, den]=numden(Gol)

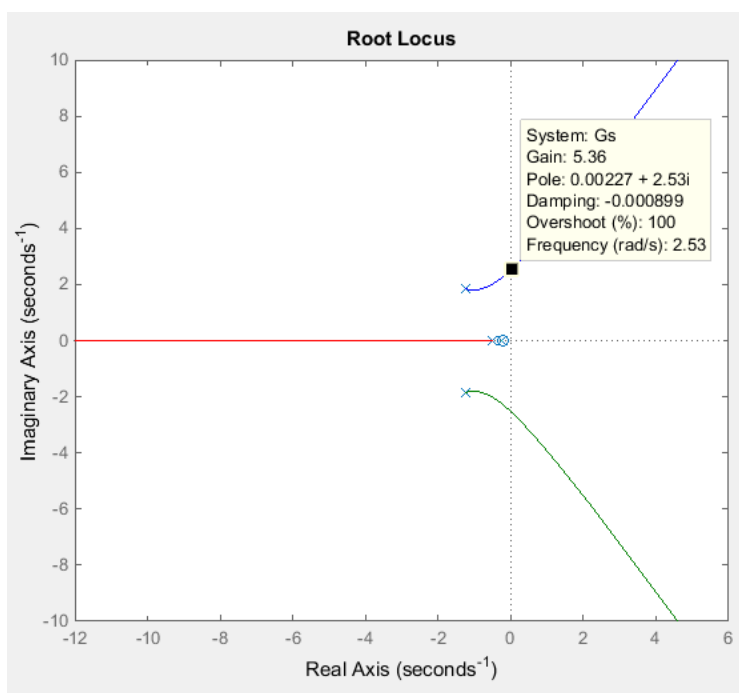
num=sym2poly(num);
den=sym2poly(den);
```

Τέλος, με την εντολή *tf(num,den)* θα δηλώσουμε την συνάρτηση μεταφοράς (transfer function) με βάση τον αριθμητή και τον παρονομαστή που βρήκαμε στο προηγούμενο βήμα. Η κατασκευή του γεωμετρικού τόπου των ριζών/πόλων γίνεται με την εντολή *rlocus*.

```
Gs=tf(num,den)

figure(1);rlocus(Gs)
```

Με την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:



Γεωμετρικός τόπος ριζών/πόλων
 $K_{cr} = 5.36$
 $P_{cr} = 2\pi/2.53 = 2.47$

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.3/1.1](#).

Στο σημείο αυτό με βάση τη ρύθμιση *Ziegler-Nichols* υπολογίζουμε τις σταθερές K_c , T_i και T_D .

Type of Controller	K_c	τ_I	τ_D
P	$0.5 \cdot K_{cr}$	∞	0
PI	$0.45 \cdot K_{cr}$	$\frac{1}{1.2} \cdot P_{cr}$	0
PID	$0.6 \cdot K_{cr}$	$0.5 \cdot P_{cr}$	$0.125 \cdot P_{cr}$



Type of Controller	K_c	τ_I	τ_D
P	2.68	∞	0
PI	2.41	2.06	0
PID	3.22	1.24	0.3

Η συνάρτηση $G_c(s)$ του PID ελεγκτή μας παίρνει τελικά την μορφή:

$$G_c(s) = 3.22 \cdot \left(1 + \frac{1}{1.24s} + 0.3s \right)$$

2^η Φάση επίλυσης

Σύστημα Ελέγχου Συστοιχίας (P & PID ελεγκτής)

Με την ίδια λογική όπως και στο σύστημα ελέγχου ανάδρασης εργαζόμαστε κι εδώ για τον βοηθητικό και τον κύριο βρόχο (Εικόνα 2.4).

Από θεωρία γνωρίζουμε ότι:

$$G_{c1}(s) = K_{c1} \cdot \left(1 + \frac{1}{\tau_I s} + \tau_D s \right)$$

και

$$G_{c2}(s) = K_{c2}$$

$$G_{c1}(s) = 2.85 \cdot \left(1 + \frac{1}{0.52s} + 0.13s \right)$$

Ξεκινάμε, με τις γνωστές εντολές *clc* και *clear all* με τις οποίες «καθαρίζουμε» το ιστορικό από τυχόν προηγούμενες εισαγωγές. Με την εντολή *syms*, δηλώνουμε ότι θα εργαστούμε με συμβολικές μεταβλητές. Στη συνέχεια, ορίζουμε τις συναρτήσεις μεταφοράς $G_{p2}(s)$ για το βοηθητικό σύστημα και τις $G_{p1}(s)$, $G_{p2}(s)$ και $G_{d1}(s)$, $G_{d2}(s)$ για το κύριο/πρωτεύων σύστημα.

<pre>% Slave Controller clc clear syms s Kc=1; Gp2=1/(0.2*s^2+0.5*s+1);</pre>	<pre>% Master Controller clc clear syms s Kc=1; Gp1=(5*s+1)/((2*s+1)*(4*s+1)); Gd1=Gp1; Gp2=1/(0.2*s^2+0.5*s+1); Gd2=Gp2;</pre>
---	---

Στη συνέχεια, ορίζουμε το γινόμενο G_{o1} που απαιτείται για την εύρεση του γεωμετρικού τόπου των πόλων, σύμφωνα με τις ακόλουθες σχέσεις:

$$G_{o1} = K_c \cdot G_{p2}(s)$$

$$G_{o1} = K_c \cdot G_{p1}(s) \cdot \frac{G_{c2}(s) \cdot G_{p2}(s)}{1 + G_{c2}(s) \cdot G_{p2}(s)}$$

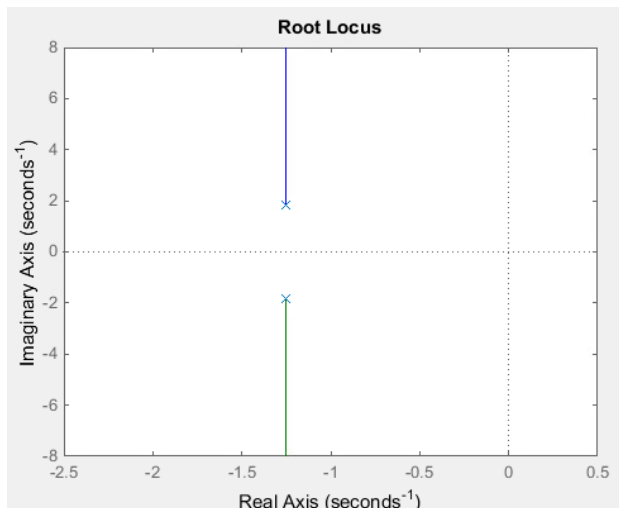
Έπειτα, προσδιορίζουμε τον αριθμητή (num) και τον παρονομαστή (den) του παραπάνω γινομένου, με τη χρήση της εντολής *numden*. Με την εντολή *sym2poly*, θα τα μετατρέψουμε από συμβολικές μεταβλητές σε πραγματικά πολυώνυμα.

<pre>% Slave Controller Gol=Kc*Gp2; [num, den]=numden(Gol) num=sym2poly(num); den=sym2poly(den);</pre>	<pre>% Master Controller Gol=Kc*Gp1*(Gc2*Gp2)/(1+Gc2*Gp2); [num, den]=numden(Gol) num=sym2poly(num); den=sym2poly(den);</pre>
--	---

Τέλος, με την εντολή *tf(num,den)* θα δηλώσουμε την συνάρτηση μεταφοράς (transfer function) με βάση τον αριθμητή και τον παρονομαστή που βρήκαμε στο προηγούμενο βήμα. Η κατασκευή του γεωμετρικού τόπου των ριζών/πόλων γίνεται με την εντολή *rlocus*.

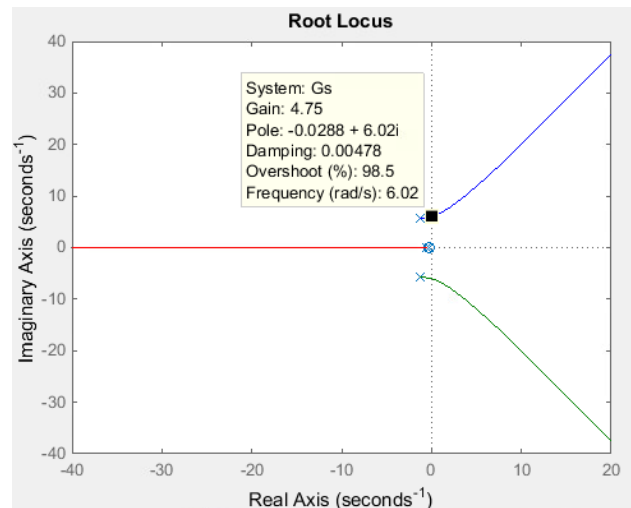
<pre>Gs=tf(num,den) figure(1);rlocus(Gs)</pre>

Με την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:



Γεωμετρικός τόπος ριζών/πόλων (Βοηθητικός βρόχος)

Το σύστημα θα είναι ευσταθές για $K_c > 0$, επομένως επιλέγουμε ένα K_c που θα εγγυάται μία γρήγορη απόκριση (έστω $K_{c2} = 6$)



Γεωμετρικός τόπος ριζών/πόλων (Κύριος βρόχος)

$$K_{cr} = 4.75$$

$$P_{cr} = 2\pi / 6.02 = 1.04$$

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.3/1.2](#).

Στο σημείο αυτό με βάση τη ρύθμιση *Ziegler-Nichols* υπολογίζουμε τις σταθερές K_c , τ_I και τ_D για τον PID ελεγκτή του κύριου βρόχου.

Type of Controller	K_c	τ_I	τ_D
P	$0.5 \cdot K_{cr}$	∞	0
PI	$0.45 \cdot K_{cr}$	$\frac{1}{1.2} \cdot P_{cr}$	0
PID	$0.6 \cdot K_{cr}$	$0.5 \cdot P_{cr}$	$0.125 \cdot P_{cr}$



Type of Controller	K_c	τ_I	τ_D
P	2.38	∞	0
PI	2.14	0.87	0
PID	2.85	0.52	0.13

Η συνάρτηση $G_{c1}(s)$ του PID ελεγκτή μας παίρνει τελικά την μορφή:

$$G_{c1}(s) = 2.85 \cdot \left(1 + \frac{1}{0.52s} + 0.13s \right)$$

Έχοντας σχεδιάσει τις συναρτήσεις μεταφοράς των ελεγκτών και για τα δύο συστήματα (ανάδρασης, συστοιχίας) συνεχίζουμε με τον υπολογισμό αυτών καθώς και με την γραφική αναπαράσταση των σημάτων εισόδου, εξόδου και του σφάλματος, συναρτήσει του χρόνου.

3^η φάση επίλυσης

Έλεγχος Με Συμβατικό Ελεγκτή $G_c=PID$

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`. Επιπλέον, δηλώνουμε τις συναρτήσεις τις $G_{p1}(s)$, $G_{p2}(s)$ και $G_{d1}(s)$, $G_{d2}(s)$ όπως μας δόθηκαν στην εκφώνηση, αλλά και τις σταθερές K_c , t_I και t_D και τον PID ελεγκτή μας ($G_c(s)$) όπως σχεδιάστηκε στην 1^η φάση επίλυσης του συγκεκριμένου προβλήματος.

```
clc
clear

syms s

Gp1=(5*s+1)/((2*s+1)*(4*s+1));
Gd1=Gp1;
Gp2=1/(0.2*s^2+0.5*s+1);
Gd2=Gp2;

Kc=3.22; tI=1.24; tD=0.3;
Gc=Kc*(1+(1/(tI*s))+tD*s);
```

Ορίζουμε τις εισόδους Y_{sp} (σήμα αναφοράς) και D_1 , D_2 (διαταραχές).

```
Ysp=(1/s);
D1=(1/s)*exp(-14*s);
D2=(1/s)*exp(-8*s);
```

Στο επόμενο βήμα προσδιορίζουμε τις σχέσεις κλειστού βρόχου $Y(s)$ και $y(t)$, το σφάλμα E και το σήμα εισόδου U που μεταβάλλεται από τον ελεγκτή μας και δηλώνουμε την διάρκεια του χρόνου:

$$E = Y_{sp}(s) - Y(s)$$

και

$$U = G_{d2}(s) \cdot D_2(s) + G_{p2}(s) \cdot G_c(s) \cdot E(s)$$

```
Y=((Gc*Gp1*Gp2)/(1+Gc*Gp1*Gp2))*Ysp+...
    (Gd1/(1+Gc*Gp1*Gp2))*D1+...
    (Gd2*Gp1/(1+Gc*Gp1*Gp2))*D2

Error=Ysp-Y;
U=Gd2*D2+Gp2*Gc*Error;

y=ilaplace(Y)
u=ilaplace(U);
error=ilaplace(Error);

Total_Time=30;
t=0:0.1:Total_Time;
```

```

y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);

```

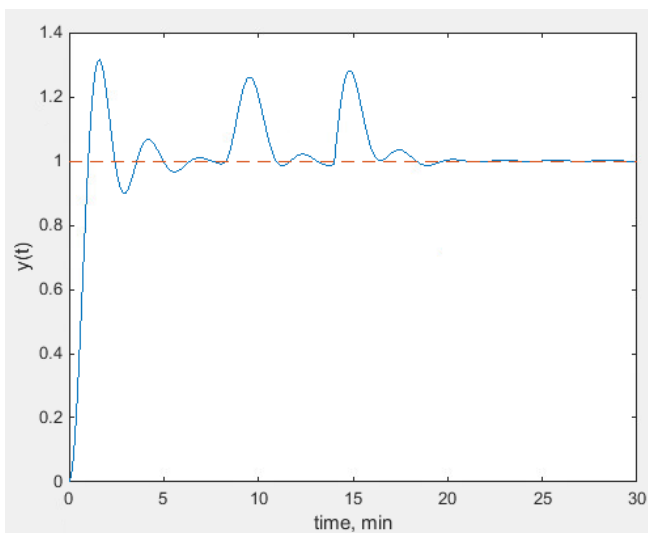
Έπειτα, προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```

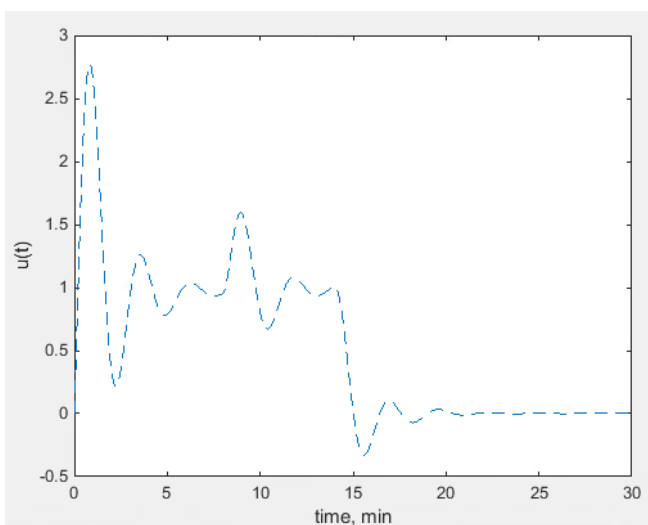
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')

```

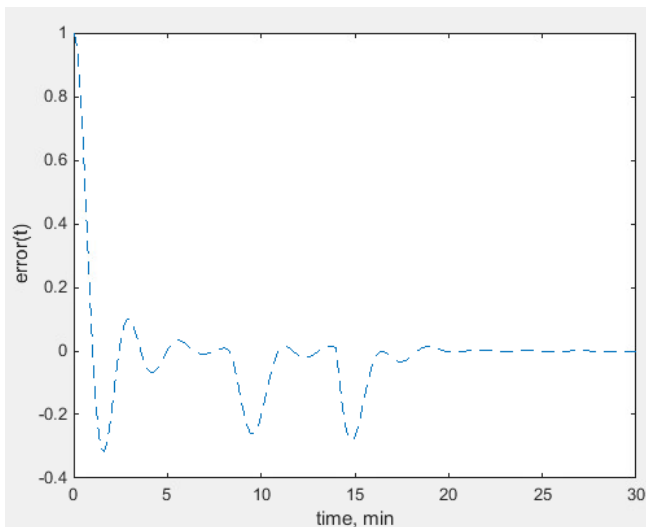
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y (Διαταραχές σε $t=8$ και $t=14$)



Σήμα εισόδου διεργασίας U (Διαταραχές σε $t=8$ και $t=14$)



Σήμα σφάλματος E (Διαταραχές σε $t=8$ και $t=14$)

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.3/1.3](#).

4^η Φάση επίλυσης

Έλεγχος συστοιχίας με $G_{c2}=P$ ελεγκτής και $G_{c1}=PID$ ελεγκτής

Ξεκινάμε όπως και στα προηγούμενα παραδείγματα με τις εντολές `clc` και `clear` για να «καθαρίσουμε» το ιστορικό από προηγούμενες εντολές. Θα εργαστούμε με συμβολικές μεταβλητές γι' αυτό και χρησιμοποιούμε την εντολή `syms`. Επιπλέον, δηλώνουμε τις συναρτήσεις τις $G_{p1}(s)$, $G_{p2}(s)$ και $G_{d1}(s)$, $G_{d2}(s)$ όπως μας δόθηκαν στην εκφώνηση, αλλά και τις σταθερές K_{c2} , K_{c1} , τ_{I1} και τ_D και τους P και PID ελεγκτές μας αντίστοιχα (G_{c2} , G_{c1}) όπως σχεδιάστηκαν στην 2^η φάση επίλυσης του συγκεκριμένου προβλήματος.

```
clc
clear

syms s

Gp1=(5*s+1)/((2*s+1)*(4*s+1));
Gd1=Gp1;
Gp2=1/(0.2*s^2+0.5*s+1);
Gd2=Gp2;
Kc2=6;
Gc2=Kc2;
Kc1=2.85; tI1=0.52; tD=0.13;
Gc1=Kc1*(1+(1/(tI1*s))+tD*s);
```

Ορίζουμε τις εισόδους Y_{sp} (σήμα αναφοράς) και D_1 , D_2 (διαταραχές).

```
Ysp=(1/s);
D1=(1/s)*exp(-14*s);
D2=(1/s)*exp(-8*s);
```

Στο επόμενο βήμα προσδιορίζουμε τις σχέσεις κλειστού βρόχου $Y(s)$ και $y(t)$, το σφάλμα E και το σήμα εισόδου U που μεταβάλλεται από τον ελεγκτή μας και δηλώνουμε την διάρκεια του χρόνου:

```
Y=((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2)))/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2))))*Ysp+...
    (Gd1/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2))))) *D1+...
```

```

(Gd2*Gp1/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2)))))*D2

Error=Ysp-Y;
U=((Gc2*Gp2)/(1+Gc2*Gp2))*(Gc1*(Ysp-Y))+(Gd2/(1+((Gc2*Gp2))))*D2;
u=ilaplace(U);
error=ilaplace(Error);
y=ilaplace(Y)

Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);

```

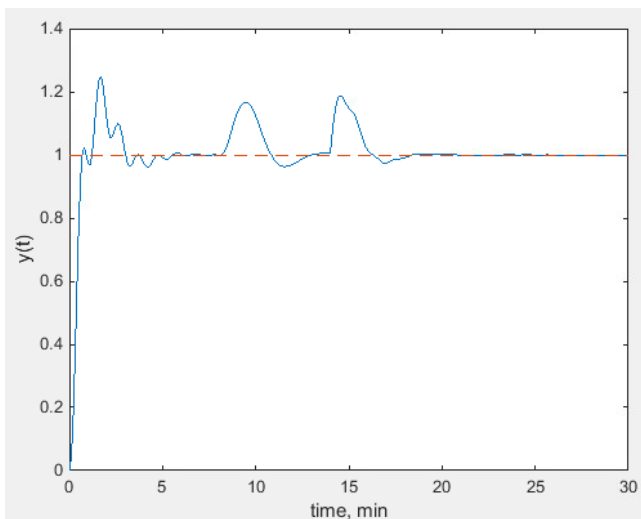
Έπειτα, προσδιορίζουμε την γραφική παράσταση της απόκρισης $y(t)$ και των σημάτων u και $error$ καθώς και την απόκριση της $ysp(t)$:

```

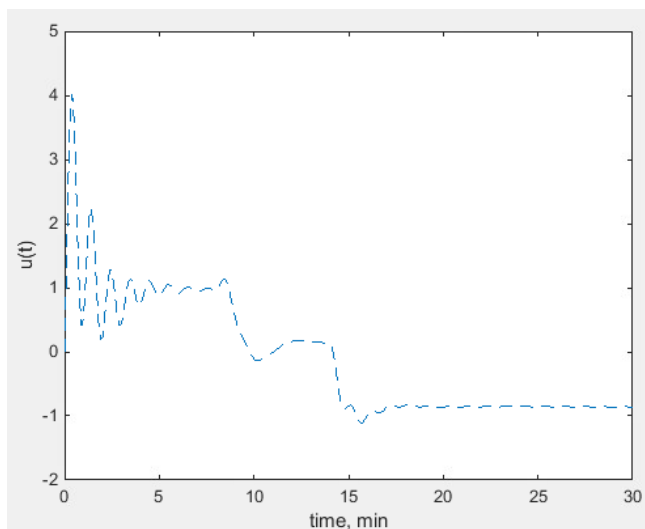
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')

```

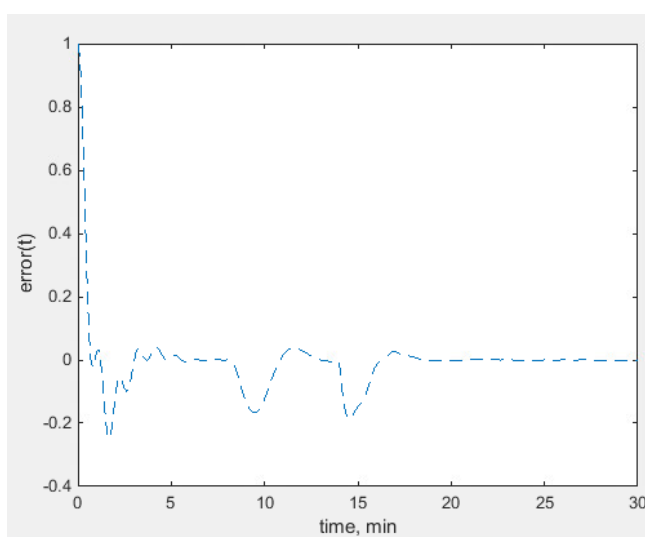
Μετά την εκτέλεση του κώδικά θα έχουμε τα παρακάτω διαγράμματα:



Σήμα εξόδου Y (Διαταραχές σε $t=8$ και $t=14$)



Σήμα εισόδου διεργασίας U (Διαταραχές σε $t=8$ και $t=14$)



Σήμα σφάλματος E (Διαταραχές σε $t=8$ και $t=14$)

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [2.3/1.4](#).

ΚΕΦΑΛΑΙΟ 3: Μοντέλα Χώρου Κατάστασης και Γραμμικοποίηση Συστημάτων

Στο παρόν κεφάλαιο με τη βοήθεια της MATLAB θα γίνει γραμμικοποίηση συστημάτων πολλών εισόδων και εξόδων καθώς και θα επιλυθούν μη-γραμμικά συστήματα δυναμικών (διαφορικών) εξισώσεων στον χώρο κατάστασης.

Όπως ορίστηκε και στη θεωρία ένα γενικευμένο μοντέλο ενός δυναμικού συστήματος περιγράφεται από τις εξής γραμμικές ή μη-γραμμικές εξισώσεις:

$$\dot{x}(t) = f_d(x, u), \quad x(t = t_0 = 0) = x_0$$

$$0 = f_a(x, u)$$

$$y(t) = g(x, u)$$

Όπου,

x = Μεταβλητές κατάστασης

y = Μεταβλητές Εξόδου

u = Μεταβλητές Εισόδου

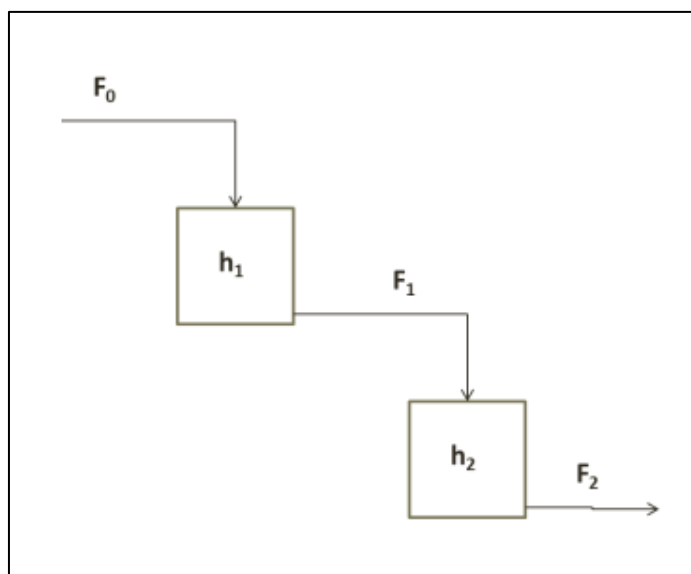
f_d = Το σύνολο των διαφορικών εξισώσεων που αφορούν τις μεταβλητές κατάστασης

f_a = Το σύνολο των αλγεβρικών εξισώσεων

g = Το σύνολο των αλγεβρικών εξισώσεων που αφορούν τις μεταβλητές εξόδου

Παράδειγμα 3.1.1: Δοχεία σε σειρά (Σημείο ισορροπίας 0)

Στο παράδειγμα αυτό θα εξεταστεί η στάθμη δύο δοχείων που βρίσκονται σε σειρά συναρτήσει του χρόνου.



Εικόνα 3.1, Δοχεία σε σειρά

Επίλυση: Αρχικά κατασκευάζουμε στο χαρτί μας τις συναρτήσεις που περιγράφουν την παραπάνω μεταβολή λαμβάνοντας υπόψιν ότι, h_1 , h_2 , οι στάθμες σε κάθε δοχείο και F_1 , F_2 οι έξοδοι που αποτελούν μη-γραμμική συνάρτηση της στάθμης h_1 , h_2 . Επομένως προκύπτουν οι εξής εξισώσεις:

$$A_1 \frac{dh_1}{dt} = F_0 - F_1 = F_0 - k_1(h_1)^{n_1}$$

$$A_2 \frac{dh_2}{dt} = F_1 - F_2 = k_1(h_1)^{n_1} - k_2(h_2)^{n_2}$$

Όπου $A_1=1$, $A_2=1$, $k_1=0.5$, $k_2=0.5$, $F_0=2$, $n_1=0.5$ και $n_2=0.5$.

Στο σημείο αυτό θα χρησιμοποιήσουμε την MATLAB για να επιλύσουμε το παραπάνω σύστημα.

Ξεκινώντας, θα δημιουργήσουμε μια συνάρτηση στην οποία θα εισάγουμε όλες τις εντολές μας για να επιλύσουμε το πρόβλημα. Επομένως με την εντολή `function` δηλώνουμε την συνάρτησή μας με δεδομένο εξόδο τη μεταβλητή S και θα καλείται με το όνομα `Tank_Height`.

```
function S = Tank_Height
```

Στη συνέχεια, κατά τα γνωστά με τις εντολές `clc` και `clear` «καθαρίζουμε» το command window από προηγούμενες εντολές και αμέσως μετά ορίζουμε τις σταθερές του προβλήματος μας A_1 , A_2 , k_1 , k_2 , F_0 , n_1 και n_2 καθώς και τις αρχικές τιμές των μεταβλητών μας (στον χρόνο $t=0$).

```
clc
clear
%Constant Parameters
A1=1;
A2=1;
k1=0.5;
k2=0.5;
F0=2;

n1=0.5;
n2=0.5;

h0(1)=0;
h0(2)=0;
```

Έπειτα, δημιουργούμε ακόμα μία συνάρτηση «μέσα» στην αρχική μας συνάρτηση για την επίλυση των διαφορικών εξισώσεων με μεταβλητή εξόδο το dx και δεδομένα εισόδου τα t, x . Επιπλέον, ορίζουμε τις μεταβλητές κατάστασης, οι οποίες στο συγκεκριμένο πρόβλημα είναι τα ύψη των δοχείων h_1 , h_2 .

```
function dx=f(t,x)

h_1= x(1,1);
h_2= x(2,1);
```

Ακολουθεί η δήλωση των αλγεβρικών εξισώσεων, δηλαδή οι σχέσεις F_1, F_2 με τις μεταβλητές h_1, h_2 .

```
%Algebraic Equations
```

```
F1=k1*(h_1)^n1;
```

```
F2=k2*(h_2)^n2;
```

Στο σημείο αυτό ορίζουμε τις διαφορικές εξισώσεις του προβλήματος όπως ακριβώς υπολογίσαμε στο χαρτί. Επιπλέον δημιουργούμε έναν μηδενικό πίνακα «dx» μεγέθους '2x1' (καθότι έχουμε δύο διαφορικές εξισώσεις), όπου «dx» ο όρος «dx/dt» και έπειτα «κλείνουμε» την συνάρτηση με την εντολή *end*.

```
dx = zeros(2,1);
```

```
%doxeiol
```

```
dx(1)= (F0-F1)*(1/A1);
```

```
dx(2)= (F1-F2)*(1/A2);
```

```
end
```

Η MATLAB διαθέτει αρκετές εντολές βιβλιοθήκης για την επίλυση συστημάτων διαφορικών εξισώσεων. Εμείς θα χρησιμοποιήσουμε την *ode45*. Τα δεδομένα εισόδου τα εντολής έχουν ως εξής:

- *f* = Η συνάρτηση που ορίσαμε τις διαφορικές εξισώσεις μας. Χρειάζεται να βάλουμε '@' καθότι δίνουμε m-file σαν δεδομένο εισόδου.
- *tspan* = Ο χρόνος επίλυσης τον οποίο θα ορίσουμε ως [0 : 1 : 150].
- *h0* = Οι αρχικές μας συνθήκες.

```
%solving parameters
```

```
tspan=[0:1:150];
```

```
[tsol,xsol]= ode45(@f,tspan,h0);
```

Τέλος, ορίζουμε τις γραφικές παραστάσεις των μεταβλητών μας, h_1 , h_2 (στάθμες δοχείων) με την εντολή *plot*. Επίσης, δημιουργούμε αρχείο excel με την εντολή *xlswrite*, όπου θα αποθηκευτούν οι τιμές των μεταβλητών μας.

```
figure(1);
```

```
plot(tsol,xsol(:,1),'--','MarkerSize',10);
```

```
hold on
```

```
plot(tsol,xsol(:,2));
```

```
hold off
```

```
xlabel('Time, min');
```

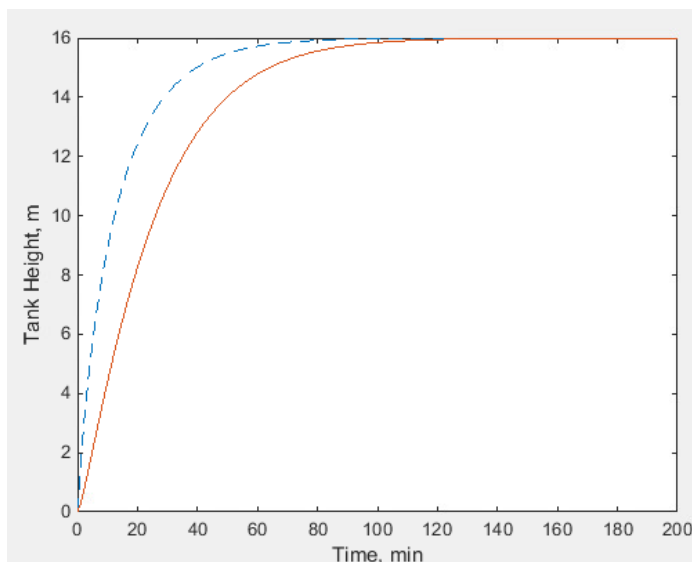
```
ylabel('Tank Height, m')
```

```
drawnow
```

```
xlswrite('height.xlsx',(tsol xsol)
```

```
end
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:



Γραφική αναπαράσταση των μεταβλητών h_1, h_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [3.1/1](#).

Παράδειγμα 3.1.2: Δοχεία σε σειρά (Σημείο ισορροπίας $\neq 0$)

Όπως και στο προηγούμενο κεφάλαιο έτσι κι εδώ, θα ασχοληθούμε με την στάθμη δύο δοχείων που βρίσκονται σε σειρά. Η διαφορά μας εδώ είναι ότι το σημείο ισορροπίας δεν είναι πλέον 0,0 αλλά $(h_{1s}, h_{2s}) = (16, 16)$.

Επίλυση: Θα κατασκευάσουμε στο χαρτί μας τις συναρτήσεις που περιγράφουν την παραπάνω μεταβολή λαμβάνοντας υπόψιν ότι, h_1, h_2 , οι στάθμες σε κάθε δοχείο και F_1, F_2 οι έξοδοι που αποτελούν μη-γραμμική συνάρτηση της στάθμης h_1, h_2 . Επομένως προκύπτουν οι εξής εξισώσεις:

$$A_1 \frac{d(h_1 - h_{1s})}{dt} = \frac{1}{A_1} [-k_1 0.5(h_{1s})^{-0.5}(h_1 - h_{1s}) + (F_0 - F_{0s})]$$

$$A_2 \frac{d(h_2 - h_{1s})}{dt} = \frac{1}{A_1} [k_1 0.5(h_{1s})^{-0.5}(h_1 - h_{1s}) - k_2 0.5(h_{2s})^{-0.5}(h_2 - h_{2s})]$$

Όπου $A_1=1, A_2=1, k_1=0.5, k_2=0.5, F_0=2, F_{0s}=2, n_1=0.5$ και $n_2=0.5$.

Αρχικά, θα δημιουργήσουμε όπως και στο προηγούμενο παράδειγμα μια συνάρτηση στην οποία θα εισάγουμε όλες τις εντολές μας για να επιλύσουμε το πρόβλημα. Επομένως, με την εντολή `function` δηλώνουμε την συνάρτησή μας με δεδομένο εξόδο τη μεταβλητή `S1` και θα καλείται με το όνομα `Tank_Height_2`.

```
Function S1 = Tank_Height_2
```

Στη συνέχεια, κατά τα γνωστά με τις εντολές `clc` και `clear` «καθαρίζουμε» το command window από προηγούμενες εντολές και αμέσως μετά ορίζουμε τις σταθερές του προβλήματος μας $A_1, A_2, k_1, k_2, F_0, n_1$ και n_2 .

Προσοχή: Δεν δηλώνουμε σε αυτό το βήμα τις αρχικές μας τιμές.

```
clc
```

```
clear
%Constant Parameters
A1=1;
A2=1;
k1=0.5;
k2=0.5;
F0=2;

n1=0.5;
n2=0.5;
```

Στη συνέχεια ορίζουμε το σημείο γραμμικοποίησης και τις αρχικές μας τιμές στο σημείο $t=0$.

```
F0s=2;

h1s= (F0s/k1)^(1/n1);
h1s= ((k1/k2)*h1s^n1)^(1/n2);

h0(1)=0-h1s;
h0(2)=0-h2s;
```

Έπειτα, δημιουργούμε και πάλι μία συνάρτηση «μέσα» στην αρχική μας συνάρτηση για την επίλυση των διαφορικών εξισώσεων με μεταβλητή εξόδου το dx και δεδομένα εισόδου τα t,x . Επιπλέον, ορίζουμε τις μεταβλητές κατάστασης.

```
function dx=f1(t,x)

h_1_new= x(1,1);
h_2_new= x(2,1);
```

Στο σημείο αυτό ορίζουμε τις διαφορικές εξισώσεις του προβλήματος όπως ακριβώς υπολογίσαμε στο χαρτί προηγουμένως. Επιπλέον δημιουργούμε έναν μηδενικό πίνακα « dx » μεγέθους '2x1' (καθότι έχουμε δύο διαφορικές εξισώσεις), όπου « dx » ο όρος « dx/dt » και έπειτα «κλείνουμε» την συνάρτηση με την εντολή *end*.

```
dx = zeros(2,1);
%doxeio1
dx(1)= (1/A1)*(-k1*n1*(h1s)^(-n1))*(h_1_new)+(F0-Fs);
%doxeio2
dx(2)=(1/A2)*(k1*n1*((h1s)^(-n1))*(h_1_new)-k2*n2*...
(h2s^(-n2))*(h_2_new));
```

Χρησιμοποιούμε την εντολή *ode45* για να επιλύσουμε το σύστημά μας και ορίζουμε το χρόνο επίλυσης $[0 : 1 : 100]$.

```
%solving parameters
tspan=[0:1:100];
[tsol,xsol]= ode45(@f,tspan,h0);
```

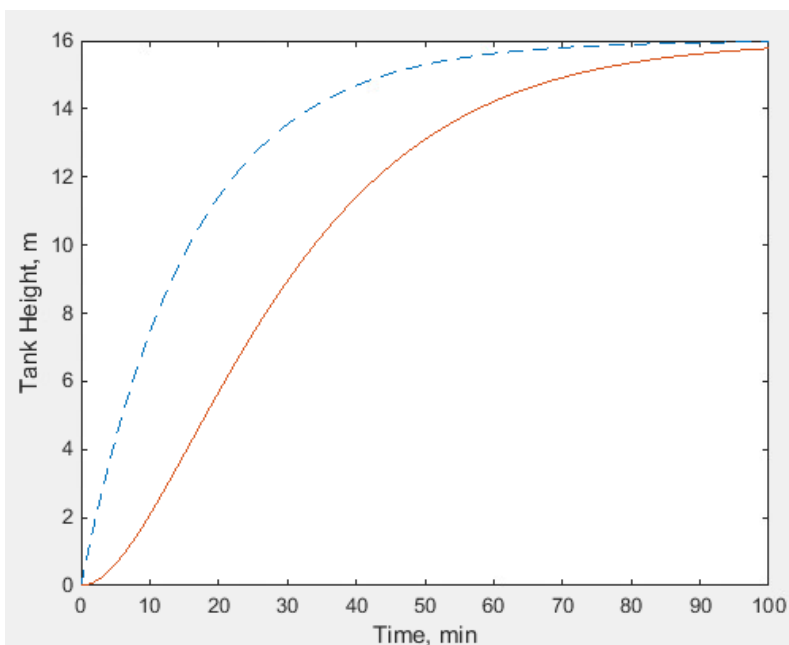
Τέλος, με την εντολή *plot* ορίζουμε τις γραφικές παραστάσεις των μεταβλητών μας και δημιουργούμε αρχείο excel όπου θα αποθηκευτούν οι τιμές τους.

Προσοχή: Τα αποτελέσματά μας θα ξεκινούν από το $(h_{1s}, h_{2s})=(16,16)$, επομένως για να μπορέσουμε να τα συγκρίνουμε τα μετατοπίζουμε στο σημείο $(0,0)$.

```
figure(1);
plot(tsol,xsol(:,1)+h1s,'--','MarkerSize',10);
hold on
plot(tsol,xsol(:,2)+h2s);
hold off
xlabel('Time, min');
ylabel('Tank Height, m')
drawnow

xlswrite('height.xlsx',[tsol,xsol])
end
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:

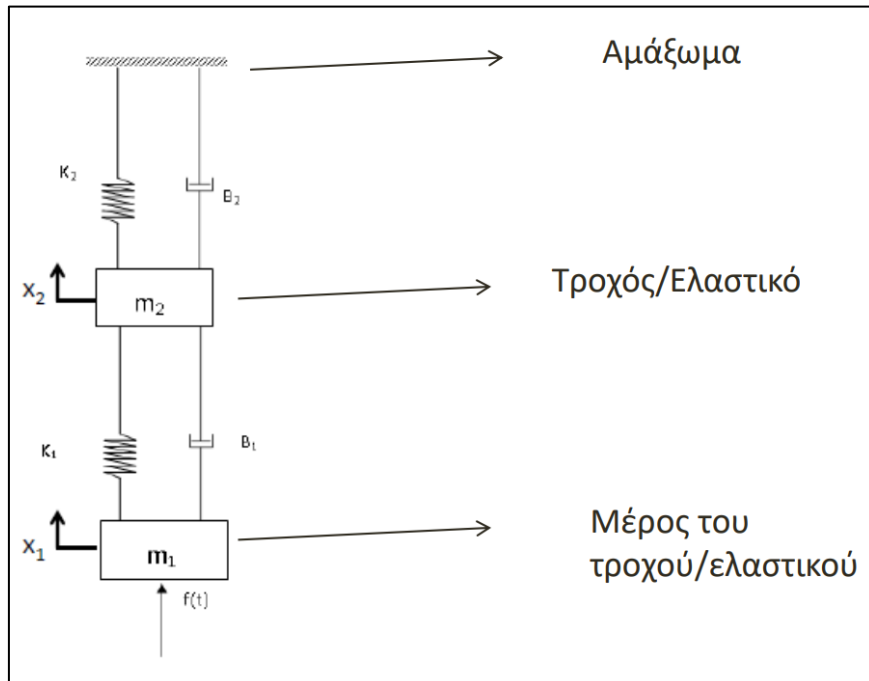


Γραφική αναπαράσταση των μεταβλητών h_1, h_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [3.1/2](#).

Παράδειγμα 3.1.3: Ανάρτηση βαρέως οχήματος

Στο συγκεκριμένο παράδειγμα, θα εργαστούμε με παρόμοιο τρόπο όπως και στα προηγούμενα, αλλά ο χώρος κατάστασης θα είναι διαφορετικός. Επομένως, θα χρησιμοποιηθούν άλλες διαφορικές εξισώσεις με διαφορετικές μεταβλητές.



Εικόνα 3.2, Ανάρτηση βαρέως οχήματος

$$m_1: -B_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2) + f = m_1\ddot{x}_1$$

$$m_2: B_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) - k_2x_2 - B_2\dot{x}_2 = m_2\ddot{x}_2 \text{ Ορίζουμε τα εξής:}$$

$$x'_1 = x_1$$

$$x'_2 = \frac{dx_1}{dt} = \dot{x}'_1$$

$$x'_3 = x_2$$

$$x'_4 = \frac{dx_2}{dt} = \dot{x}'_3$$

Με αντικατάσταση στις παραπάνω σχέσεις θα έχουμε:

$$m_1\dot{x}'_1 + B_1(x'_2 - x'_4) + k_1(x'_1 - x'_3) = f$$

$$m_2\dot{x}'_4 - B_1(x'_2 - x'_4) - k_1(x'_1 - x'_3) + k_2x'_3 + B_2x'_4 = 0$$

Επίλυση: Κατασκευάζουμε τις εξισώσεις που περιγράφουν το παραπάνω σύστημα, το οποίο θα είναι ένα γραμμικό σύστημα τεσσάρων διαφορικών εξισώσεων.

$$\dot{x}'_1 = x'_2$$

$$\dot{x}'_2 = -\frac{B_1}{m_1}(x'_2 - x'_4) - \frac{k_1}{m_1}(x'_1 - x'_3) + \frac{1}{m_1}f$$

$$\dot{x}'_3 = x'_4$$

$$\dot{x}'_4 = -\frac{B_1}{m_2}(x'_2 - x'_4) - \frac{k_1}{m_2}(x'_1 - x'_3) - \frac{k_2}{m_2}x'_3 - \frac{B_2}{m_2}x'_4$$

Όπου, ορίζουμε $y'_1 = x_1 = x'_1$ & $y'_2 = x_2 = x'_3$.

Έχοντας υπολογίσει τις εξισώσεις μας, δημιουργούμε και πάλι μια συνάρτηση με δεδομένο εξόδου S , η οποία θα καλείται με το όνομα *Car_Dispension*.

```
function S = Car_Dispension
```

Στη συνέχεια, «καθαρίζουμε» από προηγούμενες εντολές με τις εντολές *clc* και *clear* και αμέσως μετά ορίζουμε τις σταθερές του προβλήματος μας B_1 , B_2 , m_1 , m_2 , k_1 , k_2 , και F . Επιπλέον, ορίζουμε τις αρχικές μας συνθήκες στον χρόνο $t=0$ (με τις βοηθητικές μεταβλητές).

```
clc
clear
%Constant Parameters
B1=1;
B2=1;
m1=15;
m2=5;
k1=1;
k2=1;
F=2;

x0(1)=0;
x0(2)=0;
x0(3)=0;
x0(4)=0;
```

Έπειτα, δημιουργούμε και πάλι μία συνάρτηση «μέσα» στην αρχική μας συνάρτηση για την επίλυση των διαφορικών εξισώσεων με μεταβλητή εξόδου το dx και δεδομένα εισόδου τα t,x . Επιπλέον, ορίζουμε τις μεταβλητές κατάστασης και τις αλγεβρικές εξισώσεις (σχέσεις των y_1,y_2 με τις x_1,x_3).

```
function dx=f(t,x)
x_1 = x(1,1);
x_2 = x(2,1);
x_3 = x(3,1);
x_4 = x(4,1);
y1=x_1;
y2=x_3;
```

Στο σημείο αυτό ορίζουμε τις διαφορικές εξισώσεις του προβλήματος όπως ακριβώς υπολογίσαμε στο χαρτί προηγουμένως. Επιπλέον δημιουργούμε έναν μηδενικό πίνακα «dx» μεγέθους '4x1' (καθότι έχουμε τέσσερις διαφορικές εξισώσεις), όπου «dx» ο όρος «dx/dt» και έπειτα «κλείνουμε» την συνάρτηση με την εντολή *end*.

```
dx = zeros(4,1);
dx(1)= x_2;
dx(2)=(-B1/m1)*(x_2-x_4)-(k1/m1)*(x_1-x_3)+(1/m1)*F;
dx(3)=x_4;
dx(4)=(B1/m2)*(x_2-x_4)+(k1/m2)*(x_1-x_3)-(k2/m2)*x_3-(B2/m2)*x_4;
end
```

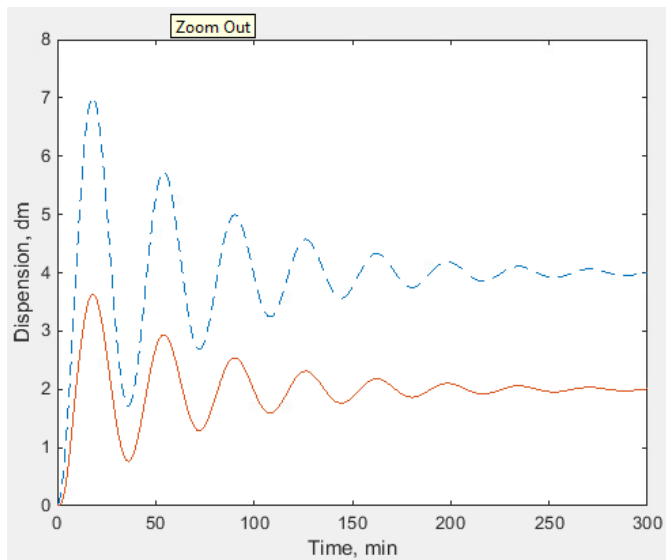
Χρησιμοποιούμε την εντολή *ode45* για να επιλύσουμε το σύστημά μας και ορίζουμε το χρόνο επίλυσης [0 : 1 : 300].

```
%solving parameters
tspan=[0:1:300];
[tsol,xsol]= ode45(@f,tspan,x0);
```

Τέλος, με την εντολή *plot* ορίζουμε τις γραφικές παραστάσεις των μεταβλητών μας και στη συνέχεια, δημιουργούμε αρχείο excel όπου θα αποθηκευτούν οι τιμές τους.

```
figure(1);
plot(tsol,xsol(:,1),'--','MarkerSize',10);
hold on
plot(tsol,xsol(:,3));
hold off
xlabel('Time, min');
ylabel('Dispension, dm');
drawnow
xlswrite('Car_dispension.xlsx',[tsol xsol])
end
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:



Γραφική αναπαράσταση των μεταβλητών
y1, y2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [3.1/3](#).

ΚΕΦΑΛΑΙΟ 4: Μοντέλα Χώρου Κατάστασης και Επίλυση Συστημάτων

Στο προηγούμενο κεφάλαιο αναλύσαμε την γραμμικοποίηση δυναμικών εξισώσεων στον χώρο κατάστασης. Τώρα καλούμαστε να επιλύσουμε αυτά τα γραμμικοποιημένα συστήματα με τρεις τρόπους:

- Αναλυτική επίλυση στο πεδίο του χρόνου, για ομογενείς και μη διαφορικές εξισώσεις.
- Επίλυση στο πεδίο Laplace, για ομογενείς και μη διαφορικές εξισώσεις.
- Επίλυση με χρήση ιδιοτιμών/ιδιοδιανυσμάτων.

4.1 Επίλυση στο πεδίο του χρόνου

Παράδειγμα 4.1.1: Εύρεση εκθετικού πίνακα e^{At} (Για ομογενείς διαφορικές εξισώσεις)

Να υπολογίσετε τον εκθετικό πίνακα e^{At} για $A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$ με την μέθοδο Caley-Hamilton όπως αυτή παρουσιάστηκε στην θεωρία.

Αρχικά θα πρέπει να υπολογίσουμε στο χαρτί μας τις ιδιοτιμές λ_i του πίνακα A.

$$|A - \lambda I| = e^{At} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} = \lambda^2 + 3\lambda + 2$$
$$\lambda_1 = -1, \lambda_2 = -2$$

Στη συνέχεια ακολουθώντας τα βήματα που παρουσιάστηκαν στη θεωρία προκύπτει:

$$f(A) = e^{At} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & e^{-t} + 2e^{-2t} \end{bmatrix}$$

Οι παραπάνω όμως υπολογισμοί μπορούν να πραγματοποιηθούν πολύ εύκολα και γρήγορα στο περιβάλλον της MATLAB.

Επίλυση: Ορίζουμε αρχικά τον πίνακα A και προσδιορίζουμε τις ιδιοτιμές λ_i με την εντολή `eig`. Επιπλέον, δηλώνουμε ότι θα εργαστούμε με συμβολικές μεταβλητές με την εντολή `syms` και προσδιορίζουμε τον εκθετικό πίνακα με την εντολή `exp`.

```
A=[0 1;-2 -3];  
L=eig(A)  
  
syms t  
exp_At=expm(A*t)
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:

```
Command Window

L =

    -1
    -2

exp_At =

[ 2*exp(-t) - exp(-2*t), exp(-t) - exp(-2*t)]
[ 2*exp(-2*t) - 2*exp(-t), 2*exp(-2*t) - exp(-t)]
```

Οι ιδιοτιμές του πίνακα A και ο εκθετικός πίνακας e^{At}

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.1/1](#).

Παράδειγμα 4.1.2: Επίλυση στο πεδίο του χρόνου (Για ομογενείς διαφορικές εξισώσεις)

Στο παράδειγμα αυτό παρουσιάζεται η επίλυση του προηγούμενου προβλήματος, καθώς και η γραφική αναπαράσταση των αποτελεσμάτων.

Επίλυση: Αρχικά, χρησιμοποιούμε τις εντολές *clc* και *clear* για να «καθαρίσουμε» το command window από προηγούμενες εντολές. Ταυτόχρονα, δηλώνουμε ότι θα εργαστούμε με συμβολικές μεταβλητές.

```
clc
clear

syms t
```

Στη συνέχεια, ορίζουμε το πίνακα A, το διάνυσμα αρχικών τιμών για $t=0$ καθώς και τον εκθετικό πίνακα με την εντολή *exp* όπως ακριβώς κάναμε και στον προηγούμενο παράδειγμα.

```
A=[0 1; -2 -3];
x0=[1;1];

exp_At=expm(A*t);
```

Τέλος, προσδιορίζουμε τη διάρκεια του χρόνου και χρησιμοποιούμε τις γνωστές εντολές *subs* και *double*, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή γραφικών παραστάσεων.

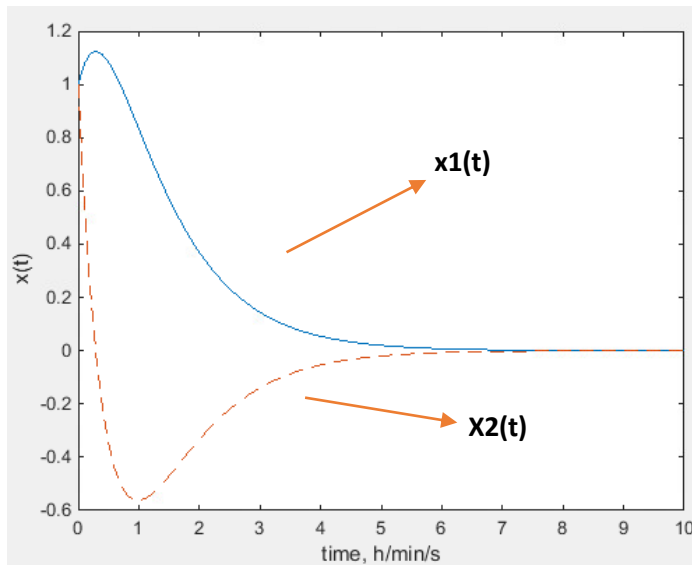
```
Total_Time=10;
t=0:0.1:Total_Time;

x_t=exp_At*x0;
x_t=subs(x_t,t);
x_t=double(x_t);

figure(1);
```

```
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');
ylabel('x(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτει το διάγραμμα:



Διάγραμμα χρονικών αποκρίσεων x_1 , x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.1/2](#).

Παράδειγμα 4.1.3: Επίλυση στο πεδίο του χρόνου (Για μη-ομογενείς διαφορικές εξισώσεις)

Σας δίνεται το σύστημα των γραμμικών μη ομογενών διαφορικών εξισώσεων:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Να υπολογίσετε τις χρονικές αποκρίσεις των $x_1(t)$ και $x_2(t)$ εάν η μεταβλητή εισόδου $u(t)$ μεταβάλλεται βηματικά, $u(t)=1H(t)$.

Επίλυση: Αρχικά, «καθαρίζουμε» από προηγούμενες εντολές με τις εντολές `clc` και `clear` και αμέσως μετά δηλώνουμε πως θα εργαστούμε με συμβολικές μεταβλητές.

```
clc
clear

syms t tau
```

Συνεχίζοντας, ορίζουμε τους πίνακες A,B και το διάνυσμα αρχικής κατάστασης

```
A=[0 1; -2 -3];
B=[0;1];
x0=[0;0];
```

Έπειτα προσδιορίζουμε τον εκθετικό πίνακα e^{At} και την είσοδο $u(t)$.

```
exp_At=expm(A*t);
u_t=1;
```

Στο σημείο αυτό θα πρέπει να επιλύσουμε την $x(t)$ για την οποία γνωρίζουμε από θεωρία ότι:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}[Bu(\tau)]d\tau$$

Για να ορίσουμε το ολοκλήρωμα της συνάρτησης $fun = e^{A(t-\tau)}[Bu(\tau)]$, χρησιμοποιούμε την εντολή *int*(συνάρτηση, μεταβλητή, κάτω όριο, άνω όριο).

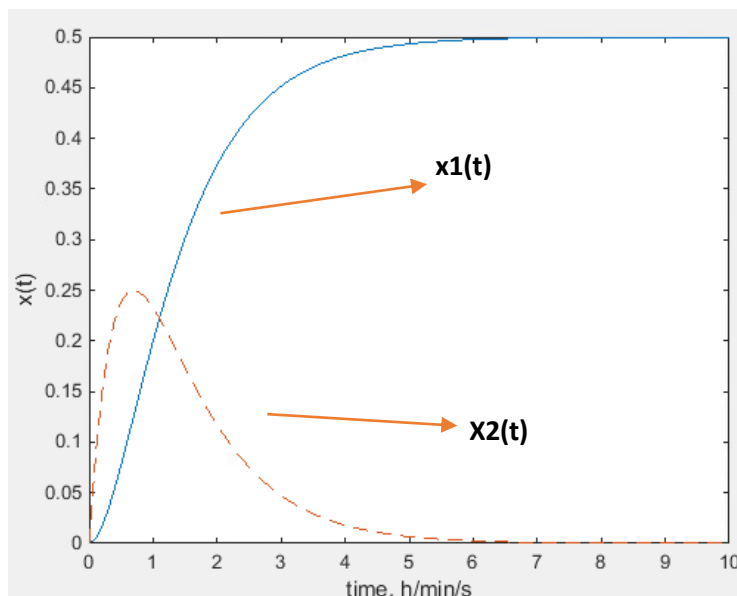
```
fun=(expm(A*(t-tau)))*B*u_t;
x_t=exp_At*x0+int(fun,tau,0,t)
```

Τέλος, λύνουμε το πρόβλημά μας στο χρόνο t και χρησιμοποιούμε τις γνωστές εντολές *subs* και *double*, ώστε να μπορέσουμε να εκφράσουμε τα αποτελέσματά μας σε μορφή γραφικών παραστάσεων.

```
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);

figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');ylabel('x(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτει το παρακάτω διάγραμμα:



Διάγραμμα χρονικών αποκρίσεων x_1 , x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.1/3](#).

4.2 Επίλυση στο πεδίο Laplace

Παράδειγμα 4.2.1: Επίλυση στο πεδίο Laplace (Για ομογενείς διαφορικές εξισώσεις)

Να βρεθεί ο εκθετικός πίνακας e^{At} για το παρακάτω γραμμικό ομογενές σύστημα διαφορικών εξισώσεων μέσω αντιστρόφων μετασχηματισμών Laplace και να λυθεί το σύστημα για $x_1(t=0)=x_2(t=0)=1$.

$$\dot{x}(t) = Ax(t), \quad x(t=0) = x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad x(s) = (sI - A)^{-1}x(0)$$

Επίλυση: Όπως κάθε φορά, «καθαρίζουμε» από προηγούμενες εντολές με τις εντολές `clc` και `clear` και αμέσως μετά δηλώνουμε πως θα εργαστούμε με συμβολικές μεταβλητές.

```
clc
clear

syms s
```

Σε αυτό το σημείο, δηλώνουμε τον πίνακα A, καθώς και την διάστασή του, το διάνυσμα αρχικών τιμών και τον μοναδιαίο πίνακα I.

Επιπλέον, με την βοήθεια του αντίστροφου μετασχηματισμού Laplace υπολογίζουμε τον εκθετικό πίνακα e^{At}

$$e^{At} = L^{-1}\{(sI - A)^{-1}\}$$

Για να υπολογιστεί ο αντίστροφος ενός πίνακα χρησιμοποιούμε την εντολή `inv` και στη συνέχεια την εντολή `ilaplace` όπως παρουσιάστηκε και σε προηγούμενο κεφάλαιο.

```
A=[0 1; -2 -3];
size_A=rank(A);
x0=[1;1];
I=eye(size_A);

exp_At=inv(s*I-A);
exp_At=ilaplace(exp_At)
```

Τέλος, λύνουμε το πρόβλημά μας στο χρόνο t και χρησιμοποιούμε τις γνωστές εντολές `subs` και `double`, για την γραφική αναπαράσταση των αποτελεσμάτων μας.

```
x_s=inv(s*I-A)*x0;
x_t=ilaplace(x_s)

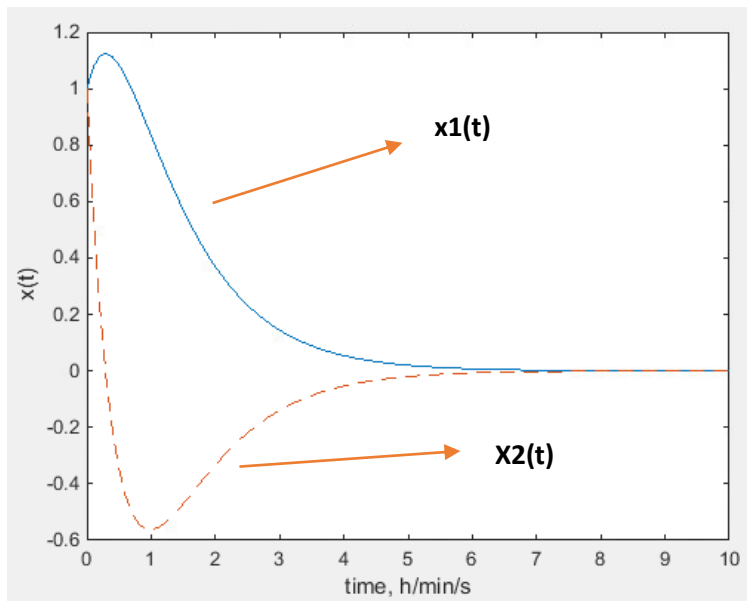
Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);
```



```
figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');
ylabel('x(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτει το παρακάτω διάγραμμα:



Διάγραμμα χρονικών αποκρίσεων x_1 , x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.2/1](#).

Παράδειγμα 4.2.2: Επίλυση στο πεδίο Laplace (Για μη-ομογενείς διαφορικές εξισώσεις)

Σε αυτό το παράδειγμα, θα επιλύσουμε ξανά το τρίτο παράδειγμα της προηγούμενης παραγράφου (Παράγραφος 4.1, Παράδειγμα 3), αλλά αυτή τη φορά στο πεδίο Laplace.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

όπου, $\dot{x}(t) = Ax(t) + Bu(t)$

Επίλυση: Αρχικά, «καθαρίζουμε» από προηγούμενες εντολές με τις εντολές *clc* και *clear* και αμέσως μετά δηλώνουμε πως θα εργαστούμε με συμβολικές μεταβλητές.

```
clc
clear

syms s
```

Συνεχίζοντας, ορίζουμε τους πίνακες A,B και το διάνυσμα αρχικής κατάστασης

```
A=[0 1; -2 -3];
B=[0;1];
x0=[0;0];
```

Έπειτα, θα πρέπει να δηλώσουμε το μέγεθος του πίνακα A, τον μοναδιαίο πίνακα I και να προσδιορίσουμε την $x(s)$ και τον αντίστροφο μετασχηματισμό Laplace όπως παρουσιάστηκε στην θεωρία.

$$x(s) = (sI - A)^{-1}x_0 + (sI - A)^{-1}Bu(s)$$

$$x(t) = L^{-1}\{x(s)\}$$

```
size_A=rank(A);
I=eye(size_A);
u_s=1/s;

x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
x_t=ilaplace(x_s)
```

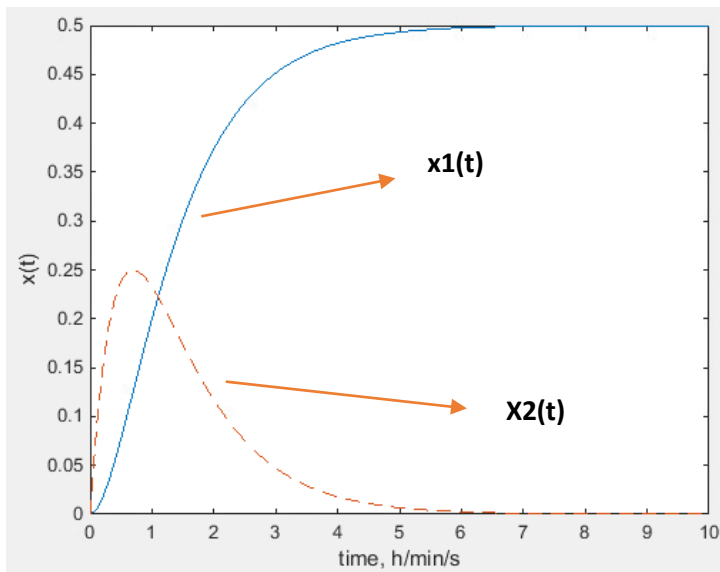
Τέλος, προσδιορίζουμε τον χρόνο επίλυσης και καταγράφουμε τα αποτελέσματα μας σε γραφικές παραστάσεις.

```
Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);

figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');
ylabel('x(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτει το παρακάτω διάγραμμα:



Διάγραμμα χρονικών αποκρίσεων x_1 , x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.2/2](#).

4.3 Επίλυση με τη χρήση ιδιοτιμών/ιδιοδιανυσμάτων

Παράδειγμα 4.3.1: Επίλυση με τη χρήση ιδιοτιμών/ιδιοδιανυσμάτων

Να επιλύσετε το παρακάτω σύστημα με τη μέθοδο ιδιοτιμών/ιδιοδιανυσμάτων γνωρίζοντας ότι, $x(t) = Me^{\lambda t}M^{-1}x(0)$.

$$\dot{x}(t) = \begin{bmatrix} -2 & 1 \\ 2 & -3 \end{bmatrix} x(t), \quad x(t=0) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Επίλυση: Αρχικά, δηλώνουμε πως θα δουλέψουμε με συμβολικές μεταβλητές κι έπειτα ορίζουμε τον πίνακα A και τις αρχικές μας τιμές στον χρόνο $t=0$.

```
clc
clear

syms t

A=[-2 1;2 -3];
x0=[1;2];
```

Στη συνέχεια με την εντολή eig θα υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα παρόμοια με προηγούμενο παράδειγμα, καθώς και με μια δομή επανάληψης προσδιορίζουμε τον πίνακα $e^{\lambda t}$. Επιπλέον προσδιορίζουμε την επίλυση της $x(t)$.

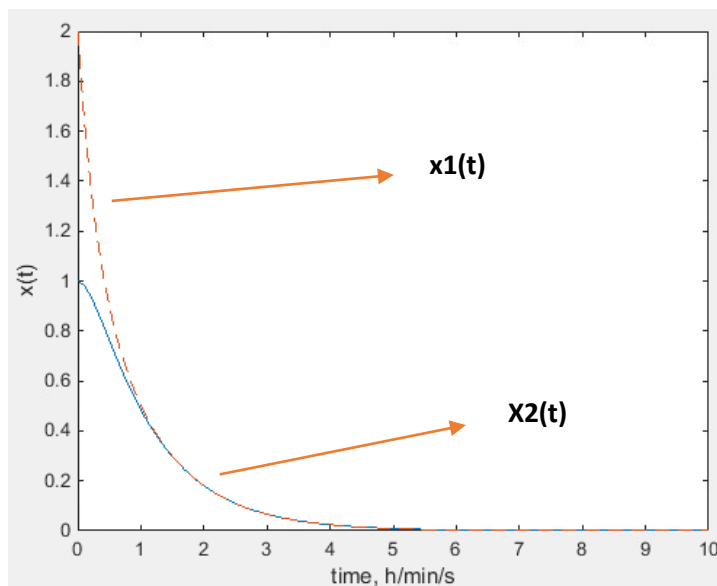
```
[M,L]=eig(A);
rank_L=rank(L);
for i=1:rank_L
exp_lambda_t(i,i)=[exp(t*L(i,i))];
end
x_t=M*exp_lambda_t*inv(M)*x0
```

Χρησιμοποιώντας τις γνωστές εντολές, προσδιορίζουμε τον χρόνο επίλυσης του προβλήματός μας και αναπαριστούμε γραφικά τα αποτελέσματά μας.

```
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτει το παρακάτω διάγραμμα:



Διάγραμμα χρονικών αποκρίσεων x_1 , x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.3/1](#).

4.4 Προσδιορισμός συνάρτησης μεταφοράς και αποκρίσεων y

Παράδειγμα 4.4.1: Προσδιορισμός $y(t)$ εάν $u(t)=1$ (χωρίς διαταραχές)

Δίνεται το ακόλουθο σύστημα γραμμικών μη ομογενών διαφορικών εξισώσεων:

$$\dot{x}_1(t) = -x_1(t) - 2x_2(t) + u(t) \text{ με } x_1(t=0) = 0 \text{ και } y_1(t) = x_1(t)$$

$$\dot{x}_2(t) = x_1(t) - 3x_2(t) \text{ με } x_2(t=0) = 0 \text{ και } y_2(t) = x_2(t)$$

Ο πίνακας $A = \begin{bmatrix} -1 & -2 \\ 1 & -3 \end{bmatrix}$.

Γνωρίζουμε επίσης από θεωρία ότι:

$$G_p(s) = \frac{y(s)}{u(s)} = [C(sI - A)^{-1}B + D]$$

όπου προκύπτουν: $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ και $D = 0$

Επίλυση: Αρχικά, δηλώνουμε πως θα εργαστούμε με συμβολικές μεταβλητές και ορίζουμε τους πίνακες A,B,C,D και τις αρχικές τιμές στον χρόνο t=0.

```
clc
clear

syms s

A=[-1 -2; 1 -3];
B=[1;0];
x0=[0;0];
C=[1 0;0 1];
D=0;
```

Στη συνέχεια, ορίζουμε το μέγεθος του πίνακα A, τον μοναδιαίο πίνακα I και προσδιορίζουμε την συνάρτηση $y(s)$ και την συνάρτηση μεταφοράς $G_p(s)$.

```
size_A=rank(A);
I=eye(size_A);
u_s=1/s;

x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s;
Gp=[C*inv(s*I-A)*B+D]
```

Όπως παρουσιάστηκε σε προηγούμενο κεφάλαιο, επιλύουμε το πρόβλημα μας με τη χρήση αντίστροφου μετασχηματισμού Laplace, ενώ παράλληλα δηλώνουμε τον χρόνο επίλυσης και κατασκευάζουμε τις γραφικές παραστάσεις των αποτελεσμάτων μας.

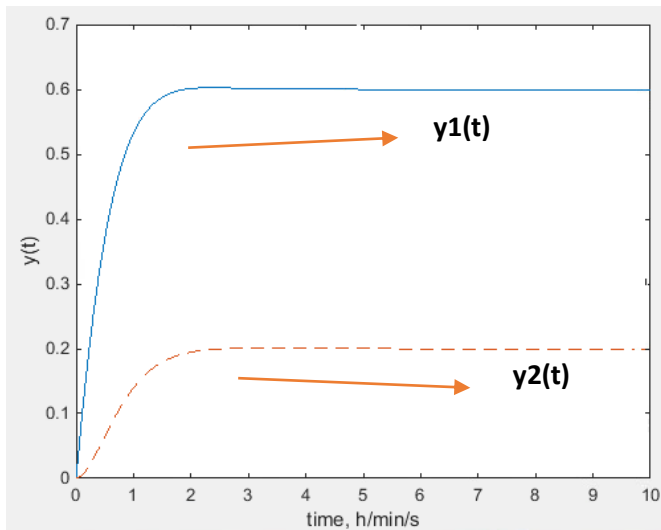
```
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)

Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:),t,y_t(2,),'--');xlabel('time, h/min/s');ylabel('y(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα :



Διάγραμμα χρονικών αποκρίσεων y_1 , y_2

```
Gp =
(s + 3) / (s^2 + 4*s + 5)
1 / (s^2 + 4*s + 5)

y_t =
3/5 - (3*exp(-2*t)*(cos(t) + sin(t)/3))/5
1/5 - (exp(-2*t)*(cos(t) + 2*sin(t)))/5
```

Οι συναρτήσεις $y(s)$, y_1 και y_2 θα έχουν τη μορφή:

```
y_s =
(s + 3) / (s*(s^2 + 4*s + 5))
1 / (s*(s^2 + 4*s + 5))
```

ή

$$y(s) = \left[\frac{s+3}{s^2+4s+5} \right] \frac{1}{s}$$

και

$$y_1(t) = 0,6 - e^{-2t}[0,6 \cos(t) + 0,2 \sin(t)]$$

$$y_2(t) = 0,2 - e^{-2t}[0,2 \cos(t) + 0,4 \sin(t)]$$

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.4/1](#).

Παράδειγμα 4.4.2: Προσδιορισμός $y(t)$ εάν $u(t)=1$ (παρουσία διαταραχών)

Δίνεται το παρακάτω σύστημα γραμμικών διαφορικών εξισώσεων:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} 1 + 3H(t-10) - 1H(t-35) \\ 3 - 2H(t-20) \end{bmatrix}$$

$$d(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix} = \begin{bmatrix} 2H(t-15) \\ 3 - 1H(t-5) + 4H(t-60) \end{bmatrix}$$

Ο πίνακας $A = \begin{bmatrix} -1 & -2 \\ 1 & -3 \end{bmatrix}$.

Γνωρίζοντας από θεωρία ότι:

$$G_p(s) = \frac{y(s)}{u(s)} = [C(sI - A)^{-1}B + D]$$

προκύπτουν: $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 3 & 1 \\ 5 & -1 \end{bmatrix}$, $D = 0$ και $E = \begin{bmatrix} 1 & 1 \\ 3 & -1 \end{bmatrix}$

Επίλυση: Λύνουμε το συγκεκριμένο παράδειγμα με παρόμοιο τρόπο όπως και στο προηγούμενο. Η διαφορά εδώ είναι η παρουσία διαταραχών. Επομένως, μαζί με τους πίνακες A,B,C και D, δηλώνουμε και τον πίνακα διαταραχών E.

```
clc
clear

syms s

A=[-1 -2; 1 -3];
B=[3 1;5 -1];
x0=[0;0];
C=[1 0;0 1];
D=0;
E=[1 1;3 -1];
```

Όπως και στο προηγούμενο παράδειγμα, δηλώνουμε το μέγεθος του πίνακα A, τον μοναδιαίο πίνακα I και προσδιορίζουμε τις $u(s)$, $d(s)$, $x(s)$, $y(s)$ και την συνάρτηση μεταφοράς $G_p(s)$.

```
size_A=rank(A);
I=eye(size_A);
u_s=[(1/s)*(1+3*exp(-10*s)-1*exp(-35*s));(1/s)*(3-2*exp(-20*s))];
d_s=[(2/s)*exp(-15*s);(1/s)*(2-1*exp(-45*s)+4*exp(-60*s))];

x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s)+(inv(s*I-A)*E*d_s);
y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s+[C*inv(s*I-A)*E]*d_s;
Gp=[C*inv(s*I-A)*B+D]
Gd=[C*inv(s*I-A)*B+E]
```

Τέλος, χρησιμοποιούμε τις γνωστές εντολές *ilaplace*, *subs* και *double* και παρουσιάζουμε γραφικά τα αποτελέσματά μας.

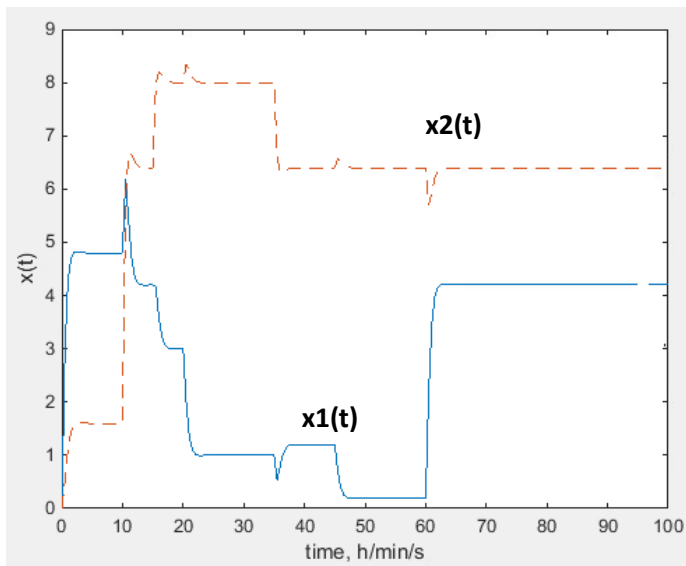
```
u_t=ilaplace(u_s);
d_t=ilaplace(d_s);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)

Total_Time=100;
t=0:0.5:Total_Time;

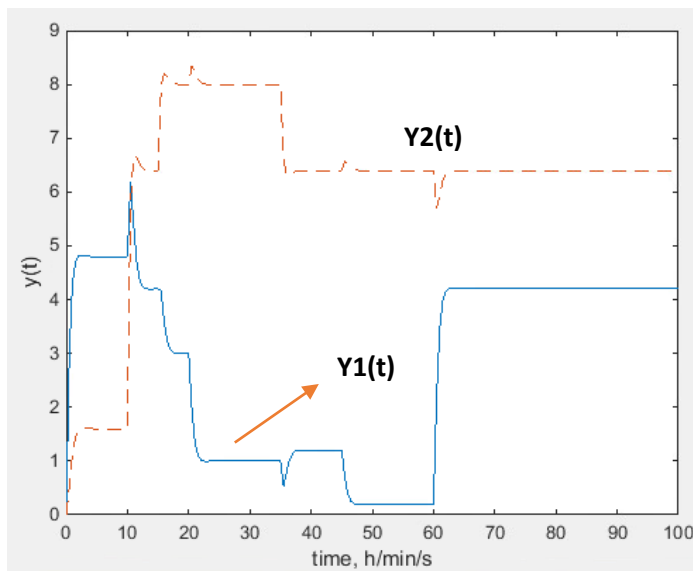
u_t=subs(u_t,t);
u_t=double(u_t);
d_t=subs(d_t,t);
d_t=double(d_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:),t,y_t(2,:), '--');xlabel('time,
h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t(1,:),t,u_t(2,:), '--');xlabel('time,
h/min/s');ylabel('u(t)')
figure(4);plot(t,d_t(1,:),t,d_t(2,:), '--');xlabel('time,
h/min/s');ylabel('d(t)')
```

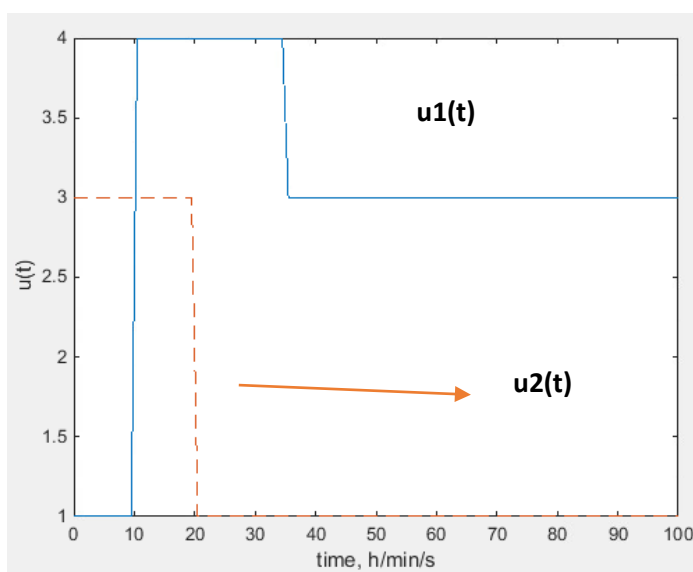
Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:



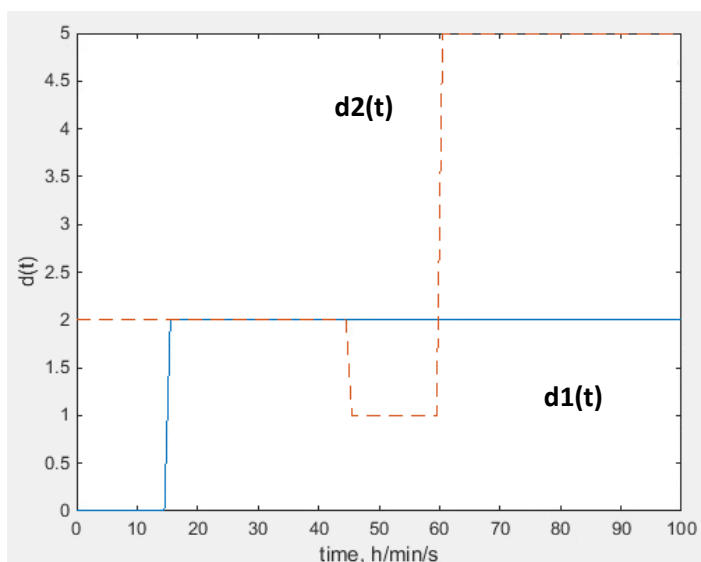
Διάγραμμα χρονικών αποκρίσεων x_1 , x_2



Διάγραμμα σημάτων εξόδου y_1 , y_2



Διάγραμμα σημάτων εισόδου u_1 , u_2



Διάγραμμα διαταραχών d_1 , d_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [4.4/2](#).

ΚΕΦΑΛΑΙΟ 5: Ιδιότητες συστημάτων πολλών εισόδων – πολλών εξόδων

Στο παρόν κεφάλαιο θα εστιάσουμε στα πολυμετάβλητα συστήματα και με την βοήθεια των ιδιοτιμών, των ιδιοδιανυσμάτων και χρησιμοποιώντας γραμμικούς μετασχηματισμούς, θα εξετάσουμε κάποιες βασικές ιδιότητες των συστημάτων αυτών. Αυτές είναι:

- Ευστάθεια συστήματος
- Ελεγχιμότητα
- Παρατηρησιμότητα

5.1 Ευστάθεια Πολυμεταβλητών Συστημάτων

Γνωρίζουμε ότι ένα σύστημα ονομάζεται ασυμπτωτικά ευσταθές, όταν η απόκριση $x(t)$ επιστρέφει στην τιμή $x(t_0=0)=0$ καθώς $t \rightarrow \infty$. Συγκεκριμένα, για να προσδιορίσουμε εάν ένα σύστημα είναι ευσταθές θα πρέπει να υπολογίσουμε τις ιδιοτιμές του και βάσει της τιμής τους θα καθοριστεί η ευστάθεια του συστήματος.

Παράδειγμα 5.1.1: Προσδιορισμός ευστάθειας πολυμετάβλητου συστήματος

Να ελέγξετε την ευστάθεια των παρακάτω συστημάτων βάσει του Πίνακα Α.

$\dot{x}(t) = Ax(t)$, $x(t=0) = x_0$ όπου A = πίνακας όπως παρακάτω ανά περίπτωση

Επίλυση: Κατά τα γνωστά ξεκινάμε με τις εντολές `clc` και `clear` και στη συνέχεια προσδιορίζουμε τον πίνακα A . Στο παρόν παράδειγμα θα χρησιμοποιήσουμε τον πίνακα A_2 , αλλά θα μπορούσαμε να δουλέψουμε και με κάποιον από τους A_1 , A_3 και A_4 .

```
clc
clear

A1=[2 1;-2 3];
A2=[-2 1;-2 -3];
A3=[-2 -1 1;2 -3 -2;-1 -3 -4];
A4=[-2 1;2 -1];
A=A2;
```

Στη συνέχεια προσδιορίζουμε τις ιδιοτιμές του πίνακα A , χρησιμοποιώντας την εντολή `eig` και υπολογίζουμε το πλήθος τους.

```
L=eig(A)
[k1,k2]=size(L);
```

Έπειτα δημιουργούμε μια δομή επανάληψης με την εντολή `for`, έτσι ώστε να ελέγξουμε το πρόσημο των ιδιοτιμών. Με τις εντολές ελέγχου `if`, `else if` και `else` θα πραγματοποιείται έλεγχος για το αν το σύστημα είναι ευσταθές, οριακά ευσταθές ή ασταθές. Σε κάθε περίπτωση με την εντολή `disp` θα εκτυπώνεται σχετικό μήνυμα για την ευστάθεια του συστήματος.

```

for i=1:1:k1
if real(L(i))<0
    disp('the system is stable')
elseif real(L(i))==0
    disp('the system is marginally stable')
else
    disp('the system is unstable')
end
end
end

```

Τέλος επιλύουμε το σύστημα διαφορικών εξισώσεων όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο.

$$\dot{x}(t) = Ax(t), \quad x(t=0) = x_0, \quad x(t) = e^{At} x_0$$

```

syms t

x0=[1;1];
exp_At=expm(A*t);

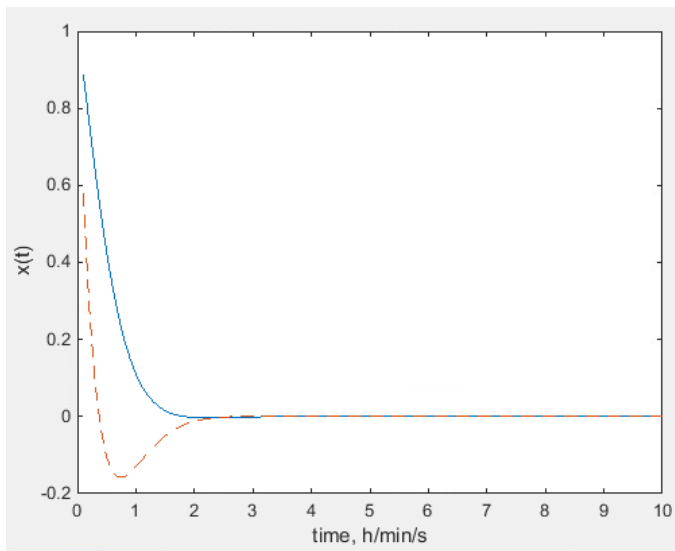
Total_Time=10;
t=0:0.1:Total_Time;

x_t=exp_At*x0;
x_t=subs(x_t,t);
x_t=double(x_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:),'--');xlabel('time,
h/min/s');ylabel('x(t)')

```

Μετά την εκτέλεση του κώδικα προκύπτει το παρακάτω διάγραμμα για τον πίνακα :



Διάγραμμα χρονικών αποκρίσεων x_1, x_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [5.1/1](#).

5.2 Πόλοι/Μηδενικά Πολυμετάβλητου Συστήματος

Για την εύρεση των πόλων και των μηδενικών ενός συστήματος θα πρέπει να υπολογίσουμε τις ιδιοτιμές καθώς και τον πίνακα Rosebrock, όπως παρουσιάστηκε στην θεωρία.

$$P = \begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix}$$

Παράδειγμα 5.2.1: Υπολογισμός πόλων/μηδενικών πολυμετάβλητου συστήματος

Πως προσδιορίζονται οι όλοι και τα μηδενικά της G_p σε ένα σύστημα διαφορικών εξισώσεων υπό μορφή μοντέλου χώρου κατάστασης (A, B, C, D).

$$G_p(s) = \frac{y(s)}{u(s)} = [C(sI - A)^{-1}B + D]$$

Με $A = \begin{bmatrix} -1 & -2 \\ 1 & -3 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ και $D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Επίλυση: Ξεκινάμε με τις κλασσικές εντολές *clc* και *clear*. Έπειτα ορίζουμε πως θα εργασθούμε με συμβολικές μεταβλητές, ενώ παράλληλα, ορίζουμε τους παραπάνω πίνακες του συστήματός μας καθώς κι έναν μοναδιαίο πίνακα, διάστασης ίσης με τη διάσταση του πίνακα A. Η εντολή που χρησιμοποιούμε για την κατασκευή του μοναδιαίου πίνακα, είναι η εντολή *eye*.

```

clc
clear
syms s

A=[-1 -2; 1 -3];
B=[1;0];
C=[1 0;0 1];
D=[0;0];

size_A=rank(A);
I=eye(size_A);

```

Στη συνέχεια, ορίζουμε και τον πίνακα Rosenbrock και υπολογίζουμε τις ιδιοτιμές του συστήματος ,χρησιμοποιώντας την εντολή *eig*, οι οποίες αντιπροσωπεύουν τους πόλους του συστήματός μας.

```

P_rosenbrock=[s*I-A -B;C D];

Eigenvalues_A=eig(A);
System_Poles=Eigenvalues_A

```

Στο σημείο αυτό, για τον υπολογισμό των μηδενικών του παραπάνω συστήματος, χρησιμοποιείται μια περίπλοκη διαδικασία για την εύρεση της ελάχιστης ορίζουσας του πίνακα Rosenbrock, καθώς δεν είναι τετραγωνικός.

```

[m_p,n_p]=size(P_rosenbrock);

if m_p>n_p
    Size_P=n_p;
else
    Size_P=m_p;
end
for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_1(i,j)=P_rosenbrock(i,j);
    end
end

Zeros=det(P_rosenbrock_1);
Zeros=sym2poly(Zeros);
Zeros=roots(Zeros)

```

Με την εκτέλεση του κώδικα βλέπουμε ότι το σύστημα μας έχει δύο πόλους και ένα μηδενικό

```
System_Poles =  
  
    -2.0000 + 1.0000i  
    -2.0000 - 1.0000i  
  
Zeros =  
  
    -3
```

ΣΗΜΕΙΩΣΗ: Καθότι έχουμε ένα μηδενικό μπορέσαμε να χρησιμοποιήσουμε την παραπάνω κωδικοποίηση για τον υπολογισμό του. Παρακάτω θα παρουσιασθεί πως χειριζόμαστε την κωδικοποίηση σε περίπτωση που έχουμε παραπάνω μηδενικά.

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [5.2/1](#).

5.3 Γραμμικοί Μετασχηματισμοί

Όπως αναφέρθηκε στην θεωρία, είναι εφικτό να μετασχηματίσουμε ένα διάνυσμα μεταβλητών x σε ένα αντίστοιχο διάνυσμα x' , χρησιμοποιώντας ένα ανεξάρτητο γραμμικό σετ. Ο μετασχηματισμός που θα παρουσιασθεί στο παρακάτω παράδειγμα, μπορεί να χρησιμοποιηθεί εάν οι βοηθητικές μεταβλητές (x_1, x_2, \dots) δεν έχουν επιλεχθεί σωστά.

Παράδειγμα 5.3.1: Προσδιορισμός συνάρτησης μεταφοράς δύο συστημάτων (αρχικό και μετασχηματισμένο)

Να προσδιορίσετε την συνάρτηση μεταφοράς, τους πόλους και τα μηδενικά των δύο συστημάτων (αρχικού και μετασχηματισμένου).

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

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

Όπου, $A = \begin{bmatrix} -2 & 1 \\ 2 & -3 \end{bmatrix}$, $B = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ και $D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Επίλυση: Ξεκινάμε με τις εντολές `clc`, `clear` και `syms` και στη συνέχεια ορίζουμε τα στοιχεία του προβλήματος, δηλαδή τους πίνακες A , B , C και D , καθώς επίσης τον πίνακα M και τον διαγώνιο πίνακα ιδιοτιμών L .

```
clc  
clear  
  
syms s t  
  
A=[-2 1; 2 -3];
```

```

B=[3;5];
C=[1 0;0 1];
D=[0;0];
[M, L]=eig(A);

```

Στη συνέχεια, ορίζουμε τους μετασχηματισμούς A' , B' , C' και D' καθώς και τις συναρτήσεις μεταφοράς και για τα δύο συστήματα λαμβάνοντας υπόψιν τους παρακάτω μετασχηματισμούς που είδαμε στην θεωρία:

$$\begin{array}{ll}
 \dot{X}(t) = AX(t) + BU(t) & \\
 Y(t) = CX(t) + BU(t) & \\
 \dot{X}'(t) = A'X'(t) + B'U(t) & \longrightarrow \dot{X}'(t) = (P^{-1}AP)X'(t) + (P^{-1}B)U(t) \\
 Y(t) = C'X'(t) + DU(t) & Y(t) = (C'P)X'(t) + DU(t) \\
 & M = P
 \end{array}$$

```

A_transform=(inv(M)*A*M)
B_transform=(inv(M)*B)
C_transform=C*M
D_transform=D

na=size(A);
I=eye(na);
Gp=[C*(inv(s*I-A))*B+D];
Gp_transform=[C_transform*(inv(s*I-A))*B_transform+D_transform];

```

Έπειτα ορίζουμε τους πίνακες Rosenbrock και για τα δύο συστήματα (αρχικό και μετασχηματισμένο), αλλά θα επιλέγουμε κάθε φορά ποιον θα χρησιμοποιήσουμε. Με τον τρόπο που παρουσιάστηκε στην προηγούμενη παράγραφο θα υπολογίσουμε τα μηδενικά μας.

```

%lo mideniko
P_rosenbrock_main_1=[s*I-A -B;C(1,:) D(1)];
P_rosenbrock_transform_1=[s*I-A_transform -B_transform;C_transform(1,:)
D_transform(1)];

P_rosenbrock_1=P_rosenbrock_main_1;
% P_rosenbrock_1=P_rosenbrock_transform_1;
[m_p,n_p]=size(P_rosenbrock_1);

if m_p>n_p

```



```

        Size_P=n_p;
else
        Size_P=m_p;
end
for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_1(i,j)=P_rosenbrock_1(i,j);
    end
end
Zeros=det(P_rosenbrock_1);
Zeros=sym2poly(Zeros);
Zeros_1=roots(Zeros)

```

Με τον ίδιο τρόπο και για το 2^ο μηδενικό.

```

% 2ο mideniko
P_rosenbrock_main_2=[s*I-A -B;C(2,:) D(2)];
P_rosenbrock_transform_2=[s*I-A_transform -B_transform;C_transform(2,:)
D_transform(2)];

P_rosenbrock_2=P_rosenbrock_main_2;
% P_rosenbrock_2=P_rosenbrock_transform_2;

[m_p,n_p]=size(P_rosenbrock_2);

if m_p>n_p
    Size_P=n_p;
else
    Size_P=m_p;
end

for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_2(i,j)=P_rosenbrock_2(i,j);
    end
end
Zeros=det(P_rosenbrock_2);
Zeros=sym2poly(Zeros);
Zeros_2=roots(Zeros)

```

Τέλος, επιλύουμε τα παραπάνω συστήματα (αρχικό και μετασχηματισμένο) και αναπαριστούμε γραφικά τις αποκρίσεις συναρτήσεως του χρόνου.

```
u_s=1/s;
x0=[0;0];

x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
x_s_transform=(inv(s*I-A_transform)*x0)+(inv(s*I-A_transform)*B_transform*u_s);

y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s;
y_s_transform=C_transform*inv(s*I-A_transform)*x0+[C_transform*inv(s*I-A_transform)*B_transform+D_transform]*u_s;

x_t=ilaplace(x_s);
y_t=ilaplace(y_s);

x_t_transform=ilaplace(x_s_transform);
y_t_transform=ilaplace(y_s_transform);

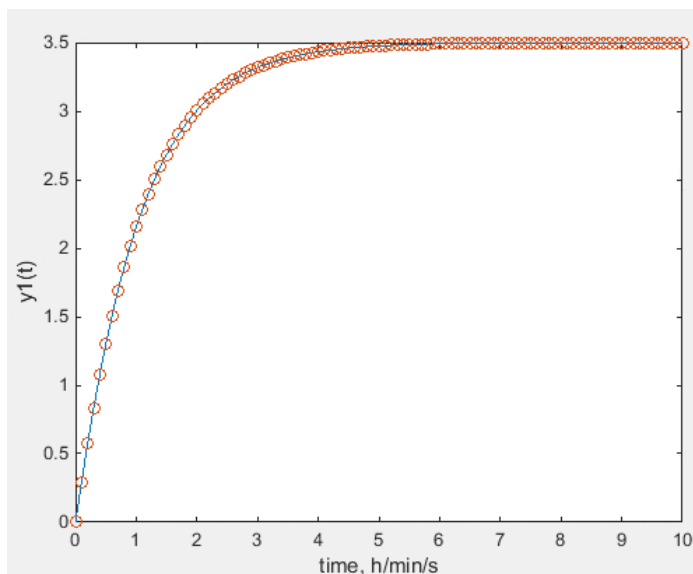
Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);

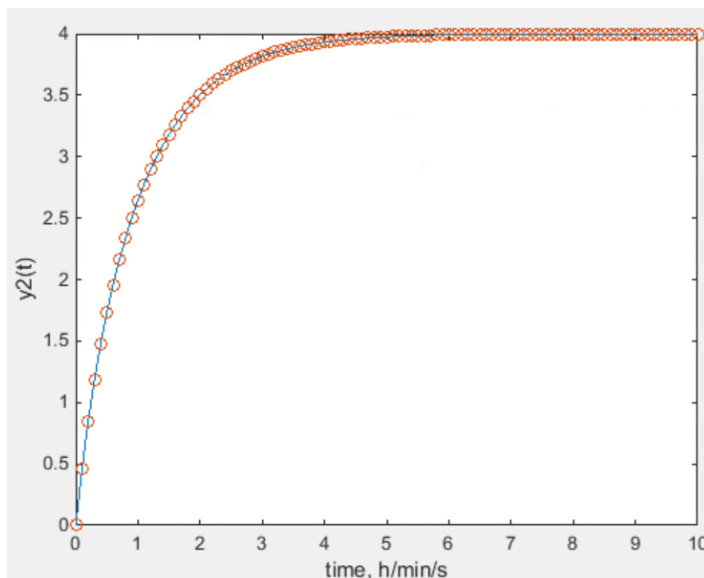
x_t_transform=subs(x_t_transform,t);
x_t_transform=double(x_t_transform);
y_t_transform=subs(y_t_transform,t);
y_t_transform=double(y_t_transform);

figure(1);plot(t,y_t(1,:),t,y_t_transform(1,),'o');xlabel('time, h/min/s');ylabel('y1(t)')
figure(2);plot(t,y_t(2,:),t,y_t_transform(2,),'o');xlabel('time, h/min/s');ylabel('y1(t)')
```

Με την εκτέλεση του κώδικα θα πρέπει να δημιουργηθούν δύο πανομοιότυπα διαγράμματα εφόσον ο μετασχηματισμός μας είναι σωστός.



Χρονική απόκριση y_1 αρχικού συστήματος



Χρονική απόκριση y_2 μετασχηματισμένου συστήματος

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [5.3/1](#).

5.4 Ελεγχιμότητα/ Παρατηρησιμότητα

Η ελεγχιμότητα και η παρατηρησιμότητα, είναι δύο βασικές ιδιότητες που ελέγχουμε στα μοντέλα χώρου κατάστασης. Από τη θεωρία γνωρίζουμε τα εξής:

«Ένα σύστημα καλείται ελέγξιμο αν για οποιαδήποτε χρονική στιγμή t_0 που χαρακτηρίζει την αρχική κατάσταση $X(t_0)$, υπάρχει ένα διάνυσμα εισόδου $u(t)$ που μπορεί να μεταφέρει την κατάσταση $X(t_0)$ στην τελική (και επιθυμητή) κατάσταση $X(t_1)$ στον χρόνο t_1 .»

«Ένα σύστημα καλείται παρατηρήσιμο αν για οποιαδήποτε χρονική στιγμή t_0 που χαρακτηρίζει την αρχική κατάσταση $X(t_0)$, υπάρχει ένα διάστημα $[t_0, t_1]$ τέτοιο ώστε αν τα διανύσματα εισόδου u και εξόδου y είναι γνωστά, να δύναται να προσδιορισθεί η αρχική κατάσταση $X(t_0)$.»

Παράδειγμα 5.4.1: Έλεγχος ελεγχσιμότητας και παρατηρησιμότητας συστημάτων

Να ελέγξετε αν το παρακάτω σύστημα είναι ελέγξιμο/παρατηρήσιμο.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 2 \\ 0 & 0 & 0 & 4 \\ 0 & -2 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = [0 \quad 1 \quad 1 \quad 1] \text{ και } D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Επίλυση: Ξεκινώντας με τις εντολές *clc* και *clear*, κατά τα γνωστά, κι έπειτα ορίζουμε τους πίνακες του συστήματος A, B, C και D.

```
clc
clear

A=[0 1 0 0;3 0 0 2;0 0 0 4;0 -2 1 0];
B=[0 0;1 0;0 0;0 1];
C=[0 1 1 1];
D=[0;0];
```

Αρχικά, για να χαρακτηριστεί το παραπάνω σύστημα ελέγξιμο, θα πρέπει το μέγεθος του πίνακα A να ισούται με το μέγεθος του επαυξημένου πίνακα *ctrb* (A, B). Όπου επαυξημένος πίνακας ορίζεται:

$$ctrb(A,B) = [B \quad AB \quad A^2B \quad A^3B \quad \dots \quad A^{n-1}B]$$

Με τις εντολές *if* και *else* θα ελέγχουμε αν το μέγεθος των δύο πινάκων είναι ίδιο και θα βγαίνει αντίστοιχο μήνυμα στην οθόνη με τη χρήση της εντολής *disp* για το αν το σύστημα είναι ελέγξιμο ή όχι.

```
Contr=ctrb(A,B);

Rank_A=rank(A);
Rank_Contr=rank(Contr);

if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
end
```

Με όμοιο τρόπο ελέγχουμε και την παρατηρησιμότητα του συστήματος, χρησιμοποιώντας τον επαυξημένο πίνακα *obsn* (A, C).

Όπου,

$$obsn(A,C) = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ \dots \\ CA^{n-1} \end{bmatrix}$$

```

Observ=obsv(A,C);
Rank_Observ=rank(Observ);

if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
end

```

Μετά την εκτέλεση του κώδικα βλέπουμε ότι το σύστημα μας είναι ελέγξιμο και παρατηρήσιμο. Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [5.4/1](#).

Εξάσκηση:

Βάσει του παραπάνω παραδείγματος:

Να ελέγξετε αν το παρακάτω σύστημα είναι ελέγξιμο και παρατηρήσιμο.

$$A = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 2 \\ 0 & 0 & 4 & 0 \\ 0 & -2 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, C = [1 \ 0 \ 0 \ 0] \text{ και } D = 0.$$

Επίλυση: Θα πρέπει στο πρώτο βήμα του παραπάνω παραδείγματος να αντικαταστήσετε τους πίνακες A, B, C και D.

Μετά την εκτέλεση το κώδικα θα διαπιστώσετε ότι το παραπάνω σύστημα είναι μη ελέγξιμο και μη παρατηρήσιμο.

5.5 Συνάρτηση Μεταφοράς & Μοντέλο Χώρου Κατάστασης

Στο παρακάτω παράδειγμα παρουσιάζεται η κατασκευή μιας συνάρτησης μεταφοράς προς ένα μοντέλο χώρου κατάστασης που είναι ελέγξιμο και παρατηρήσιμο.

Παράδειγμα 5.5.1: Προσδιορισμός Συνάρτησης Μεταφοράς

Επίλυση: Ξεκινάμε με τις εντολές *clc* και *clear*. Συνεχίζοντας δηλώνουμε τον αριθμητή και τον παρονομαστή (*num*, *den*) της συνάρτησης μεταφοράς και την κατασκευάζουμε με την εντολή *tf*. Έπειτα μετατρέπουμε την συνάρτηση μεταφοράς σε μοντέλο χώρου κατάστασης χρησιμοποιώντας την εντολή *tf2ss*.

```

clc
clear

num=[3 2 -23 -22];
den=[1 0 -7 -6];
G=tf(num,den)

```

```
[A, B, C, D]=tf2ss(num,den)
```

Στη συνέχεια, κάνουμε το αντίστροφο. Δηλαδή ορίζουμε τον μοντέλο χώρου κατάστασης (πίνακες A1, B1, C1 και D1), με την εντολή *ss2tf* το μετατρέπουμε σε συνάρτηση μεταφοράς και με την *tf* την κατασκευάζουμε.

```
A1=[-2 1; 2 -3]
B1=[0;1]
C1=[0 1]
D1=0

[num1,den1]=ss2tf(A1,B1,C1,D1);
G=tf(num1,den1)
```

Στο σημείο αυτό, πραγματοποιούμε τους σχετικούς ελέγχους για την ελεγχιμότητα και την παρατηρησιμότητα του συστήματος που παρουσιάστηκαν στο προηγούμενο παράδειγμα.

```
Contr=ctrb(A,B);

Rank_A=rank(A);
Rank_Contr=rank(Contr);

if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
end

Observ=obsv(A,C);

Rank_Observ=rank(Observ);

if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
end
```

Μετά την εκτέλεση του κώδικα έχουμε τα εξής αποτελέσματα:

Command Window

```
G =  
  
      3 s^3 + 2 s^2 - 23 s - 22  
      -----  
      s^3 - 7 s - 6  
  
Continuous-time transfer function.  
  
A =  
  
      0      7      6  
      1      0      0  
      0      1      0  
  
B =  
  
      1  
      0  
      0  
  
C =  
  
      2      -2      -4  
  
D =  
  
      3
```

```
A1 =  
  
      -2      1  
      2      -3  
  
B1 =  
  
      0  
      1  
  
C1 =  
  
      0      1  
  
D1 =  
  
      0  
  
G =  
  
      s + 2  
      -----  
      s^2 + 5 s + 4  
  
Continuous-time transfer function.
```

Μετατροπή συνάρτησης μεταφοράς σε μοντέλο χώρου κατάστασης

Μετατροπή μοντέλου χώρου κατάστασης σε συνάρτηση μεταφοράς

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [5.5/1](#).

ΚΕΦΑΛΑΙΟ 6: Σχεδιασμός Ελεγκτών/Γραμμική Ανάδραση Καταστάσεων

Στο παρόν κεφάλαιο θα παρουσιασθούν παραδείγματα σχεδιασμού ελεγκτών/γραμμικών αναδράσεων κατάστασης.

6.1 Γραμμική Ανάδραση Καταστάσεων

Γνωρίζουμε ότι με την γραμμική ανάδραση κατάστασης «Κ», μπορούμε να τοποθετήσουμε τους πόλους του κλειστού βρόχου όπου εμείς επιθυμούμε. Επομένως, ένα ασταθές σύστημα μπορεί να μετατραπεί σε ευσταθές.

Παράδειγμα 6.1.1: Προσδιορισμός ανύσματος κατάστασης «Κ»

Να προσδιορισθεί το άνυσμα κατάστασης K , έτσι ώστε οι πόλοι του κλειστού βρόχου να μεταφερθούν στα ευσταθή σημεία -1 , -2 .

$$\dot{x}(t) = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

$$y = [1 \quad 2] x(t) + [0] u(t)$$

Όπου $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $C = [1 \quad 2]$ και $D = 0$.

Επίλυση: Ξεκινάμε με τις κλασσικές εντολές `clc` και `clear`. Έπειτα προσδιορίζουμε το μοντέλο χώρου κατάστασης με τους πίνακες A , B , C , D .

```
clc
clear

A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
```

Στη συνέχεια μπορούμε να προσδιορίσουμε την ευστάθεια του συστήματος ανοικτού βρόχου όπως παρουσιάστηκε και στο προηγούμενο κεφάλαιο.


```

L_open=eig(A);
[k1,k2]=size(L_open);
for i=1:1:k1
if real(L_open(i))<0
    disp('the open loop system is stable')
elseif real(L_open(i))==0
    disp('the open loop system is marginally stable')
Else
    disp('the open loop system is unstable')
End

```

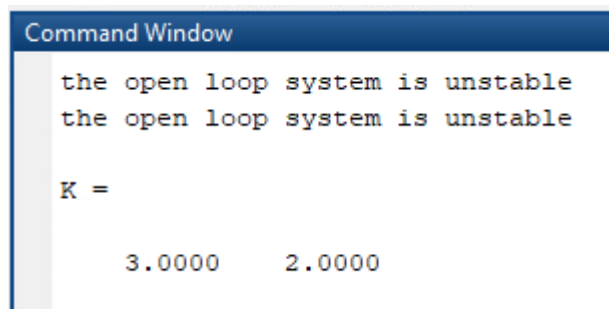
Τέλος προσδιορίζουμε το άνωσμα κατάστασης K για τους επιθυμητούς πόλους -1, -2 με την εντολή `place(A, B, P)`.

```

% % Epithymittoi Poloi Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
K=place(A,B,P)
Return

```

Μετά την εκτέλεση του κώδικα έχουμε τα εξής αποτελέσματα:



```

Command Window

the open loop system is unstable
the open loop system is unstable

K =

    3.0000    2.0000

```

Το άνωσμα κατάστασης K και η ευστάθεια του συστήματος.

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.1/1](#).

6.2 Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης

Από θεωρία γνωρίζουμε ότι ισχύουν οι παρακάτω σχέσεις:

$$\dot{x}(t) = (A - B \cdot K) x(t) + Bv$$

$$y(t) = (C - D \cdot K) x(t) + Dv$$

Χαρακτηριστικό πολυώνυμο κλειστού βρόχου “ $sl-(A-BK)$ ”

Στην συνάρτηση μεταφοράς του κλειστού βρόχου, τα μηδενικά παραμένουν ίδια με τα μηδενικά της συνάρτησης μεταφοράς του ανοικτού βρόχου, ενώ οι πόλοι μεταβάλλονται.

Παράδειγμα 6.2.1: Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης ($v=0$)

Για το προηγούμενο παράδειγμα να βρεθεί η απόκριση των $x_1(t)$, $x_2(t)$, $y(t)$ για το σύστημα κλειστού βρόχου. Θεωρήστε την αρχική κατάσταση στο $t=0$, $x_0=[1;1]$ και είσοδο $u(t)=v-K \cdot x(t)$ και $v=0$.

Επίλυση: Ξεκινάμε με τις εντολές `clc` και `clear` και ορίζουμε πως θα εργαστούμε με συμβολικές μεταβλητές με την εντολή `syms`.

```
% Solution to a closed loop system with a state feedback
clc
clear
syms s
```

Στο σημείο αυτό ορίζουμε το μοντέλο χώρου κατάστασης, δηλαδή τους πίνακες A, B, C, D όπως προσδιορίστηκαν στο προηγούμενο παράδειγμα.

```
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
```

Στη συνέχεια ορίζουμε το άνωσμα κατάστασης K για τους επιθυμητούς πόλους -1, -2.

```
% % Epithymitoi Poloi Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
K=place(A,B,P)
```

Έπειτα θα πρέπει να προσδιορίσουμε τους πίνακες κλειστού βρόχου καθώς επίσης το διάνυσμα αρχικής κατάστασης και το επιθυμητό σήμα αναφοράς (set point) υπό μορφή Laplace.

Σύμφωνα με τις σχέσεις:

$$\dot{x}(t) = (A - B \cdot K) x(t) + Bv$$

$$y(t) = (C - D \cdot K) x(t) + Dv$$

Προκύπτουν:

$$A_{new} = (A - B \cdot K)$$

$$C_{new} = (C - D \cdot K)$$

$$B_{new} = B$$

$$D_{new} = D$$

```

x0=[1;1];
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;
n_Set_point=0/s;

```

Συνεχίζοντας μπορούμε να προσδιορίσουμε την ευστάθεια του συστήματος κλειστού βρόχου και να επαληθεύσουμε την σωστή τοποθέτηση των νέων πόλων του κλειστού βρόχου στα επιθυμητά σημεία 1, -2.

```

L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
if real(L_closed(i))<0
    disp('the closed loop system is stable')
elseif real(L_closed(i))==0
    disp('the closed loop system is marginally stable')
Else
    disp('the closed loop system is unstable')
End
End

```

Τέλος, επιλύουμε το σύστημα κλειστού βρόχου με γραμμική ανάδραση κατάστασης, προσδιορίζοντας τις σχέσεις $x(s)$ και $y(s)$ και χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace. Παράλληλα δηλώνουμε τον χρόνο επίλυσης και με τη χρήση των γνωστών εντολών *subs* και *double*, θα αναπαριστήσουμε γραφικά τα αποτελέσματά μας.

```

%Epilysi Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);

x_s=(inv(s*I-A_new)*(x0+B*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B*n_Set_point))+D*n_Set_point;

x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
set_point=ilaplace(n_Set_point);

Total_Time=10;
t=0:0.1:Total_Time;

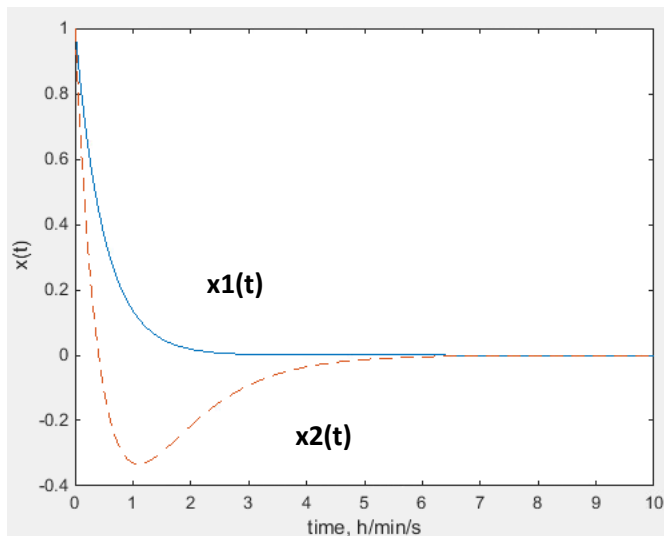
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--
');xlabel('time,h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t,'--');xlabel('time, h/min/s');ylabel('y(t)')

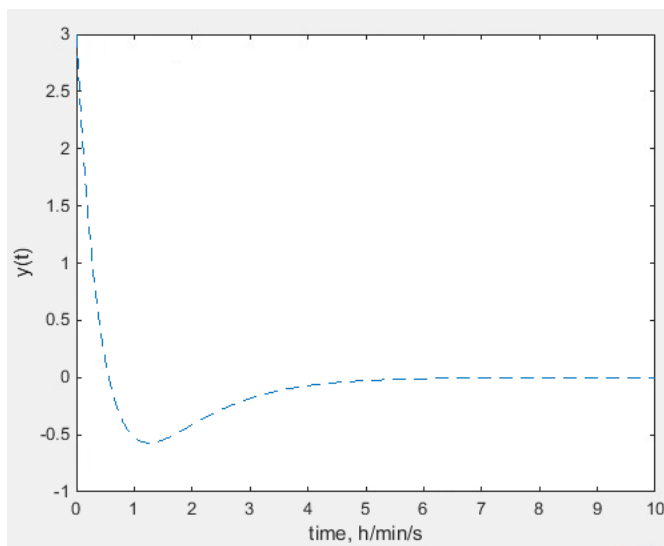
Return

```

Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:



Χρονικές αποκρίσεις $x_1(t)$, $x_2(t)$ για το σύστημα κλειστού βρόχου



Χρονική απόκριση $y(t)$ για το σύστημα κλειστού βρόχου

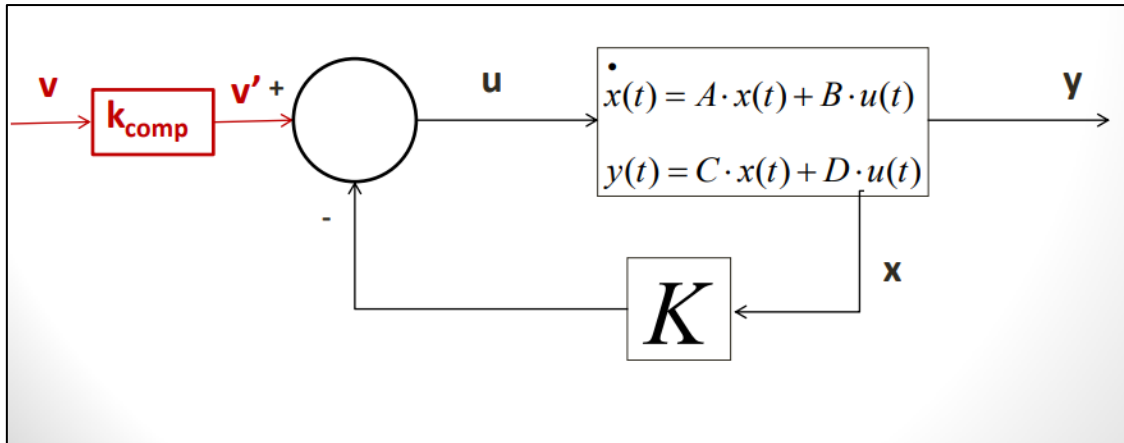
Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.2/1](#).

Παράδειγμα 6.2.2: Επίλυση Συστήματος Κλειστού Βρόχου με Γραμμική Ανάδραση Κατάστασης ($v \neq 0$)

Για το προηγούμενο παράδειγμα σας ζητείτε να βρεθεί η απόκριση των $x_1(t)$, $x_2(t)$, $y(t)$ για το σύστημα κλειστού βρόχου. Θεωρήστε την αρχική κατάσταση στο $t=0$, $x_0=[1;1]$ και είσοδο $u(t)=v \cdot k_{comp} - K \cdot x(t)$ και $v(t)=5$ ή $v(s)=5/s$.

Για την επίλυση του συγκεκριμένου παραδείγματος εισάγεται η έννοια του προ-αντισταθμιστή (k_{comp}), όπως αναλύθηκε στην θεωρία.

$$k_{comp} = [\text{SetPoint} / y(t \rightarrow \infty, \text{ εάν αρχικά } k_{comp} = 1)] = 5 / -2,4994$$



Εικόνα 6.1 Εισαγωγή προ-αντισταθμιστή σε σύστημα κλειστού βρόχου

Επίλυση: Εργαζόμαστε με παρόμοιο τρόπο όπως στο προηγούμενο παράδειγμα, ξεκινώντας και πάλι με τις εντολές *clc*, *clear* και *syms*. Ορίζουμε τους πίνακες A, B, C και D, το άνωσμα κατάστασης K για τους επιθυμητούς πόλους -1, -2 και την αρχική συνθήκη x_0 .

```

Clc
Clear
syms s
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
% Epithymitoi Poloi Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
K=place(A,B,P)
x0=[1;1];
% Anysma katastasis
K=place(A,B,P)

```

Ορίζουμε τον προ-αντισταθμιστή καθώς επίσης και τους πίνακες κλειστού βρόχου μαζί με το διάνυσμα αρχικής κατάστασης.

```

kcomp=5/(-2.4993);
x0=[1;1];
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;

```

Συνεχίζοντας, ορίζουμε το διαφορετικό setpoint σε μορφή Laplace κι έπειτα προσδιορίζουμε την ευστάθεια του συστήματος κλειστού βρόχου και επαληθεύουμε την σωστή τοποθέτηση των νέων πόλων, όπως ακριβώς κάναμε και στα προηγούμενα παραδείγματα για τα επιθυμητά σημεία -1,-2.

```
n_Set_point=5/s;

L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
    if real(L_closed(i))<0
        disp('the closed loop system is stable')
    elseif real(L_closed(i))==0
        disp('the closed loop system is marginally stable')
    else
        disp('the closed loop system is unstable')
    End
End
```

Τέλος, επιλύουμε το σύστημα κλειστού βρόχου με γραμμική ανάδραση κατάστασης, προσδιορίζοντας τις σχέσεις $x(s)$ και $y(s)$ και χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace. Παράλληλα δηλώνουμε τον χρόνο επίλυσης και με τη χρήση των γνωστών εντολών *subs* και *double*, θα αναπαριστήσουμε γραφικά τα αποτελέσματά μας.

```
%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);

x_s=(inv(s*I-A_new)*(x0+B_new*kcomp*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*kcomp*n_Set_point))+D_new*kcomp*n_Set_point;

x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
set_point=ilaplace(n_Set_point);

Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
```

```

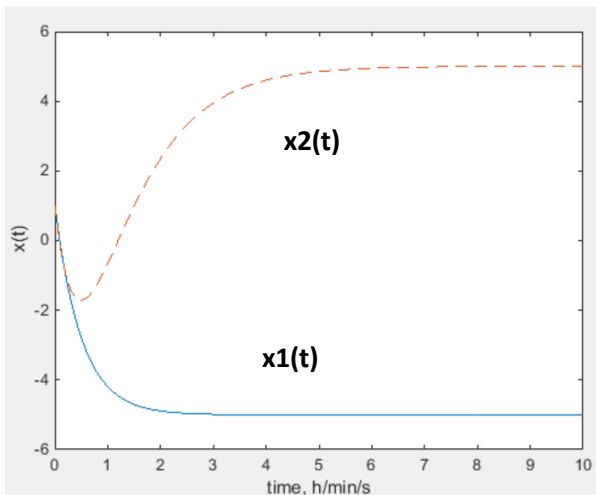
y_t=double(y_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t, '--');xlabel('time, h/min/s');ylabel('y(t)')

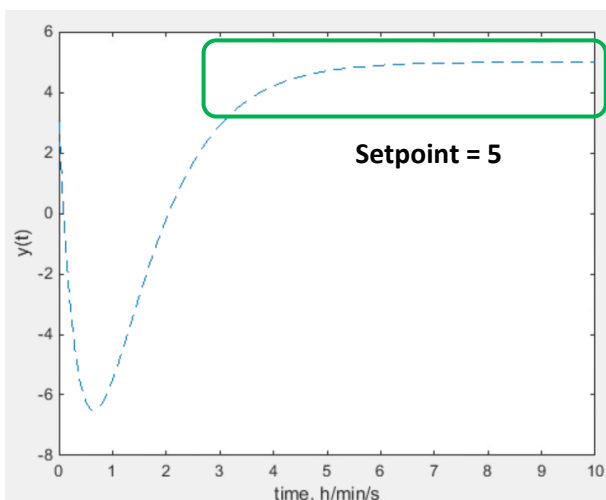
Return

```

Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:



Χρονικές αποκρίσεις $x_1(t)$, $x_2(t)$ για το σύστημα κλειστού βρόχου



Χρονική απόκριση $y(t)$ για το σύστημα κλειστού βρόχου

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.2/2](#).

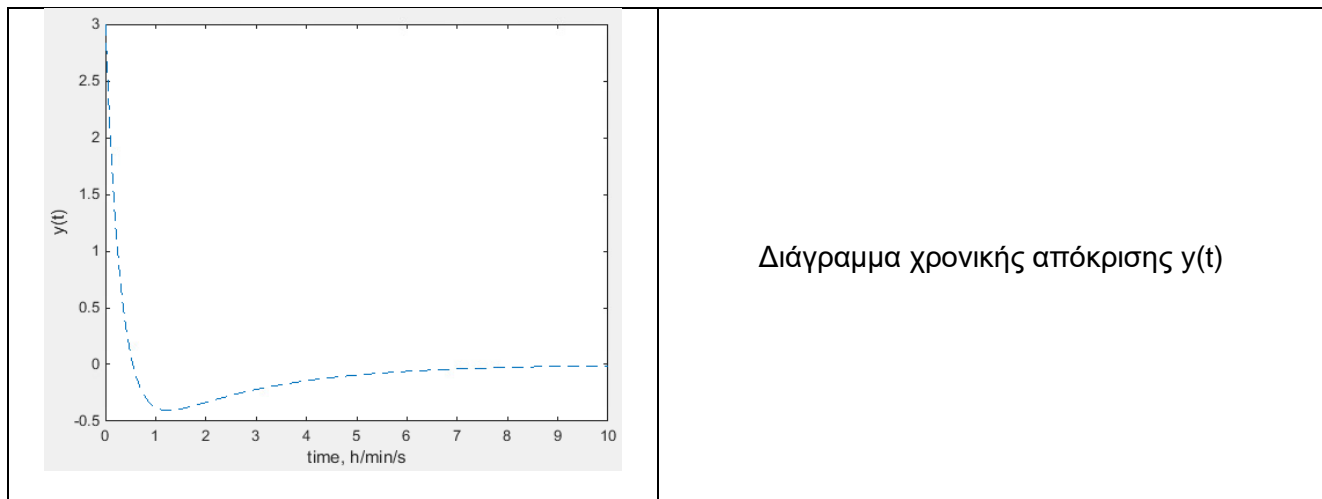
Παράδειγμα 6.2.3: Προσδιορισμός ανύσματος κατάστασης «Κ» με χρήση LQR ελεγκτή

Το πρώτο παράδειγμα της ενότητας (Παράδειγμα 6.2.1) θα μπορούσε να επιλυθεί με τη χρήση LQR ελεγκτή όπως παρουσιάστηκε και στην θεωρία. Ο κώδικας παραμένει ίδιος το μόνο που θα χρειαστεί να αλλάξει, είναι το σημείο προσδιορισμού των επιθυμητών πόλων κλειστού βρόχου και του ανύσματος κατάστασης K .

Συγκεκριμένα θα αντικατασταθούν οι παρακάτω γραμμές κώδικα:

Άνυσμα κατάσταση K χωρίς LQR ελεγκτή	Άνυσμα κατάσταση K με LQR ελεγκτή
<pre>% Epithymitoi Poloi Kleistou Broxou p1=-1; p2=-2; P=[p1 p2]; K=place(A,B,P) x0=[1;1];</pre>	<pre>% Anysma katastasis G = -B*B'; Q = C'*C; [T,POLES,K] = care(A, B, Q, 1)</pre>

Μετά την εκτέλεση του κώδικα προκύπτουν τα εξής αποτελέσματα:



Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.2/3](#).

6.3 Γραμμική Ανάδραση Καταστάσεων (ΠΕΠΕ)

Στην παρούσα παράγραφο θα παρουσιασθεί ο τρόπος επίλυσης συστημάτων πολλών εισόδων και πολλών εξόδων με γραμμική ανάδραση.

Παράδειγμα 6.3.1: Προσδιορισμός ανύσματος κατάστασης «K» για συστήματα ΠΕΠΕ

Να προσδιορισθεί το άνυσμα κατάστασης K, έτσι ώστε οι πόλοι του κλειστού βρόχου να μεταφερθούν στα ευσταθή σημεία $\theta_1=-4$, $\theta_2=-5$.

$$\dot{X}(t) = \begin{bmatrix} -1 & 0 \\ -3 & -2 \end{bmatrix} X(t) + \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} U(t)$$

Επίλυση: Ξεκινώντας χρησιμοποιούμε τις εντολές *clc*, *clear* και *syms* και έπειτα ορίζουμε το σύστημα μας (A, B, C, D).

```
clc
clear
syms s
A=[-1 0;-3 -2];
B=[1 2;2 0];
C=[1 0;0 1];
size_C=rank(C);
D=zeros(size_C); D=D1;%( :,1);
```

Στη συνέχεια προσδιορίζουμε το άνωσμα κατάστασης K για τους επιθυμητούς πόλους -4, -5 και την αρχική συνθήκη x_0 . Στο σημείο αυτό επειδή η εντολή $K=\text{place}(A,B,P)$ θα μας δώσει διαφορετικό K από αυτό που βρήκαμε στην θεωρία, μπορούμε να το προσθέσουμε σε σχόλιο και σε περίπτωση που θέλουμε να το αλλάξουμε.

```
% Epithymittoi Poloi Kleistou Broxou
p1=-4; p2=-5;
P=[p1 p2];
% Anysma katastasis
% K=[2.364 -0.545; 2.364 -0.545];
K=place(A,B,P)
x0=[1;1];
```

Έπειτα θα πρέπει να προσδιορίσουμε τους πίνακες κλειστού βρόχου καθώς επίσης και το επιθυμητό σήμα αναφοράς (set point) υπό μορφή Laplace για τις Y_1 και Y_2 .

```
A_new=A-B*K;
C_new=C-D'*K;
B_new=B;
D_new=D';
n_Set_point=[0/s;0/s];
```

Συνεχίζοντας, προσδιορίζουμε την ευστάθεια του συστήματος κλειστού βρόχου και επαληθεύουμε την σωστή τοποθέτηση των πόλων, όπως ακριβώς κάναμε και στα προηγούμενα παραδείγματα για τα επιθυμητά σημεία -4,-5.

```
L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
if real(L_closed(i))<0
    disp('the closed loop system is stable')
elseif real(L_closed(i))==0
    disp('the closed loop system is marginally stable')
else
    disp('the closed loop system is unstable')
end
end
```

Τέλος, επιλύουμε το σύστημα κλειστού βρόχου με γραμμική ανάδραση κατάστασης, προσδιορίζοντας τις σχέσεις $x(s)$, $y(s)$ και $u(s)$ χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace. Παράλληλα δηλώνουμε τον χρόνο επίλυσης και με τη χρήση των γνωστών εντολών *subs* και *double*, θα αναπαριστήσουμε γραφικά τα αποτελέσματά μας.

```

%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);

x_s=(inv(s*I-A_new)*(x0+B*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B*n_Set_point))+D*n_Set_point;
u_s=n_Set_point-K*x_s;

x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);

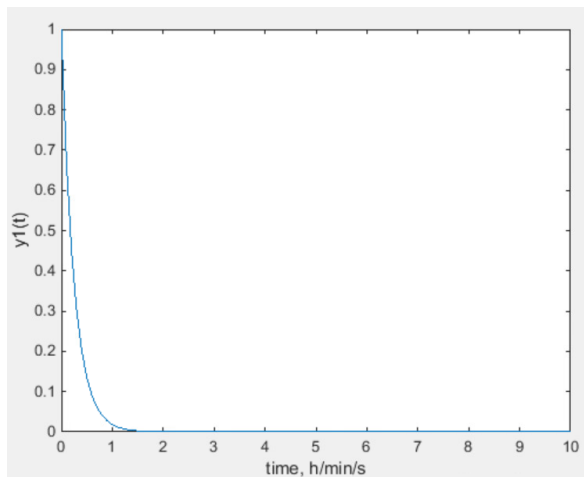
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:));xlabel('time, h/min/s');ylabel('y1(t)')
figure(3);plot(t,y_t(2,:));xlabel('time, h/min/s');ylabel('y2(t)')
figure(4);plot(t,u_t(1,:),t,u_t(2,:), '--');xlabel('time, h/min/s');ylabel('u(t)')
return

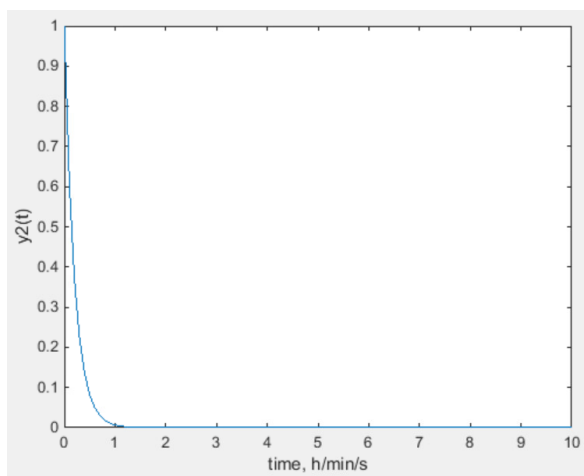
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:

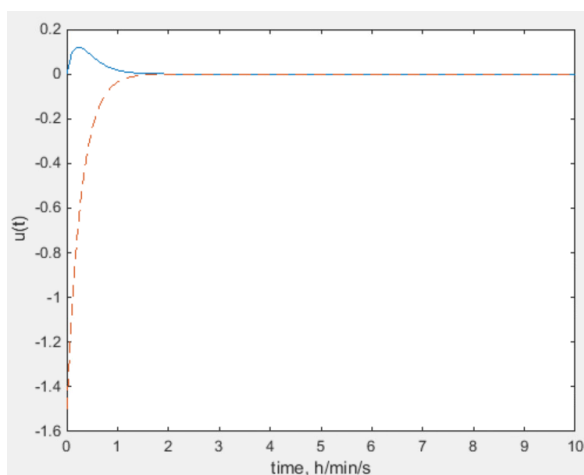
1) Αποτελέσματα MATLAB με υπολογισμένο K από τις διαθέσιμες εντολές



Χρονική απόκριση $y_1(t)$ για το σύστημα κλειστού βρόχου

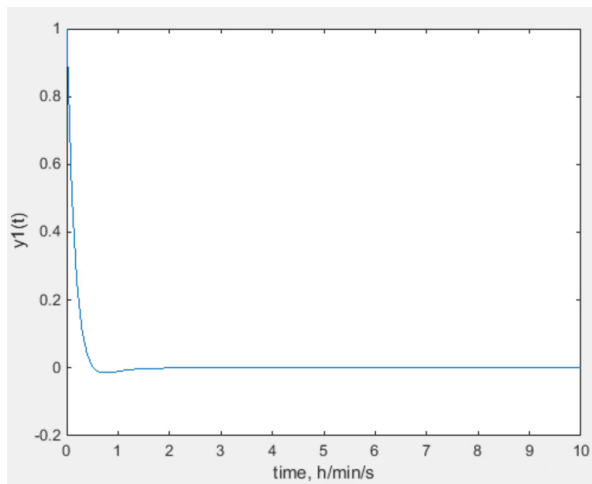


Χρονική απόκριση $y_2(t)$ για το σύστημα κλειστού βρόχου

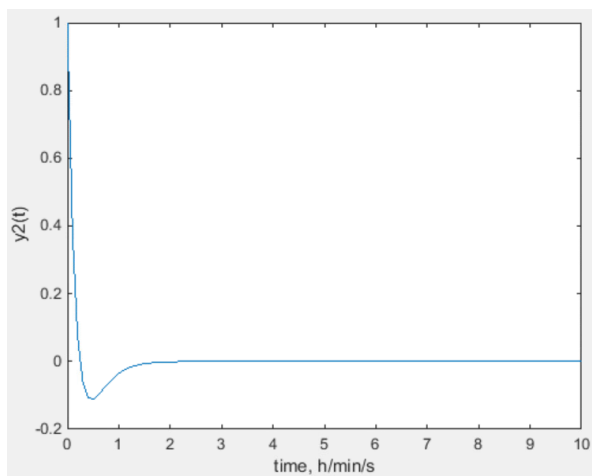


Διάγραμμα σημάτων εισόδου u_1 , u_2

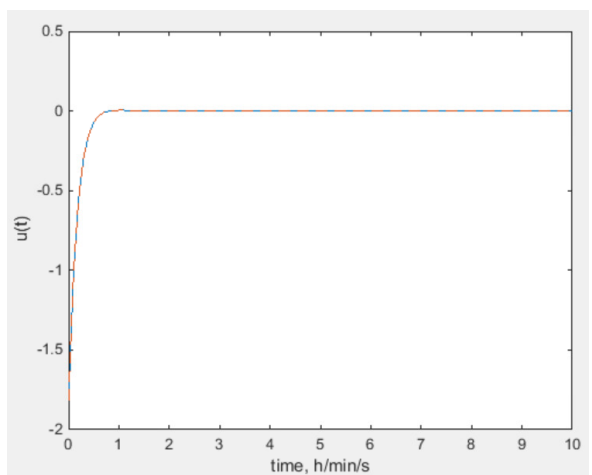
2) Αποτελέσματα MATLAB με υπολογισμένο K από τη θεωρία



Χρονική απόκριση $y_1(t)$ για το σύστημα κλειστού βρόχου



Χρονική απόκριση $y_2(t)$ για το σύστημα κλειστού βρόχου



Διάγραμμα σημάτων εισόδου u_1 , u_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.3/1](#).

Παράδειγμα 6.3.2: Προσδιορισμός ανύσματος κατάστασης «K» για συστήματα ΠΕΠΕ με χρήση LQR ελεγκτή

Το προηγούμενο παράδειγμα θα μπορούσε να επιλυθεί με τη χρήση LQR ελεγκτή και της εξίσωσης Riccati, όπως παρουσιάστηκε και στην θεωρία. Ο κώδικας παραμένει ίδιος το μόνο που θα χρειαστεί να αλλάξει, είναι το σημείο προσδιορισμού του ανύσματος κατάστασης K.

Εξίσωση Riccati:

$$TA + A^T T - \frac{1}{d} T B B^T T + Q = 0$$

Με $Q = C^T C$, $G = -B B^T$

Επομένως χρησιμοποιώντας την εντολή care δηλώνουμε την γενική μορφή της εξίσωσης Riccati, και για να υπολογιστεί το άνυσμα κατάστασης K εισάγουμε τις παρακάτω εντολές βάζοντας τους δύο προηγούμενους τρόπους υπολογισμού του ως σχόλια.

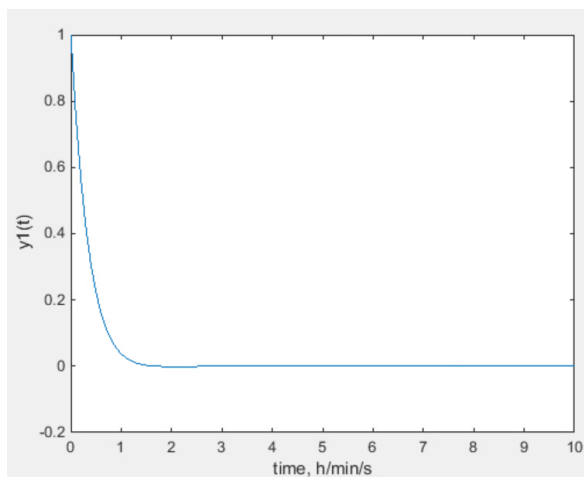
```
% Anysma katastasis
% K=[2.366 -0.545; 2.366 -0.545];
% K=place(A,B,P)

% Anysma katastasis RICCATI
G = -B*B';
Q = C'*C;

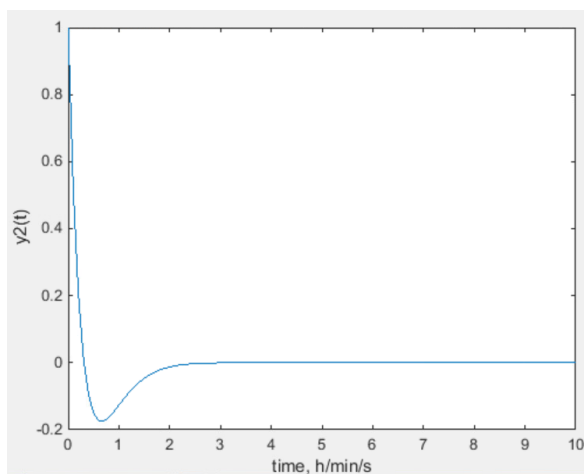
size_A=rank(A);
R=eye(size_A);
S=zeros(size_A);
E=R;

[T,POLES,K] = care(A, B, Q, R, S, E, 10)
```

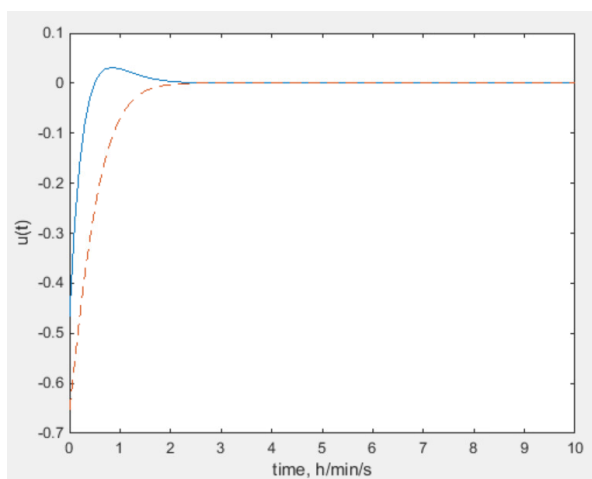
Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:



Χρονική απόκριση $y_1(t)$ για το σύστημα κλειστού βρόχου



Χρονική απόκριση $y_2(t)$ για το σύστημα κλειστού βρόχου

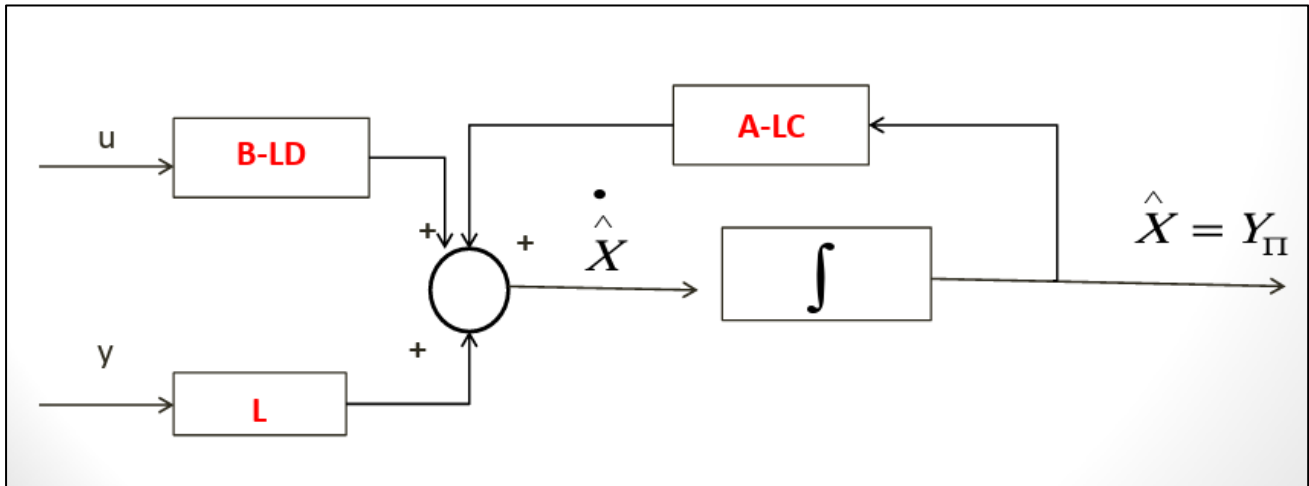


Διάγραμμα σημάτων εισόδου u_1 , u_2

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [6.3/2](#).

ΚΕΦΑΛΑΙΟ 7: Παρατηρητές Κατάστασης

Στο παρόν κεφάλαιο θα παρουσιασθούν παραδείγματα σχεδιασμού παρατηρητών κατάστασης.



Εικόνα 7.1, Διάγραμμα βαθμίδων με παρατηρητή κατάστασης

Ο παρατηρητής είναι μία συσκευή ή ένας αλγόριθμος (όπως και ο ελεγκτής δηλαδή) που «εκτιμάει» τις μεταβλητές κατάστασης ενός συστήματος.

$$\frac{d\hat{X}}{dt} = (A - LC)\hat{X} + (B - LD)u + Ly$$

Από την θεωρία έχουμε ένα σύνθετο-επαυξημένο σύστημα που περιλαμβάνει το αρχικό σύστημα (μοντέλο χώρου κατάστασης) αλλά και τον παρατηρητή μας.

Όπου αρχικό σύστημα:

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

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

Παρατηρητής:

$$\frac{d\hat{X}}{dt} = (A - LC)\hat{X} + (B - LD)u + Ly$$

ή

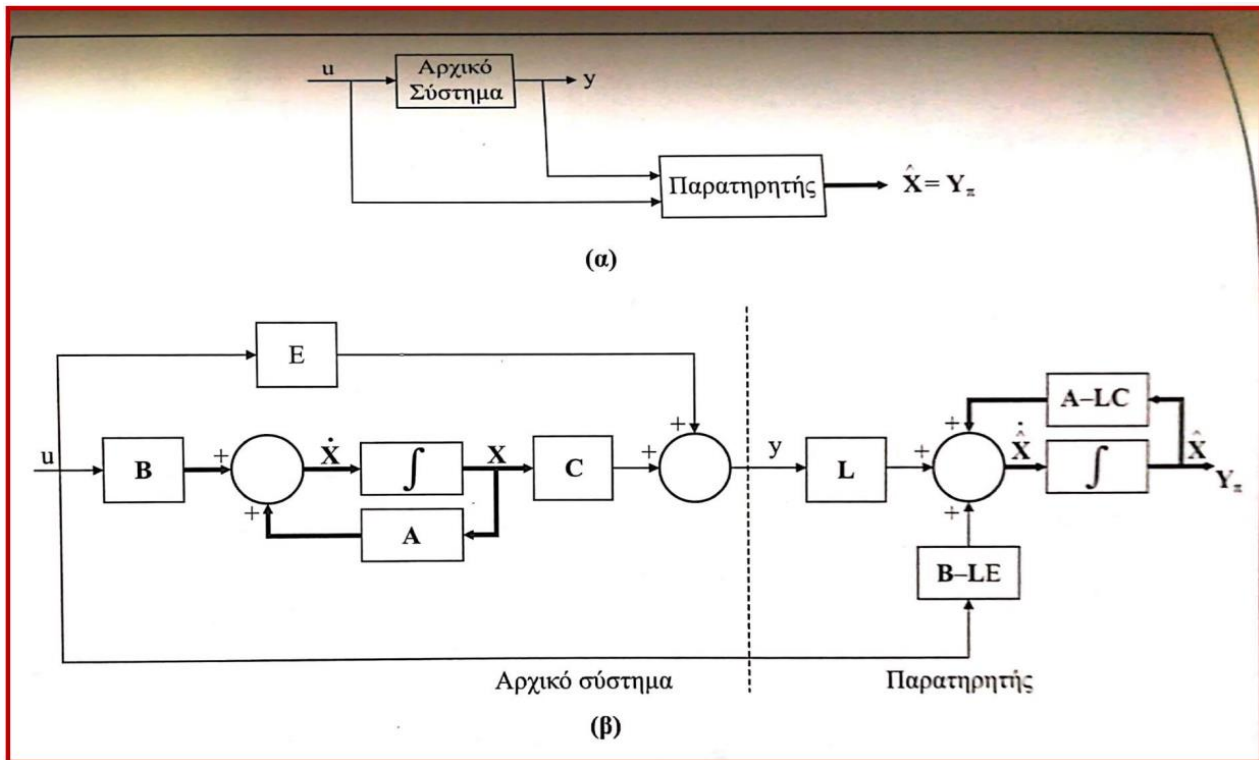
$$\frac{d\hat{X}}{dt} = (A - LC)\hat{X} + LCX + Bu$$

$$Y_{\pi} = \hat{X}$$

Σύνθετο επαυξημένο σύστημα:

$$\frac{d}{dt} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u(t)$$

$$\begin{bmatrix} y \\ Y_{\pi} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} u(t)$$



Εικόνα 7.2, Διάγραμμα βαθμίδων σύνθετου-επαυξημένου συστήματος (όπου $E=D$)

Παράδειγμα 7.1.1: Προσδιορισμός ανύσματος L (παρατηρητής)

Να προσδιορισθεί το άνυσμα L , τέτοιο ώστε οι ιδιοτιμές του πίνακα $A-LC$ να είναι $-5+2i$, $-5-2i$. Στη συνέχεια χρησιμοποιήστε το για την εκτίμηση καταστάσεων και σφάλματος.

$$\dot{x}(t) = \begin{bmatrix} 3 & 1 \\ 4 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(t)$$

$$y = [0 \quad 1]x(t) + [0]u(t)$$

Επίλυση: Ξεκινώντας και πάλι με τις εντολές `clc`, `clear` και `syms`. Ορίζουμε το μοντέλο του χώρου κατάστασης με τους πίνακες A, B, C, D καθώς και τις αρχικές συνθήκες x_0 .

```
clc
syms s
A=[3 1;4 0];
B=[1;-1];
C=[0 1];
D=[0];
x0=[0.5;0];
```

Έπειτα ορίζουμε τους επιθυμητούς πόλους για τον παρατηρητή μας. Το πρόβλημα ανάγεται στην ισοδύναμη μορφή επίλυσης γραμμικής ανάδρασης ($A-BK$), όπου πλέον ισοδύναμα μπορεί να μετασχηματιστεί ως $(A-LC) == (A^T - C^T L^T)$ χρησιμοποιώντας την εντολή `place`.

```
p1=-5+2i;
```

```
p2=-5-2i;
p=[p1 p2];
L=place(A',C', p);
L=L'
```

Στη συνέχεια, επιλύουμε το σφάλμα ω με τη χρήση του πίνακα μετάβασης.

$$\omega(t) = X(t) - \hat{X}(t)$$

$$\dot{\omega}(t) = (A - LC)\omega(t)$$

```
size_A=rank(A);
I=eye(size_A);
omega_0=[0.5;0];
%Error Omega
omega_s=((inv(s*I-(A-L*C)))*omega_0);
omega_t=ilaplace(omega_s)
```

Στο σημείο αυτό προσδιορίζουμε το επαυξημένο σύστημα μας.

$$\frac{d}{dt} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u(t)$$

$$\begin{bmatrix} y \\ y_\pi \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} u(t)$$

```
%The system with the observations
A_new=[A zeros(size(A));L*C A-L*C]
B_new=[B;B]
C_new=[C zeros(size(C));zeros(size(A)) eye(size(A))]
D_new=[D;zeros(size(B))]
```

Τέλος, επιλύουμε το πρόβλημα μας χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace και αναπαριστούμε γραφικά τα αποτελέσματά μας.

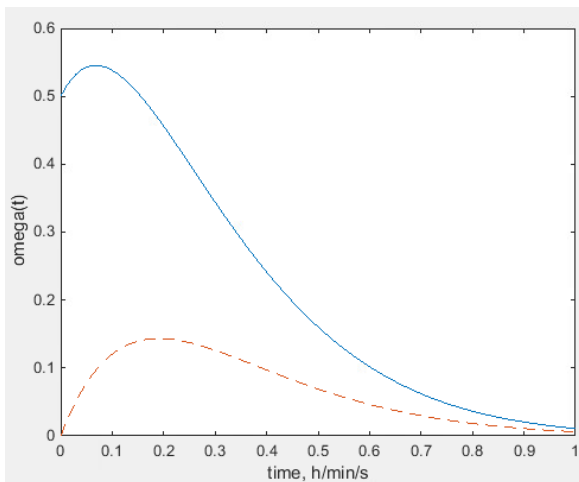
```
I_new=eye(size(A_new));
x0=[0.5;0;0;0];
u_s=0/s;
x_s=(inv(s*I_new-A_new)*x0)+(inv(s*I_new-A_new)*B_new*u_s);
y_s=C_new*inv(s*I_new-A_new)*x0+[C_new*inv(s*I_new-A_new)*B_new+D_new]*u_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)
Total_Time=1;
t=0:0.01:Total_Time;
omega_t=subs(omega_t,t);
omega_t=double(omega_t);
```

```

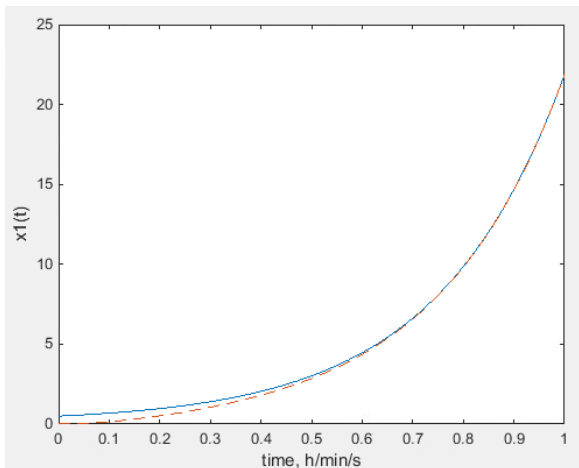
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,omega_t(1,:),t,omega_t(2,:), '--');xlabel('time,
h/min/s');ylabel('omega(t)')
figure(2);plot(t,x_t(1,:),t,x_t(3,:), '--');xlabel('time,
h/min/s');ylabel('x1(t)')
figure(3);plot(t,x_t(2,:),t,x_t(4,:), '--');xlabel('time,
h/min/s');ylabel('x2(t)')
figure(4);plot(t,y_t(1,:),t,y_t(2,:), '--',t,y_t(3,:), 'o');xlabel('time,
h/min/s');ylabel('y(t)')

```

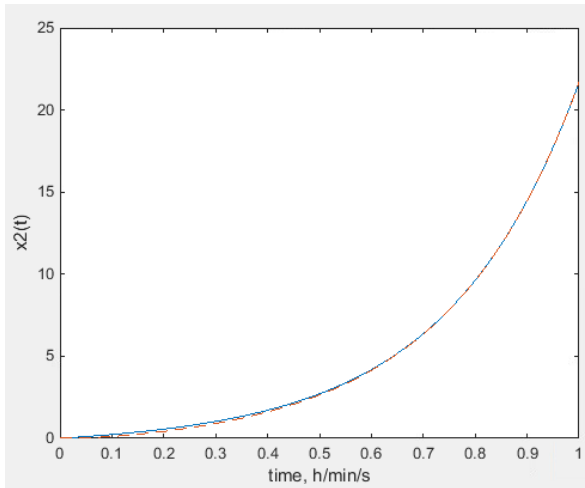
Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω διαγράμματα:



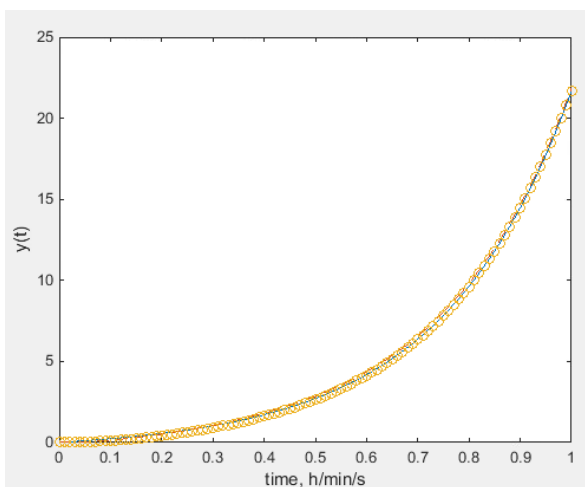
Διάγραμμα σφάλματων ω_1 , ω_2 για τις μεταβλητές κατάστασης x_1 , x_2 .



Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_1



Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_2



Διάγραμμα $y(t)$, $Y_{\pi}(t)$ κλειστού βρόχου

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [7.1/1](#).

Παράδειγμα 7.1.2: Πολλαπλά ερωτήματα χωρίς διαταραχή

Έστω το σύστημα:

$$\dot{x}(t) = \begin{bmatrix} 3 & 1 \\ 4 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(t)$$

$$y = [0 \quad 1]x(t) + [0]u(t)$$

$$x(0) = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

Να απαντηθούν τα παρακάτω ερωτήματα για το παραπάνω σύστημα:

A) Είναι το σύστημα ανοικτού βρόχου ελέγξιμο, παρατηρήσιμο και ευσταθές?

B) Να βρεθεί ο πίνακας μεταβατικής κατάστασης $\Phi(t)$ ή αλλιώς ο εκθετικός πίνακας e^{At} (μπορείτε να χρησιμοποιήσετε όποια μέθοδο θα θέλατε)

Γ) Να σχεδιάσετε έναν κατάλληλο ελεγκτή γραμμικής ανάδρασης που να μεταφέρει τους πόλους του κλειστού βρόχου σε $\theta_1 = -1+i$, $\theta_2 = -1-i$

Δ) Με βάση τον ελεγκτή γραμμικής ανάδρασης να δώσετε το σύστημα κλειστού βρόχου υπό μορφή μοντέλου χώρου κατάστασης.

Ε) Με βάση τον ελεγκτή γραμμικής ανάδρασης να επιλύσετε (υπολογιστικά...) το σύστημα κλειστού βρόχου ως προς x_1 , x_2 , u , y . Θεωρήστε $v=0$ (setpoint)

ΣΤ) Έστω τώρα, ότι κάποιες μεταβλητές κατάστασης δεν δύνανται να μετρηθούν και πρέπει να εκτιμηθούν. Να σχεδιάσετε έναν παρατηρητή (με βάση το σύστημα ανοικτού βρόχου) που να καταλήγει σε ιδιοτιμές $-5+2i$ και $-5-2i$.

Ζ) Να δώσετε το επαυξημένο σύστημα πινάκων που να συμπεριλαμβάνει το σύστημα κλειστού βρόχου αλλά και τον παρατηρητή.

Η) Με βάση τον ελεγκτή γραμμικής ανάδρασης και τον παρατηρητή σας να επιλύσετε (υπολογιστικά...) το σύστημα κλειστού βρόχου ως προς x_1 , x_2 , \hat{x}_1 , \hat{x}_2 , u , y . Θεωρήστε $v=0$ (setpoint).

Επίλυση ερωτήματος Α:

Ξεκινάμε με τις γνωστές εντολές `clc`, `clear` και `syms`. Ορίζουμε τους πίνακες A, B, C και D του συστήματος καθώς και τις αρχικές συνθήκες x_0 . Ελέγχουμε αρχικά εάν το σύστημα ανοικτού βρόχου είναι ευσταθές και στην συνέχεια αν είναι παρατηρήσιμο.

```
clc
Clear
syms s
A=[3 1;4 0];
B=[1;-1];
C=[0 1];
D=[0];
x0=[0.5;1];
%STABILITY OF THE OPEN LOOP SYSTEM
L=eig(A)
[k1,k2]=size(L);
for i=1:1:k1
    if real(L(i))<0
        disp('the pole is stable')
    elseif real(L(i))==0
        disp('the pole is marginally stable')
    else
        disp('the pole is unstable')
    end
end
%CONTROLLABILITY/OBSERVABILITY OF THE OPEN LOOP SYSTEM
Contr=ctrb(A,B)
Rank_A=rank(A);
Rank_Contr=rank(Contr);
```

```

if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
end
Observ=obsv(A,C)
Rank_Observ=rank(Observ);
if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
end

```

Επίλυση ερωτημάτων Β, Γ, Δ και Ε:

Υπολογίζουμε τον πίνακα $\Phi(t)$ και ορίζουμε τους επιθυμητούς πόλους. Επιπλέον, προσδιορίζουμε τους πίνακες κλειστού βρόχου και έπειτα επιλύουμε το σύστημα κλειστού βρόχου με γραμμική ανάδραση κατάστασης, δεδομένου ότι $v=0$ (set point), χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace. Παράλληλα δηλώνουμε τον χρόνο επίλυσης και με τη χρήση των γνωστών εντολών *subs* και *double*, θα αναπαριστήσουμε γραφικά τα αποτελέσματά μας.

```

%TRANSITION MATRIX
syms t
F_t=expm(A*t)
%STATE FEEDBACK CONTROL
p1=-1+1j;
p2=-1-1j;
p=[p1 p2];

K=place(A,B,p)

A_new=A-B*K
C_new=C-D*K
B_new=B
D_new=D

x0=[0.5;1];

%CLOSED LOOP WITH STATE FEEDBACK
size_A_new=rank(A_new);

```

```

I=eye(size_A_new);
n_Set_point=0/s;

x_s=(inv(s*I-A_new)*(x0+B_new*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*n_Set_point))+D_new*n_Set_point;
u_s=n_Set_point-K*x_s;

x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);

Total_Time=10;
t=0:0.1:Total_Time;

x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t, '--');xlabel('time, h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t);xlabel('time, h/min/s');ylabel('u(t)')

```

Επίλυση ερωτημάτων ΣΤ, Ζ και Η:

Ορίζουμε τους επιθυμητούς πόλους του παρατηρητή μας και επιλύουμε το σφάλμα ω με χρήση του πίνακα μετάβασης. Έπειτα προσδιορίζουμε το επαυξημένο σύστημά μας και επιλύουμε το πρόβλημά μας χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace και αναπαριστούμε γραφικά τα αποτελέσματά μας.

```

%STATE OBSERVER
op1=-5+2i;
op2=-5-2i;
op=[op1 op2];
L=place(A',C',op);
L=L'

```

```

size_A=rank(A);
I=eye(size_A);
omega_0=[0.0;0];
%Error OMEGA FOR THE OBSERVATION
omega_s=((inv(s*I-(A-L*C)))*omega_0);
omega_t=ilaplace(omega_s);
A_new=[A -B*K;L*C A-L*C-B*K]
B_new=[B;B]
C_new=[C -D*K;zeros(size(A)) eye(size(A)) ]
D_new=[D;zeros(size(B)) ]
I_new=eye(size(A_new));
x0=[0.5;1;0;0];
x_s=(inv(s*I_new-A_new)*(x0+B_new*n_Set_point));
y_s=(C_new)*(inv(s*I_new-A_new)*(x0+B_new*n_Set_point))+D_new*n_Set_point;
u1_s=n_Set_point-K*x_s(1:2,:);
u2_s=n_Set_point-K*x_s(3:4,:);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u1_t=ilaplace(u1_s);
u2_t=ilaplace(u2_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
omega_t=subs(omega_t,t);
omega_t=double(omega_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u1_t=subs(u1_t,t);
u1_t=double(u1_t);
u2_t=subs(u2_t,t);
u2_t=double(u2_t);
figure(4);plot(t,omega_t(1,:),t,omega_t(2,),'--');xlabel('time, h/min/s');ylabel('omega(t)')
figure(5);plot(t,x_t(1,:),t,x_t(3,),'--');xlabel('time, h/min/s');ylabel('x1(t)')

```



```
figure(6);plot(t,x_t(2,:),t,x_t(4,:), '--');xlabel('time, h/min/s');ylabel('x2(t)')
figure(7);plot(t,y_t(1,:),t,y_t(2,:), '--',t,y_t(3,:), 'o');xlabel('time, h/min/s');ylabel('y(t)')
figure(8);plot(t,u1_t,t,u2_t, '--');xlabel('time, h/min/s');ylabel('u(t)')
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω αποτελέσματα:

```
Command Window

L =

     4
    -1

the pole is unstable
the pole is stable
the system is controllable
the system is observable

F_t =

[ exp(-t)/5 + (4*exp(4*t))/5, exp(4*t)/5 - exp(-t)/5]
[ (4*exp(4*t))/5 - (4*exp(-t))/5, (4*exp(-t))/5 + exp(4*t)/5]

K =

    6.8333    1.8333

A_new =

   -3.8333   -0.8333
   10.8333    1.8333

C_new =

     0     1

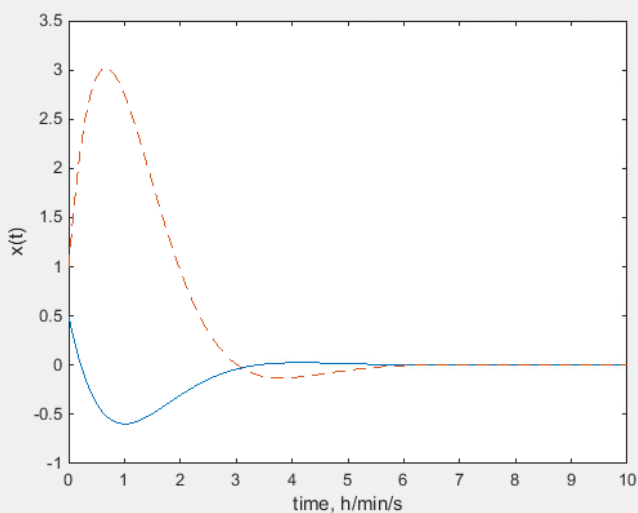
B_new =

     1
    -1

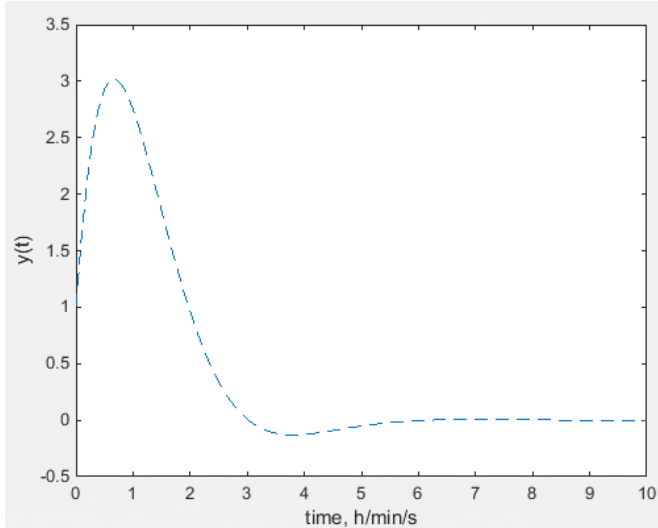
D_new =

     0
```

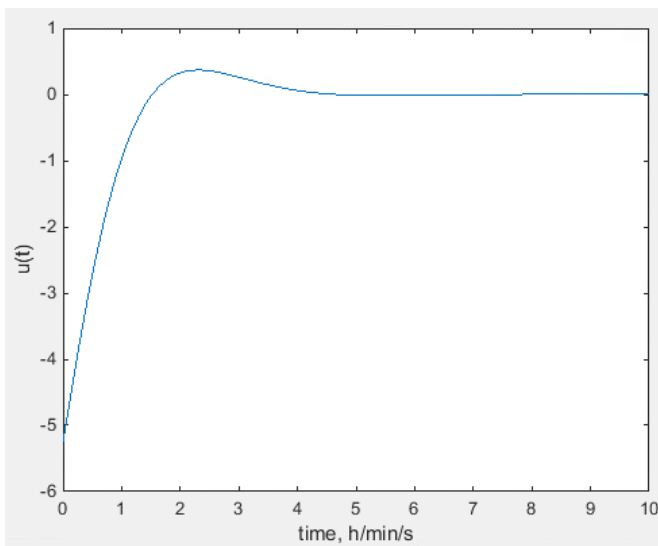
Απαντήσεις ερωτημάτων Α, Β, Γ και Δ



Ερώτημα Ε: Χρονικές αποκρίσεις $x_1(t)$, $x_2(t)$ για το σύστημα κλειστού βρόχου



Ερώτημα Ε: Χρονική απόκριση $y(t)$ για το σύστημα κλειστού βρόχου (Setpoint = 0)



Ερώτημα Ε: Διάγραμμα σήματος εισόδου $u(t)$

Command Window

```

L =

    18
    13

A_new =

    3.0000    1.0000   -6.8333   -1.8333
    4.0000     0      6.8333    1.8333
     0    18.0000   -3.8333   -18.8333
     0    13.0000   10.8333   -11.1667

B_new =

     1
    -1
     1
    -1

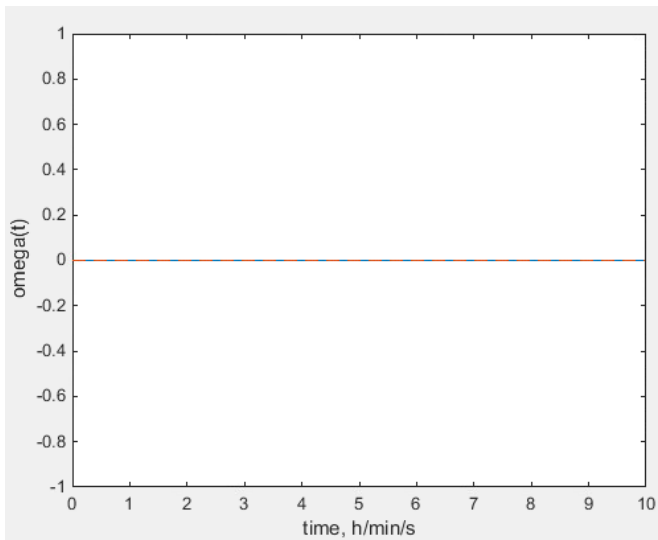
C_new =

     0     1     0     0
     0     0     1     0
     0     0     0     1

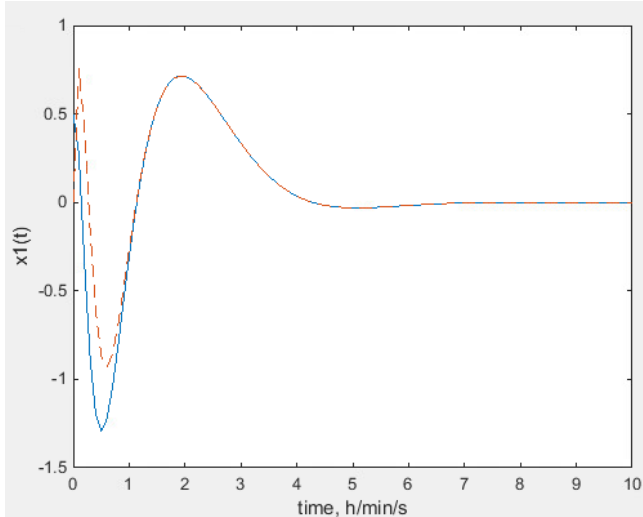
D_new =

```

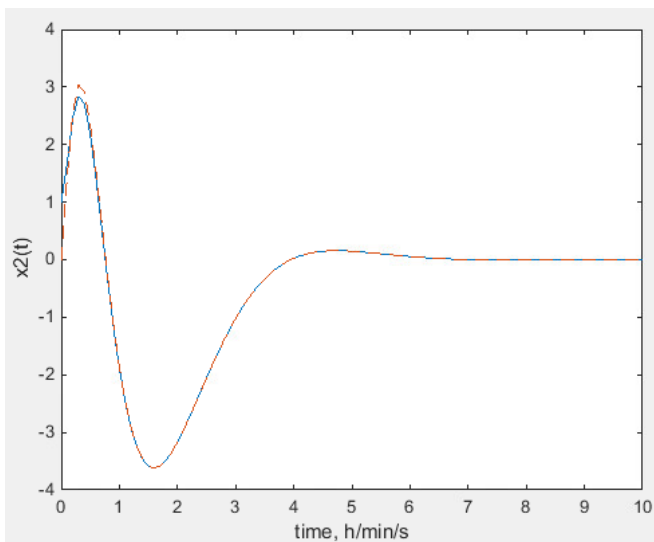
Ερωτήματα ΣΤ και Ζ



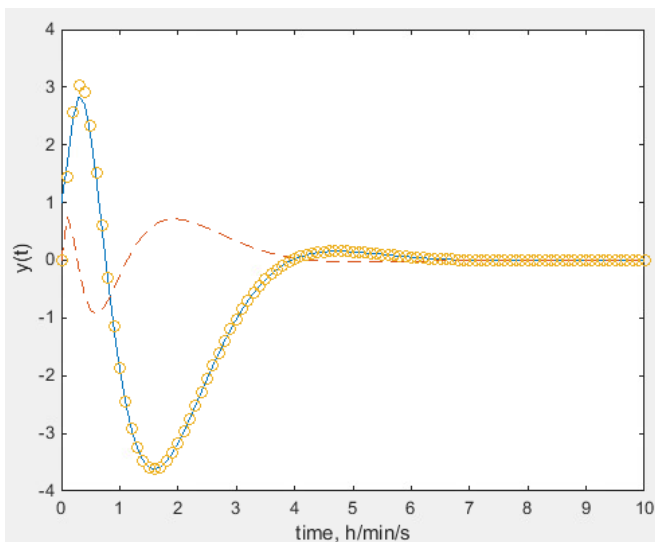
Ερώτημα Η: Διάγραμμα σφάλματος ω_1 , ω_2 για τις μεταβλητές κατάστασης x_1 , x_2 . (Μη-δενικό σφάλμα)



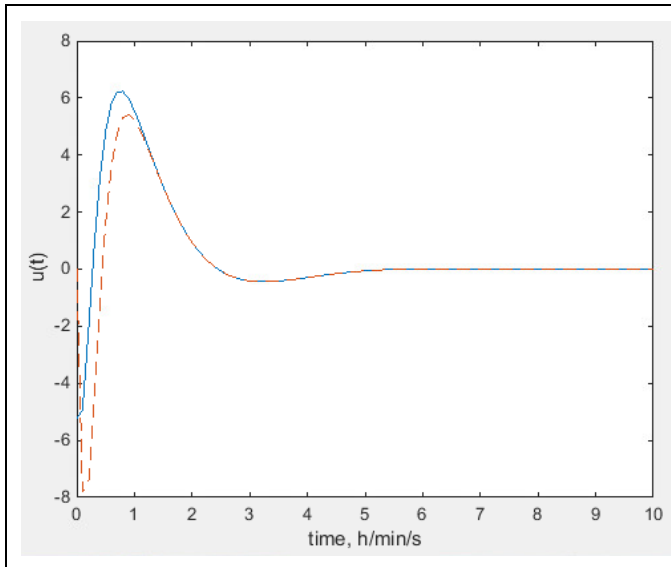
Ερώτημα Η: Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_1



Ερώτημα Η: Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_2



Ερώτημα Η: Διάγραμμα $y(t)$, $Y_{\pi}(t)$ κλειστού βρόχου



Διάγραμμα σημάτων εισόδου $u(t)$, με παρατηρητή (διακεκομμένες) και χωρίς

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [7.1/2](#).

Παράδειγμα 7.1.3: Πολλαπλά ερωτήματα με διαταραχή

Έστω το σύστημα με διαταραχή:

$$\dot{x}(t) = \begin{bmatrix} -2 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 3 \end{bmatrix} u(t) + \begin{bmatrix} 4 \\ 0 \end{bmatrix} d(t)$$

$$y = [1 \quad 0]x(t) + [0]u(t)$$

$$x(0) = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

Να απαντηθούν τα παρακάτω ερωτήματα για το παραπάνω σύστημα:

Α) Είναι το σύστημα ανοικτού βρόχου ελέγξιμο, παρατηρήσιμο και ευσταθές?

Β) Να βρεθεί ο πίνακας μεταβατικής κατάστασης $\Phi(t)$ ή αλλιώς ο εκθετικός πίνακας e^{At} (μπορείτε να χρησιμοποιήσετε όποια μέθοδο θα θέλατε)

Γ) Να σχεδιάσετε έναν κατάλληλο ελεγκτή γραμμικής ανάδρασης που να μεταφέρει τους πόλους του κλειστού βρόχου σε $\theta_1 = -5$, $\theta_2 = -6$

Δ) Με βάση τον ελεγκτή γραμμικής ανάδρασης να δώσετε το σύστημα κλειστού βρόχου υπό μορφή μοντέλου χώρου κατάστασης.

Ε) Με βάση τον ελεγκτή γραμμικής ανάδρασης να επιλύσετε (υπολογιστικά...) το σύστημα κλειστού βρόχου ως προς x_1 , x_2 , u , y . Θεωρήστε $v=0$ (setpoint), $d=(1/s)*\exp(-5*s)$.

ΣΤ) Έστω τώρα, ότι κάποιες μεταβλητές κατάστασης δεν δύναται να μετρηθούν και πρέπει να εκτιμηθούν. Να σχεδιάσετε έναν παρατηρητή (με βάση το σύστημα ανοικτού βρόχου) που να καταλήγει σε ιδιοτιμές -3 και -4 .

Ζ) Να δώσετε το επαυξημένο σύστημα πινάκων που να συμπεριλαμβάνει το σύστημα κλειστού βρόχου αλλά και τον παρατηρητή.

Η) Με βάση τον ελεγκτή γραμμικής ανάδρασης και τον παρατηρητή σας να επιλύσετε (υπολογιστικά...) το σύστημα κλειστού βρόχου ως προς x_1 , x_2 , \hat{x}_1 , \hat{x}_2 , u , y . Θεωρήστε $v=0$.

Επίλυση ερωτήματος A:

Ξεκινάμε με τις γνωστές εντολές `clc`, `clear` και `syms`. Ορίζουμε τους πίνακες A,B,C, D και E του συστήματος καθώς και τις αρχικές συνθήκες x_0 . Ελέγχουμε αρχικά εάν το σύστημα ανοιχτού βρόχου είναι ευσταθές και στην συνέχεια αν είναι παρατηρήσιμο.

```
clc
clear
syms s
A=[-2 4;0 -1];
B=[0;3];
C=[1 0];
D=[0];
E=[0;0];
x0=[0.5;1];
%STABILITY OF THE OPEN LOOP SYSTEM
L=eig(A)
[k1,k2]=size(L);
for i=1:1:k1
    if real(L(i))<0
        disp('the pole is stable')
    elseif real(L(i))==0
        disp('the pole is marginally stable')
    else
        disp('the pole is unstable')
    end
end
%CONTROLLABILITY/OBSERVABILITY OF THE OPEN LOOP SYSTEM
Contr=ctrb(A,B)
Rank_A=rank(A);
Rank_Contr=rank(Contr);
if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
end
Observ=obsv(A,C)
Rank_Observ=rank(Observ);
if Rank_A==Rank_Observ
```

```

        disp('the system is observable')
else
        disp('the system is unobservable')
end

```

Επίλυση ερωτημάτων Β, Γ, Δ και Ε:

Υπολογίζουμε τον πίνακα $\Phi(t)$ και ορίζουμε τους επιθυμητούς πόλους. Επιπλέον, προσδιορίζουμε τους πίνακες κλειστού βρόχου και έπειτα επιλύουμε το σύστημα κλειστού βρόχου με γραμμική ανάδραση κατάστασης, δεδομένου ότι $v=0$ (set point) και $d=(1/s)*\exp(-5*s)$, χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace. Παράλληλα δηλώνουμε τον χρόνο επίλυσης και με τη χρήση των γνωστών εντολών *subs* και *double*, θα αναπαριστήσουμε γραφικά τα αποτελέσματά μας.

```

%TRANSITION MATRIX
syms t
F_t=expm(A*t)
%STATE FEEDBACK CONTROL
p1=-5;
p2=-6;
p=[p1 p2];
K=place(A,B,p)
A_new=A-B*K
C_new=C-D*K
B_new=B
D_new=D
E_new=E
x0=[0.5;1];
%CLOSED LOOP WITH STATE FEEDBACK
size_A_new=rank(A_new);
I=eye(size_A_new);
n_Set_point=0/s;
dist=(1/s)*exp(-5*s);
x_s=(inv(s*I-A_new)*(x0+B_new*n_Set_point+E_new*dist));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*n_Set_point+E_new*dist))+D_new*n_Set_point;
u_s=n_Set_point-K*x_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;

```

```

t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t, '--');xlabel('time, h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t);xlabel('time, h/min/s');ylabel('u(t)')

```

Επίλυση ερωτημάτων ΣΤ, Ζ και Η:

Ορίζουμε τους επιθυμητούς πόλους του παρατηρητή μας και επιλύουμε το σφάλμα ω με χρήση του πίνακα μετάβασης. Έπειτα προσδιορίζουμε το επαυξημένο σύστημά μας και επιλύουμε το πρόβλημά μας χρησιμοποιώντας αντίστροφο μετασχηματισμό Laplace και αναπαριστούμε γραφικά τα αποτελέσματά μας.

```

%STATE OBSERVER
op1=-3;
op2=-4;
op=[op1 op2];
L=place(A',C',op);
L=L'
size_A=rank(A);
I=eye(size_A);
omega_0=[0;0];
%Error OMEGA FOR THE OBSERVATION
omega_s=((inv(s*I-(A-L*C)))*omega_0);
omega_t=ilaplace(omega_s);
A_new=[A -B*K;L*C A-L*C-B*K]
B_new=[B;B]
C_new=[C -D*K;zeros(size(A)) eye(size(A))]
D_new=[D;0;0]
E_new=[E;zeros(size(B))]
n_Set_point=0/s;
dist=(1/s)*exp(-5*s);
I_new=eye(size(A_new));
x0=[0.5;1;0;0];

```



```

x_s=(inv(s*I_new-A_new)*(x0+B_new*n_Set_point+E_new*dist));
y_s=(C_new)*(inv(s*I_new-A_new)*(x0+B_new*n_Set_point+E_new*dist))+D_new*n_Set_point;
u1_s=n_Set_point-K*x_s(1:2,:);
u2_s=n_Set_point-K*x_s(3:4,:);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u1_t=ilaplace(u1_s);
u2_t=ilaplace(u2_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
omega_t=subs(omega_t,t);
omega_t=double(omega_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u1_t=subs(u1_t,t);
u1_t=double(u1_t);
u2_t=subs(u2_t,t);
u2_t=double(u2_t);

figure(4);plot(t,omega_t(1,:),t,omega_t(2,),'--');xlabel('time, h/min/s');ylabel('omega(t)')
figure(5);plot(t,x_t(1,:),t,x_t(3,),'--');xlabel('time, h/min/s');ylabel('x1(t)')
figure(6);plot(t,x_t(2,:),t,x_t(4,),'--');xlabel('time, h/min/s');ylabel('x2(t)')
figure(7);plot(t,y_t(1,:),t,y_t(2,),'--',t,y_t(3,),'o');xlabel('time, h/min/s');ylabel('y(t)')
figure(8);plot(t,u1_t,t,u2_t,'--');xlabel('time, h/min/s');ylabel('u(t)')

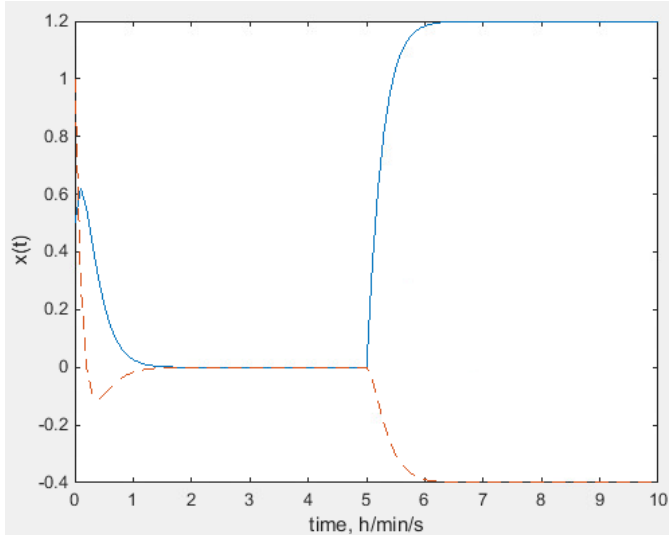
```

Μετά την εκτέλεση του κώδικα προκύπτουν τα παρακάτω αποτελέσματα:

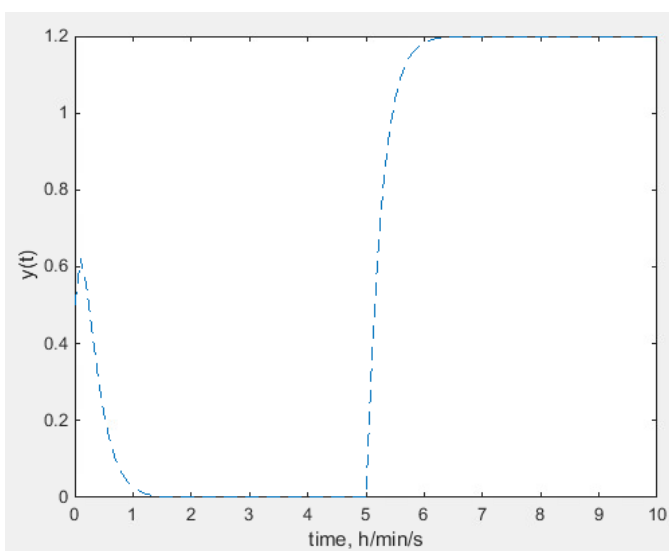
Command Window

```
L =  
  
    -2  
    -1  
  
the pole is stable  
the pole is stable  
the system is controllable  
the system is observable  
  
F_t =  
  
[ exp(-2*t), 4*exp(-t) - 4*exp(-2*t)]  
[      0,      exp(-t)]  
  
K =  
  
    1.0000    2.6667  
  
A_new =  
  
    -2.0000    4.0000  
    -3.0000   -9.0000  
  
C_new =  
  
    1    0  
  
B_new =  
  
    0  
    3  
  
D_new =  
  
    0  
  
E_new =  
  
    4  
    0
```

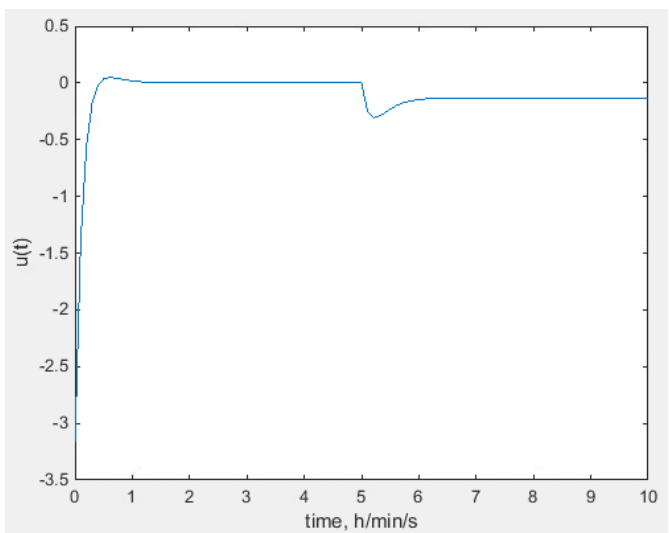
Απαντήσεις ερωτημάτων Α, Β, Γ και Δ



Ερώτημα Ε: Χρονικές αποκρίσεις $x_1(t)$, $x_2(t)$ για το σύστημα κλειστού βρόχου



Ερώτημα Ε: Χρονική απόκριση $y(t)$ για το σύστημα κλειστού βρόχου (Setpoint = 0)



Ερώτημα Ε: Διάγραμμα σήματος εισόδου $u(t)$

Command Window

L =

4.0000
1.5000

A_new =

-2.0000	4.0000	0	0
0	-1.0000	-3.0000	-8.0000
4.0000	0	-6.0000	4.0000
1.5000	0	-4.5000	-9.0000

B_new =

0
3
0
3

C_new =

1	0	0	0
0	0	1	0
0	0	0	1

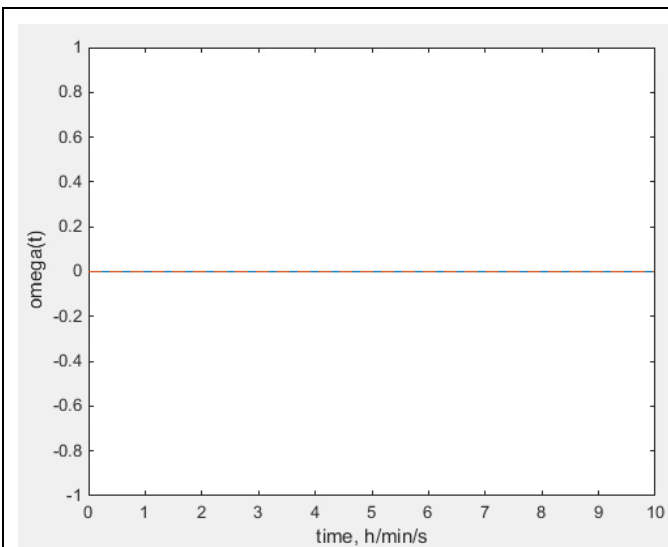
D_new =

0
0
0

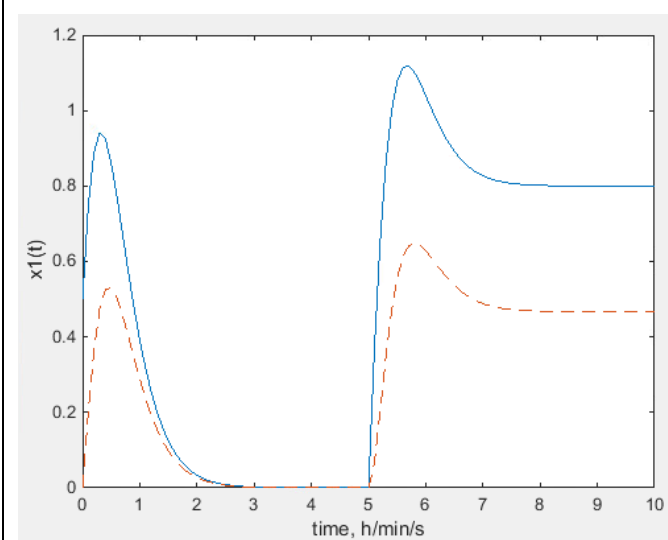
E_new =

4
0
0
0

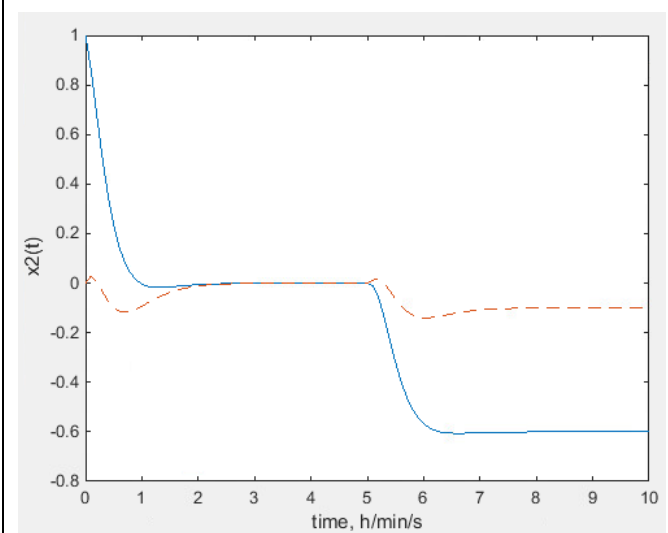
Ερωτήματα ΣΤ και Ζ



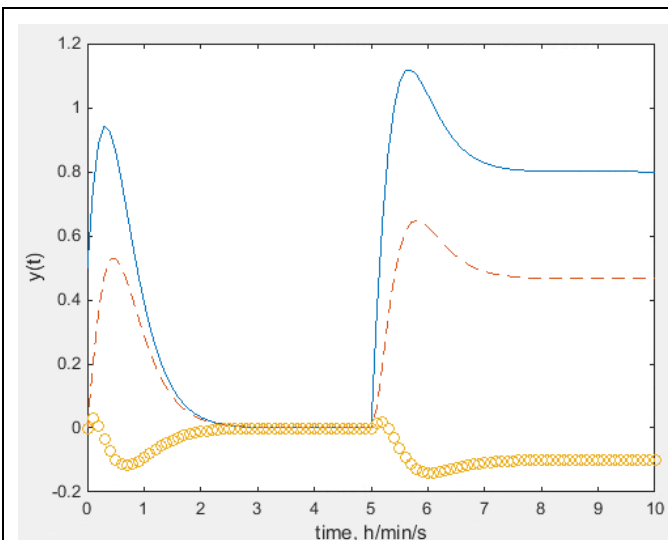
Ερώτημα Η: Διάγραμμα σφάλματων ω_1 , ω_2 για τις μεταβλητές κατάστασης x_1 , x_2 . (Μη-δενικό σφάλμα)



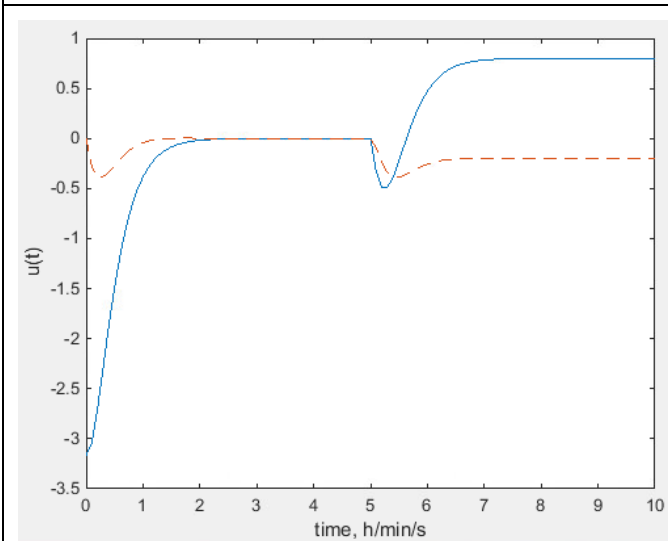
Ερώτημα Η: Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_1



Ερώτημα Η: Διάγραμμα παρατηρητή (διακεκομμένες) και χρονικής μεταβλητής x_2



Ερώτημα Η: Διάγραμμα $y(t)$, $Y_{\pi}(t)$ κλειστού βρόχου



Διάγραμμα σημάτων εισόδου $u(t)$, με παρατηρητή (διακεκομμένες) και χωρίς

Ολοκληρωμένος ο κώδικας του παραδείγματος βρίσκεται στο Παράρτημα [7.1/3](#).

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η διπλωματική αυτή επιδίωξε να αναπτύξει ολοκληρωμένο εκπαιδευτικό υλικό ειδικά σχεδιασμένο για την θεωρία Μοντέρνου Ελέγχου με έμφαση στην ενίσχυση των θεωρητικών εννοιών μέσω της πρακτικής εφαρμογής χρησιμοποιώντας το MATLAB. Τα παρακάτω βασικά σημεία συνοψίζουν τις συνεισφορές και τα ευρήματα κάθε κεφαλαίου:

- Ξεκινώντας από την εισαγωγή, στόχος ήταν η εξοικείωση των αναγνωστών με το MATLAB, παρουσιάζοντας τον τρόπο δημιουργίας αρχείων script, ώστε να τους παρέχει σωστή καθοδήγηση για τη διασφάλιση ορθής εκτέλεσης του κώδικα μέσω κατάλληλης ονοματοδοσίας αρχείων και διαχείρισης διαδρομής προορισμού αυτών.
- Το πρώτο κεφάλαιο παρουσίασε εκτενώς τον σχεδιασμό των ελεγκτών ανάδρασης με βάση το πρότυπο μοντέλο της συνάρτησης μεταφοράς διεργασίας $G_p(s)$ σε συστήματα ελάχιστης και μη-ελάχιστης φάσης $[G_p(s)$ και $G_p+(s)]$., εξασφαλίζοντας την ύπαρξη ευστάθειας κλειστού βρόχου, μηδενικού σφάλματος και την επιθυμητή δυναμική απόδοση. Συγκεκριμένα παρουσιάστηκαν παραδείγματα σχεδιασμού ελεγκτών 1^{ης}, 2^{ης} και ν^{ης} τάξης, με αρνητικό και θετικό μηδενικό καθώς και με ύπαρξη νεκρού χρόνου. Αποτέλεσμα αυτών ήταν η αναπαράσταση των σημάτων εισόδου/εξόδου και του σφάλματος σε διαγραμματική μορφή. Κατά την ανάλυση των παραδειγμάτων σχεδίασης ελεγκτή ανάδρασης, χρησιμοποιήθηκε η μεθοδολογία Morari και Zafiriou.
- Το δεύτερο κεφάλαιο αντιμετώπισε ειδικές περιπτώσεις των συστημάτων ελέγχου, όπως τα Συστήματα Ελέγχου Πρόδρασης, Πρόδρασης/Ανάδρασης και Συστοιχίας. Παρέχονται παραδείγματα σχεδιασμού ελεγκτών πρόδρασης με τη χρήση συναρτήσεων μεταφοράς και γίνεται γραφική αναπαράσταση των σημάτων εισόδου, εξόδου και του σφάλματος σε σχέση με το χρόνο. Επιπλέον, αναλύονται παραδείγματα σχεδιασμού ελεγκτών πρόδρασης/ανάδρασης, όπου συνδυάζονται οι δύο τεχνικές ελέγχου. Τέλος, παρουσιάζεται η τεχνική του ελέγχου συστοιχίας, ιδανική για περιπτώσεις με πολλαπλές διαταραχές και γίνεται σύγκριση με τα συστήματα ελέγχου ανάδρασης.
- Το τρίτο κεφάλαιο αποσκοπεί στην εφαρμογή μοντέλων χώρου κατάστασης και γραμμικοποίησης συστημάτων με τη χρήση του MATLAB. Συγκεκριμένα, εφαρμόζεται η διαδικασία της γραμμικοποίησης σε συστήματα πολλαπλών εισόδων και εξόδων με μη-γραμμικές διαφορικές εξισώσεις στον χώρο κατάστασης. Παρουσιάζονται και αναλύονται τρία παραδείγματα για την καλύτερη κατανόηση και εξάσκηση, στη κατασκευή και επίλυση γραμμικών και μη-γραμμικών συναρτήσεων. Τα δύο πρώτα αφορούν τον ίδιο χώρο κατάστασης, δύο δοχεία σε σειρά. Εξετάζεται η μεταβολή της στάθμης του νερού αυτών, συναρτήσει του χρόνου, καθώς και πως επηρεάζονται τα αποτελέσματα, αν το σημείο ισορροπίας δεν είναι μηδενικό. Στο τρίτο παράδειγμα ο χώρος κατάστασης είναι διαφορετικός και αφορά την ανάρτηση ενός βαρέως οχήματος. Παρόλα αυτά ο τρόπος επίλυσης είναι όμοιος με τα προηγούμενα.
- Στο Κεφάλαιο 4, εξετάζουμε την επίλυση γραμμικοποιημένων συστημάτων χρησιμοποιώντας τρεις διαφορετικές μεθόδους. Αρχικά, «Επίλυση στο Χρόνο», χρησιμοποιώντας τη μέθοδο Caley-Hamilton για την εύρεση του εκθετικού πίνακα. Έπειτα, «Επίλυση στο πεδίο Laplace» όπου χρησιμοποιήθηκαν οι μετασχηματισμοί Laplace και «Επίλυση με χρήση Ιδιοτιμών/Ιδιοδιανυσμάτων». Τέλος, παρουσιάζονται παραδείγματα όπου εξετάζονται οι αποκρίσεις με την παρουσία ή μη διαταραχών. Τα περιλαμβάνουν τόσο ομογενείς όσο και μη-ομογενείς διαφορικές εξισώσεις.

- Το πέμπτο κεφάλαιο εστιάζει στις ιδιότητες των πολυμεταβλητών συστημάτων, χρησιμοποιώντας έννοιες όπως οι ιδιοτιμές, τα ιδιοδιανύσματα και γραμμικοί μετασχηματισμοί. Αναλύεται η ευστάθεια των συστημάτων, οι πόλοι και τα μηδενικά τους, καθώς και οι γραμμικοί μετασχηματισμοί. Χρησιμοποιείται επίσης ο υπολογισμός του πίνακα Rosebrock, όπως παρουσιάστηκε στη θεωρία, για την εύρεση των πόλων και των μηδενικών ενός συστήματος.
- Το έκτο κεφάλαιο προσφέρει μια εμβάθυνση στον σχεδιασμό ελεγκτών και τη γραμμική ανάδραση καταστάσεων. Μέσα από παραδείγματα και θεωρητική ανάλυση, εξετάζονται οι δυνατότητες της γραμμικής ανάδρασης καταστάσεων να μετατρέψει ασταθή συστήματα σε ευσταθή. Μετά την προσδιορισμό του ανύσματος κατάστασης K , επιλύονται τα συστήματα κλειστού βρόχου, επιβεβαιώνοντας τη σωστή τοποθέτηση των πόλων. Τέλος, παρουσιάζονται παραδείγματα προσδιορισμού του ανύσματος κατάστασης K με τη χρήση LQR ελεγκτή και της εξίσωσης Riccati, παρέχοντας έναν επιπλέον τρόπο σχεδιασμού ελεγκτών.
- Στο έβδομο και τελευταίο κεφάλαιο παρουσιάζεται η έννοια του παρατηρητή και επιλύονται σχετικά παραδείγματα σχεδιασμού του. Συγκεκριμένα, αναλύονται τρία παραδείγματα, από τα οποία αναφέρονται στην προσδιορισμό του ανύσματος L για τον παρατηρητή, στην επίλυση πολλαπλών ερωτημάτων που αφορούν συστήματα χωρίς διαταραχή και την επίλυση πολλαπλών ερωτημάτων για συστήματα με διαταραχή. Στόχος των πολλαπλών ερωτημάτων, ήταν να παρέχεται μια ολοκληρωμένη εξάσκηση, καλύπτοντας θέματα που αναφέρθηκαν σε προηγούμενα κεφάλαια, όπως ο προσδιορισμός ενός συστήματος σαν ελέγξιμο, παρατηρήσιμο ή ευσταθές.

ΠΑΡΑΡΤΗΜΑ

ΚΕΦΑΛΑΙΟ 1:

1.1/1:

```
Clc
clear
syms s
K=5;t=2;lamda=1;
Gp=K/(t*s+1);
Gc=(t/(K*lamda))*(1+(1/(t*s)));
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

1.1/2:

```
Clc
clear
syms s
%1os tropos
% K=5;t=2;z=3/4;lamda=1;r=2;
% Gp=K/((t^2)*(s^2)+2*z*t*s+1);
% Gp_plus=1;
% Gp_minus=Gp;
% Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
% Gd=1;
% 2os tropos
K=5;t=2;z=3/4;lamda=1;
Gp=K/((t^2)*(s^2)+2*z*t*s+1);
Gc=(z*t/(K*lamda))*(1+(1/(2*z*t*s)))+(t/(2*z))*s*(1/((lamda/2)*s+1));
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
```

```
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

1.1/3:

```
Clc
clear
syms s
K=5;ta=3;t1=2;t2=5;r=1;lamda=1;
Gp=K*(ta*s+1)/((t1*s+1)*(t2*s+1));
Gp_plus=1;
Gp_minus=Gp;
Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

1.1/4:

```
clc
clear
syms s
K=5;ta=3;t1=2;t2=5;r=1;lamda=1;
Gp=K*(-ta*s+1)/((t1*s+1)*(t2*s+1));
Gp_plus=(1-ta*s)/(1+ta*s);
Gp_minus=K*(ta*s+1)/((t1*s+1)*(t2*s+1));
Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

1.1/5:

```
Clc
clear
syms s
K=5;t=3;theta=2;r=1;lamda=1;
Gp=(K/(t*s+1))*(1-theta*0.5*s)/(1+theta*0.5*s);
Gp_plus=(1-theta*0.5*s)/(1+theta*0.5*s);
Gp_minus=K/(t*s+1);
Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus));
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

1.1/6:

```
Clc
clear
syms s
K=5;t1=2;t2=3;t3=5;r=3;lamda=1;
Gp=K/((t1*s+1)*(t2*s+1)*(t3*s+1));
Gp_plus=1;
Gp_minus=Gp;
Gc=(1/Gp_minus)*(1/(((lamda*s+1)^r)-Gp_plus))
Gd=1;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gc/(1+Gp*Gc))*Ysp+(Gd/(1+Gp*Gc))*D
Error=Ysp-Y;
U=Gc*Error;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

ΚΕΦΑΛΑΙΟ 2:

2.1/1:

```
Clc
clear
syms s
K=5;t=2;lamda=1;
Gp=K/(t*s+1);
Gd=1/(t*s+1);
Gsp=(t*s+1)/(K*(lamda*s+1));
Gff=1/K;
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=(Gp*Gsp)*Ysp+(Gd-Gp*Gff)*D
U=Gsp*Ysp-Gff*D;
Error=Ysp-Y;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
xlswrite('y_signal.xlsx',[t' y' ysp'])
```

2.2/1:

```
Clc
clear
syms s
%SXEDIASMOS ELEGKTI PRODRASIS/ANADRASIS GIA SYSTIMA 2is TAXIS
Gp=(1-s)/((s+2)*(s+3));
Gd=1/(s+2);
lamda1=0.1;lamda2=0.1;
Gc=((s+2)*(s+3))/((2*s)*(lamda1*s+1));
Gff=((1/3)*(s+3))/(lamda2*s+1);
Ysp=(1/s)*(1+3*exp(-8*s));
D=(5/s)*exp(-3*s);
Y=((Gc*Gp)/(1+Gc*Gp))*Ysp+((Gd-Gp*Gff)/(1+Gc*Gp))*D
Error=Ysp-Y;
U=Gc*Error-Gff*D;
y=ilaplace(Y);
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
xlswrite('y_signal.xlsx',[t' y' ysp'])
```


2.3/1.1:

```
Clc
clear
syms s
Kc=1;
Gp1=(5*s+1)/((2*s+1)*(4*s+1));
Gd1=Gp1;
Gp2=1/(0.2*s^2+0.5*s+1);
Gd2=Gp2;
Gol=Kc*Gp1*Gp2;
[num, den]=numden(Gol)
num=sym2poly(num);
den=sym2poly(den);
Gs=tf(num,den)
figure(1);rlocus(Gs)
```

2.3/1.2:

% Slave Controller	% Master Controller
<pre>Clc clear syms s Kc=1; Gp2=1/(0.2*s^2+0.5*s+1); Gol=Kc*Gp2; [num, den]=numden(Gol) num=sym2poly(num); den=sym2poly(den); Gs=tf(num,den) figure(1);rlocus(Gs)</pre>	<pre>clc clear syms s Kc=1; Kc2=6; Gc2=Kc2; Gp1=(5*s+1)/((2*s+1)*(4*s+1)); Gd1=Gp1; Gp2=1/(0.2*s^2+0.5*s+1); Gd2=Gp2; Gol=Kc*Gp1*(Gc2*Gp2)/(1+Gc2*Gp2); [num, den]=numden(Gol) num=sym2poly(num); den=sym2poly(den); Gs=tf(num,den) figure(1);rlocus(Gs)</pre>

2.3/1.3:

```
Clc
clear
syms s
Gp1=(5*s+1)/((2*s+1)*(4*s+1));
Gd1=Gp1;
Gp2=1/(0.2*s^2+0.5*s+1);
Gd2=Gp2;
Kc=3.22; tI=1.24; tD=0.3;
Gc=Kc*(1+(1/(tI*s))+tD*s);
Ysp=(1/s);
D1=(1/s)*exp(-14*s);
D2=(1/s)*exp(-8*s);
Y=((Gc*Gp1*Gp2)/(1+Gc*Gp1*Gp2))*Ysp+...
    (Gd1/(1+Gc*Gp1*Gp2))*D1+...
    (Gd2*Gp1/(1+Gc*Gp1*Gp2))*D2
Error=Ysp-Y;
U=Gd2*D2+Gp2*Gc*Error;
y=ilaplace(Y)
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

2.3/1.4:

```
Clc
clear
syms s
Gp1=(5*s+1)/((2*s+1)*(4*s+1));
Gd1=Gp1;
Gp2=1/(0.2*s^2+0.5*s+1);
Gd2=Gp2;
Kc2=6;
Gc2=Kc2;
Kc1=2.85; tI1=0.52;tD=0.13;
Gc1=Kc1*(1+(1/(tI1*s))+tD*s);
Ysp=(1/s);
D1=(1/s)*exp(-14*s);
D2=(1/s)*exp(-8*s);
Y=((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2)))/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2))))) *Ysp+...
    (Gd1/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2))))) *D1+...
    (Gd2*Gp1/(1+((Gc1*Gp1)*((Gc2*Gp2)/(1+Gc2*Gp2))))) *D2
Error=Ysp-Y;
U=((Gc2*Gp2)/(1+Gc2*Gp2))*Gc1*(Ysp-Y)+(Gd2/(1+((Gc2*Gp2))))*D2;
y=ilaplace(Y)
u=ilaplace(U);
error=ilaplace(Error);
Total_Time=30;
t=0:0.1:Total_Time;
y=subs(y,t);
y=double(y);
u=subs(u,t);
u=double(u);
error=subs(error,t);
error=double(error);
ysp1=ilaplace(Ysp);
ysp1=subs(ysp1,t);
ysp1=double(ysp1);
ysp=ysp1;
figure(1);plot(t,y,t,ysp,'--');xlabel('time, min');ylabel('y(t)')
```

```
figure(2);plot(t,u);xlabel('time, min');ylabel('u(t)')  
figure(3);plot(t,error);xlabel('time, min');ylabel('error(t)')
```

ΚΕΦΑΛΑΙΟ 3:

3.1/1:

```
function S = Tank_Height
Clc
clear
% Constant Parameters
A1=1;
A2=1;
k1=0.5;
k2=0.5;
F0=2;
n1=0.5;
n2=0.5;
h0(1)=0;
h0(2)=0;
function dx=f(t,x)
h_1= x(1.1);
h_2= x(2,1);
% Algebraic Equations
F1=k1*(h_1)^n1;
F2=k2*(h_2)^n2;
dx = zeros(2,1);
%doxeiol
dx(1)= (F0-F1)*(1/A1);
dx(2)= (F1-F2)*(1/A2);
End
% solving parameters
tspan=[0:1:150];
[tsol,xsol]= ode45(@f,tspan,h0);
figure(1);
plot(tsol,xsol(:,1),'--','MarkerSize',10);
hold on
plot(tsol,xsol(:,2));
hold off
xlabel('Time, min');
ylabel('Tank Height, m')
```

```

drawnow
xlswrite('height.xlsx', (tsol,xsol)
End

```

3.1/2:

```

function S = Tank_Height
Clc
clear
%Constant Parameters
A1=1;
A2=1;
k1=0.5;
k2=0.5;
F0=2;
n1=0.5;
n2=0.5;
h0(1)=0;
h0(2)=0;
function dx=f(t,x)
h_1= x(1,1);
h_2= x(2,1);
%Algebraic Equations
F1=k1*(h_1)^n1;
F2=k2*(h_2)^n2;
dx = zeros(2,1);
%doxeiol
dx(1)= (F0-F1)*(1/A1);
dx(2)= (F1-F2)*(1/A2);
End
% solving parameters
tspan=[0:1:150];
[tsol,xsol]= ode45(@f,tspan,h0);
figure(1);
plot(tsol,xsol(:,1),'--','MarkerSize',10);
hold on
plot(tsol,xsol(:,2));
hold off

```

```

xlabel('Time, min');
ylabel('Tank Height, m')
drawnow
xlswrite('height.xlsx',[tso1 xso1])
End

```

3.1/3:

```

function S = Car_Dispension
Clc
clear
%Constant Parameters
B1=1;
B2=1;
m1=15;
m2=5;
k1=1;
k2=1;
F=2;
x0(1)=0;
x0(2)=0;
x0(3)=0;
x0(4)=0;
function dx=f(t,x)
x_1 = x(1,1);
x_2 = x(2,1);
x_3 = x(3,1);
x_4 = x(4,1);
y1=x_1;
y2=x_3;
dx = zeros(4,1);
dx(1)= x_2;
dx(2)=(-B1/m1)*(x_2-x_4)-(k1/m1)*(x_1-x_3)+(1/m1)*F;
dx(3)=x_4;
dx(4)=(B1/m2)*(x_2-x_4)+(k1/m2)*(x_1-x_3)-(k2/m2)*x_3-(B2/m2)*x_4;
End
%solving parameters
tspan=[0:1:300];

```

```
[tsol,xsol]= ode45(@f,tspan,x0);  
figure(1);  
plot(tsol,xsol(:,1),'--','MarkerSize',10);  
hold on  
plot(tsol,xsol(:,3));  
hold off  
xlabel('Time, min');  
ylabel('Dispension, dm');  
drawnow  
xlswrite('Car_dispension.xlsx',[tsol xsol])  
End
```


ΚΕΦΑΛΑΙΟ 4:

4.1/1:

```
A=[0 1;-2 -3];  
L=eig(A)  
syms t  
exp_At=expm(A*t)
```

4.1/2:

```
Clc  
clear  
syms t  
A=[0 1; -2 -3];  
x0=[1;1];  
exp_At=expm(A*t);  
A=[0 1; -2 -3];  
exp_At=expm(A*t)  
Total_Time=10;  
t=0:0.1:Total_Time;  
x_t=exp_At*x0;  
x_t=subs(x_t,t);  
x_t=double(x_t);  
figure(1);  
plot(t,x_t(1,:),t,x_t(2,:), '--');  
xlabel('time, h/min/s');  
ylabel('x(t)')
```

4.1/3:

```
Clc  
clear  
syms t tau  
A=[0 1; -2 -3];  
B=[0;1];  
x0=[0;0];  
exp_At=expm(A*t);  
u_t=1;  
fun=(expm(A*(t-tau)))*B*u_t;  
x_t=exp_At*x0+int(fun,tau,0,t)
```

```

Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');ylabel('x(t)')

```

4.2/1:

```

clc
clear
syms s
A=[0 1; -2 -3];
size_A=rank(A);
x0=[1;1];
I=eye(size_A);
exp_At=inv(s*I-A);
exp_At=ilaplace(exp_At)
x_s=inv(s*I-A)*x0;
x_t=ilaplace(x_s)
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');
ylabel('x(t)')

```

4.2/2:

```

Clc
clear
syms s
A=[0 1; -2 -3];
B=[0;1];
x0=[0;0];
size_A=rank(A);

```

```

I=eye(size_A);
u_s=1/s;
x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
x_t=ilaplace(x_s)
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
figure(1);
plot(t,x_t(1,:),t,x_t(2,:), '--');
xlabel('time, h/min/s');
ylabel('x(t)')
xlabel('time, h/min/s');
ylabel('x(t)')

```

4.3/1:

```

Clc
clear
syms t
A=[-2 1;2 -3];
x0=[1;2];
[M,L]=eig(A);
rank_L=rank(L);
for i=1:rank_L
    exp_lamda_t(i,i)=[exp(t*L(i,i))];
End
x_t=M*exp_lamda_t*inv(M)*x0
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')

```

4.4/1:

```

Clc
clear
syms s

```

```

A=[-1 -2; 1 -3];
B=[1;0];
x0=[0;0];
C=[1 0;0 1];
D=0;
size_A=rank(A);
I=eye(size_A);
u_s=1/s;
x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s;
Gp=[C*inv(s*I-A)*B+D]
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:),t,y_t(2,),'--');xlabel('time,
h/min/s');ylabel('y(t)')

```

4.4/2:

```

clc
clear
syms s
A=[-1 -2; 1 -3];
B=[3 1;5 -1];
x0=[0;0];
C=[1 0;0 1];
D=0;
E=[1 1;3 -1];
size_A=rank(A);
I=eye(size_A);
u_s=[(1/s)*(1+3*exp(-10*s)-1*exp(-35*s));(1/s)*(3-2*exp(-20*s))];
d_s=[(2/s)*exp(-15*s);(1/s)*(2-1*exp(-45*s)+4*exp(-60*s))];

```

```

x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s)+(inv(s*I-A)*E*d_s);
y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s+[C*inv(s*I-A)*E]*d_s;
Gp=[C*inv(s*I-A)*B+D]
Gd=[C*inv(s*I-A)*B+E]
u_t=ilaplace(u_s);
d_t=ilaplace(d_s);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)
Total_Time=100;
t=0:0.5:Total_Time;
u_t=subs(u_t,t);
u_t=double(u_t);
d_t=subs(d_t,t);
d_t=double(d_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:),t,y_t(2,:), '--');xlabel('time,
h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t(1,:),t,u_t(2,:), '--');xlabel('time,
h/min/s');ylabel('u(t)')
figure(4);plot(t,d_t(1,:),t,d_t(2,:), '--');xlabel('time,
h/min/s');ylabel('d(t)')

```

ΚΕΦΑΛΑΙΟ 5:

5.1/1:

```
Clc
clear
A1=[2 1;-2 3];
A2=[-2 1;-2 -3];
A3=[-2 -1 1;2 -3 -2;-1 -3 -4];
A4=[-2 1;2 -1];
A=A2;
L=eig(A)
[k1,k2]=size(L);
for i=1:1:k1
    if real(L(i))<0
        disp('the system is stable')
    elseif real(L(i))==0
        disp('the system is marginally stable')
    else
        disp('the system is unstable')
    End
End
syms t
x0=[1;1];
exp_At=expm(A*t);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=exp_At*x0;
x_t=subs(x_t,t);
x_t=double(x_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
```

5.2/1:

```
Clc
clear
syms s
A=[-1 -2; 1 -3];
B=[1;0];
```

```

C=[1 0;0 1];
D=[0;0];
size_A=rank(A);
I=eye(size_A);
P_rosenbrock=[s*I-A -B;C D];
Eigenvalues_A=eig(A);
System_Poles=Eigenvalues_A
[m_p,n_p]=size(P_rosenbrock);
if m_p>n_p
    Size_P=n_p;
else
    Size_P=m_p;
End
for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_1(i,j)=P_rosenbrock(i,j);
    end
End
Zeros=det(P_rosenbrock_1);
Zeros=sym2poly(Zeros);
Zeros=roots(Zeros)

```

5.3/1:

```

Clc
clear
syms s t
A=[-2 1; 2 -3];
B=[3;5];
C=[1 0;0 1];
D=[0;0];
[M, L]=eig(A);
A_transform=(inv(M)*A*M)
B_transform=(inv(M)*B)
C_transform=C*M
D_transform=D
na=size(A);
I=eye(na);

```

```

Gp=[C*(inv(s*I-A))*B+D];
Gp_transform=[C_transform*(inv(s*I-A))*B_transform+D_transform];
%1o mideniko
P_rosenbrock_main_1=[s*I-A -B;C(1,:) D(1)];
P_rosenbrock_transform_1=[s*I-A_transform -B_transform;C_transform(1,:)
D_transform(1)];

P_rosenbrock_1=P_rosenbrock_main_1;
% P_rosenbrock_1=P_rosenbrock_transform_1;
[m_p,n_p]=size(P_rosenbrock_1);
if m_p>n_p
    Size_P=n_p;
else
    Size_P=m_p;
End
for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_1(i,j)=P_rosenbrock_1(i,j);
    end
End
Zeros=det(P_rosenbrock_1);
Zeros=sym2poly(Zeros);
Zeros_1=roots(Zeros)
% 2o mideniko
P_rosenbrock_main_2=[s*I-A -B;C(2,:) D(2)];
P_rosenbrock_transform_2=[s*I-A_transform -B_transform;C_transform(2,:)
D_transform(2)];
P_rosenbrock_2=P_rosenbrock_main_2;
% P_rosenbrock_2=P_rosenbrock_transform_2;
[m_p,n_p]=size(P_rosenbrock_2);
if m_p>n_p
    Size_P=n_p;
else
    Size_P=m_p;
End
for i=1:1:Size_P
    for j=1:1:Size_P
        P_rosenbrock_2(i,j)=P_rosenbrock_2(i,j);
    end
End

```



```

        end
End
Zeros=det(P_rosenbrock_2);
Zeros=sym2poly(Zeros);
Zeros_2=roots(Zeros)
u_s=1/s;
x0=[0;0];
x_s=(inv(s*I-A)*x0)+(inv(s*I-A)*B*u_s);
x_s_transform=(inv(s*I-A_transform)*x0)+(inv(s*I-A_transform)*B_transform*u_s);
y_s=C*inv(s*I-A)*x0+[C*inv(s*I-A)*B+D]*u_s;
y_s_transform=C_transform*inv(s*I-A_transform)*x0+[C_transform*inv(s*I-A_transform)*B_transform+D_transform]*u_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
x_t_transform=ilaplace(x_s_transform);
y_t_transform=ilaplace(y_s_transform);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
x_t_transform=subs(x_t_transform,t);
x_t_transform=double(x_t_transform);
y_t_transform=subs(y_t_transform,t);
y_t_transform=double(y_t_transform);
figure(1);plot(t,y_t(1,:),t,y_t_transform(1,:), 'o');xlabel('time, h/min/s');ylabel('y1(t)')
figure(2);plot(t,y_t(2,:),t,y_t_transform(2,:), 'o');xlabel('time, h/min/s');ylabel('y1(t)')

```

5.4/1:

```

Clc
clear
A=[0 1 0 0;3 0 0 2;0 0 0 4;0 -2 1 0];
B=[0 0;1 0;0 0;0 1];
C=[0 1 1 1];
D=[0;0];
Contr=ctrb(A,B);

```

```

Rank_A=rank(A);
Rank_Contr=rank(Contr);

if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
End
Observ=obsv(A,C);
Rank_Observ=rank(Observ);

if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
End

```

5.5/1:

```

Clc
clear
num=[3 2 -23 -22];
den=[1 0 -7 -6];
G=tf(num,den)
[A, B, C, D]=tf2ss(num,den)
A1=[-2 1; 2 -3]
B1=[0;1]
C1=[0 1]
D1=0
[num1,den1]=ss2tf(A1,B1,C1,D1);
G=tf(num1,den1)
Contr=ctrb(A,B);
Rank_A=rank(A);
Rank_Contr=rank(Contr);
if Rank_A==Rank_Contr
    disp('the system is controllable')
else

```

```
        disp('the system is uncontrollable')
End
Observ=obsv(A,C);
Rank_Observ=rank(Observ);
if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
End
```

ΚΕΦΑΛΑΙΟ 6:

6.1/1:

```
Clc
clear
% % PART I of SLIDES
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
L_open=eig(A);
[k1,k2]=size(L_open);
for i=1:1:k1
    if real(L_open(i))<0
        disp('the open loop system is stable')
    elseif real(L_open(i))==0
        disp('the open loop system is marginally stable')
    else
        disp('the open loop system is unstable')
    End
End
% % Epithymittoi Poloi Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
K=place(A,B,P)
```

6.2/1:

```
Clc
clear
syms s
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
% % Epithymittoi Poloi Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
```

```

K=place(A,B,P)
x0=[1;1];
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;
n_Set_point=0/s;
L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
    if real(L_closed(i))<0
        disp('the closed loop system is stable')
    elseif real(L_closed(i))==0
        disp('the closed loop system is marginally stable')
    else
        disp('the closed loop system is unstable')
    End
End
%Epilysi Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);
x_s=(inv(s*I-A_new)*(x0+B*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B*n_Set_point))+D*n_Set_point;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t,'--');xlabel('time, h/min/s');ylabel('y(t)')
return

```

6.2/2:

```
Clc
clear
syms s
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
% Epithymitoti Poloti Kleistou Broxou
p1=-1; p2=-2;
P=[p1 p2];
K=place(A,B,P)
x0=[1;1];
% Anysma katastasis
K=place(A,B,P)
kcomp=5/(-2.4993);
x0=[1;1];
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;
n_Set_point=5/s;
L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
    if real(L_closed(i))<0
        disp('the closed loop system is stable')
    elseif real(L_closed(i))==0
        disp('the closed loop system is marginally stable')
    else
        disp('the closed loop system is unstable')
    End
End
%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);
```

```

x_s=(inv(s*I-A_new)*(x0+B_new*kcomp*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*kcomp*n_Set_point))+D_new*kcomp*n_Set_point;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t, '--');xlabel('time, h/min/s');ylabel('y(t)')
return

```

6.2/3:

```

clc
clear
syms s
A=[1 2;0 1];
B=[1;1];
C=[1 2];
D=0;
% Anysma katastasis
G = -B*B';
Q = C'*C;
[T,POLES,K] = care(A, B, Q, 1)
x0=[1;1];
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;
n_Set_point=0/s;
L_closed=eig(A_new);
[k1,k2]=size(L_closed);
for i=1:1:k1

```

```

if real(L_closed(i))<0
    disp('the closed loop system is stable')
elseif real(L_closed(i))==0
    disp('the closed loop system is marginally stable')
else
    disp('the closed loop system is unstable')
End
End
%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);
x_s=(inv(s*I-A_new)*(x0+B_new*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*n_Set_point))+D_new*n_Set_point;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t,'--');xlabel('time, h/min/s');ylabel('y(t)')
return

```

6.3/1:

```

Clc
clear
syms s
A=[-1 0;-3 -2];
B=[1 2;2 0];
C=[1 0;0 1];
size_C=rank(C);
D1=zeros(size_C);
D=D1;%(:,1)
% Epithymittoi Poloi Kleistou Broxou

```



```

p1=-4; p2=-5;
P=[p1 p2];
% Anysma katastasis
K=[2.364 -0.545; 2.364 -0.545];
% K=place(A,B,P)
x0=[1;1];
A_new=A-B*K;
C_new=C-D'*K;
B_new=B;
D_new=D';
n_Set_point=[0/s;0/s];
L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
if real(L_closed(i))<0
    disp('the closed loop system is stable')
elseif real(L_closed(i))==0
    disp('the closed loop system is marginally stable')
else
    disp('the closed loop system is unstable')
End
End
%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);
x_s=(inv(s*I-A_new)*(x0+B*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B*n_Set_point))+D*n_Set_point;
u_s=n_Set_point-K*x_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);

```

```

y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:));xlabel('time, h/min/s');ylabel('y1(t)')
figure(3);plot(t,y_t(2,:));xlabel('time, h/min/s');ylabel('y2(t)')
figure(4);plot(t,u_t(1,:),t,u_t(2,:), '--');xlabel('time,
h/min/s');ylabel('u(t)')
return

```

6.3/2:

```

Clc
Clear
syms s
A=[-1 0;-3 -2];
B=[1 2;2 0];
C=[1 0;0 1];
size_C=rank(C);
D1=zeros(size_C);
D=D1;%(:,1);
% Epithymitoi Poloi Kleistou Broxou
p1=-4; p2=-5;
P=[p1 p2];
% Anysma katastasis
% K=[2.366 -0.545; 2.366 -0.545];
% K=place(A,B,P)
% Anysma katastasis RICCATI
G = -B*B';
Q = C'*C;
size_A=rank(A);
R=eye(size_A);
S=zeros(size_A);
E=R;
[T,POLES,K] = care(A, B, Q, R, S, E, 10)
x0=[1;1];
A_new=A-B*K;

```

```

C_new=C-D'*K;
B_new=B;
D_new=D';
n_Set_point=[0/s;0/s];
L_closed=eig(A_new)
[k1,k2]=size(L_closed);
for i=1:1:k1
if real(L_closed(i))<0
    disp('the closed loop system is stable')
elseif real(L_closed(i))==0
    disp('the closed loop system is marginally stable')
else
    disp('the closed loop system is unstable')
End
End
%Epilysis Kleistou Broxou
size_A_new=rank(A_new);
I=eye(size_A_new);
x_s=(inv(s*I-A_new)*(x0+B*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B*n_Set_point))+D*n_Set_point;
u_s=n_Set_point-K*x_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t(1,:));xlabel('time, h/min/s');ylabel('y1(t)')
figure(3);plot(t,y_t(2,:));xlabel('time, h/min/s');ylabel('y2(t)')

```

```
figure(4);plot(t,u_t(1,:),t,u_t(2,:), '--');xlabel('time,  
h/min/s');ylabel('u(t)')  
return
```

ΚΕΦΑΛΑΙΟ 7:

7.1/1:

```
clc
clear
syms s
A=[3 1;4 0];
B=[1;-1];
C=[0 1];
D=[0];
x0=[0.5;0];
p1=-5+2i;
p2=-5-2i;
p=[p1 p2];
L=place(A',C', p);
L=L'
size_A=rank(A);
I=eye(size_A);
omega_0=[0.5;0];
%Error Omega
omega_s=((inv(s*I-(A-L*C)))*omega_0);
omega_t=ilaplace(omega_s)
%The system with the observations
A_new=[A zeros(size(A));L*C A-L*C]
B_new=[B;B]
C_new=[C zeros(size(C));zeros(size(A)) eye(size(A)) ]
D_new=[D;zeros(size(B)) ]
I_new=eye(size(A_new));
x0=[0.5;0;0;0];
u_s=0/s;
x_s=(inv(s*I_new-A_new)*x0)+(inv(s*I_new-A_new)*B_new*u_s);
y_s=C_new*inv(s*I_new-A_new)*x0+[C_new*inv(s*I_new-A_new)*B_new+D_new]*u_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s)
Total_Time=1;
t=0:0.01:Total_Time;
```

```

omega_t=subs(omega_t,t);
omega_t=double(omega_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
figure(1);plot(t,omega_t(1,:),t,omega_t(2,:), '--');xlabel('time,
h/min/s');ylabel('omega(t)')
figure(2);plot(t,x_t(1,:),t,x_t(3,:), '--');xlabel('time,
h/min/s');ylabel('x1(t)')
figure(3);plot(t,x_t(2,:),t,x_t(4,:), '--');xlabel('time,
h/min/s');ylabel('x2(t)')
figure(4);plot(t,y_t(1,:),t,y_t(2,:), '--',t,y_t(3,:), 'o');xlabel('time,
h/min/s');ylabel('y(t)')

```

7.1/2:

```

clc
clear
syms s
A=[3 1;4 0];
B=[1;-1];
C=[0 1];
D=[0];
x0=[0.5;1];
%STABILITY OF THE OPEN LOOP SYSTEM
L=eig(A)
[k1,k2]=size(L);
for i=1:1:k1
    if real(L(i))<0
        disp('the pole is stable')
    elseif real(L(i))==0
        disp('the pole is marginally stable')
    else
        disp('the pole is unstable')
    end
end
%CONTROLLABILITY/OBSERVABILITY OF THE OPEN LOOP SYSTEM
Contr=ctrb(A,B);
Rank_A=rank(A);
Rank_Contr=rank(Contr);

```

```

if Rank_A==Rank_Contr
    disp('the system is controllable')
else
    disp('the system is uncontrollable')
end
Observ=obsv(A,C);
Rank_Observ=rank(Observ);
if Rank_A==Rank_Observ
    disp('the system is observable')
else
    disp('the system is unobservable')
end
%TRANSITION MATRIX
syms t
F_t=expm(A*t)
%STATE FEEDBACK CONTROL
p1=-1+1j;
p2=-1-1j;
p=[p1 p2];
K=place(A,B,p)
A_new=A-B*K;
C_new=C-D*K;
B_new=B;
D_new=D;
x0=[0.5;1];
%CLOSED LOOP WITH STATE FEEDBACK
size_A_new=rank(A_new);
I=eye(size_A_new);
n_Set_point=0/s;
x_s=(inv(s*I-A_new)*(x0+B_new*n_Set_point));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*n_Set_point))+D_new*n_Set_point;
u_s=n_Set_point-K*x_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);

```

```

Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);

figure(1);plot(t,x_t(1,:),t,x_t(2,),'--');xlabel('time,
h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t,'--');xlabel('time, h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t);xlabel('time, h/min/s');ylabel('u(t)')

%STATE OBSERVER
op1=-5+2i;
op2=-5-2i;
op=[op1 op2];
L=place(A',C',op);
L=L'
size_A=rank(A);
I=eye(size_A);
omega_0=[0.0;0];
%Error OMEGA FOR THE OBSERVATION
omega_s=((inv(s*I-(A-L*C)))*omega_0)
omega_t=ilaplace(omega_s)
A_new=[A -B*K;L*C A-L*C-B*K]
B_new=[B;B]
C_new=[C -D*K;zeros(size(A)) eye(size(A)) ]
D_new=[D;zeros(size(B)) ]
I_new=eye(size(A_new));
x0=[0.5;1;0;0];
x_s=(inv(s*I_new-A_new)*(x0+B_new*n_Set_point));
y_s=(C_new)*(inv(s*I_new-A_new)*(x0+B_new*n_Set_point))+D_new*n_Set_point;
u1_s=n_Set_point-K*x_s(1:2,:);
u2_s=n_Set_point-K*x_s(3:4,:);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u1_t=ilaplace(u1_s);

```



```

u2_t=ilaplace(u2_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
omega_t=subs(omega_t,t);
omega_t=double(omega_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u1_t=subs(u1_t,t);
u1_t=double(u1_t);
u2_t=subs(u2_t,t);
u2_t=double(u2_t);

figure(4);plot(t,omega_t(1,:),t,omega_t(2,:), '--');xlabel('time, h/min/s');ylabel('omega(t)')
figure(5);plot(t,x_t(1,:),t,x_t(3,:), '--');xlabel('time, h/min/s');ylabel('x1(t)')
figure(6);plot(t,x_t(2,:),t,x_t(4,:), '--');xlabel('time, h/min/s');ylabel('x2(t)')
figure(7);plot(t,y_t(1,:),t,y_t(2,:), '--',t,y_t(3,:), 'o');xlabel('time, h/min/s');ylabel('y(t)')
figure(8);plot(t,u1_t,t,u2_t, '--');xlabel('time, h/min/s');ylabel('u(t)')

```

7.1/3:

```

clc
clear
syms s
A=[-2 4;0 -1];
B=[0;3];
C=[1 0];
D=[0];
E=[4;0];
x0=[0.5;1];
%STABILITY OF THE OPEN LOOP SYSTEM
L=eig(A)
[k1,k2]=size(L);
for i=1:1:k1
if real(L(i))<0

```

```

        disp('the pole is stable')
elseif real(L(i))==0
        disp('the pole is marginally stable')
else
        disp('the pole is unstable')
end
end
%CONTROLLABILITY/OBSERVABILITY OF THE OPEN LOOP SYSTEM
Contr=ctrb(A,B);
Rank_A=rank(A);
Rank_Contr=rank(Contr);
if Rank_A==Rank_Contr
        disp('the system is controllable')
else
        disp('the system is uncontrollable')
end
Observ=obsv(A,C);
Rank_Observ=rank(Observ);
if Rank_A==Rank_Observ
        disp('the system is observable')
else
        disp('the system is unobservable')
end
%TRANSITION MATRIX
syms t
F_t=expm(A*t)
%STATE FEEDBACK CONTROL
p1=-5;
p2=-6;
p=[p1 p2];
K=place(A,B,p)
A_new=A-B*K
C_new=C-D*K
B_new=B
D_new=D
E_new=E

```

```

x0=[0.5;1];
%CLOSED LOOP WITH STATE FEEDBACK
size_A_new=rank(A_new);
I=eye(size_A_new);
n_Set_point=0/s;
dist=(1/s)*exp(-5*s);
x_s=(inv(s*I-A_new)*(x0+B_new*n_Set_point+E_new*dist));
y_s=(C_new)*(inv(s*I-A_new)*(x0+B_new*n_Set_point+E_new*dist))+D_new*n_Set_point;
u_s=n_Set_point-K*x_s;
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u_t=ilaplace(u_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u_t=subs(u_t,t);
u_t=double(u_t);
figure(1);plot(t,x_t(1,:),t,x_t(2,:), '--');xlabel('time, h/min/s');ylabel('x(t)')
figure(2);plot(t,y_t, '--');xlabel('time, h/min/s');ylabel('y(t)')
figure(3);plot(t,u_t);xlabel('time, h/min/s');ylabel('u(t)')
%STATE OBSERVER
op1=-3;
op2=-4;
op=[op1 op2];
L=place(A',C',op);
L=L'
size_A=rank(A);
I=eye(size_A);
omega_0=[0;0];
%Error OMEGA FOR THE OBSERVATION
omega_s=((inv(s*I-(A-L*C)))*omega_0);
omega_t=ilaplace(omega_s);

```

```

A_new=[A -B*K;L*C A-L*C-B*K]
B_new=[B;B]
C_new=[C -D*K;zeros(size(A)) eye(size(A)) ]
D_new=[D;0;0]
E_new=[E;zeros(size(B)) ]
n_Set_point=0/s;
dist=(1/s)*exp(-5*s);
I_new=eye(size(A_new));
x0=[0.5;1;0;0];
x_s=(inv(s*I_new-A_new)*(x0+B_new*n_Set_point+E_new*dist));
y_s=(C_new)*(inv(s*I_new-A_new)*(x0+B_new*n_Set_point+E_new*dist))+D_new*n_Set_point;
u1_s=n_Set_point-K*x_s(1:2,:);
u2_s=n_Set_point-K*x_s(3:4,:);
x_t=ilaplace(x_s);
y_t=ilaplace(y_s);
u1_t=ilaplace(u1_s);
u2_t=ilaplace(u2_s);
set_point=ilaplace(n_Set_point);
Total_Time=10;
t=0:0.1:Total_Time;
omega_t=subs(omega_t,t);
omega_t=double(omega_t);
x_t=subs(x_t,t);
x_t=double(x_t);
y_t=subs(y_t,t);
y_t=double(y_t);
u1_t=subs(u1_t,t);
u1_t=double(u1_t);
u2_t=subs(u2_t,t);
u2_t=double(u2_t);
figure(4);plot(t,omega_t(1,:),t,omega_t(2,),'--');xlabel('time, h/min/s');ylabel('omega(t)')
figure(5);plot(t,x_t(1,:),t,x_t(3,),'--');xlabel('time, h/min/s');ylabel('x1(t)')
figure(6);plot(t,x_t(2,:),t,x_t(4,),'--');xlabel('time, h/min/s');ylabel('x2(t)')
figure(7);plot(t,y_t(1,:),t,y_t(2,),'--',t,y_t(3,),'o');xlabel('time, h/min/s');ylabel('y(t)')
figure(8);plot(t,u1_t,t,u2_t,'--');xlabel('time,

```

```
h/min/s');ylabel('u(t)')
```