ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
"ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ"

Υπολογισμός Αντισταθμιστή Ανοιχτού και Κλειστού Βρόγχου με τη Χρήση του GUI

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεωργία Γ. Πεχλιβανίδου

Επιβλέπων: Αντώνιος Ιωάννης Βαρδουλάκης
Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Μάρτιος 2010

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
" ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ "

Υπολογισμός Αντισταθμιστή Ανοιχτού και Κλειστού Βρόγχου με τη Χρήση του GUI

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεωργία Γ. Πεχλιβανίδου

Επιβλέπων: Αντώνιος Ιωάννης Βαρδουλάκης
Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την η Μαρτίου 2010.

…………………………	…………………………	…………………………
Α. Ι. Βαρδουλάκης	Ν. Καραμπετάκης	Ε. Αντωνίου
Καθηγητής Α.Π.Θ.	Αν. Καθηγητής Α.Π.Θ.	Επ. Καθηγητής Α.Τ.Ε.Ι. Θεσσαλονίκης

Θεσσαλονίκη, Μάρτιος 2010

…………………………………………..
Γεωργία Γ. Πεχλιβανίδου
Πτυχιούχος Μαθηματικός Α.Π.Θ.

ΠΕΡΙΛΗΨΗ

Η λειτουργία πολλών συστημάτων δεν είναι σταθερή. Το πρόβλημα αυτό αντιμετωπίζεται μέσω ενός νέου συστήματος, του αντισταθμιστή και της μοναδιαίας ανάδρασης. Το σύστημα αυτό είναι σύστημα κλειστού βρόχου, έχει εσωτερική ευστάθεια και οι έξοδοι του συστήματος έχουν επιθυμητή συμπεριφορά και δεν υπάρχουν ανεπιθύμητες αλληλεπιδράσεις μεταξύ των σημάτων εισόδων και των εξόδων.

Σκοπός της εργασίας είναι η κατασκευή δύο προγραμμάτων για τον υπολογισμό του αντισταθμιστή ενός συστήματος κλειστού βρόχου στο GUI του Matlab. Στο πρώτο πρόγραμμα είναι να υπολογίζεται η συνάρτηση μεταφοράς του ανοιχτού βρόχου συστήματος, το οποίο σύστημα δέχεται μία είσοδο, θεωρώντας ως γνωστούς τους πίνακες ενός συστήματος αυτομάτου ελέγχου οι οποίοι συνδέονται μέσω των εξισώσεων στο χώρο των καταστάσεων. Στη συνέχεια υπολογίζεται η συνάρτηση μεταφοράς κλειστού βρόχου, το οποίο επιτυγχάνεται με τη χρήση αντισταθμιστού. Προηγουμένος εισάγονται οι επιθυμητοί πόλοι.

Στο δεύτερο πρόγραμμα δίνεται η συνάρτηση μεταφοράς ανοιχτού βρόχου και υπολογίζεται η συνάρτηση μεταφοράς κλειστού βρόχου και του αντισταθμιστή.

Η παρούσα εργασία χωρίζεται σε δύο κεφάλαια. Στο πρώτο κεφάλαιο αναφέρατε η θεωρία την οποία κάνουν χρήση τα προγράμματα. Στο δεύτερο κεφάλαιο δίνονται τα προγράμματα και εξηγείται ο τρόπος με τον οποίο τρέχουν.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

GUI-Matlab, Αντισταθμιστής, Ανάδραση, Συνάρτηση μεταφοράς, Σταθεροποιήσημο σύστημα

ABSTRACT

The operation of multiple systems is not stable. This issue is treated using a new system consisting of a compensator and a unit feedback. The closed loop system is with inner stability, and the system outputs have desirable behaviour. No undesirable reactions/interrelations between input and output are observed.

The main objective is to develop two Matlab programs, using the GUI environment, able to estimate the compensator of a closed loop system. The transfer function of the open loop system is estimated in the first program named “compensatorabcd”. This system receives an input, assuming the matrices of an automated control system to be known. These matrices are connected using the functions in the device area. The transfer function is further estimated using a compensator. The desired poles are inserted a priori.

The transfer function of an open loop is inserted in the second program named “compensator”, aiming to estimate the transfer function of a closed loop and a compensator.

The current study consists of two chapters. The theory and literature review is mentioned in the first chapter. The programs and their manuals are presented in the second chapter.

KEY WORDS

GUI-Matlab, Compensator, Feedback, Transfer function, Stabilizable system

Ευχαριστίες προς

τον επιβλέποντα καθηγητή μου, κ. Αντώνη Βαρδουλάκη

την οικογένεια μου

και την φίλη μου Barbara!

Π Ι Ν Α Κ Α Σ   Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Ω Ν

ΠΕΡΙΛΗΨΗ..4

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ...4

ABSTRACT..5

KEY WORDS...5

ΠΡΟΛΟΓΟΣ..6

ΠΕΡΙΕΧΟΜΕΝΑ...7

Σελίδα

ΚΕΦΑΛΑΙΟ I...9

1.1 Εισαγωγή.. 9

1.1.1. Σήματα και Συστήματα...9

1.1.2. Η Μαθηματική Ερμηνεία του Συστήματος....................................11

1.1.3. Συνδεσμολογία Συστημάτων...18

1.2 Υπολογισμός Αντισταθμιστή...21

1.2.1. Βασική Θεωρία...21

1.2.2. Ο Αλγόριθμός Των Προγραμμάτων..26

ΚΕΦΑΛΑΙΟ II..28

2.1. Το GUI...28

2.2 Προγραμματισμός..30

2.2.1. Πρόγραμμα compensatorabcd...30

2.2.1.1. Εισαγωγή..30

2.2.1.2. Οι Συναρτήσεις του Προγράμματος compensatorabcd........32

2.2.1.3. Το Πρόγραμμα compensatorabcd…….................................38

2.2.1.4. Τρέξιμο Προγράμματος compensatorabcd...........................50

2.2.2. Πρόγραμμα compensator 52

2.2.2.1. Εισαγωγή..52

2.2.2.2 Οι Συναρτήσεις του Προγράμματος compensator.................54

2.2.2.3. Το Πρόγραμμα compensator...59

2.2.2.4. Τρέξιμο Πρόγραμμα compensator.....................................68

2.3. Εφαρμογή..69

Βιβλιογραφία / Εργασίες / Σημειώσεις...74

<u>ΚΕΦΑΛΑΙΟ Ι</u>

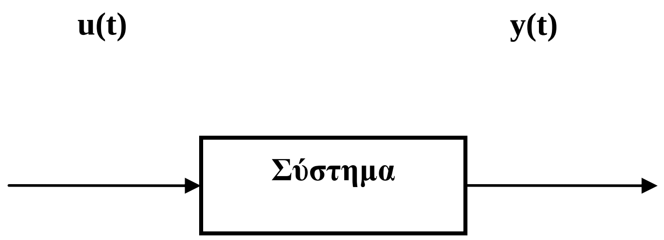
1.1. Εισαγωγή

1.1.1. <u>*Σήματα και Συστήματα*</u>

Οι λέξεις σήματα και συστήματα ακούγονται συχνά στη σημερινή εποχή. Έτσι θεωρούνται τα ηλεκτρικά σήματα και τα συστήματα αυτομάτου ελέγχου. Τα σήματα περιγράφονται από μία μαθηματική συνάρτηση η οποία αποτελείται από μία ή περισσότερες ανεξάρτητες μεταβλητές, μία από τις οποίες είναι υποχρεωτικά ο χρόνος, δηλαδή συμβολίζονται ως f(x,y,z,t,….w). Το σήμα περιέχει πληροφορίες για την χρονική εξέλιξη μίας ποσότητας η οποία περιγράφει ένα φαινόμενο ή μία διαδικασία.

Για το σύστημα δεν μπορεί να δοθεί ακριβής ορισμός . Περιγραφικά θα ήταν δυνατόν να διατυπωθεί ότι είναι μια διάταξη ή σύνολο ή συλλογή αντικειμένων συνδεμένων κατά τέτοιο τρόπο ώστε να αποτελούν μία ενότητα ή να δρουν ως ολοκληρωμένη ομάδα δηλαδή ότι είναι συνδεδεμένα έτσι ώστε να μπορούν να πετύχουν ένα συγκεκριμένο σκοπό.

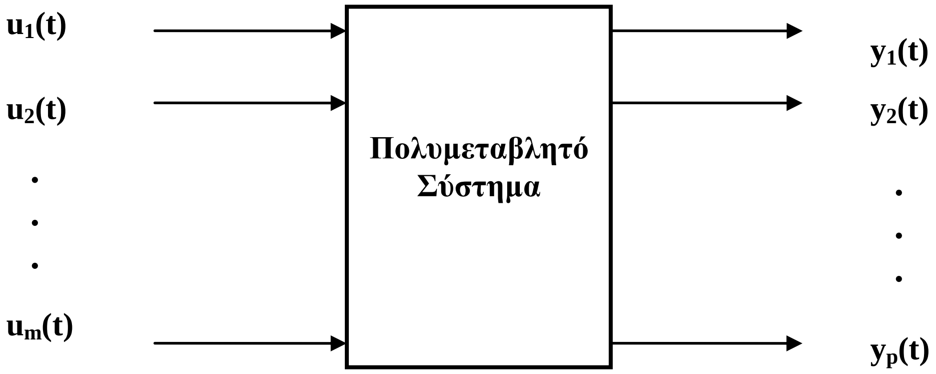
Στο Σχήμα 1 (Μπλοκ διάγραμμα) φαίνεται η σύνδεση των σημάτων με το σύστημα.



όπου u(t) είναι το σήμα της εισόδου του συστήματος και y(t) το σήμα που παράγει το σύστημα στην έξοδο το οποίο ονομάζεται πραγματική απόκριση y(t) του συστήματος ελέγχου.

Στα σήματα όταν εφαρμόζεται ηθελημένα στο σύστημα μπορεί να είναι είτε κινητήρια δύναμη, είτε διέγερση, είτε εντολή και προέρχεται από μία εξωτερική πηγή ενέργειας. Οι διεγέρσεις οι οποίες δεν δρουν ηθελημένα στο σύστημα ονομάζονται διαταραχές.

Εκτός των συστημάτων μιας εισόδου και μιας εξόδου υπάρχουν και τα συστήματα τα οποία δέχονται παραπάνω από μία εισόδους και εξόδους. Ο αριθμός των εισόδων δεν είναι απαραίτητο να είναι ίσος με τον αριθμό των εξόδων. Ένα τέτοιο σύστημα όπου είτε ο αριθμός των εισόδων είτε ο αριθμός των εξόδων είναι μεγαλύτερος της μονάδας ονομάζεται πολυμεταβλητό σύστημα και γραφικά περιγράφεται από το Σχήμα 2.



Σχήμα 2

Ένα σύστημα με m διεγέρσεις και p αποκρίσεις εκφράζεται μαθηματικός είναι ένας τελεστής F, ο οποίος ορίζεται (Fu)(t)=y(t)

$$\text{όπου } u(t)=\begin{bmatrix} u_1(t) \\ u_2(t) \\ \ldots\ldots \\ u_m(t) \end{bmatrix} \text{ και } y(t)=\begin{bmatrix} y_1(t) \\ y_2(t) \\ \ldots\ldots \\ y_p(t) \end{bmatrix}.$$

Μία διάταξη φυσικών στοιχείων συνδεδεμένων ή συσχετισμένων κατά τέτοιο τρόπο ώστε να κατευθύνουν, να ρυθμίζουν ή να σταθεροποιούν αυτό το ίδιο το σύστημα, ή κάποιο άλλο σύστημα ονομάζεται σύστημα αυτομάτου ελέγχου.

Ένα σύστημα ελέγχου χαρακτηρίζεται ανοικτού βρόχου όταν η δράση ελέγχου είναι ανεξάρτητη της εξόδου ενώ είναι κλειστού βρόχου όταν η δράση του εξαρτάται από την έξοδο. Τα συστήματα κλειστού βρόγχου ονομάζονται και ως συστήματα ανάδρασης. Ανάδραση είναι εκείνη η συγκεκριμένη ιδιότητα ενός συστήματος κλειστού βρόχου η οποία επιτρέπει τη σύγκριση της εξόδου του με την είσοδο στο σύστημα, έτσι ώστε η επιθυμητή δράση ελέγχου να αποτελεί συνάρτηση της εξόδου και της εισόδου του συστήματος. Πρέπει να σημειωθεί ότι η έξοδος μπορεί να προέρχεται από κάποια άλλη ελεγχόμενη μεταβλητή και η είσοδος εκτός από το σύστημα μπορεί να επιδρά και σε κάποιο υποσύστημα. Επίσης η ανάδραση είναι εκείνο το χαρακτηριστικό των συστημάτων ελέγχου κλειστού βρόγχου το οποίο τα κάνει να ξεχωρίζουν από τα συστήματα του ανοικτού βρόχου.

Τα συστήματα χωρίζονται σε δύο κατηγορίες:

- Στα συστήματα συνεχούς χρόνου τα οποία δέχονται ως είσοδο ένα ή περισσότερα σήματα συνεχούς χρόνου. Αυτά παριστάνονται από μία πραγματική συνάρτηση $x_i(t): R \to R$, όπου i=1,…m και το t ο χρόνος σε συνεχή μορφή.
- Και στα συστήματα διακριτού χρόνου τα οποία δέχονται ως είσοδο σήματα διακριτού χρόνου y[t] όπου το t παίρνει διακριτές τιμές , t=k*T
  όπου k=…-2, -1, 0, 1, 2, … και T>0 είναι η περίοδος δειγματοληψίας.

1.1.2. *Η Μαθηματική Ερμηνεία του Συστήματος*

Η μαθηματική περιγραφή ενός συστήματος μπορεί να παρασταθεί από n διαφορικές εξισώσεις πρώτης τάξης της μορφής:

$$\dot{x}_1(t) = f_1(x_1(t), x_2(t), \ldots., x_n(t), t; u_1(t), u_2(t), \ldots., u_m(t))$$

$$\dot{x}_2(t) = f_2(x_1(t), x_2(t), \ldots., x_n(t), t; u_1(t), u_2(t), \ldots., u_m(t))$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\dot{x}_n(t) = f_n(x_1(t), x_2(t), \ldots., x_n(t), t; u_1(t), u_2(t), \ldots., u_m(t))$$

όπου $u_i(t)$ είναι οι μεταβλητές εισόδου, $f_i : R^n x R^m x R^+ \longrightarrow R$ είναι οι μεταβλητές εξόδου και $x_i(t)$ είναι οι μεταβλητές κατάστασης από τις οποίες υπολογίζεται η μελλοντική απόκριση του συστήματος. Όταν είναι γνωστή η παρούσα κατάσταση $x_i(t)$ και οι είσοδοι u(t) οι παραπάνω εξισώσεις περιγράφουν την δυναμική τους.

Ένας άλλος τρόπος που μπορεί να παρασταθεί ένα σύστημα είναι από μία διανυσματική διαφορική εξίσωσης πρώτης τάξης.

$$\dot{x}(t) = f(x(t), t; u(t))$$

όπου x(t)= $\begin{bmatrix} x_1(t) \\ x_2(t) \\ \ldots\ldots \\ x_m(t) \end{bmatrix} : R^+ \to R^n$ είναι το διάνυσμα κατάστασης του συστήματος,

$f(x(t),t;u(t)) = \begin{bmatrix} f_1(x(t),t,u(t)) \\ f_2(x(t),t,u(t)) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x(t),t,u(t)) \end{bmatrix} : R^n x R^m x R^+ \to R^n$ είναι το διάνυσμα

έξοδος του συστήματος

και όπου u(t)= $\begin{bmatrix} u_1(t) \\ u_2(t) \\ \ldots\ldots \\ u_m(t) \end{bmatrix} : R^+ \to R^m$ είναι το διάνυσμα εισόδου του συστήματος

Ένα τέτοιο μαθηματικό πρότυπο ονομάζεται μοντέλο του χώρου καταστάσεως.

Έστω x(t): $\mathbb{R} \to \mathbb{R}$ τότε **ο μετασχηματισμός Laplace** υπολογίζεται από τη σχέση

$$L\{x(t)\} = X(s) := \int_{-\infty}^{+\infty} x(t) e^{-st} dt$$

ενώ **ο μετασχηματισμός Z** μετασχηματίζει τη σχέση x(t): $\mathbb{Z} \to \mathbb{R}$ στη σχέση

$$Z\{x(t)\} = X(z) := \sum_{t=-\infty}^{\infty} x(t) z^{-t}\ .$$

Γενικά ένα πολυμεταβλητό σύστημα S **συνεχούς χρόνου** περιγράφεται από τις **εξισώσεις στο χώρο των καταστάσεων** από τις παρακάτω εξισώσεις ως εξής:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$
$$A \in R^{nxn}, B \in R^{nxm}, C \in R^{pxn}, D \in R^{pxm}$$

Μετασχηματίζοντας το παραπάνω σύστημα κατά Laplace και θέτοντας ως:

X(s)= L{x(t)}

U(s)= L{u(t)}

Y(s)= L{y(t)}

προκύπτει

sX(s)-x(0)=AX(s)+BU(s)

Y(s)=CX(s)+DU(s)

(sI-A)X(s)=x(0)+BU(s) οπότε

$X(s)=(sI-A)^{-1}\ x(0)+(sI-A)^{-1}BU(s)$

$Y(s)=C(sI-A)^{-1}x(0)+C(sI-A)^{-1}BU(s)+DU(s)$

Αν x(0)=0

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

Ο πίνακας $G(s)=[C(sI-A)^{-1}B+D] \in R_{pr}(s)^{pxm}$ ονομάζεται πίνακας συνάρτησης μεταφοράς του συστήματος.

Εάν η συνάρτηση είναι της μορφής $G(s) = k\dfrac{(s-z_1)(s-z_2)....(s-z_n)}{(s-p_1)(s-p_2)....(s-p_n)}$ τότε οι τιμές που μηδενίζουν τον παρονομαστή ονομάζονται πόλοι (pole) και οι τιμές που μηδενίζουν τον αριθμητή ονομάζονται μηδενικά (zero) και η k είναι μία σταθερά που εκφράζει το κέρδος (gain).

Όταν ο βαθμός του πολυωνύμου του παρονομαστή είναι μεγαλύτερος από τον βαθμό του πολυωνύμου του αριθμητή τότε η συνάρτηση μεταφοράς ονομάζεται strictly proper.

Ένα πολυμεταβλητό σύστημα S **διακριτού χρόνου** περιγράφεται από τις **εξισώσεις του χώρου των καταστάσεων** ως:

$$
\begin{aligned}
&x[n+1] = Ax[n] + Bu[n] \\
&y[n] = Cx[n] + Du[n] \\
&A \in R^{nxn}, B \in R^{nxm}, C \in R^{pxn}, D \in R^{pxm}
\end{aligned}
$$

Όπου η συνάρτηση μεταφοράς παριστάνεται ως :

$$
H(z) = \frac{b_0 z^n + b_1 z^{n-1} + ... + b_{n-1} z + b_n}{a_0 z^m + a_1 z^{m-1} + ... + a_{m-1} z + a_m}
$$

Ομοίως και σ' αυτήν την περίπτωση οι τιμές που μηδενίζουν τον παρονομαστή ονομάζονται πόλοι και οι τιμές που μηδενίζουν τον αριθμητή ονομάζονται μηδενικά.

Ένα πολυμεταβλητό σύστημα S **συνεχούς χρόνου** περιγράφεται από το **γενικευμένος χώρο καταστάσεων** όταν είναι της μορφής:

$$
\begin{aligned}
&E\dot{x}(t) = Ax(t) + Bu(t) \\
&y(t) = Cx(t) + Du(t) \\
&A \in R^{nxn}, B \in R^{nxm}, C \in R^{pxn}, D \in R^{pxm}, E \in R^{nxn}
\end{aligned}
$$

όπου ο E ένας μη αντιστρέψιμος πίνακας
Αν E αντιστρέψιμος τότε το σύστημα ανάγεται στον χώρο καταστάσεων.

Ένα σύστημα λέγεται ότι έχει **καθυστέρηση** όταν είναι της μορφής:

$$\dot{x}(t) = Ax(t) + Bu(t-\theta)$$
$$y(t+\phi) = Cx(t) + Du(t-\theta)$$
$$A \in R^{nxn}, B \in R^{nxm}, C \in R^{pxn}, D \in R^{pxm}$$

Όπου θ είναι η διαφορά φάσης εισόδου και φ η διαφορά φάσης εξόδου.

Όλα τα συστήματα έχουν κάποιες βασικές ιδιότητας:

1. Ένα σύστημα ονομάζεται **προσθετικό** αν για κάθε ζεύγος εισόδων x(t), u(t) (x[k],u[k]) ισχύει

[F(x+u)](t)=(Fx)(t)+(Fu)(t) , ([F[x[k]+u[k]]=F[x[k]]+F[u[k]])

Δηλαδή ένα σύστημα είναι προσθετικό αν η απόκριση στην έξοδο y(t)= [F(x+y)](t) (y[k]= [F[x+y]][k]) η οποία προκύπτει από την είσοδο x(t)+u(t) (x[k]+y[k]) είναι ίση με το άθροισμα της απόκρισης (Fx)(t) ([Fx][k]) στην έξοδο λόγω της εισόδου x(t) συν την απόκριση στην έξοδο (Fu)(t) ([Fu][k])λόγω της εισόδου u(t) (u[k]).

2. Χαρακτηρίζεται ως **ομογενές** αν για κάθε μιγαδική σταθερά a και για είσοδο x[k] ισχύει

[F(ax)](t)=a(Fx)(t) (F[ax[k]]=aF[x[k]])

Η σχέση αυτή δείχνει ότι στα ομογενή συστήματα στην έξοδο y(t)=[F(ax)](t) (y[k]=[F[ax][k]]) η οποία προκύπτει από την είσοδο ax(t) (ax[k]) ισούται με a-φορές την απόκριση (Fx)(t) ([Fx][k]) στην έξοδο λόγω της εισόδου x(t) (x[k]).

3. Ένα σύστημα y(t)=(Fx)(t) (y[k] = [Fx][k]) το οποίο είναι προσθετικό και ομογενές ονομάζεται **γραμμικό.**

Οι δύο παραπάνω συνθήκες είναι ισοδύναμες με την συνθήκη

[F(ax+bu)](t)=a(Fx)(t)+b(Fu)(t) ([F[ax[k]+bu[k]]= aFx[k] + bFu[k])

με x(t) και u(t) (x[k] και u[k]) ένα ζεύγος εισόδων και a,b πραγματικές σταθερές.

4. **Αιτιατό** καλείται ένα σύστημα αν για κάθε χρονική στιγμή t_0 και για κάθε είσοδο x(t) η απόκριση $y(t_0)$ του συστήματος τη χρονική στιγμή t_0 εξαρτάται μόνο από την είσοδο μέχρι την χρονική στιγμή t_0.

5. Και χαρακτηρίζεται ως **χρονικά αναλλοίωτο ή σταθερό** προς το χρόνο αν για κάθε είσοδο x(t) και για κάθε t_1 η έξοδος, σε είσοδο $x(t-t_1)$,ισούται με $y(t-t_1)$.

6. Ένα σύστημα συνεχούς χρόνου ονομάζεται **πεπερασμένης διάστασης** αν για κάποιο μη αρνητικό ακέραιο n η παράγωγος n τάξεως της είναι συνάρτηση των $y^{(i)}(t)$ και $x^{(i)}(t)$ για $0 \leq i \leq n-1$. Η n-στης τάξης παράγωγος $y^{(n)}(t)$ της εξόδου y(t) μπορεί να εξαρτάται και από την i-τάξεως παράγωγο $x^{(i)}(t)$ της εισόδου x(t) για $i \geq n$.

7. **Ασυμπτωτικά ευσταθές** ονομάζεται το σύστημα αν και μόνο αν για κάθε σύνολο αρχικών συνθηκών $y(0^-)$, $y^{(1)}(0^-)$,…, $y^{(n-1)}(0^-)$ η ελεύθερη απόκριση $y_a(t)$ του συστήματος ικανοποιεί την σχέση

$$\lim_{t \to \infty} y_a(t) \to 0 .$$

Ο όρος ασυμπτωτικά σημαίνει ότι η γραφική παράσταση της ελεύθερης απόκρισης $y_a(t)$ ως προς το χρόνο t, ασυμπτωτικά τείνει στην ευθεία y=0 καθώς ο χρόνος t τείνει στο άπειρο.

8. Η **μόνιμη απόκριση** ενός συστήματος είναι το μέρος της ολικής απόκρισής y(t) το οποίο δεν τείνει στο μηδέν, ενώ **μεταβατική απόκριση** είναι το μέρος της ολικής απόκρισής y(t) το οποίο τείνει στο μηδέν όταν ο χρόνος t τείνει στο άπειρο.

9. Ένα σύστημα ονομάζεται **ελέγξιμο** όταν περιγράφεται από τις εξισώσεις του χώρου των καταστάσεων, διαφορετικά το ζεύγος (A, B) είναι ελέγξιμο αν για κάθε αρχική κατάσταση x(0) υπάρχει χρόνος $t_1>0$ και είσοδος u(t), $t \in [0, t_1]$ τέτοια ώστε η κατάσταση $x(t_1)=0$.

10. Ένα σύστημα ονομάζεται **παρατηρήσημο** το οποίο περιγράφεται από τις εξισώσεις του χώρου των καταστάσεων διαφορετικά το ζεύγος (A, C) ονομάζεται παρατηρήσιμο, αν υπάρχει πεπερασμένος χρόνος $t_1 \geq t_0$ τέτοιος ώστε για κάθε αρχική κατάσταση $x_0=x(t_0)$ η γνώση της εισόδου $\{u(t), t_0 \leq t \leq t_1\}$ και της εξόδου $\{y(t), t_0 \leq t \leq t_1\}$ να επιτρέπουν τον προσδιορισμό της αρχικής κατάστασης x_0.

11. Ένα κλειστό σύστημα χαρακτηρίζεται ως **εσωτερικά ευσταθές** όταν η επιλογή του αντισταθμιστή είναι τέτοια ώστε όλα τα μηδενικά του να βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο ($\mathbb{C}^- = \{s \in \mathbb{C}, \mathrm{Re} < 0\}$).

12. Ένα σύστημα είναι **σταθεροποιήσιμο** αν τα αποσυζευτικά μηδενικά του συστήματος είναι εντός του $\mathbb{C}^-$ (αποσυζευτικά μηδενικά ονομάζονται οι λύσεις του μέγιστου κοινού διαιρέτη του αριθμητή και παρονομαστή της συνάρτησης μεταφοράς).

Μέσω του αντισταθμιστή και της μοναδιαίας ανάδρασης γίνεται η προσπάθεια σταθεροποίησης του συστήματος.

Οι επιθυμητές ιδιότητες του συστήματος είναι:

- Η εσωτερική ευστάθεια
- Οι έξοδοι του συστήματος να έχουν επιθυμητή συμπεριφορά
- Να μην υπάρχουν ανεπιθύμητες αλληλεπιδράσεις μεταξύ των εισόδων και των εξόδων.

13. Τέλος **συνέλιξη** δύο σημάτων h(t) και x(t) (h[k], x[k]) συνεχούς χρόνου (διακριτού χρόνου) συμβολίζεται ως h*x και ορίζεται ως

για την περίπτωση συνεχούς χρόνου ως το ολοκλήρωμα

$$y(t) = (h * x)(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau$$

και για την περίπτωση διακριτού χρόνου ως το άθροισμα

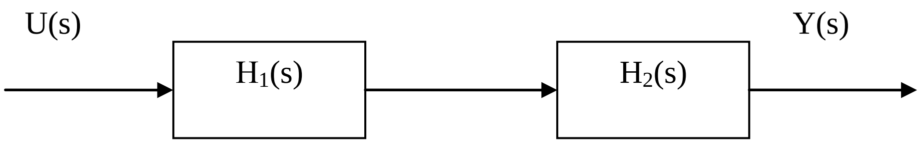
$$y[k] = (h[k] * x[k]) = \sum_{n=-\infty}^{\infty} x[n]h[k-n] = \sum_{n=-\infty}^{\infty} h[k]x[k-n]$$

και είναι η πράξη μέσω της οποίας υπολογίζεται η έξοδος του συστήματος.

1.1.3. <u>Συνδεσμολογία Συστημάτων</u>

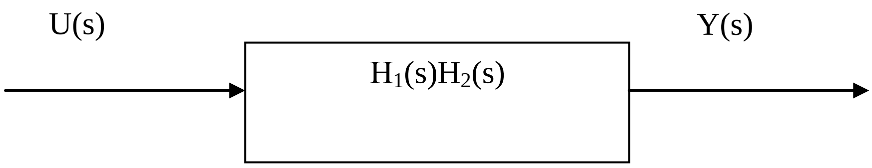
Οι τρόποι συνδεσμολογίας των συστημάτων είναι τρεις.

1 Σε σειρά (Σχήμα 3)



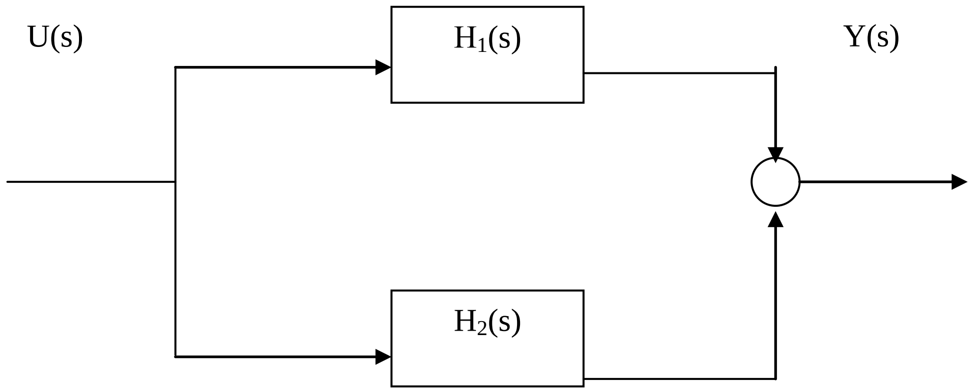
Σχήμα 3

όπου η νέα συνάρτηση μεταφοράς είναι ίση με το γινόμενο των συναρτήσεων μεταφοράς των συναρτήσεων μεταφοράς των δύο συστημάτων (Σχήμα 4).



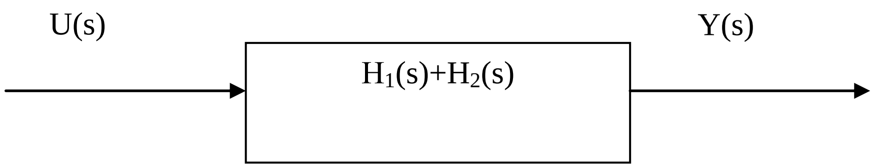
Σχήμα 4

2 Παράλληλη σύνδεση (Σχήμα 5)



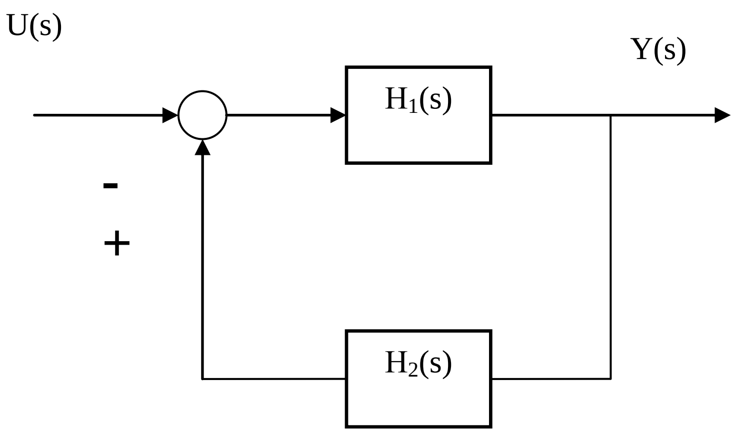
Σχήμα 5

όπου η νέα συνάρτηση μεταφοράς είναι ίση με το άθροισμα των συναρτήσεων μεταφοράς των συναρτήσεων μεταφοράς των δύο συστημάτων (Σχήμα 6).



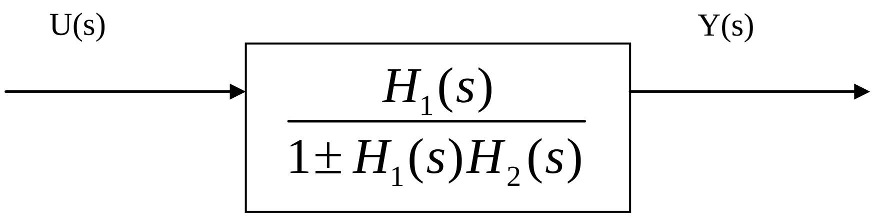
Σχήμα 6

3 Με ανάδραση (Σχήμα 7)



Σχήμα 7

όπου η νέα συνάρτηση μεταφοράς ισούται με $\frac{H_1(s)}{1 \pm H_1(s)H_2(s)}$ (Σχήμα 8)



Σχήμα 8

1.2. Υπολογισμός Αντισταθμιστή

1.2.1. *Βασική Θεωρία*

Για την μελέτη των συναρτήσεων μεταφοράς και των αντίστοιχων αντισταθμιστών των συστημάτων κλειστού βρόχου είναι απαραίτητα να οριστούν τα παρακάτω σύνολα.

- $\mathbb{C}$ συμβολίζεται το σύνολο των μιγαδικών αριθμών,
- $\mathbb{R}$ το σύνολο των πραγματικών αριθμών,
- $\mathbb{Z}$ το σύνολο των ακεραίων,
- $\mathbb{Z}^+$ το σύνολο των μη αρνητικών ακεραίων.
- $\mathbb{R}(s)$ απεικονίζει το σύνολο των ρητών συναρτήσεων που είναι της μορφής

$$p(s) = \frac{n(s)}{d(s)} \text{ όπου } n(s), d(s) \in \mathbb{R}[s]$$

όπου ο αριθμητής και ο παρονομαστής είναι πολυώνυμα.

- $\mathbb{R}_{pr}(s)$ το σύνολο όλων των proper πραγματικών συναρτήσεων
- $\mathbb{R}_{pr0}(s)$ το σύνολο των strictly proper πραγματικών συναρτήσεων
- Έστω K ένα σύνολο τότε ως $K^{p\times m}$ συμβολίζεται το σύνολο όλων των $p\times m$ πινάκων του οποίου τα στοιχεία είναι στοιχεία του συνόλου K.

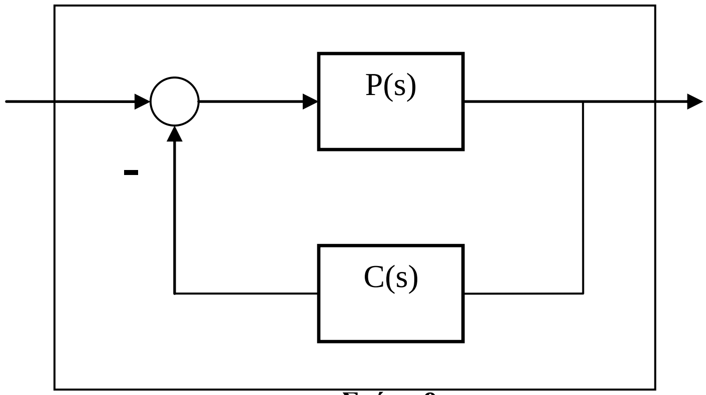
Ένας πίνακας ονομάζεται ομαλός ή μη ιδιάζων όταν η ορίζουσα του είναι διάφορη του μηδενός.

Το κλάσμα $p(s) = \frac{n(s)}{d(s)} \in \mathbb{R}(s)$ καλείται **proper** αν $\lim_{s\to\infty} p(s) = k \in \mathbb{R}$ και **strictly proper** στη περίπτωση όπου το k=0.

Μία συνάρτηση $P(s) \in \mathbb{R}_{pr}(s)$ ονομάζεται **biproper** αν και μόνο αν $P^{-1}(s) \in \mathbb{R}_{pr}(s)$ και ισοδύναμα αν και μόνο αν $\lim_{s\to\infty} P(s) = E \in \mathbb{R}$ όπου $|E| \neq 0$.

Έστω S ένα γραμμικό, χρονικά αναλλοίωτο πολυμεταβλητό σύστημα του οποίου η συνάρτηση μεταφοράς P(s) χαρακτηρίζεται ως strictly proper. Τότε μπορεί να κατασκευαστεί ένα κλειστό σύστημα ανάδρασης Σ_c όπου C(s) θεωρείται η συνάρτηση μεταφοράς του αντισταθμιστή (Σχήμα 10).

ΟΡΙΣΜΟΣ 1. Αν για ένα σύστημα P(s) υπάρχει ένας σταθεροποιητικός αντισταθμιστής C(s), τότε λέμε ότι το P(s) είναι σταθεροποιήσιμο και ότι το C(s) σταθεροποιεί το P(s).



Σχήμα 9

Ένα γραμμικό χρονικά αναλλοίωτο σύστημα Σ, χαρακτηρίζεται ως strictly proper όταν η συνάρτηση μεταφοράς του συστήματος $P(s) \in \mathbb{R}_{pr0}(s)$ που δίνεται από τη σχέση

$$P(s) = \frac{n(s)}{d(s)}$$

όπου $\deg(n(s)) < \deg(d(s))$

Σημαντική προυπόθεση είναι το μηδενικά του P(s) να βρίσκονται στο αρνητικό μιγαδικό ημιεπίπεδο.

Υπολογίζοντας την αντίστροφη συνάρτηση της συνάρτησης μεταφοράς $P(s)^{-1} \in \mathbb{R}(s)$ και εκτελώντας την ευκλείδεια διαίρεση προκύπτουν οι δύο περιπτώσεις

1) το υπόλοιπο της διαίρεσης να είναι μηδέν

2) το υπόλοιπο να είναι ένα πολυώνυμο μικρότερου βαθμού από τον παρονομαστή αντίστοιχα.

Στη πρώτη περίπτωση η διαίρεση είναι τέλεια και το $P(s)^{-1}$ γράφεται ως

$$p^{-1}(s) = \frac{d(s)}{n(s)} \in \mathbb{R}(s)$$

Και στη δεύτερη περίπτωση η διαίρεση είναι ατελής οπότε γράφεται ως

$$p^{-1}(s) = \frac{d(s)}{n(s)} = q(s) + \frac{r(s)}{n(s)}$$

με $d(s) = q(s)n(s) + r(s)$, όπου το υπόλοιπο της διαίρεσης $r(s)$ είναι διάφορο του μηδενικού πολυωνύμου.

Το πολυώνυμο $P(s)^{-1}$ μπορεί να διασπαστεί σε άθροισμα ενός πολυωνύμου και ενός strictly proper κλάσματος με βαθμό παρονομαστή μεγαλύτερο από βαθμό αριθμητή, δηλαδή

$$P(s)^{-1} = Q(s) + \frac{r(s)}{n(s)} = [P(s)^{-1}]_{pol} + [P(s)^{-1}]_{sp} \ , \qquad (1)$$

όπου $Q(s) = [P(s)^{-1}]_{pol} \in \mathbb{R}(s) \kappa\alpha\iota [P(s)^{-1}]_{sp} \in \mathbb{R}(s)$.

Η συνάρτηση μεταφοράς ενός συστήματος κλειστού βρόχου Σ_c υπολογίζεται από τη σχέση

$$\begin{aligned} &P_C(s) = \frac{P(s)}{1 + P(s)C(s)} \\ &\acute{\alpha}\rho\alpha \\ &P_C(s)^{-1} = \frac{1 + P(s)C(s)}{P(s)} = P(s)^{-1} + C(s) \end{aligned} \qquad (2)$$

όπου C(s)$\in \mathbb{R}_{pr}(s)$

Συνεπώς $C(s) = P_C(s)^{-1} - P(s)^{-1}$

ΠΡΟΤΑΣΗ 2. Η σχέση $P_1(s)^{-1} - P_2(s)^{-1} = C(s) \in \mathbb{R}_{pr}(s)$ είναι σχέση ισοδυναμίας στο $R(\mathbb{R}_{pr0}(s) \times \mathbb{R}_{pr0}(s))$

Απόδειξη

- $(P_1(s), P_1(s)) \in R$ άρα είναι ανακλαστική
- $$(P_1(s), P_2(s)) \in R \Leftrightarrow P_1(s)^{-1} - P_2(s)^{-1} = C(s) \in \mathbb{R}_{pr}(s) \Leftrightarrow$$
  $$\Leftrightarrow -P_1(s)^{-1} + P_2(s)^{-1} = -C(s) \in \mathbb{R}_{pr}(s)$$

άρα είναι συμμετρικό

- $$(P_1(s), P_2(s)) \in R \Leftrightarrow P_1(s)^{-1} - P_2(s)^{-1} = C_1(s) \in \mathbb{R}_{pr}(s)$$
  $$\kappa\alpha\iota$$
  $$(P_2(s), P_3(s)) \in R \Leftrightarrow P_2(s)^{-1} - P_3(s)^{-1} = C_2(s) \in \mathbb{R}_{pr}(s) \Leftrightarrow$$
  $$\Leftrightarrow (P_1(s), P_3(s)) \in R \Leftrightarrow P_1(s)^{-1} - P_3(s)^{-1} = C_3(s) \in \mathbb{R}_{pr}(s)$$
  $$\alpha\varphi o\acute{\upsilon}$$
  $$(P_1(s)^{-1} - P_2(s)^{-1}) + (P_2(s)^{-1} - P_3(s)^{-1}) = P_1(s)^{-1} - P_3(s)^{-1} =$$
  $$= C_1(s) + C_2(s) = C_3(s) \in \mathbb{R}_{pr}(s)$$

άρα είναι μεταβατική

Και επειδή ισχύουν και οι τρείς προϋποθέσεις είναι σχέση ισοδυναμίας.

▲

Το μηδενικά της συνάρτησης μεταφοράς του συστήματος του κλειστού βρόχου πρέπει να ανήκουν στο αρνητικό μιγαδικό ημιεπίπεδο ($\mathbb{C}^- = \{s \in \mathbb{C}, \mathrm{Re} < 0\}$). Συνεπώς το πολυώνυμο $D_{RC}(s) = Q(s)N_{RC}(s) + R_{RC}(s) \in \mathbb{R}(s)$ να έχει μηδενικά στο $\mathbb{C}^-$.

Άρα $Q(s)^{-1}D_{RC}(s) = N_{RC}(s) + Q(s)^{-1}R_{RC}(s)$

και θέτοντας ως $T(s) = Q(s)^{-1}D_{RC}(s)$ (3).

Ο πίνακας T(s) είναι strictly proper.

Από τα παραπάνω προκύπτει ότι

$$P(s) \in \mathbb{R}_{pr0}(s) .$$

Από αυτήν υπολογίζεται η αντίστροφη της συνάρτησης μεταφοράς η οποία διασπώντας την σε πολυωνυμικό και strictly proper μέρος (Σχέση 6)

$$P(s)^{-1} = [P(s)^{-1}]_{pol} + [P(s)^{-1}]_{sp} \tag{4}$$

Q(s) ορίζεται ως το πολυωνυμικό κομμάτι της αντίστροφης της συνάρτησης μεταφοράς

$$Q(s) = [P(s)^{-1}]_{pol} .$$

Ο πίνακας $D_{RC}(s)$ (Σχέση 3) πρέπει να έχει

$$\deg D_{RC}(s) >= 2\deg Q(s) - 1$$

Η αντίστροφη συνάρτηση της συνάρτησης μεταφοράς του συστήματος κλειστού βρόχου υπολογίζεται από τη σχέση

$$P_C(s)^{-1} = D_{RC}(s) N_{RC}(s)^{-1}$$

Με αντισταθμιστή

$$C(s) = P_C(s)^{-1} - P(s)^{-1}$$

όπου $T(s) = Q(s)^{-1} D_{RC}(s) = [T(s)]_{pol} + [T(s)]_{sp}$ και $N_{RC}(s) = [T(s)]_{pol}$.

1.2.2. <u>Ο Αλγόριθμος των Προγραμμάτων</u>

Στους παρακάτω αλγορίθμους, όπως και στα προγράμματα θεωρείται ότι τα συστήματα μας έχουν μία είσοδο.
Και ο πίνακας $D_{RC}(s)$ συμβολίζεται DC.

Πρόγραμμα compensatorabcd

- Δώσε πίνακες A, B, C, D
- Υπολόγισε συνάρτηση μεταφοράς tf και εκτύπωσε
- Διάσπασε την αντίστροφη της tf σε pol και sp κομμάτια και θέσε το pol ως Q και εκτύπωσε
- Υπολόγισε το βαθμό του πολυωνύμου Q και εκτύπωσε 2degQ-1
- Δώσε τον πίνακα DC (στοιχεία του είναι οι πόλοι που επιθυμούμε να έχει το νέο σύστημα) όπου degDC>=2degQ-1
- Θέσε $T(s)=Q(s)^{-1}DC(s)$
- Διάσπασε T(s) σε pol και sp και θέσε το pol ως N_{RC}
- Υπολόγισε $P_C(s)^{-1}= DC(s)N_{RC}(s)^{-1}$
- Εκτύπωσε $P_C(s)$
- Υπολόγισε $C(s)=P_C(s)^{-1}- P(s)^{-1}$ και εκτύπωσε

Πρόγραμμα compensator

- Δώσε συνάρτηση μεταφοράς tf
- Διάσπασε την αντίστροφη της tf σε pol και sp κομμάτια και θέσε το pol ως Q και εκτύπωσε
- Υπολόγισε το βαθμό του πολυωνύμου Q και εκτύπωσε 2degQ-1
- Δώσε τον πίνακα DC (στοιχεία του είναι οι πόλοι που επιθυμούμε να έχει το νέο σύστημα) όπου degDC>=2degQ-1

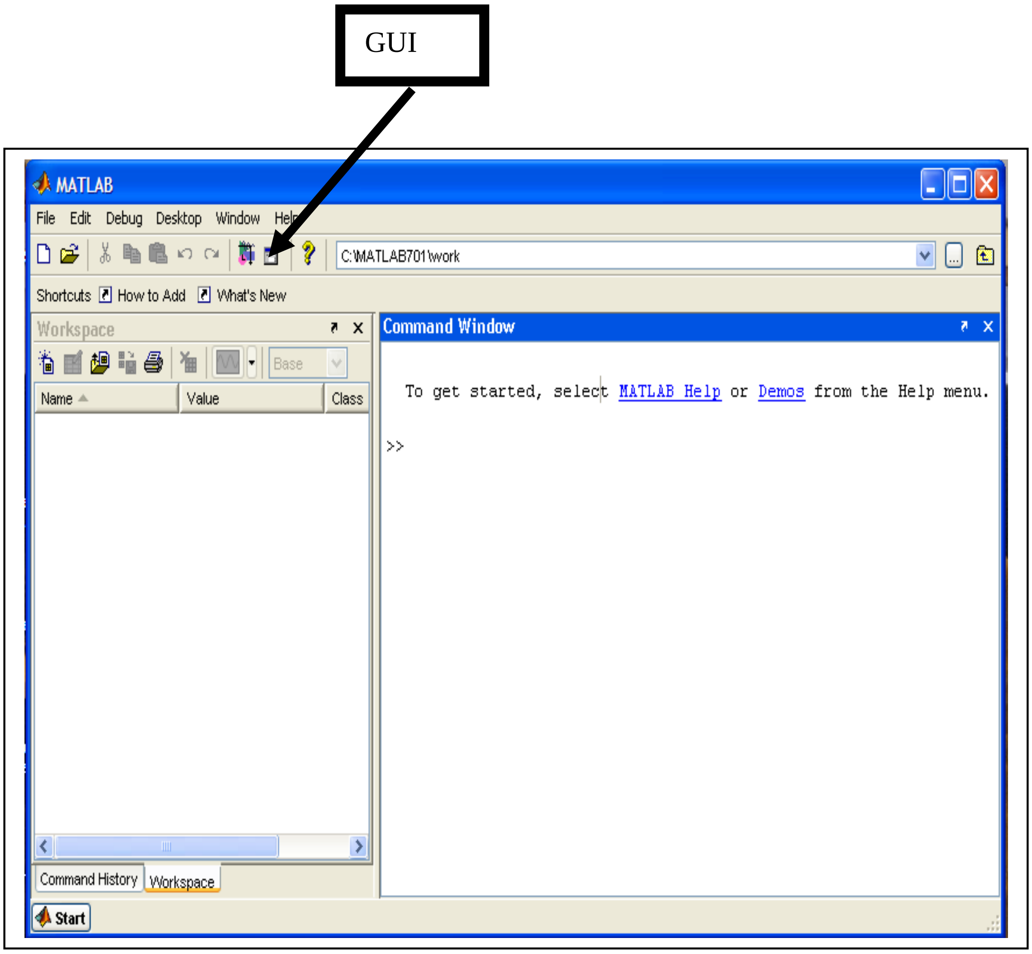
- Θέσε $T(s)=Q(s)^{-1}DC(s)$
- Διάσπασε T(s) σε pol και sp και θέσε το pol ως N_{RC}
- Υπολόγισε $P_C(s)^{-1}= DC(s)N_{RC}(s)^{-1}$
- Εκτύπωσε $P_C(s)$
- Υπολόγισε $C(s)=P_C(s)^{-1}- P(s)^{-1}$ και εκτύπωσε

ΚΕΦΑΛΑΙΟ ΙΙ

2.1. Το GUI

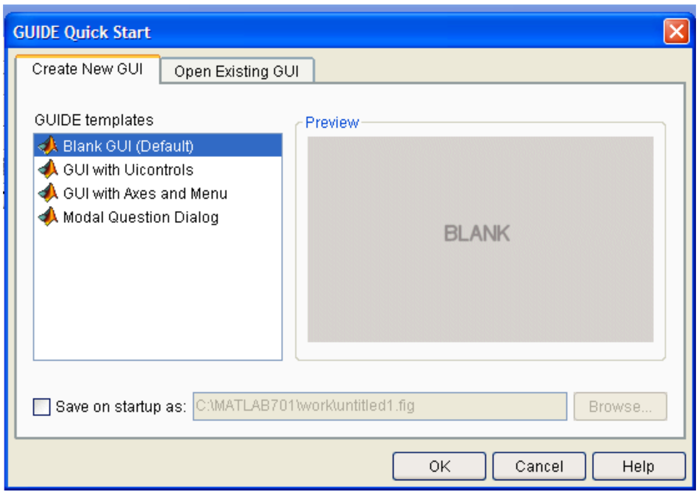
Το Matlab είναι μία γλώσσα προγραμματισμού το οποίο περιέχει στοιχεία από πολλές επιστήμες. Ένα υποπρόγραμμα γραφικών του Matlab είναι το Graphical User Interface (GUI). O κάθε προγραμματιστής και χρήστης των ήδη έτοιμων προγραμμάτων που δουλεύει στο περιβάλλον του GUI, έχει την δυνατότητα να σχεδιάσει και να εφαρμόσει προγράμματα. Τα αποτελέσματα των προγραμμάτων αυτών μπορούν να εξαχθούν με τη χρήση του ¨Pushbuttons¨.

Ανοίγοντας το περιβάλλον του Matlab εμφανίζεται η παρακάτω εικόνα (Σχήμα 10)



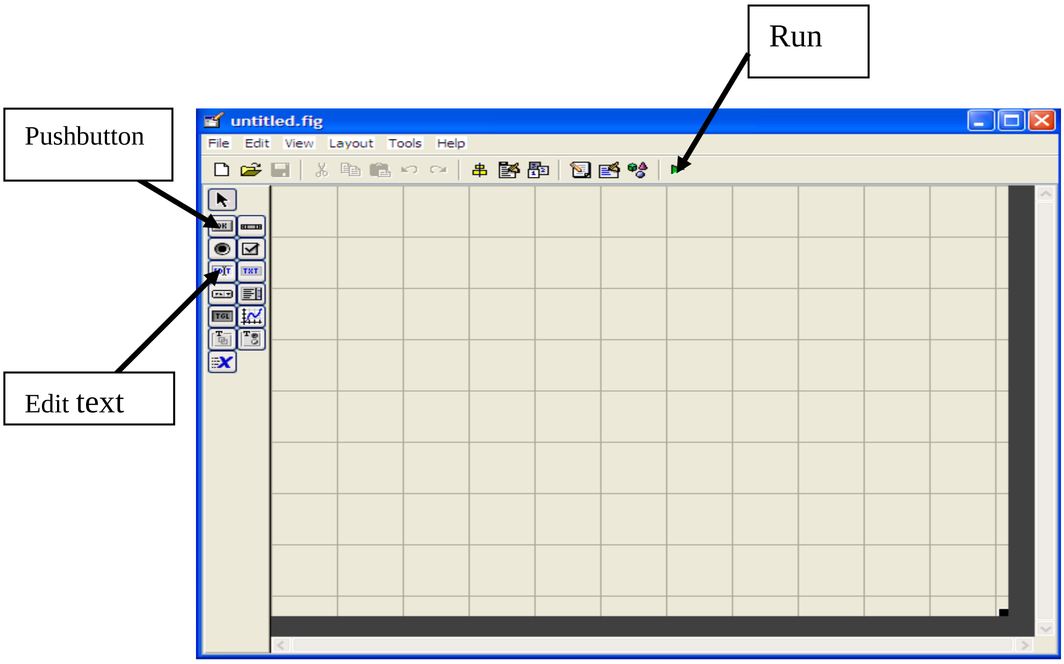
Σχήμα 10

Στη γραμμή των εντολών του προγράμματος υπάρχει ένα σύμβολο με ένα παράθυρο και ένα μολύβι. Πατώντας το κουμπί αυτό ανοίγει το πρόγραμμα GUI και εμφανίζεται το παρακάτω παράθυρο (Σχήμα 11).



Σχήμα 11

Στη συνέχεια επιλέγοντας την επιλογή Blank GUI(Default) και πατώντας OK ανοίγει το παράθυρο επεξεργασίας του GUI (Σχήμα 12).



Σχήμα 12

Στην αριστερή στήλη υπάρχει μία λίστα εικονιδίων – εντολών. Με την εντολή Pushbutton εισάγεται διακόπτης σε οπουδήποτε ορθογώνιο επιλεγεί. Με το Edit text εισάγεται ένα ορθογώνιο παραλληλόγραμμο στο οποίο ή γράφονται δεδομένα ή λαμβάνονται αποτελέσματα (Σχήμα 13 και Σχήμα 18). Εάν χρειαστεί να γραφούν σχόλια πρέπει να γίνει η μετατροπή του Edit text σε Text σε μέσω των Property Inspector. Το παράθυρο του Property Inspector ανοίγει κάνοντας διπλό αριστερό κλικ πάνω στο εικονίδιο το οποίο έχει μαρκαριστεί , είτε επιλέγοντας το μέσω των εντολών του View.

2.2 Προγραμματισμός

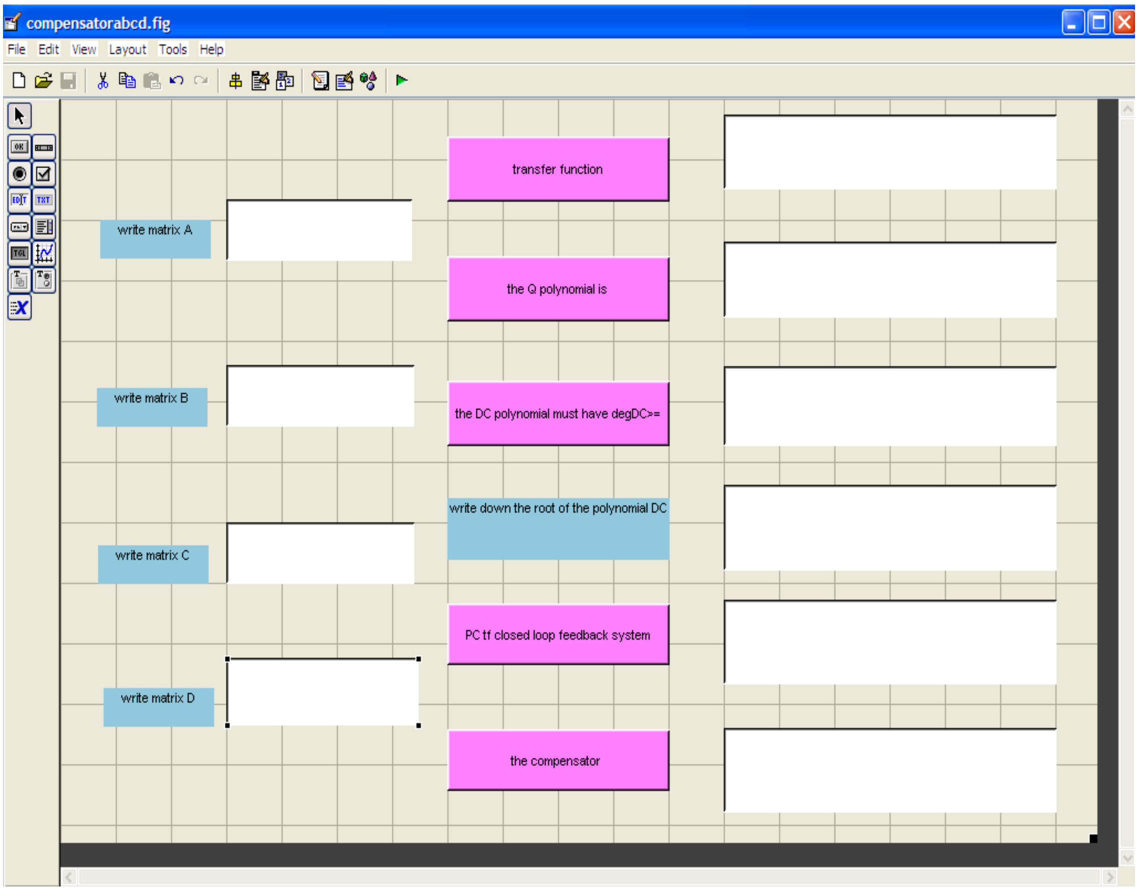
Στόχος των προγραμμάτων είναι να υπολογιστεί ο αντισταθμιστής και η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου.

2.2.1. Πρόγραμμα compensatorabcd

2.2.1.1. *Εισαγωγή*

Στο πρώτο πρόγραμμα της παρούσας εργασίας υπολογίζεται η συνάρτηση μεταφοράς, η συνάρτηση μεταφοράς του κλειστού συστήματος καθώς και τον αντισταθμιστή του, όταν έχει το σύστημα μία είσοδο και είναι γνωστοί οι πίνακες A, B, C και D του συστήματος στο χώρο των καταστάσεων (Σχήμα 13).

Συνεπώς ο χρήστης θα πρέπει να εισάγει τους πίνακες A, B, C και D όπως φαίνεται στο παράθυρο (Σχήμα 13). Τα αποτελέσματα δίνονται στη τελευταία στήλη με τα λευκά κουτιά.



Σχήμα 13

- Με το πάτημα του κουμπιού με τίτλο 'transfer function' εμφανίζεται η συνάρτηση μεταφοράς του συστήματος στο χώρο των καταστάσεων.
- Με το πάτημα του κουμπιού με τίτλο 'the Q polynomial is' εμφανίζεται το πολυώνυμο Q (λόγο ότι το σύστημα έχει μία μόνο είσοδο ο πίνακα Q από πολυωνυμικός γίνεται απλό πολυώνυμο).
- Με το πάτημα του κουμπιού με τίτλο 'the DC polynomial must have degDC>=' εμφανίζεται από ποια τιμή και πάνω πρέπει να είναι ο βαθμός του πολυωνύμου DC.
- Με το πάτημα του κουμπιού με τίτλο 'write down the root of the polynomial DC' στο διπλανό πίνακα πρέπει να γίνει η εισαγωγή των ριζών με τη μορφή πίνακα του πολυωνύμου DC.
- Με το πάτημα του κουμπιού με τίτλο 'PC tf closed loop feedback system' εμφανίζεται η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου.

- Με το πάτημα του κουμπιού με τίτλο 'the compensator' εμφανίζεται ο αντισταθμιστής του συστήματος.

2.2.1.2. *Οι Συναρτήσεις του Προγράμματος compensatorabcd*

Για την καλύτερη κατανόηση του προγράμματος compensatorabcd εξηγούνται παρακάτω μερικές συναρτήσεις οι οποίες χρησιμοποιούνται στο πρόγραμμα.

Υπολογισμός της συνάρτησης μεταφοράς του συστήματος (tf)

- **Ορισμός συνάρτησης tf**

```
function [tf]=calculationstf(a,b,c,d)      % ορισμός συνάρτησης tf έχοντας ως δεδομένα: τους πίνακες a, b, c και d

syms s                      % δήλωση μεταβλητής s
m=size(a);                  % m η διάσταση του πίνακα a
s2=s*eye(m(1,1),m(1,1))-a;  % υπολογισμός του s*I-a για οποιαδήποτε τιμή και αν έχει το m
s3=inv(s2);                 % υπολογισμός του αντιστρόφου της παραπάνω παράστασης
tf=simple(c*s3*b+d);        % υπολογισμός της συνάρτησης μεταφοράς και μέσω της εντολής simple γίνεται η απλοποίηση της παράστασης
```

Υπολογισμός του πολυωνυμικού μέρους Q της αντίστροφης συνάρτησης μεταφοράς (tf)

- **Ορισμός συνάρτησης Q**

```
function [Q]=calculationsQ(a,b,c,d)      % ορισμός συνάρτησης tf έχοντας ως δεδομένα: τους πίνακες a, b, c και d
```

```
syms s                        % δήλωση μεταβλητής s
m=size(a);                    % m η διάσταση του πίνακα a
s2=s*eye(m(1,1),m(1,1))-a; % υπολογισμός του s*I-a για οποιαδήποτε τιμή και
αν έχει το m
s3=inv(s2);                   % υπολογισμός του αντιστρόφου της παραπάνω
παράστασης
s4=simple(c*s3*b+d);          % υπολογισμός της συνάρτησης μεταφοράς και μέσω
της εντολής  simple γίνεται η απλοποίηση της παράστασης
s5=s4^(-1);                   % αντιστροφή της συνάρτησης μεταφοράς
de=(det(s3))^(-1);            % αντιστροφή της ορίζουσας της s3
y1=(s5/de)^(-1);               % αντιστροφή της διαίρεσης των πολυωνύμων s5 με
το de (στόχος της διαίρεσης να γίνει ο διαχωρισμός του αριθμητή και του
παρονομαστή του αντιστρόφου της συνάρτησης μεταφοράς)
de1=sym2poly(de);              % με την εντολή sym2poly( ) το πολυώνυμο το οποίο
βρίσκεται μέσα στη παρένθεση γίνεται πίνακας οπότε το πολυώνυμο de
μετατρέπεται σε πίνακα
y=sym2poly(y1);                % το πολυώνυμο y1 μετατρέπεται σε πίνακα
    [z,p]=deconv(de1,y);  % με την εντολή deconv γίνεται η διαίρεση των δύο
πολυωνύμων (τα οποία έχουν τη μορφή πίνακα) και υπολογίζεται το πηλίκο και το
υπόλοιπο της διαίρεσης (σύμφωνα με τη θεωρία το πηλίκο είναι το πολυωνυμικό
κομμάτι της αντιστρόφου της συνάρτησης μεταφοράς )
     z1=poly2sym(z,s);   % με την εντολή poly2sym( , ) γίνεται η μετατροπή
του πίνακα z σε πολυώνυμο έχοντας ως μεταβλητή το s
     Q=z1;                    % το Q της συνάρτησης είναι ίσο με το z1
```

Υπολογισμός της τιμής που πρέπει να είναι μεγαλύτερος ή ίσος ο βαθμός του πολυωνύμου DC

- **<u>Ορισμός συνάρτησης DegDC</u>**

```
function [DegDC]=calculationsDegDC(a,b,c,d)    % ορισμός της συνάρτησης
DegDC έχοντας ως δεδομένα: τους πίνακες a, b, c και d
```

```
syms s                          %δήλωση μεταβλητής s
m=size(a);                      % m η διάσταση του πίνακα a
s2=s*eye(m(1,1),m(1,1))-a;      % υπολογισμός του s*I-a για οποιαδήποτε τιμή και αν έχει το m
s3=inv(s2);                     % υπολογισμός του αντιστρόφου της παραπάνω παράστασης
s4=simple(c*s3*b+d);            % υπολογισμός της συνάρτησης μεταφοράς και μέσω της εντολής simple γίνεται η απλοποίηση της παράστασης
s5=s4^(-1);                     % αντιστροφή της συνάρτησης μεταφοράς
de=(det(s3))^(-1);              % αντιστροφή της ορίζουσας της s3
y1=(s5/de)^(-1);                % αντιστροφή της διαίρεσης των πολυωνύμων s5 με το de (στόχος της διαίρεσης να γίνει ο διαχωρισμός του αριθμητή και του παρονομαστή του αντιστρόφου της συνάρτησης μεταφοράς)
de1=sym2poly(de);               % με την εντολή sym2poly( ) το πολυώνυμο το οποίο βρίσκεται μέσα στη παρένθεση γίνεται πίνακας οπότε το πολυώνυμο de μετατρέπεται σε πίνακα
y=sym2poly(y1);                 % το πολυώνυμο y1 μετατρέπεται σε πίνακα
    [z,p]=deconv(de1,y);        % με την εντολή deconv γίνεται η διαίρεση των δύο πολυωνύμων (τα οποία έχουν τη μορφή πίνακα) και υπολογίζεται το πηλίκο και το υπόλοιπο της διαίρεσης (σύμφωνα με τη θεωρία το πηλίκο είναι το πολυωνυμικό κομμάτι της αντιστρόφου της συνάρτησης μεταφοράς)
     z1=poly2sym(z,s);          % με την εντολή poly2sym( , ) γίνεται η μετατροπή του πίνακα z σε πολυώνυμο έχοντας ως μεταβλητή το s
     Q=z1;                      % το Q της συνάρτησης είναι ίσο με το z1
Q1=sym2poly(Q);                 % μετατροπή του πολυωνύμου Q σε πίνακα
w=size(Q1);                     % ονομάζω w τις διαστάσεις του πίνακα Q1
DegDC=2*(w(1,2)-1)-1;           % ο βαθμός του πολυωνύμου DC πρέπει να είναι μεγαλύτερος ή ίσος από 2*degQ-1 (ο πίνακας w είναι πίνακας γραμμή οπότε ο βαθμός του πολυωνύμου στο οποίο αντιστοιχεί θα πρέπει να είναι κατά 1 μικρότερος επειδή η τελευταία στήλη του αντιστοιχεί στο κομμάτι του πολυωνύμου του οποίου ο εκθέτης του s είναι το 0).
```

Υπολογισμός αντισταθμιστή του συστήματος κλειστού βρόχου Com

- <u>**Ορισμός συνάρτησης Com**</u>

`function [Com]=calculationsCom(a,b,c,d,DC)` **% ορισμός συνάρτησης Com έχοντας ως δεδομένα: τους πίνακες a, b, c και d**

`syms s` **%δήλωση μεταβλητής s**

`m=size(a);` **% m η διάσταση του πίνακα a**

`s2=s*eye(m(1,1),m(1,1))-a;` **% υπολογισμός του s*I-a για οποιαδήποτε τιμή και αν έχει το m**

`s3=inv(s2);` **% υπολογισμός του αντιστρόφου της παραπάνω παράστασης**

`s4=simple(c*s3*b+d);` **% υπολογισμός της συνάρτησης μεταφοράς και μέσω της εντολής simple( ) γίνεται η απλοποίηση της παράστασης**

`s5=s4^(-1);` **% αντιστροφή της συνάρτησης μεταφοράς**

`de=(det(s3))^(-1);` **% αντιστροφή της ορίζουσας της s3**

`y1=(s5/de)^(-1);` **% αντιστροφή της διαίρεσης των πολυωνύμων s5 με το de (στόχος της διαίρεσης να γίνει ο διαχωρισμός του αριθμητή και του παρονομαστή του αντιστρόφου της συνάρτησης μεταφοράς)**

`de1=sym2poly(de);` **% με την εντολή sym2poly( ) το πολυώνυμο το οποίο βρίσκεται μέσα στη παρένθεση γίνεται πίνακας οπότε το πολυώνυμο de μετατρέπεται σε πίνακα**

`y=sym2poly(y1);` **% το πολυώνυμο y1 μετατρέπεται σε πίνακα**

`[z,p]=deconv(de1,y);` **% με την εντολή deconv γίνεται η διαίρεση των δύο πολυωνύμων (τα οποία έχουν τη μορφή πίνακα) και υπολογίζεται το πηλίκο και το υπόλοιπο της διαίρεσης (σύμφωνα με τη θεωρία το πηλίκο είναι το πολυωνυμικό κομμάτι της αντιστρόφου της συνάρτησης μεταφοράς)**

`z1=poly2sym(z,s);` **% με την εντολή poly2sym(,) γίνεται η μετατροπή του πίνακα z σε πολυώνυμο έχοντας ως μεταβλητή το s**

`Q=z1;` **% το Q της συνάρτησης είναι ίσο με το z1**

`QW=s-DC;` **% ο QW είναι πίνακας με στοιχεία τη διαφορά s- DC**

```
po=prod(QW);
```
% με την εντολή prod( ) γίνεται ο υπολογισμός του γινομένου όλων των στοιχείων του πίνακα QW

```
poo=expand(po);
```
% με την εντολή expand( ) γίνεται ο υπολογισμός του γινομένου του πολυωνύμου po

```
t1=sym2poly(Q);
```
% μετατροπή του πολυωνύμου Q σε πίνακα

```
po1=sym2poly(poo);
```
% μετατροπή του πολυωνύμου poo σε πίνακα

```
[z4,p4]=deconv(po1,t1);
```
% διαίρεση των πολυωνύμων po1 και t1 έτσι ώστε να διασπαστεί το πολυώνυμο σε pol και σε sp κομμάτια

```
z5=poly2sym(z4,s);
```
% μετατροπή του πίνακα z4 σε πολυώνυμο με μεταβλητή το s

```
pc=poo/z5;
```
% διαιρούμε τα δύο πολυώνυμα έτσι ώστε να υπολογιστεί η αντίστροφη συνάρτηση της συνάρτησης μεταφοράς

```
g1=pc-s5;
```
% υπολογισμός της διαφοράς

```
Com=simplify(g1);
```
% με την εντολή simplify( ) γίνονται οι πράξεις τις παράστασης του πολυωνύμου το οποίο βρίσκεται μέσα στη παρένθεση και έτσι υπολογίζεται ο αντισταθμιστής

Υπολογισμός της συνάρτησης μεταφοράς του συστήματος κλειστού βρόχου PC

- **Ορισμός συνάρτησης PC**

```
function [PC]=calculationsPC(a,b,c,d,DC)
```
%Ορισμός συνάρτησης PC έχοντας ως δεδομένα: τους πίνακες a, b, c και d

```
syms s
```
%δήλωση μεταβλητής s

```
m=size(a);
```
% m η διάσταση του πίνακα a

```
s2=s*eye(m(1,1),m(1,1))-a;
```
% υπολογισμός του s*I-a για οποιαδήποτε τιμή και αν έχει το m

```
s3=inv(s2);
```
% υπολογισμός του αντιστρόφου της παραπάνω παράστασης

```
s4=simple(c*s3*b+d);
```
% υπολογισμός της συνάρτησης μεταφοράς και μέσω της εντολής simple γίνεται η απλοποίηση της παράστασης

`s5=s4^(-1);` **% αντιστροφή της συνάρτησης μεταφοράς**

`de=(det(s3))^(-1);` **% αντιστροφή της ορίζουσας της s3**

`y1=(s5/de)^(-1);` **% αντιστροφή της διαίρεσης των πολυωνύμων s5 με το de (στόχος της διαίρεσης να γίνει ο διαχωρισμός του αριθμητή και του παρονομαστή του αντιστρόφου της συνάρτησης μεταφοράς)**

`de1=sym2poly(de);` **% με την εντολή sym2poly( ) το πολυώνυμο το οποίο βρίσκεται μέσα στη παρένθεση γίνεται πίνακας, οπότε το πολυώνυμο de μετατρέπεται σε πίνακα**

`y=sym2poly(y1);` **% το πολυώνυμο y1 μετατρέπεται σε πίνακα**

`[z,p]=deconv(de1,y);` **% με την εντολή deconv γίνεται η διαίρεση των δύο πολυωνύμων (τα οποία έχουν τη μορφή πίνακα) και υπολογίζεται το πηλίκο και το υπόλοιπο της διαίρεσης (σύμφωνα με τη θεωρία το πηλίκο είναι το πολυωνυμικό κομμάτι της αντιστρόφου της συνάρτησης μεταφοράς)**

`z1=poly2sym(z,s);` **% με την εντολή poly2sym(,) γίνεται η μετατροπή του πίνακα z σε πολυώνυμο έχοντας ως μεταβλητή το s**

`Q=z1;` **% το Q της συνάρτησης είναι ίσο με το z1**

`QW=s-DC;` **% ο QW είναι πίνακας με στοιχεία τη διαφορά s- DC**

`po=prod(QW);` **% με την εντολή prod( ) γίνεται ο υπολογισμός του γινομένου όλων των στοιχείων του πίνακα QW**

`poo=expand(po);` **% με την εντολή expand( ) γίνεται ο υπολογισμός του γινομένου του πολυωνύμου po**

`t1=sym2poly(Q);` **% μετατροπή του πολυωνύμου Q σε πίνακα**

`po1=sym2poly(poo);` **% μετατροπή του πολυωνύμου poo σε πίνακα**

`[z4,p4]=deconv(po1,t1);` **% διαίρεση των πολυωνύμων po1 και t1 έτσι ώστε να διασπαστεί το πολυώνυμο σε pol και σε sp κομμάτια**

`z5=poly2sym(z4,s);` **% μετατροπή του πίνακα z4 σε πολυώνυμο με μεταβλητή το s**

`pc1=poo/z5;` **% διαιρούμε τα δύο πολυώνυμα έτσι ώστε να υπολογιστεί η αντίστροφη συνάρτηση της συνάρτησης μεταφοράς**

`PC=pc1^(-1);` **% υπολογισμός της συνάρτησης μεταφοράς του κλειστού συστήματος**

2.2.1.3. *Το πρόγραμμα compensatorabcd*

```
function varargout = compensatorabcde(varargin)
% COMPENSATORABCDE M-file for compensatorabcde.fig
%
%
% NOTICE : THE UNKNOWN SIGNED s.
%
% THE MATRIX FORMS ARE
% EXAMPLE :
% WRITE MATRIX A : [3 -2;1 0]
% WRITE MATRIX B : [2;0]
% WRITE MATRIX C : [0.5 0.5]
% WRITE MATRIX D : [0]
% WRITE DOWN THE ROOT OF THE POLYNOMIAL DC : [-2;-3]
%
%
%      COMPENSATORABCDE, by itself, creates a new COMPENSATORABCDE or
raises the existing
%      singleton*.
%
%      H = COMPENSATORABCDE returns the handle to a new
COMPENSATORABCDE or the handle to
%      the existing singleton*.
%
%      COMPENSATORABCDE('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in COMPENSATORABCDE.M with the given
input arguments.
%
%      COMPENSATORABCDE('Property','Value',...) creates a new
COMPENSATORABCDE or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before compensatorabcde_OpeningFunction gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to compensatorabcde_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help compensatorabcde

% Last Modified by GUIDE v2.5 13-Oct-2009 18:33:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compensatorabcde_OpeningFcn, ...
```

```
                   'gui_OutputFcn',  @compensatorabcde_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before compensatorabcde is made visible.
function compensatorabcde_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to compensatorabcde (see VARARGIN)

% Choose default command line output for compensatorabcde
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes compensatorabcde wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = compensatorabcde_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

%όλα τα παραπάνω δηλώνονται από το ίδιο το GUI κατά το σχεδιασμό του προγράμματος

```
syms s
```
% δήλωση μεταβλητής s

%δήλωση στο πρόγραμμα του πίνακα A

```
function edit1_Callback(hObject, eventdata, handles)
```

```
a=str2num(get(hObject,'string'));


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%δήλωση στο πρόγραμμα του πίνακα B

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double



% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%δήλωση στο πρόγραμμα του πίνακα C

```
function edit3_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%δήλωση στο πρόγραμμα του πίνακα D

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του Pushbuttontf και κάνοντας τη χρήση της συνάρτησης tf και όλων των παραπάνω στοιχείων όπου εισήγαγε ο χρήστης, το πρόγραμμα εξάγει τη συνάρτηση μεταφοράς

```
% --- Executes on button press in pushbuttontf.
function pushbuttontf_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε ονομασίες a, b, c και d στα στοιχεία που γράφτηκαν στα edit1, edit2, edit3 και edit4 αντίστοιχα έτσι ώστε να δηλωθούν στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθούν για την εύρεση και των υπολοίπων στοιχείων

```
a=str2num(get(handles.edit1,'string'));
b=str2num(get(handles.edit2,'string'));
c=str2num(get(handles.edit3,'string'));
d=str2num(get(handles.edit4,'string'));

[tf]=calculationstf(a,b,c,d);
```

% χρήση της συνάρτησης tf

```
answer=char(tf);
```

% με την εντολή char( ) μετασχηματίζεται ο πίνακας ο οποίος βρίσκεται μέσα στη παρένθεση και βάζει τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s (γίνεται η χρήση της εντολής char γιατί το αποτέλεσμα tf είναι πολυωνυμικό κλάσμα)

```
set(handles.edittf,'string',answer)
```
% εκτύπωση του answer στο edittf

```
function edittf_Callback(hObject, eventdata, handles)
% hObject    handle to edittf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edittf as text
%        str2double(get(hObject,'String')) returns contents of edittf
as a double


% --- Executes during object creation, after setting all properties.
function edittf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edittf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonQ και κάνοντας τη χρήση της συνάρτησης Q και όλων των παραπάνω στοιχείων όπου εισήγαγε ο χρήστης, το πρόγραμμα εξάγει το πολυώνυμο Q

```
% --- Executes on button press in pushbutton1.
function pushbuttonQ_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε ονομασίες a, b, c και d στα στοιχεία που γράφτηκαν στα edit1, edit2, edit3 και edit4 αντίστοιχα έτσι ώστε να δηλωθούν στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθούν για την εύρεση και των υπολοίπων στοιχείων

```
a=str2num(get(handles.edit1,'string'));
b=str2num(get(handles.edit2,'string'));
c=str2num(get(handles.edit3,'string'));
d=str2num(get(handles.edit4,'string'));

[Q]=calculationsQ(a,b,c,d);   % χρήση της συνάρτησης Q

answer=char(Q);               % μετασχηματισμός του πίνακα Q βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

set(handles.editQ,'string',answer)  % εκτύπωση του answer στο editQ

function editQ_Callback(hObject, eventdata, handles)
% hObject    handle to editGP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editGP as text
%        str2double(get(hObject,'String')) returns contents of editGP as a double



% --- Executes during object creation, after setting all properties.
function editQ_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editGP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonDegDC και κάνοντας τη χρήση της συνάρτησης DegDC και όλων των παραπάνω στοιχείων όπου εισήγαγε ο χρήστης το πρόγραμμα εξάγει παραπάνω από ποια τιμή πρέπει να είναι ο βαθμός του πολυωνύμου DC.

```
% --- Executes on button press in pushbuttonDeg.
function pushbuttonDegDC_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε ονομασίες a, b, c και d στα στοιχεία που γράφτηκαν στα edit1, edit2, edit3 και edit4 αντίστοιχα έτσι ώστε να δηλωθούν στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθούν για την εύρεση και των υπολοίπων στοιχείων

```
a=str2num(get(handles.edit1,'string'));
b=str2num(get(handles.edit2,'string'));
c=str2num(get(handles.edit3,'string'));
d=str2num(get(handles.edit4,'string'));
```

`[DegDC]=calculationsDegDC(a,b,c,d);` **% χρήση της συνάρτησης DegDC**

`set(handles.editDegDC,'string',DegDC)` **% εκτύπωση του DegDC στο editDegDC (σ΄αυτό το σημείο δεν γίνεται η χρήση της εντολής char γιατί το αποτέλεσμα της συνάρτησης DegDC είναι φυσικός αριθμός)**

```
function editDegDC_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double


% --- Executes during object creation, after setting all properties.
function editDegDC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
 % Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%δήλωση στο πρόγραμμα των ριζών του πολυωνύμου DC με τη μορφή πίνακα

```
function editDC_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double



% --- Executes during object creation, after setting all properties.
function editDC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonCom και κάνοντας τη χρήση της συνάρτησης Com και όλων των παραπάνω στοιχείων όπου εισήγαγε ο χρήστης, το πρόγραμμα εξάγει τον αντισταθμιστή του συστήματος

```
% --- Executes on button press in pushbutton3.
```

```
function pushbuttonCom_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε ονομασίες a, b, c, d και DC στα στοιχεία που γράφτηκαν στα edit1, edit2, edit3, edit4 και editDC αντίστοιχα έτσι ώστε να δηλωθούν στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθούν για την εύρεση και των υπολοίπων στοιχείων

```
a=str2num(get(handles.edit1,'string'));
b=str2num(get(handles.edit2,'string'));
c=str2num(get(handles.edit3,'string'));
d=str2num(get(handles.edit4,'string'));
DC=str2num(get(handles.editDC,'string'));
```

```
[Com]=calculationsCom(a,b,c,d,DC);
```
% χρήση της συνάρτησης Com

```
answer=char(Com);
```
% μετασχηματισμός του πίνακα Com βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

```
set(handles.editCom,'string',answer)
```
% εκτύπωση του answer στο editCom

```
function editCom_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9
as a double


% --- Executes during object creation, after setting all properties.
function editCom_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonPC και κάνοντας τη χρήση της συνάρτησης PC και όλων των παραπάνω στοιχείων όπου εισήγαγε ο χρήστης, το πρόγραμμα εξάγει τη συνάρτηση μεταφοράς του κλειστού συστήματος

```
% --- Executes on button press in pushbuttonPC.
function pushbuttonPC_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε ονομασίες a, b, c, d και DC στα στοιχεία που γράφτηκαν στα edit1, edit2, edit3, edit4 και edit DC αντίστοιχα έτσι ώστε να δηλωθούν στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθούν για την εύρεση και των υπολοίπων στοιχείων

```
a=str2num(get(handles.edit1,'string'));
b=str2num(get(handles.edit2,'string'));
c=str2num(get(handles.edit3,'string'));
d=str2num(get(handles.edit4,'string'));
DC=str2num(get(handles.editDC,'string'))
[PC]=calculationsPC(a,b,c,d,DC);        % χρήση της συνάρτησης PC
    answer=char(PC);                    % μετασχηματισμός του πίνακα PC
βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

 set(handles.editPC,'string',answer)    % εκτύπωση του answer στο
editPC
```

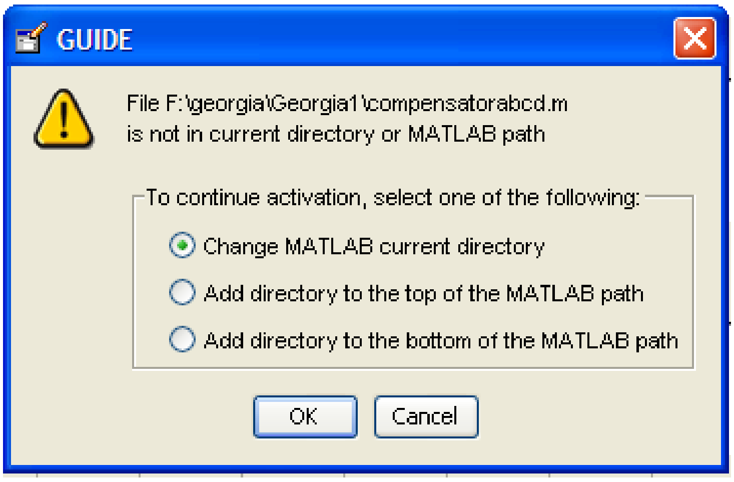
```
 function editPC_Callback(hObject, eventdata, handles)
% hObject    handle to editPC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editPC as text
%        str2double(get(hObject,'String')) returns contents of editPC
as a double
% --- Executes during object creation, after setting all properties.
function editPC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editPC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB%
handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

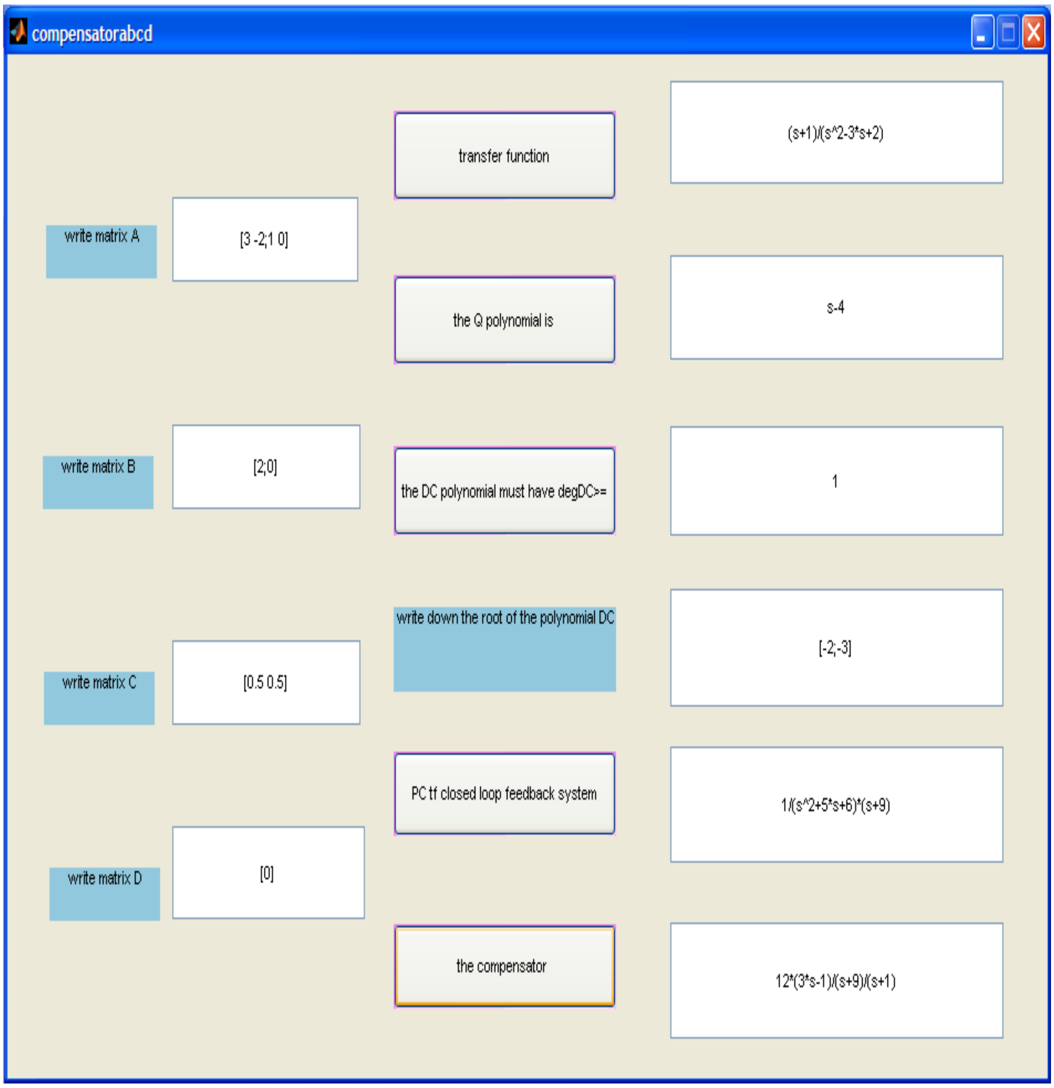
2.2.1.4. *Τρέξιμο Προγράμματος compensatorabcd*

Για να τρέξει το πρόγραμμα ο χρήστης πρέπει να πατήσει το βελάκι (Σχήμα 12). Στη συνέχεια θα εμφανιστεί το παρακάτω παράθυρο.



Σχήμα 14

Και επιλέγοντας την επιλογή που είναι ήδη μαρκαρισμένη και πατώντας το OK ανοίγει το παρακάτω παράθυρο στο οποίο ο χρήστης εισάγει τα δεδομένα.



Σχήμα 15

Οι πίνακες ορίζονται:

Παράδειγμα :

- ο πίνακας A διάστασης nxn : [3 -2;1 0],
- ο πίνακας B διάστασης nx1 : [2;0],
- ο πίνακας C διάστασης 1xn : [0.5 0.5],
- ο πίνακας D διαστάσεων 1x1 : [0]
- ο πίνακας DC διαστάσεων kx1 όπου k αριθμός των ριζών που επιθυμούμε να έχει το πολυώνυμο DC : [-2;-3].

Η δήλωση ενός πίνακα στο Matlab γίνεται γράφοντας τις γραμμές και τις στήλες του πίνακα στην ίδια γραμμή βάζοντας μεταξύ των γραμμών το ελληνικό ερωτηματικό ; και μεταξύ των στηλών αφήνοντας κενά

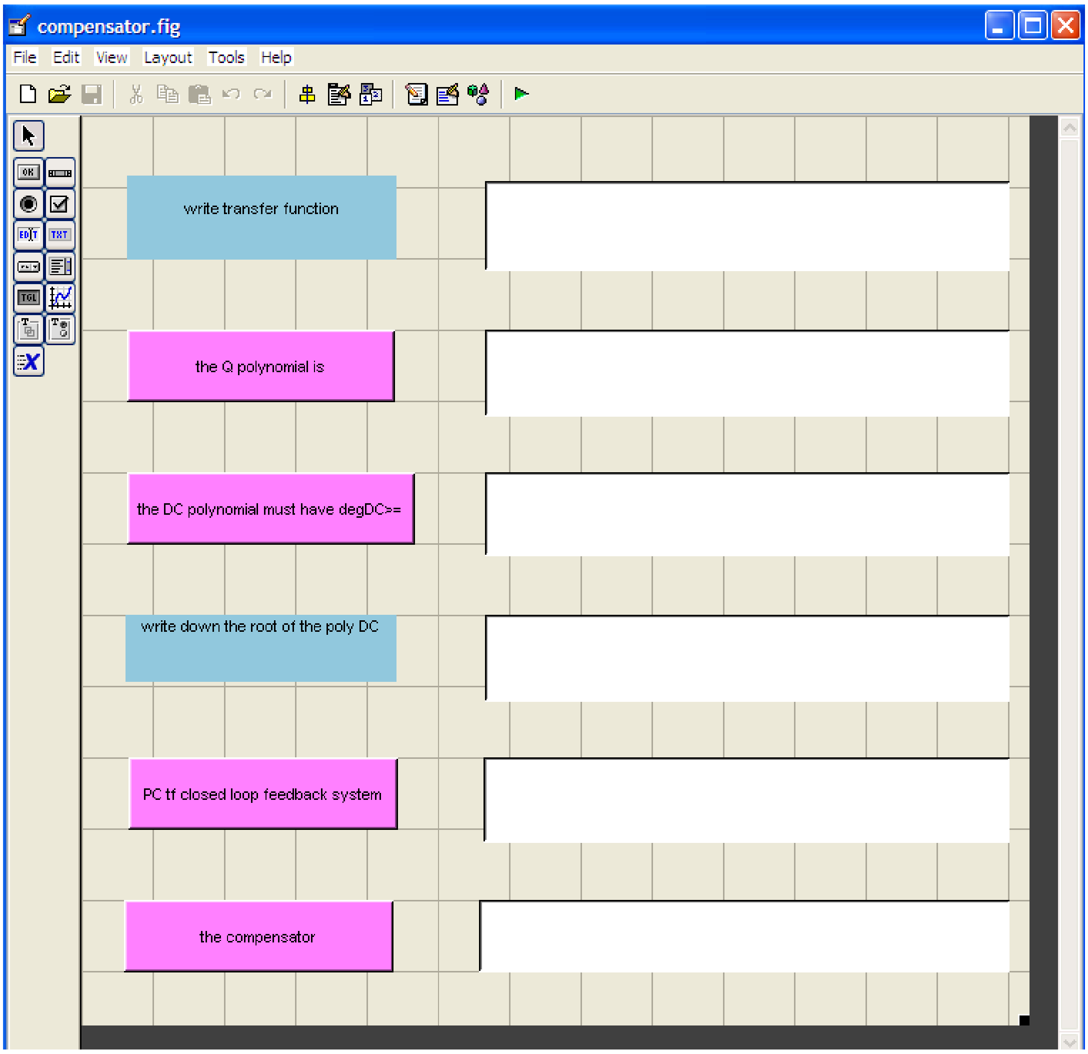
2.2.2. Πρόγραμμα compensator

2.2.2.1. *Εισαγωγή*

Στο δεύτερο πρόγραμμα υπολογίζεται η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου καθώς και τον αντισταθμηστή γνωρίζοντας τη συνάρτηση μεταφοράς ανοιχτού βρόχου.

Η συνάρτηση μεταφοράς είναι ένα πολυώνυμικο κλάσμα επειδή θεωρείται ότι το σύστημα είναι μιας εισόδου.

Συνεπώς ο χρήστης θα πρέπει να εισάγει το πολυωνυμικό κλάσμα tf.



Σχήμα 18

- Με το πάτημα του κουμπιού με τίτλο ‘the Q polynomial is’ εμφανίζεται το πολυώνυμο Q (λόγο ότι το σύστημα έχει μία μόνο είσοδο ο πίνακα Q από πολυωνυμικός γίνεται απλό πολυώνυμο).
- Με το πάτημα του κουμπιού με τίτλο ‘the DC polynomial must have degDC>=’ εμφανίζεται από ποια τιμή και πάνω πρέπει να είναι ο βαθμός του πολυωνύμου DC.
- Με το πάτημα του κουμπιού με τίτλο ‘write down the root of the polynomial DC’ στο διπλανό πίνακα πρέπει να γίνει η εισαγωγή των ριζών με τη μορφή πίνακα του πολυωνύμου DC.
- Με το πάτημα του κουμπιού με τίτλο ‘PC tf closed loop feedback system’ εμφανίζεται η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου.

- Με το πάτημα του κουμπιού με τίτλο 'the compensator' εμφανίζεται ο αντισταθμιστής του συστήματος.

2.2.2.2 *Οι Συναρτήσεις του Προγράμματος compensator*

Στην εφαρμογή του προγράμματος compensator γίνεται η χρήση μερικών συναρτήσεων των οποίων εξηγούνται παρακάτω.

Υπολογισμός του πολυωνυμικού μέρους Q1 της αντίστροφης συνάρτησης μεταφοράς (tf)

- **Ορισμός συνάρτησης Q1**

```
function [Q1]=calculationsQ1(tf)     % ορισμός συνάρτησης Q1 έχοντας ως δεδομένη τη συνάρτηση tf

syms s                               % δήλωση μεταβλητής s
a=solve(tf);                         % εύρεση των ριζών (των μηδενικών) της συνάρτησης μεταφοράς
QW=s-a;                              % υπολογισμός του πίνακα με συντελεστές τη διαφορά s-a
po=prod(QW);                         % δημιουργία πολυωνύμου πολλαπλασιάζοντας όλα τα στοιχεία του πίνακα po
poo=expand(po);                      % υπολογισμός του γινομένου του πολυωνύμου po
Q2=(tf/poo)^(-1);                    % διαίρεση της συνάρτησης μεταφοράς με το πολυώνυμο poo και αντιστροφή του αποτελέσματος (στόχος όλων των παραπάνω είναι να διαχωρίσουμε τον αριθμητή με τον παρονομαστή
t1=sym2poly(Q2);                     % μετατροπή του πολυωνύμου Q2 σε πίνακα
```

```
po1=sym2poly(poo);
```
% μετατροπή του πολυωνύμου poo σε πίνακα

```
[z4,p4]=deconv(t1,po1);
```
% διαίρεση των πολυωνύμων t1 και po1 έτσι ώστε να βρεθεί το πηλίκο και το υπόλοιπο της διαίρεσης (το πηλίκο της διαίρεσης είναι το πολυωνυμικό μέρος του αντιστρόφου της συνάρτησης μεταφοράς

```
Q1=poly2sym(z4,s);
```
% μετατροπή του πίνακα z4 σε πίνακα έχοντας ως μεταβλητή το s

Υπολογισμός της τιμής που πρέπει να είναι μεγαλύτερος ή ίσος ο βαθμός του πολυωνύμου DC1

- **<u>Ορισμός συνάρτησης DegDC1</u>**

```
function [DegDC1]=calculationsDegDC1(tf)
```
% ορισμός συνάρτησης DegDC1 έχοντας ως δεδομένη τη συνάρτηση tf

```
syms s
```
% δήλωση μεταβλητής s

```
a=solve(tf);
```
% εύρεση των ριζών (των μηδενικών) της συνάρτησης μεταφοράς

```
QW=s-a;
```
% υπολογισμός του πίνακα με συντελεστές τη διαφορά s-a

```
po=prod(QW);
```
% δημιουργία πολυωνύμου πολλαπλασιάζοντας όλα τα στοιχεία του πίνακα po

```
poo=expand(po);
```
% υπολογισμός του γινομένου του πολυωνύμου po

```
Q2=(tf/poo)^(-1);
```
% διαίρεση της συνάρτησης μεταφοράς με το πολυώνυμο poo και αντιστροφή του αποτελέσματος (στόχος όλων των παραπάνω είναι να διαχωρίσουμε τον αριθμητή με τον παρονομαστή

```
t1=sym2poly(Q2);
```
% μετατροπή του πολυωνύμου Q2 σε πίνακα

```
pol=sym2poly(poo);
```
% μετατροπή του πολυωνύμου poo σε πίνακα

```
[z4,p4]=deconv(t1,pol);
```
% διαίρεση των πολυωνύμων t1 και pol έτσι ώστε να βρεθεί το πηλίκο και το υπόλοιπο της διαίρεσης (το πηλίκο της διαίρεσης είναι το πολυωνυμικό μέρος του αντιστρόφου της συνάρτησης μεταφοράς

```
m=size(z4);
```
% ονομάζω m τις διαστάσεις του πίνακα z4

```
DegDC1=2*(m(1,2)-1)-1;
```
% ο βαθμός του πολυωνύμου DC1 πρέπει να είναι μεγαλύτερος ή ίσος από 2*degQ1-1 (ο πίνακας m είναι πίνακας γραμμή οπότε ο βαθμός του πολυωνύμου στο οποίο αντιστοιχεί θα πρέπει να είναι κατά 1 μικρότερος επειδή η τελευταία στήλη του αντιστοιχεί στο κομμάτι του πολυωνύμου του οποίου ο εκθέτης του s είναι το 0)

Υπολογισμός της συνάρτησης μεταφοράς του συστήματος κλειστού βρόχου (PC1)

- **<u>Ορισμός συνάρτησης PC1</u>**

```
function [PC1]=calculationsPC1(tf,DC)
```
% ορισμός συνάρτησης PC1 έχοντας ως δεδομένες τις συναρτήσεις tf και DC

```
syms s
```
% δήλωση μεταβλητής s

```
a=solve(tf);
```
% εύρεση των ριζών (των μηδενικών) της συνάρτησης μεταφοράς

```
QW=s-a;
```
% υπολογισμός του πίνακα με συντελεστές τη διαφορά s-a

```
po=prod(QW);
```
% δημιουργία πολυωνύμου πολλαπλασιάζοντας όλα τα στοιχεία του πίνακα po

```
poo=expand(po);
```
% υπολογισμός του γινομένου του πολυωνύμου po

`Q2=(tf/poo)^(-1);` **% διαίρεση της συνάρτησης μεταφοράς με το πολυώνυμο poo και αντιστροφή του αποτελέσματος (στόχος όλων των παραπάνω είναι να διαχωρίσουμε τον αριθμητή με τον παρονομαστή**

`t1=sym2poly(Q2);` **% μετατροπή του πολυωνύμου Q2 σε πίνακα**

`po1=sym2poly(poo);` **% μετατροπή του πολυωνύμου poo σε πίνακα**

`[z4,p4]=deconv(t1,po1);` **% διαίρεση των πολυωνύμων t1 και po1 έτσι ώστε να βρεθεί το πηλίκο και το υπόλοιπο της διαίρεσης (το πηλίκο της διαίρεσης είναι το πολυωνυμικό μέρος του αντιστρόφου της συνάρτησης μεταφοράς**

`Q1=poly2sym(z4,s);` **% μετατροπή του πίνακα z4 σε πίνακα έχοντας ως μεταβλητή το s**

`QW1=s-DC;` **% ο QW1 είναι πίνακας με στοιχεία τη διαφορά s- DC**

`po11=prod(QW1);` **% με την εντολή prod( ) γίνεται ο υπολογισμός του γινομένου όλων των στοιχείων του πίνακα QW1**

`poo11=expand(po11);` **% με την εντολή expand( ) γίνεται ο υπολογισμός του γινομένου του πολυωνύμου poo1**

`t11=sym2poly(Q1);` **% μετατροπή του πολυωνύμου Q1 σε πίνακα**

`po111=sym2poly(poo11);` **% μετατροπή του πολυωνύμου poo11 σε πίνακα**

`[z44,p44]=deconv(po111,t11);` **% διαίρεση των πολυωνύμων poo111 και t11 έτσι ώστε να διασπαστεί το πολυώνυμο σε pol και σε sp κομμάτια**

`z55=poly2sym(z44,s);` **% μετατροπή του πίνακα z44 σε πολυώνυμο με μεταβλητή το s**

`pc=poo11/z55;` **% διαιρούμε τα δύο πολυώνυμα έτσι ώστε να υπολογιστεί η αντίστροφη συνάρτηση της συνάρτησης μεταφοράς**

`PC1=pc^(-1);` **% υπολογισμός της συνάρτησης μεταφοράς του κλειστού συστήματος**

Υπολογισμός αντισταθμηστή του συστήματος κλειστού βρόχου Com1

- **<u>Ορισμός συνάρτησης Com1</u>**

```
function [Com1]=calculationsCom1(tf,DC)
```
% ορισμός συνάρτησης Com1 έχοντας ως δεδομένες τις συναρτήσεις tf και DC

```
syms s
```
% δήλωση μεταβλητής s

```
a=solve(tf);
```
% εύρεση των ριζών (των μηδενικών) της συνάρτησης μεταφοράς

```
QW=s-a;
```
% υπολογισμός του πίνακα με συντελεστές τη διαφορά s-a

```
po=prod(QW);
```
% δημιουργία πολυωνύμου πολλαπλασιάζοντας όλα τα στοιχεία του πίνακα po

```
poo=expand(po);
```
% υπολογισμός του γινομένου του πολυωνύμου po

```
Q2=(tf/poo)^(-1);
```
% διαίρεση της συνάρτησης μεταφοράς με το πολυώνυμο poo και αντιστροφή του αποτελέσματος (στόχος όλων των παραπάνω είναι να διαχωρίσουμε τον αριθμητή με τον παρονομαστή

```
t1=sym2poly(Q2);
```
% μετατροπή του πολυωνύμου Q2 σε πίνακα

```
po1=sym2poly(poo);
```
% μετατροπή του πολυωνύμου poo σε πίνακα

```
[z4,p4]=deconv(t1,po1);
```
% διαίρεση των πολυωνύμων t1 και po1 έτσι ώστε να βρεθεί το πηλίκο και το υπόλοιπο της διαίρεσης (το πηλίκο της διαίρεσης είναι το πολυωνυμικό μέρος του αντιστρόφου της συνάρτησης μεταφοράς

```
Q1=poly2sym(z4,s);
```
% μετατροπή του πίνακα z4 σε πίνακα έχοντας ως μεταβλητή το s

```
QW1=s-DC;
```
% ο QW1 είναι πίνακας με στοιχεία τη διαφορά s- DC

```
po11=prod(QW1);
```
% με την εντολή prod( ) γίνεται ο υπολογισμός του γινομένου όλων των στοιχείων του πίνακα QW1

pool1=expand(pol1); **% με την εντολή expand( ) γίνεται ο υπολογισμός του γινομένου του πολυωνύμου poo1**

t11=sym2poly(Q1); **% μετατροπή του πολυωνύμου Q1 σε πίνακα**

pol11=sym2poly(pool1); **% μετατροπή του πολυωνύμου poo11 σε πίνακα**

[z44,p44]=deconv(pol11,t11); **% διαίρεση των πολυωνύμων poo111 και t11 έτσι ώστε να διασπαστεί το πολυώνυμο σε pol και σε sp κομμάτια**

z55=poly2sym(z44,s); **% μετατροπή του πίνακα z44 σε πολυώνυμο με μεταβλητή το s**

pc=pool1/z55; **% διαιρούμε τα δύο πολυώνυμα έτσι ώστε να υπολογιστεί η αντίστροφη συνάρτηση της συνάρτησης μεταφοράς**

g1=pc-Q2/poo; **% υπολογισμός της διαφοράς**

Com1=simplify(g1); **% με την εντολή simplify( ) γίνονται οι πράξεις τις παράστασης του πολυωνύμου το οποίο βρίσκεται μέσα στη παρένθεση και έτσι υπολογίζεται ο αντισταθμιστής**

2.2.2.3. *Το Πρόγραμμα compensator*

```
function varargout = compensator(varargin)
% COMPENSATOR M-file for compensator.fig
%
%
% WRITE TRANSFER FUNCTION
% EXAMPLE : (s+1)/(s^2-3*s+2)
%
% NOTICE : THE UNKNOWN SIGNED s.
%
% NOTICE : THE TRANSFER FUNCTION IS RATIONAL FUNCTION.
% AT THE NUMERATOR'S POLYNOMIAL, THE COEFFICIENT OF THE BIGGEST
EXPONENT OF
% THE UNKNOWN s FACTOR, HAS TO BE 1.
%
% IF THE TF IS  (44*s^2-242*s+380)/(s^3-10*s^2+39*s-64)
% YOU MUST WRITE
% (s^2-(242/44)^s+(380/44)/((1/44)*s^3-(10/44)*s^2+(39/44)*s-(64/44))
%
%
%      COMPENSATOR, by itself, creates a new COMPENSATOR or raises the
existing
%      singleton*.
%
```

```
%      H = COMPENSATOR returns the handle to a new COMPENSATOR or the
handle to
%      the existing singleton*.
%
%      COMPENSATOR('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in COMPENSATOR.M with the given input
arguments.
%
%      COMPENSATOR('Property','Value',...) creates a new COMPENSATOR
or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before compensator_OpeningFunction gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to compensator_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help compensator

% Last Modified by GUIDE v2.5 13-Oct-2009 18:54:50

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @compensator_OpeningFcn, ...
                   'gui_OutputFcn',  @compensator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before compensator is made visible.
function compensator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to compensator (see VARARGIN)
```

```
% Choose default command line output for compensator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes compensator wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = compensator_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

%όλα τα παραπάνω δηλώνονται από το ίδιο το GUI κατά το σχεδιασμό του προγράμματος

```
syms s
```

% δήλωση μεταβλητής s

% στη συνέχεια γίνεται ο ορισμός στο πρόγραμμα της συνάρτησης μεταφοράς που εισήγαγε ο χρήστης

```
function edit1_Callback(hObject, eventdata, handles)
tf=str2num(get(hObject,'string'));


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonQ1 και κάνοντας τη χρήση της συνάρτησης tf το πρόγραμμα εξάγει τη συνάρτηση Q1

```
% --- Executes on button press in pushbutton1.
function pushbuttonQ1_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε τη ονομασία tf στο στοιχεία που γράφτηκε στο edit1 έτσι ώστε να δηλωθεί στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθεί για την εύρεση και των υπολοίπων στοιχείων

```
tf=get(handles.edit1,'string');

[Q1]=calculationsQ1(tf);             % χρήση της συνάρτησης Q1

answer=char(Q1);                     % μετασχηματισμός του πίνακα Q1
βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

set(handles.editQ1,'string',answer)    % εκτύπωση του answer στο editQ1

function editQ1_Callback(hObject, eventdata, handles)
% hObject    handle to editGP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editGP as text
%        str2double(get(hObject,'String')) returns contents of editGP
as a double
% --- Executes during object creation, after setting all properties.
function editQ1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editGP (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonDegDC1 και κάνοντας τη χρήση της συνάρτησης tf το πρόγραμμα εξάγει τη συνάρτηση DegDC1

```
% --- Executes on button press in pushbuttonDeg1.
function pushbuttonDegDC1_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε τη ονομασία tf στο στοιχεία που γράφτηκε στο edit1 έτσι ώστε να δηλωθεί στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθεί για την εύρεση και των υπολοίπων στοιχείων

```
tf=get(handles.edit1,'string');
```

```
[DegDC1]=calculationsDegDC1(tf);
```
% χρήση της συνάρτησης DegDC1

```
set(handles.editDegDC1,'string',DegDC1)
```
% εκτύπωση του answer στο editDegDC1(δεν κάνουμε χρήση της συνάρτησης char γιατί η συνάρτηση DegDC1 δίνει ως αποτέλεσμα φυσικό αριθμό και όχι πολυώνυμο)

```
function editDegDC1_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
```

```
%        str2double(get(hObject,'String')) returns contents of edit6
as a double

% --- Executes during object creation, after setting all properties.
function editDegDC1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

% ορισμό του πίνακα DC1

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonPC1 και κάνοντας τη χρήση της συνάρτησης tf και τον πίνακα DCτο πρόγραμμα εξάγει τη συνάρτηση μεταφοράς του κλειστού συστήματος

```
% --- Executes on button press in pushbuttonPC1.
function pushbuttonPC1_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε τη ονομασία tf και DC στα στοιχεία που γράφτηκαν στα edit1, edit2 αντίστοιχα έτσι ώστε να δηλωθεί στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθεί για την εύρεση και των υπολοίπων στοιχείων

```
tf=get(handles.edit1,'string');
DC=str2num(get(handles.edit2,'string'));
```

```
[PC1]=calculationsPC1(tf,DC);
```
% χρήση της συνάρτησης PC1

```
answer=char(PC1);
```
% μετασχηματισμός του πίνακα PC1 βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

```
set(handles.editPC1,'string',answer)
```
% εκτύπωση του answer στο editPC1

```
function editPC1_Callback(hObject, eventdata, handles)
% hObject    handle to editPC1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editPC1 as text
%        str2double(get(hObject,'String')) returns contents of editPC1
as a double


% --- Executes during object creation, after setting all properties.
```

```
function editPC1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editPC1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

%μέσω του PushbuttonCom1 και κάνοντας τη χρήση της συνάρτησης tf και τον πίνακα DC το πρόγραμμα εξάγει τον αντισταθμηστή του συστήματος

```
% --- Executes on button press in pushbutton3.
function pushbuttonCom1_Callback(hObject, eventdata, handles)
```

% με τις παρακάτω εντολές δίνουμε τη ονομασία tf και DC στα στοιχεία που γράφτηκαν στα edit1, edit2 αντίστοιχα έτσι ώστε να δηλωθεί στο πρόγραμμα και στη συνέχεια να χρησιμοποιηθεί για την εύρεση και των υπολοίπων στοιχείων

```
tf=get(handles.edit1,'string');
DC=str2num(get(handles.edit2,'string'));

[Com1]=calculationsCom1(tf,DC);
```

% χρήση της συνάρτησης Com1

```
answer=char(Com1);
```

% μετασχηματισμός του πίνακα Com1 βάζοντας τα στοιχεία του πίνακα σε γραμμή κάνοντας τη χρήση της μεταβλητής s

```
set(handles.editCom1,'string',answer)
```

% εκτύπωση του answer στο editCom1

```
function editCom1_Callback(hObject, eventdata, handles)
% hObject    handle to editPC1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

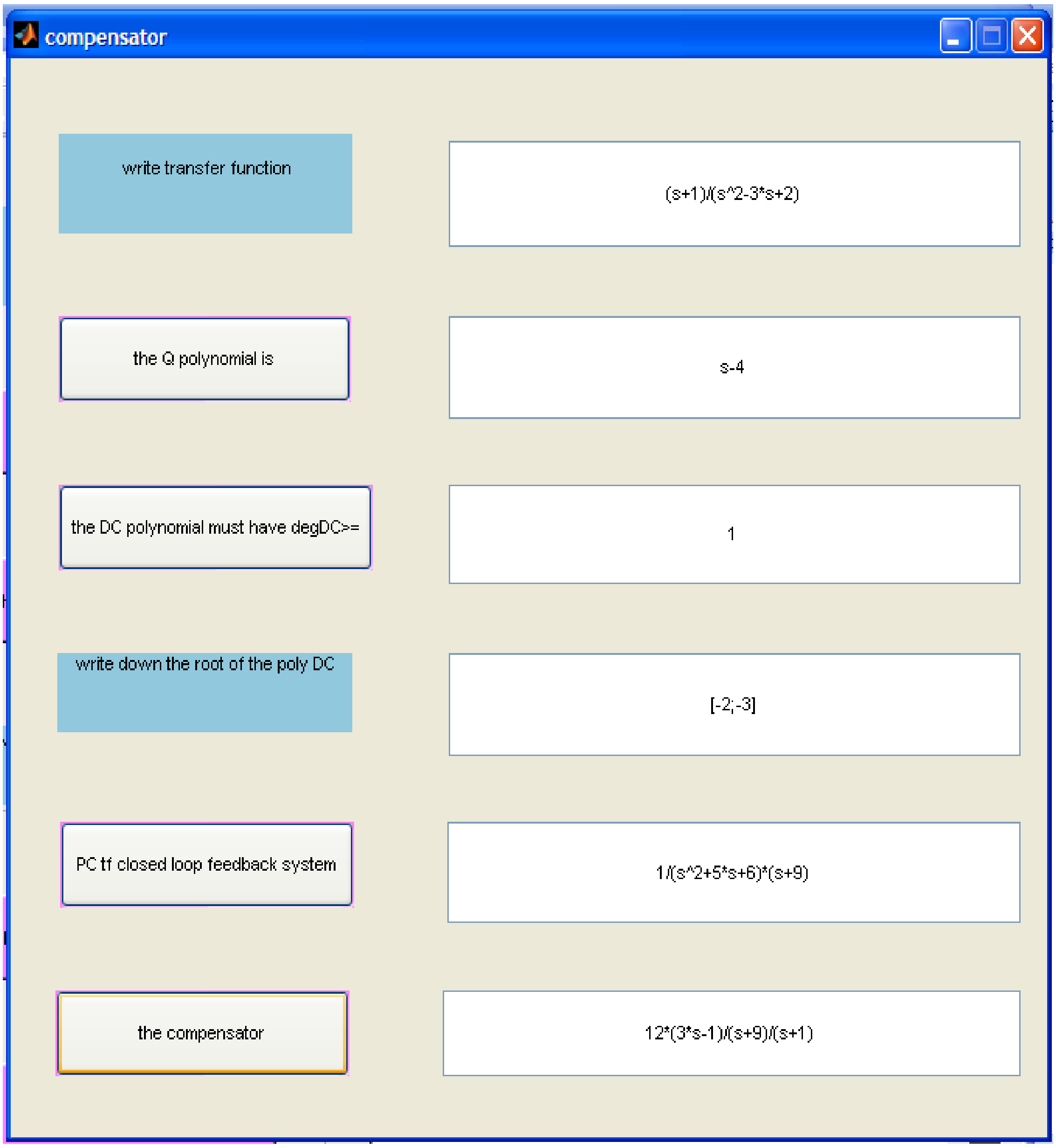
```
% Hints: get(hObject,'String') returns contents of editPC1 as text
%        str2double(get(hObject,'String')) returns contents of editPC1
as a double


% --- Executes during object creation, after setting all properties.
function editCom1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editPC1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

2.2.2.4. *Τρέξιμο Πρόγραμμα compensator*

Για να τρέξει το πρόγραμμα ο χρήστης πρέπει να πατήσει το βελάκι. Στη συνέχεια εμφανίζεται το παρακάτω παράθυρο (Σχήμα 17).



Σχήμα 17

Για το ορισμό των δεδομένων ο χρήστης θα πρέπει να ακολουθήσει τους παρακάτω κανόνες.

- ο ορισμός της συνάρτησης μεταφοράς γίνεται γράφοντας απλά το πολυώνυμο σε μορφή κλάσματος. Πολύ σημαντικό όμως είναι ότι στον αριθμητή ο συντελεστής του μεγαλύτερου εκθέτη του s πρέπει να είναι η μονάδα.

Παραδείγματος χάρη αν η συνάρτηση μεταφοράς είναι

(2*s^2+3*s+6)/(s^3+7*s^2+9*s-3)

θα πρέπει να γραφτεί

(s^2+(3/2)*s+(6/2))/((1/2)s^3+(7/2)*s^2+(9/2)*s-(3/2))

- ο πίνακας DC διάστασης kx1 όπου k αριθμός των ριζών που επιθυμούμε να έχει το πολυώνυμο DC : [-2;-3].

2.3. *Εφαρμογή*

Πρόγραμμα compensatorabcd

Δίνονται οι πίνακες

A= [-7 -2.25 0.75;4 0 0;0 1 0] είναι 3x3

B= [2;0;0] ,

C= [1 0.375 0.75]

και D= [0].

Το πρόγραμμα υπολογίζει την συνάρτηση μεταφοράς ίση με

(2*s^2+3*s+6)/(s^3+7*s^2+9*s-3)

το πολυώνυμο Q ίσο με

$$\frac{1}{2}\times s+\frac{11}{4}$$

οπότε το πολυώνυμο του οποίου πρέπει να δώσει ο χρήστης πρέπει να είναι βαθμού μεγαλύτερου του

$$1$$

έτσι εισάγοντας ο χρήστης του προγράμματος τον πίνακα DC

[-3;-3]

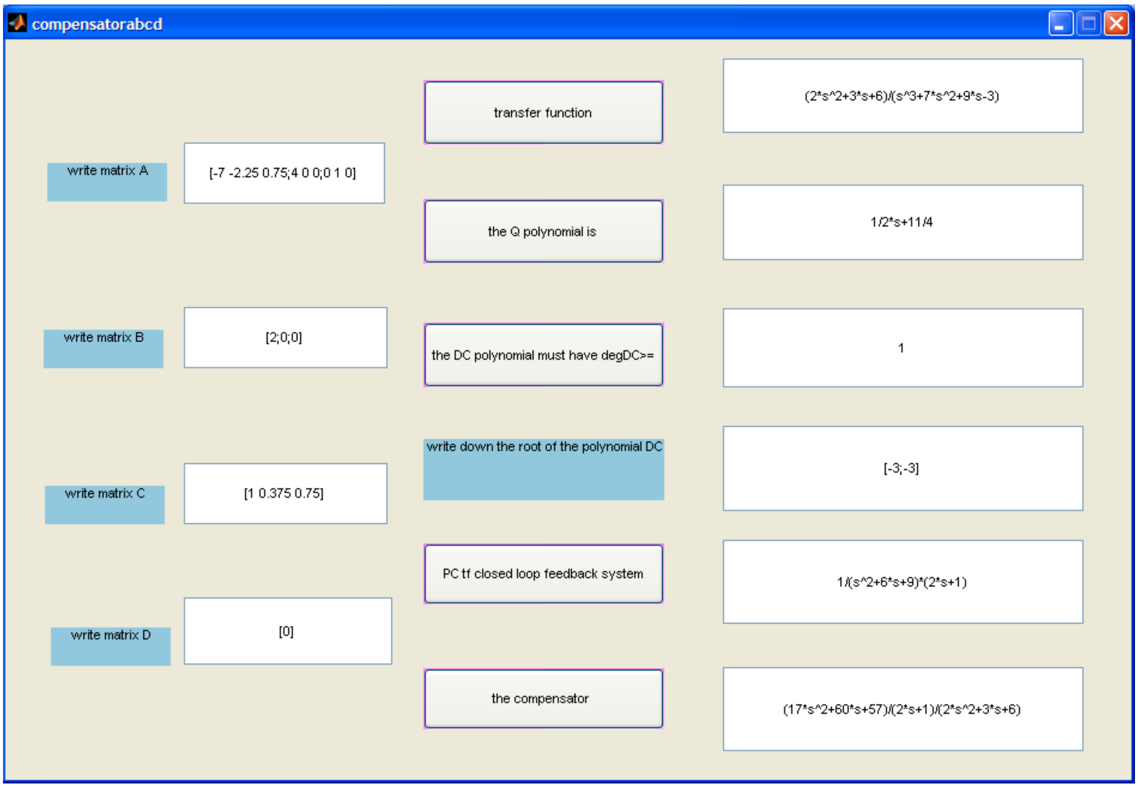
το πρόγραμμα υπολογίζει τη συνάρτηση μεταφοράς του κλειστού βρόχου ίση με

$$1/(s\,^\wedge 2+6*s+9)*(2*s+1)$$

και τον αντισταθμηστή ίσο με

$$(17*s\,^\wedge 2+60*s+57)/(2*s+1)/(2*s\,^\wedge 2+3*s+6)$$

Τα αποτελέσματα φαίνονται στο παρακάτω παράθυρο Σχήμα 18



Σχήμα 18

Πρόγραμμα compensator

Δίνεται η συνάρτηση μεταφοράς ίση με

(2*s^2+3*s+6)/(s^3+7*s^2+9*s-3)

Όπως έχει εξηγηθεί πρέπει να εισαχθεί με την παρακάτω μορφή

(s^2+(3/2)*s+(6/2))/((1/2)s^3+(7/2)*s^2+(9/2)*s-(3/2))

το πρόγραμμα υπολογίζει ότι το πολυώνυμο Q ίσο με

$$\frac{1}{2}\times s+\frac{11}{4}$$

οπότε το πολυώνυμο του οποίου πρέπει να δώσει ο χρήστης πρέπει να είναι βαθμού μεγαλύτερου του

$$1$$

εισάγοντας τον πίνακα DC

[-3;-3]

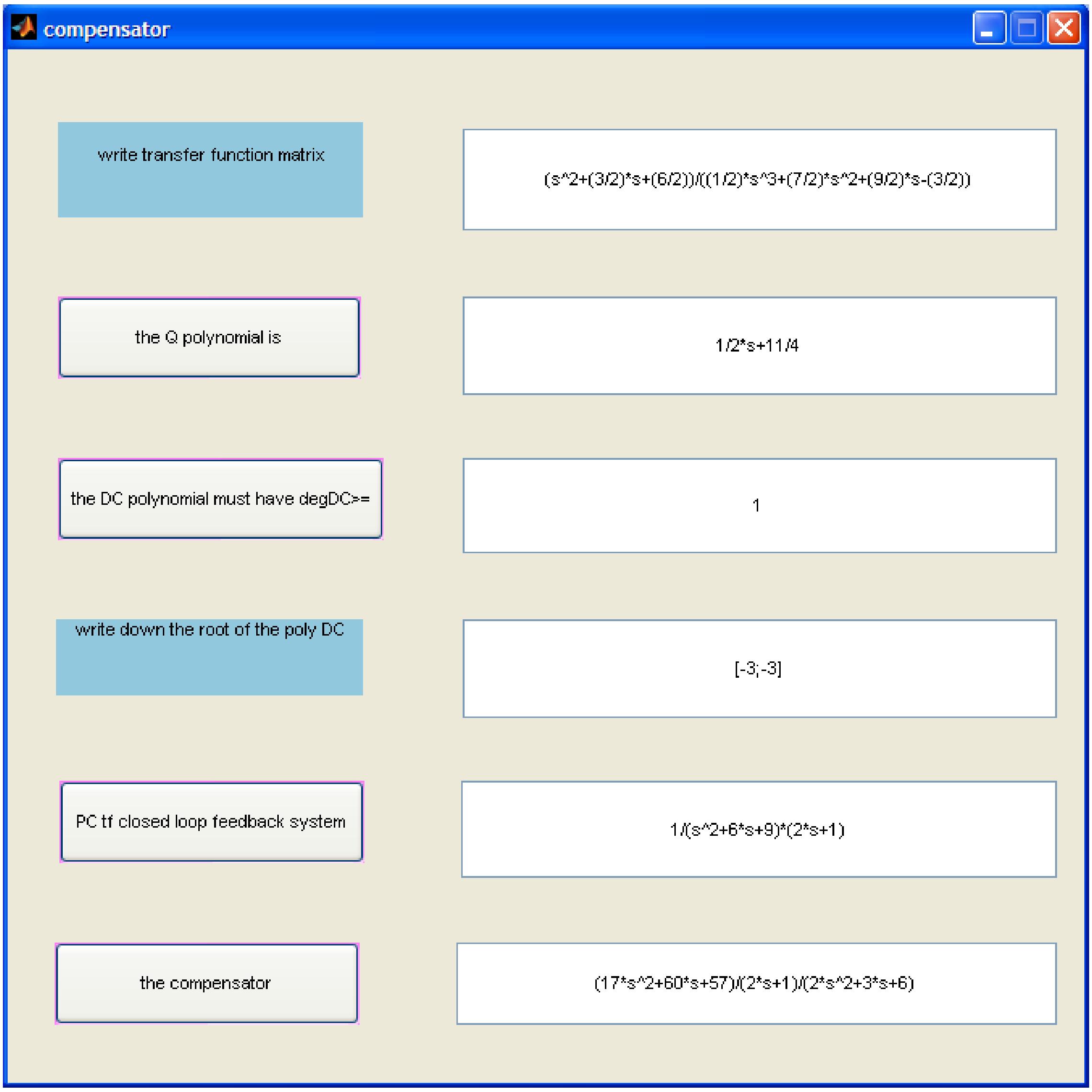
το πρόγραμμα υπολογίζει τη συνάρτηση μεταφοράς του κλειστού βρόχου ίση με

$$1/(s\wedge 2+6*s+9)*(2*s+1)$$

και τον αντισταθμηστή ίσο με

$$(17*s\wedge 2+60*s+57)/(2*s+1)/(2*s\wedge 2+3*s+6)$$

Τα αποτελέσματα φαίνονται στο παρακάτω παράθυρο Σχήμα 19.


compensator
write transfer function matrix
(s^2+(3/2)*s+(6/2))/((1/2)*s^3+(7/2)*s^2+(9/2)*s-(3/2))
the Q polynomial is
1/2*s+11/4
the DC polynomial must have degDC>=
1
write down the root of the poly DC
[-3;-3]
PC tf closed loop feedback system
1/(s^2+6*s+9)*(2*s+1)
the compensator
(17*s^2+60*s+57)/(2*s+1)/(2*s^2+3*s+6)

Σχήμα 19

<u>Βιβλιογραφία / Εργασίες / Σημειώσεις</u>

[1] Denominator Assignment, Invariants and Canonical forms under dynamic feedback Compensation in Linear Multivariable Systems, A.I.G. Vardulakis and C. Kazantzidou, in Proceedings of the 17^{th} Mediterranean Conference on Control and Automation (MED09), Thessaloniki, Greece, pp. 336-341.

[2] Linear Multivariable Control, Algebraic Analysis and Synthesis Methods, A. I. S. Vardulakis, Aristotle University of Thessaloniki, Greece, John Wiley & Sons, Chichester New York Brisbane Toronto Singapore, 1991

[3] Linear System, Thomas Kailath, Department of Electrical Engineering Stanford University, Prentice Hall, 07632, 1980

[4] Matlab για Μηχανικούς 5, Adrian Bira&Moshe Breiner Εκδόσεις Τζιόλας, 1999

[5] Matlab για Μηχανικούς 7, Ευάγγελος Β. Χατζίκος, Εκδόσεις Τζιόλας, 2007

[6] Modeling Analysis and Control of Dynamic Systems, William J.Palm, University of Rhode Island 2^{nd} Edition, John Wiley & Sons, Inc, New York, 2000

[7] Modern Control design with Matlab and Simuling, Ashish Tewari, Indian Institute of Technology, Kanpur, India, John Wiley 8 Sons, Ltd 2002

[8] Γραμμική Άλγεβρα , Κ. Λάκκη, Τρίτη έκδοση, Θεσσαλονίκη 1993

[9] Σημειώσεις μαθήματος Κλασική Θεωρία Ελέγχου, Α.Ι. Βαρδουλάκης

[10] Σημειώσεις μαθήματος Μοντέρνα Θεωρία Ελέγχου, Α.Ι. Βαρδουλάκης

[11] Σημειώσεις μαθήματος Πολυμεταβλητά Συστήματα, Α.Ι. Βαρδουλάκης

[12] Σύγχρονα συστήματα αυτομάτου ελέγχου, Richard C. Dorf Robert H. Bishop, Τζιόλα Εκδόσεις, $9^{η}$ Έκδοση, Θεσσαλονίκη, 2008

[13] Το πρόγραμμα MATLAB