



*ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ*

Συστήματα Αυτόματου Ελέγχου

*Ανάπτυξη Σημειώσεων για τις Ασκήσεις Πράξης
Σ.Α.Ε. Ι*

*Υπεύθυνος καθηγητής : Αριστοτέλης Γιαννακουδάκης
Πτυχιακή εργασία : Γιώργου Κάκκαρη
Αριθμός Μητρώου: 3494*

Πρόλογος

Η έκδοση των σημειώσεων με τίτλο «Συστήματα Αυτόματου Ελέγχου» απευθύνονται στους σπουδαστές του τμήματος Ηλεκτρολογίας του Τ.Ε.Ι. Ηρακλείου ως συμπληρωματικές για την κατανόηση του ομότιτλου μαθήματος Συστήματα Αυτόματου Ελέγχου.

Σκοπός των σημειώσεων αυτών είναι να εισάγει τον νέο σπουδαστή στις έννοιες των συστημάτων αυτόματου ελέγχου και να τον βοηθήσει στην κατανόηση, στη μελέτη και την επίλυση προβλημάτων που έχουν σχέση με αυτά.

Περιεχόμενα

Εισαγωγή.....	5
Ιστορική Αναδρομή	7
Μάθημα	13
1.Εισαγωγή στο MatLab	14
1.1 Δηλώσεις και Μεταβλητές.....	16
1.2 Πίνακες και Διανύσματα	21
1.3 Γραφικές Παραστάσεις και Διαγράμματα	26
1.4 Πολύωνυμα	30
1.5 Αρχεία εντολών και Αρχεία συναρτήσεων	33
1.6 Δομές Ελέγχου ροής.....	34
1.6.1 Εντολή if.....	35
1.6.2 Εντολή for.....	36
1.6.3 Εντολή while	36
1.7 Βασικές συναρτήσεις MatLab	38
2. Εισαγωγή στο Simulink	39
3. Μαθηματικά Μοντέλα.....	47
3.1 Μετασχηματισμός Laplace	47
3.2 Αντίστροφος Μετασχηματισμός Laplace	50
3.3 Μετασχηματισμός Laplace - MatLab	53
3.4 Ασκήσεις Πράξης Μετασχηματισμού Laplace.....	54
4. Βασικά Σήματα Συστημάτων Αυτόματου Ελέγχου	65
4.1 Βηματική(step) συνάρτηση.....	65
4.2 Κρουστική συνάρτηση	67
4.3 Συνάρτηση Αναρρίχησης	67
4.4 Εκθετική Συνάρτηση	69
4.5 Ημιτονοειδής Συνάρτηση.....	69
4.6 Βασικά Σήματα-MatLab	69
4.7 Ασκήσεις Πράξης Βασικών Σημάτων	70
5.Ανάλυση Συστημάτων Αυτόματου Ελέγχου.....	89
5.1 Συνάρτηση Μεταφοράς.....	89
5.2 Δομικά Διαγράμματα Συστημάτων	94

5.3 Διασύνδεση Συστημάτων	95
5.4 Συστήματα Αυτόματου ελέγχου ανοιχτού και κλειστού βρόγχου	99
5.5 Ασκήσεις Πράξης Ανάλυσης Συστημάτων Αυτόματου Ελέγχου	103
6. Χρονική απόκριση Συστημάτων Αυτόματου Ελέγχου	109
6.1 Τυποποιημένα Σήματα Εισόδου.....	109
6.2 Χαρακτηριστικά της γραφικής παράστασης της χρονικής απόκρισης.....	111
6.3 Συστήματα πρώτης τάξης	112
6.3.1 Βηματική απόκριση.....	113
6.3.2 Κρουστική απόκριση	114
6.3.3 Απόκριση Γενικά	115
6.4 Συστήματα δεύτερης τάξης	116
6.5 Σφάλματα Συστημάτων Αυτόματου Ελέγχου.....	118
6.6 Σταθερές Σφάλματος μόνιμης κατάστασης.....	119
6.6.1 Σταθερά σφάλματος θέσης K_p	120
6.6.2 Σταθερά σφάλματος ταχύτητας K_v	120
6.6.3 Σταθερά σφάλματος επιτάχυνσης K_a	120
6.7 Ασκήσεις Πράξης Χρονικής απόκρισης Συστημάτων Αυτόματου Ελέγχου	121
7. Ευστάθεια Συστημάτων Αυτόματου Ελέγχου	131
8. Ελεγκτές Συστημάτων Αυτόματου Ελέγχου	135
8.1 Αναλογικός (P) ελεγκτής	135
8.2 Ολοκληρωτικός (I) ελεγκτής.....	137
8.3 Διαφορικός (D) ελεγκτής	138
8.4 Αναλογικός – Ολοκληρωτικός (PI) ελεγκτής	138
8.5 Αναλογικός – Διαφορικός (PD) ελεγκτής	139
8.6 Αναλογικός – Ολοκληρωτικός – Διαφορικός (PID) ελεγκτής	141
8.7 Ασκήσεις Πράξης ελεγκτών Συστημάτων Αυτόματου Ελέγχου.....	142
9. Ασκήσεις Πράξης Συστημάτων Αυτόματου Ελέγχου	153
9.1 Φίλτρα.....	153
9.2 DC Κινητήρας.....	163
Βιβλιογραφία.....	174

Εισαγωγή

Τις έννοιες ενός σήματος και ενός συστήματος τις συναντάμε σε πολλούς τομείς των τεχνολογικών και εφαρμοσμένων επιστημών. Η χρήση των εννοιών αυτών από τους ερευνητές κατά τα τελευταία 50 περίπου χρόνια βοήθησε κατ' αρχή στην μαθηματική διατύπωση ερωτημάτων τα οποία προέκυπταν από την προσπάθεια για καλύτερη και βαθύτερη κατανόηση πολλών φυσικών, μηχανικών ή οικονομικών φαινομένων και διαδικασιών και στην συνέχεια στην διερεύνηση αντιστοίχων προβλημάτων.

Οι έννοιες ενός σήματος και ενός συστήματος έχουν βασικές κοινές ιδιότητες. Αυτό που ονομάζουμε σήμα αποτελεί πάντα μία μαθηματική συνάρτηση μίας ή περισσότερων ανεξαρτήτων μεταβλητών μία από τις οποίες είναι υποχρεωτικά ο χρόνος και τυπικά περιέχει πληροφορίες για τη χρονική εξέλιξη μιας ποσότητας η οποία περιγράφει ένα φαινόμενο ή μία διαδικασία. Ο ακριβής ορισμός της έννοιας του συστήματος είναι πιο δύσκολος. Ένα σύστημα αναγνωρίζεται πιο εύκολα από ότι ορίζεται. Με τον όρο σύστημα εννοούμε ένα μέρος του φυσικού κόσμου το οποίο θεωρούμε ότι αποτελείται από ένα σύνολο στοιχείων τα οποία λειτουργούν συγχρόνως κατά προδιαγεγραμμένο τρόπο έτσι ώστε να επιτυγχάνεται κάποιος στόχος. Ένα σύστημα επικοινωνεί με το περιβάλλον μέσω σημάτων. Τα σήματα που δέχεται ένα σύστημα ονομάζονται διεγέρσεις ή είσοδοι και τα σήματα που παράγει ένα σύστημα λόγω των διεγέρσεων και των μη μηδενικών αρχικών συνθηκών ονομάζονται αποκρίσεις ή έξοδοι (Δρ. Βολογιαννίδης Σταύρος, «Συστήματα Αυτόματου Ελέγχου Θεωρία και Εφαρμογές-Διδακτικές σημειώσεις Τμήματος Πληροφορικής και Επικοινωνιών Τ.Ε.Ι. Σερρών»).

Τα Συστήματα Αυτόματου Ελέγχου (Σ.Α.Ε.) είναι σήμερα μία από τις σημαντικότερες περιοχές της επιστήμης και της τεχνολογίας, κάτι που οφείλεται στο γεγονός ότι ο αυτοματισμός είναι συνυφασμένος με την ανάπτυξη σχεδόν κάθε μορφής τεχνολογίας.

Οι τηλεπικοινωνίες, η ηλεκτρονική, η παραγωγή και κατανομή ηλεκτρικής ενέργειας, η ρομποτική, η αεροναυτική και αστροναυτική, η οικονομία, και οικονομετρία, και ακόμη η νευρολογία, η βιολογία και η ιατρική είναι μερικά μόνο παραδείγματα επιστημονικών περιοχών για τις οποίες τα Σ.Α.Ε. έπαιξαν και παίζουν συνεχώς πολύ σημαντικό ρόλο στην διατύπωση, ανάλυση, διερεύνηση και λύση προβλημάτων τα οποία τις απασχολούν.

Συνοψίζοντας τα παραπάνω, σύστημα αυτόματου ελέγχου είναι ένα σύστημα που τα διάφορα μέρη του είναι συνδεδεμένα μεταξύ τους με τέτοιο τρόπο ώστε να συμπεριφέρονται αυτόματα κατά ένα προκαθορισμένο επιθυμητό τρόπο (Πηγή: Π.Ν. Παρασκευόπουλος, «Εισαγωγή στον Αυτόματο Έλεγχο – Τόμος Α: Θεωρία», Αθήνα 2001).

Το σύστημα λοιπόν αποτελείται από τρία στοιχεία :

- Είσοδος
- Σύστημα
- Έξοδος

Αν συμβολίσουμε το σύστημα με το γράμμα «Σ» τότε η συμπεριφορά του συστήματος (δηλαδή η έξοδος ή η απόκριση αυτού) συνδέεται με την είσοδο με μία μαθηματική σχέση :

$$y(t) = \Sigma * u(t)$$

Όπου

$u(t)$: η είσοδος του συστήματος συναρτήσει του χρόνου και ,
 $y(t)$: η έξοδος του συστήματος συναρτήσει του χρόνου .

Σ' ένα Σ.Α.Ε. είναι πιθανό να έχουμε περισσότερες από μία εισόδους, οι εισοδοί ονομάζονται και παράμετροι ενώ η έξοδος ονομάζεται και προδιαγραφές. Οι εισοδοί (παράμετροι) μπορεί να είναι δύο ειδών είτε ελέγχου είτε διαταραχής, δηλαδή υπάρχουν εισοδοί τις οποίες μπορούμε να ελέγξουμε(ελέγχου) και εισοδοί τις οποίες δεν μπορούμε να ελέγξουμε(διαταραχής). Σε αντίθεση με τις παραμέτρους που τις ελέγχουμε ή δεν τις ελέγχουμε τις παραμέτρους μπορούμε να τις ορίσουμε. Αν το επιθυμούμε μπορούμε να επεξεργαστούμε την έξοδο του συστήματος και να δούμε αν ταυτίζεται με την προδιαγραφή που έχουμε ορίσει, σε περίπτωση που δεν ισχύει αυτό τότε πρέπει να δράσουμε καταλλήλως στις παραμέτρους ώστε να πετύχουμε έξοδο ίδια με τις προδιαγραφές. Για να γίνουν περισσότερο αντιληπτά τα παραπάνω θα αναφέρουμε ένα απλό παράδειγμα.

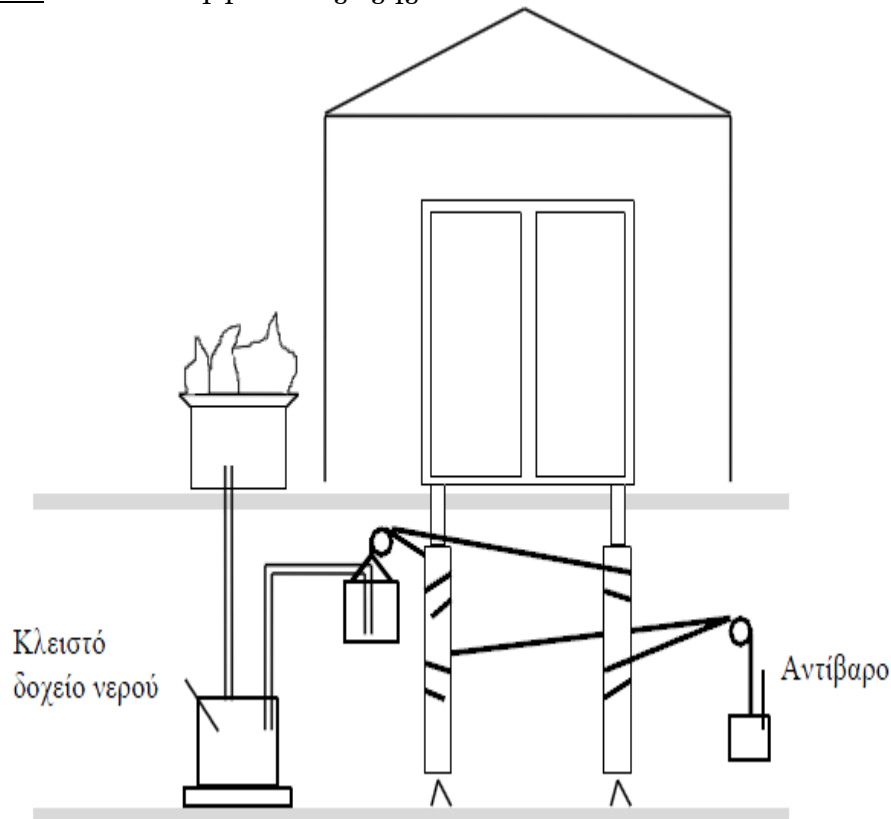
Θεωρούμε ότι υπάρχει ένα σπίτι στο οποίο θέλουμε να ελέγξουμε τη θερμοκρασία. Αρχικά ορίζουμε στον θερμοστάτη τη θερμοκρασία την οποία θέλουμε δηλαδή ορίζουμε την προδιαγραφή. Στη συνέχεια για να λειτουργήσει ο καυστήρας εισάγουμε το πετρέλαιο που αποτελεί είσοδο ελέγχου, όμως δεν μπορούμε να ελέγξουμε την εξωτερική θερμοκρασία που επηρεάζει την θερμοκρασία στο εσωτερικό του σπιτιού, άρα η εξωτερική θερμοκρασία είναι είσοδος διαταραχής. Τέλος ο καυστήρας θα σταματήσει να λειτουργεί όταν ο θερμοστάτης μετρήσει στο εσωτερικό του σπιτιού τη θερμοκρασία που έχουμε ορίσει δηλαδή όταν η έξοδος του συστήματος ταυτιστεί με την προδιαγραφή.

Τα συστήματα που θα μας απασχολήσουν στο μάθημα των Συστημάτων Αυτόματου Ελέγχου είναι :

- Σταθερά, συστήματα που είναι αμετάβλητα σε σχέση με το χρόνο.
- Γραμμικά, συστήματα που έχουν την ιδιότητα κάθε γραμμικός συνδυασμός εισόδων να αντιστοιχεί σε ανάλογο γραμμικό συνδυασμό εξόδων.

Ιστορική Αναδρομή

Ένα από τα πρώτα συστήματα αυτομάτου ελέγχου που αναφέρονται στην ιστορία είναι ο μηχανισμός που επινόησε ο Ήρωνας ο Αλεξανδρινός για το αυτόματο άνοιγμα των θυρών ενός αρχαίου ναού. Η διάταξη φαίνεται στην Εικόνα 1 και λειτουργούσε ως εξής:

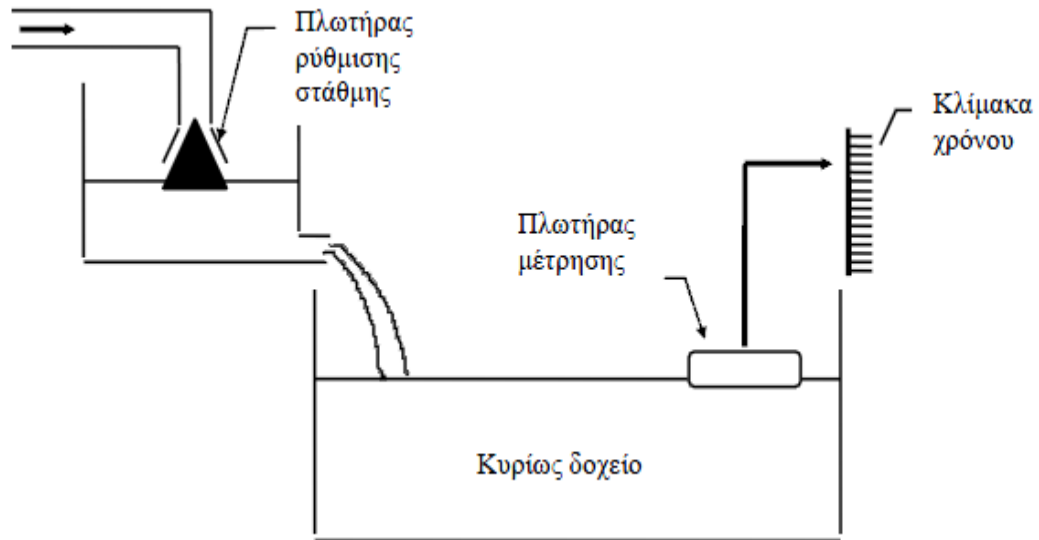


Εικόνα 1

Με το άναμμα της φωτιάς στο βωμό ο αέρας κάτω απ' αυτόν θερμαίνεται και διαστέλλεται και πιέζει το νερό από το κλειστό δοχείο να ανέβει στο δοχείο. Το δοχείο γίνεται βαρύτερο και κατέρχεται ανοίγοντας τις θύρες με τη βοήθεια σχοινιών, παρασύρει δε και ανυψώνει το αντίβαρο. Η διαδικασία εξελίσσεται αντίστροφα όταν σβήσει η φωτιά. Ο αέρας κρυώνει, η πίεση του κλειστού δοχείου μικραίνει και συνεπώς το νερό κυλάει από το δοχείο στο κλειστό δοχείο λόγω βαρύτητας. Το δοχείο λοιπόν γίνεται ελαφρύτερο οπότε κάποια στιγμή το αντίβαρο αρχίζει να κατεβαίνει κλείνοντας τις πόρτες.

Είναι πιθανόν ότι ο όλος μηχανισμός ενεργοποιείται όταν οι ιερείς αρχίζουν να ανεβαίνουν τα σκαλοπάτια του ναού. Η διάταξη βέβαια δεν ήταν ορατή στους κοινούς θνητούς και έτσι το άνοιγμα αποδιδόταν στους Θεούς! (Πηγές: 1. Π.Ν. Παρασκευόπουλος, «Εισαγωγή στον Αυτόματο Έλεγχο – Τόμος Α: Θεωρία», Αθήνα 2001. 2. Singer et al : «A history of Technology» vol.2. Oxford University Press, Fair Lawn N.J. 1956).

Το πρώτο σύστημα ελέγχου με ανάδραση θεωρείται το υδάτινο ρολόι του Κτησιβίου που κατασκευάστηκε στην Αλεξάνδρεια τον 3^ο π.Χ. αιώνα.

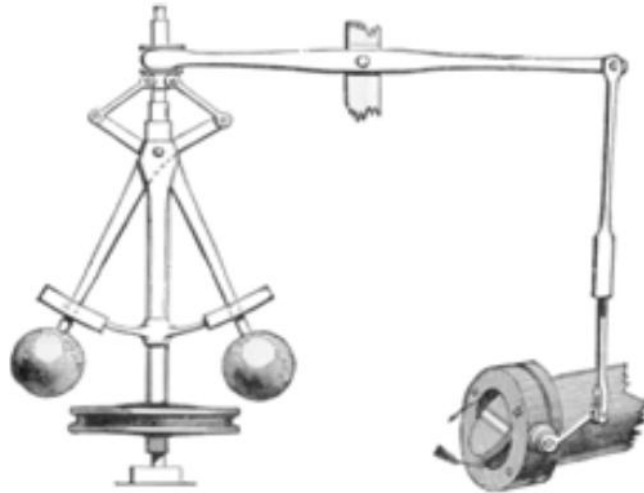


Εικόνα 2

Στη διάταξη αυτή, που φαίνεται στην Εικόνα 2 ένα κύριο δοχείο χρησιμεύει για την συγκέντρωση νερού. Αν η ροή προς τη δεξαμενή διατηρείται σταθερή τότε η στάθμη της - κατάλληλα βαθμονομημένη - μπορεί να χρησιμοποιηθεί για την μέτρηση του χρόνου. Προκειμένου να διατηρήσει τη ροή προς την δεξαμενή σταθερή, ο Κτησίβιος χρησιμοποίησε ένα δεύτερο δοχείο ρύθμισης. Του δοχείου αυτού έλεγχε την στάθμη με την βοήθεια ενός κωνικού πλωτήρα: Όταν η στάθμη κατέρχεται ο πλωτήρας κατερχόμενος αποκάλυπτει περισσότερο το στόμιο και επιτρέπει να περάσει περισσότερο νερό, άρα η στάθμη ξανανεβαίνει. Η σταθερή στάθμη του νερού στο δευτερεύον δοχείο επιβάλλει σταθερή ροή προς το κύριο (Η ροή από σταθερό στόμιο δοχείου εξαρτάται μόνο από τη στάθμη του νερού σ' αυτό). Ένα μάλλον ογκώδες ρολόι ιδιαίτερα αν συγκριθεί με τα σύγχρονα ηλεκτρονικά!

Στο ρολόι του Κτησίβιου συναντάμε την αρχή της ανάδρασης: Η ελεγχόμενη ποσότητα (στάθμη του βοηθητικού δοχείου) ανατροφοδοτείται (επιηρεάζει) στην είσοδο του συστήματος που είναι η ροή του νερού προς αυτό (Πηγές: 1.Μ. Σφακιωτάκης, «Συστήματα Αυτόματου Ελέγχου II-Διαλέξεις Θεωρίας». 2. Π.Ν. Παρασκευόπουλος, «Εισαγωγή στον Αυτόματο Έλεγχο – Τόμος Α: Θεωρία», Αθήνα 2001).

Κατά καιρούς εμφανίστηκαν και άλλα πρωτόγονα συστήματα αυτόματου ελέγχου. Το πρώτο όμως που βρήκε (και βρίσκει) ευρεία χρήση, θεωρείται ο φυγοκεντρικός ρυθμιστής ταχύτητας που εφευρέθηκε το 1769 από τον James Watt για τον έλεγχο ταχύτητας των ατμομηχανών .



Εικόνα 3

Ο μηχανισμός, που φαίνεται στην Εικόνα 3, παρακολουθεί βασικά την περιστροφική ταχύτητα της μηχανής με τη βοήθεια στρεφομένων (από τον άξονα της μηχανής) σφαιριδίων. Μείωση της ταχύτητας περιστροφής της μηχανής -λόγω π.χ. αυξημένου φορτίου- συνεπάγεται πτώση των σφαιριδίων που μέσω του υδραυλικού μηχανισμού οδηγεί στο άνοιγμα της βαλβίδας του ατμού. Όσο αυξανόταν η ταχύτητα περιστροφής η σφαίρα εξαιτίας της φυγόκεντρης δύναμης ανασηκωνόταν και η κίνηση της απομακρυνόταν από τον άξονα συμμετρίας του κυλινδρικού άξονα κίνησης προκαλώντας με τον τρόπο αυτό το κλείσιμο της βάνας. Ακριβώς το αντίστροφο συμβαίνει όταν η μηχανή τείνει να αυξήσει τις στροφές της. Για την κίνηση της σφαίρας απαιτείται ένα ποσό ενέργειας που προσφέρεται από την ίδια την μηχανή, οπότε η μέτρηση της περιστροφικής ταχύτητας διεξάγεται με σχετικά μικρή ακρίβεια.

Ο ρυθμιστής του Watt εξελιγμένος φυσικά χρησιμοποιείται ακόμη και σήμερα για την ρύθμιση στροφών μοντέρνων στροβιλομηχανών. Το σύστημα είναι ενδιαφέρον από άποψη ευστάθειας, και μόνο το 1868 ο Maxwell το μελέτησε αναλυτικά (Πηγές: 1.Μ. Σφακιωτάκης, «Συστήματα Αυτόματου Ελέγχου II-Διαλέξεις Θεωρίας». 2.Μ. Καβουσιανός «Συστήματα Αυτόματου Ελέγχου»).

Η χρονική περίοδος πριν το 1868 χαρακτηρίστηκε γενικά από την ανάπτυξη διαφόρων εμπνευσμένων και έξυπνων συστημάτων αυτόματου ελέγχου. Οι διάφορες προσπάθειες με σκοπό την αύξηση της ακρίβειας των συστημάτων οδήγησαν στην επιτακτική ανάγκη της ανάπτυξης μιας αντίστοιχης θεωρίας για τον αυτόματο έλεγχο. Ο J.C. Maxwell διετύπωσε μια θεωρία σχετικά με τον αυτόματο έλεγχο, βασισμένη σε ένα μαθηματικό μοντέλο διαφορικής εξίσωσης ενός ελεγκτή. Η μελέτη αυτή του Maxwell αφορούσε κυρίως στην επίδραση των διαφόρων παραμέτρων του συστήματος πάνω στη συνολική του συμπεριφορά. Κατά την ίδια περίοδο ο I.A. Vyshnegradskii διετύπωσε μια αντίστοιχη θεωρία βασισμένη στα μαθηματικά, σχετικά με τα συστήματα ρυθμιστών.

Πριν από τον Δεύτερο Παγκόσμιο Πόλεμο, οι διάφορες θεωρίες που αναπτύχθηκαν στις Η.Π.Α. και στην Δυτική Ευρώπη είχαν διατυπωθεί με αρκετά διαφορετικό τρόπο σε σχέση με εκείνες που αναπτύχθηκαν από την

πλευρά της Ανατολικής Ευρώπης και της Ρωσίας. Την ώθηση για την χρήση συστημάτων ανάδρασης στις Η.Π.Α. αποτέλεσε η ανάπτυξη της τηλεφωνίας και των ηλεκτρονικών ενισχυτών ανάδρασης από τους Bode, Nyquist και Black της εταιρίας Bell Telephone Laboratories. Η έννοια του πεδίου της συχνότητας χρησιμοποιήθηκε πρωταρχικά με σκοπό να περιγράψει την λειτουργία των ενισχυτών ανάδρασης σε σχέση με το εύρος ζώνης καθώς επίσης και με άλλες αντίστοιχες μεταβλητές. Σε αντίθεση με όλα αυτά διάφοροι Μαθηματικοί αλλά και διάφοροι εμπειρικοί μηχανικοί στην πρώην Σοβιετική Ένωση ενέπνευσαν και κυριάρχησαν στον χώρο της θεωρίας του αυτόματου ελέγχου. Η θεωρητική προσέγγιση των Σοβιετικών διακατέχεται από μια τάση προς το πεδίο του χρόνου διατυπωμένη κατάλληλα με τη βοήθεια διαφορικών εξισώσεων (Πηγή: Richard C. Dorf, Robert H. Bishop, «Σύγχρονα Συστήματα Αυτόματου Ελέγχου», 9^η Έκδοση, Εκδόσεις Τζιόλα 2003).

Ισχυρή ώθηση δόθηκε στον Αυτόματο Έλεγχο κατά τη διάρκεια του 2ου Παγκοσμίου Πολέμου. Τότε η επιτακτική ανάγκη για τους εμπόλεμους να κατασκευάζουν συνεχώς τελειότερα όπλα, οδήγησε στην κατασκευή αυτομάτων συστημάτων σκόπευσης πυροβόλων, συστημάτων αυτόματης κατεύθυνσης κεραιών radar, συστημάτων αυτόματης πλοήγησης αεροσκαφών και υποβρυχίων. Δυστυχώς για μια ακόμη φορά ο Ηράκλειτος επαληθεύθηκε : «πόλεμος πατήρ πάντων...». Πριν από το 1940 οι διαδικασίες σχεδίασης των συστημάτων αυτόματου ελέγχου βρισκόταν ως επί το πλείστον σε επίπεδο εμπειρικών μεθόδων βασισμένων σε τεχνικές δοκιμής και σφάλματος. Στην διάρκεια της δεκαετίας αυτής διατυπώνονταν με γοργούς ρυθμούς διάφορες μαθηματικές και αναλυτικές μέθοδοι και προοδευτικά το αντικείμενο του αυτόματου ελέγχου μετατράπηκε σε επίπεδο ιδιαίτερης εφαρμοσμένης επιστήμης.

Οι διάφορες τεχνικές ανάλυσης στο πεδίο της συχνότητας συνέχισαν να κυριαρχούν στον χώρο του αυτόματου ελέγχου και μετά τον 2^ο Παγκόσμιο Πόλεμο με ιδιαίτερη έμφαση στην χρήση των μετασχηματισμών Laplace καθώς και του πεδίου της μιγαδικής συχνότητας. Κατά τη διάρκεια της δεκαετίας του 1950 δόθηκε αρκετή έμφαση στην ανάπτυξη και την χρήση μεθόδων του πεδίου της μιγαδικής μεταβλητής s και ιδιαίτερα της μεθόδου του γεωμετρικού τόπου ριζών. Στη δεκαετία '50-'60 εμφανίστηκαν οι αναλογικοί και στη συνέχεια οι ψηφιακοί υπολογιστές. Τα εργαλεία αυτά έχοντας την ικανότητα να μετρούν μεταβλητές και να εκτελούν ταχύτατα υπολογισμούς, έδωσαν τεράστια ώθηση στον αυτόματο έλεγχο. Τη δεκαετία του 1980 η χρήση των ψηφιακών υπολογιστών και των αντίστοιχων σχετικών διεργασιών ως στοιχεία ελέγχου γίνεται ρουτίνα. Η τεχνολογία των σύγχρονων για την εποχή εκείνη συστημάτων ήταν απαγορευτική για το σύνολο των μηχανικών σχεδίασης.

Σήμερα...

Με την αποστολή του διαστημοπλοίου Sputnik και την εμφάνιση της εποχής του διαστήματος δόθηκε μια νέα μεγάλη ώθηση στον χώρο του αυτόματου ελέγχου, έτσι δημιουργήθηκε μια νέα ανάγκη σχεδιασμού πολύπλοκων συστημάτων αυτόματου ελέγχου υψηλής ακρίβειας για πυραύλους και διατάξεις ανιχνευτών για διαστημική χρήση. Επίσης η ανάγκη ελέγχου υψηλής ακρίβειας έφερε στο προσκήνιο το αντικείμενο του βέλτιστου

ελέγχου. Αποτέλεσμα των παραπάνω απαιτήσεων τα τελευταία χρόνια αναπτύχθηκαν μέθοδοι στο πεδίο του χρόνου από τους Liapunov, Minorsky και άλλους όπως και διάφορες μελέτες σχετικά με την τεχνολογία των εύρωστων συστημάτων. Είναι ξεκάθαρο ότι τόσο η ανάλυση στο πεδίο του χρόνου, όσο και στο πεδίο της συχνότητας θα πρέπει να λαμβάνονται εξίσου υπόψη κατά τις διαδικασίες ανάλυσης και σχεδίασης συστημάτων αυτόματου ελέγχου.

Το αυτόματο πλυντήριο, η αυτόματη κουζίνα, η αυτοματοποιημένη κεντρική θέρμανση είναι μόνο λίγα από τα παραδείγματα αυτοματοποιημένων συστημάτων που βρίσκονται ήδη στη διάθεση του ανθρώπου.

Σήμερα οι ψηφιακοί υπολογιστές έχουν ολοκληρωτικά επικρατήσει των αναλογικών. Εκατομμύρια απ' αυτούς είναι εγκατεστημένοι στη βιομηχανία όπου ελέγχουν διεργασίες παρακολουθώντας και ελέγχοντας πλήθος μεταβλητών. Ο αυτοματισμός είναι πολύ διαδεδομένος στον τεχνικό κόσμο και δεν είναι υπερβολή να πούμε ότι σ' ένα μεγάλο βαθμό η αυτοματοποίηση με τον ένα ή τον άλλο τρόπο έχει εισβάλλει στη ζωή μας και στο μέλλον θα κυριαρχεί όλο και περισσότερο. Αυτό οφείλεται στο γεγονός ότι ένας από τους κύριους σκοπούς της σύγχρονης τεχνολογίας είναι η κατά το δυνατό μείωση της συμμετοχής του ανθρώπου στη λειτουργία των συστημάτων. Η αυτοματοποίηση στις γραμμές παραγωγής των εργοστασίων συντελεί στη συνεχή μείωση του κόστους των καταναλωτικών προϊόντων με παράλληλη βελτίωση της ποιότητάς τους. Αυτοκίνητα, ηλεκτρικές-ηλεκτρονικές συσκευές, υπολογιστές παράγονται σε πλήρως αυτοματοποιημένες γραμμές και κατακλύζουν τις αγορές του κόσμου. Τα τελευταία χρόνια τα συστήματα αυτόματου ελέγχου παρουσιάζουν μία εντυπωσιακή εξέλιξη που οφείλεται στους ηλεκτρονικούς υπολογιστές. Εξαιτίας της γρήγορης ανάπτυξης και κυρίως στο χαμηλό κόστος των υπολογιστών χρησιμοποιούνται ολοένα και περισσότερο ώστε να αυτοματοποιηθεί ένα σύστημα ή μια διεργασία, κάτι που μπορεί να απαλλάξει τον άνθρωπο από μονότονες και δύσκολες εργασίες, ήδη τα βιομηχανικά ρομπότ αναλαμβάνουν ακούραστα και αδιαμαρτύρητα να εκτελέσουν πολλές τέτοιες εργασίες. Υπάρχει πάντως και ο φόβος ότι η πλατιά εξάπλωση των συστημάτων αυτόματου ελέγχου στους χώρους δουλειάς οδηγεί στην ανεργία, αφού χάνονται συνεχώς θέσεις εργασίας ανειδίκευτου προσωπικού. Αναμφισβήτητα πολλά επαγγέλματα περνούν κρίση, η λύση είναι η συνεχής μετεκπαίδευση στις νέες τεχνολογίες. Παρ' όλα αυτά, η τεχνολογική ανεργία όπως ονομάζεται απειλεί σοβαρά ένα μεγάλο μέρος των εργαζομένων και αυτό φαίνεται να συνειδητοποιείται σιγά - σιγά παγκοσμίως.

Παρ' όλα αυτά, η τεχνολογική ανεργία όπως ονομάζεται απειλεί σοβαρά ένα μεγάλο μέρος των εργαζομένων και αυτό φαίνεται να συνειδητοποιείται σιγά - σιγά παγκοσμίως. Φαίνεται οξύμωρο αλλά αυτό που οραματίστηκαν κάποιοι ρομαντικοί της τεχνολογίας κάποτε, ένα κόσμο δηλαδή στον οποίο οι μηχανές θα δουλεύουν και οι άνθρωποι θα έχουν αρκετό ελεύθερο χρόνο για να φιλοσοφούν, να παίζουν και να ερωτεύονται, αυτό δηλαδή που σήμερα θα μπορούσε να συμβαίνει λόγω της τεχνολογικής έκρηξης, φαντάζει ακόμη μακρινό.

Απαιτείται σίγουρα για κάτι τέτοιο συναίνεση σε παγκόσμιο επίπεδο, αφού οι αγορές είναι πλέον εντελώς διεθνοποιημένες και κάθε χώρα

προβάλλει το επιχείρημα της ανταγωνιστικότητας των προϊόντων της. Η συνείδηση αυτή δεν είναι και τόσο εύκολη υπόθεση ασφαλώς. Φαίνεται και πάλι παρανοϊκό, αλλά λόγω ακριβώς της περιφημης αυτής ανταγωνιστικότητας, οι έχοντες εργασία δουλεύουν σκληρότερα για να την διατηρήσουν και να παράγουν όλο και περισσότερα και φτηνότερα προϊόντα - πολλά από τα οποία παντελώς άχρηστα για τις πραγματικές μας ανάγκες - ενώ κάποιοι άλλοι συσσωρεύονται στις λίστες των ανέργων.

Το πρώτο βήμα για μια κάποια λύση είναι η συνειδητοποίηση της τεράστιας αντίφασης κάτι που φαίνεται να έχει γίνει. Μένει να δούμε αν θα παρθούν οι σωστές αποφάσεις, έτσι που η τεχνολογία και οι τεράστιες παραγωγικές δυνατότητες που προσφέρει, να γίνουν ευλογία και όχι κατάρα.

Τέλος δυστυχώς ακόμη και σήμερα, χρόνια μετά τον ψυχρό πόλεμο, μεγάλα ποσά χρημάτων κατευθύνεται στην έρευνα οπλικών συστημάτων και γενικά της βιομηχανίας του πολέμου. Τουλάχιστον το ενθαρρυντικό στην ιστορία είναι ότι μετά από κάποια χρονική περίοδο, τα ευρήματα των ερευνών στα εργαστήρια της πολεμικής βιομηχανίας κατευθύνονται σε ειρηνικούς σκοπούς...(!)

Σύντομη Ιστορική Αναδρομή στην Εξέλιξη των Συστημάτων Αυτόματου Ελέγχου	
1769	Ο James Watt παρουσιάζει μια συσκευή ελέγχου κινούμενης σφαίρας για τον έλεγχο της περιστροφικής ταχύτητας μιας ατμομηχανής.
1800	Παρουσιάζεται η ιδέα της Eli Whitney για την κατασκευή ανταλλακτικών στην παραγωγή μουσκέτων. Η πρόταση της Whitney αναφέρεται συχνά και ως σημείο αφετηρίας της μαζικής παραγωγής.
1868	Ο James Maxwell διατυπώνει σχετική θεωρία για το σύστημα ελέγχου της ατμομηχανής του James Watt.
1913	Ο Henry Ford κατασκευάζει μια μηχανή συναρμολόγησης για την παραγωγή αυτοκινήτων.
1927	Ο H.W. Bode παρουσιάζει την ανάλυση των ενισχυτών ανάδρασης.
1932	Ο H. Nyquist παρουσιάζει μια μέθοδο ανάλυσης της ευστάθειας των συστημάτων.
1952	Ανάπτυξη του Αριθμητικού Ελέγχου στο Πανεπιστήμιο του M.I.T. για τον έλεγχο εργαλειομηχανών.
1954	Ο G. Devol εξελίσσει το σύστημα «προγραμματιζόμενης μεταφοράς άρθρου» που θεωρείται το πρώτο βήμα στην σχεδίαση των βιομηχανικών ρομπότ.
1960	Εμφάνιση του πρώτου αυτόνομου ρομπότ, βασισμένο στην σχεδίαση Devol. Τέτοια συστήματα εγκαταστάθηκαν το 1961 για την εξυπηρέτηση μηχανών καλουπώματος.
1970	Εξέλιξη της μεθόδου Ανάδρασης Καταστάσεων και της θεωρίας του Βέλτιστου Ελέγχου.
1980	Μελετάται ευρύτατα η σχεδίαση Εύρωστων Συστημάτων Ελέγχου.
1990	Οι κατασκευάστριες εξαγωγικές εταιρίες δίνουν έμφαση στον αυτοματισμό.
1994	Ευρεία χρήση του αυτόματου ελέγχου στην αυτοκινητοβιομηχανία. Η παραγωγή απαιτεί αξιόπιστα και σθεναρά συστήματα.
1997	Το πρώτο αυτόνομο διαστημικό όχημα, γνωστό με την ονομασία «Sojourner», εξερευνά την επιφάνεια του πλανήτη Άρη.

Πηγή : Richard C. Dorf, Robert H. Bishop, «Σύγχρονα Συστήματα Αυτόματου Ελέγχου», 9^η Έκδοση, Εκδόσεις Τζιόλα 2003.

Μάθημα

Το μάθημα των Συστημάτων Αυτόματου Ελέγχου του τμήματος Ηλεκτρολογίας γίνεται με τη χρήση ηλεκτρονικών υπολογιστών (για τους λόγους που αναφέρονται στην ιστορική αναδρομή) παραπάνω και πιο συγκεκριμένα στο περιβάλλον εργασίας του προγράμματος MatLab. Το MatLab είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο που χρησιμοποιείται σε πανεπιστημιακά μαθήματα αλλά και ερευνητικές και άλλες εφαρμογές με επιστημονικούς υπολογισμούς. Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων Matrix Laboratory που μεταφράζεται ως εργαστήριο πινάκων και είναι ένα **διαδραστικό** πρόγραμμα για αριθμητικούς υπολογισμούς και οπτικοποίηση δεδομένων με δυνατότητες προγραμματισμού που το καθιστούν ένα ισχυρό και χρήσιμο εργαλείο στις μαθηματικές και φυσικές επιστήμες. Όπως υποδηλώνεται και από το όνομά του είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικών πινάκων κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κ.α.

Η MatLab είναι μία υψηλού επιπέδου γλώσσα προγραμματισμού που έχει χρησιμοποιηθεί εκτενώς για επίλυση δύσκολων μηχανικών προβλημάτων. Δίνει τη δυνατότητα ανάπτυξης ενός πρωτοτύπου μίας μεθόδου γρήγορα και εύκολα καθώς και την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Το MatLab είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε *αριθμητική πεπερασμένη ακρίβειας*, δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος.

Στόχος του πρώτου κεφαλαίου είναι η εξοικείωση του σπουδαστή με τα βασικά χαρακτηριστικά και εντολές του MatLab. Εδώ όμως θα πρέπει να τονίσουμε ότι ο καλύτερος τρόπος εκμάθησης του MatLab είναι η συστηματική ενασχόληση με αυτό και η διερεύνησή του από τον ίδιο τον σπουδαστή. Το πρόγραμμα είναι εφοδιασμένο με ένα εκτενές σύστημα βοήθειας όπου κάθε εντολή επεξηγείται αναλυτικά και με αντιπροσωπευτικά παραδείγματα, τα οποία καλούνται με την πιο «σημαντική» εντολή του MatLab είναι η **Help** (βοήθεια)! Επίσης, κάποιος μπορεί να βρει πλήθος χρήσιμων πληροφοριών τόσο για αρχάριους όσο και για προχωρημένους στην επίσημη ιστοσελίδα της MatLab <http://www.mathworks.com>.

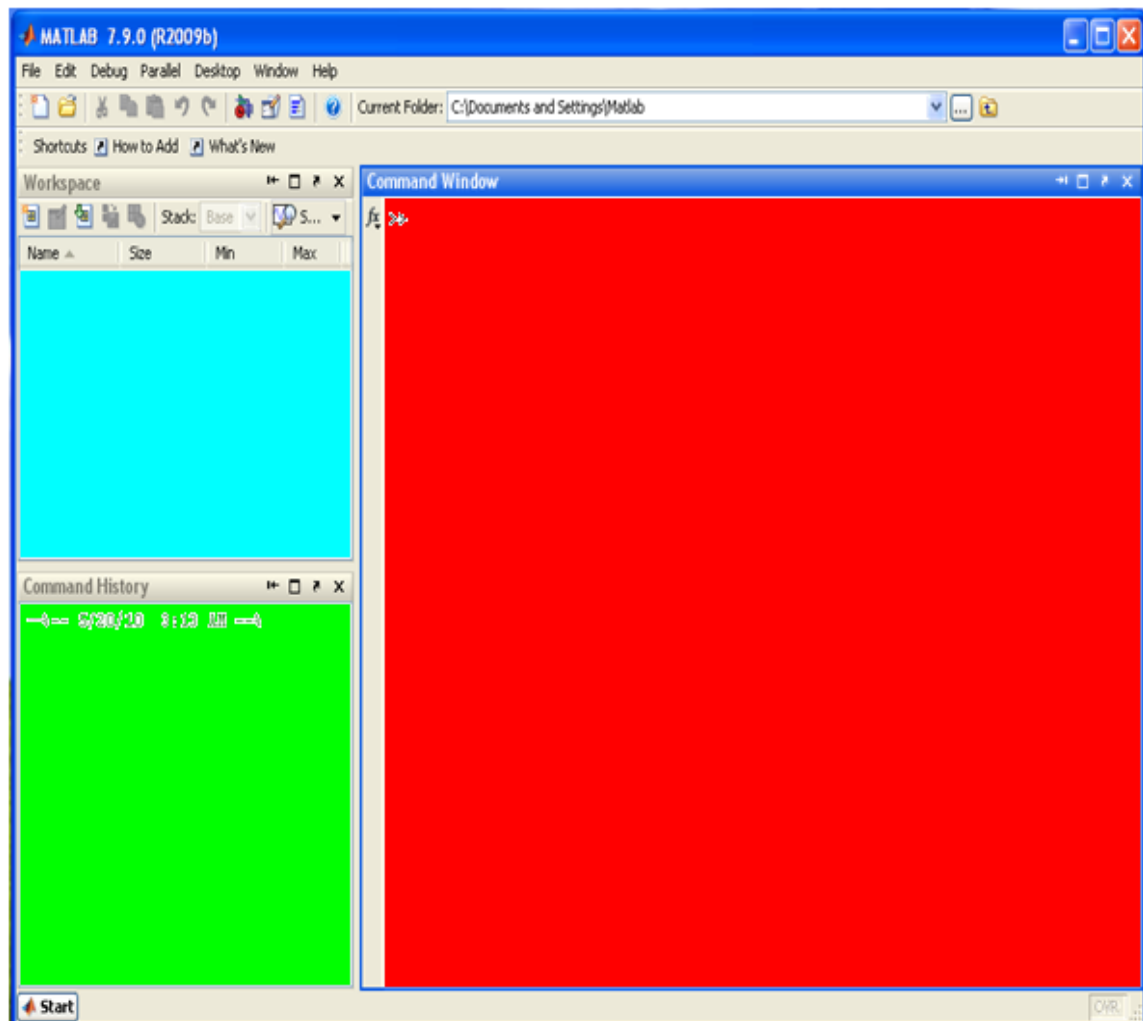
1.Εισαγωγή στο MatLab

Όπως αναφέρθηκε και στην εισαγωγή το πρώτο κεφάλαιο στοχεύει στο να εισάγει τον σπουδαστή στο περιβάλλον του MatLab, το οποίο είναι το μόνο, αλλά υπεραρκετό, «εργαλείο» που θα χρησιμοποιήσουμε στο μάθημα των Συστημάτων Αυτόματου Ελέγχου. Το πρόγραμμα του MatLab περιέχει πλήθος εντολών, εμείς θα αναλύσουμε αυτές τις οποίες πρόκειται να χρησιμοποιήσουμε κατά τη διάρκεια του μαθήματος.

Αρχικά για να ξεκινήσουμε το πρόγραμμα στον υπολογιστή θα πρέπει να επιλέξουμε από την επιφάνεια εργασίας με διπλό κλικ το παρακάτω εικονίδιο :



Μετά από λίγο, αφού το πρόγραμμα φορτώσει, θα εμφανιστεί στην οθόνη μας το παράθυρο έναρξης της MATLAB (MATLAB opening window):

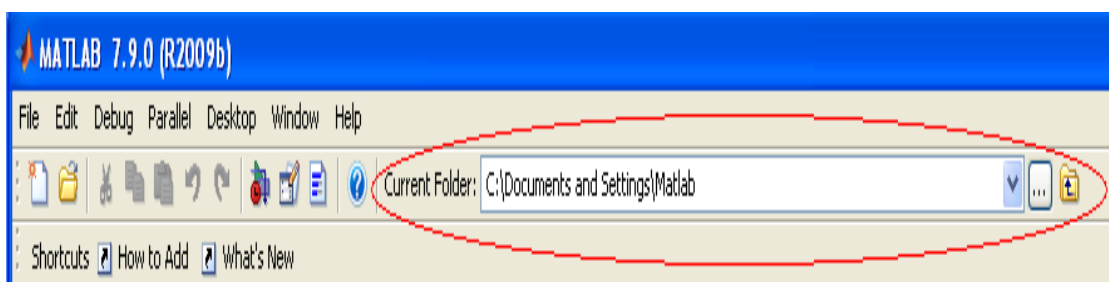


Στο περιβάλλον του προγράμματος εμφανίζονται τέσσερα παράθυρα:

1. Το **κόκκινο** παράθυρο εντολών (Command Window) στα δεξιά. Οι εντολές της MATLAB εισάγονται σ' αυτό το παράθυρο μετά την *προτροπή* (prompt) `>>`. Τα αποτελέσματα επίσης τυπώνονται-εμφανίζονται στο παράθυρο αυτό.
2. Το **σιέλ** παράθυρο που δείχνει τα στοιχεία, αλλά και πληροφορίες για αυτά σε ότι αφορά μέγεθος, διαστάσεις κ.τ.λ., που εισάγονται και υπολογίζονται στο παράθυρο εντολών είναι το παράθυρο του χώρου εργασίας (Workspace).
3. Στο **σιέλ** παράθυρο μπορεί να γίνει εναλλαγή του παραθύρου χώρου εργασίας(Workspace) με τον τρέχοντα φάκελο(Current Directory) που μας δείχνει τα αρχεία που περιέχει ο προσωπικός φάκελος που θα εξηγήσουμε στη συνέχεια γιατί είναι απαραίτητο να δημιουργηθεί στο φάκελο «Τα έγγραφα μου» του υπολογιστή κάθε σπουδαστή.
4. Το **πράσινο** παράθυρο κάτω αριστερά δείχνει το ιστορικό εντολών (Command History). Αν δεν εμφανιστεί το παράθυρο αυτό επιλέξτε *Command History* στην επιλογή *View*.

Κατά τη διάρκεια μιας εργασίας στη MATLAB μπορεί να εμφανιστούν αυτόματα και άλλα παράθυρα όταν αυτό απαιτείται όπως παράθυρα κειμένου (document windows), παράθυρα γραφικών (graphics windows) και παράθυρα σύνταξης αρχείων (editing windows).

Σημειώνεται ότι στο χώρο χρήσης ηλεκτρονικών υπολογιστών του τμήματος Ηλεκτρολογίας όπου διεξάγεται το μάθημα των Συστημάτων Αυτόματου Ελέγχου, πρέπει να αλλάξετε τον τρέχοντα φάκελο (Current Directory) επιλέγοντας τον προσωπικό σας φάκελο με την ονομασία «SAE-Im-files» ή όπως εσείς επιθυμείτε, τον οποίο έχετε προηγουμένως δημιουργήσει στον φάκελο του υπολογιστή «Τα έγγραφα μου», κάνοντας την κατάλληλη επιλογή στο χωρίο που φαίνεται στην παρακάτω εικόνα.



Προσοχή : Αν δεν ακολουθηθεί η παραπάνω οδηγία τότε όλη η εργασία και η αποθήκευση αρχείων κατά τη διάρκεια του μαθήματος θα διαγράφεται αμέσως μετά την αποσύνδεση του χρήστη. Με αποτέλεσμα την απώλεια κάθε εργασίας. Αυτό συμβαίνει εξαιτίας του τρόπου λειτουργίας των υπολογιστών στο χώρο του μαθήματος.

Το περιβάλλον MatLab αποτελεί μια ολοκληρωμένη εφαρμογή επιστημονικών και τεχνικών υπολογισμών. Το πακέτο λογισμικού MatLab αποτελείται από μια ομάδα προγραμμάτων στην οποία περιλαμβάνεται το βασί-

κό εκτελέσιμο πρόγραμμα (εφαρμογή) καθώς επίσης και ένα πλήθος από εργαλειοθήκες (Toolboxes) μια συλλογή ειδικών αρχείων με την επέκταση «.m» (M-files), τα οποία συμβάλλουν σημαντικά στην λειτουργικότητα του προγράμματος εφαρμογής. Το ομώνυμο πρόγραμμα εφαρμογής σε συνδυασμό με την εργαλειοθήκη αυτόματου ελέγχου(control system toolbox), παρέχουν την δυνατότητα χρήσης του MatLab στην ανάλυση και σχεδίαση συστημάτων αυτόματου ελέγχου.

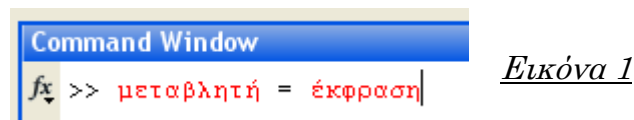
Οι περισσότερες από τις δηλώσεις, τις συναρτήσεις και τις εντολές που συναντούμε στο περιβάλλον αυτό, είναι ανεξάρτητες από το χρησιμοποιούμενο υπολογιστικό περιβάλλον. Γίνεται χρήση διαφόρων αντικειμένων με την βοήθεια των οποίων μπορούμε να επικοινωνούμε με το βασικό πρόγραμμα :

1. Δηλώσεις και Μεταβλητές
2. Πίνακες και Διανύσματα
3. Παράθυρα Γραφικών – σχήματα γραφικών παραστάσεων και
4. Κείμενα συνόλου εντολών (Scripts), τα οποία για συντομία τα καλούμε «προγράμματα».

Το περιβάλλον του MatLab λαμβάνει τις απαραίτητες πληροφορίες εισόδου από ένα ή περισσότερα από τα παραπάνω αντικείμενα.

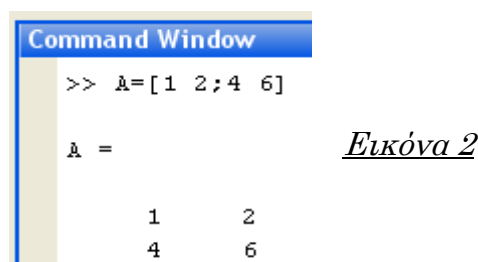
1.1 Δηλώσεις και Μεταβλητές

Οι δηλώσεις έχουν τη μορφή



Εικόνα 1

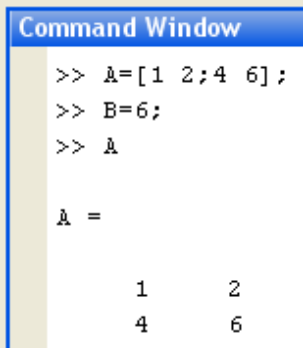
Στο περιβάλλον του MatLab χρησιμοποιείται η λογική της εκχώρησης, οπότε το σύμβολο της ισότητας ('=') υπονοεί την εκχώρηση μιας έκφρασης σε μια μεταβλητή. Παρατηρούμε ότι στη γραμμή εντολών (command prompt) του MatLab εμφανίζεται το σύμβολο '>>'.



Εικόνα 2

Στην παραπάνω εικόνα βλέπουμε μια τυπική δήλωση, όπου εισάγουμε έναν πίνακα 2*2 στον οποίο δίνουμε το όνομα της μεταβλητής 'A'. Η δήλωση αυτή εκτελείται αφού πιάσουμε το πλήκτρο τροφοδοσίας γραμμής (Enter) μετά την δεύτερη αγκύλη(στα επόμενα παραδείγματα και κατά τα υπόλοιπα κεφάλαια θα εννοείται ότι χρησιμοποιούμε το πλήκτρο τροφοδοσίας γραμμής στο τέλος κάθε εντολής). Ο πίνακας A εμφανίζεται αυτόματα στην

οθόνη όταν εκτελεστεί η συγκεκριμένη δήλωση μετά την πίεση του πλήκτρου τροφοδοσίας γραμμής. Αν η παραπάνω δήλωση ακολουθείται από το σύμβολο ‘;’ Τότε μετά την εκτέλεση της δεν εμφανίζεται η μορφή του πίνακα ‘A’, όπως φαίνεται στην Εικόνα 2. Η εκχώρηση της μεταβλητής A λαμβάνει χώρα ανεξάρτητα από την αναίρεση της εμφάνισης του αποτελέσματος με τη χρήση του συμβόλου ‘;’, όπως φαίνεται στην Εικόνα 3 όπου εκχωρούμε τον πίνακα ‘A’ συνεχίζουμε με άλλη εντολή και όταν καλέσουμε μας τον εμφανίζει κανονικά στην οθόνη.



```

Command Window
>> A=[1 2;4 6];
>> B=6;
>> A

A =

     1     2
     4     6
  
```

Εικόνα 3

Κατά τη χρήση του MatLab συμβαίνει συχνά να περιλαμβάνονται ενδιαμέσοι υπολογισμοί των οποίων τα επιμέρους αποτελέσματα δεν μας ενδιαφέρουν. Κάνοντας χρήση του συμβόλου ‘;’ όπου θεωρείται αναγκαίο, μειώνουμε σημαντικά τον φόρτο των δεδομένων εξόδου που εμφανίζονται στην οθόνη. Με τον τρόπο αυτό έχουμε το πλεονέκτημα της αύξησης της ταχύτητας των υπολογισμών εφόσον η διαδικασία απεικόνισης των αποτελεσμάτων στην οθόνη απαιτεί κάποιον χρόνο.

Στις εκφράσεις χρησιμοποιούνται οι συνήθεις μαθηματικοί τελεστές.

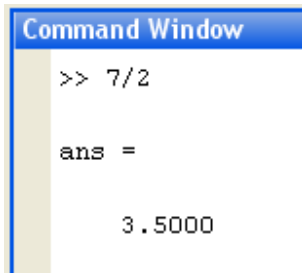
Μαθηματικοί Τελεστές	
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη

Η προτεραιότητα των πράξεων είναι αυτή που γνωρίζουμε από τους γνωστούς αλγεβρικούς κανόνες για τη σειρά εκτέλεσης πράξεων.

1. Πρώτα εκτελούνται οι πράξεις μέσα στις παρενθέσεις από τα μέσα προς τα έξω,
2. Στη συνέχεια εκτελούνται οι υψώσεις σε δύναμη,
3. Ακολουθούν οι εκτελέσεις των πολλαπλασιασμών και των διαιρέσεων από τα αριστερά προς τα δεξιά και
4. Τέλος εκτελούνται οι προσθέσεις και αφαιρέσεις από τα αριστερά προς τα δεξιά.

Στην Εικόνα 4 βλέπουμε πως μπορούμε να χρησιμοποιήσουμε το MatLab σε λειτουργία αριθμομηχανής. Όταν παραλείπεται το όνομα της μεταβλητής και το σύμβολο της ισότητας από μια έκφραση τότε το αντίστοιχο

αποτέλεσμα εκχωρείται σε μια μεταβλητή γενικής χρήσης του συστήματος με την ονομασία *ans*.

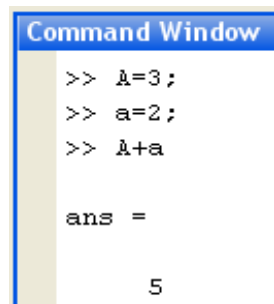


```
Command Window
>> 7/2
ans =
    3.5000
```

Εικόνα 4

Το MatLab περιλαμβάνει τις περισσότερες από τις στοιχειώδεις και τριγωνομετρικές συναρτήσεις που συναντούμε σε μια επιστημονική αριθμομηχανή.

Τα ονόματα των μεταβλητών μπορεί να είναι ένας χαρακτήρας ή ένα σύνολο από χαρακτήρες. Το μέγιστο μήκος του ονόματος μιας μεταβλητής που αντιλαμβάνεται το MatLab είναι 19 χαρακτήρες, σε περίπτωση που εισάγουμε περισσότερους το πρόγραμμα λαμβάνει υπόψη μόνο τους 19 πρώτους χαρακτήρες. Παρά την δυνατότητα του MatLab θα ήταν καλό να επιλέγουμε μικρά ονόματα για την εισαγωγή μεταβλητών. Η μεταβλητή *A* είναι διαφορετική από τη μεταβλητή *a* αφού το MatLab λειτουργεί με διαχωρισμό πεζών και κεφαλαίων, στην Εικόνα 5 παρουσιάζεται η παραπάνω λειτουργία.



```
Command Window
>> A=3;
>> a=2;
>> A+a
ans =
    5
```

Εικόνα 5

Σε αυτό το σημείο θα πρέπει να προσέξουμε τους χαρακτήρες *i*, *j* και *pi* οι οποίοι όπως θα δούμε παρακάτω χρησιμοποιούνται ως προκαθορισμένες μεταβλητές από το ίδιο το MatLab και θα πρέπει να αποφεύγουμε τη χρήση τους. Αν για οποιονδήποτε λόγο τις χρησιμοποιήσουμε, δεν δημιουργείται πρόβλημα αλλά αφού υπάρχουν αμέτρητοι τρόποι να δηλώσουμε μεταβλητές δεν υπάρχει λόγος να το κάνουμε, αλλά στη συνέχεια θέλουμε να τις χρησιμοποιήσουμε με τον προκαθορισμένο τρόπο που τις χρησιμοποιεί το MatLab τότε εισάγουμε την εντολή >>clear μεταβλητή όπου στη θέση μεταβλητή εισάγουμε *i*, *j* ή *pi* ανάλογα ποια μεταβλητή θέλουμε να επαναφέρουμε στην προκαθορισμένη τιμή της.

Συνοψίζοντας τα παραπάνω για την ονομασία μεταβλητών θα προσέχουμε τα εξής :

- Το όνομα αρχίζει με χαρακτήρα(αγγλικού αλφαβήτου)
- Το όνομα μπορεί να αποτελείται από χαρακτήρες, αριθμούς και υποπαύλες(underscore)

- Δεν χρησιμοποιούμε ονόματα που έχουν δεσμευτεί από το MatLab(π.χ. συναρτήσεις βιβλιοθήκης και εργαλείοι)
- Προτιμάμε σύντομα ονόματα αν και δεν υπάρχει περιορισμός στο μήκος ονομάτων

Κατά τη διάρκεια της εργασίας μας στο περιβάλλον του MatLab είναι πιθανή η χρήση αριθμού μεταβλητών περισσότερων από μιας. Σε περίπτωση όπου θέλουμε να εμφανίσουμε όλες τις μεταβλητές που έχουμε χρησιμοποιήσει τη δεδομένη στιγμή πληκτρολογούμε την εντολή `who` η οποία μας επιστρέφει όλες τις μεταβλητές. Υπάρχει όμως και η εντολή `whos` η οποία εκτός ότι εμφανίζει τις μεταβλητές όπως και η `who` ταυτόχρονα εμφανίζει και πρόσθετες πληροφορίες σχετικά με τις μεταβλητές σε ότι αφορά τις διαστάσεις(size), το μέγεθος(bytes) της χρησιμοποιούμενης μνήμης και τον τύπο(class). Όπως έχουμε τη δυνατότητα να εισάγουμε μεταβλητές έτσι έχουμε τη δυνατότητα να απομακρύνουμε μεταβλητές από το περιβάλλον εργασίας χρησιμοποιώντας την εντολή `clear`, η οποία απομακρύνει όλες τις μεταβλητές που ήταν φορτωμένες στο σύστημα μέχρι εκείνη τη στιγμή. Μπορούμε αν επιθυμούμε να απομακρύνουμε μόνο μια από τις μεταβλητές που έχουμε εισάγει με την εντολή `clear μεταβλητή` και απομακρύνουμε τη μεταβλητή το όνομα της οποίας ακολουθεί την εντολή. Με τον ίδιο τρόπο μπορούμε να απομακρύνουμε περισσότερες από μια μεταβλητές από το σύστημα `clear μεταβλητή1 μεταβλητή2`, με αυτόν τον τρόπο απομακρύνουμε τη μεταβλητή1 και τα μεταβλητή2 αν θέλουμε να απομακρύνουμε και άλλες μεταβλητές ακολουθούμε τον ίδιο τρόπο.

Το MatLab χρησιμοποιεί(αναγνωρίζει) το γνωστό σε όλους μας δεκαδικό συμβολισμό (π.χ για τη μαθηματική σταθερά $\pi = 3,14$) αλλά και το λεγόμενο επιστημονικό συμβολισμό (π.χ. για τη μαθηματική σταθερά $\pi = 314 \cdot 10^{-2}$). Ο επιστημονικός συμβολισμός μας εξυπηρετεί όταν θέλουμε να εισάγουμε πολύ μεγάλους αριθμούς ή πολύ μικρούς. Για να γίνουν αντιληπτοί οι παραπάνω συμβολισμοί παίρνουμε ως παράδειγμα τον αριθμό 0,00015 και τον αριθμό 150000, σύμφωνα με το δεκαδικό συμβολισμό θα τους γράφαμε όπως παραπάνω στο περιβάλλον εργασίας του MatLab ενώ σύμφωνα με τον επιστημονικό οι εντολές που θα δίναμε στο περιβάλλον εργασίας είναι οι εξής `15e-5` και `15e4`, δηλαδή γράφουμε τον αριθμό και το e δηλώνει ότι ακολουθεί δύναμη με βάση το 10 και εκθέτη τον αριθμό που ακολουθεί μετά το e.

Οι υπολογισμοί που επιτελεί το MatLab εκτελούνται με διπλή ακρίβεια. Εκτός ότι το MatLab μας επιτρέπει να γράψουμε στο περιβάλλον εργασίας με διάφορους τρόπους όπως π.χ. δεκαδικό ή επιστημονικό συμβολισμό έχει τη δυνατότητα τα δεδομένα (αποτελέσματα) που εμφανίζονται στην οθόνη να παρουσιάζονται με διάφορες μορφοποιήσεις. Η συνήθης μορφοποίηση αποτελείται από τέσσερα ψηφία μετά την υποδιαστολή για μη ακέραιους αριθμούς. Σε αυτό το σημείο πρέπει να τονίσουμε ότι η μορφοποίηση της παρουσίασης των δεδομένων δεν επηρεάζει σε καμία περίπτωση την ακρίβεια των υπολογισμών που εκτελεί το MatLab, επίσης η παρουσίαση των δεδομένων δεν είναι σίγουρο ότι θα μας εμφανίζει όλο το σύνολο των σημαντικών ψηφίων των δεδομένων, δηλαδή το τι μας εμφανίζει η οθόνη εξαρτά-

ται από τον εκάστοτε χρήστη. Η μορφοποίηση με την οποία εμφανίζονται οι αριθμοί στην οθόνη μεταβάλλονται μέσω της εντολής *format*, όταν θέσουμε μέσω της εντολής μια μορφοποίηση τότε η συγκεκριμένη μορφοποίηση θα ισχύει έως να την αλλάξουμε ξανά χρησιμοποιώντας την ίδια εντολή. Στην παρακάτω εικόνα παρουσιάζονται οι πιο συνήθεις μορφοποιήσεις με τη βοήθεια της μαθηματικής σταθερά π. Εκτός από αυτές τις οποίες παρουσιάζονται στην εικόνα υπάρχουν και οι λιγότερο διαδεδομένες που είναι η *format long g* με ακρίβεια 15 σημαντικών ψηφίων κινητής υποδιαστολής, η *format short g* με ακρίβεια 4 σημαντικών ψηφίων κινητής υποδιαστολής, η *format hex* δεκαεξαδική μορφή και η *format bank* τραπεζικός συμβολισμός με 2 δεκαδικά ψηφία για τα λεπτά του ευρώ. Για τις παραπάνω μορφοποιήσεις δεν θα επεκταθούμε περισσότερο αφού δεν θα τις χρησιμοποιήσουμε. Υπενθυμίζουμε ότι επαναφέρουμε την προεπιλεγμένη μορφοποίηση απλώς με την εντολή *format*.

```

Command Window

>> pi

ans =

    3.1416      Βασική μορφοποίηση με ακρίβεια 4 ψηφίων μετά την υποδιαστολή

>> format long
>> pi

ans =

    3.141592653589793  Ακρίβεια 15 ψηφίων σταθερή υποδιαστολής

>> format short e
>> pi

ans =

    3.1416e+000      Ακρίβεια 5 ψηφίων κινητής υποδιαστολής

>> format long e
>> pi

ans =

    3.141592653589793e+000  Ακρίβεια 15 ψηφίων κινητής υποδιαστολής

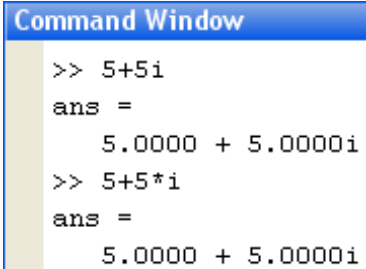
```

Εικόνα 6

Τέλος όσο αφορά την εντολή *format* εκτός από την μορφοποίηση των αριθμών που εμφανίζονται στην οθόνη μας, μπορεί να μορφοποιήσει και την απεικόνιση αυτών. Τι εννοούμε όταν λέμε απεικόνιση; Κατά τη διάρκεια της χρήσης του παραθύρου εντολών θα παρατηρήσετε ότι οι δηλώσεις των μεταβλητών ή των υπολογισμών απέχουν από τα αποτελέσματα κάποιες «σειρές» δηλαδή έχουμε αραιή απεικόνιση αφού υπάρχουν κάποιες κενές σει-

ρές. Η παραπάνω απεικόνιση είναι και η προεπιλογή του MatLab, σε περίπτωση που εμείς όμως επιθυμούμε να παραλείψουμε τις κενές γραμμές και να έχουμε μια πιο πυκνή απεικόνιση το επιτυγχάνουμε με την εντολή *format compact*, με την εντολή *format loose* επαναφέρουμε την αρχική απεικόνιση.

Παραπάνω αναφερθήκαμε για τις προκαθορισμένες μεταβλητές i και j , οι συγκεκριμένοι χαρακτήρες χρησιμοποιούνται από το MatLab για το συμβολισμό της φανταστικής μονάδας και τις χρησιμοποιούμε για την εισαγωγή μιγαδικών αριθμών. Η εισαγωγή γίνεται με την μορφή $a+bi$ των μιγαδικών αριθμών, είτε χρησιμοποιήσουμε το σύμβολο του πολλαπλασιασμού είτε όχι μεταξύ της φανταστική μονάδας και του φανταστικού μέρος του μιγαδικού αριθμού είναι το ίδιο, το MatLab το αναγνωρίζει όπως φαίνεται στην Εικόνα 7.



```

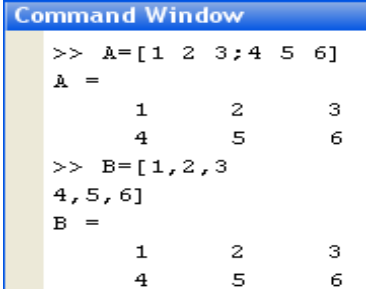
Command Window
>> 5+5i
ans =
    5.0000 + 5.0000i
>> 5+5*i
ans =
    5.0000 + 5.0000i
  
```

Εικόνα 7

1.2 Πίνακες και Διανύσματα

Όπως έχουμε αναφέρει το MatLab προέρχεται από την ένωση των λέξεων Matrix και Laboratory που ελεύθερα μεταφράζεται σε εργαστήριο πινάκων. Στην ουσία το MatLab είναι ένα πρόγραμμα το οποίο χρησιμοποιεί πίνακες και πράξεις μεταξύ αυτών, ακόμα και ένας ακέραιος αριθμός θεωρείται «λαμβάνεται» από το MatLab ως ένας πίνακας ενός στοιχείου. Σε ένα πίνακα τα στοιχεία δεν είναι απαραίτητο να είναι του ίδιου τύπου μπορεί να έχει αριθμούς, χαρακτήρες, διανύσματα(που αποτελούν και αυτά πίνακες), τριγωνομετρικές ή στοιχειώδεις συναρτήσεις, μιγαδικοί αριθμοί κ.α. .

Η δήλωση των στοιχείων ενός πίνακα γίνεται τοποθετώντας τα στοιχεία μέσα σε αγκύλες([]). Ένας πίνακας είναι μπορεί να είναι μονοδιάστατος ή και περισσότερων διαστάσεων, οπότε για τη δήλωση των στοιχείων ακολουθούμε τα εξής, τα στοιχεία μιας στήλης χωρίζονται αφήνοντας μεταξύ τους ένα κενό ή με ένα κόμμα ενώ τα στοιχεία μιας γραμμής διαχωρίζονται από αυτά της προηγούμενης γραμμής με ένα ερωτηματικό ή πατώντας το πλήκτρο «enter». Στην Εικόνα 8 παρουσιάζονται και οι δύο παραπάνω τρόποι.



```

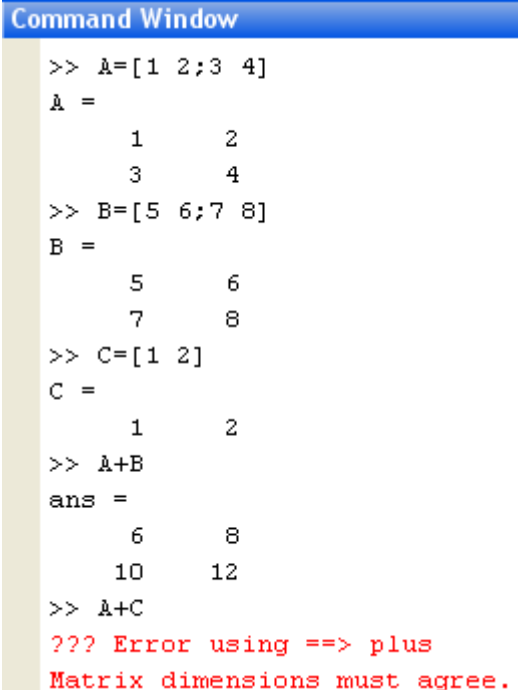
Command Window
>> A=[1 2 3;4 5 6]
A =
     1     2     3
     4     5     6
>> B=[1,2,3
4,5,6]
B =
     1     2     3
     4     5     6

```

Εικόνα 8

Κατά τη διάρκεια της δήλωσης ενός πίνακα δεν χρειάζεται να δηλώνουμε και τις διαστάσεις του ή κάποιο άλλο από τα χαρακτηριστικά του αφού αυτές οι λειτουργίες γίνονται αυτόματα από το σύστημα αναλόγως τα στοιχεία και τον τρόπο που τα έχουμε δηλώσει.

Οι πράξεις οι οποίες μπορεί να επιτελέσει το MatLab μεταξύ πινάκων είναι οι ίδιες που επιτελεί και για απλούς αριθμούς, δηλαδή πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση και ύψωση σε δύναμη. Εκτός όμως από αυτές επιτελεί και τις ονομαζόμενες διανυσματικές πράξεις οι οποίες είναι πράξεις που γίνονται μεταξύ πινάκων στοιχείο προς στοιχείο. Για να είναι εφικτή η υλοποίηση μιας πράξης μεταξύ πινάκων θα πρέπει οι διαστάσεις τους να είναι συμβατές. Όσο αφορά τη πρόσθεση και την αφαίρεση για να υλοποιηθούν μεταξύ δύο πινάκων θα πρέπει οι διαστάσεις των πινάκων να είναι ίδιες, δηλαδή α έχουμε ένα πίνακα A με διαστάσεις $n \times m$ και ένα πίνακα B με αντίστοιχες διαστάσεις $k \times q$ τότε θα πρέπει $n=k$ και $m=q$.



```

Command Window
>> A=[1 2;3 4]
A =
     1     2
     3     4
>> B=[5 6;7 8]
B =
     5     6
     7     8
>> C=[1 2]
C =
     1     2
>> A+B
ans =
     6     8
    10    12
>> A+C
??? Error using ==> plus
Matrix dimensions must agree.

```

Εικόνα 9

Στην *Εικόνα 9* βλέπουμε ότι σε περίπτωση που δε είναι συμβατές οι διαστάσεις των πινάκων για την εκάστοτε πράξη το πρόγραμμα μας επιστρέφει σφάλμα και μας εξηγεί ότι πρέπει οι διαστάσεις των πινάκων να συμφωνούν. Στον πολλαπλασιασμό των πινάκων συμβατότητα έχουμε μόνο όταν, για τους ίδιους πίνακες A και B, ισχύει $m=k$ και το αποτέλεσμα είναι ένας πίνακας με διαστάσεις $n \times q$. Υπάρχει όμως και ο διανυσματικός πολλαπλα-

σιασμός όπου πολλαπλασιάζεται ένας πίνακας με ένα διάνυσμα. Στη περίπτωση του πίνακα A και ενός διανύσματος D με διάσταση d , ο διανυσματικός πολλαπλασιασμός μεταξύ τους είναι δυνατός μόνο αν $m=d$ και το αποτέλεσμα είναι ένα διάνυσμα με διαστάσεις $n \times 1$. Εκτός από τις πράξεις μεταξύ πινάκων μπορούμε να κάνουμε πράξεις μεταξύ αριθμών και πινάκων, δηλαδή μπορούμε να προσθέσουμε ή να αφαιρέσουμε ένα αριθμό x σε όλα τα στοιχεία ενός πίνακα ή ενός διανύσματος. Επίσης μπορούμε να πολλαπλασιάσουμε όλα τα στοιχεία ενός πίνακα A ή ενός διανύσματος D με έναν αριθμό x , αυτό το είδος πολλαπλασιασμού ονομάζεται βαθμωτός πολλαπλασιασμός. Με τον ίδιο τρόπο έχουμε τη δυνατότητα να διαιρέσουμε όλα τα στοιχεία ενός πίνακα ή ενός διανύσματος με έναν αριθμό x , προσοχή όμως στην περίπτωση της διαίρεσης ο αριθμός θα πρέπει να είναι μη μηδενικός. Τα παραπάνω παρουσιάζονται στην παρακάτω εικόνα.

```

Command Window
>> A=[1 2;3 4]
A =
     1     2
     3     4
>> A+1
ans =
     2     3
     4     5
>> A-1
ans =
     0     1
     2     3
>> A*5
ans =
     5    10
    15    20
>> A/2
ans =
    0.5000    1.0000
    1.5000    2.0000

```

Εικόνα 10

Με τη χρήση της αποστρόφου ($'$) μπορούμε να βρούμε τον αντίστροφο πίνακα ενός πραγματικού πίνακα. Στο παρακάτω παράδειγμα της [Εικόνας 11](#) βρίσκουμε τον αντίστροφο πίνακα του πραγματικού πίνακα A που χρησιμοποιήσαμε και σε παραπάνω παραδείγματα.

```

Command Window
>> A=[1 2;3 4]
A =
     1     2
     3     4
>> A'
ans =
     1     3
     2     4

```

Εικόνα 11

Η παραπάνω χρήση της αποστρόφου ισχύει για οποιονδήποτε πραγματικό πίνακα. Σε περίπτωση που ο πίνακας δεν είναι πραγματικός αλλά μιγαδικός η χρήση της αποστρόφου μας δίνει διαφορετικό αποτέλεσμα από την απλή αναστροφή του πίνακα. Η χρήση λοιπόν της αποστρόφου σε ένα μιγαδικό πίνακα μας δίνει τον ανάστροφο-συζυγή πίνακα του μιγαδικού πίνακα. Η χρήση τελείας πριν την απόστροφο (') μας δίνει τον ανάστροφο του μιγαδικού πίνακα χωρίς να παίρνουμε τα συζυγή στοιχεία, ενώ με τη χρήση τελείας και δύο αποστρόφων (.'') παίρνουμε ένα πίνακα που αποτελεί το συζυγή του μιγαδικού πίνακα χωρίς όμως αναστροφή.

```

Command Window
>> M=[1+6i 5+4i 3-9i;10-3i 2+6i 7-8i]
M =
 1.0000 + 6.0000i  5.0000 + 4.0000i  3.0000 - 9.0000i
10.0000 - 3.0000i  2.0000 + 6.0000i  7.0000 - 8.0000i
>> M'
ans =
 1.0000 - 6.0000i  10.0000 + 3.0000i
 5.0000 - 4.0000i  2.0000 - 6.0000i
 3.0000 + 9.0000i  7.0000 + 8.0000i
>> M.'
ans =
 1.0000 + 6.0000i  10.0000 - 3.0000i
 5.0000 + 4.0000i  2.0000 + 6.0000i
 3.0000 - 9.0000i  7.0000 - 8.0000i
>> M.``
ans =
 1.0000 - 6.0000i  5.0000 - 4.0000i  3.0000 + 9.0000i
10.0000 + 3.0000i  2.0000 - 6.0000i  7.0000 + 8.0000i

```

Εικόνα 12

Με συνδυασμό της χρήσης της πράξης του πολλαπλασιασμού και της πράξης της αναστροφής μπορούμε να εκτελέσουμε το εσωτερικό γινόμενο διανυσμάτων καθώς και το εξωτερικό γινόμενο δύο διανυσμάτων. Υπενθυμίζουμε σε αυτό το σημείο ότι το εσωτερικό γινόμενο δύο διανυσμάτων είναι ένα βαθμωτό μέγεθος ενώ το εξωτερικό γινόμενο δύο διανυσμάτων είναι ένα διάνυσμα πρώτου βαθμού και διαστάσεων $d \times d$ (όπου d οι διαστάσεις των διανυσμάτων). Αν είχαμε λοιπόν δύο διανύσματα D και Q με διαστάσεις $d \times 1$ αντιστοίχως(σε περίπτωση που εισάγετε τα στοιχεία του διανύσματος με διαστάσεις $1 \times d$ τότε οι εντολές που ακολουθούν θα δίνουν διαφορετικό α-

ποτέλεσμα), το εσωτερικό γινόμενο των δύο διανυσμάτων υπολογίζεται δίνοντας την εντολή $D'*Q$ και μας επιστρέφει αριθμό, ενώ το εξωτερικό γινόμενο των δύο διανυσμάτων υπολογίζεται με την εντολή $D*Q'$ και μας επιστρέφει ένα πίνακα. Ας δούμε δύο παραδείγματα για εσωτερικό και εξωτερικό γινόμενο αντιστοιχώς.

```

Command Window
>> D=[1;2;3;5];Q=[7;8;4;6];
>> D'*Q
ans =
    65
>> D*Q'
ans =
     7     8     4     6
    14    16     8    12
    21    24    12    18
    35    40    20    30

```

Εικόνα 13

Οι πράξεις μεταξύ πινάκων, όπως αναφέραμε και στην αρχή της ενότητας, μπορούν να εκτελεστούν και σε επίπεδο στοιχείο προς στοιχείο χρησιμοποιώντας τη τελεία πριν από τον αντίστοιχο τελεστή της πράξης που θέλουμε να εκτελέσουμε, για παράδειγμα για τον πολλαπλασιασμό δύο πινάκων στοιχείο προς στοιχείο χρησιμοποιούμε τη τελεία πριν το επί(*).

Το MatLab μας δίνει τη δυνατότητα να ορίσουμε τα στοιχεία ενός πίνακα ή ενός διανύσματος με συγκεκριμένο βήμα. Σε αυτή τη περίπτωση δηλώνουμε μέσα σε αγκύλες το πρώτο στοιχείο, το βήμα και το τελευταίο στοιχείο του πίνακα μεταξύ των οποίων παρεμβάλουμε το σύμβολο της άνω κάτω τελείας. Αν για παράδειγμα θέλαμε να φτιάξουμε ένα πίνακα με όλους τους άρτιους αριθμούς από το μηδέν έως το δέκα θα δίναμε την εντολή της επόμενης εικόνας.

```

Command Window
>> A=[0:2:10]
A =
     0     2     4     6     8    10

```

Εικόνα 14

Πίνακες χρησιμοποιούμε ακόμα και για τη δημιουργία γραφικών παραστάσεων. Μια γραφική παράσταση όπως στη γνωρίζουμε είναι στην ουσία η απεικόνιση της μεταβολής μιας συνάρτησης σε σχέση με μία μεταβλητή την οποία μεταβάλλουμε. Για αυτήν την απεικόνιση μιας συνάρτησης αρχικά θα πρέπει να δημιουργήσουμε ένα πίνακα με τις τιμές που θα παίρνει η μεταβλητή μας και στη συνέχεια να δηλώσουμε τη σχέση που τη συνδέει με τη συνάρτηση μας. Στην ουσία έχουμε δύο πίνακες ίδιων διαστάσεων ένα της μεταβλητής και ένα με τις τιμές που λαμβάνει η συνάρτηση για την εκάστοτε τιμή της μεταβλητής. Αφού λοιπόν έχουμε δύο πίνακες το τελευταίο στάδιο που απομένει για τη γραφική παράσταση είναι δώσουμε εντολή να σχεδιαστεί ένα διάγραμμα χρησιμοποιώντας τα στοιχεία των πινάκων (αναλυτικά στα διαγράμματα και στις παραστάσεις θα αναφερθούμε στην επόμενη ενότητα).

1.3 Γραφικές Παραστάσεις και Διαγράμματα

Οι γραφικές παραστάσεις και τα διαγράμματα έχουν πολύ σημαντικό ρόλο στη σχεδίαση και στην ανάλυση των συστημάτων αυτόματου ελέγχου. Το MatLab έχει εξαιρετικές δυνατότητες σε αυτόν τον τομέα και περιέχει αρκετές συναρτήσεις για σχεδίαση. Το MatLab εμφανίζει τις παραστάσεις και τα διαγράμματα σε ένα νέο ανεξάρτητο παράθυρο το οποίο ενεργοποιείται αυτόματα κάθε φορά που δίνουμε εντολή να εκτελεστεί κάποια από τις συναρτήσεις που αφορούν τη σχεδίαση παραστάσεων.

Μια από τις συναρτήσεις σχεδίασης είναι η εντολή-συνάρτηση *plot*, η οποία «ανοίγει» νέο παράθυρο το οποίο ονομάζεται παράθυρο σχήματος (figure). Η συνάρτηση *plot* σχεδιάζει τη κλασική μορφή ενός διαγράμματος, που γνωρίζουμε όλοι, σε καρτεσιανές συντεταγμένες. Τα δεδομένα-τιμές που αντιστοιχούν στους άξονες X και Y είναι αποθηκευμένα στα ισομήκη διανύσματα που έχουμε προηγουμένως αποθηκεύσει τις τιμές που παίρνει η μεταβλητή μας και τις τιμές που θα παίρνει για την εκάστοτε τιμή της μεταβλητής η συνάρτηση που έχουμε ορίσει. Η συνάρτηση *plot* στην ουσία ενώνει τα σημεία που ορίζονται από τα στοιχεία των διανυσμάτων με συνεχείς γραμμές, η κλίμακα των αξόνων και ο τύπος των χρησιμοποιούμενων γραμμών επιλέγεται αυτόματα από το ίδιο το πρόγραμμα. Βέβαια μας δίνεται η δυνατότητα αν το επιθυμούμε να αλλάξουμε τον προεπιλεγμένο τύπου γραμμής που χρησιμοποιεί το πρόγραμμα, οι διάφοροι τύποι που είναι διαθέσιμοι παρουσιάζονται στον παρακάτω πίνακα.

Τύποι γραμμών για τις καμπύλες των διαγραμμάτων	
-	Συνεχής γραμμή
- -	Διακεκομμένη γραμμή
:	Γραμμή με στίγματα
-.	Διακεκομμένη γραμμή με στίγματα

Ας υποθέσουμε ότι έχουμε μια συνάρτηση $f(x) = 5 * x + 2$ και θέλουμε να σχεδιάσουμε τη γραφική της παράσταση τιμές της μεταβλητή X από 0 έως 100, θα πρέπει αρχικά να δώσουμε ένα διάνυσμα με τις τιμές τις μεταβλητές, στη συνέχεια να «τρέξουμε» τη συνάρτηση για τις τιμές της μεταβλητής οπότε στην ουσία φτιάχνουμε ένα δεύτερο διάνυσμα ίδιου μήκους με αυτό της μεταβλητής και τέλος καλούμε τη συνάρτηση *plot*, για να γίνει κατανοητό ακολουθεί εικόνα με τις παραπάνω εντολές.

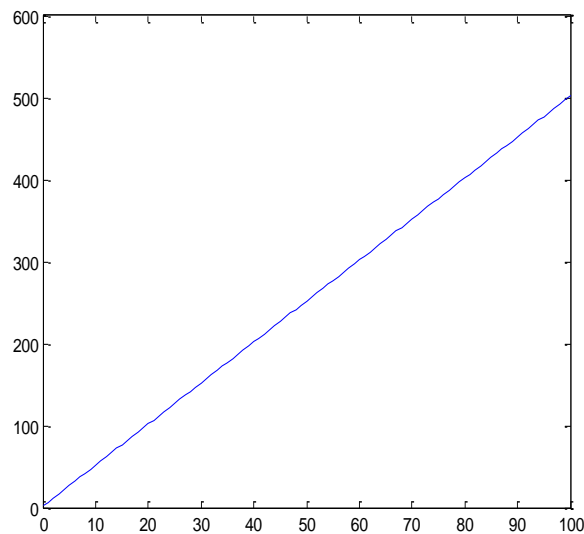
```

Command Window
>> x=[0:100];
>> y=5*x+2;
>> plot(x,y)

```

Εικόνα 15

Μετά την εντολή *plot* ανοίγει ένα νέο παράθυρο που παρουσιάζει την παρακάτω γραφική παράσταση.



Εικόνα
16

Η *plot* είναι η συνάρτηση την οποία θα χρησιμοποιήσουμε σε μεγάλο βαθμό κατά τη διάρκεια του μαθήματος των συστημάτων αυτόματου ελέγχου. Εκτός όμως από αυτή υπάρχουν και άλλες συναρτήσεις που θα ήταν καλό να τις αναφέρουμε. Η συνάρτηση *loglog* χρησιμοποιείται με τον ίδιο τρόπο που χρησιμοποιούμε την *plot*, δηλαδή μας επιστρέφει το διάγραμμα του διανύσματος x συναρτήσει του διανύσματος y με τη διαφορά ότι και οι δυο άξονες του γραφήματος είναι σε λογαριθμική κλίμακα. Η συνάρτηση *semilogx* μας επιστρέφει τη γραφική παράσταση του διανύσματος x συναρτήσει του διανύσματος y με διαφορά ότι ο άξονας x είναι σε λογαριθμική κλίμακα ενώ ο άξονας y είναι σε γραμμική κλίμακα. Τέλος η *semilogy* μας επιστρέφει τη γραφική παράσταση του διανύσματος x συναρτήσει του διανύσματος y με διαφορά ότι ο άξονας y είναι σε λογαριθμική κλίμακα ενώ ο άξονας x είναι σε γραμμική κλίμακα.

Παρατηρώντας το παραπάνω γράφημα που μας επιστρέφει η εντολή *plot* βλέπουμε ότι είναι ένα γράφημα «γυμνό» από πληροφορίες. Μπορούμε λοιπόν με τη χρήση άλλων συναρτήσεων να εμπλουτίσουμε το γράφημα με πληροφορίες που θέλουμε, όπως ονομασίες αξόνων, ετικέτα, υπόμνημα κτλ. Οι συναρτήσεις που διαχειριζόμαστε τις γραφικές παραστάσεις αλλά και τι εκτελούν αυτές ακολουθούν στον παρακάτω πίνακα.

Συναρτήσεις διαχείρισης γραφικών παραστάσεων	
title('κείμενο')	Εισάγει μια επικεφαλίδα στο πάνω μέρος του παραθύρου της γραφικής παράστασης
legend('ονομα1','ονομα2',...)	Εισάγει ένα υπόμνημα στο διάγραμμα χρησιμοποιώντας ως ετικέτες για τις γραφικές τις ονομασίες που έχουμε δώσει.
xlabel('κείμενο')	Εισάγει στον άξονα x ετικέτα με την ονομασία που έχουμε δώσει.
ylabel('κείμενο')	Εισάγει στον άξονα y ετικέτα με την ονομασία που έχουμε δώσει.
grid on	Προσθέτει γραμμές πλέγματος στο διάγραμμα
grid off	Αφαιρεί τις γραμμές πλέγματος στο διάγραμμα

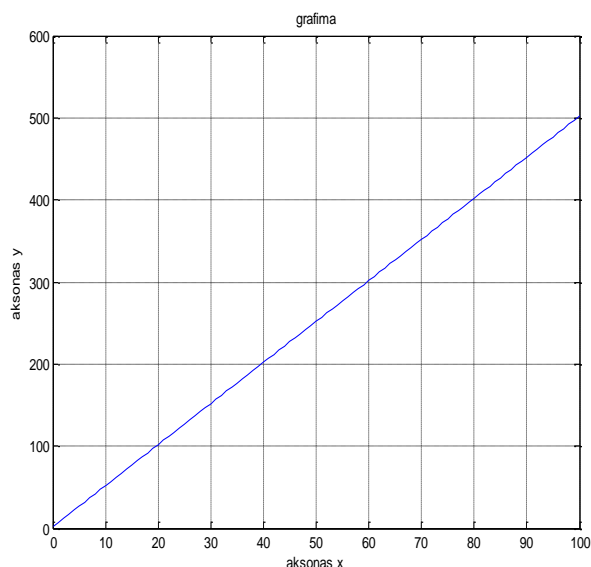
Τις παραπάνω εντολές τις τοποθετούμε μετά την εντολή plot όπως φαίνεται στην Εικόνα 17.



```
>> plot(x,y);title('grafima');xlabel('aksonas x');ylabel('aksonas y');grid on;
```

Εικόνα 17

Οι παραπάνω εντολές μας δίνουν ως αποτέλεσμα το γράφημα το οποίο εμφανίζεται στην επόμενη σελίδα. Παρατηρούμε ότι είναι αρκετά διαφορετικό σε σχέση με αυτό που μας δίνει αυτόματα το πρόγραμμα ως προεπιλογή.






Εικόνα 18

Οι παραπάνω προσθήκες μπορούν να γίνουν είτε μέσω του παραθύρου εντολών και δίνοντας τις κατάλληλες εντολές είτε από το ίδιο το παράθυρο γραφικών που εμφανίζεται μέσω του μενού Insert, επιλέγοντας το με

το ποντίκι μας θα εμφανιστούν οι επιλογές μέσω των οποίων μπορούμε να αλλάξουμε τα στοιχεία του γραφήματος.

Σε περίπτωση που θελήσουμε να δημιουργήσουμε νέα γραφική παράσταση χωρίς όμως να χάσουμε την προηγούμενη, η εντολή *figure* μας δίνει τη δυνατότητα να σχεδιαστεί μια γραφική παράσταση σε νέο παράθυρο σχεδίασης, η εντολή αυτή σε αντίθεση με τις εντολές που αναφέραμε για τη διαχείριση των γραφικών παραστάσεων δεν ακολουθεί αλλά προηγείται της εντολής *plot*. Αν επιθυμούμε όμως να σχεδιάσουμε μια νέα γραφική παράσταση στο ήδη υπάρχων παράθυρο σχεδίασης χωρίς όμως να χάσουμε την προηγούμενη γραφική ώστε να μπορούμε να τις συγκρίνουμε μεταξύ τους υπάρχει η εντολή *hold* που μας δίνει αυτή τη δυνατότητα. Με τη χρήση αυτής της εντολής μπορούμε να σχεδιάσουμε αρκετές γραφικές παραστάσεις στο ίδιο παράθυρο σχεδίασης.

Τέλος θα αναφερθούμε στην εντολή *subplot* η οποία μας επιτρέπει να διαιρέσουμε το παράθυρο που εμφανίζεται μια γραφική παράσταση σε μικρότερα παράθυρα. Η εντολή δίνεται με τη μορφή *subplot(a,b,c)* όπου

-  a : ο αριθμός των γραμμών που θα διαιρεθεί το παράθυρο
-  b : ο αριθμός των γραφημάτων ανά γραμμή
-  c : η θέση του γραφήματος στο οποίο θέλουμε να εισάγουμε την παράσταση.

Ειδικότερα μπορούμε να πούμε ότι η παραπάνω εντολή διαμερίζει το παράθυρο σχεδίασης σε a x b υπο-παράθυρα και τοποθετεί το επόμενο γράφημα στη θέση c. Για να γίνει αντιληπτή η παραπάνω εντολή ακολουθεί παράδειγμα όπου δηλώνουμε τέσσερις διαφορετικές συναρτήσεις ως προς x και στη συνέχεια σχεδιάζουμε τη γραφική παράσταση κάθε συνάρτησης σε ένα παράθυρο χωρισμένο σε υπο-παράθυρα, για να γίνει πιο εύκολη η κατανόηση ως προς το πώς εισάγουμε τα γραφήματα στην εκάστοτε θέση έχουμε δώσει τίτλους σε κάθε γράφημα ώστε να είναι εμφανής η θέση του.

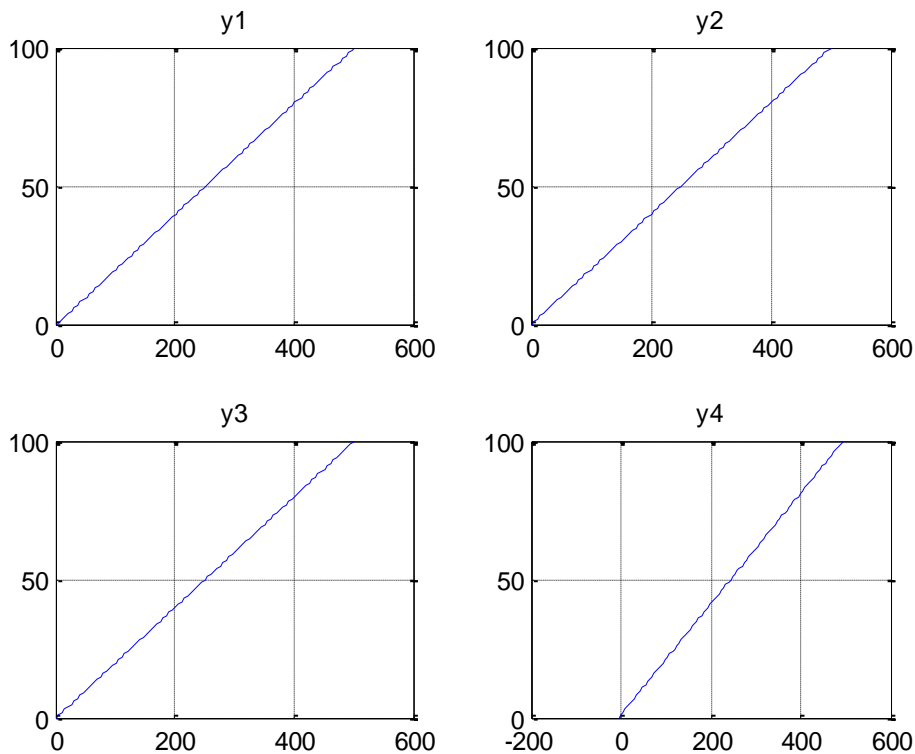
```

Command Window
>> x=[0:100];
>> y1=5*x+2;y2=5*x;y3=5*x+1;y4=5*x-6;
>> subplot(2,2,1);plot(y1,x);grid on;title('y1');
>> subplot(2,2,2);plot(y2,x);grid on;title('y2');
>> subplot(2,2,3);plot(y3,x);grid on;title('y3');
>> subplot(2,2,4);plot(y4,x);grid on;title('y4');

```

Εικόνα 19

Με τις παραπάνω εντολές λαμβάνουμε σε νέο παράθυρο τα εξής γραφήματα :

Εικόνα 20

1.4 Πολυώνυμα

Τα πολυώνυμα στο MatLab αναπαρίστανται ως διανύσματα που περιέχουν τους συντελεστές των δυνάμεων των όρων του πολυωνύμου διατεταγμένους κατά φθίνουσα σειρά. Στα πολυώνυμα για μεταβλητή συνηθίζεται να χρησιμοποιείται ο χαρακτήρας s του αγγλικού αλφαβήτου, οπότε αν είχαμε το πολυώνυμο $y(s) = 2 * s^2 + 5 * s + 6$ θα το δηλώναμε στο MatLab ως εξής :

```
Command Window
>> y=[2 5 6]
y =
    2     5     6
```

Εικόνα 21

Είναι προφανές ότι να κάποιος από του συντελεστές κάποιου όρου του πολυωνύμου είναι μηδενικός θα πρέπει να ληφθεί υπόψη και στην θέση του να τοποθετήσουμε το μηδέν αλλιώς θα έχουμε ως αποτέλεσμα ένα τελείως διαφορετικό πολυώνυμο. Δηλαδή αν επιθυμούμε να εισάγουμε το πολυώνυμο $y(s) = 2 * s^4 + 2 * s^2 + 6$ και δώσουμε την εντολή :

```
Command Window
>> y=[2 2 6]
y =
    2     2     6
```

Εικόνα 22

Τότε το αποτέλεσμα θα είναι λάθος αφού το MatLab αντιλαμβάνεται με την παραπάνω εντολή το πολυώνυμο $y_{(s)} = 2 * s^2 + 2 * s + 6$ το οποίο όμως δεν είναι αυτό που θέλουμε, οπότε η σωστή εντολή για να πάρουμε ως αποτέλεσμα το πολυώνυμο $y_{(s)} = 2 * s^4 + 2 * s^2 + 6$ είναι :

```
Command Window
>> y=[2 0 2 0 6]
y =
     2     0     2     0     6
```

Εικόνα 23

Παρατηρούμε λοιπόν ότι πρέπει να τοποθετήσουμε το μηδενικό συντελεστή της τρίτης δύναμης αλλά και της πρώτης δύναμης στις αντίστοιχες θέσεις των συντελεστών στο διάνυσμα δηλαδή στην ουσία το πολυώνυμο το γράφουμε $y_{(s)} = 2 * s^4 + 0 * s^3 + 2 * s^2 + 0 * s + 6$.

Το MatLab διαθέτει κάποιες συναρτήσεις οι οποίες μας βοηθούν στις πράξεις όπου μετέχουν πολυώνυμα. Αν θέλουμε να βρούμε τις ρίζες ενός πολυωνύμου καλούμε τη συνάρτηση roots(όνομα πολυωνύμου) η οποία μας επιστρέφει ένα διάνυσμα στήλη που περιέχει τις ρίζες του πολυωνύμου. Έστω ότι έχουμε το πολυώνυμο $y_{(s)} = s^2 + 5 * s + 6$ με τη χρήση της εντολής roots(y) βρίσκουμε ότι το πολυώνυμο έχει δύο ρίζες την $y_1 = -3$ και την $y_2 = -2$.

```
Command Window
>> y=[1 5 6]
y =
     1     5     6
>> roots(y)
ans =
    -3.0000
    -2.0000
```

Εικόνα 24

Αν τώρα αντίστροφα γνωρίζουμε τις ρίζες κάποιου πολυωνύμου αλλά δεν ξέρουμε το πολυώνυμο τότε εισάγοντας τις ρίζες σε ένα διάνυσμα στήλη το MatLab με τη χρήση της συνάρτησης poly(όνομα διανύσματος) μας επιστρέφει το πολυώνυμο που έχει ρίζες τις ρίζες που έχουμε εισάγει στο διάνυσμα. Αν όπως στο προηγούμενο παράδειγμα είχαμε δύο ρίζες την $y_1 = -3$ και την $y_2 = -2$ και ψάχναμε το πολυώνυμο θα δίναμε τις εντολές της Εικόνας 24.

```
Command Window
>> x=[-2;-3]
x =
    -2
    -3
>> poly(x)
ans =
     1     5     6
```

Εικόνα 25

Εκτός από τις ρίζες ενός πολυωνύμου ή την εύρεση του πολυωνύμου μέσω των ριζών μπορούμε να υπολογίσουμε τη τιμή ενός πολυωνύμου για μια συγκεκριμένη τιμή της μεταβλητής του. Η συνάρτηση *polyval(y,x)* υπολογίζει τη τιμή του πολυωνύμου $y(s)$ για $s=x$.

```
Command Window
>> y=[1 5 6];
>> polyval(y,5) Εικόνα 26
ans =
    56
```

Στην Εικόνα 25 υπολογίζουμε τη τιμή του πολυωνύμου $y(s)$ για $s=5$, άρα $y(5) = 56$.

Η συνάρτηση *polyder(πολυώνυμο)* μας επιστρέφει την παράγωγο του πολυωνύμου.

```
Command Window
>> polyder(y) Εικόνα 27
ans =
     2     5
```

Αυτό που αντιλαμβανόμαστε από την Εικόνα 26 είναι ότι η παράγωγος του πολυωνύμου $y(s) = s^2 + 5 * s + 6$ ισούται με $y'(s) = 2 * s + 5$.

Το MatLab μπορεί να πραγματοποιήσει πράξεις μεταξύ πολυωνύμων. Η πρόσθεση και η αφαίρεση πολυωνύμων ίδιου μήκους (δηλαδή πολυωνύμων ίδιου βαθμού) γίνεται απλά με την τοποθέτηση του απαραίτητου τελεστή μεταξύ των δύο πολυωνύμων. Σε περίπτωση που τα πολυώνυμα δεν είναι ίδιου μήκους τότε στο διάνυσμα του πολυωνύμου με την μικρότερη τάξη προσθέτουμε στην αρχή του διανύσματος κατάλληλο αριθμό μηδενικών ώστε να πάρουμε διανύσματα συντελεστών ίδιου μήκους και στην συνέχεια εκτελούμε την πράξη που επιθυμούμε. Για τα πολυώνυμα $y_{1(s)} = 4 * s + 6$ και $y_{2(s)} = 5 * s^4 + 3 * s + 2$ η πρόσθεση θα γινόταν όπως παρουσιάζεται στην Εικόνα 28.

```
Command Window
>> y1=[0 0 0 4 6];
>> y2=[5 0 0 3 2];
>> y1+y2 Εικόνα 28
ans =
     5     0     0     7     8
```

Ο πολλαπλασιασμός επιτυγχάνεται με τη χρήση της συνάρτησης *conv(πολυώνυμο1,πολυώνυμο2)*. Για τα ίδια πολυώνυμα που χρησιμοποιήσαμε στο παράδειγμα της πρόσθεσης, ο πολλαπλασιασμός τους παρουσιάζεται στην Εικόνα 29.


```

Command Window
>> y1=[4 6];y2=[5 0 0 3 2];
>> conv(y1,y2)
ans =
    20    30     0    12    26    12

```

Εικόνα 29

Βλέπουμε ότι στον πολλαπλασιασμό δεν χρειάζεται να μετατρέψουμε τα διανύσματα των πολυωνύμων σε ισομήκη, επίσης στον πολλαπλασιασμό δεν μπορούμε να εισάγουμε παραπάνω από δύο πολυώνυμα, στη περίπτωση που θέλουμε να πολλαπλασιάσουμε τρία πολυώνυμα τότε θα πολλαπλασιάσουμε δύο πολυώνυμα μεταξύ τους και το αποτέλεσμα αυτών το πολλαπλασιάζουμε με το τρίτο πολυώνυμο, ακολουθούμε όμοια διαδικασία αν θέλουμε να πολλαπλασιάσουμε παραπάνω από τρία πολυώνυμα. Η εντολή $[P,Y]=deconv(\text{πολυώνυμο1},\text{πολυώνυμο2})$ διαιρεί το πολυώνυμο1 με το πολυώνυμο2 και επιστρέφει το πολυώνυμο P που αποτελεί το πηλίκο της διαίρεσης και το πολυώνυμο Y που αποτελεί το υπόλοιπο της διαίρεσης.

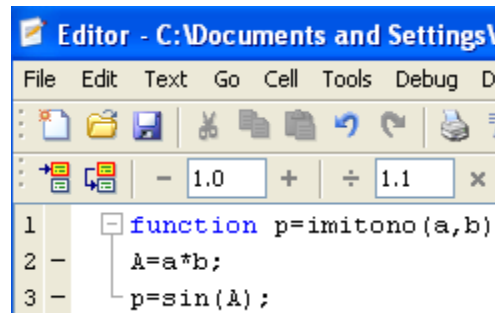
1.5 Αρχεία εντολών και Αρχεία συναρτήσεων

Το MatLab έχει στη βάση δεδομένων του εκατοντάδες εντολές και συναρτήσεις που δύσκολα κανείς μπορεί να θυμάται ή να τις χρησιμοποιήσει όλες με την ίδια ευκολία, παρόλα αυτά το MatLab δίνει τη δυνατότητα στο χρήστη να φτιάξει τις δικές του εντολές και τις δικές του συναρτήσεις. Η δυνατότητα αυτή μας εξυπηρετεί σε περίπτωση που για την επίλυση ενός προβλήματος έχουμε προς εκτέλεση μια μεγάλη σειρά από εντολές, το MatLab μας επιτρέπει να αποθηκεύσουμε τις εντολές αυτές σε ένα αρχείο και να το εκτελούμε όποτε αυτό χρειάζεται. Τέτοιου είδους αρχεία τα αποθηκεύουμε στο φάκελο που έχουμε δημιουργήσει στο φάκελο «Τα έγγραφα μου» δηλαδή στον τρέχοντα φάκελο (οπότε όλα αυτά τα αρχεία είναι εμφανή στο αντίστοιχο παράθυρο του MatLab) το καθένα με μια ξεχωριστή ονομασία. Όταν θελήσουμε γράφουμε το όνομα του αρχείου στο παράθυρο εντολών(και πατώντας enter) το MatLab αυτόματα τρέχει όλες τις εντολές που εμείς έχουμε αποθηκεύσει στο αρχείο.

Εκτός όμως από τα αρχεία εντολών μπορούμε να δημιουργήσουμε και αρχεία συναρτήσεων. Αρχεία συναρτήσεων το μπορούν να εισαχθούν από το χρήστη στη βάση δεδομένων του MatLab αν εκφραστούν σωστά με τη βοήθεια άλλων συναρτήσεων ή εντολών που ήδη υπάρχουν στη βάση δεδομένων του. Το αρχείο της συνάρτησης μπορεί να αποτελείται από ένα σύνολο εντολών και συναρτήσεων καθώς και να απαιτεί την εισαγωγή από το χρήστη κάποιων τιμών κάθε φορά που θα εκτελείται. Στην αρχή του αρχείου στη πρώτη γραμμή πρέπει να υπάρχει μία γραμμή που περιέχει την σύνταξη της συνάρτησης και πάντα αρχίζουμε με τη λέξη *function* (συνάρτηση). Πρέπει να τονισθεί ότι το όνομα της συνάρτησης πρέπει να είναι ίδιο με το

όνομα του αρχείου που την περιέχει γιατί με αυτό το όνομα θα την καλούμε από το παράθυρο εντολών.

Τα αρχεία εντολών όπως και τα αρχεία συναρτήσεων τα ονομάζουμε m-files και για να δημιουργήσουμε νέο αρχείο επιλέγουμε στο menu εντολών του MatLab File → New → M-file και εμφανίζεται στην οθόνη μας ένα νέο παράθυρο στο οποίο γράφουμε τις εντολές που επιθυμούμε. Στην Εικόνα 30 ακολουθεί παράδειγμα ενός αρχείου συνάρτησης που δέχεται ως είσοδο δύο αριθμούς και υπολογίζει το ημίτονο του γινομένου τους.



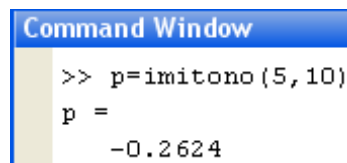
```

1 function p=imitono(a,b)
2     A=a*b;
3     p=sin(A);

```

Εικόνα 30

Και στη Εικόνα 31 παρουσιάζεται ο τρόπος που την καλούμε στο παράθυρο εντολών και ο τρόπος που αυτή εκτελείται και μας δίνει αποτέλεσμα.



```

>> p=imitono(5,10)
p =
    -0.2624

```

Εικόνα 31

Τέλος θα αναφέρουμε ότι και σε αρχείο εντολής αλλά και σε αρχείο συνάρτησης μπορούμε να εισάγουμε κάποια σχόλια, αυτό το επιτυγχάνουμε τοποθετώντας στην αρχή της έκφρασης που αποτελεί το σχόλιο μας το χαρακτήρα (%), στη περίπτωση που το σχόλιο μας ξεπεράσει τη μία γραμμή πρέπει στην αρχή κάθε γραμμής που αποτελεί σχόλιο να εισάγουμε τον παραπάνω χαρακτήρα.

1.6 Δομές Ελέγχου ροής

Το MatLab περιλαμβάνει κάποιες εντολές οι οποίες μας βοηθάνε να δημιουργήσουμε δικές μας ακολουθίες συναρτήσεων με τις οποίες να επιτύχουμε την επίλυση προβλημάτων υψηλού δείκτη δυσκολίας. Από αυτές τις εντολές εμείς θα επεκταθούμε στις εξής τρεις γιατί αυτές είναι που θα χρειαστούμε στα συστήματα αυτόματου ελέγχου :

- ✚ Η εντολή if
- ✚ Η εντολή for και

✚ Η εντολή `while`.

Τις εντολές αυτές συνηθίζεται να τις συντάσσουμε μέσα σε αρχεία συναρτήσεων, αυτό όμως δεν σημαίνει ότι δεν μπορούν να χρησιμοποιηθούν και απευθείας στο παράθυρο εντολών. Εμείς κατά τη διάρκεια του μαθήματος θα τις χρησιμοποιούμε μέσα σε αρχεία εντολών ή συναρτήσεων (`M-files`).

1.6.1 Εντολή `if`

Η εντολή `if` κάνει ακριβώς αυτό που λέει η ίδια η ονομασία της «εάν» ισχύει μια συνθήκη τότε εκτέλεσε την εκάστοτε εντολή, η μορφή της εντολής `if` είναι η εξής:

```
if συνθήκη
    εντολή;
end
```

Η εντολή ελέγχει τη συνθήκη και αν ικανοποιείται τότε εκτελείται η εντολή ενώ αν δεν ικανοποιείται η συνθήκη δεν εκτελείται η εντολή και η ροή του προγράμματος συνεχίζεται με τον τερματισμό της εντολής `if`. Η εντολή `if` μπορεί να περιέχει περισσότερες από μια εντολές τις οποίες τις γράφουμε τη μια κάτω από την άλλη.

```
if συνθήκη
    εντολή1;
    ....
    εντολή N;
end
```

Εκτός από την παραπάνω κλασική μορφή, η εντολή `if` μπορεί να έχει και την παρακάτω μορφή.

```
if συνθήκη
    εντολή 1;
else
    εντολή 2;
end
```

Σε αυτή τη περίπτωση ελέγχουμε τη συνθήκη και αν ικανοποιείται εκτελείται η *εντολή 1* αλλιώς αν δεν ικανοποιείται εκτελείται η εντολή που ακολουθεί την *else* δηλαδή η *εντολή 2* και στη συνέχεια τερματίζεται η εντολή `if`. Επίσης υπάρχει και ο συνδυασμός των δύο παραπάνω μορφών που έχουμε δει μέχρι στιγμής.

```
if συνθήκη
    εντολή 1;
elseif
    εντολή 2;
else
```

```
εντολή 3;
end
```

Ακολουθούμε την ίδια λογική που έχουμε αναφέρει στην εντολή *if*, αυτό που πρέπει να προσέξουμε είναι ότι το *elseif* γράφεται χωρίς κενό σαν να είναι μια λέξη.

1.6.2 Εντολή for

Η εντολή *for* επαναλαμβάνει την εκτέλεση μιας εντολής ή μιας σειράς εντολών για ένα αριθμό επαναλήψεων που έχουμε ορίσει με τη χρήση ενός μετρητή. Παρατηρώντας την μορφή της εντολής *for* θα γίνει πιο αντιληπτή η λειτουργία της.

```
for μετρητής = αρχική τιμή: βήμα μέτρησης: τελική τιμή
    εντολή1;
    ...
    εντολήN;
end
```

Κατανοούμε λοιπόν ότι η εντολή επαναλαμβάνεται για όλες τις τιμές του μετρητή και κάθε φορά ο μετρητής αυξάνεται κατά το βήμα μέτρησης που έχουμε ορίσει μέχρι και την τελική τιμή. Μπορούμε στην εντολή *for* να παραλείψουμε το βήμα μέτρησης, τότε σε αυτή τη περίπτωση το MatLab αντιλαμβάνεται ως βήμα μέτρησης την μονάδα.

1.6.3 Εντολή while

Η εντολή *while* θα μπορούσαμε να πούμε ότι αποτελεί ένα συνδυασμό των δύο παραπάνω εντολών. Η εντολή *while* επαναλαμβάνει μια εντολή ή μια σειρά εντολών για όσο ισχύει κάποια συνθήκη. Η μορφή της εντολή είναι η εξής :

```
while συνθήκη
    εντολή 1;
    ...
    εντολή N;
end
```

Στις παραπάνω εντολές αναφερθήκαμε σε συνθήκες οι οποίες έχουν άμεση σχέση με την εκτέλεση ή μη κάποιων εντολών. Ένας από τους τρόπους για να εκφράσουμε μια συνθήκη αποτελεί η χρήση σχεσιακών τελεστών,

βέβαια δεν αποτελούν το μόνο τρόπο σε περίπτωση που σε κάποιο κεφάλαιο χρησιμοποιηθεί κάποιος άλλος τρόπος θα υπάρξει επεξήγηση της λειτουργίας του. Στον παρακάτω πίνακα παρουσιάζονται σχεσιακοί τελεστές.

Σχεσιακοί τελεστές	
==	Έλεγχος ισότητας
~=	Έλεγχος ανισότητας
<	Μικρότερος
<=	Μικρότερος ίσος
>	Μεγαλύτερος
>=	Μεγαλύτερος ίσος

1.7 Βασικές συναρτήσεις MatLab

Στον παρακάτω πίνακα παρουσιάζονται οι σημαντικότερες συναρτήσεις και εντολές του MatLab τις οποίες πρόκειται να χρησιμοποιήσουμε καθώς και η περιγραφή λειτουργία της καθεμιάς. Εκτενέστερη ανάλυση των συναρτήσεων που ακολουθούν θα γίνει στα αντίστοιχα κεφάλαια που θα τις χρησιμοποιήσουμε.

Βασικές συναρτήσεις του MatLab	
sin(x)	Ημίτονο του x
cos(x)	Συνημίτονο του x
tan(x)	Εφαπτομένη του x
asin(x)	Τόξο ημιτόνου του x
acos(x)	Τόξο συνημίτονου του x
atan(x)	Τόξο εφαπτομένης του x
exp(x)	Εκθετική συνάρτηση (e^x)
log(x)	Συνάρτηση φυσικού λογαρίθμου ($\ln(x)$)
log10(x)	Συνάρτηση δεκαδικού λογαρίθμου του x
log2(x)	Συνάρτηση λογαρίθμου του x με βάση το 2
sqrt(x)	Τετραγωνική ρίζα του x
abs(x)	Απόλυτη τιμή του x
complex(x,y)	Σύνθεση μιγαδικών δεδομένων από πραγματικό (x) και φανταστικό μέρος (y)
conj(x)	Συζυγής μιγαδικός
imag(x)	Φανταστικό μέρος μιγαδικής ποσότητας
real(x)	Πραγματικό μέρος μιγαδικής ποσότητας
ans	Μεταβλητή συστήματος γενικής χρήσης
clear	Καθολικό σβήσιμο δεδομένων της οθόνης του τρέχοντος περιβάλλοντος εργασίας
help	Εμφάνιση θεμάτων βοήθειας
i	Φανταστική μονάδα ($\sqrt{-1}$)
j	
impulse(F)	Υπολογισμός της κρουστικής της συνάρτησης F
lsim(f,u,t)	Υπολογισμός της χρονικής απόκρισης ενός συστήματος f με είσοδο u και χρονικό διάστημα t
max(x)	Προσδιορίζει το μεγαλύτερο στοιχείο ενός πίνακα x
min(x)	Προσδιορίζει το μικρότερο στοιχείο ενός πίνακα x
pole(x)	Υπολογισμός των πόλων του συστήματος x
residue	Υπολογισμός ανάλυσης σε μερικά κλάσματα
rlocus	Υπολογισμός του γεωμετρικού τόπου ριζών
step(F)	Υπολογισμός της βηματικής απόκρισης ενός συστήματος

2. Εισαγωγή στο Simulink

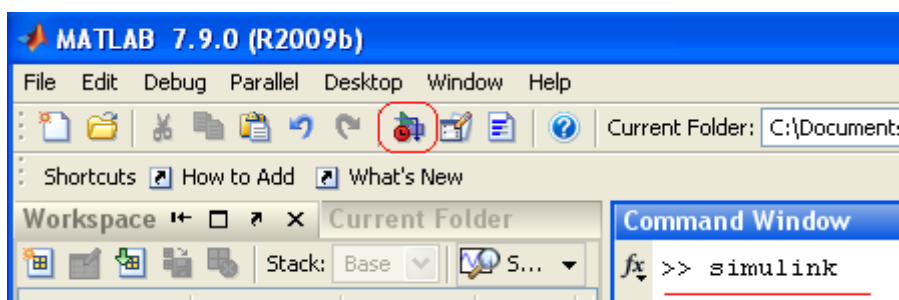
Το Simulink είναι ενσωματωμένο με το MatLab, είναι μια γραφική επέκταση για την μοντελοποίηση και προσομοίωση συστημάτων. Μας παρέχει άμεση πρόσβαση σε ένα ευρύ φάσμα εργαλείων που μας επιτρέπουν να αναπτύξουμε με ταχύτητα και ακρίβεια υπολογιστικά μοντέλα συστημάτων χρησιμοποιώντας μπλοκ διαγράμματα. Στο περιβάλλον του Simulink είναι εύκολο να υλοποιούνται πολύπλοκα μοντέλα μη γραμμικών συστημάτων, να επεξεργαζόμαστε αναλογικά ή ψηφιακά σήματα και είναι δυνατή η δημιουργία γραφικών παραστάσεων που δείχνουν τη συμπεριφορά της προσομοίωσης κάτι που μας βοηθά στην κατανόηση της συμπεριφοράς του φυσικού συστήματος.

Το Simulink όπως αναφέραμε είναι ενσωματωμένο με το MatLab και αυτό έχει ως αποτέλεσμα να υπάρχει άμεση πρόσβαση μεταξύ των δύο με αποτέλεσμα να μεγαλώνει το φάσμα των λειτουργιών που μπορεί να εκτελέσει. Στο πρόγραμμα υπάρχει μια βιβλιοθήκη με ποικιλία μπλοκ διαγραμμάτων για το σχεδιασμό, την προσομοίωση, την εφαρμογή και δοκιμή χρονικών μεταβαλλόμενων σημάτων εισόδου αλλά και δυνατότητα παρακολούθησης της εξόδου των συστημάτων κάτι που το καθιστά ένα πάρα πολύ καλό εργαλείο για την ανάλυση συστημάτων αυτόματου ελέγχου.

Για να εκκινήσουμε το πρόγραμμα του Simulink υπάρχουν δύο τρόποι.

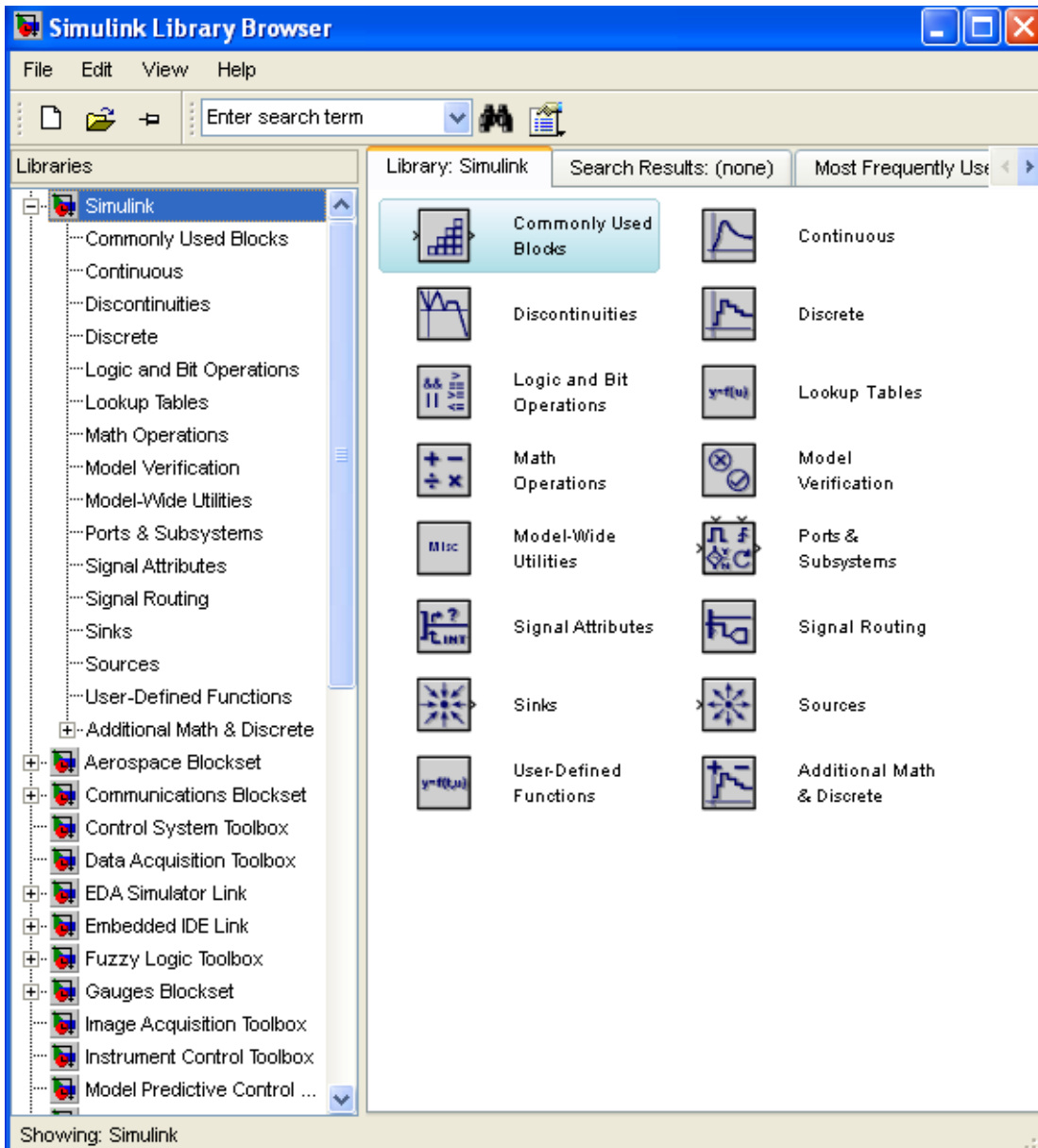
- ✚ Στο παράθυρο εντολών του MatLab πληκτρολογούμε την εντολή Simulink και πατάμε το πλήκτρο enter,
- ✚ Ο δεύτερος τρόπος είναι να πατήσουμε με τον κέρσορα στο εικονίδιο του Simulink που βρίσκεται στην γραμμή εργαλείων του MatLab.

Οι δύο αυτοί τρόποι παρουσιάζονται στην Εικόνα 1.



Εικόνα 1

Όταν ενεργοποιηθεί το πρόγραμμα του Simulink θα εμφανιστεί στην οθόνη μας το πρόγραμμα περιήγησης Simulink που είναι το εξής παράθυρο :

*Εικόνα 2*

Το παράθυρο παρατηρούμε ότι έχει δομή δέντρου και είναι το βασικό παράθυρο αναζήτησης αντικειμένων των βιβλιοθηκών του Simulink που είναι εγκατεστημένες στον υπολογιστή. Από όλες αυτές τις βιβλιοθήκες που παρουσιάζονται στο παράθυρο εμείς θα χρησιμοποιήσουμε κατά κύριο την βιβλιοθήκη που φαίνεται στην Εικόνα 2 επιλεγμένη μέσα σε μπλε πλαίσιο με την ονομασία Simulink. Βλέπουμε ότι περιέχει 16 υποκατηγορίες που περιέχουν μπλοκ διαγράμματα υπεραρκετά για τα μοντέλα που θα σχεδιάσουμε τόσο στα Συστήματα Αυτόματου Ελέγχου I αλλά και στα Συστήματα Αυτόματου Ελέγχου II. Για να δημιουργήσουμε ένα μοντέλο υπάρχουν δύο τρόποι.

- ✚ Στο menu εντολών του παραθύρου περιήγησης βιβλιοθηκών Simulink επιλέγουμε File → New → Model και στην συνέχεια ανοίγει ένα μικρότερο παράθυρο μέσα στο οποίο σχεδιάζουμε το μοντέλο.

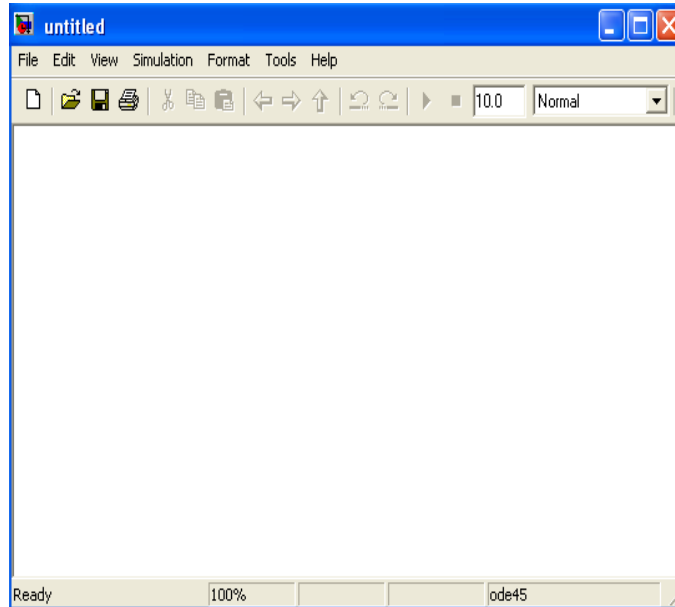
- ✚ Στο παράθυρο περιήγησης βιβλιοθηκών Simulink στη γραμμή εργαλείων και ακριβώς κάτω από την επιλογή File του menu εντολών υπάρχει ένα εικονίδιο μιας λευκής σελίδας, πατάμε αυτό το εικονίδιο και ανοίγει το παράθυρο σχεδίασης.

Ένα μοντέλο είναι ένα σύνολο από μπλοκ διαγράμματα που θα πρέπει να αντιπροσωπεύει σε γενικές γραμμές το σύστημα που θέλουμε να προσομοιώσουμε. Για να μεταφέρουμε στο νέο παράθυρο σχεδίασης μοντέλου τα μπλοκ διαγράμματα που επιθυμούμε είτε τα μετακινούμε με το ποντίκι από τη βιβλιοθήκη του Simulink είτε επιλέγοντας τα με δεξί κλικ στο ποντίκι επιλέγουμε την εντολή 'Add to untitled'. 'Untitled' είναι η ονομασία που δίνει το πρόγραμμα σε ένα μοντέλο που δεν έχουμε ακόμα ονομάσει, σε περίπτωση που ήδη έχουμε αποθηκεύσει το μοντέλο μας τότε στη θέση του 'untitled' θα υπάρχει η ονομασία που έχουμε δώσει στο μοντέλο μας.

Στη συνέχεια του κεφαλαίου θα παρουσιάσουμε την κατασκευή ενός πολύ απλού μοντέλου βήμα προς βήμα έτσι ώστε να βοηθήσουμε στην εξοικείωση του προσομοιωτή και του τρόπου που τον χειριζόμαστε.

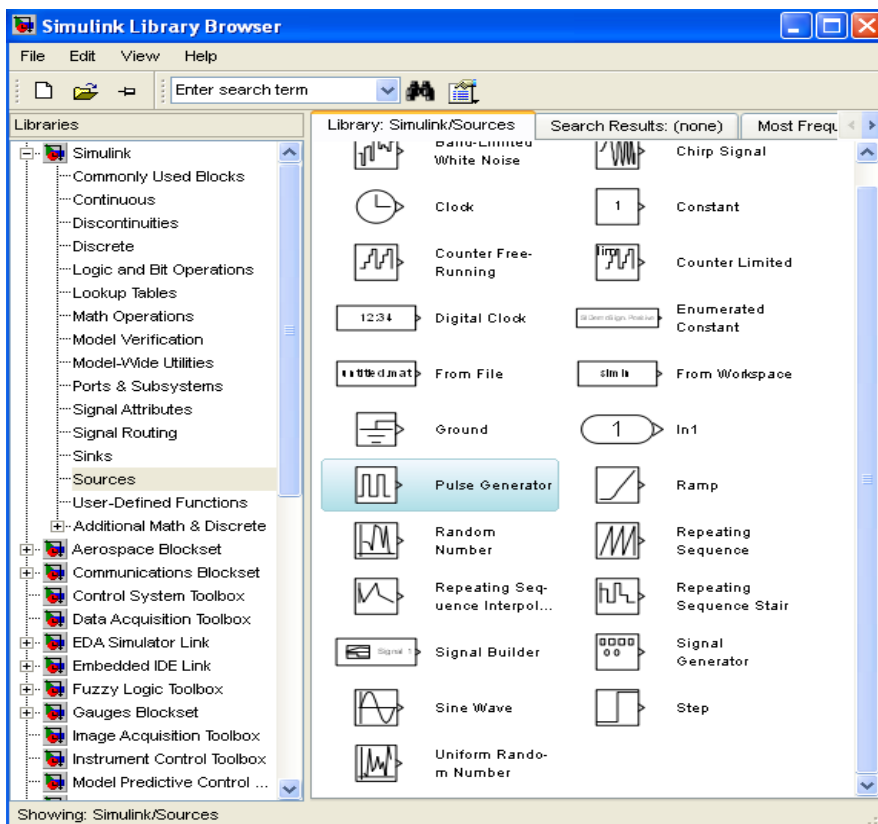
Το μοντέλο που θέλουμε να σχεδιάσουμε θέλουμε να αποτελείται από μια πηγή η οποία θα παράγει τετραγωνικό παλμό με πλάτος ίσο με 5 μονάδες, περίοδο 6 δευτερόλεπτα και ο παλμός να διαρκεί 3 δευτερόλεπτα. Τον παραγόμενο παλμό επιθυμούμε να τον παρατηρήσουμε με τη χρήση ενός παλμογράφου.

Πρώτο βήμα για τη δημιουργία του μοντέλου είναι το άνοιγμα ενός νέου παραθύρου σχεδίασης μέσα στο οποίο θα τοποθετήσουμε τα κατάλληλα μπλοκ διαγράμματα, τους τρόπους που θα ανοίξουμε το παράθυρο σχεδίασης τους έχουμε αναφέρει παραπάνω, επομένως το παράθυρο σχεδίασης όταν το ανοίξουμε είναι το παρακάτω.



Εικόνα 3

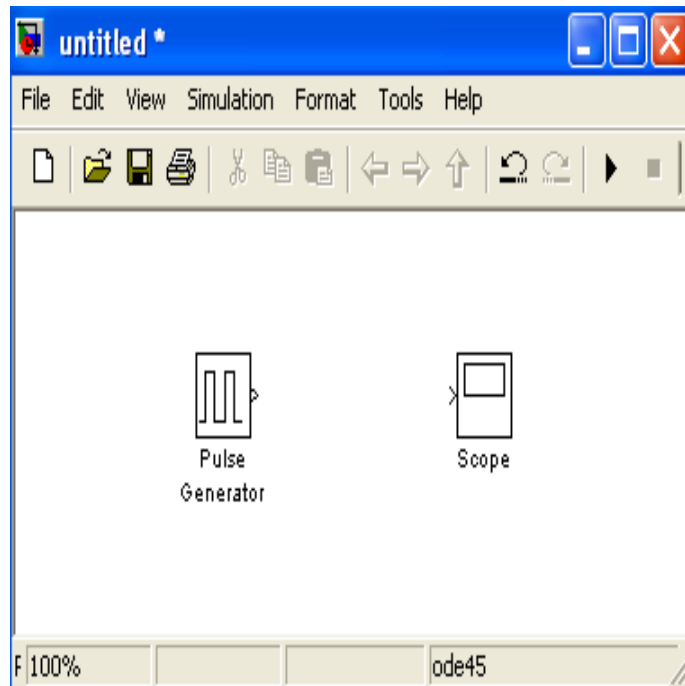
Στη συνέχεια πηγαίνουμε στο παράθυρο περιήγησης βιβλιοθηκών του Simulink και στη βιβλιοθήκη «Simulink» και στην υποκατηγορία «Sources» (πηγές) βρίσκουμε και τοποθετούμε (με έναν από τους δύο τρόπους που έχουμε προαναφέρει) στο μοντέλο το μπλοκ διάγραμμα με την ονομασία «Pulse Generator» (γεννήτρια παλμού) όπως παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 4

Για να εισάγουμε τον παλμογράφο στο μοντέλο επιλέγουμε την υποκατηγορία «Sinks» (σήματα) και επιλέγουμε το μπλοκ διάγραμμα «Scope»

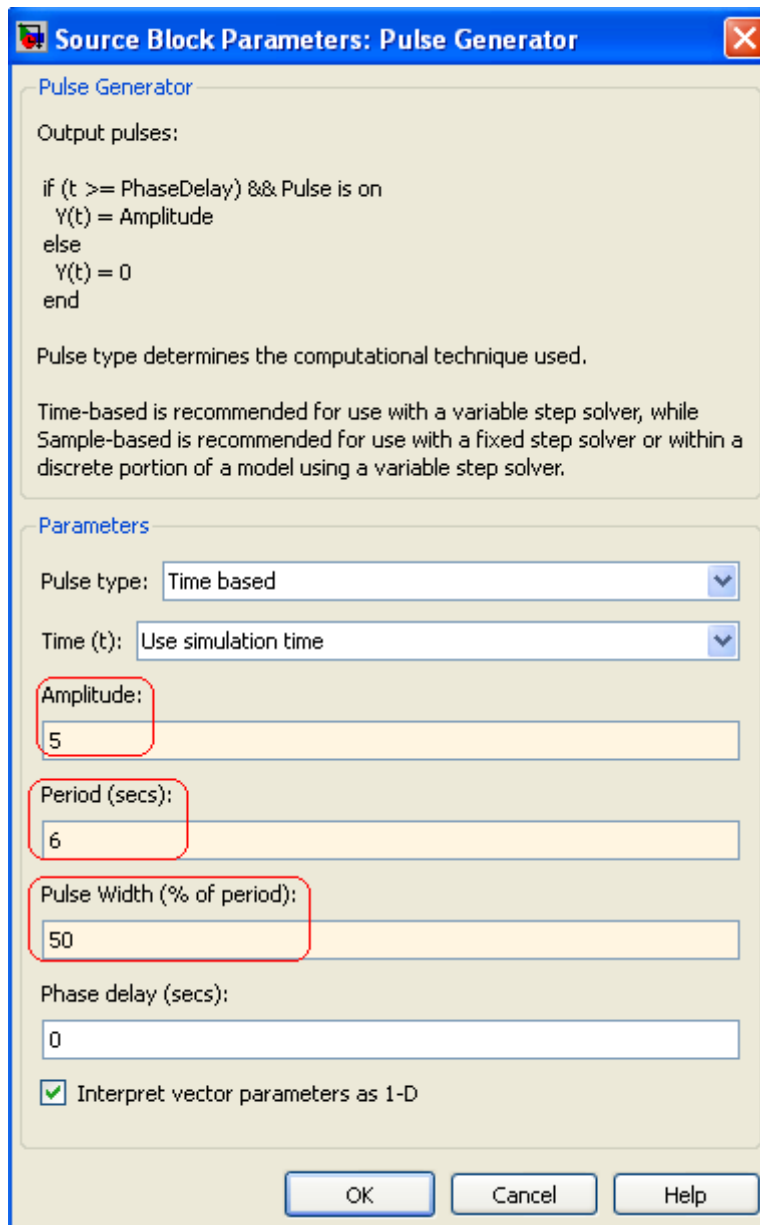
(παλμογράφος) το οποίο εισάγουμε στο μοντέλο το οποίο πλέον έχει την εξής μορφή.



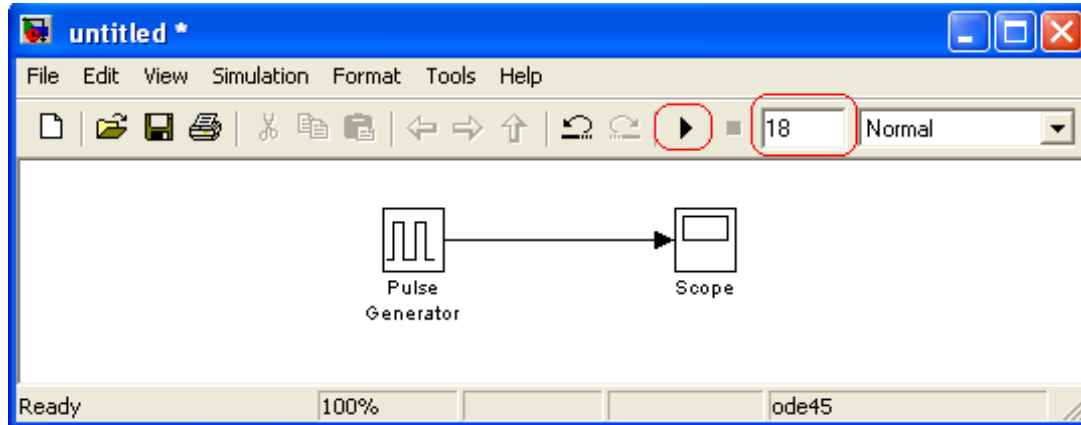
Εικόνα 5

Τα δύο μπλοκ διαγράμματα έχουν τοποθετηθεί στο παράθυρο σχεδίασης του μοντέλου και το μόνο που απομένει είναι να τα συνδέσουμε μεταξύ τους. Παρατηρούμε ότι και τα δύο μπλοκ διαγράμματα έχουν στο πλάι αλλά σε διαφορετικές πλευρές ένα βέλος, τραβώντας με το ποντίκι μια γραμμή πιέζοντας συνεχώς το αριστερό πλήκτρο ενώνουμε αυτά τα δύο βέλη και τα δύο μπλοκ διαγράμματα έχουν ενωθεί. Σε αυτό το σημείο θα πρέπει να πούμε ότι υπάρχουν μπλοκ διαγράμματα με περισσότερες από ένα βελάκι, αυτό που θα πρέπει να προσέξουμε είναι ότι το βελάκι άλλοτε έχει φορά προς το εσωτερικό του μπλοκ διαγράμματος και άλλοτε προς το εξωτερικό του. Τα βελάκια με φορά προς το εσωτερικό του μπλοκ αποτελούν θύρες εισόδου ενώ αυτά με προς το εξωτερικό αποτελούν τις θύρες εξόδου ενός μπλοκ διαγράμματος.

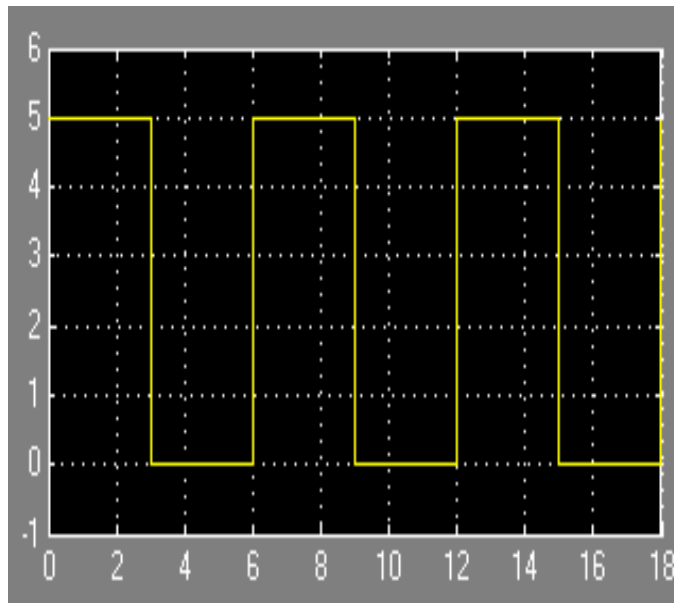
Αφού λοιπόν έχουμε συνδέσει τα δύο μπλοκ διαγράμματα ρυθμίζουμε, βάζοντας την τιμή που επιθυμούμε στο χωρίο, το χρόνο προσομοίωσης στα 18 δευτερόλεπτα. Κάθε μπλοκ διάγραμμα έχει τις δικές του ιδιότητες τις οποίες μπορούμε να ρυθμίσουμε όπως εμείς επιθυμούμε, κάνοντας διπλό «κλικ» πάνω στο εκάστοτε μπλοκ διάγραμμα εμφανίζεται το παράθυρο των ιδιοτήτων. Στο δικό μας μοντέλο εμείς θα αλλάξουμε τις ιδιότητες του «Pulse Generator» ώστε ο παλμός να έχει τα χαρακτηριστικά που θέλουμε και αναφέραμε στην αρχή του παραδείγματος, οι αλλαγές των ιδιοτήτων φαίνονται στην επόμενη εικόνα και παρατηρούμε ότι υπάρχουν αναλυτικές εξηγήσεις για κάθε παράμετρο.

Εικόνα 6

Πλέον είμαστε έτοιμοι και μπορούμε να τρέξουμε τη προσομοίωση του μοντέλου πατώντας το πλήκτρο «start», στην επόμενη εικόνα φαίνεται που ακριβώς είναι το πλήκτρο και το χωρίο μέσα στο οποίο τοποθετούμε το χρόνο που επιθυμούμε να διαρκέσει η προσομοίωση.

*Εικόνα 7*

Αφού τελειώσει η προσομοίωση τότε μπορούμε να επιλέξουμε το «Scope» με διπλό κλικ και μπορούμε να δούμε την παλμοσειρά που έχουμε δημιουργήσει και αν είναι αυτή που επιθυμούμε. Η παλμοσειρά που βλέπουμε μέσω του «Scope» παρουσιάζεται στην Εικόνα 8.

*Εικόνα 8*

Παρατηρούμε ότι ο παλμός έχει περίοδο 6 δευτερόλεπτα, πλάτος 5 και διάρκεια 3 δευτερόλεπτα άρα πληροί τις προδιαγραφές που θέσαμε στην αρχή.

Παρακολουθήσαμε λοιπόν μια απλή σχεδίαση ενός μοντέλου, η ίδια μεθοδολογία ακολουθούμε και για την σχεδίαση - μοντελοποίηση πολυπλοκότερων συστημάτων.

Είναι κατανοητό ότι στα 2 πρώτα κεφάλαια προσπαθήσαμε να δείξουμε τις βασικές και όχι όλες τις λειτουργίες του MatLab και του Simulink ώστε να βοηθηθούν οι σπουδαστές. Οποιαδήποτε εντολή, λειτουργία ή

μπλοκ διάγραμμα δεν έχει αναφερθεί αλλά χρησιμοποιηθεί στη συνέχεια του μαθήματος θα γίνεται ειδική αναφορά στο εκάστοτε κεφάλαιο.

3. Μαθηματικά Μοντέλα

Κατά τη μελέτη, την ανάλυση και το σχεδιασμό συστημάτων αυτόματου ελέγχου χρησιμοποιούμε μαθηματικά μοντέλα που αντικαθιστούν τα διάφορα φυσικά πολύπλοκα συστήματα. Η συμπεριφορά των φυσικών συστημάτων μπορεί να περιγραφεί με τη βοήθεια διαφορικών εξισώσεων επειδή τα διάφορα συστήματα που μας απασχολούν είναι συνήθως δυναμικά συστήματα. Οι διαφορικές εξισώσεις, που χρησιμοποιούμε στο ισοδύναμο μοντέλο του συστήματος που θέλουμε να μελετήσουμε, καταγράφονται με τη βοήθεια των φυσικών νόμων που διέπουν τη λειτουργία του συστήματος. Στη συνέχεια με διάφορα μαθηματικά εργαλεία βρίσκουμε μια λύση η οποία μας περιγράφει μαθηματικά τον τρόπο με τον οποίο λειτουργεί το σύστημα. Η διαδικασία με την οποία μελετάμε τα διάφορα δυναμικά συστήματα αποτελείται από τα εξής βήματα :

1. Ορίζουμε το σύστημα και τα στοιχεία που το αποτελούν.
2. Ορίζουμε το μαθηματικό μοντέλο του συστήματος.
3. Διατυπώνουμε τις διαφορικές εξισώσεις που περιγράφουν την λειτουργία του συστήματος.
4. Εξετάζουμε και μελετάμε τις λύσεις.
5. Αν δεν μας ικανοποιούν οι λύσεις και θεωρηθεί αναγκαίο αναλύουμε και επανασχεδιάζουμε το σύστημα από την αρχή.

Τα περισσότερα φυσικά συστήματα είναι γραμμικά. Γραμμικό θεωρούμε ένα σύστημα το οποίο ικανοποιεί τις αρχές της ομογένειας και της επαλληλίας. Παρά το γεγονός ότι τα περισσότερα συστήματα είναι γραμμικά σχεδόν όλα συμπεριφέρονται μη γραμμικά όταν οι μεταβλητές που επηρεάζουν τα συστήματα αυξάνονται απεριόριστα. Ένα πολύ χρήσιμο μαθηματικό εργαλείο για τη μελέτη και τη σχεδίαση γραμμικών μη χρονικά μεταβαλλόμενων συστημάτων, με τα οποία θα ασχοληθούμε κατά τη διάρκεια του μαθήματος, είναι ο μετασχηματισμός Laplace. Ο μετασχηματισμός Laplace μας βοηθά στην ευκολότερη επίλυση των εξισώσεων του μαθηματικού μοντέλου.

3.1 Μετασχηματισμός Laplace

Ο μετασχηματισμός Laplace ανάγει την λύση μίας δύσκολης γραμμικής διαφορικής εξίσωσης με σταθερούς συντελεστές σε λύση μιας αλγεβρικής εξίσωσης. Αυτό επιτυγχάνεται παίρνοντας τις διαφορικές εξισώσεις του μοντέλου που μελετάμε και υπολογίζουμε τον αντίστοιχο μετασχηματισμό Laplace των εξισώσεων αυτών, από αυτή τη διαδικασία προκύπτουν αλγεβρικές εξισώσεις τις οποίες επιλύουμε ως προς τη μεταβλητή που μας απασχολεί. Ο μετασχηματισμός Laplace μας δίνει τη δυνατότητα να αλλάξουμε το πεδίο ορισμού μιας συνάρτησης, μας επιτρέπει δηλαδή μια συνάρτηση

του χρόνου $f(t)$ να τη μεταβάλλουμε σε συνάρτηση της μιγαδικής συχνότητας $f(s)$. Αυτό μας επιτρέπει να μελετήσουμε τη συνάρτηση ενός συστήματος και στο πεδίο του χρόνου αλλά και στο πεδίο της μιγαδικής συχνότητας και να λάβουμε περισσότερες πληροφορίες και συμπεράσματα για τη συμπεριφορά της συνάρτησης άρα και του συστήματος.

Ο μετασχηματισμός Laplace μιας συνάρτησης $f(t)$ ορίζεται ως:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt$$

Όπου, $F(s)$ είναι η συνάρτηση στο πεδίο της μιγαδικής συχνότητας,

$f(t)$ είναι η συνάρτηση στο πεδίο του χρόνου,

$\mathcal{L}\{ \}$ συμβολίζει το μετασχηματισμό Laplace και

s είναι η μιγαδική μεταβλητή που ισούται με $s = \sigma + j\omega$ όπου σ πεπερασμένος θετικός αριθμός.

Ο μετασχηματισμός Laplace ορίζεται με την προϋπόθεση ότι ισχύει η σχέση:

$$\int_0^{\infty} f(t)e^{-st} dt < \infty$$

Ο μετασχηματισμός Laplace έχει πολλές ιδιότητες από τις οποίες εμείς θα αναφέρουμε τις πιο χρήσιμες για τη μελέτη συστημάτων. Όπως αναφέραμε τα συστήματα που θα μας απασχολήσουν είναι γραμμικά οπότε γίνεται κατανοητό ότι η γραμμικότητα του μετασχηματισμού είναι μια από τις ιδιότητες που το καθιστούν ένα σημαντικό εργαλείο για την ανάλυση μοντέλων. Σύμφωνα με τη γραμμικότητα ισχύει:

$$\text{Αν } \mathcal{L}\{f(t)\} = F(s) \text{ και } \mathcal{L}\{y(t)\} = Y(s) \text{ τότε}$$

$$\mathcal{L}\{c_1 f(t) + c_2 y(t)\} = \mathcal{L}\{c_1 f(t)\} + \mathcal{L}\{c_2 y(t)\} = c_1 F(s) + c_2 Y(s)$$

Όπου c_1 και $c_2 \in \mathbb{R}$.

Αν έχουμε μια συνάρτηση $f(t)$ τότε ο μετασχηματισμός Laplace της παραγώγου της συνάρτησης ($f'(t)$) δίνεται από τη σχέση:

$$\mathcal{L}\{f'(t)\} = sF(s) - f(0).$$

Ομοίως για το μετασχηματισμό Laplace της δεύτερης παραγώγου της συνάρτησης θα έχουμε:

$$\mathcal{L}\{f''(t)\} = s^2 F(s) - sf(0) - f'(0)$$

Προκύπτει λοιπόν η γενικευμένη σχέση για το μετασχηματισμό Laplace της n παραγώγου μιας συνάρτησης, η οποία είναι η εξής:

$$\begin{aligned} \mathcal{L}\{f^{(n)}(t)\} &= s^n F(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \dots - f^{(n-1)}(0) \leftrightarrow \\ &\leftrightarrow \mathcal{L}\{f^{(n)}(t)\} = s^n F(s) - \sum_{i=0}^{n-1} s^{n-i-1} \frac{d^i}{dt^i} f(0) \end{aligned}$$

Έστω μια συνάρτηση $f(t)y(t)$ η μετατοπισμένη προς τα δεξιά κατά t_0 μονάδες συνάρτηση θα είναι $f_{(t-t_0)}y_{(t-t_0)}$. Ο μετασχηματισμός Laplace της

αρχικής συνάρτησης και ο μετασχηματισμός Laplace της μετατοπισμένης συνάρτησης (μετατόπιση στο πεδίο του χρόνου) συνδέονται με τη σχέση:

$$\mathcal{L}\{f_{(t-t_0)}y_{(t-t_0)}\} = e^{-st_0}F(s)$$

Τέλος θα αναφερθούμε στην ιδιότητα αλλαγής κλίμακας στο πεδίο του χρόνου. Έστω οι συναρτήσεις $f_{(t)}$ και $f_{(at)}$, όπου a ένας θετικός αριθμός. Η συνάρτηση $f_{(at)}$ διαφέρει από την $f_{(t)}$ στην κλίμακα του χρόνου κατά a μονάδες, και ισχύει η σχέση:

$$\mathcal{L}\{f_{(at)}\} = \frac{1}{a}F\left(\frac{s}{a}\right)$$

Εκτός από τις παραπάνω ιδιότητες που θα μας βοηθήσουν για τη μελέτη συστημάτων υπάρχουν και τα θεωρήματα αρχικής και τελικής τιμής. Το θεώρημα αρχικής τιμής αναφέρεται στη συμπεριφορά μιας συνάρτησης $f_{(t)}$ για t τείνει στο μηδέν και διατυπώνεται με τη σχέση:

$$\lim_{t \rightarrow 0} f_{(t)} = \lim_{s \rightarrow \infty} sF(s)$$

Αντιστοίχως το θεώρημα τελικής τιμής αναφέρεται στη συμπεριφορά μιας συνάρτησης $f_{(t)}$ για t τείνει στο άπειρο και διατυπώνεται με τη σχέση:

$$\lim_{t \rightarrow \infty} f_{(t)} = \lim_{s \rightarrow 0} sF(s)$$

Βασικές Ιδιότητες Μετασχηματισμού Laplace	
$a * f_{(t)} \pm \beta * y_{(t)}$	$a * F(s) \pm \beta * Y(s)$
$f'_{(t)}$	$sF(s) - f_{(0)}$
$f''_{(t)}$	$s^2F(s) - sf_{(0)} - f'_{(0)}$
$f^n_{(t)}$	$s^nF(s) - s^{n-1}f_{(0)} - s^{n-2}f'_{(0)} - \dots - f^{n-1}_{(0)}$
$\int_0^t f_{(t)} dt$	$\frac{F(s)}{s} + \frac{f_{(0)}^{-1}}{s}$
$\int_0^t \int_0^t f_{(t)} dt$	$\frac{F(s)}{s} + \frac{f_{(0)}^{-1}}{s^2} + \frac{f_{(0)}^{-2}}{s}$
$f_{(t-t_0)}y_{(t-t_0)}$	$e^{-st_0}F(s)$
$\frac{f_{(t)}}{t}$	$\int_s^\infty F(\sigma) d\sigma$
$f_{(t)} * y_{(t)}$	$F(s) * Y(s)$
$e^{-at}f_{(t)}$	$F(s+a)$
$f\left(\frac{t}{a}\right)$	$aF(as)$
$f_{(at)}$	$\frac{F(s/a)}{a}$

3.2 Αντίστροφος Μετασχηματισμός Laplace

Ο μετασχηματισμός Laplace αναφέραμε ότι μετατρέπει μία συνάρτηση χρόνου σε συνάρτηση μιγαδικής συχνότητας, εκτός όμως από αυτή την μετατροπή θα χρειαστούμε και την αντίστροφη δηλαδή από το πεδίο της μιγαδικής συχνότητας στο πεδίο του χρόνου. Για αυτή τη μετατροπή θα χρησιμοποιήσουμε τον αντίστροφο μετασχηματισμό Laplace ο οποίος ορίζεται ως:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s)e^{+st} ds$$

Όπου, $\mathcal{L}^{-1}\{ \}$ συμβολίζει τον αντίστροφο μετασχηματισμό Laplace και

c είναι η πραγματική σταθερά που είναι μεγαλύτερη από τα πραγματικά μέρη όλων των πόλων της $F(s)$.

Η επίλυση του παραπάνω ολοκληρώματος είναι συχνά δύσκολη, γι' αυτό το λόγο για τον υπολογισμό του αντίστροφου μετασχηματισμού Laplace γίνεται με τη χρήση τύπων οι οποίοι αντιστοιχούν τιμές της $F(s)$ με τιμές της $f(t)$. Ακολουθεί σε επόμενη σελίδα πίνακας με κάποια από τα ζεύγη μετασχηματισμών Laplace.

Τα ζεύγη μετασχηματισμών όμως δεν καλύπτουν όλες τις περιπτώσεις εύρεσης του αντίστροφου μετασχηματισμού Laplace, επομένως σε περίπτωση που η συνάρτηση $F(s)$ δεν είναι σε μορφή που δίνεται στο πίνακα τότε χρησιμοποιούμε μεθόδους ώστε μέσω κατάλληλης μετατροπής να πάρει τέτοια μορφή η $F(s)$ ώστε ο αντίστροφος της να υπολογίζεται μέσω των πινάκων. Μια τέτοια μέθοδος είναι η ανάλυση σε απλά κλάσματα, η συγκεκριμένη μέθοδος ισχύει σε περιπτώσεις ρητών συναρτήσεων δηλαδή στο είδος συναρτήσεων με το οποίο θα ασχοληθούμε στη διάρκεια του μαθήματος.

Η ανάλυση σε απλά κλάσματα μας απλοποιεί την συνάρτηση $F(s)$ οπότε μας δίνει τη δυνατότητα να υπολογίσουμε τον αντίστροφο μετασχηματισμό Laplace της $F(s)$ απευθείας από τους πίνακες ζευγών. Ο λόγος που ακολουθούμε αυτή τη διαδικασία είναι γιατί στα Συστήματα Αυτόματου Ελέγχου συνήθως επιθυμούμε να προσδιορίσουμε την έξοδο ενός συστήματος όταν μας δίνεται συγκεκριμένη είσοδος, επομένως το πρόβλημα ανάγεται στον υπολογισμό ενός αντίστροφου μετασχηματισμού Laplace μιας ρητής συνάρτησης του s , την $F(s)$.

Έστω η εξής συνάρτηση:

$$F(s) = \frac{n(s)}{d(s)} = \frac{n_i s^i + n_{i-1} s^{i-1} + \dots + n_0}{d_k s^k + d_{k-1} s^{k-1} + \dots + d_0}$$

Με τις συναρτήσεις που θα ασχοληθούμε θα ισχύει $i < k$ δηλαδή ο βαθμός του πολυωνύμου του αριθμητή θα είναι μικρότερο από αυτόν του παρανομαστή, σε αντίθετη περίπτωση η μέθοδος ανάλυσης σε απλά κλάσματα δεν ισχύει.

Όπου $n(s)$ το πολυώνυμο του αριθμητή

$d_{(s)}$ το πολυώνυμο του παρανομαστή
 i ο βαθμός του πολυωνύμου του αριθμητή και
 k ο βαθμός του πολυωνύμου του παρανομαστή.

Η ανάλυση σε απλά κλάσματα της $F_{(s)}$ διαφέρει ανάλογα με τις ρίζες του πολυωνύμου του παρανομαστή. Οι περιπτώσεις που θα μας απασχολήσουν είναι η εξής δυο:

- ✚ Αν το πολυώνυμο του παρανομαστή έχει πλήθος n διαφορετικών ριζών, δηλαδή $\rho_1 \neq \rho_2 \neq \rho_3 \neq \dots \neq \rho_n$ τότε η ανάλυση σε απλά κλάσματα της $F_{(s)}$ θα είναι της μορφής:

$$F_{(s)} = \frac{n_{(s)}}{d_{(s)}} = \frac{c_1}{s - \rho_1} + \frac{c_2}{s - \rho_2} + \frac{c_3}{s - \rho_3} + \dots + \frac{c_n}{s - \rho_n} = \sum_{i=1}^n \frac{c_i}{s - \rho_i}$$

Όπου $c_1, c_2, c_3, \dots, c_n$ σταθερές οι οποίες προσδιορίζονται μέσω από τη μέθοδο ανάλυσης απλών κλασμάτων η οποία είναι γνωστή αφού περιλαμβάνεται στην ύλη του μαθήματος των Εφαρμοσμένων Μαθηματικών του 3ου εξαμήνου όπως προβλέπεται από το πρόγραμμα σπουδών του τμήματος. Ο λόγος που δεν την αναλύουμε είναι και το γεγονός ότι με τη χρήση του MatLab θα υπολογίζουμε άμεσα αυτές τις σταθερές και τις ρίζες του πολυωνύμου.

Αφού λοιπόν η $F_{(s)}$ έχει πάρει την μορφή που παρουσιάζουμε παραπάνω, χρησιμοποιώντας του πίνακες του μετασχηματισμού Laplace μπορούμε να υπολογίσουμε τον αντίστροφο μετασχηματισμό Laplace της $F_{(s)}$, δηλαδή την $f_{(t)}$:

$$f_{(t)} = \mathcal{L}^{-1}\{F_{(s)}\} = \mathcal{L}^{-1}\left\{\frac{c_1}{s - \rho_1} + \frac{c_2}{s - \rho_2} + \frac{c_3}{s - \rho_3} + \dots + \frac{c_n}{s - \rho_n}\right\} \leftrightarrow \\ \leftrightarrow f_{(t)} = c_1 e^{\rho_1 t} + c_2 e^{\rho_2 t} + c_3 e^{\rho_3 t} + \dots + c_n e^{\rho_n t}$$

- ✚ Αν όμως το πολυώνυμο του παρανομαστή έχει επαναλαμβανόμενη ή επαναλαμβανόμενες ρίζες τότε η ανάλυση σε απλά κλάσματα της $F_{(s)}$ θα είναι της μορφής:

$$F_{(s)} = \frac{n_{(s)}}{d_{(s)}} = \frac{c_1}{s - \rho_1} + \frac{c_2}{(s - \rho_1)^2} + \dots + \frac{c_k}{(s - \rho_1)^k} + \frac{c_{k+1}}{s - \rho_2} + \dots + \frac{c_n}{s - \rho_n} \leftrightarrow \\ \leftrightarrow F_{(s)} = \sum_{i=1}^n \sum_{j=1}^{k_i} \frac{c_{ij}}{(s - \rho_i)^j}$$

Χρησιμοποιώντας πάλι τους πίνακες βρίσκουμε την $f_{(t)}$:

$$f_{(t)} = \mathcal{L}^{-1}\{F_{(s)}\} = \mathcal{L}^{-1}\left\{\frac{c_1}{s - \rho_1} + \frac{c_2}{(s - \rho_1)^2} + \dots + \frac{c_k}{(s - \rho_1)^k} + \frac{c_{k+1}}{s - \rho_2} + \dots + \frac{c_n}{s - \rho_n}\right\} \leftrightarrow \\ \leftrightarrow f_{(t)} = c_1 e^{\rho_1 t} + c_2 t e^{\rho_1 t} + \dots + \frac{c_k}{(k - 1)!} t^{k-1} e^{\rho_1 t} + c_{k+1} e^{\rho_2 t} + \dots + c_n e^{\rho_n t}$$

Επομένως αναλόγως την περίπτωση καλούμαστε να επιλέξουμε το σωστό τρόπο επίλυσης της μεθόδου ανάλυσης σε απλά κλάσματα. Η χρήση του MatLab στις παραπάνω διαδικασίες απλοποιεί σε ένα μεγάλο βαθμό το δικό μας έργο αφού με τις ανάλογες εντολές το πρόγραμμα εκτελεί για εμάς όλες τις πολύπλοκες πράξεις, αλλά εμείς καλούμαστε να διαχειριστούμε με τον κατάλληλο τρόπο τα αποτελέσματα που λαμβάνουμε ώστε να καταλήξουμε στα σωστά συμπεράσματα κάθε φορά.

Ακολουθεί ο πίνακας με τα ζεύγη του μετασχηματισμού Laplace.

Ζεύγη Μετασχηματισμών Laplace	
$F(s)$	$f(t)$
1	$\delta(t)$
$\frac{1}{s}$	$u(t)$
$\frac{1}{s^2}$	t
$\frac{1}{s+a}$	e^{-at}
$\frac{1}{(s+a)^n}$	$\frac{1}{(n-1)!} t^{n-1} e^{-at}$
$\frac{\omega}{s^2 + \omega^2}$	$\sin \omega t$
$\frac{s}{s^2 + \omega^2}$	$\cos \omega t$
$\frac{n!}{s^{n+1}}$	t^n
$\frac{\omega}{(s+a)^2 + \omega^2}$	$e^{-at} \sin \omega t$
$\frac{(s+\alpha)}{(s+a)^2 + \omega^2}$	$e^{-at} \cos \omega t$
$\frac{1}{\sqrt{s}}$	$\frac{1}{\sqrt{\pi t}}$
$\frac{e^{-as}}{s}$	$u(t-\alpha)$
$\frac{1}{s}(1 - e^{-as})$	$u(t)-u(t-\alpha)$
$\frac{1}{s(s+a)}$	$\frac{1 - e^{-at}}{a}$
$\frac{1}{s(s+a)(s+\beta)}$	$\frac{1 - \frac{\beta}{\beta-\alpha} e^{-at} + \frac{\alpha}{\beta-\alpha} e^{-\beta t}}{a\beta}$
$\frac{s+\gamma}{s(s+a)(s+\beta)}$	$\frac{1}{\alpha\beta} (\gamma - \frac{\beta(\gamma-\alpha)}{\beta-\alpha} e^{-at} + \frac{\alpha(\gamma-\beta)}{\beta-\alpha} e^{-\beta t})$
$\frac{1}{(s+a)(s+\beta)}$	$\frac{e^{-at} - e^{-\beta t}}{\beta-\alpha}$

$\frac{s}{(s+a)(s+\beta)}$	$\frac{\alpha e^{-at} - \beta e^{-\beta t}}{\beta - \alpha}$
$\frac{s+\gamma}{(s+a)(s+\beta)}$	$\frac{(\gamma - \alpha)e^{-at} - (\gamma - \beta)e^{-\beta t}}{\beta - \alpha}$
$\frac{1}{(s+a)(s+\beta)(s+\gamma)}$	$\frac{e^{-at}}{(\beta - \alpha)(\gamma - \alpha)} + \frac{e^{-\beta t}}{(\gamma - \beta)(\alpha - \beta)} + \frac{e^{-\gamma t}}{(\alpha - \gamma)(\beta - \gamma)}$
$\frac{(s+\delta)}{(s+a)(s+\beta)(s+\gamma)}$	$\frac{(\delta - \alpha)e^{-at}}{(\beta - \alpha)(\gamma - \alpha)} + \frac{(\delta - \beta)e^{-\beta t}}{(\gamma - \beta)(\alpha - \beta)} + \frac{(\delta - \gamma)e^{-\gamma t}}{(\alpha - \gamma)(\beta - \gamma)}$
$\frac{1}{s(s^2 + \omega^2)}$	$\frac{1 - \cos \omega t}{\omega^2}$
$\frac{1}{(s+\alpha)^2 + \beta^2}$	$\frac{e^{-at} \sin \beta t}{\beta}$
$\frac{1}{s^2(s+\alpha)}$	$\frac{1}{\alpha^2} (\alpha t - 1 + e^{-at})$
$\frac{1}{s(s+\alpha)^2}$	$\frac{1}{\alpha^2} (1 - e^{-at} - \alpha t e^{-at})$
$\frac{s+\beta}{s(s+\alpha)^2}$	$\frac{1}{\alpha^2} (\beta - \beta e^{-at} + \alpha(\alpha - \beta)t e^{-at})$

Για τα μαθηματικά μοντέλα με τα οποία θα αντικαθιστούμε τα φυσικά συστήματα που θέλουμε να μελετήσουμε θα χρειαστούμε να γνωρίζουμε τις βασικές διαφορικές εξισώσεις των ηλεκτρικών στοιχείων και το μετασχηματισμό τους κατά Laplace.

Στοιχείο	Διαφορική εξίσωση στο πεδίο του χρόνου	Μετασχηματισμένη εξίσωση στο πεδίο της συχνότητας s
Ηλεκτρική αντίσταση	$u_{R(t)} = R * i_{(t)}$	$V_{R(s)} = R * I_{(s)}$
Πηνίο	$u_{L(t)} = L * \frac{di_{(t)}}{dt}$ $i_{(t)} = \frac{1}{L} * \int_0^t u_{L(t)} dt$	$V_{L(s)} = s * L * I_{(s)} - L * i_{(0)}$ $I_{(s)} = \frac{V_{L(s)}}{s * L} + \frac{i_{(0)}}{s}$
Πυκνωτής	$i_{(t)} = C * \frac{du_{C(t)}}{dt}$ $u_{C(t)} = \frac{1}{C} * \int_0^t i_{(t)} dt$	$I_{(s)} = s * C * V_{C(s)} - C * u_{C(0)}$ $V_{C(s)} = \frac{I_{(s)}}{s * C} + \frac{u_{C(0)}}{s}$

3.3 Μετασχηματισμός Laplace - MatLab

Ο λόγος που χρησιμοποιούμε τον μετασχηματισμό Laplace στα συστήματα αυτόματου ελέγχου είναι ότι μπορούμε να δημιουργήσουμε μια συνάρτηση η οποία να εκφράζει τη λειτουργία του συστήματος στο πεδίο της συχνότητας και εμείς να τη μετατρέψουμε στο πεδίο του χρόνου και να μας επιστρέψει το MatLab γράφημα με τη συμπεριφορά του συστήματος σε χρο-

νική περίοδο που εμείς έχουμε ορίσει. Μέσω αυτών των γραφημάτων μπορούμε να βγάλουμε ασφαλή συμπεράσματα για τη λειτουργία του πραγματικού συστήματος σε πραγματικό χρόνο.

3.4 Ασκήσεις Πράξης Μετασχηματισμού Laplace

Για να κατανοήσουμε τα του κεφαλαίου αλλά και τις εντολές του MatLab τις οποίες θα χρησιμοποιούμε στο εκάστοτε κεφάλαιο θα ακολουθούν στο τέλος του ασκήσεις πράξης.

Άσκηση 1: *Να βρεθούν οι αντίστροφοι μετασχηματισμοί Laplace των παρακάτω συναρτήσεων:*

$$\text{i) } F(s) = \frac{2}{s(s+1)(s+2)}$$

$$\text{ii) } F(s) = \frac{s+3}{(s+1)(s^2+4s+5)}$$

$$\text{iii) } F(s) = \frac{10}{(s+4)^3(s+3)^2}$$

$$\text{iv) } F(s) = \frac{2s^2-4}{(s+1)(s-2)^2(s-3)}$$

Θα αναλύσουμε βήμα-βήμα τις εντολές που πρόκειται να χρησιμοποιήσουμε ώστε να γνωρίζουμε το λόγο που τη χρησιμοποιήσαμε. Αρχικά θα πρέπει να εισάγουμε αυτές τις συναρτήσεις στο περιβάλλον του MatLab, για το λόγο αυτό θα χρησιμοποιήσουμε την εντολή `syms s` με την οποία δηλώνουμε ότι το s είναι συμβολική μεταβλητή (μπορούμε να χρησιμοποιήσουμε οποιοδήποτε γράμμα επιθυμούμε αλλά συνηθίζεται η χρήση του s). Δηλώνουμε τη συνάρτηση $F(s)$ και στη συνέχεια με την εντολή `numden` το MatLab ξεχωρίζει τον αριθμητή και το παρανομαστή της συνάρτησης. Με την εντολή `sym2poly` μετατρέπουμε το πολυώνυμο του αριθμητή και του παρανομαστή αντιστοίχως σε διπλό πολυώνυμο (double polynomial), ένα διάνυσμα δηλαδή με συντεταγμένες μόνο τους συντελεστές του πολυωνύμου. Αφού ορίσουμε τα διπλά πολυώνυμα αριθμητή και παρανομαστή της συνάρτησης καλούμε την εντολή `residue` η οποία εκτελεί την ανάλυση σε απλά κλάσματα και μας δίνει τις ρίζες και τους σταθερούς όρους. Αναλόγως τις ρίζες επιλέγουμε την κατάλληλη περίπτωση για να εκφράσουμε τα απλά κλάσματα του αντιστρόφου Laplace. Τέλος μέσω του πίνακα ζευγών υπολογίζουμε την $f(t)$.

```

i)  $F(s) = \frac{2}{s(s+1)(s+2)}$ 
>> syms s
>> F=2/(s*(s+1)*(s+2))
F =
2/(s*(s + 1)*(s + 2))
>> [num den]=numden(F)
num =
2
den =
s*(s + 1)*(s + 2)
>> nd=sym2poly(num)
nd =
2
>> dd=sym2poly(den)
dd =
1 3 2 0
>> [C R Z]=residue(nd,dd)
C =
1
-2
1
R =
-2
-1
0
Z =
[]

```

Παρατηρούμε ότι δεν υπάρχει κάποια επαναλαμβανόμενη ρίζα(R) οπότε η $F(s)$ σε απλά κλάσματα έχει τη μορφή:

$$F(s) = \frac{1}{s - (-2)} + \frac{-2}{s - (-1)} + \frac{1}{s - 0} = \frac{1}{s + 2} - \frac{2}{s + 1} + \frac{1}{s}$$

Οπότε μέσω του πίνακα η $f(t)$ ισούται με:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \mathcal{L}^{-1}\left\{\frac{1}{s+2} - \frac{2}{s+1} + \frac{1}{s}\right\} = \mathcal{L}^{-1}\left\{\frac{1}{s+2}\right\} - \mathcal{L}^{-1}\left\{\frac{2}{s+1}\right\} + \mathcal{L}^{-1}\left\{\frac{1}{s}\right\}$$

$$\Leftrightarrow f(t) = e^{-2t} - 2 * e^{-t} + u(t)$$

ii) $F(s) = \frac{s+3}{(s+1)(s^2+4s+5)}$

```

>> syms s
>> F=(s+3)/((s+1)*(s^2+4*s+5));
>> [num den]=numden(F);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
    -0.5000 - 0.0000i
    -0.5000 + 0.0000i
     1.0000
R =
    -2.0000 + 1.0000i
    -2.0000 - 1.0000i
    -1.0000
Z =
     []

```

Δεν έχουμε κάποια διπλή ρίζα άρα η $F(s)$ σε απλά κλάσματα έχει τη μορφή:

$$F(s) = \frac{-0.5}{s - (-2 + i)} + \frac{-0.5}{s - (-2 - i)} + \frac{1}{s - (-1)} \leftrightarrow$$

$$\leftrightarrow F(s) = -\frac{0.5}{s + 2 - i} - \frac{0.5}{s + 2 + i} + \frac{1}{s + 1}$$

Οπότε μέσω του πίνακα η $f(t)$ ισούται με:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \mathcal{L}^{-1}\left\{-\frac{0.5}{s + 2 - i} - \frac{0.5}{s + 2 + i} + \frac{1}{s + 1}\right\} \leftrightarrow$$

$$\leftrightarrow f(t) = -0.5 * e^{-(2-i)t} - 0.5 * e^{-(2+i)t} + e^{-t}$$


```

iii) F(s) = 10 / ((s+4)^3 * (s+3)^2)
>> syms s
>> F=10/((s+4)^3*(s+3)^2);
>> [num den]=numden(F);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
    30.0000
    20.0000
    10.0000
   -30.0000
    10.0000
R =
   -4.0000
   -4.0000
   -4.0000
   -3.0000
   -3.0000
Z =
     []

```

Σε αυτή τη περίπτωση παρατηρούμαι ότι έχουμε μια ρίζα το -4 επαναλαμβανόμενη 3 φορές και μια διπλή ρίζα το -3. Οπότε είναι διαφορετική περίπτωση ανάλυσης σε απλά κλάσματα από τις προηγούμενες δύο συναρτήσεις. Σε αυτή τη περίπτωση η $F(s)$ σε απλά κλάσματα θα έχει τη μορφή:

$$\begin{aligned}
 F(s) &= \frac{30}{s - (-4)} + \frac{20}{(s - (-4))^2} + \frac{10}{(s - (-4))^3} + \frac{-30}{s - (-3)} + \frac{10}{(s - (-3))^2} \leftrightarrow \\
 &\leftrightarrow F(s) = \frac{30}{s + 4} + \frac{20}{(s + 4)^2} + \frac{10}{(s + 4)^3} - \frac{30}{s + 3} + \frac{10}{(s + 3)^2}
 \end{aligned}$$

Οπότε μέσω του πίνακα η $f(t)$ ισούται με:

$$\begin{aligned}
 f(t) &= \mathcal{L}^{-1}\{F(s)\} = \mathcal{L}^{-1}\left\{\frac{30}{s + 4} + \frac{20}{(s + 4)^2} + \frac{10}{(s + 4)^3} - \frac{30}{s + 3} + \frac{10}{(s + 3)^2}\right\} \leftrightarrow \\
 \leftrightarrow f(t) &= 30 * e^{-4t} + 20 * t * e^{-4t} + \frac{10}{2} * t^2 * e^{-4t} - 30 * e^{-3t} + 10 * t * e^{-3t} \leftrightarrow \\
 &\leftrightarrow f(t) = 10 * e^{-4t}(3 + 2 * t + 0.5 * t^2) + 10 * e^{-3t}(t - 3)
 \end{aligned}$$

iv) $F(s) = \frac{2s^2-4}{(s+1)(s-2)^2(s-3)}$

```

>> syms s
>> F=(2*s^2-4)/((s+1)*(s-2)^2*(s-3));
>> [num den]=numden(F);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
    3.5000
   -3.5556
   -1.3333
    0.0556
R =
    3.0000
    2.0000
    2.0000
   -1.0000
Z =
    []

```

Βλέπουμε ότι υπάρχει μια διπλή ρίζα το -2, οπότε η $F(s)$ σε απλά κλάσματα θα έχει τη μορφή:

$$F(s) = \frac{3.5}{s-3} + \frac{-3.5556}{s-2} + \frac{-1.3333}{(s-2)^2} + \frac{0.0556}{s-(-1)} \leftrightarrow$$

$$\leftrightarrow F(s) = \frac{3.5}{s-3} - \frac{3.5556}{s-2} - \frac{1.3333}{(s-2)^2} + \frac{0.0556}{s+1}$$

Οπότε μέσω του πίνακα η $f(t)$ ισούται με:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \mathcal{L}^{-1}\left\{\frac{3.5}{s-3} - \frac{3.5556}{s-2} - \frac{1.3333}{(s-2)^2} + \frac{0.0556}{s+1}\right\} \leftrightarrow$$

$$\leftrightarrow f(t) = 3.5 * e^{3t} - 3.5556 * e^{2t} - 1.3333 * t * e^{2t} + 0.0556 * e^{-t}$$

Άσκηση 2: Να λυθεί το παρακάτω σύστημα διαφορικών εξισώσεων γνωρίζοντας ότι αρχικές συνθήκες $f(0) = 8$ και $y(0) = 3$:

$$f'(t) = 2 * f(t) - 3 * y(t)$$

$$y'(t) = y(t) - 2 * f(t)$$

Σύμφωνα με τις ιδιότητες Laplace έχουμε,

$$f'(t) = s * F(s) - f(0) \quad \text{και} \quad 2 * f(t) - 3 * y(t) = 2 * F(s) - 3 * Y(s)$$

$$\text{Άρα, } s * F_{(s)} - f_{(0)} = 2 * F_{(s)} - 3 * Y_{(s)} \leftrightarrow F_{(s)} = \frac{8-3*Y_{(s)}}{s-2} \quad (1)$$

$$y'_{(t)} = s * Y_{(s)} - y_{(0)} \text{ και } y_{(t)} - 2 * f_{(t)} = Y_{(s)} - 2 * F_{(s)}$$

$$\text{Άρα, } s * Y_{(s)} - y_{(0)} = Y_{(s)} - 2 * F_{(s)} \leftrightarrow Y_{(s)} = \frac{3-2*F_{(s)}}{s-1} \quad (2)$$

Από τις (1),(2) έχουμε :

$$F_{(s)} = \frac{8-3*\frac{3-2*F_{(s)}}{s-1}}{s-2} \leftrightarrow F_{(s)} = \frac{8-\frac{9+6*F_{(s)}}{s-1}}{s-2} \leftrightarrow F_{(s)} = \frac{8*s-17}{s^2-3*s-4}$$

Και

$$Y_{(s)} = \frac{3-2*\frac{8*s-17}{s^2-3*s-4}}{s-1} \leftrightarrow Y_{(s)} = \frac{3*s^2-25*s+22}{(s^2-3*s-4)(s-1)}$$

```
>> syms s
>> F=(8*s-17)/(s^2-3*s-4);
>> [num den]=numden(F);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
     3
     5
R =
     4
    -1
Z =
     []
```

$$\text{Επομένως, } F_{(s)} = \frac{3}{s-4} + \frac{5}{s-(-1)} = \frac{3}{s-4} + \frac{5}{s+1} \leftrightarrow f_{(t)} = \mathcal{L}^{-1}\{F_{(s)}\} = 3 * e^{4t} + 5 * e^{-t}$$

```
>> syms s
>> Y=(3*s^2-25*s+22)/((s^2-3*s-4)*(s-1));
>> [num den]=numden(Y);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
   -2.0000
    5.0000
    0.0000
R =
    4.0000
   -1.0000
    1.0000
Z =
     []
```

$$\text{Άρα, } Y_{(s)} = \frac{-2}{s-4} + \frac{5}{s-(-1)} + \frac{0}{s-1} \leftrightarrow y_{(t)} = \mathcal{L}^{-1}\{Y_{(s)}\} = -2 * e^{4t} + 5 * e^{-t}$$

Άσκηση 3: Να υπολογιστεί ο αντίστροφος Laplace των παρακάτω ρητών συναρτήσεων και να γίνει η γραφική αναπαράσταση για την χρονική περίοδο $Time=0:01:20$ (που σημαίνει από 0 έως 20 δευτερόλεπτα με βήμα 0,01 του δευτερολέπτου).

$$i) \frac{1}{(s+1)(s+2)(s+3)}$$

$$ii) \frac{1}{(s+1)(s+2)^2}$$

$$iii) \frac{2s^2+1}{s-5}$$

$$iv) \frac{1}{s^2+1}$$

$$v) \frac{s+1}{s(s^2+4)}$$

$$vi) \frac{1}{((s+1)^2+4)^2}$$

Για την επίλυση της άσκησης να χρησιμοποιηθεί το παρακάτω m-file το οποίο περιλαμβάνει τις παραπάνω διαδικασίες που γνωρίζουμε ώστε να μην είναι χρονοβόρα η διαδικασία. Το m-file μας επιστρέφει τον αντίστροφο Laplace και εμείς θα πρέπει να σχεδιάσουμε τη γραφική παράσταση.

```
function af=alapla(F, Tim)
[num den]=numden(F); %Σεχωρίζει τον αριθμητή και τον παρονομαστή της συνάρτησης
nd=sym2poly(num); %Μετατρέπει τον αριθμητή σε διπλό πολυώνυμο
dd=sym2poly(den); %Μετατρέπει τον παρονομαστή σε διπλό πολυώνυμο
[C R Z]=residue(nd, dd); %Ανάλυση σε απλά κλάσματα
if ~isempty(Z)
af='function is not proper'
%Δεν μπορεί να υπολογιστεί η ανάλυση σε απλά κλάσματα γιατί ο βαθμός του
%αριθμητή είναι μεγαλύτερος από αυτόν του παρονομαστή
else
K=polaplo(R); %Βρες αν υπάρχει επαναλαμβανόμενη ρίζα
%η συντεταγμένη i του K είναι η πολλαπλότητα της διαφορετικής ρίζας i του παρονομαστή
%sum(K)=numel(R)
af=Tim*0; %αρχική τιμή αντιστρόφου Laplace
for I=1:numel(K)
for J=1:K(I)
if I==1
JJ=J;
else
JJ=sum(K(1:I-1))+J;
end
af=af+C(JJ)*Tim.^ (J-1) .*exp(R(JJ)*Tim)/factorial(J-1);
end;end
end
```

Με τη βοήθεια του παραπάνω m-file έχουμε:

$$i) F(s) = \frac{1}{(s+1)(s+2)(s+3)}$$

```
>> Time=0:.01:20;
```

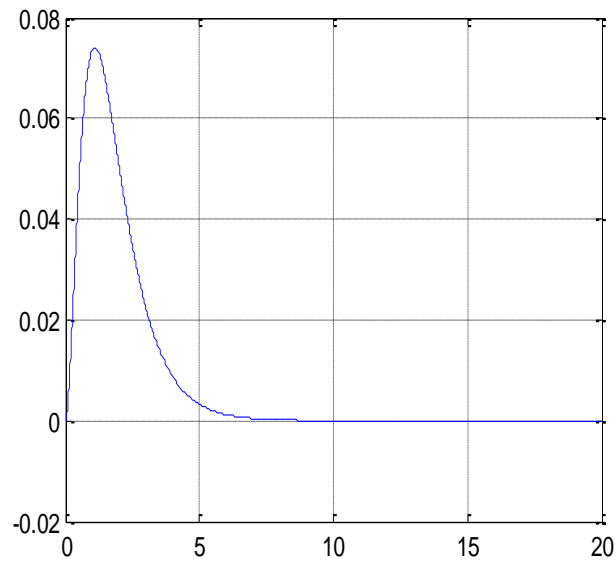
```
>> syms s
```

```
>> F1=1/((s+1)*(s+2)*(s+3));
```

```
>> af1=alapla(F1, Time);
```

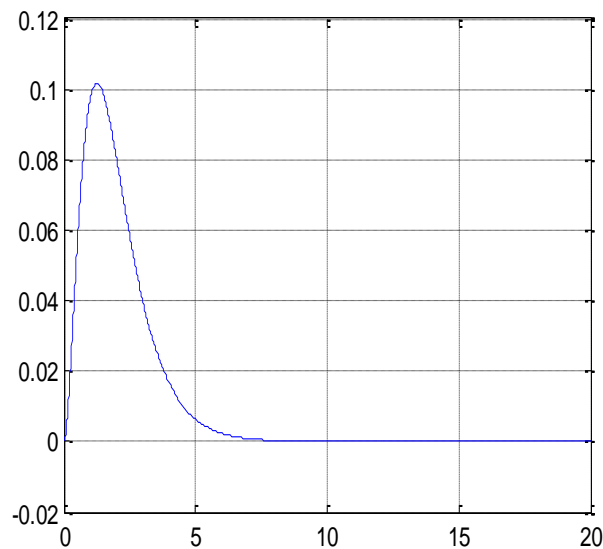
```
>> plot(Time, af1); grid on;
```

και το MatLab μας επιστρέφει την παρακάτω γραφική



ii) $F(s) = \frac{1}{(s+1)(s+2)^2}$

```
>> F2=1/((s+1)*(s+2)^2);
>> af2=alapla(F2,Time);
>> plot(Time,af2);grid on;
```



iii) $F(s) = \frac{2*s^2+1}{s-5}$

```
>> F3=(2*s^2+1)/(s-5);
>> af3=alapla(F3,Time);
```

af =

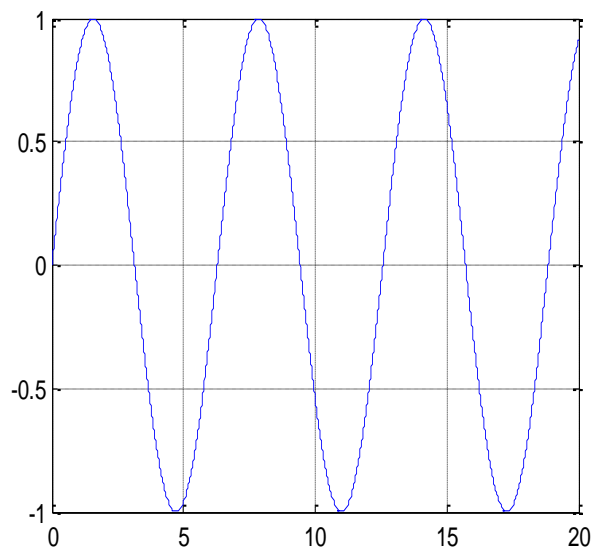
function is not proper

Το MatLab μας εμφανίζει το μήνυμα «η συνάρτηση δεν είναι κατάλληλη», αυτό συμβαίνει γιατί στη ρουτίνα που φτιάξαμε με το παραπάνω m-file ελέγχουμε αν ο βαθμός του πολυωνύμου του παρανομαστή είναι μεγαλύτερος από τον βαθμό του αριθμητή.

λύτερος από αυτόν του αριθμητή κάτι που αποτελεί αναγκαία προϋπόθεση για την ανάλυση σε απλά κλάσματα. Στη παραπάνω συνάρτηση η ανάλυση σε απλά κλάσματα δεν είναι δυνατή αφού ο βαθμός του πολυωνύμου του αριθμητή είναι μεγαλύτερος από του πολυωνύμου του παρανομαστή.

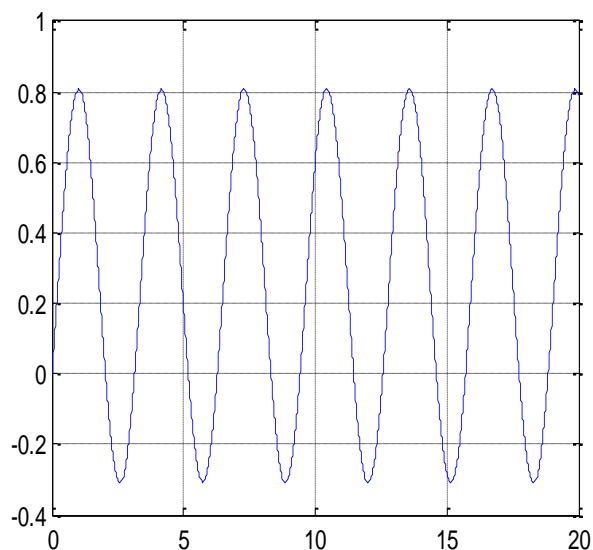
iv) $F(s) = \frac{1}{s^2+1}$

```
>> F4=1/(s^2+1);
>> af4=alapla(F4,Time);
>> plot(Time,af4);grid on;
```

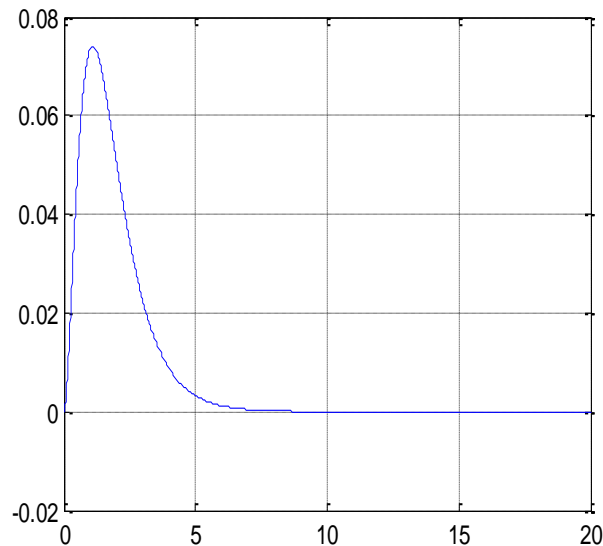


v) $F(s) = \frac{s+1}{s(s^2+4)}$

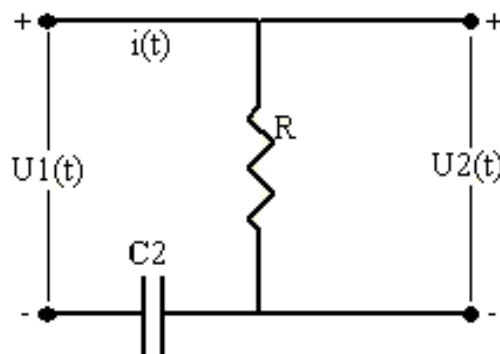
```
>> F5=(s+1)/(s*(s^2+4));
>> af5=alapla(F5,Time);
>> plot(Time,af5);grid on;
```



vi) $F(s) = \frac{1}{((s+1)^2+4)^2}$
 >> F6=1/(((s+1)^2+4)^2);
 >> af6=alapla(F6,Time);
 >> plot(Time,af);grid on;



Άσκηση 4: Στο κύκλωμα του παρακάτω σχήματος ζητείται να βρεθεί η τάση $u_{2(t)}$ στα άκρα της αντίστασης, γνωρίζοντας ότι $u_{1(t)} = 2 * e^{-t}$, $R = 1 \Omega$, $C = 1 F$ και $u_{C(0)} = 0 Volt$.



Από το ηλεκτρικό κύκλωμα του σχήματος προκύπτουν οι εξισώσεις:

$$u_{1(t)} = R * i(t) + \frac{1}{C} * \int_0^t i(t) dt$$

και

$$u_{2(t)} = R * i(t)$$

Μετασχηματίζοντας τις δύο παραπάνω σχέση κατά Laplace έχουμε:

$$V_{1(s)} = R * I(s) + \frac{I(s)}{s * C} = \frac{s * R * C + 1}{s * C} * I(s) \leftrightarrow I(s) = \frac{V_{1(s)} * s * C}{s * R * C + 1}$$

και

$$V_{2(s)} = R * I_{(s)}$$

Άρα,

$$V_{2(s)} = R * \frac{V_{1(s)} * s * C}{s * R * C + 1} = \frac{R * s * C}{s * R * C + 1} * V_{1(s)} = \frac{1 * s * 1}{s * 1 * 1 + 1} * V_{1(s)} \leftrightarrow$$

$$\leftrightarrow V_{2(s)} = \frac{s}{s + 1} * V_{1(s)}$$

Γνωρίζουμε την $u_{1(t)}$ επομένως από το πίνακα ζευγών του μετασχηματισμού Laplace μπορούμε να βρούμε την $V_{1(s)}$, η οποία ισούται $V_{1(s)} = \frac{2}{s+1}$.

Επομένως,

$$V_{2(s)} = \frac{s}{s + 1} * \frac{2}{s + 1} = \frac{2 * s}{(s + 1)^2}$$

Με τη βοήθεια του MatLab αναλύουμε τη $V_{2(s)}$ σε απλά κλάσματα.

```
>> syms s
>> F=2*s/(s+1)^2;
>> [num den]=numden(F);
>> nd=sym2poly(num);
>> dd=sym2poly(den);
>> [C R Z]=residue(nd,dd)
C =
     2
    -2
R =
    -1
    -1
Z =
     []
```

Άρα η $V_{2(s)} = \frac{2}{(s-(-1))} + \frac{-2}{(s-(-1))^2} \leftrightarrow V_{2(s)} = \frac{2}{(s+1)} - \frac{2}{(s+1)^2}$ και με τη βοήθεια του πίνακα ζευγών του μετασχηματισμού Laplace υπολογίζουμε τον αντίστροφο μετασχηματισμό Laplace της $V_{2(s)}$, οπότε η τάση στα άκρα της αντίστασης ισούται:

$$u_{2(t)} = 2 * e^{-t} - 2 * t * e^{-t}$$

4. Βασικά Σήματα Συστημάτων Αυτόματου Ελέγχου

Τις έννοιες ενός σήματος και ενός συστήματος τις συναντάμε σε πολλούς τομείς των τεχνολογικών και εφαρμοσμένων επιστημών. Αυτές οι βασικές έννοιες είναι απαραίτητες για την κατανόηση του αντικειμένου της μαθηματικής θεωρίας των συστημάτων.

Το σήμα αποτελεί μία μαθηματική συνάρτηση μίας ή περισσότερων ανεξάρτητων μεταβλητών μία από τις οποίες είναι υποχρεωτικά ο χρόνος και τυπικά περιέχει πληροφορίες για τη χρονική εξέλιξη μιας ποσότητας η οποία περιγράφει ένα φαινόμενο ή μία διαδικασία. Ένα σήμα συνεχούς χρόνου είναι μία πραγματική συνάρτηση $x(t) : \mathcal{R} \rightarrow \mathcal{R}$ της ανεξάρτητης μεταβλητής t η οποία εκφράζει το συνεχές χρόνο. Παραδείγματα σημάτων είναι η ηλεκτρική τάση $u(t)$ στους ακροδέκτες ενός ηλεκτρικού κυκλώματος ή η ένταση του ρεύματος $i(t)$ σε ένα κλάδο ηλεκτρικού κυκλώματος. Άλλα παραδείγματα σημάτων συνεχούς χρόνου είναι π.χ. η θέση $x(t)$ ή η ταχύτητα $v(t)$ ενός κινητού ως προς κάποια αρχή συντεταγμένων.

Με τον όρο σύστημα εννοούμε ένα μέρος του φυσικού κόσμου το οποίο θεωρούμε ότι αποτελείται από ένα σύνολο στοιχείων τα οποία λειτουργούν συγχρόνως κατά προδιαγεγραμμένο τρόπο έτσι ώστε να επιτυγχάνεται κάποιος στόχος.

Ένα σύστημα επικοινωνεί με το περιβάλλον μέσω σημάτων. Τα σήματα που δέχεται ένα σύστημα ονομάζονται διεγέρσεις ή εισοδοί και τα σήματα που παράγει ένα σύστημα λόγω των διεγέρσεων και των μη μηδενικών αρχικών συνθηκών ονομάζονται αποκρίσεις ή έξοδοι. Οι εισοδοί και οι έξοδοι των συστημάτων είναι συναρτήσεις του χρόνου. Οι συναρτήσεις που μας ενδιαφέρουν και έχουν ευρεία εφαρμογή στα συστήματα αυτόματου ελέγχου είναι γραμμικοί συνδυασμοί ορισμένων βασικών συναρτήσεων τις οποίες καλούμε βασικά σήματα, τα σήματα αυτά είναι οι εξής συναρτήσεις:

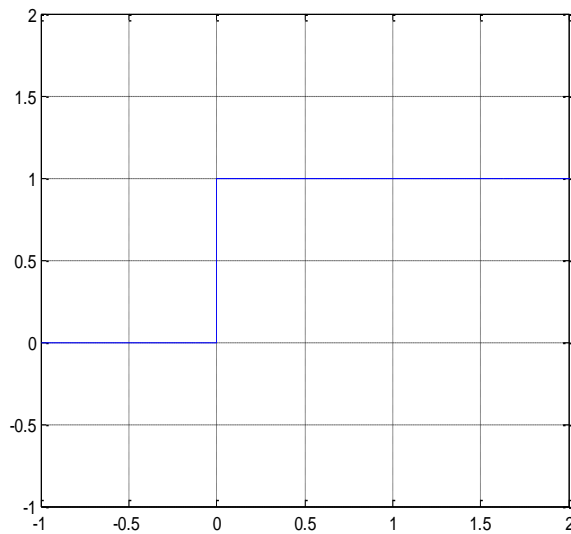
- ✚ Βηματική
- ✚ Κρουστική
- ✚ Αναρρίχηση
- ✚ Εκθετική
- ✚ Ημιτονοειδής

4.1 Βηματική(step) συνάρτηση

Η βηματική συνάρτηση συμβολίζεται με το $u(t)$ και συνήθως αναφέρεται και ως μοναδιαία βηματική συνάρτηση, όρος ο οποίος προέρχεται από το πλάτος της το οποίο είναι ίσο με τη μονάδα. Η βηματική συνάρτηση ορίζεται ως εξής:

$$u(t) = \begin{cases} 0, & -\infty < t \leq 0 \\ 1, & 0 < t < \infty \end{cases}$$

Η γραφική της βηματικής παρουσιάζεται στο παρακάτω γράφημα.

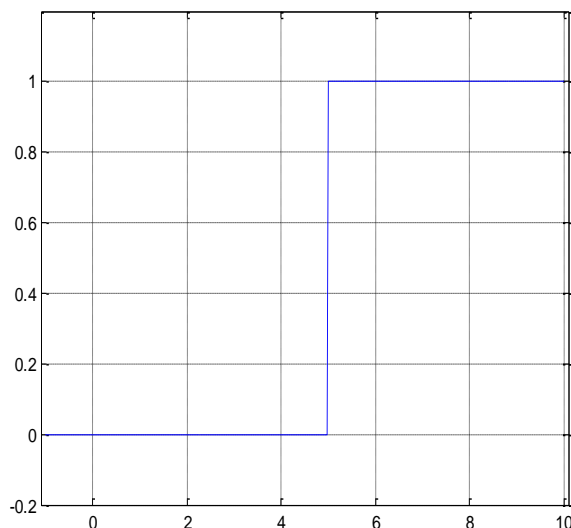


Ένα φυσικό παράδειγμα της βηματικής συνάρτησης είναι ο διακόπτης ενός δικτύου, αφού ο ρόλος που έχει ο διακόπτης εκφράζεται με τη συνάρτηση $u(\tau)$.

Η μοναδιαία βηματική εκτός από την παραπάνω μορφή μπορεί να παρουσιαστεί και με καθυστέρηση δηλαδή αντί να παίρνει την τιμή 1 μετά τη χρονική στιγμή $t = 0$ μπορούμε να ορίσουμε εμείς τη χρονική στιγμή όπου θα λαμβάνει την τιμή 1. Σε αυτή τη περίπτωση η βηματική ορίζεται:

$$u_{(t)} = \begin{cases} 0, & t \leq T \\ 1, & t > T \end{cases}$$

Όποτε για $T = 5$ η γραφική παράσταση της βηματικής είναι:



4.2 Κρουστική συνάρτηση

Η κρουστική συνάρτηση, η οποία ονομάζεται και συνάρτηση Dirac, συμβολίζεται με το $\delta_{(t)}$ και ορίζεται για κάθε συνάρτηση συνεχή στο μηδέν ως εξής:

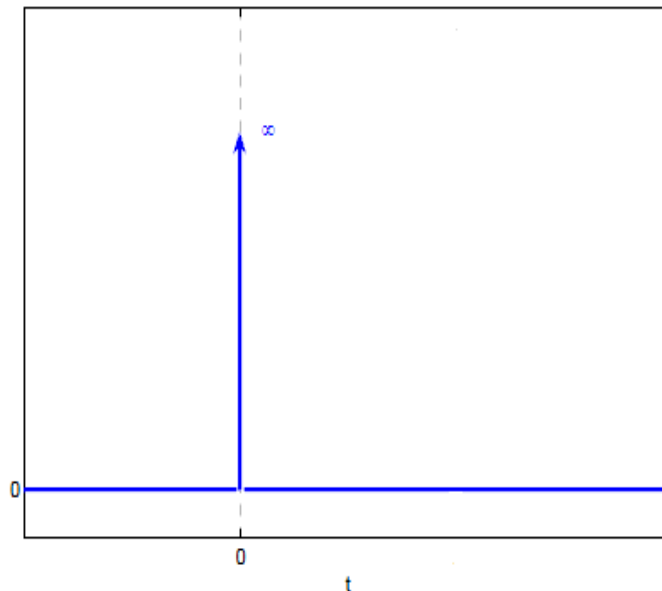
$$\delta_{(t)} = \begin{cases} 0, & t \neq 0 \\ \infty, & t = 0 \end{cases}$$

ή

$$\int_{-\infty}^{\infty} f_{(t)} * \delta_{(t)} * dt = f_{(0)}$$

Από φυσική άποψη η κρουστική συνάρτηση αναπαριστά την μεταφορά ενέργειας σε μηδενικό χρόνο. Από μαθηματική άποψη είναι μια συνάρτηση μηδενική παντού εκτός από το μηδέν όπου απειρίζεται. Η συνάρτηση αυτή έχει πάρα πολύ μεγάλη σημασία για την θεωρία ελέγχου των συστημάτων αλλά δεν μπορεί να υλοποιηθεί.

Η γραφική παράσταση που περιγράφει την κρουστική συνάρτηση ακολουθεί στο επόμενο γράφημα.

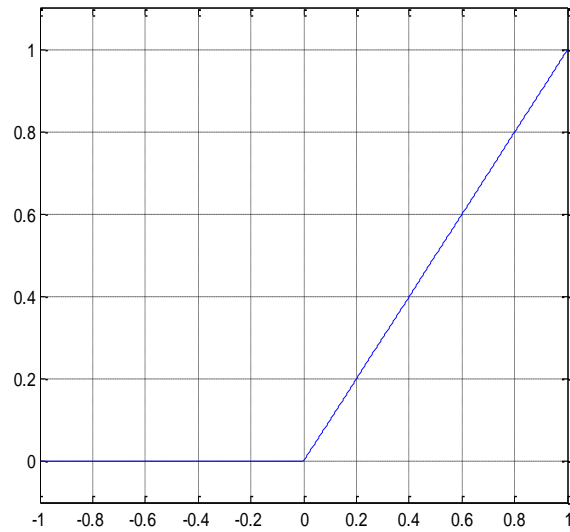


4.3 Συνάρτηση Αναρρίχησης

Η συνάρτηση αναρρίχησης συμβολίζεται με το $r_{(t)}$ και ορίζεται ως εξής:

$$r_{(t)} = \begin{cases} 0, & -\infty < t \leq 0 \\ t, & 0 < t < \infty \end{cases}$$

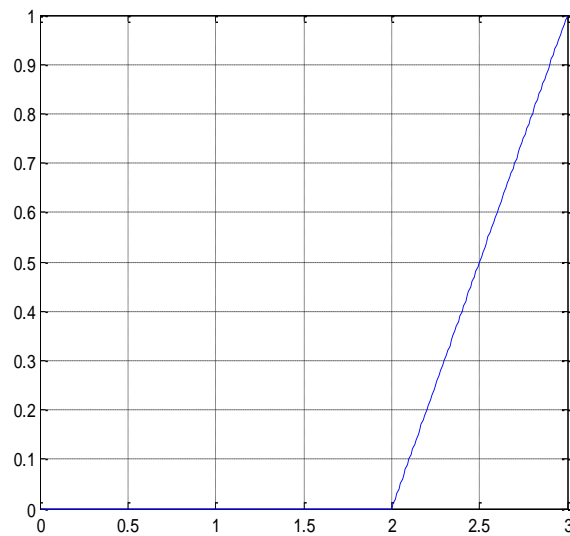
Η γραφική παράσταση που περιγράφει τη συνάρτηση αναρρίχησης είναι η εξής:



Όπως και βηματική η συνάρτηση αναρρίχησης εκτός από την παραπάνω μορφή μπορεί να παρουσιαστεί και με καθυστέρηση δηλαδή αντί να ξεκινάει η «αναρρίχηση» μετά το μηδέν μπορούμε να ορίσουμε εμείς τη χρονική στιγμή που θα ξεκινά και τότε η συνάρτηση αναρρίχησης ορίζεται:

$$r_{(t)} = \begin{cases} 0, & t \leq T \\ t, & t > T \end{cases}$$

Όποτε για $T = 2$ η γραφική παράσταση της αναρρίχησης είναι:



4.4 Εκθετική Συνάρτηση

Η γνωστή μαθηματική εκθετική συνάρτηση συμβολίζεται με $f(t) = A * e^{at}$, όμως στην πραγματικότητα η συνάρτηση που ενδιαφέρει τα συστήματα αυτόματου ελέγχου είναι η $f(t) = e^t * u(t)$.

Η εκθετική συνάρτηση παίζει σημαντικό ρόλο γιατί όλες οι συναρτήσεις που αναφέρονται παραπάνω είναι αποτελέσματα της χρήσης της εκθετικής. Η μοναδιαία βηματική συνάρτηση είναι η εκθετική όταν $A = 1$ και $a = 0$. Οι κρουστική συνάρτηση και η συνάρτηση αναρρίχησης παράγονται από τη βηματική συνάρτηση, η οποία όπως αναφέραμε προέρχεται από την εκθετική. Επίσης μια περιοδική συνάρτηση μπορεί να εκφραστεί ως γραμμικός συνδυασμός εκθετικών συναρτήσεων. Επιπλέον η εκθετική συνάρτηση περιγράφει πολλά φυσικά φαινόμενα όπως η απόκριση συστημάτων, η ακτινοβολία ραδιενεργών ισotόπων κ.α.

4.5 Ημιτονοειδής Συνάρτηση

Τέλος η γνωστή από τα μαθηματικά ημιτονοειδής συνάρτηση συμβολίζεται με $f(t) = A * \sin(\omega * t + \theta)$.

4.6 Βασικά Σήματα-MatLab

Ο λόγος που αναφέραμε τις παραπάνω συναρτήσεις είναι ότι αυτές θα αποτελέσουν τη βάση για να συνθέσουμε άλλες πιο πολύπλοκες που θα χρησιμοποιήσουμε ως σήματα διέγερσης στα συστήματα που θα προσομοιώσουμε. Αυτό μας βοηθάει στο να δούμε την συμπεριφορά του προσομοιωμένου κάθε φορά συστήματος ανάλογα με το σήμα εισόδου που θα διεγείρουμε.

Εξαιτίας ότι της συναρτήσεις αυτές θα τις χρησιμοποιήσουμε αρκετές φορές για τη δημιουργία σημάτων δημιουργήσαμε αρχεία m-file τα οποία θα μας επιστρέφουν την κάθε συνάρτηση όποτε την χρειαζόμαστε και με τα χαρακτηριστικά τα οποία επιθυμούμε να έχει. Τα m-files που θα χρησιμοποιούμε είναι τα εξής:

✚ Για την μοναδιαία βηματική

```
function u=aplistep(A)
u=A*0+1;
x=find(A<0);
u(x)=u(x)*0;
```

όπου A ο χρονικός ορίζοντας

- ✚ Για την βηματική με καθυστέρηση

```
function u=unistep(T,t)
u=T*0+1;
x=find(T<t);
u(x)=u(x)*0;
```

όπου T ο χρονικός ορίζοντας και t η χρονική καθυστέρηση

- ✚ Για την αναρρίχηση

```
function r=apriramp(T)
r=T;
x=find(T<0);
r(x)=r(x)*0;
```

όπου T ο χρονικός ορίζοντας

- ✚ Για την αναρρίχηση με καθυστέρηση

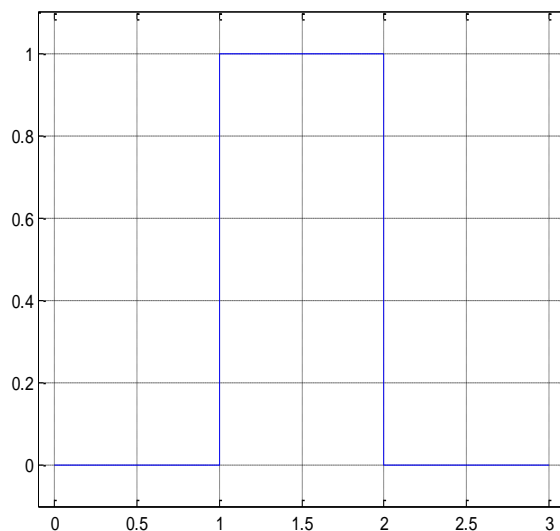
```
function r=uniramp(T,T0)
r=T-T0;
x=find(T<T0);
r(x)=r(x)*0;
```

όπου T ο χρονικός ορίζοντας και T_0 η χρονική καθυστέρηση

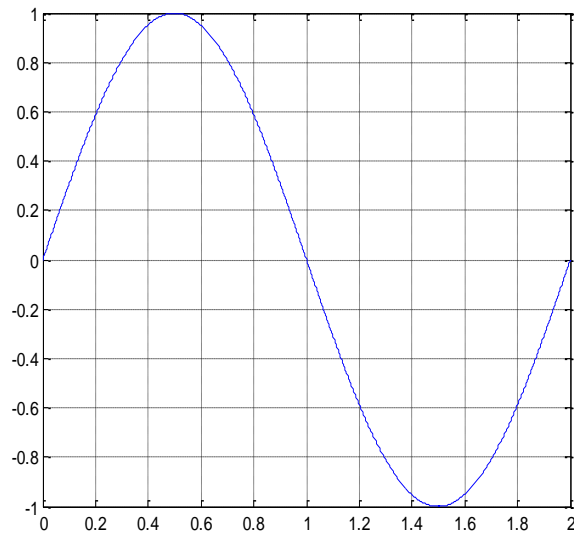
4.7 Ασκήσεις Πράξης Βασικών Σημάτων

Άσκηση 1: Με τη χρήση των βασικών σημάτων να δοθούν συναρτήσεις ώστε να παράγονται τα εξής γραφήματα:

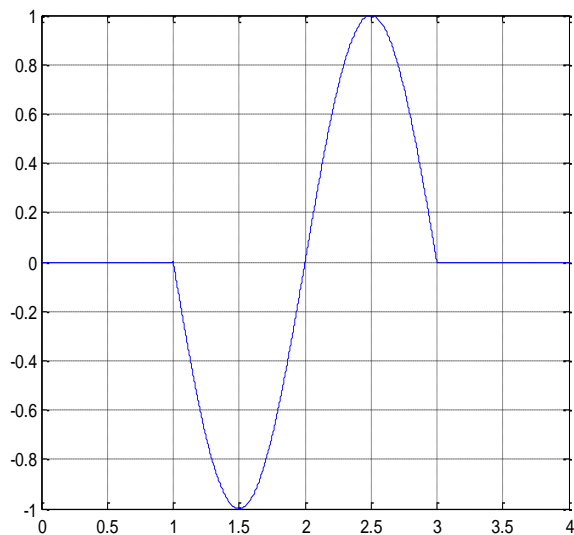
i)



ii)



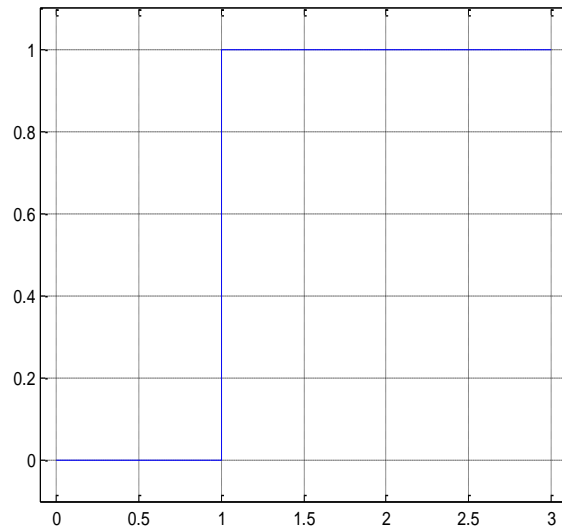
iii)



i) Παρατηρούμε ότι ο παλμός που μας ζητείται να δημιουργήσουμε μέχρι τη χρονική στιγμή 2 sec είναι μια βηματική συνάρτηση με $T=1$. Δίνουμε στο περιβάλλον του MatLab τις παρακάτω εντολές:

```
T=0:.001:3;
u=unistep(T,1);
plot(T,u);grid on;axis([-0.1 3.1 -0.1 1.1])
```

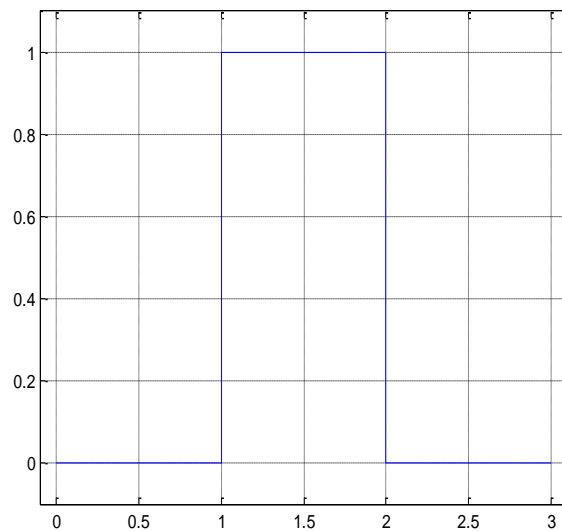
όπου δηλώνουμε το χρονικό ορίζοντα που επιθυμούμε και το βήμα αυτού και στη συνέχεια σχεδιάζουμε τη βηματική με $T=1$ και παίρνουμε το εξής γράφημα:



Βλέπουμε ότι ο παλμός που θέλουμε «κατεβαίνει» τη χρονική στιγμή στο μηδέν σε αντίθεση με αυτόν που έχουμε δημιουργήσει μέχρι στιγμής. Επομένως για να επιτύχουμε το σωστό παλμό θα πρέπει να τη χρονική στιγμή $T=2$ να αφαιρέσουμε μια «βηματική» ώστε να πάμε στο μηδέν.

```
u=unistep(T,1)-unistep(T,2);
plot(T,u);grid on;axis([-0.1 3.1 -0.1 1.1])
```

Μετά τις παραπάνω εντολές παίρνουμε το γράφημα που ακολουθεί στην επόμενη σελίδα.

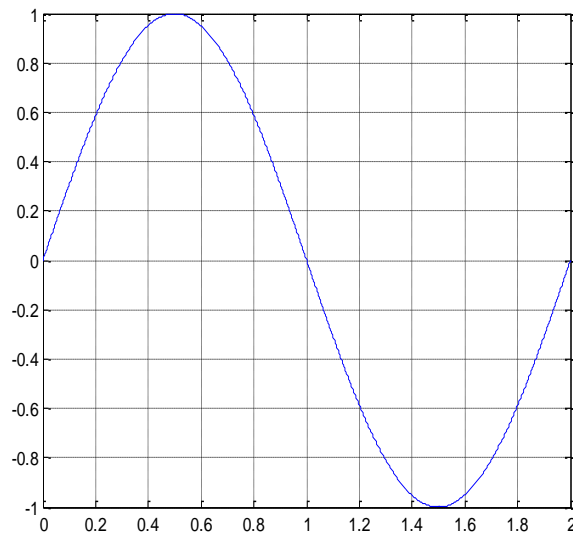


Ο παραπάνω παλμός είναι αυτός που μας ζητήθηκε να δημιουργήσουμε και αυτό το επιτύχαμε χρησιμοποιώντας τη βηματική συνάρτηση και πιο συγκεκριμένα αφαιρώντας μια βηματική από μια άλλη. Ανάλογους συνδυασμούς θα χρησιμοποιούμε για την δημιουργία παλμοσειρών.

ii) Η γραφική που μας ζητείται είναι αντιληπτό ότι είναι μια απλή ημιτονική παλμοσειρά η δημιουργία της οποίας επιτυγχάνεται με τις παρακάτω εντολές:

```
T=0:.001:2;  
p=sin(pi*T);  
plot(T,p);grid on;
```

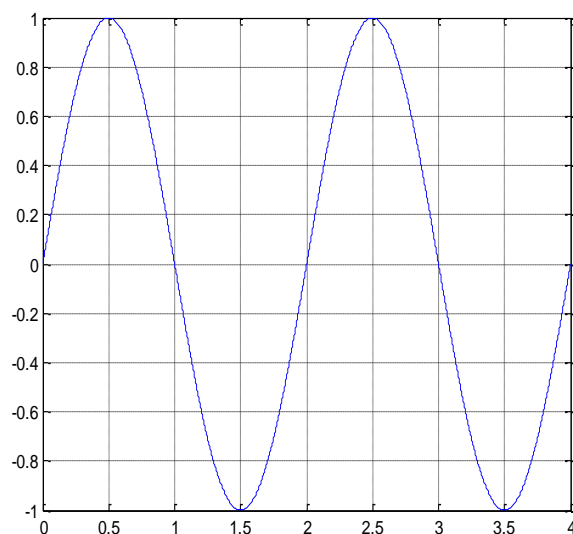
Με αποτέλεσμα το παρακάτω γράφημα που είναι αυτό που μας ζητήθηκε.



iii) Δίνουμε τις παρακάτω εντολές στο περιβάλλον του MatLab

```
T=0:.001:4;  
p=sin(pi*T);  
plot(T,p);grid on;
```

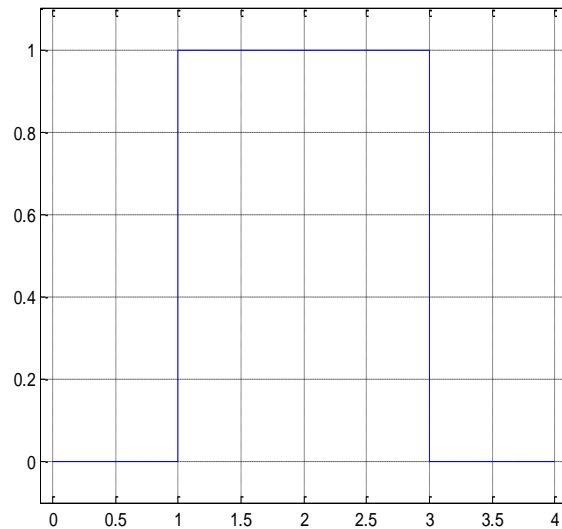
Αποτέλεσμα των οποίων είναι το παρακάτω γράφημα



Για να πετύχουμε το σωστό γράφημα θα πρέπει να μηδενιστούν οι περιοχές 0 έως 1 sec και 3 έως 4 sec. Σε αυτή την περίπτωση δεν μπορούμε να

κάνουμε ότι και προηγουμένως που αφαιρέσαμε μια συνάρτηση από μια άλλη γιατί θα έχουμε τελείως διαφορετικό αποτέλεσμα από το επιθυμητό. Θα πρέπει να βρούμε μια συνάρτηση που με την κατάλληλη πράξη να μας μηδενίζει την ημιτονική συνάρτηση στα διαστήματα που θέλουμε και να την αφήνει ως έχει στο υπόλοιπο διάστημα του χρονικού οριζοντα. Θέλουμε ένα παλμό λοιπόν όπως το πρώτο ερώτημα με τους κατάλληλους χρόνους.

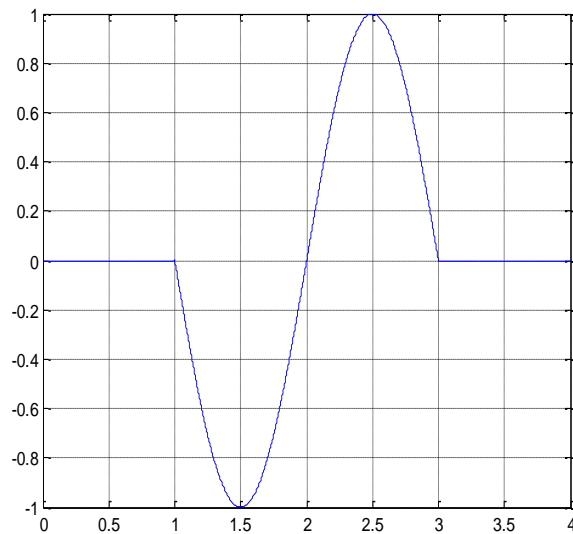
```
u=unistep(T,1)-unistep(T,3);
plot(T,u);grid on;axis([-0.1 4.1 -0.1 1.1])
```



Η πράξη που θα μας δώσει το σωστό γράφημα είναι ο πολλαπλασιασμός μεταξύ των δυο συναρτήσεων γιατί με αυτόν τον τρόπο στα διαστήματα 0-1 και 3-4 που ο παλμός έχει μηδενική τιμή θα μηδενίσει και την ημιτονική συνάρτηση ενώ στο διάστημα 1-3 που ο παλμός ισούται με 1 η ημιτονική συνάρτηση θα διατηρήσει την τιμή της, οπότε:

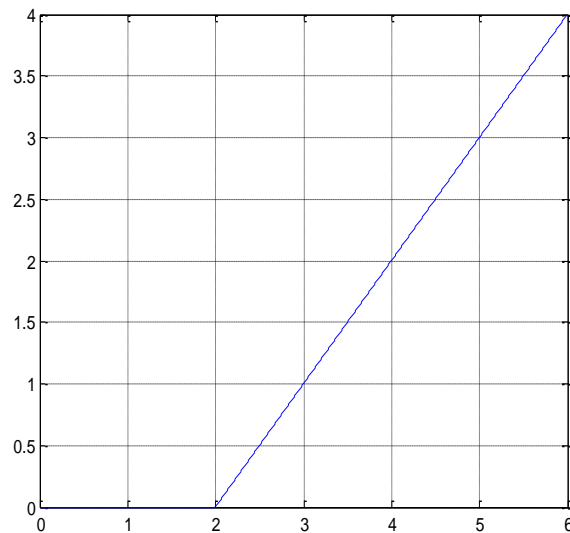
```
P=u.*p;
plot(T,P);grid on;
```

Η τελεία που τοποθετήθηκε πριν τον τελεστή του πολλαπλασιασμού είναι γιατί η δύο συναρτήσεις u και p είναι στην πραγματικότητα πίνακες που περιέχουν τις τιμές των συναρτήσεων για κάθε χρονική στιγμή τους οποίους εμείς θέλουμε να πολλαπλασιάσουμε στοιχείο προς στοιχείο, κάτι που δηλώνεται με αυτήν την τελεία.



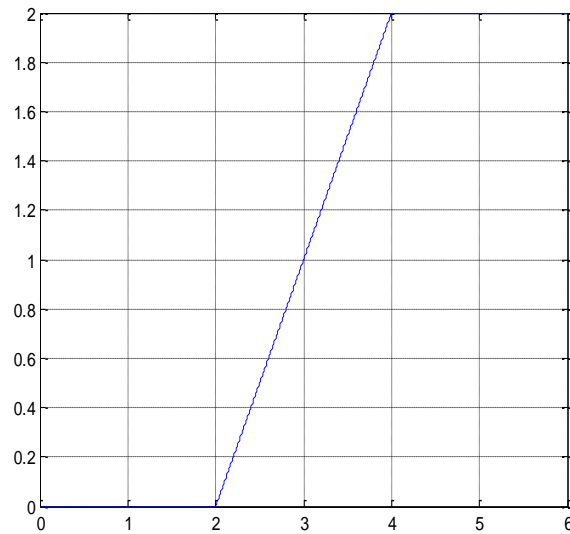
Άσκηση 2: Να δημιουργηθεί συνάρτηση που η γραφική της παράσταση να είναι τριγωνικός παλμός με πλάτος 2 δευτερόλεπτα και να ξεκινά τη χρονική στιγμή 2 δευτερόλεπτα. Στη συνέχεια να δημιουργηθεί νέα συνάρτηση τριγωνικό παλμό με πλάτος 3 δευτερόλεπτα και να ξεκινά από τη χρονική στιγμή 1.5 δευτερόλεπτα. Χρονικός ορίζοντας άσκησης 6 δευτερόλεπτα.

```
T=0:.0001:6;
tr=uniramp(T,2);
plot(T,tr);grid on;
```



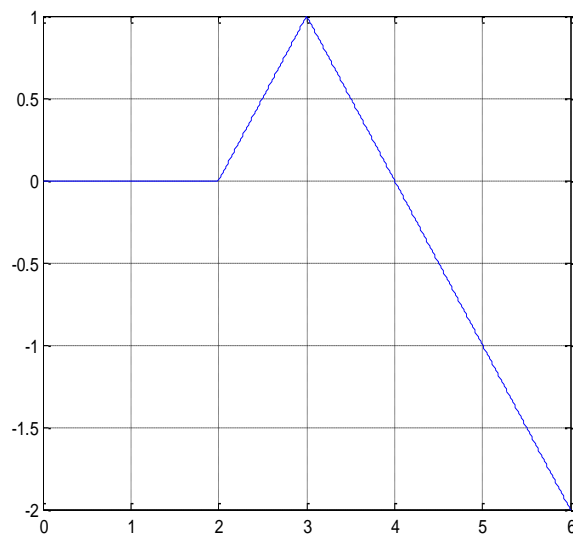
Πετύχαμε με τις παραπάνω εντολές ο τριγωνικός παλμός να ξεκινά τη χρονική στιγμή 2 δευτερόλεπτα. Αφού ο παλμός θα έχει πλάτος δύο δευτερόλεπτα τη χρονική στιγμή 4 δευτερόλεπτα θέλουμε να έχει τη τιμή μηδέν.

```
tr=uniramp(T,2)-uniramp(T,4);
plot(T,tr);grid on;
```



Παρατηρούμε στο παραπάνω γράφημα ότι τη χρονική στιγμή 4 δευτερόλεπτα αφαιρεί την κλίση που εισάγαμε τη χρονική στιγμή 2 δευτερόλεπτα. Άρα με βάση τις εντολές που δώσαμε αν αφαιρέσουμε μια συνάρτηση αναρρίχησης από μια άλλη στην ουσία αφαιρούμε κλίση ομοίως και στη περίπτωση της πρόσθεσης. Από το παραπάνω γράφημα αντιλαμβανόμαστε ότι η χρονική στιγμή που επιλέξαμε δεν ήταν η σωστή αφού θέλουμε η γραφική να έχει τιμή 0 τη χρονική στιγμή 4 δευτερόλεπτα. Οπότε τη χρονική στιγμή 3 δευτερόλεπτα θα πρέπει να αφαιρέσουμε 2 και όχι μία συναρτήσεις αναρρίχησης αφού εμείς θέλουμε να αποκτήσουμε αρνητική κλίση προς το μηδέν.

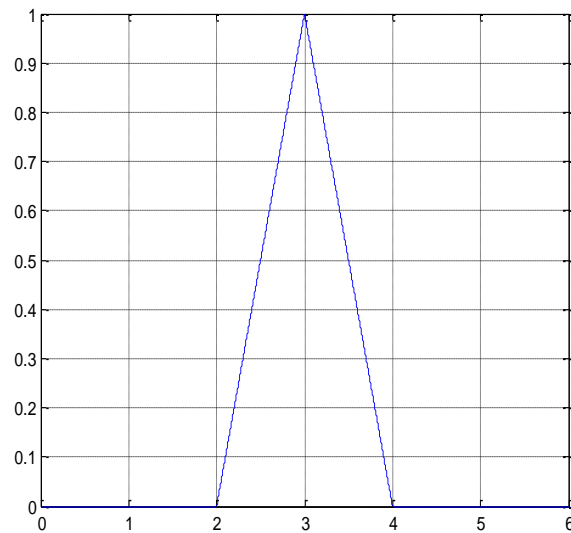
```
tr=uniramp(T,2)-2*uniramp(T,3);
plot(T,tr);grid on;
```



Τέλος μας μένει να σταθεροποιήσουμε τη συνάρτηση ώστε μετά τη χρονική στιγμή 4 δευτερόλεπτα να έχει σταθερή τιμή ίση με το μηδέν. Αυτό επιτυγχάνεται προσθέτοντας αυτή τη χρονική στιγμή κλίση, οπότε η τελική συνάρτηση που μας δίνει τον απαιτούμενο τριγωνικό παλμό είναι:

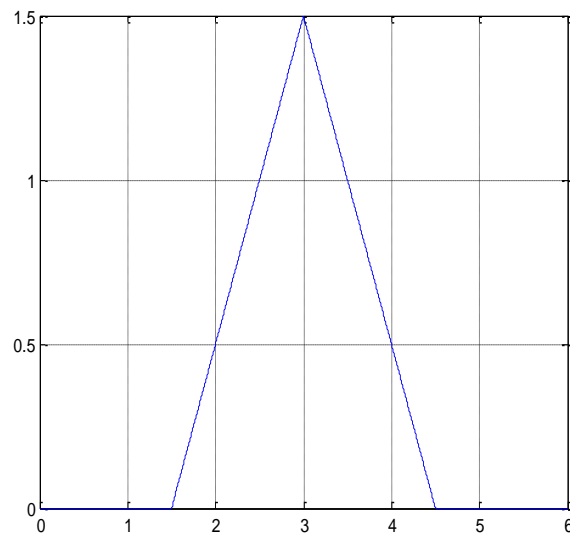
```
tr=uniramp(T,2)-2*uniramp(T,3)+uniramp(T,4);
plot(T,tr);grid on;
```

Και η γραφική της συνάρτησης είναι η εξής:



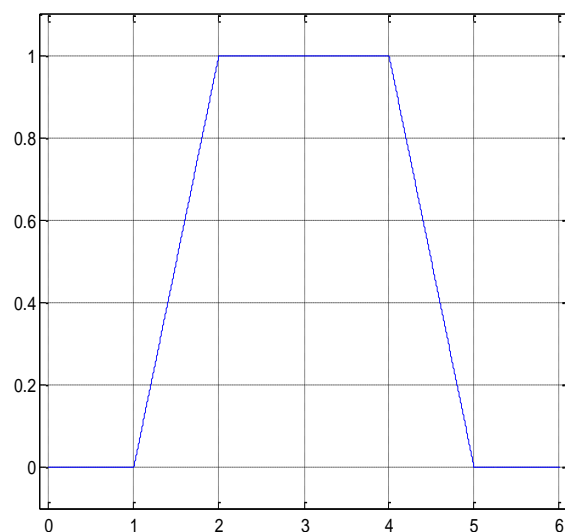
Δουλεύοντας με την ίδια ακριβώς μεθοδολογία για την κατασκευή του δεύτερου τριγωνικού παλμού θα έχουμε μια συνάρτηση αναρρίχησης με $T=1.5$ ώστε να πετύχουμε την εκκίνηση στα 1.5 δευτερόλεπτα, στη συνέχεια και αφού ο παλμός θα έχει πλάτος 3 δευτερόλεπτα στα 3 δευτερόλεπτα που είναι η κορυφή του τριγωνικού παλμού θα αφαιρέσουμε 2 φορές μια συνάρτηση αναρρίχησης και τη χρονική στιγμή 4.5 δευτερόλεπτα θα προσθέσουμε μια συνάρτηση αναρρίχησης ώστε να σταθεροποιηθεί στη τιμή μηδέν. Για να επαληθεύσουμε τα παραπάνω δίνουμε στο MatLab τις εντολές:

```
tri=uniramp(T,1.5)-2*uniramp(T,3)+uniramp(T,4.5);
plot(T,tri);grid on;
```



Άσκηση 3: Να αναλυθούν οι παρακάτω παλμοί και να δοθούν οι συναρτήσεις που τους περιγράφουν με τη χρήση των βασικών σημάτων.

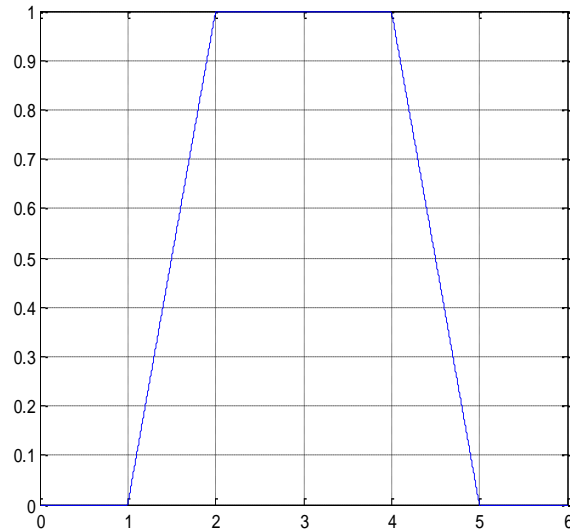
i)



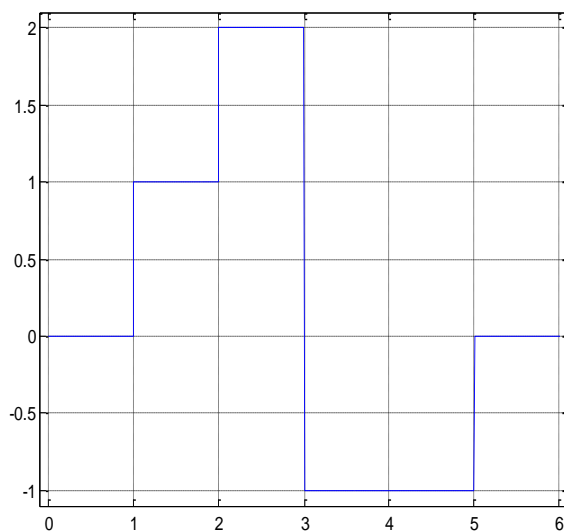
Παρατηρώντας το παλμό συμπεραίνουμε ότι αφού αρχίζει να έχει κλίση τη χρονική στιγμή 1 δευτερόλεπτο η συνάρτηση που ψάχνουμε θα είναι μια συνάρτηση αναρρίχησης με $T=1$, από την οποία θα αφαιρέσουμε μια συνάρτηση αναρρίχησης με $T=2$ αφού τη χρονική στιγμή 2 δευτερόλεπτα και μέχρι τη χρονική στιγμή 4 δευτερόλεπτα ο παλμός ισορροπεί. Στη συνέχεια θα αφαιρέσουμε μια συνάρτηση αναρρίχησης με $T=4$ αφού εκείνη τη χρονική στιγμή ο παλμός παίρνει αρνητική κλίση, τέλος θα προσθέσουμε μια συνάρτηση αναρρίχησης με $T=5$ αφού τη χρονική στιγμή 5 δευτερόλεπτα ο παλμός

σταθεροποιείται στη τιμή μηδέν μέχρι το τέλος του χρονικού οριζοντα. Όλα τα παραπάνω τα εκφράζουμε στο MatLab με τις παρακάτω εντολές:

```
T=0:.0001:6;
p=uniramp(T,1)-uniramp(T,2)-uniramp(T,4)+uniramp(T,5);
plot(T,p);grid on;
```

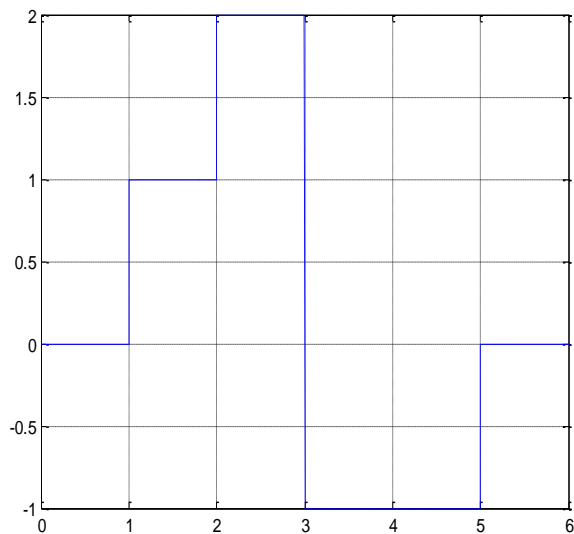


ii)

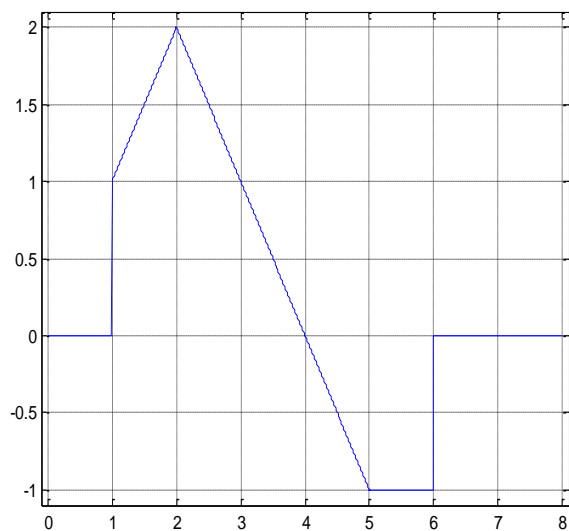


Αρχικά θα προσθέσουμε δύο βηματικές συναρτήσεις η πρώτη με $T=1$ και η δεύτερη με $T=2$, στη συνέχεια ο παλμός πέφτει στην τιμή -1 ενώ είχε την τιμή 2 οπότε θα πρέπει να αφαιρέσουμε 3 φορές τη βηματική συνάρτηση με $T=3$ ώστε να πετύχουμε τη πτώση στην επιθυμητή τιμή ($2 - (-1) = 3$). Τη χρονική στιγμή 5 δευτερόλεπτα ο παλμός ανεβαίνει από την τιμή -1 στο μηδέν οπότε θα προσθέσουμε μια βηματική συνάρτηση με $T=5$.

```
T=0:.0001:6;
p=unistep(T,1)+unistep(T,2)-3*unistep(T,3)+unistep(T,5);
plot(T,p);grid on;
```

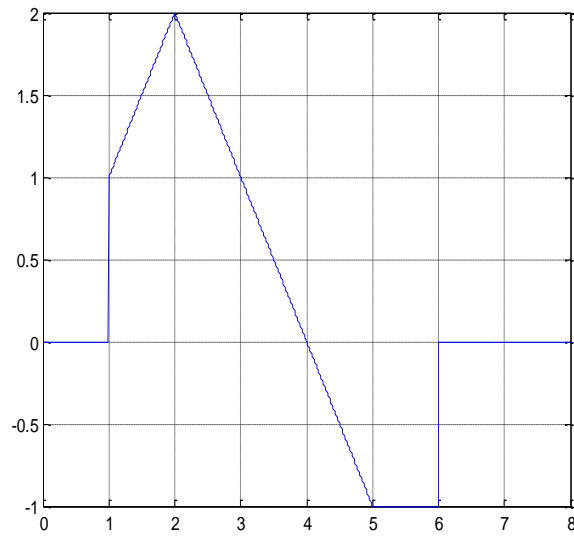


iii)

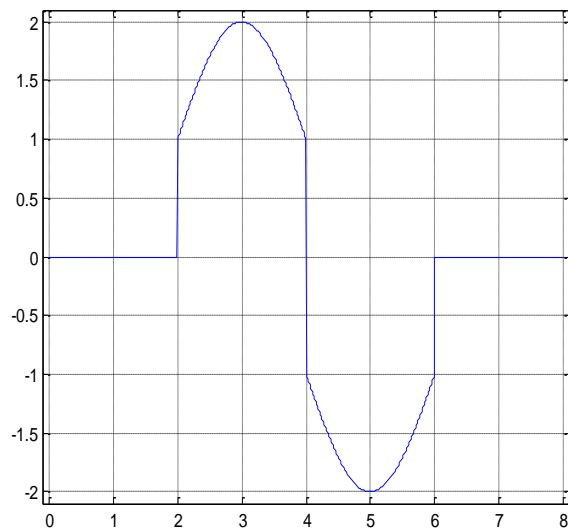


Θα χρειαστούμε μια βηματική συνάρτηση με $T=1$ και θα της προσθέσουμε μια συνάρτηση αναρρίχησης με $T=1$. Στη συνέχεια θα αφαιρέσουμε 2 φορές τη συνάρτηση αναρρίχησης με $T=2$ ώστε να πετύχουμε την αρνητική κλίση τη χρονική στιγμή 2 και για να πετύχουμε την σταθεροποίηση της χρονικής στιγμής 5 θα προσθέσουμε μια συνάρτηση αναρρίχησης με $T=5$ ώστε να «σβήσουμε» την προηγούμενη αρνητική κλίση. Τέλος θα προσθέσουμε μια βηματική συνάρτηση με $T=6$ ώστε τη χρονική στιγμή 6 να ανέβουμε από τη τιμή -1 στο μηδέν.


```
T=0:.0001:8;
p=unistep(T,1)+uniramp(T,1)-2*uniramp(T,2)+uniramp(T,5)+unistep(T,6);
plot(T,p);grid on;
```

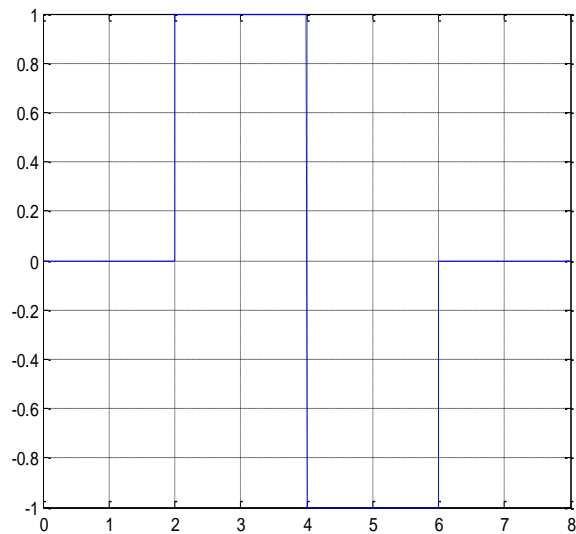


iv)



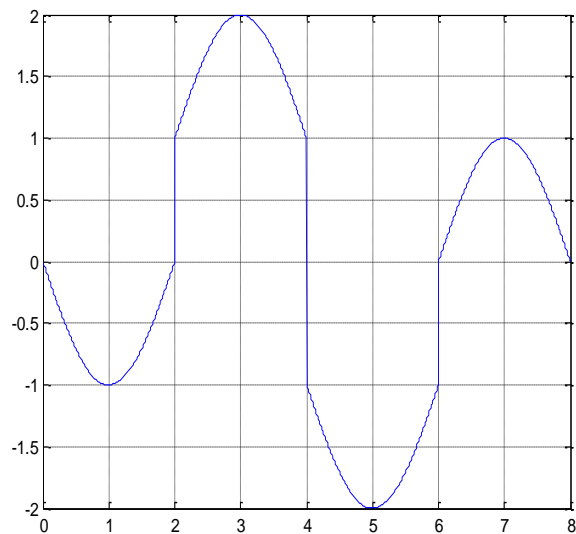
Διακρίνουμε ότι η γραφική είναι συνδυασμός βηματικών συναρτήσεων και ημιτονικής συνάρτησης. Αν δεν είχαμε την ημιτονική συνάρτηση τότε θα μπορούσαμε με τις παρακάτω εντολές να πάρουμε το ακόλουθο γράφημα.

```
T=0:.0001:8;
p1=unistep(T,2)-2*unistep(T,4)+unistep(T,6);
plot(T,p1);grid on;
```



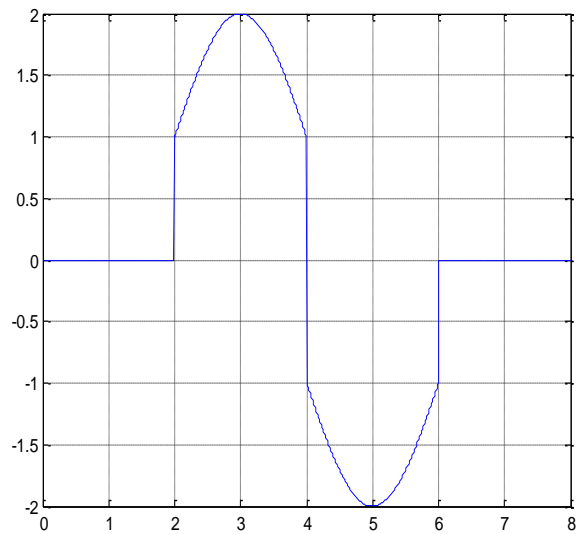
Τώρα προσθέτοντας την ημιτονική συνάρτηση έχουμε:

```
p2=sin(pi*(T-2)/2);
p3=(p1+p2);
plot(T,p3);grid on;
```

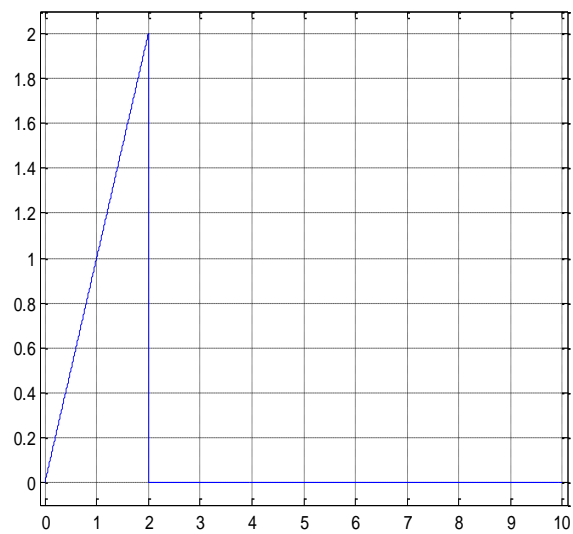


Στα χρονικά διαστήματα 0-2 και 6-8 το γράφημα που δημιουργήσαμε διαφέρει από αυτό που μας ζητήθηκε να δημιουργήσουμε αφού σε αυτά τα διαστήματα η συνάρτηση θα έπρεπε να μηδενίζεται. Όπως πράξαμε σε παραπάνω άσκηση θα πολλαπλασιάσουμε την παραπάνω συνάρτηση που έχουμε φτιάξει μέχρι τώρα με έναν παλμό ο οποίος τα χρονικά διαστήματα 0-2 και 6-8 θα μας δίνει 0 και το χρονικό διάστημα 2-6 θα μας δίνει 1. Οπότε η συνάρτηση που αναζητούμε είναι:

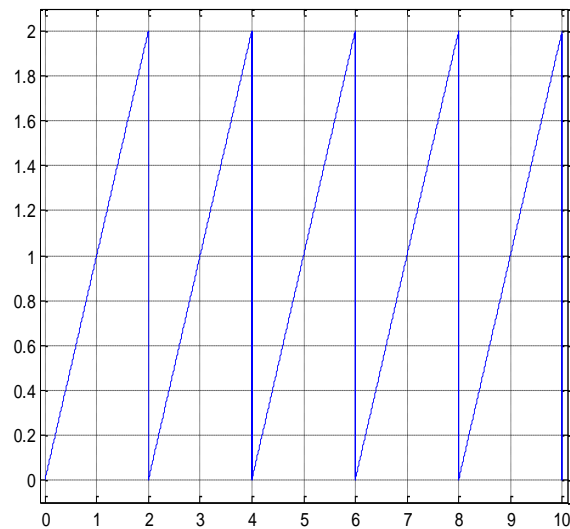
```
P=(p1+p2) .* (unistep(T,2)-unistep(T,6));
plot(T,P);grid on;
```



Άσκηση 4: Να βρεθεί συνάρτηση που να επιστρέφει τον παρακάτω παλμό:

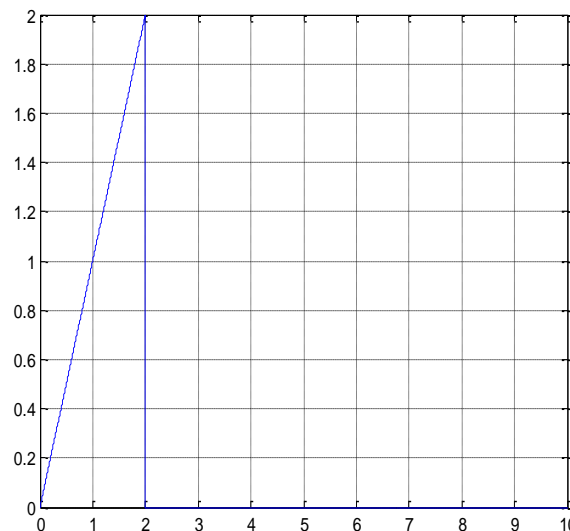


Στη συνέχεια να βρεθεί νέα συνάρτηση η οποία να επιστρέφει μια παλμοσειρά που να αποτελείται από 5 τέτοιους παλμούς, όπως παρουσιάζεται στην παρακάτω γραφική.



Αρχικά θα πρέπει να βρούμε τη συνάρτηση για να τη δημιουργία του μονού παλμού, η οποία είναι η εξής(δεν γίνεται εκτενής αναφορά αφού πλέον γνωρίζουμε τον τρόπο ώστε να βρούμε τη κατάλληλη συνάρτηση για τον εκάστοτε παλμό που ψάχνουμε):

```
T=0:.001:10;
P=uniramp(T,0)-uniramp(T,2)-2*unistep(T,2);
plot(T,P);grid on;
```



Για δικιά μας διευκόλυνση θα φτιάξουμε ένα m-file το οποίο όποτε το καλούμε θα μας επιστρέφει ένα τέτοιου είδους παλμό ανάλογα με το χρονικό ορίζοντα και τις χρονικές στιγμές που θα επιθυμούμε κάθε φορά.

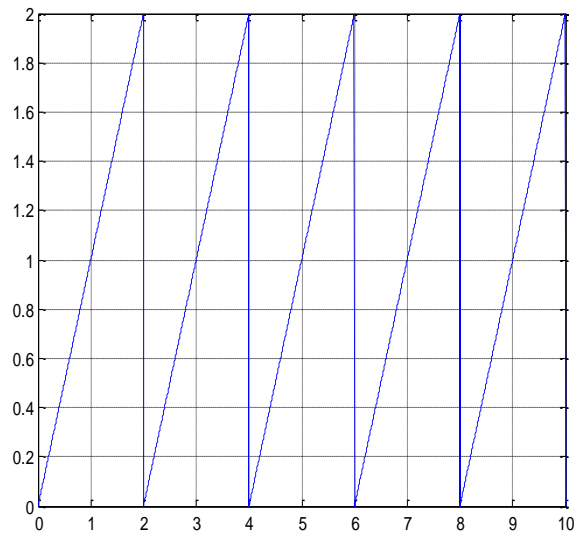
```
function p=palmos(T,T1,T2)
p=uniramp(T,T1)-uniramp(T,T2)-(T2-T1)*unistep(T,T2);
```

Ο όρος $(T2-T1)$ που πολλαπλασιάζουμε την βηματική συνάρτηση είναι για να εισάγουμε σωστά τη πτώση του παλμού, στη συγκεκριμένη περίπτωση

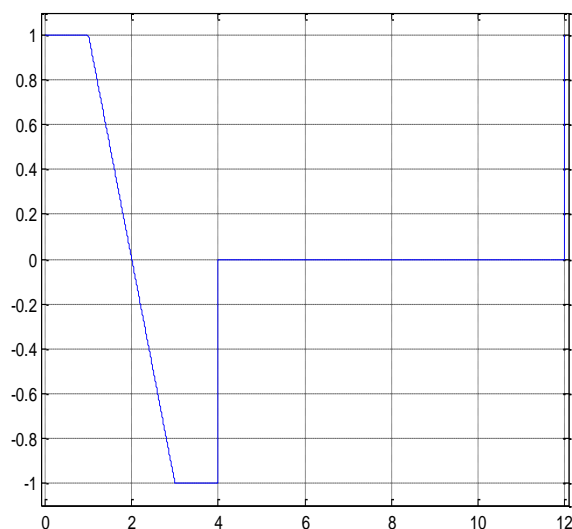
εμείς πολλαπλασιάσαμε με 2 γιατί αυτό είναι το αποτέλεσμα της διαφοράς των χρονικών στιγμών του παλμού($2-0=2$). Τώρα θα πρέπει να φτιάξουμε μια ρουτίνα η οποία θα παίρνει τον αρχικό παλμό με τα δεδομένα που εμείς θέλουμε και να τον επαναλαμβάνει όσες φορές επιθυμούμε, αυτό το επιτυγχάνουμε με το παρακάτω m-file.

```
function p=palmoσειρα(T,N) %T ο χρονικός ορίζοντας και N οι επαναλήψεις που επιθυμούμε.
K=T(end)/N; %Διαιρεί το χρονικό ορίζοντα με τις επαναλήψεις του παλμου που επιθυμούμε
p=T*0; % Μηδενίζει τον παλμό
for L=1:N %Χρησιμοποιώντας την εντολή for θα επαναλάβει τη σειρά εντολών από 1 έως N
p=p+palmos(T,(L-1)*K,L*K); %Εκτελεί τη ρουτίνα που είχαμε προδημιουργήσει για κάθε επανάληψη
end %παλμού που επιθυμούμε και κάθε φορά τον προσθέτει στο προηγούμενο αποτέλεσμα
```

Αφού ολοκληρώσουμε τα παραπάνω παίρνουμε την επιθυμητή παλμοσειρά με τις εξής εντολές.



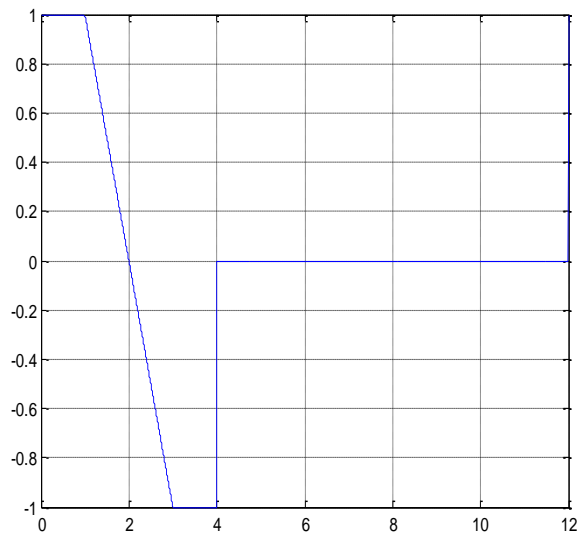
Άσκηση 5: Να βρεθεί συνάρτηση που να επιστρέφει τον παρακάτω παλμό:



Και στη συνέχεια η ρουτίνα που θα επιστρέφει μια παλμοσειρά με 3 επαναλαμβανόμενους παλμούς ίδιους με τον παραπάνω.

Η συνάρτηση που μας δίνει τον παλμό είναι:

```
T=0:.001:12;
p=unistep(T,0)-uniramp(T,1)+uniramp(T,3)+unistep(T,4)+unistep(T,12);
plot(T,H);grid on;
```

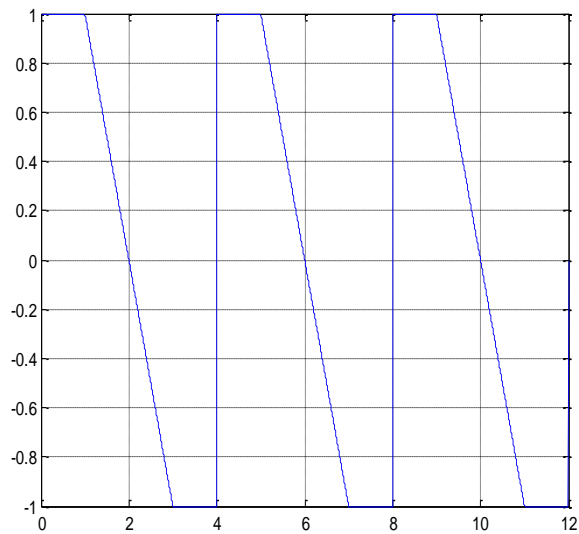


Μπορούμε να δημιουργήσουμε το παρακάτω m-file μέσα στο οποίο αποθηκεύουμε και τον χρονικό οριζόντιο, το οποίο διαφοροποιεί τον τρόπο αυτόν από αυτόν της προηγούμενης άσκησης.

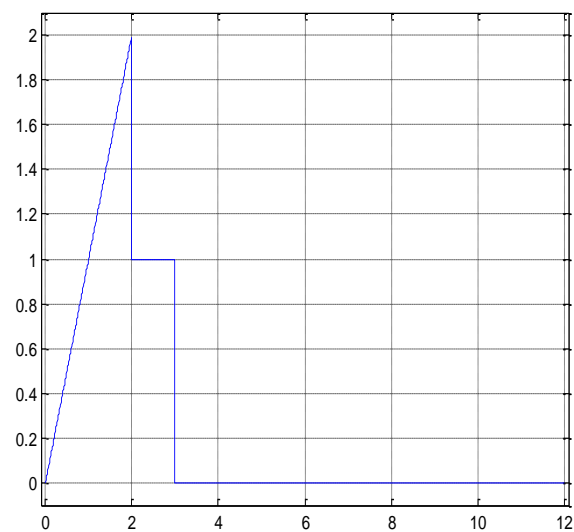
```
function p=palmos2(T0,T1,T2,T3)
t=0:.01:12;
p=unistep(t,T0)-uniramp(t,T1)+uniramp(t,T2)+unistep(t,T3);
```

Χρησιμοποιώντας την παραπάνω ρουτίνα που δημιουργήσαμε δίνουμε στο περιβάλλον του MatLab τις παρακάτω εντολές που μας επιστρέφουν την ακόλουθη και επιθυμητή παλμοσειρά.

```
p=palmos2(0,1,3,4)+palmos2(4,5,7,8)+palmos2(8,9,11,12);
plot(T,p);grid on;
```



Άσκηση 6: Να βρεθεί συνάρτηση που να επιστρέφει τον παρακάτω παλμό και στη συνέχεια να βρεθεί η ρουτίνα που θα επιστρέφει μια παλμοσειρά που θα αποτελείται από 4 επαναλήψεις του παλμού.

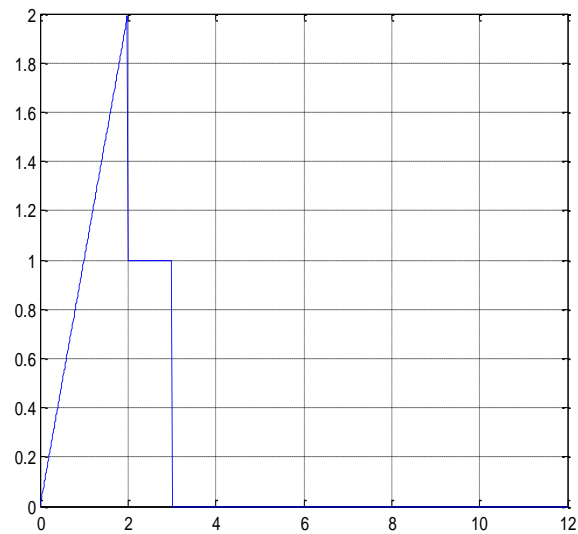


Η συνάρτηση που μας δίνει τον παλμό είναι:

```
T=0:.01:12;
p=uniramp(T,0)-uniramp(T,2)-unistep(T,2)-unistep(T,3);
plot(T,p);grid on;
```

Μπορούμε να φτιάξουμε μια ρουτίνα σε m-file που θα μας επιστρέφει τον παλμό.

```
function p=palmos3(T,T1,T2,T3)
p=uniramp(T,T1)-uniramp(T,T2)-unistep(T,T2)-unistep(T,T3);
```

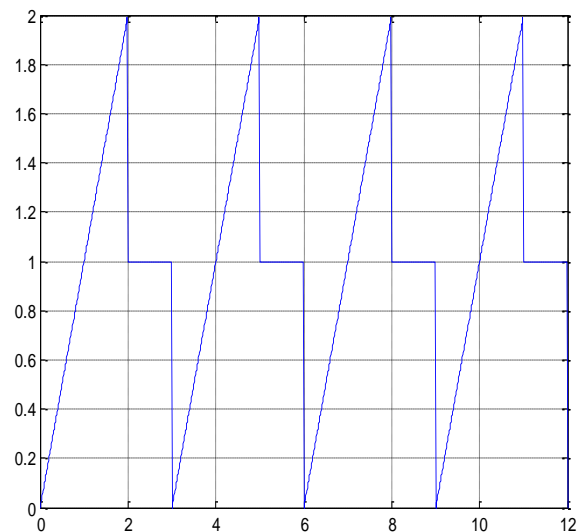


Για να πάρουμε την παλμοσειρά που μας ζητείται θα εισάγουμε την παραπάνω ρουτίνα του παλμού σε μια άλλη ρουτίνα που θα μας επιστρέφει την παλμοσειρά.

```
function P=palmoσειρα3 (T, T1, T2, T3, N)
P=T*0;
for k=0:N-1
    P=P+palmos3 (T, T1+T3 *k, T2+T3 *k, T3+T3 *k) ;
end
```

Και εκτελώντας την ρουτίνα έχουμε:

```
P=palmoσειρα3 (T, 0, 2, 3, 4) ;
plot (T,P) ;grid on;
```



5. Ανάλυση Συστημάτων Αυτόματου Ελέγχου

Σε αυτό το κεφάλαιο θα αναφερθούμε στον ορισμό των συστημάτων αυτόματου ελέγχου μέσω της συνάρτησης μεταφοράς, θα εξηγήσουμε τι είναι ένα σύστημα ανοιχτού βρόχου και τι σύστημα κλειστού βρόχου καθώς και στη διασύνδεση συστημάτων.

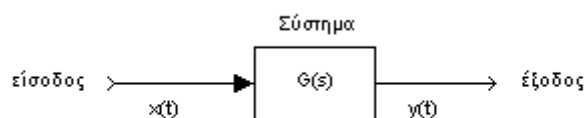
5.1 Συνάρτηση Μεταφοράς

Κατά τη μελέτη των συστημάτων ελέγχου χρησιμοποιούνται συχνά οι συναρτήσεις μεταφοράς που χαρακτηρίζουν τις σχέσεις εισόδου-εξόδου των συστημάτων.

Η συνάρτηση μεταφοράς είναι μια περιγραφή στο πεδίο της μιγαδικής συχνότητας και ισχύει για μια περιορισμένη κατηγορία συστημάτων και συγκεκριμένα για τα γραμμικά, μη χρονικά μεταβαλλόμενα συστήματα που έχουν μηδενικές αρχικές συνθήκες. Η συνάρτηση μεταφοράς ορίζεται μόνο στη περίπτωση ενός γραμμικού συστήματος με σταθερές παραμέτρους γιατί σε ένα σύστημα μεταβλητών παραμέτρων, το οποίο καλείται συνήθως σύστημα μεταβλητού χρόνου, περιέχει μια ή περισσότερες χρονικά μεταβαλλόμενες μεταβλητές και για τον λόγο αυτόν ενδέχεται να μην ορίζεται ο αντίστοιχος μετασχηματισμός Laplace.

Η συνάρτηση μεταφοράς ενός συστήματος ορίζεται ως ο λόγος μετασχηματισμού Laplace της μεταβλητής που εκφράζει την έξοδο $y(t)$ προς τον μετασχηματισμό Laplace της μεταβλητής που εκφράζει την είσοδο $x(t)$, θεωρώντας μηδενικές αρχικές συνθήκες. Η συνάρτηση μεταφοράς ενός συστήματος αντιστοιχεί σε μια σχέση με την οποία περιγράφεται η δυναμική του υπό εξέταση συστήματος, επομένως η συνάρτηση μεταφοράς δεν περιέχει καμιά πληροφορία σχετικά με την εσωτερική δομή του συστήματος και της συμπεριφοράς του γενικότερα. Η συνάρτηση συμβολίζεται συνήθως με κεφαλαίο γράμμα του αγγλικού αλφαβήτου σε μορφή συνάρτησης τη μιγαδικής συχνότητας π.χ. $G(s)$.

Αν θεωρήσουμε το σύστημα του παρακάτω σχήματος:



Τότε η συνάρτηση μεταφοράς του συστήματος θα είναι:

$$G(s) = \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{x(t)\}} = \frac{Y(s)}{X(s)}$$

Η χρήση της συνάρτησης μεταφοράς στην ανάλυση συστημάτων ελέγχου είναι ένα σημαντικό εργαλείο γιατί μας επιτρέπει να ελέγχουμε τη συμπεριφορά του πραγματικού συστήματος χωρίς να το θέτουμε σε κίνδυνο αφού μπορούμε να θέσουμε διαφορετικές συναρτήσεις ως εισόδους και με τη βοήθεια της συνάρτησης μεταφοράς να υπολογίζουμε την πιθανή έξοδο με την επίλυση μιας απλής εξίσωσης. Θα μπορούσαμε να πούμε ότι θέτουμε το σύστημα σε «μαθηματικές μετρήσεις» (μέσω επίλυσης απλών εξισώσεων, αφού το μόνο ζητούμενο-άγνωστο θα είναι η έξοδος του συστήματος) αντί να θέτουμε το σύστημα σε πειραματικές μετρήσεις όπου σε κάποιες περιπτώσεις λανθασμένων διεγέρσεων εισόδου να κινδύνευε το ίδιο το σύστημα.

Μπορούμε όταν μας δίνεται η διαφορική εξίσωση ενός συστήματος να βρούμε τη συνάρτηση μεταφοράς του αλλάζοντας, με τη χρήση του μετασχηματισμού Laplace, την εξίσωση από το πεδίο του χρόνου στο πεδίο της μιγαδικής συχνότητας. Αντιστρόφως όταν μας δίνεται η συνάρτηση μεταφοράς ενός συστήματος μπορούμε να βρούμε τη διαφορική εξίσωση του συστήματος ακολουθώντας την αντίστροφη διαδικασία δηλαδή μετασχηματίζοντας την εξίσωση της συνάρτησης μεταφοράς από το πεδίο της συχνότητας στο πεδίο του χρόνου.

Τα γραμμικά χρονικά αμετάβλητα συστήματα όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο μπορούν να περιγραφούν από γραμμικές συναρτήσεις της μορφής:

$$a_i \frac{d^i c(t)}{dt^i} + a_{i-1} \frac{d^{i-1} c(t)}{dt^{i-1}} + \dots + a_0 c(t) = b_k \frac{d^k r(t)}{dt^k} + b_{k-1} \frac{d^{k-1} r(t)}{dt^{k-1}} + \dots + b_0 r(t)$$

Όπου $c(t)$ η συνάρτηση εξόδου και $r(t)$ η συνάρτηση εισόδου, με χρήση του μετασχηματισμού Laplace η παραπάνω διαφορική μετατρέπεται:

$$\begin{aligned} a_i s^i C(s) + a_{i-1} s^{i-1} C(s) + \dots + a_0 C(s) &= b_k s^k R(s) + b_{k-1} s^{k-1} R(s) + \dots + b_0 R(s) \leftrightarrow \\ \leftrightarrow (a_i s^i + a_{i-1} s^{i-1} + \dots + a_0) C(s) &= (b_k s^k + b_{k-1} s^{k-1} + \dots + b_0) R(s) \leftrightarrow \\ \leftrightarrow \frac{C(s)}{R(s)} &= \frac{a_i s^i + a_{i-1} s^{i-1} + \dots + a_0}{b_k s^k + b_{k-1} s^{k-1} + \dots + b_0} \end{aligned}$$

Ο παραπάνω λόγος εξόδου-εισόδου στην πραγματικότητα μας δίνει τη συνάρτηση μεταφοράς του συστήματος οπότε η συνάρτηση μεταφοράς από μαθηματική άποψη είναι ένας πίνακας i γραμμών και k στηλών με στοιχεία ρητές συναρτήσεις με μεταβλητή τη μιγαδική συχνότητα s που έχουν βαθμό αριθμητή μέχρι και ίσο με τον βαθμό του παρονομαστή δηλαδή $i \leq k$ (βλ. κεφάλαιο 3.2).

Η εντολή με την οποία ορίζουμε στο περιβάλλον του MatLab μια συνάρτηση μεταφοράς είναι η tf η οποία προέρχεται από τα αρχικά της αγγλικής φράσης *transfer function* που μεταφράζεται συνάρτηση μεταφοράς. Η σύνταξη της είναι η ακόλουθη:

$$G = \text{tf}([num], [den])$$

Όπου, num πολυώνυμο με τους σταθερούς όρους του αριθμητή και

den πολυώνυμο με τους σταθερούς όρους του παρανομαστή.

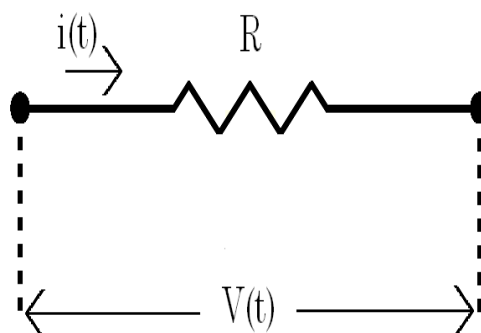
Η συνάρτηση μεταφοράς ενός συστήματος είναι ο λόγος δύο πολυωνύμων του αριθμητή και του παρανομαστή, οι ρίζες των πολυωνύμων αυτών έχουν σημασία στην ανάλυση των συστημάτων αυτόματου ελέγχου. Η ευστάθεια ενός συστήματος και η ταχύτητα απόκρισης (θα αναφερθούμε αναλυτικά για αυτούς τους όρους σε επόμενο κεφάλαιο) είναι κάποια από τα χαρακτηριστικά του συστήματος που επηρεάζονται από τις ρίζες αυτών των δύο πολυωνύμων. Οι ρίζες του αριθμητή ονομάζονται μηδενικά της συνάρτησης μεταφοράς ή του συστήματος ενώ οι ρίζες του παρανομαστή ονομάζονται πόλοι της συνάρτησης μεταφοράς ή του συστήματος. Σε επόμενο κεφάλαιο θα αναφερθούμε αναλυτικότερα για τη σημασία των πόλων και των μηδενικών ενός συστήματος, ενώ εκτενέστερη αναφορά προβλέπεται στο μάθημα των Συστημάτων Αυτόματου Ελέγχου II.

Η τιμή της συνάρτησης μεταφοράς ενός συστήματος για $s = 0$ ονομάζεται συνεχές κέρδος (dc-gain) και τις περισσότερες φορές είναι μεγαλύτερο από τη μονάδα.

Τέλος από τη συνάρτηση μεταφοράς ενός συστήματος εντοπίζεται και η τάξη του συστήματος. Η τάξη του συστήματος δίνεται από τη μεγαλύτερη δύναμη στην οποία είναι υψωμένη η μιγαδική συχνότητα s στο πολυώνυμο του παρανομαστή, στην ουσία η τάξη ενός συστήματος μας δίνει τον αριθμό των πόλων του συστήματος.

Για να κατανοήσουμε τη χρήση και τη λειτουργία της συνάρτησης μεταφοράς ακολουθούν απλά παραδείγματα με χρήση των βασικών διαφορικών εξισώσεων των ηλεκτρικών στοιχείων και το μετασχηματισμό τους κατά Laplace που παρουσιάζονται σε πίνακα στη σελίδα 52.

Παράδειγμα 1: Να βρεθεί η συνάρτηση μεταφοράς μιας ηλεκτρικής αντίστασης.



Αρχικά θα πρέπει να ορίσουμε ποια θα είναι η είσοδος και ποια η έξοδος κάτι το οποίο έχει μεγάλη σημασία στην ανάλυση του συστήματος αφού για διαφορετική έξοδο θα έχουμε και διαφορετική συνάρτηση αναφοράς.

i) Για είσοδο τη τάση στα άκρα της αντίστασης και έξοδο το ρεύμα που διαρρέει την αντίσταση έχουμε:

$$u(t) = R * i(t) \xrightarrow{\mathcal{L}} U(s) = R * I(s) \leftrightarrow \frac{I(s)}{U(s)} = \frac{1}{R}$$

Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{1}{R}$.

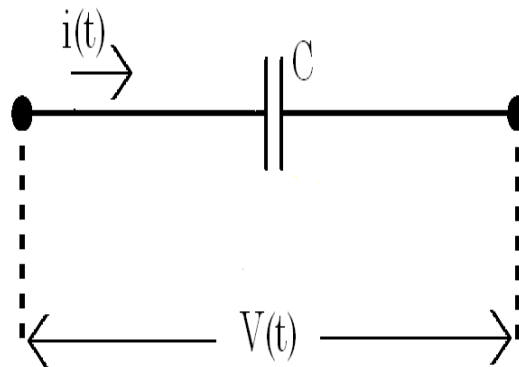
ii) Για είσοδο το ρεύμα που διαρρέει την αντίσταση και έξοδο τη τάση στα άκρα της αντίστασης έχουμε:

$$u(t) = R * i(t) \xrightarrow{\mathcal{L}} U(s) = R * I(s) \leftrightarrow \frac{U(s)}{I(s)} = R$$

Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = R$.

Γίνεται αντιληπτό ότι για το ίδιο σύστημα η συνάρτηση μεταφοράς μπορεί να διαφέρει αν επιλέξουμε διαφορετικές μεταβλητές για είσοδο και έξοδο.

Παράδειγμα 2: Να βρεθεί η συνάρτηση μεταφοράς του παρακάτω πυκνωτή.



i) Για είσοδο τη τάση στα άκρα του πυκνωτή και έξοδο το ρεύμα που τον διαρρέει έχουμε:

$$i(t) = C * \frac{du(t)}{dt} \xrightarrow{\mathcal{L}} I(s) = s * C * U(s) - C * u(0)$$

Έχουμε αναφέρει ότι η συνάρτηση μεταφοράς ορίζεται για μηδενικές αρχικές συνθήκες οπότε η παραπάνω σχέση γίνεται:

$$I(s) = s * C * U(s) - C * 0 \leftrightarrow \frac{I(s)}{U(s)} = s * C$$

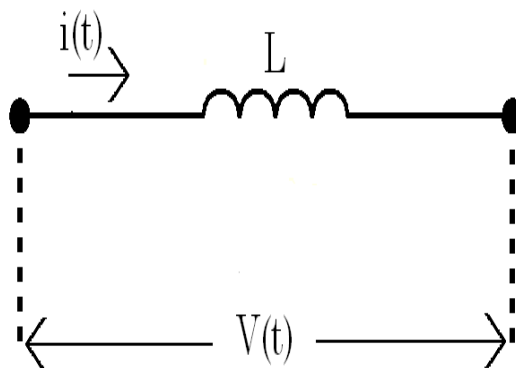
Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = s * C$

ii) Για είσοδο το ρεύμα που διαρρέει τον πυκνωτή και έξοδο τη τάση στα άκρα του έχουμε:

$$u(t) = \frac{1}{C} * \int_0^t i(t) dt \xrightarrow{\mathcal{L}} U(s) = \frac{I(s)}{s * C} + \frac{u(0)}{s} \leftrightarrow U(s) = \frac{I(s)}{s * C} + 0 \leftrightarrow \frac{U(s)}{I(s)} = \frac{1}{s * C}$$

Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{1}{s * C}$.

Παράδειγμα 3: Να βρεθεί η συνάρτηση μεταφοράς του παρακάτω πηνίου.



i) Για είσοδο τη τάση στα άκρα του πηνίου και έξοδο το ρεύμα που το διαρρέει έχουμε:

$$i(t) = \frac{1}{L} * \int_0^t u(t) dt \xrightarrow{\mathcal{L}} I(s) = \frac{U(s)}{s * L} * + \frac{i(0)}{s} \leftrightarrow I(s) = \frac{U(s)}{s * L} * + 0 \leftrightarrow \frac{I(s)}{U(s)} = \frac{1}{s * L}$$

Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{1}{s * L}$

ii) Για είσοδο το ρεύμα που διαρρέει τον πυκνωτή και έξοδο τη τάση στα άκρα του έχουμε:


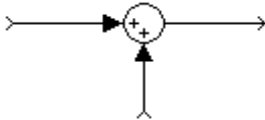
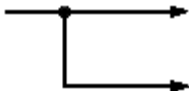
$$u(t) = L * \frac{di(t)}{dt} \xrightarrow{\mathcal{L}} U(s) = s * L * I(s) - L * i(0) \leftrightarrow U(s) = s * L * I(s) \leftrightarrow \frac{U(s)}{I(s)} = s * L$$

Άρα η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = s * L$.

5.2 Δομικά Διαγράμματα Συστημάτων

Για την περιγραφή συστημάτων συνηθίζεται να χρησιμοποιούνται τα δομικά διαγράμματα και τα διαγράμματα ροής. Η διαφορά αυτών των δύο μεθόδων εντοπίζεται στο γεγονός ότι τα διαγράμματα ροής αναφέρονται σε συναρτήσεις στο πεδίο του χρόνου ενώ τα δομικά διαγράμματα σε συναρτήσεις μεταφοράς στο πεδίο της μιγαδικής συχνότητας.

Θα προτιμήσουμε τη χρήση των δομικών διαγραμμάτων από τα διαγράμματα ροής επειδή το Simulink του MatLab για τη σχεδίαση των βαθμίδων των συστημάτων χρησιμοποιεί τα δομικά διαγράμματα και επειδή το πεδίο της μιγαδικής συχνότητας παρουσιάζει ευκολία στο χειρισμό των διαγραμμάτων. Τα βασικά δομικά στοιχεία ενός δομικού διαγράμματος ή διαγράμματος βαθμίδων είναι τα εξής.

Βασικά δομικά στοιχεία δομικών διαγραμμάτων	
	<p>Η βαθμίδα που σχεδιάζεται ως ορθογώνιο παραλληλόγραμμο και αντιστοιχεί σε ένα σύστημα στο οποίο δρα ένα σήμα εισόδου και ένα σήμα εξόδου που απορρέει.</p>
	<p>Ο κύκλος που συνοδεύεται από κατάλληλο αριθμό πρόσημων και σημάτων που καταλήγουν σε αυτόν ονομάζεται σημείο άθροισης (μπορεί όμως να πραγματοποιηθεί και αφαίρεση σημάτων).</p>
	<p>Σημείο λήψης</p>

Είναι φανερό ότι σε ένα διάγραμμα βαθμίδων τα βέλη αντιπροσωπεύουν την κατεύθυνση ροής του σήματος. Τα διαγράμματα βαθμίδων συναρτήσεων μεταφοράς είναι μια εικονική αναπαράσταση της σχέσης εισόδου και εξόδου ενός συστήματος και κάθε βαθμίδα χαρακτηρίζεται από μια περιγραφή εισόδου και μια εξόδου όπου και τοποθετούνται αντιστοίχως το σήμα εισόδου και το σήμα εξόδου.

Τα δομικά διαγράμματα περιγράφουν ένα σύστημα εποπτικά και αποτελούν τη σχηματική παράσταση των λειτουργιών των συντελεστών του συστήματος. Στα πλεονεκτήματα της χρήσης των δομικών διαγραμμάτων κατατάσσονται η ευκολία με την οποία καταρτίζονται τα διαγράμματα και οι πληροφορίες που παρέχουν για τη δυναμική συμπεριφορά του συστήματος. Η μορφή του δομικού διαγράμματος ενός συστήματος μπορεί να απλουστευ-

θεί ώστε να αποδίδεται με λιγότερες βαθμίδες κάτι που επιτυγχάνεται με την συμπύκνωση των διαδοχικών διασυνδεμένων βαθμίδων σε ισοδύναμα διαγράμματα. Η απλοποίηση του διαγράμματος μας παρουσιάζει με καλύτερο τρόπο τη σχέση μεταξύ της εισόδου που δρα στο σύστημα αλλά και της εξόδου που απορρέει από αυτό, καθώς και το βαθμό που επιδρούν στην έξοδο τα στοιχεία του συστήματος. Όμως η απλοποίηση συστημάτων έχει ως βασικό μειονέκτημα ότι αποκρύπτει σε ένα βαθμό, εξαιτίας των συμπυκνώσεων που έχουν διενεργηθεί, πληροφορίες για την διασύνδεση επιμέρους στοιχείων του συστήματος.

Τέλος τα βασικά βήματα για τη σχεδίαση του διαγράμματος βαθμίδων ενός συστήματος είναι τα εξής:

1. Εντοπισμός των εξισώσεων που περιγράφουν τη δυναμική συμπεριφορά των επιμέρους στοιχείων του συστήματος.
2. Υπολογισμός του μετασχηματισμού Laplace των παραπάνω εξισώσεων λαμβάνοντας ως μηδενικές τις αρχικές συνθήκες.
3. Απεικονίζουμε κάθε μία από τις παραπάνω μετασχηματισμένες κατά Laplace εξισώσεις σε μορφή βαθμίδων.
4. Τέλος, συνδυασμός όλων των στοιχείων του συστήματος σε ένα ενιαίο διάγραμμα βαθμίδων.

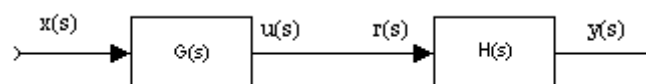
5.3 Διασύνδεση Συστημάτων

Αναφέραμε η απλοποίηση ενός διαγράμματος βαθμίδων μπορεί να μας παρουσιάσει με καλύτερο τρόπο τη σχέση μεταξύ εισόδου και εξόδου του συστήματος το οποίο μελετάμε, για να πετύχουμε αυτή την απλοποίηση πρέπει να κατανοήσουμε τον τρόπο διασύνδεσης των βαθμίδων-στοιχείων των συστημάτων. Τα είδη διασύνδεσης είναι δύο:

- ✚ Διασύνδεση σε σειρά
- ✚ Παράλληλη διασύνδεση

Ανάλογα με τη διασύνδεση των συστημάτων πραγματοποιείται και η ανάλογη απλοποίηση με συγκεκριμένες πράξεις μεταξύ των συναρτήσεων μεταφοράς των συστημάτων και έτσι οι βαθμίδες συμπυκνώνονται σε μια.

Στη περίπτωση που έχουμε δύο συστήματα(βαθμίδες) συνδεδεμένα σε σειρά με συναρτήσεις μεταφοράς $G(s)$ και $H(s)$ αντιστοίχως όπως αυτά που παρουσιάζονται στην παρακάτω εικόνα:



Τότε μπορούν να απλοποιηθούν σε μία βαθμίδα με συνάρτηση μεταφοράς $F(s) = G(s) * H(s)$, το παραπάνω αποδεικνύεται ως εξής:

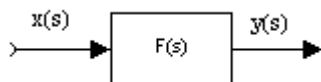
$$G(s) = \frac{u(s)}{x(s)} \leftrightarrow u(s) = G(s) * x(s)$$

και

$$H(s) = \frac{y(s)}{r(s)} \leftrightarrow y(s) = H(s) * r(s)$$

Είναι εμφανές όμως ότι ισχύει $u(s) = r(s)$ οπότε:

$$y(s) = H(s) * G(s) * x(s) \leftrightarrow \frac{Y(s)}{X(s)} = H(s) * G(s) \leftrightarrow F(s) = G(s) * H(s)$$



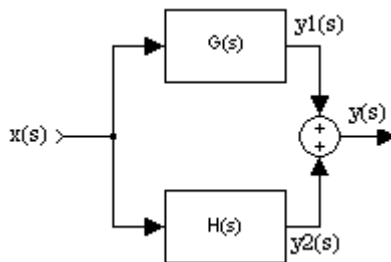
Επομένως αν δύο ή περισσότερες βαθμίδες είναι συνδεδεμένες σε σειρά τότε απλοποιούνται σε μία βαθμίδα με συνάρτηση μεταφοράς το γινόμενο των επιμέρους συναρτήσεων μεταφοράς των βαθμίδων.

Στο περιβάλλον του MatLab μπορούμε να υπολογίσουμε τη συνάρτηση μεταφοράς δύο βαθμίδων που είναι συνδεδεμένες σε σειρά είτε πολλαπλασιάζοντας τις συναρτήσεις μεταφοράς τους είτε χρησιμοποιώντας την συνάρτηση *series* η οποία στη γενική της σύνταξη είναι:

`F=series(G,H)`

όπου G και H η συναρτήσεις μεταφοράς των σε σειρά συνδεδεμένων συστημάτων.

Έστω τώρα η περίπτωση που τα δύο παραπάνω συστήματα είναι συνδεδεμένα παράλληλα, ομοίως με συναρτήσεις μεταφοράς $G(s)$ και $H(s)$ όπως φαίνεται στην παρακάτω εικόνα.



Οι δύο βαθμίδες συμπυκνώνονται σε μια βαθμίδα με συνάρτηση μεταφοράς $F(s) = G(s) + H(s)$, το οποίο αποδεικνύεται από τις παρακάτω σχέσεις:

$$G(s) = \frac{y1(s)}{x(s)} \leftrightarrow y1(s) = G(s) * x(s)$$

$$H(s) = \frac{y2(s)}{x(s)} \leftrightarrow y2(s) = H(s) * x(s)$$

και

$$\begin{aligned} y(s) = y1(s) + y2(s) &\leftrightarrow y(s) = G(s) * x(s) + H(s) * x(s) \leftrightarrow y(s) = x(s) * (G(s) + H(s)) \\ &\leftrightarrow \frac{Y(s)}{X(s)} = G(s) + H(s) \leftrightarrow F(s) = G(s) + H(s) \end{aligned}$$

Οπότε αν δύο ή περισσότερες βαθμίδες είναι συνδεδεμένες παράλληλα τότε απλοποιούνται σε μία βαθμίδα με συνάρτηση μεταφοράς το άθροισμα

ή τη διαφορά(ανάλογα με τα πρόσημα στο σημείο άθροισης) των επιμέρους συναρτήσεων μεταφοράς των βαθμίδων.

Στο περιβάλλον του MatLab μπορούμε να υπολογίσουμε τη συνάρτηση μεταφοράς δύο βαθμίδων που είναι συνδεδεμένες παράλληλα είτε προσθέτοντας τις συναρτήσεις μεταφοράς τους είτε χρησιμοποιώντας την συνάρτηση *parallel* η οποία στη γενική της σύνταξη είναι:

```
F=parallel(G,H)
```

όπου G και H η συναρτήσεις μεταφοράς των σε σειρά συνδεδεμένων συστημάτων.

Παράδειγμα 1: Δίνονται οι συναρτήσεις μεταφοράς των παρακάτω συστημάτων:

$$G(s) = \frac{1}{s+1} \text{ και } H(s) = \frac{2*s}{s+2}$$

Να βρεθεί η συνάρτηση μεταφοράς του απλοποιημένου συστήματος όταν:

- i) Τα συστήματα είναι συνδεδεμένα σε σειρά και
- ii) Τα συστήματα είναι παράλληλα συνδεδεμένα.

i) Όταν τα συστήματα είναι συνδεδεμένα σε σειρά πολλαπλασιάζουμε τις συναρτήσεις μεταφοράς τους ή χρησιμοποιούμε την εντολή *series*.

```
G=tf([1],[1 1]);
H=tf([2 0],[1 2]);
F=G*H
Transfer function:
      2 s
-----
s^2 + 3 s + 2

>> F=series(G,H)

Transfer function:
      2 s
-----
s^2 + 3 s + 2
```

Το απλοποιημένο σύστημα έχει συνάρτηση μεταφοράς $F(s) = \frac{2*s}{s^2+3*s+2}$.

ii) Όταν τα συστήματα είναι παράλληλα συνδεδεμένα προσθέτουμε τις συναρτήσεις μεταφοράς τους ή χρησιμοποιούμε την εντολή *parallel*.

$$F=G+H$$

Transfer function:

$$\frac{2s^2 + 3s + 2}{s^2 + 3s + 2}$$

$$F=\text{parallel}(G, H)$$

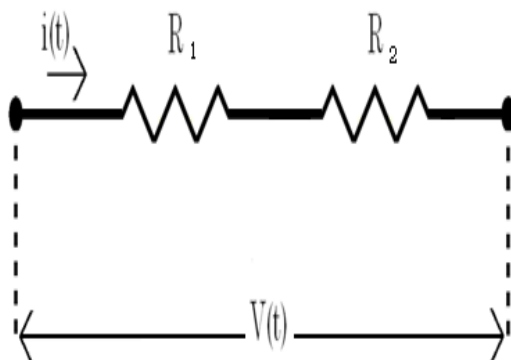
Transfer function:

$$\frac{2s^2 + 3s + 2}{s^2 + 3s + 2}$$

$$\frac{2s^2 + 3s + 2}{s^2 + 3s + 2}$$

Το απλοποιημένο σύστημα έχει συνάρτηση μεταφοράς $F(s) = \frac{2s^2+3s+2}{s^2+3s+2}$.

Παράδειγμα 2: Δίνονται οι δύο αντιστάσεις του παρακάτω σχήματος και ζητείται να βρεθεί η συνάρτηση μεταφοράς του συστήματος για είσοδο το ρεύμα $i(t)$ που διαρρέει τις αντιστάσεις και έξοδο τη τάση $u(t)$ στα άκρα των αντιστάσεων.



Ισχύει ότι

$$u(t) = (R_1 + R_2) * i(t) \xrightarrow{\mathcal{L}} U(s) = (R_1 + R_2) * I(s) \leftrightarrow \frac{U(s)}{I(s)} = R_1 + R_2$$

Άρα η συνάρτηση του απλοποιημένου συστήματος είναι $F(s) = R_1 + R_2$.

Παράδειγμα 3: Για το παραπάνω σχήμα να επαναληφθεί η ίδια διαδικασία αλλά για είσοδο τη τάση $u(t)$ στα άκρα των αντιστάσεων και έξοδο το ρεύμα $i(t)$ που τις διαρρέει.

Ισχύει ότι

$$u(t) = (R_1 + R_2) * i(t) \xrightarrow{\mathcal{L}} U(s) = (R_1 + R_2) * I(s) \leftrightarrow \frac{I(s)}{U(s)} = \frac{1}{(R_1 + R_2)}$$

Άρα η συνάρτηση του απλοποιημένου συστήματος είναι $F(s) = \frac{1}{R_1+R_2}$.

Όπως έχουμε αναφέρει η εύρεση συνάρτησης μεταφοράς ενός συστήματος επηρεάζεται από τον καθορισμό εισόδου και εξόδου όμως στο παραπάνω παράδειγμα δεν παρατηρούμε μόνο αυτό. Βλέπουμε ότι ενώ ηλεκτρολογικά οι δύο αντιστάσεις είναι συνδεδεμένες σε σειρά δεν σημαίνει ότι και στη γλώσσα των συστημάτων είναι συνδεδεμένες σε σειρά. Αν ίσχυε τότε θα έπρεπε στο παράδειγμα 2 να βρούμε ως αποτέλεσμα $F(s) = R_1 * R_2$, σύμφωνα με το αποτέλεσμα που βρήκαμε οι δύο αντιστάσεις στη γλώσσα των συστημάτων είναι συνδεδεμένες παράλληλα δηλαδή έχουν κοινή είσοδο και διαφορετική έξοδο, το οποίο ισχύει αφού δύο αντιστάσεις σε σειρά ηλεκτρολογικά έχουν κοινό ρεύμα αλλά διαφορετική τάση στα άκρα τους. Πρέπει λοιπόν να κατανοήσουμε ότι οι κανόνες της ηλεκτρολογικής συνδεσμολογίας είναι διαφορετικοί από αυτούς που ισχύουν στη διασύνδεση συστημάτων.

5.4 Συστήματα Αυτόματου ελέγχου ανοιχτού και κλειστού βρόγχου

Σε ένα σύστημα αυτόματου ελέγχου το σήμα εισόδου δίνεται στο σύστημα από ένα «πρόσθετο σύστημα» που το ονομάζουμε αντισταθμιστή ή ρυθμιστή ή ελεγκτή. Στα ανοικτά συστήματα ο αντισταθμιστής είναι κατασκευασμένος ώστε το σήμα που παράγει και τροφοδοτεί το σύστημα να αποτελεί την κατάλληλη διέγερση για να αποφέρει την επιθυμητή έξοδο του συστήματος. Στα κλειστά συστήματα ο αντισταθμιστής διεγείρεται και από την έξοδο του συστήματος οπότε το σήμα που παράγει ο αντισταθμιστής και διεγείρει το σύστημα επηρεάζεται αποφασιστικά από την έξοδο του ίδιου του συστήματος.

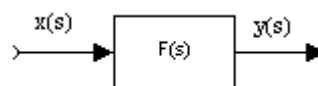
Με απλά λόγια θα μπορούσαμε να ορίσουμε πως ανοικτό σύστημα είναι ένα σύστημα που η είσοδος δεν είναι συνάρτηση της εξόδου και κλειστό σύστημα είναι ένα σύστημα που η είσοδος είναι συνάρτηση της εξόδου.

Τα κλειστά συστήματα διαφέρουν από τα ανοικτά συστήματα στην αρχή λειτουργίας τους και η διαφορά αυτή οφείλεται στη λήψη πληροφοριών από την έξοδο στην είσοδο του συστήματος. Επειδή η επιθυμητή απόκριση ενός συστήματος κλειστού βρόγχου είναι γνωστή η έξοδος της ελεγχόμενης διαδικασίας παρακολουθείται, η μέτρηση της οποίας χρησιμοποιείται ως σήμα ανάδρασης στο σύστημα ελέγχου όπου και συγκρίνεται με την επιθυμητή τιμή της εξόδου. Στη συνέχεια το σήμα σφάλματος που υπολογίζεται αξιοποιείται κατάλληλα από τον ελεγκτή με σκοπό να μειωθεί ώστε η διαδικασία να οδηγηθεί προς το επιθυμητή τιμή εξόδου και παράγεται ένα σήμα το οποίο αντιστοιχεί στο σφάλμα-διαφορά που εμφανίζεται μεταξύ της επιθυμητής και της πραγματικής μετρούμενης απόκρισης του συστήματος. Η χρήση αυτού του σήματος για τον έλεγχο όλης της διεργασίας έχει ως αποτέλεσμα τη δημιουργία μιας σειράς λειτουργιών μέσα στο κλειστό βρόγχο, η

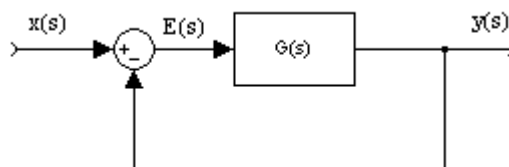
σειρά λειτουργιών αυτή ονομάζεται σύστημα ελέγχου με ανάδραση ή σύστημα αυτόματου ελέγχου κλειστού βρόγχου και διαδραματίζει βασικό ρόλο στα συστήματα αυτόματου ελέγχου. Στην επιθυμητή τιμή εξόδου της διαδικασίας η έξοδος ταυτίζεται με την επιθυμητή έξοδο αναφοράς και το σφάλμα είναι μηδενικό, κάτι που σημαίνει ότι με τη χρήση κλειστού βρόγχου με ανάδραση η συμπεριφορά ενός συστήματος βελτιώνεται με τη χρήση ανάδρασης.

Τα κλειστά συστήματα χρησιμοποιούνται σε περιπτώσεις όπου οι απαιτήσεις αυτοματισμού είναι μεγάλες ενώ τα ανοικτά συστήματα χρησιμοποιούνται σε απλές περιπτώσεις αυτοματισμού. Για τον παραπάνω λόγο τα κλειστά συστήματα είναι πιο πολύπλοκα στη σχεδίαση και στην υλοποίηση από τα ανοικτά. Τα ανοικτά που είναι λιγότερα πολύπλοκα έχουν χαμηλότερο κόστος, δεν αντιμετωπίζουν προβλήματα ευστάθειας σε αντίθεση με τα κλειστά. Για τα ανοικτά συστήματα η ακρίβεια που πετυχαίνουν στον έλεγχο της διεργασίας εξαρτάται από το κατά πόσο είναι γνωστή η συμπεριφορά της διεργασίας για δεδομένο σήμα εισόδου ενώ τα κλειστά έχουν υψηλό βαθμός ακρίβειας ακόμα και με περιορισμένη γνώση των παραμέτρων της διεργασίας. Τυχόν διαταραχές και μεταβολές της ακρίβειας των στοιχείων ενός ανοικτού συστήματος συνεπάγονται με εκτροπή της εξόδου από την επιθυμητή απόκριση ενώ τα κλειστά μπορούν να αντεπεξέλθουν σε εξωτερικές διαταραχές και σε μεταβολές των παραμέτρων της ελεγχόμενης διεργασίας. Αντιλαμβανόμαστε ότι τα κλειστά συστήματα υπερτερούν των ανοικτών συστημάτων και για αυτό χρησιμοποιούνται περισσότερο στη πράξη.

Ένα σύστημα ανοικτού βρόγχου που λειτουργεί χωρίς ανάδραση και παράγει απευθείας σήμα εξόδου ως απόκριση του συστήματος σε συγκεκριμένο σήμα εισόδου, παρουσιάζεται στην παρακάτω εικόνα.



Ένα σύστημα κλειστού βρόγχου που λαμβάνει ένα δείγμα του σήματος εξόδου το οποίο συγκρίνεται συνεχώς με την επιθυμητή έξοδο έτσι ώστε να παράγεται ένα σήμα σφάλματος που εφαρμόζεται ως σήμα εισόδου, παρουσιάζεται στην παρακάτω εικόνα.



Το παραπάνω σύστημα αν θέλουμε να το απλοποιήσουμε σε μια βαθμίδα τότε η νέα βαθμίδα θα έχει συνάρτηση μεταφοράς $\frac{y(s)}{x(s)} = \frac{G(s)}{1+G(s)}$. Το σύστημα αυτό ονομάζεται και σύστημα μοναδιαίας ανάδρασης αφού η ανάδραση

που έχουμε είναι ίση με τη μονάδα. Η συνάρτηση μεταφοράς προκύπτει από τα εξής:

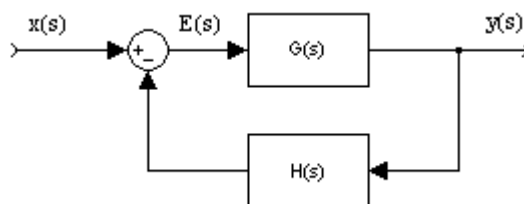
$$\begin{aligned} \frac{y(s)}{E(s)} = G(s) &\leftrightarrow y(s) = G(s) * E(s) \leftrightarrow y(s) = G(s) * (x(s) - y(s)) \leftrightarrow \\ &\leftrightarrow y(s) = G(s) * x(s) - G(s) * y(s) \leftrightarrow y(s) + G(s) * y(s) = G(s) * x(s) \leftrightarrow \\ &\leftrightarrow (1 + G(s)) * y(s) = G(s) * x(s) \leftrightarrow y(s) = \frac{G(s)}{1 + G(s)} * x(s) \leftrightarrow \frac{y(s)}{x(s)} = \frac{G(s)}{1 + G(s)} \end{aligned}$$

Το σήμα σφάλματος του παραπάνω κλειστού βρόγχου,

$$E(s) = \frac{1}{1 + G(s)} * x(s)$$

Οπότε για να ελαχιστοποιήσουμε το σήμα σφάλματος θα πρέπει $1 + G(s) > 1$ για όλες τις τιμές της μιγαδικής συχνότητας s .

Εκτός από τη μοναδιαία ανάδραση έχουμε και τη μορφή ανάδρασης που παρουσιάζεται στην επόμενη εικόνα.



Με συνάρτηση μεταφοράς $\frac{y(s)}{x(s)} = \frac{G(s)}{1 + H(s) * G(s)}$ το οποίο αποδεικνύεται ως εξής:

$$\begin{aligned} \frac{y(s)}{E(s)} = G(s) &\leftrightarrow y(s) = G(s) * E(s) \leftrightarrow y(s) = G(s) * (x(s) - H(s) * y(s)) \leftrightarrow \\ &\leftrightarrow y(s) = G(s) * x(s) - G(s) * H(s) * y(s) \leftrightarrow y(s) + G(s) * H(s) * y(s) = G(s) * x(s) \leftrightarrow \\ &\leftrightarrow (1 + G(s) * H(s)) * y(s) = G(s) * x(s) \leftrightarrow y(s) = \frac{G(s)}{1 + H(s) * G(s)} * x(s) \leftrightarrow \\ &\leftrightarrow \frac{y(s)}{x(s)} = \frac{G(s)}{1 + H(s) * G(s)} \end{aligned}$$

Σε αυτή τη περίπτωση το σήμα σφάλματος είναι,

$$E(s) = \frac{1}{1 + H(s) * G(s)} * x(s)$$

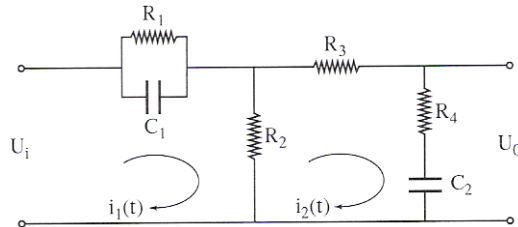
Επομένως για να ελαχιστοποιήσουμε το σήμα σφάλματος θα πρέπει $1 + H(s) * G(s) > 1$.

Στη επόμενη σελίδα ακολουθεί πίνακας με τους πιο σημαντικούς μετασχηματισμούς δομικών διαγραμμάτων.

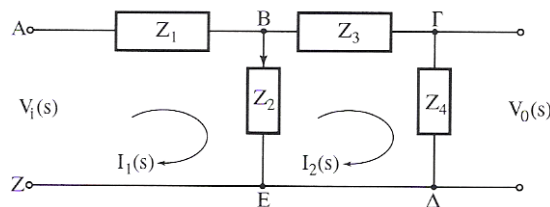
Μετασχηματισμοί δομικών διαγραμμάτων		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

5.5 Ασκήσεις Πράξης Ανάλυσης Συστημάτων Αυτόματου Ελέγχου

Άσκηση 1: Να βρεθεί η συνάρτηση μεταφοράς του κυκλώματος του παρακάτω σχήματος θεωρώντας ως σήμα εισόδου την τάση στην είσοδο του κυκλώματος και ως έξοδο τη τάση στην έξοδο του κυκλώματος.



Αρχικά θα υπολογίσουμε την σύνθετη αντίσταση Z_1 που αποτελείται από την αντίσταση R_1 και από τον πυκνωτή C_1 και την Z_2 που αποτελείται από την αντίσταση R_2 και από τον πυκνωτή C_2 . Οπότε το κύκλωμα μετασχηματίζεται ως εξής:



Όπου οι σύνθετες αντιστάσεις μετασχηματισμένες κατά Laplace είναι,

$$Z_{1(s)} = \frac{R_1 * \frac{1}{s * C_1}}{R_1 + \frac{1}{s * C_1}} = \frac{R_1}{R_1 * C_1 * s + 1}$$

$$Z_{2(s)} = R_2$$

$$Z_{3(s)} = R_3$$

και

$$Z_{4(s)} = R_4 + \frac{1}{s * C_2} = \frac{R_4 * C_2 * s + 1}{s * C_2}$$

Από το παραπάνω κύκλωμα προκύπτουν οι παρακάτω εξισώσεις κατά μετασχηματισμό Laplace:

$$V_{i(s)} = Z_{1(s)} * I_{1(s)} + Z_{2(s)} * (I_{1(s)} - I_{2(s)})$$

$$V_{o(s)} = Z_{4(s)} * I_{2(s)}$$

και

$$(I_{1(s)} - I_{2(s)}) * Z_{2(s)} = I_{2(s)} * (Z_{3(s)} + Z_{4(s)})$$

Επομένως η συνάρτηση μεταφοράς του κυκλώματος είναι:

$$G_{(s)} = \frac{V_{o(s)}}{V_{i(s)}} = \frac{Z_{4(s)} * I_{2(s)}}{Z_{1(s)} * I_{1(s)} + Z_{2(s)} * (I_{1(s)} - I_{2(s)})} \leftrightarrow$$

$$\Leftrightarrow G(s) = \frac{\frac{R_4 * C_2 * s + 1}{s * C_2} * I_{2(s)}}{\frac{R_1}{R_1 * C_1 * s + 1} * I_{1(s)} + R_2 * (I_{1(s)} - I_{2(s)})}$$

Λύνοντας τη σχέση $(I_{1(s)} - I_{2(s)}) * Z_{2(s)} = I_{2(s)} * (Z_{3(s)} + Z_{4(s)})$ έχουμε:

$$I_{1(s)} = I_{2(s)} * \frac{(R_2 + R_3 + R_4) * s * C_2}{R_2 * C_2 * s}$$

Άρα η συνάρτηση μεταφοράς γίνεται:

$$G(s) = \frac{\frac{R_4 * C_2 * s + 1}{s * C_2} * I_{2(s)}}{I_{2(s)} * \left(\left(\frac{R_1}{R_1 * C_1 * s + 1} * \frac{(R_2 + R_3 + R_4) * s * C_2}{R_2 * C_2 * s} \right) + (R_2 * \left(\frac{(R_2 + R_3 + R_4) * s * C_2}{R_2 * C_2 * s} - 1 \right)) \right)}$$

$$\Leftrightarrow G(s) = \frac{R_4 * C_2 * s + 1}{\frac{R_1 * [(R_2 + R_3 + R_4) * s * C_2 + 1]}{R_2 * (R_1 * C_1 * s + 1)} + (R_2 + R_3 + R_4) * C_2 * s + 1 - R_2 * C_2 * s}$$

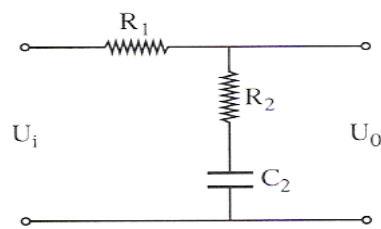
$$\Leftrightarrow G(s) = \frac{R_2 * (R_4 * C_2 * s + 1) * (R_1 * C_1 * s + 1)}{[R_1 * R_2 * (R_3 + R_4) * C_1 * C_2] s^2 + [R_1 * (R_2 + R_3 + R_4) * C_2 + (R_3 + R_4) * R_2 * C_2 + R_1 * R_2 * C_1] * s + R_1 + R_2}$$

Ο παρανομαστής είναι πολυώνυμο 2^{ου} βαθμού, οπότε βρίσκουμε τις ρίζες του μέσω της διακρίνουσας και η συνάρτηση μεταφοράς απλοποιείται και ισούται με:

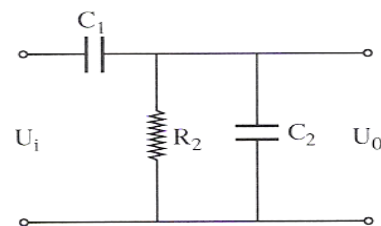
$$G(s) = \frac{R_2 * (R_4 * C_2 * s + 1) * (R_1 * C_1 * s + 1)}{R_1 * R_2 * (R_3 + R_4) * C_1 * C_2 * \left[s + \frac{1}{(R_3 + R_4) * C_2} \right] \left[s + \frac{R_1 * R_2 + R_1 * R_3 + R_1 * R_4 + R_2 * R_3 + R_2 * R_4}{R_1 * R_2 * C_1 * (R_3 + R_4)} \right]}$$

Άσκηση 2: Να βρεθούν οι συναρτήσεις μεταφοράς $G(s) = \frac{V_o(s)}{V_i(s)}$ των παρακάτω ηλεκτρικών κυκλωμάτων.

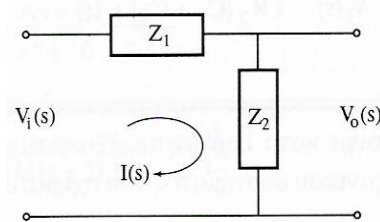
i)



ii)



Τα δύο κυκλώματα της άσκησης θα θεωρήσουμε ότι έχουν και τα δύο τη μορφή του παρακάτω σχήματος.



i) Για αυτό το κύκλωμα ισχύει:

$$Z_{1(s)} = R_1$$

$$Z_{2(s)} = R_2 + \frac{1}{s * C_2} = \frac{R_2 * C_2 * s + 1}{s * C_2}$$

$$V_{o(s)} = Z_{2(s)} * I(s)$$

και

$$V_{i(s)} = (Z_{1(s)} + Z_{2(s)}) * I(s)$$

Άρα η συνάρτηση μεταφοράς του κυκλώματος είναι:

$$G(s) = \frac{V_{o(s)}}{V_{i(s)}} = \frac{Z_{2(s)} * I(s)}{(Z_{1(s)} + Z_{2(s)}) * I(s)} = \frac{Z_{2(s)}}{Z_{1(s)} + Z_{2(s)}} \leftrightarrow$$

$$\leftrightarrow G(s) = \frac{\frac{R_2 * C_2 * s + 1}{s * C_2}}{R_1 + \frac{R_2 * C_2 * s + 1}{s * C_2}} = \frac{R_2 * C_2 * s + 1}{R_1 * C_2 * s + R_2 * C_2 * s + 1} \leftrightarrow$$

$$\leftrightarrow G(s) = \frac{R_2 * C_2 * s + 1}{(R_1 + R_2) * C_2 * s + 1}$$

ii) Για το δεύτερο κύκλωμα ισχύουν ομοίως

$$V_{o(s)} = Z_{2(s)} * I(s)$$

και

$$V_{i(s)} = (Z_{1(s)} + Z_{2(s)}) * I(s)$$

Αλλά διαφέρουν οι σύνθετες αντιστάσεις μετασχηματισμένες κατά Laplace,

$$Z_{1(s)} = \frac{1}{s * C_1}$$

και

$$Z_{2(s)} = \frac{R_2 * \frac{1}{s * C_2}}{R_2 + \frac{1}{s * C_2}} = \frac{R_2}{R_2 * C_2 * s + 1}$$

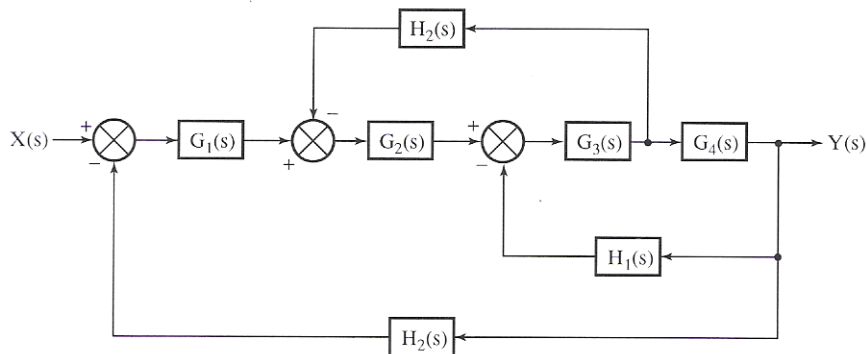
Άρα η συνάρτηση μεταφοράς του κυκλώματος είναι:

$$G(s) = \frac{V_{o(s)}}{V_{i(s)}} = \frac{Z_{2(s)} * I(s)}{(Z_{1(s)} + Z_{2(s)}) * I(s)} = \frac{Z_{2(s)}}{Z_{1(s)} + Z_{2(s)}} \leftrightarrow$$

$$\Leftrightarrow G(s) = \frac{\frac{R_2}{R_2 * C_2 * s + 1}}{\frac{1}{s * C_1} + \frac{R_2}{R_2 * C_2 * s + 1}} = \frac{R_2 * C_1 * s}{R_2 * C_2 * s + 1 + R_2 * C_1 * s} \Leftrightarrow$$

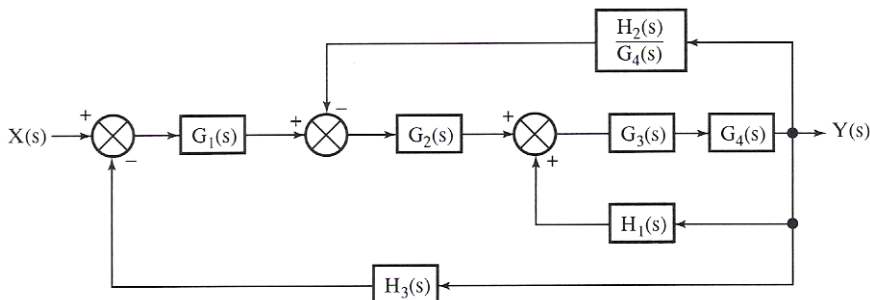
$$\Leftrightarrow G(s) = \frac{R_2 * C_1 * s}{(C_1 + C_2) * R_2 * s + 1}$$

Άσκηση 3: Να βρεθεί η συνάρτηση μεταφοράς του δομικού διαγράμματος του σχήματος απλοποιώντας αρχικά το παρακάτω δομικό διάγραμμα.

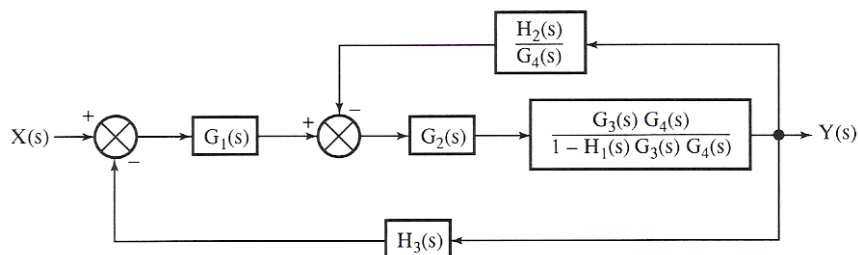


Για να λύσουμε την άσκηση απλοποιώντας το δομικό διάγραμμα θα χρησιμοποιήσουμε τον πίνακα Μετασχηματισμών δομικών διαγραμμάτων που βρίσκεται στη σελίδα 102.

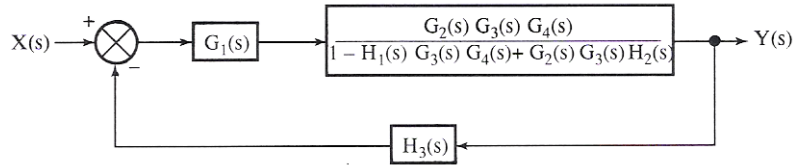
Με χρήση του μετασχηματισμού 7 των δομικών διαγραμμάτων μεταφέρουμε το σημείο λήψης δεξιά της βαθμίδας με συνάρτηση μεταφοράς $G_4(s)$ και προκύπτει το νέο δομικό διάγραμμα.



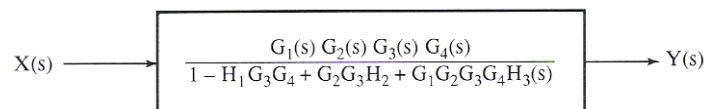
Με την εφαρμογή των μετασχηματισμών 1 και 3 για τις βαθμίδες με συναρτήσεις μεταφοράς $G_3(s)$, $G_4(s)$ και $H_1(s)$ το δομικό διάγραμμα απλοποιείται στην παρακάτω μορφή.



Με τους μετασχηματισμούς 1 και 3 για το βρόγχο που περιέχει τη βαθμίδα με συνάρτηση μεταφοράς $\frac{H_2(s)}{G_4(s)}$ ως ανατροφοδότηση, προκύπτει το παρακάτω απλοποιημένο δομικό διάγραμμα.



Τέλος, ομοίως με τους μετασχηματισμούς 1 και 3 καταλήγουμε στην τελική απλοποιημένη μορφή του αρχικού δομικού διαγράμματος που παρουσιάζεται στην παρακάτω εικόνα.

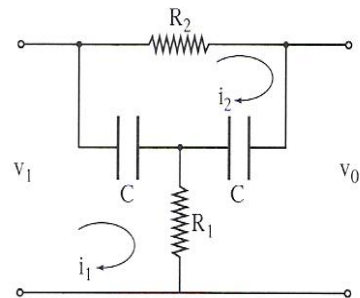


Όπως φαίνεται η συνάρτηση μεταφοράς του απλοποιημένου δομικού διαγράμματος ισούται με,

$$F(s) = \frac{G_1(s) * G_2(s) * G_3(s) * G_4(s)}{1 - H_1(s) * G_3(s) * G_4(s) + G_2(s) * G_3(s) * H_2(s) + G_1(s) * G_2(s) * G_3(s) * G_4(s) * H_3(s)}$$

Άσκηση 4: Να βρεθούν οι συναρτήσεις μεταφοράς των παρακάτω ηλεκτρικών κυκλωμάτων.

i)



Για το κύκλωμα του σχήματος ισχύει:

$$\begin{cases} V_{i(s)} = \left(R_1 + \frac{1}{C * s} \right) * I_{1(s)} - \frac{1}{C * s} * I_{2(s)} \\ 0 = -\frac{1}{C * s} * I_{1(s)} + \left(R_2 + \frac{2}{C * s} \right) * I_{2(s)} \end{cases}$$

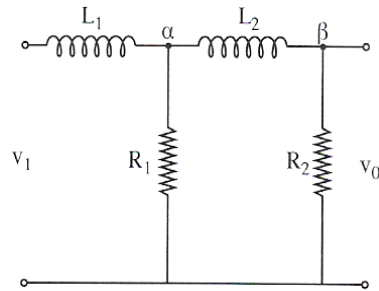
Λύνοντας το παραπάνω σύστημα ως προς $I_{2(s)}$ έχουμε:

$$I_{2(s)} = \frac{V_{i(s)}}{R_1 * R_2 * C^2 * s^2 + (2 * R_1 + R_2) * C * s + 1}$$

Αντικαθιστώ στη σχέση $V_{o(s)} = V_{i(s)} - I_{2(s)} * R_2$, το $I_{2(s)}$ και προκύπτει η συνάρτηση μεταφοράς του κυκλώματος, που ισούται με:

$$\frac{V_{o(s)}}{V_{i(s)}} = \frac{R_1 * R_2 * C^2 * s^2 + 2 * R_1 * C * s + 1}{R_1 * R_2 * C^2 * s^2 + (2 * R_1 + R_2) * C * s + 1}$$

ii)



Εφαρμόζοντας τη μέθοδο των κόμβων, οι εξισώσεις για τους κόμβους α και β είναι:

$$\frac{V_{\alpha(s)} - V_{i(s)}}{s * L_1} + \frac{V_{\alpha(s)}}{R_1} + \frac{V_{\alpha(s)} - V_{\beta(s)}}{s * L_2} = 0$$

και

$$\frac{V_{\beta(s)} - V_{\alpha(s)}}{s * L_2} + \frac{V_{\beta(s)}}{R_2} = 0$$

Ισχύει $V_{o(s)} = V_{\beta(s)}$, επομένως καταλήγουμε στη σχέση,

$$\frac{V_{o(s)}}{V_{i(s)}} = \frac{R_1 * R_2}{L_1 * L_2 * s^2 + (L_1 * R_1 + L_1 * R_2 + L_2 * R_1) * s + R_1 * R_2}$$

που είναι και η συνάρτηση μεταφοράς του κυκλώματος.

6. Χρονική απόκριση Συστημάτων Αυτόματου Ελέγχου

Με τον όρο χρονική απόκριση ενός συστήματος εννοούμε τη συμπεριφορά του συστήματος με τη πάροδο του χρόνου σε μια συγκεκριμένη είσοδο. Μπορούμε δηλαδή αν μας δίνεται ένα σύστημα περιγραφόμενο από την συνάρτηση μεταφοράς του και μας δοθεί η είσοδος $x(t)$ ως συνάρτηση του χρόνου να υπολογιστεί η έξοδος $y(t)$. Η χρονική απόκριση ενός συστήματος ελέγχου αποτελείται από δύο μέρη:

1. Την μεταβατική απόκριση (transient response) και
2. Την απόκριση μόνιμης κατάστασης (steady state response)

Με τον όρο μεταβατική εννοούμε την απόκριση του συστήματος αμέσως μετά τη διέγερση του και πριν από τη σταθεροποίηση της εξόδου του, είναι δηλαδή το μέρος εκείνο της απόκρισης που τείνει στο μηδέν καθώς ο χρόνος μεγαλώνει.

Με τον όρο μόνιμη απόκριση εννοούμε το μέρος της απόκρισης το οποίο εμφανίζεται αφού πρώτα έχει εξασθενήσει το μεταβατικό μέρος. Η μόνιμη απόκριση μπορεί να είναι μια σταθερά, μια ταλάντωση με σταθερό πλάτος, μια συνάρτηση που αυξάνει με το χρόνο κ.λπ.

Επειδή τα περισσότερα συστήματα είναι από τη φύση τους δυναμικά, η αντίστοιχη συμπεριφορά τους καθορίζεται συνήθως συναρτήσει τόσο της μεταβατικής τους απόκρισης, όσο και της απόκρισης τους στη μόνιμη κατάσταση.

Οι προδιαγραφές της συμπεριφοράς ενός συστήματος που ορίζονται στο πεδίο του χρόνου αποτελούν πολύ σημαντικές παραμέτρους επειδή τα διάφορα συστήματα ελέγχου είναι από την ίδια τους την φύση συστήματα που λειτουργούν στο πεδίο του χρόνου. Αυτό σημαίνει ουσιαστικά ότι η χρονική απόκριση ενός συστήματος είναι κατά κύριο λόγο εκείνο το χαρακτηριστικό που μας ενδιαφέρει περισσότερο. Η απόκριση ενός συστήματος σε ένα σήμα εισόδου μας παρέχει πολλές πληροφορίες σχετικά με την γενικότερη συμπεριφορά του συστήματος.

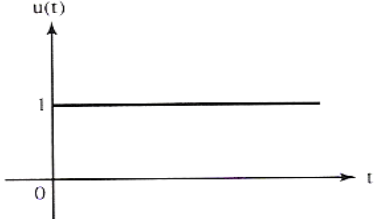
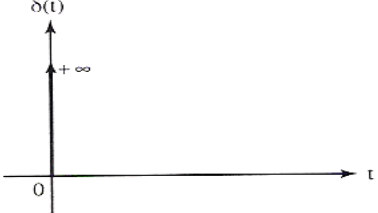
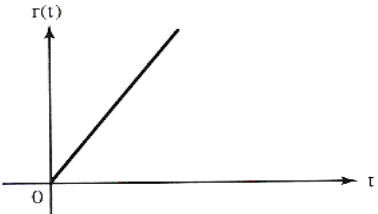
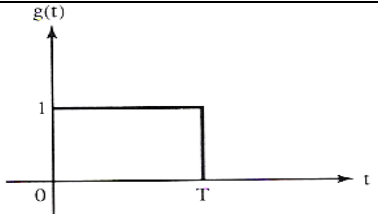
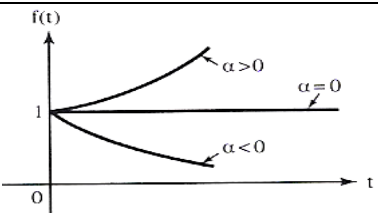
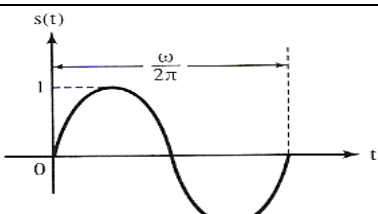
6.1 Τυποποιημένα Σήματα Εισόδου

Επειδή η πραγματική μορφή των διαφόρων σημάτων εισόδου μας είναι συνήθως άγνωστη, χρησιμοποιούμε κάποια τυποποιημένα σήματα διέγερσης. Τα συνηθέστερα σήματα διέγερσης είναι η συνάρτηση βαθμίδας, η συνάρτηση αναρρίχησης, η κρουστική συνάρτηση, η εκθετική συνάρτηση και η ημιτονοειδής συνάρτηση οι οποίες έχουν αναφερθεί στο 4^ο κεφάλαιο.

Η μέθοδος χρήσης τυποποιημένων σημάτων είναι αρκετά χρήσιμη αφού μας προσδίδει σε μεγάλο βαθμό μια συσχέτιση μεταξύ της απόκρισης

ενός συστήματος στην τυποποιημένη διέγερση και της ικανότητας λειτουργίας του συστήματος υπό κανονικές συνθήκες επομένως μας επιτρέπει να συγκρίνουμε διάφορες σχεδιαστικές προτάσεις για το ίδιο σύστημα.

Στον παρακάτω πίνακα παρουσιάζονται τα τυποποιημένα σήματα εισόδου, οι γραφικές τους παραστάσεις και οι μετασχηματισμοί Laplace.

Σήμα	Γραφική παράσταση	Μετασχηματισμός Laplace
Μοναδιαία Βηματική συνάρτηση $u(t)$		$U(s) = \frac{1}{s}$
Μοναδιαία Κρουστική συνάρτηση $\delta(t)$		$D(s) = 1$
Μοναδιαία συνάρτηση Αναρρίχησης $r(t)$		$R(s) = \frac{1}{s^2}$
Μοναδιαίος Παλμός $y(t)$		$Y(s) = \frac{1}{s} - \frac{1}{s} * e^{-Ts}$
Εκθετική συνάρτηση $y(t)$		$Y(s) = \frac{1}{s - \alpha}$
Ημιτονοειδής συνάρτηση $y(t)$		$Y(s) = \frac{\omega}{s^2 + \omega^2}$

6.2 Χαρακτηριστικά της γραφικής παράστασης της χρονικής απόκρισης

Τα χαρακτηριστικά της γραφικής παράστασης της χρονικής απόκρισης είναι τα στοιχεία αυτά τα οποία μας δίνουν μια εποπτική εικόνα της συμπεριφοράς του συστήματος. Αυτά τα χαρακτηριστικά είναι τα εξής:

✚ Μέγιστη υπερύψωση (M_p): Είναι η διαφορά της μέγιστης τιμής και της τελικής τιμής της εξόδου και συνηθίζεται να εκφράζεται σε ποσοστό επί τις εκατό. Έστω y_m η μέγιστη τιμή της εξόδου και y_f η τελική τιμή της εξόδου τότε το ποσοστό υπερύψωσης ορίζεται:

$$M_p = \frac{y_m - y_f}{y_f} * 100 \%$$

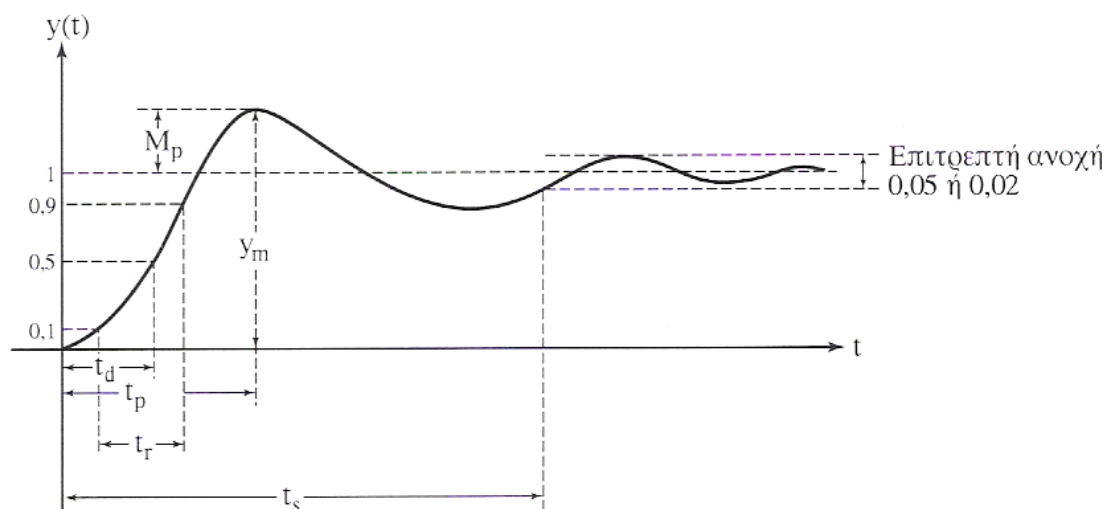
✚ Χρόνος ανύψωσης (t_r , rise time): Είναι ο χρόνος που χρειάζεται για να ανέλθει η απόκριση από το 10% στο 90% της τελικής της τιμής.

✚ Χρόνος κορυφής (t_p , peak time): Είναι ο χρόνος που χρειάζεται για να φτάσει η απόκριση στην μέγιστη τιμή της εξόδου.

✚ Χρόνος αποκατάστασης (t_s , settling time): Είναι ο χρόνος που χρειάζεται για να φτάσει και να παραμείνει η καμπύλη απόκρισης ανάμεσα στο $\pm 2\%$ ή στο $\pm 5\%$ της τελικής τιμής.

✚ Χρόνος καθυστέρησης (t_d , delay time): Είναι ο χρόνος που χρειάζεται για να φτάσει η απόκριση για πρώτη φορά το 50% της τελικής τιμής.

Στο παρακάτω σχήμα παρουσιάζεται μια γενική μορφή της απόκρισης ενός γραμμικού συστήματος, στο οποίο παριστάνονται και τα χαρακτηριστικά της γραφικής παράστασης της χρονικής απόκρισης.



6.3 Συστήματα πρώτης τάξης

Υπενθυμίζουμε ότι η τάξη του συστήματος δίνεται από τη μεγαλύτερη δύναμη στην οποία είναι υψωμένη η μιγαδική συχνότητα s στο πολυώνυμο του παρανομαστή, στην ουσία η τάξη ενός συστήματος μας δίνει τον αριθμό των πόλων του συστήματος.

Η γενική μορφή της συνάρτησης μεταφοράς ενός συστήματος πρώτης τάξης είναι,

$$G(s) = \frac{b}{s + a}$$

την οποία εμείς θα φέρνουμε στη μορφή,

$$G(s) = \frac{K}{T * s + 1}$$

όπου, K είναι η σταθερά κέρδους ή ενίσχυσης και

T είναι η σταθερά χρόνου, που είναι ο χρόνος που η απόκριση του συστήματος φτάνει το 63,21% της τελικής τιμής της.

Θα περιγράψουμε τη διαδικασία με τη χρήση παραδειγμάτων όπου θα αλλάζουμε το σήμα εισόδου, ώστε να γίνει κατανοητή.

Η παραπάνω συνάρτηση μεταφοράς για $K=1$ θα είναι,

$$G(s) = \frac{1}{T * s + 1}$$

Σε περίπτωση που το σύστημα με την παραπάνω συνάρτηση μεταφοράς διεγείροταν με σήμα στην είσοδο μοναδιαία βηματική συνάρτηση ($X(s) = \frac{1}{s}$) τότε η απόκριση θα τη βρίσκαμε ως εξής:

$$G(s) = \frac{Y(s)}{X(s)} \leftrightarrow \frac{1}{T * s + 1} = \frac{Y(s)}{\frac{1}{s}} \leftrightarrow Y(s) = \frac{1}{T * s + 1} * \frac{1}{s}$$

Αναλύοντας την $Y(s)$ σε απλά κλάσματα και βρίσκοντας τον αντίστροφο μετασχηματισμό Laplace υπολογίζουμε την απόκριση του συστήματος στο πεδίο του χρόνου:

$$Y(s) = \frac{1}{s} - \frac{1}{s + \frac{1}{T}} \xrightarrow{\mathcal{L}^{-1}} y(t) = 1 - e^{-\frac{t}{T}}$$

Αντιλαμβανόμαστε λοιπόν ότι η απόκριση του συστήματος επηρεάζεται σε μεγάλο βαθμό από τη σταθερά χρόνου T . Όσο πιο μικρή είναι η σταθερά χρόνου τόσο πιο μεγάλη είναι η ταχύτητα απόκρισης του συστήματος. Με τον όρο ταχύτητα απόκρισης εννοούμε το χρόνο που απαιτείται για να φτάσει η έξοδος το 63,21% της τελικής τιμής της, ο χρόνος αυτός συμπίπτει με τη σταθερά χρόνου. Αυτό αποδεικνύεται αν χρησιμοποιήσουμε τη συνάρτηση της εξόδου για $t=T$:

$$y(T) = 1 - e^{-\frac{T}{T}} = 1 - e^{-1} = 1 - 0.3679 = 0.6321$$

Στη περίπτωση που στο παραπάνω σύστημα έχουμε για είσοδο το σήμα της μοναδιαίας κρουστικής συνάρτησης ($X_{(s)} = 1$), τότε η απόκριση του συστήματος είναι:

$$G_{(s)} = \frac{Y_{(s)}}{X_{(s)}} \leftrightarrow \frac{1}{T * s + 1} = \frac{Y_{(s)}}{1} \leftrightarrow Y_{(s)} = \frac{1}{T * s + 1}$$

$$Y_{(s)} = \frac{1}{T * s + 1} \xrightarrow{\mathcal{L}^{-1}} y_{(t)} = \frac{1}{T} e^{-\frac{t}{T}}$$

Τέλος σε περίπτωση που για είσοδο δώσουμε σήμα μοναδιαίας συνάρτησης αναρρίχησης ($X_{(s)} = \frac{1}{s^2}$) η απόκριση του συστήματος θα ισούται:

$$G_{(s)} = \frac{Y_{(s)}}{X_{(s)}} \leftrightarrow \frac{1}{T * s + 1} = \frac{Y_{(s)}}{\frac{1}{s^2}} \leftrightarrow Y_{(s)} = \frac{1}{T * s + 1} * \frac{1}{s^2}$$

$$Y_{(s)} = \frac{1}{T * s + 1} * \frac{1}{s^2} \xrightarrow{\mathcal{L}^{-1}} y_{(t)} = t - T + T * e^{-\frac{t}{T}}$$

Παρατηρούμε λοιπόν ότι μπορούμε να υπολογίσουμε την έξοδο του συστήματος που μελετάμε για διαφορετικές εισόδους. Με αυτό τον τρόπο μπορούμε να μελετήσουμε και τη συμπεριφορά του με την πάροδο του χρόνου.

6.3.1 Βηματική απόκριση

Θα χρησιμοποιούμε τον όρο *βηματική απόκριση* όταν θα θέλουμε να υπολογίσουμε την έξοδο-απόκριση ενός συστήματος αυτόματου ελέγχου το οποίο διεγείρουμε με σήμα εισόδου τη μοναδιαία βηματική συνάρτηση. Δηλαδή αν $G_{(s)}$ η συνάρτηση μεταφοράς του συστήματος τότε η βηματική απόκριση του θα ισούται με:

$$G_{(s)} = \frac{Y_{(s)}}{X_{(s)}} \leftrightarrow Y_{(s)} = \frac{G_{(s)}}{s} \leftrightarrow y_{(t)} = \mathcal{L}^{-1} \left\{ \frac{G_{(s)}}{s} \right\}$$

Εκτός από το να υπολογίζουμε την απόκριση του συστήματος το MatLab μας δίνει τη δυνατότητα με την εντολή *step* και να τη σχεδιάσουμε. Για το παράδειγμα που χρησιμοποιήσαμε παραπάνω σχεδιάζουμε την βηματική απόκριση του συστήματος με τις παρακάτω εντολές(για σταθερά χρόνου $T=2$).

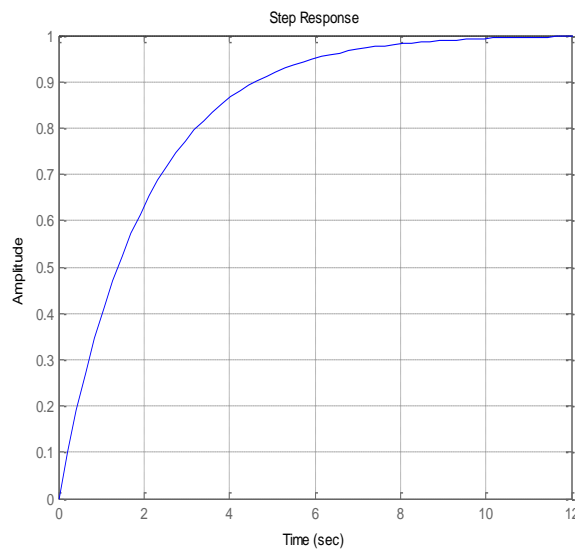
```
G=tf([1],[2 1])
```

```
Transfer function:
```

```
1
-----
2 s + 1
```

```
step(G);grid on;
```

Η βηματική απόκριση του συστήματος φαίνεται στο παρακάτω διάγραμμα.



6.3.2 Κρουστική απόκριση

Θα χρησιμοποιούμε τον όρο *κρουστική απόκριση* όταν θα θέλουμε να υπολογίσουμε την έξοδο-απόκριση ενός συστήματος αυτόματου ελέγχου το οποίο διεγείρουμε με σήμα εισόδου τη μοναδιαία κρουστική συνάρτηση. Αν $G(s)$ η συνάρτηση μεταφοράς του συστήματος τότε η κρουστική απόκριση του θα ισούται με:

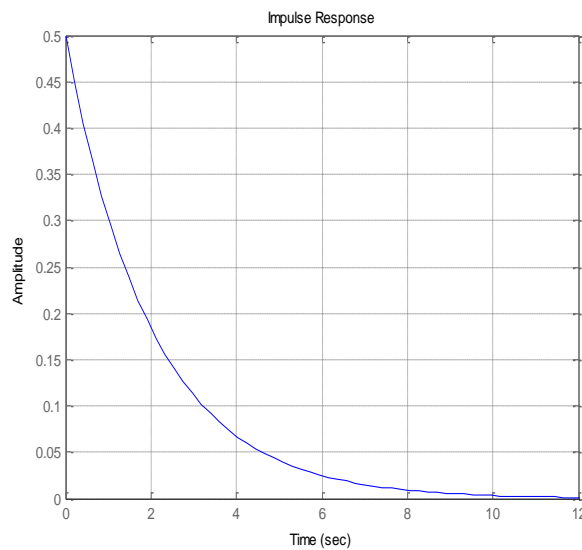
$$G(s) = \frac{Y(s)}{X(s)} \leftrightarrow Y(s) = \frac{G(s)}{1} \leftrightarrow y(t) = \mathcal{L}^{-1}\{G(s)\}$$

Στη ουσία η κρουστική απόκριση ενός συστήματος είναι ο αντίστροφος μετασχηματισμός Laplace της συνάρτησης μεταφοράς του συστήματος.

Όπως και με τη βηματική το MatLab μας δίνει τη δυνατότητα με την εντολή *impulse* και να τη σχεδιάσουμε. Ομοίως για το παράδειγμα που χρησιμοποιήσαμε παραπάνω σχεδιάζουμε την κρουστική απόκριση του συστήματος με τις παρακάτω εντολές (για σταθερά χρόνου $T=2$).

```
G=tf([1],[2 1]);
impulse(G);grid on;
```

Η κρουστική απόκριση του συστήματος είναι η εξής:

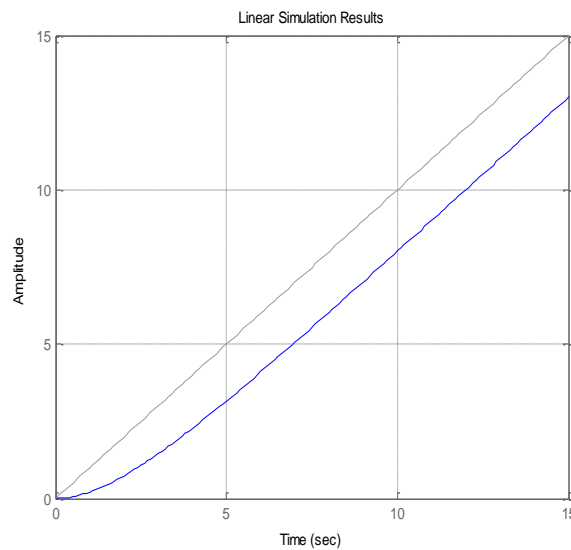


6.3.3 Απόκριση Γενικά

Σε παράδειγμα που αναφέρθηκε είδαμε πως μπορούμε να υπολογίσουμε την έξοδο-απόκριση του συστήματος για οποιοδήποτε σήμα εισόδου (στο παράδειγμα που δώσαμε αναφερόταν στη μοναδιαία συνάρτηση αναρρίχησης). Αν όμως επιθυμούμε να σχεδιάσουμε την απόκριση του συστήματος για οποιοδήποτε σήμα εισόδου εκτός της μοναδιαίας βηματικής και κρουστικής συνάρτησης το MatLab μας δίνει τη δυνατότητα αυτή με τη χρήση της εντολής *lsim*. Η ιδιαιτερότητα αυτής της εντολής σε σχέση με τις προηγούμενες δύο είναι ότι δεν αρκεί να γνωρίζουμε μόνο την συνάρτηση μεταφοράς του συστήματος. Θα πρέπει να γνωρίζουμε και την είσοδο την οποία θα εισάγουμε με τη μορφή διάνυσματος (σ' αυτό χρησιμεύουν τα βασικά σήματα) και θα πρέπει να δηλώσουμε και το χρονικό ορίζοντα της απόκρισης κάτι που στις προηγούμενες δύο περιπτώσεις το MatLab έκανε αυτόματα. Η εντολή έχει την εξής μορφή *lsim(G,u,T)* όπου G η συνάρτηση μεταφοράς του συστήματος του οποίου θέλουμε να σχεδιάσουμε την απόκριση, u το διάνυσμα εισόδου και T ο χρονικός ορίζοντας. Για το παράδειγμα της μοναδιαίας συνάρτησης αναρρίχησης και για σταθερά χρόνο ίση με 2, θα σχεδιάζαμε την απόκριση του συστήματος με τις εξής εντολές:

```
G=tf([1],[2 1]);
T=0:0.1:15;
u=T;
lsim(G,u,T);grid on;
```

Η απόκριση του συστήματος είναι:



Η γκρι γραμμή που εμφανίζεται στο διάγραμμα είναι η γραφική της εισόδου που δώσαμε στο σύστημα και θα εξηγήσουμε στη συνέχεια γιατί η έξοδος παρεκκλίνει από αυτή.

6.4 Συστήματα δεύτερης τάξης

Η γενικευμένη μορφή της συνάρτησης μεταφοράς των συστημάτων 2^{ns} τάξης είναι η εξής:

$$G(s) = \frac{\omega_n^2}{s^2 + 2 * \zeta * \omega_n * s + \omega_n^2}$$

Όπου, ω_n η κυκλική ιδιοσυχνότητα του συστήματος και

ζ ο συντελεστής απόσβεσης του συστήματος.

Οι ρίζες του παρανομαστή δηλαδή οι πόλοι του συστήματος είναι:

- ✚ Για $\zeta = 0$ το σύστημα έχει μηδενική απόσβεση και έχει δύο φανταστικούς και συζυγείς πόλους που δίνονται από τη σχέση:

$$s_{1,2} = \pm j * \omega_n$$

- ✚ Για $0 < \zeta < 1$ το σύστημα έχει υπο-απόσβεση και δύο συζυγείς μιγαδικούς πόλους που δίνονται από τη σχέση:

$$s_{1,2} = -\zeta * \omega_n \pm j * \omega_n * \sqrt{1 - \zeta^2}$$

$$t_p = \frac{\pi}{\omega_d}$$

$$t_s = \frac{4}{\zeta * \omega_n}$$

$$t_r = \frac{\pi - \tan^{-1}\left(\frac{\omega_d}{\zeta * \omega_n}\right)}{\omega_d}$$

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

$$\zeta = \frac{-\ln M_p}{\sqrt{\pi^2 + \ln^2(M_p)}}$$

το φανταστικό τους μέρος των πόλων ονομάζεται με φυσική ιδιοσυχνότητα και συμβολίζεται ως $\omega_d = \omega_n * \sqrt{1 - \zeta^2}$.

- ✚ Για $\zeta = 1$ το σύστημα έχει κρίσιμη απόσβεση και ως διπλό πόλο την τιμή της κυκλικής συχνότητας με διαφορετικό πρόσημο:

$$s_{1,2} = -\omega_n$$

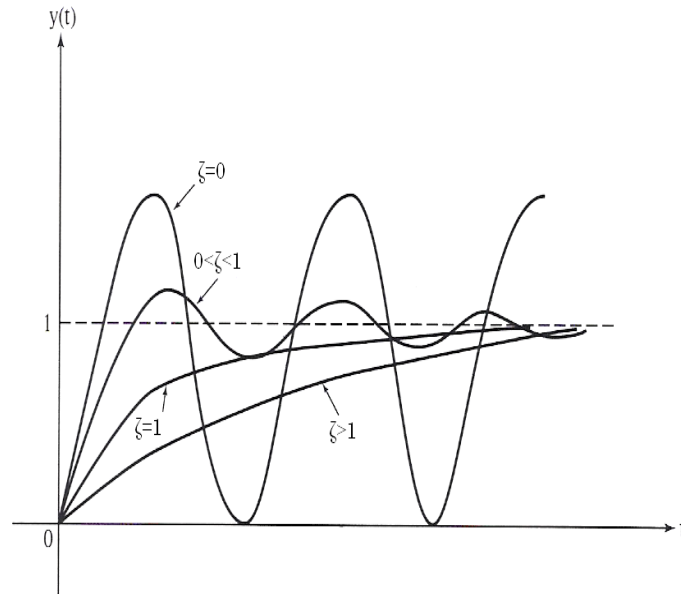
- ✚ Για $\zeta > 1$ το σύστημα έχει υπερ-απόσβεση και δύο πραγματικούς πόλους που δίνονται από τη σχέση:

$$s_{1,2} = -\zeta * \omega_n \pm \omega_n * \sqrt{\zeta^2 - 1}$$

Τα συστήματα που θα μας απασχολήσουν είναι αυτά με συντελεστή απόσβεσης $0 < \zeta < 1$. Ο συντελεστής απόσβεσης ζ μας δίνει πληροφορίες-εκτίμηση για τη μέγιστη υπερύψωση της απόκρισης των συστημάτων αλλά και το αντίστροφο. Στον παρακάτω πίνακα παρουσιάζονται οι αντιστοιχίες του συντελεστή απόσβεσης με τη μέγιστη υπερύψωση.

Συντελεστής απόσβεσης	Ποσοστό μέγιστης υπερύψωσης
0.3	37.2%
0.4	25.4%
0.5	16.3%
0.6	9.5%
0.7	4.6%
0.8	1.5%
0.9	0.2%

Παραπάνω αναφερθήκαμε στους όρους μηδενική απόσβεση, υπο-απόσβεση, υπερ-απόσβεση και κρίσιμη απόσβεση, οι όροι αυτοί αναφέρονται στην απόκλιση του συστήματος δηλαδή στη μορφή του γραφήματος της απόκλισης του συστήματος. Στο παρακάτω γράφημα παρουσιάζονται οι παραπάνω περιπτώσεις ώστε να γίνουν και αντιληπτές οι έννοιες.



6.5 Σφάλματα Συστημάτων Αυτόματου Ελέγχου

Ένας σημαντικός παράγοντας στη λειτουργία και την ανάλυση των συστημάτων αυτόματου ελέγχου είναι το σφάλμα στη μόνιμη κατάσταση λειτουργίας. Το σφάλμα στη μόνιμη κατάσταση λειτουργίας είναι το σφάλμα που εμφανίζεται στην έξοδο του συστήματος μετά τη πάροδο της μεταβατικής απόκρισης του συστήματος.

Το σφάλμα στη μόνιμη κατάσταση λειτουργίας ενός συστήματος ανοιχτού βρόχου με συνάρτηση μεταφοράς $G(s)$ είναι:

$$E(s) = X(s) - Y(s) \leftrightarrow E(s) = X(s) - G(s) * X(s) \leftrightarrow E(s) = X(s) * (1 - G(s))$$

Για να υπολογίσουμε τη τιμή του σφάλματος χρησιμοποιούμε το θεώρημα τελικής τιμής και είσοδο μοναδιαία βηματική συνάρτηση επομένως:

$$\begin{aligned} \lim_{t \rightarrow \infty} e(t) &= \lim_{s \rightarrow 0} (s * E(s)) \leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} (s * X(s) * (1 - G(s))) \leftrightarrow \\ &\leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} \left(s * \frac{1}{s} * (1 - G(s)) \right) \leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} ((1 - G(s))) \\ &\leftrightarrow e_{(\infty)} = 1 - G(0) \end{aligned}$$

Η τιμή της συνάρτησης μεταφοράς για $s = 0$, όπως έχουμε αναφέρει και στο 5^ο κεφάλαιο, είναι συνήθως μεγαλύτερη από τη μονάδα οπότε το σύστημα παρουσιάζει μια αξιοπρόσεκτη τιμή σφάλματος στη μόνιμη κατάσταση λειτουργίας. Όμως αν ρυθμίσουμε κατάλληλα τον όρο $G(0)$ έτσι ώστε να είναι ίσος με τη μονάδα μπορεί το σύστημα ανοιχτού βρόχου να παρουσιάσει μηδενικό σφάλμα. Σε ένα σύστημα ανοιχτού βρόχου αυτό μπορεί να επιτευχθεί μόνο με μεταβολή των παραμέτρων της συνάρτησης μεταφοράς οπότε σε περίπτωση πιθανών εξωτερικών μεταβολών-διαταραχών που επηρεάζουν το σύστημα κατά τη λειτουργία του, κάποια στιγμή η τιμή του όρου $G(0)$ δεν θα είναι ίση με τη μονάδα οπότε το σφάλμα πάλι δε θα είναι μη μη-

δενικό έως να μεταβάλουμε πάλι τις παραμέτρους της συνάρτησης μεταφοράς.

Ακολουθώντας όμοια μέθοδο με τα συστήματα ανοιχτού βρόγχου στα συστήματα κλειστού βρόγχου το σφάλμα μόνιμης κατάστασης λειτουργίας ισούται (για μοναδιαίο κλάδο ανάδραση):

$$\begin{aligned} E_{(s)} = X_{(s)} - Y_{(s)} &\leftrightarrow E_{(s)} = X_{(s)} - \frac{G_{(s)}}{1 + G_{(s)}} * X_{(s)} \leftrightarrow E_{(s)} = X_{(s)} * \left(1 - \frac{G_{(s)}}{1 + G_{(s)}}\right) \leftrightarrow \\ &\leftrightarrow E_{(s)} = X_{(s)} * \left(\frac{1 + G_{(s)} - G_{(s)}}{1 + G_{(s)}}\right) \leftrightarrow E_{(s)} = X_{(s)} * \frac{1}{1 + G_{(s)}} \end{aligned}$$

Με το θεώρημα τελικής τιμής και είσοδο μοναδιαία βηματική συνάρτηση υπολογίζουμε το σφάλμα:

$$\begin{aligned} \lim_{t \rightarrow \infty} e_{(t)} &= \lim_{s \rightarrow 0} (s * E_{(s)}) \leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} \left(s * X_{(s)} * \frac{1}{1 + G_{(s)}} \right) \leftrightarrow \\ &\leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} \left(s * \frac{1}{s} * \frac{1}{1 + G_{(s)}} \right) \leftrightarrow e_{(\infty)} = \lim_{s \rightarrow 0} \left(\frac{1}{1 + G_{(s)}} \right) \leftrightarrow \\ &\leftrightarrow e_{(\infty)} = \frac{1}{1 + G_{(0)}} \end{aligned}$$

Αντίθετα με τα συστήματα ανοιχτού βρόγχου τα συστήματα κλειστού βρόγχου παρουσιάζουν μικρό σφάλμα μόνιμης κατάστασης αφού όσο μεγαλύτερος είναι ο όρος $G_{(0)}$ δηλαδή το dc κέρδος του συστήματος τόσο μειώνεται το σφάλμα. Επίσης τα συστήματα κλειστού βρόγχου μπορούν να κρατούν σε χαμηλά επίπεδα ανεξάρτητα από τυχών εξωτερικές μεταβολές-διαταραχές αφού συνεχώς λαμβάνουν πληροφορίες από την έξοδο στην είσοδο του συστήματος οπότε ελέγχουν συνεχώς τη τιμή του σφάλματος και σκοπεύουν την μείωση του.

Το σφάλμα στη μόνιμη κατάσταση λειτουργίας ενός συστήματος κλειστού βρόγχου είναι αρκετά μικρότερο από ότι σε ένα σύστημα ανοιχτού βρόγχου γεγονός που αποτελεί το βασικότερο λόγο που χρησιμοποιούμε την ανάδραση παρά τη πολυπλοκότητα και το κόστος της.

6.6 Σταθερές Σφάλματος μόνιμης κατάστασης

Η απόδοση ενός συστήματος αυτόματου ελέγχου κλειστού βρόγχου σε σχέση με το σφάλμα μόνιμης κατάστασης εξαρτάται από τις τρεις ακόλουθες σταθερές σφάλματος που περιγράφουν την ικανότητα του συστήματος να περιορίσει ή να εξαλείψει το σφάλμα μόνιμης κατάστασης:

- ✚ Σταθερά σφάλματος θέσης K_p
- ✚ Σταθερά σφάλματος ταχύτητας K_v
- ✚ Σταθερά σφάλματος επιτάχυνσης K_a

Η ορολογία των παραπάνω σταθερών σφάλματος οφείλεται στο γεγονός ότι προέρχονται από εφαρμογές ελέγχου κίνησης.

6.6.1 Σταθερά σφάλματος θέσης K_p

Η σταθερά σφάλματος θέσης αναφέρεται σε βηματική είσοδο. Το σφάλμα στη μόνιμη κατάσταση ενός συστήματος με διέγερση στην είσοδο βηματικής συνάρτησης με πλάτος A είναι:

$$e_{ss} = \lim_{s \rightarrow 0} \left(s * X(s) * \frac{1}{1 + G(s)} \right) = \lim_{s \rightarrow 0} \left(s * \frac{A}{s} * \frac{1}{1 + G(s)} \right) = \frac{A}{1 + G(0)}$$

Ο σταθερός όρος $G(0)$ συμβολίζεται με K_p και αποτελεί το συντελεστή σφάλματος θέσης, οπότε το σφάλμα μόνιμης κατάστασης για βηματική είσοδο πλάτους A γράφεται:

$$e_{ss} = \frac{A}{1 + K_p}$$

6.6.2 Σταθερά σφάλματος ταχύτητας K_v

Η σταθερά σφάλματος ταχύτητας αναφέρεται σε είσοδο αναρρίχησης. Το σφάλμα στη μόνιμη κατάσταση ενός συστήματος με διέγερση στην είσοδο συνάρτησης αναρρίχησης με κλίση A είναι:

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} \left(s * X(s) * \frac{1}{1 + G(s)} \right) = \lim_{s \rightarrow 0} \left(s * \frac{A}{s^2} * \frac{1}{1 + G(s)} \right) = \lim_{s \rightarrow 0} \left(\frac{A}{s} * \frac{1}{1 + G(s)} \right) \leftrightarrow \\ &\leftrightarrow e_{ss} = \lim_{s \rightarrow 0} \left(\frac{A}{s + s * G(s)} \right) \leftrightarrow e_{ss} = \lim_{s \rightarrow 0} \left(\frac{A}{s * G(s)} \right) \end{aligned}$$

Το K_v ονομάζεται συντελεστής σφάλματος ταχύτητας και ορίζεται με τη σχέση:

$$K_v = \lim_{s \rightarrow 0} (s * G(s))$$

Οπότε το σφάλμα μόνιμης κατάστασης για είσοδο αναρρίχησης με κλίση A γράφεται:

$$e_{ss} = \frac{A}{K_v}$$

6.6.3 Σταθερά σφάλματος επιτάχυνσης K_a

Η σταθερά σφάλματος ταχύτητας αναφέρεται σε παραβολική είσοδο. Το σφάλμα στη μόνιμη κατάσταση ενός συστήματος με διέγερση στην είσοδο παραβολική συνάρτηση της μορφής $x(t) = \frac{A * t^2}{2}$ είναι:

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} \left(s * X(s) * \frac{1}{1 + G(s)} \right) = \lim_{s \rightarrow 0} \left(s * \frac{A}{s^3} * \frac{1}{1 + G(s)} \right) = \lim_{s \rightarrow 0} \left(\frac{A}{s^2} * \frac{1}{1 + G(s)} \right) \leftrightarrow \\ &\leftrightarrow e_{ss} = \lim_{s \rightarrow 0} \left(\frac{A}{s^2 + s^2 * G(s)} \right) \leftrightarrow e_{ss} = \lim_{s \rightarrow 0} \left(\frac{A}{s^2 * G(s)} \right) \end{aligned}$$

Το K_α ονομάζεται συντελεστής σφάλματος ταχύτητας και ορίζεται με τη σχέση:

$$K_\alpha = \lim_{s \rightarrow 0} (s^2 * G(s))$$

Οπότε το σφάλμα μόνιμης κατάστασης για είσοδο αναρρίχησης με κλίση A γράφεται:

$$e_{ss} = \frac{A}{K_\alpha}$$

Η σημασία των παραπάνω σταθερών σφάλματος είναι μεγάλη αφού οι τιμές τους ορίζουν και το σφάλμα μόνιμης κατάστασης του συστήματος. Όσο μεγαλύτερες είναι οι τιμές των σταθερών σφάλματος τόσο μικρότερο είναι το σφάλμα μόνιμης κατάστασης του συστήματος για το αντίστοιχο σήμα εισόδου. Εάν κάποια από τις παραπάνω σταθερές είναι μηδενική τότε το σφάλμα μόνιμης κατάστασης είναι άπειρο στο αντίστοιχο σήμα εισόδου, κάτι που σημαίνει ότι το σύστημα δεν μπορεί να παρακολουθήσει το σήμα αναφοράς.

Αυτό επιδιώκουμε είναι η αύξηση των τιμών και των τριών σταθερών σφάλματος, σε περίπτωση που δεν είναι δυνατή η ταυτόχρονη αύξηση της σταθερά σφάλματος ταχύτητας (K_v) και της σταθερά σφάλματος επιτάχυνσης (K_α) προτεραιότητα θα δίνουμε στη σταθερά σφάλματος ταχύτητας (K_v).

6.7 Ασκήσεις Πράξης Χρονικής απόκρισης Συστημάτων Αυτόματου Ελέγχου

Άσκηση 1: Ο μετασχηματισμός Laplace της βηματικής απόκρισης ενός συστήματος είναι $Y(s) = \frac{1}{s*(s+1)}$, να βρεθούν η συνάρτηση μεταφοράς και η κρουστική απόκριση του συστήματος.

Αφού η $Y(s)$ είναι ο μετασχηματισμός Laplace της βηματικής απόκρισης και ο μετασχηματισμός Laplace της βηματικής εισόδου είναι $X(s) = \frac{1}{s}$ ισχύει:

$$Y(s) = G(s) * X(s) \leftrightarrow \frac{1}{s * (s + 1)} = G(s) * \frac{1}{s} \leftrightarrow G(s) = \frac{1}{s + 1}$$

Οπότε η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{1}{s+1}$

Γνωρίζουμε ότι η κρουστική απόκριση ενός συστήματος είναι ο αντίστροφος μετασχηματισμός Laplace της συνάρτησης μεταφοράς, άρα:

$$y_{(t)\text{κρουστικής}} = \mathcal{L}^{-1}\{G(s)\} = \mathcal{L}^{-1}\left\{\frac{1}{s+1}\right\} = e^{-t}$$

Άσκηση 2: Ο μετασχηματισμός Laplace της κρουστικής απόκρισης ενός συστήματος είναι $Y(s) = \frac{1}{s*(s+1)}$, να βρεθούν η συνάρτηση μεταφοράς και η βηματική απόκριση του συστήματος.

Αφού η $Y(s)$ είναι ο μετασχηματισμός Laplace της κρουστικής απόκρισης και ο μετασχηματισμός Laplace της βηματικής εισόδου είναι $X(s) = 1$ ισχύει:

$$Y(s) = G(s) * X(s) \leftrightarrow \frac{1}{s * (s + 1)} = G(s) * 1 \leftrightarrow G(s) = \frac{1}{s * (s + 1)}$$

Οπότε η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{1}{s*(s+1)}$

Η βηματική απόκριση του συστήματος είναι:

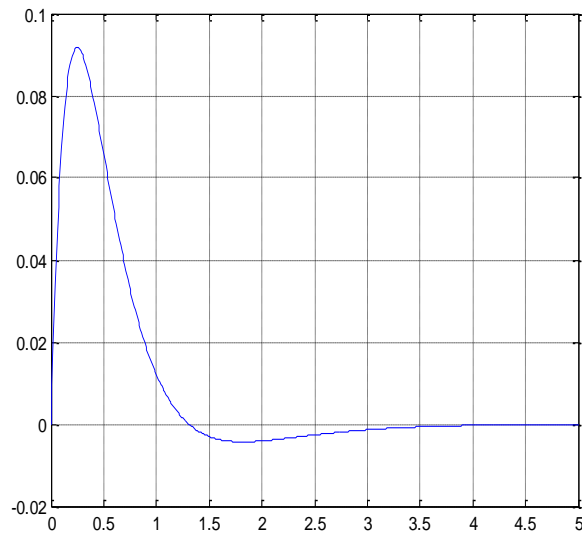
$$G(s) = \frac{Y(s)}{X(s)} \leftrightarrow \frac{1}{s * (s + 1)} = \frac{Y(s)}{\frac{1}{s}} \leftrightarrow Y(s) = \frac{1}{s * (s + 1)} * \frac{1}{s} \leftrightarrow Y(s) = \frac{1}{s^2 * (s + 1)} \leftrightarrow$$

$$\leftrightarrow y_{(t)\beta\eta\mu\alpha\tau\iota\kappa\eta\varsigma} = \mathcal{L}^{-1}\{Y(s)\} = \mathcal{L}^{-1}\left\{\frac{1}{s^2 * (s + 1)}\right\} = t * e^{-t}$$

Άσκηση 3: Ο μετασχηματισμός Laplace της κρουστικής απόκρισης ενός συστήματος είναι $Y(s) = \frac{s+1}{(s+5)^2*(s+5)}$, να σχεδιαστούν η κρουστική και η βηματική απόκριση του συστήματος για τον ίδιο χρονικό ορίζοντα.

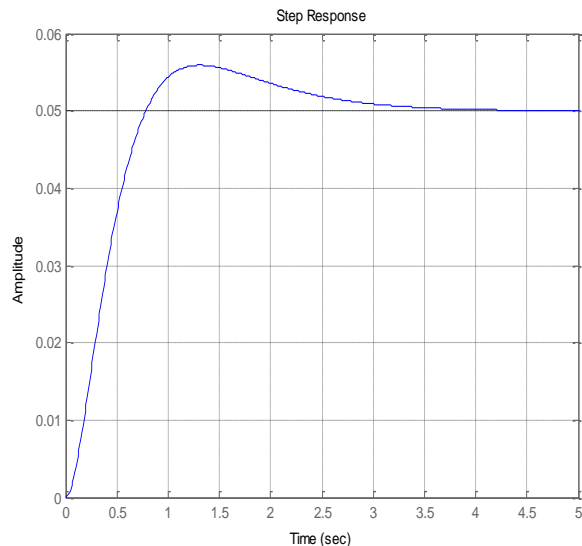
Η $Y(s)$ αφού είναι ο μετασχηματισμός Laplace της κρουστικής απόκρισης του συστήματος είναι ίση με τη συνάρτηση μεταφοράς του συστήματος. Αφού γνωρίζουμε τη συνάρτηση μεταφοράς θα χρησιμοποιήσουμε το m-file alapla.m και με τις παρακάτω εντολές σχεδιάζουμε την κρουστική απόκριση του συστήματος.

```
syms s
F=(s+1)/((s+2)^2*(s+5));
T=0:.01:5;
y=alapla(F,T);
plot(T,y);grid on;
```



Η βηματική απόκριση σχεδιάζεται με την εξής εντολή στην οποία ορίζουμε και το χρονικό ορίζοντα που επιθυμούμε να έχει το διάγραμμα.

```
syms s
F=(s+1)/((s+2)^2*(s+5));
[n d]=numden(F);
nd=sym2poly(n);
dd=sym2poly(d);
G=tf(nd,dd);
step(G,T);grid on;
```



Άσκηση 4: Θεωρήστε ότι η $Y(s) = \frac{s+1}{s*(s+2)*(s+3)^2}$ είναι η μετασχηματισμένη κατά Laplace της βηματικής απόκρισης ενός συστήματος. Σχεδιάστε τη βηματική απόκριση του συστήματος, υπολογίστε τον αντίστροφο Laplace της συνάρτησης μεταφοράς και σχεδιάστε την κρουστική του απόκριση για τον ίδιο χρονικό ορίζοντα με τον αντίστροφο Laplace.

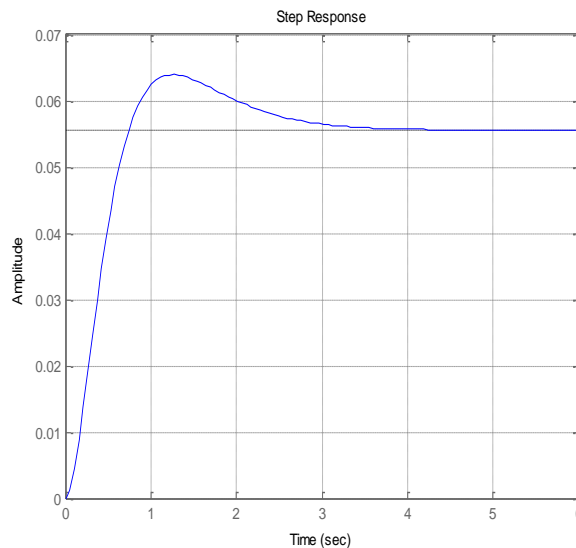
Αρχικά θα βρούμε τη συνάρτηση μεταφοράς του συστήματος.

$$Y(s) = G(s) * X(s) \leftrightarrow \frac{s+1}{s * (s+2) * (s+3)^2} = G(s) * \frac{1}{s} \leftrightarrow G(s) = \frac{s * (s+1)}{s * (s+2) * (s+3)^2}$$

$$\leftrightarrow G(s) = \frac{(s+1)}{(s+2) * (s+3)^2}$$

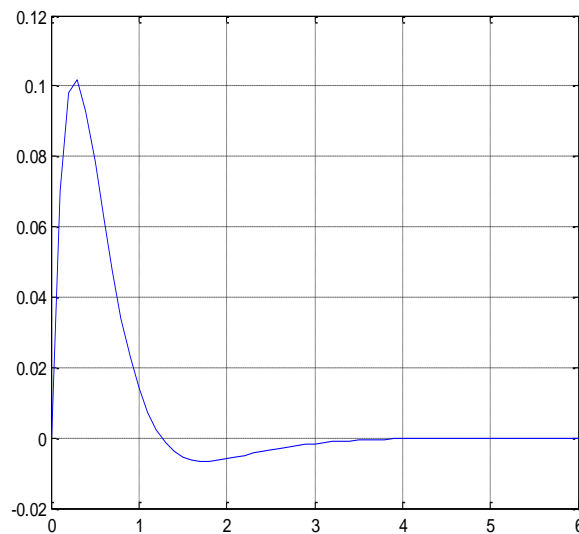
Άρα η βηματική απόκριση δίνεται με τις εντολές:

```
syms s
G=(s+1)/((s+2)*(s+3)^2);
[n d]=numden(G);
nd=sym2poly(n);
dd=sym2poly(d);
G=tf(nd,dd);
step(G);grid on;
```



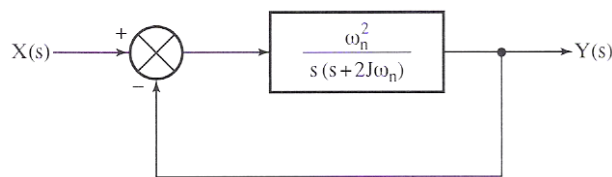
Ο αντίστροφος Laplace της συνάρτησης μεταφοράς που είναι η κρουστική απόκριση υπολογίζεται και σχεδιάζεται με τις παρακάτω εντολές:

```
syms s
G=(s+1)/((s+2)*(s+3)^2);
T=0:0.1:6;
K=alapa(G,T);
plot(T,K);grid on;
```



Υπενθυμίζουμε ότι σε περίπτωση που γνωρίζουμε τη συνάρτηση μεταφοράς του συστήματος μπορούμε να χρησιμοποιήσουμε την εντολή *impulse* για να πάρουμε την κρουστική απόκριση του συστήματος.

Άσκηση 5: Για το παρακάτω σύστημα με είσοδο μοναδιαία βηματική να υπολογιστούν τα ω_d , t_r , t_p , M_p και t_s αν $\zeta = 0.6$ και $\omega_n = 5 \text{ rad/sec}$.



Το σύστημα του δομικού διαγράμματος έχει συνάρτηση μεταφοράς:

$$G(s) = \frac{\frac{\omega_n^2}{s(s+2\zeta\omega_n)}}{1 + \frac{\omega_n}{s(s+2\zeta\omega_n)}} = \frac{\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s}}{\frac{s^2 + 2\zeta\omega_n s + \omega_n}{s^2 + 2\zeta\omega_n s}} \leftrightarrow$$

$$\leftrightarrow G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Το σύστημα είναι 2^{ης} τάξης με $0 < \zeta < 1$ οπότε:

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} = 5 * \sqrt{1 - 0.6^2} = 5 * \sqrt{0.64} \leftrightarrow \omega_d = 4 \text{ rad/sec}$$

$$t_r = \frac{\pi - \tan^{-1}\left(\frac{\omega_d}{\zeta * \omega_n}\right)}{\omega_d} = \frac{\pi - \tan^{-1}\left(\frac{4}{0.6 * 5}\right)}{4} = \frac{\pi - 0.9261}{4} \leftrightarrow t_r = 0.5539 \text{ sec}$$

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{4} \leftrightarrow t_p = 0.7854 \text{ sec}$$

$$M_p = e^{-\frac{\pi * \zeta}{\sqrt{1 - \zeta^2}}} = e^{-\frac{\pi * 0.6}{\sqrt{1 - 0.6^2}}} = e^{-\frac{\pi * 0.6}{0.8}} = e^{-2.3562} \leftrightarrow M_p = 0.0948$$

$$\leftrightarrow M_p = 9.48\%$$

$$t_s = \frac{4}{\zeta * \omega_n} = \frac{4}{0.6 * 5} = \frac{4}{3} \leftrightarrow t_s = 1.33 \text{ sec}$$

Άσκηση 6: Η συνάρτηση μεταφοράς ενός συστήματος είναι η εξής:

$$G(s) = \frac{\kappa}{s^2 + 10 * s + \kappa}$$

Όπου κ παράμετρος ενίσχυσης του συστήματος το οποίο διεγείρεται από είσοδο της μορφής μοναδιαίας βηματικής. Για $\kappa = 10, 100$ και 1000 να βρεθούν:

- i) Η κυκλική ιδιοσυχνότητα του συστήματος ω_n
- ii) Ο συντελεστής απόσβεσης ζ
- iii) Η φυσική ιδιοσυχνότητα του συστήματος ω_d
- iv) Οι πόλοι του συστήματος
- v) Το ποσοστό υπερύψωσης M_p
- vi) Η χρονική απόκριση του συστήματος.

Το σύστημα είναι 2^{ης} τάξης με βηματική είσοδο άρα έχουμε:

- i) $\omega_n = \sqrt{\kappa} \leftrightarrow \omega_n = \begin{cases} 3.16 \text{ rad/sec για } \kappa = 10 \\ 10 \text{ rad/sec για } \kappa = 100 \\ 31.6 \text{ rad/sec για } \kappa = 1000 \end{cases}$
- ii) $2 * \zeta * \omega_n = 10 \leftrightarrow \zeta = \frac{10}{2 * \omega_n} \leftrightarrow \zeta = \begin{cases} 1.58 \text{ για } \kappa = 10 \\ 0.5 \text{ για } \kappa = 100 \\ 0.158 \text{ για } \kappa = 1000 \end{cases}$
- iii) $\omega_d = \omega_n * \sqrt{1 - \zeta^2} \leftrightarrow \omega_d = \begin{cases} \text{δεν υπάρχει γιατί για } \kappa = 10, \zeta > 1 \\ 8.66 \text{ rad/sec για } \kappa = 100 \\ 31.2 \text{ rad/sec για } \kappa = 1000 \end{cases}$
- iv) $s_{1,2} = -\zeta * \omega_n \pm \omega_n * \sqrt{\zeta^2 - 1} \leftrightarrow$
 $\leftrightarrow s_{1,2} = -1.58 * 3.16 \pm 3.16 * \sqrt{1.58^2 - 1} \leftrightarrow$
 $\leftrightarrow s_1 = -1.272 \text{ και } s_2 = -8.8584 \text{ για } \kappa = 10.$

$$s_{1,2} = -\zeta * \omega_n \pm j * \omega_n * \sqrt{1 - \zeta^2} \leftrightarrow$$

$$\leftrightarrow s_{1,2} = \begin{cases} s_1 = -5 + j * 8.66 \text{ και } s_2 = -5 - j * 8.66 \text{ για } \kappa = 100 \\ s_1 = -5 + j * 31.2 \text{ και } s_2 = -5 - j * 31.2 \text{ για } \kappa = 1000 \end{cases}$$

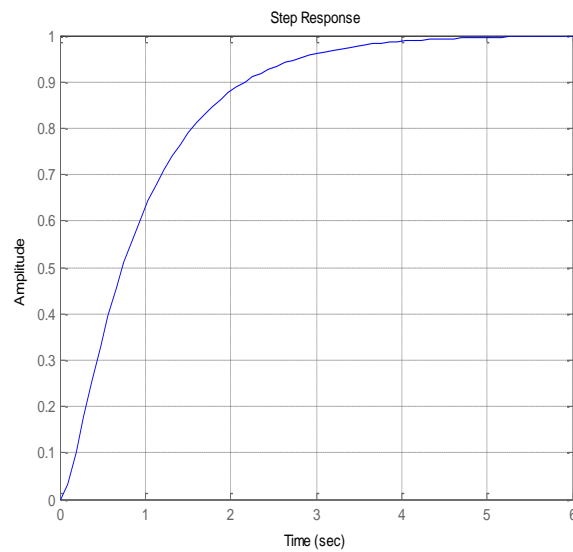
- v) $M_p \% = e^{-\pi * \zeta / \sqrt{1 - \zeta^2}} * 100 \leftrightarrow \begin{cases} \text{δεν υπάρχει γιατί για } \kappa = 10, \zeta > 1 \\ 16,3\% \text{ για } \kappa = 100 \\ 60\% \text{ για } \kappa = 1000 \end{cases}$

vi) Για $\kappa = 10$ η συνάρτηση μεταφοράς του συστήματος είναι

$$G(s) = \frac{10}{s^2 + 10 * s + 10}, \text{ με χρήση των παρακάτω εντολών η α-}$$

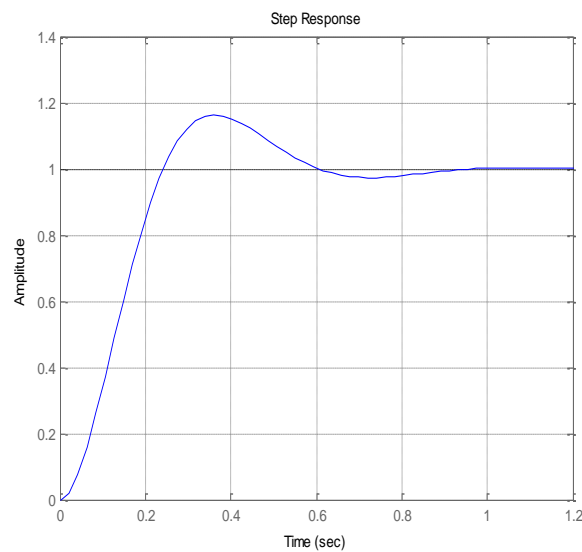
πόκριση του συστήματος είναι:

```
G=tf([10],[1 10 10]);
step(G);grid on;
```



Για $\kappa = 100$ η συνάρτηση μεταφοράς του συστήματος είναι $G(s) = \frac{100}{s^2+10*s+100}$, με χρήση των παρακάτω εντολών η απόκριση του συστήματος είναι:

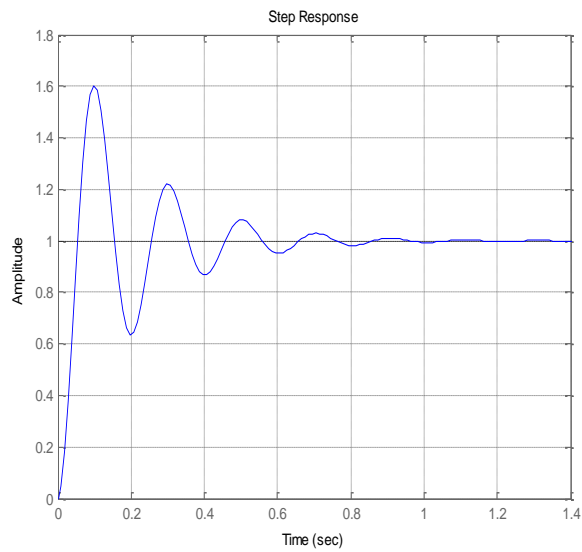
```
G=tf([100],[1 10 100]);
step(G);grid on;
```



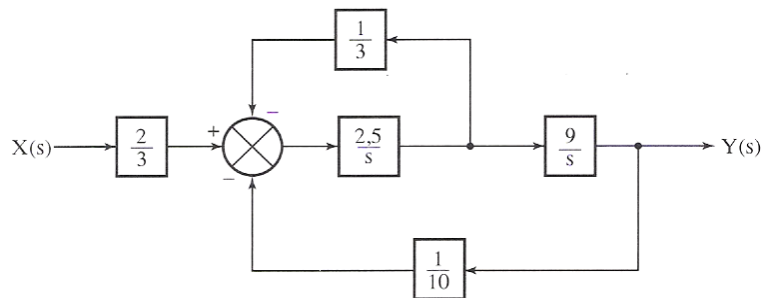
Για $\kappa = 1000$ η συνάρτηση μεταφοράς του συστήματος είναι

$G(s) = \frac{1000}{s^2+10*s+1000}$, με χρήση των παρακάτω εντολών η απόκριση του συστήματος είναι:

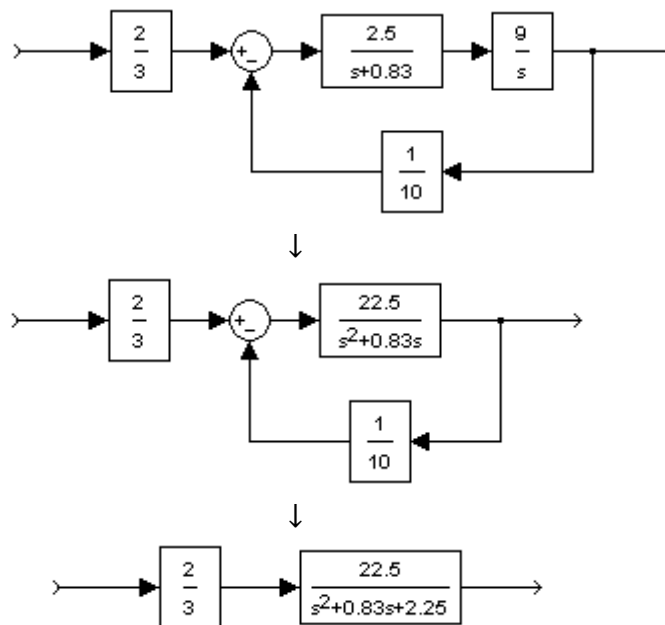
```
G=tf([1000],[1 10 1000]);
step(G);grid on;
```



Άσκηση 7: Για το παρακάτω δομικό διάγραμμα να βρεθούν τα ω_n , ζ , ω_d , και να σχεδιαστεί η βηματική απόκριση του συστήματος.



Αρχικά θα απλοποιήσουμε το σύστημα.





Από τη συνάρτηση μεταφοράς του συστήματος συμπεραίνουμε ότι είναι σύστημα 2^{ης} τάξης με:

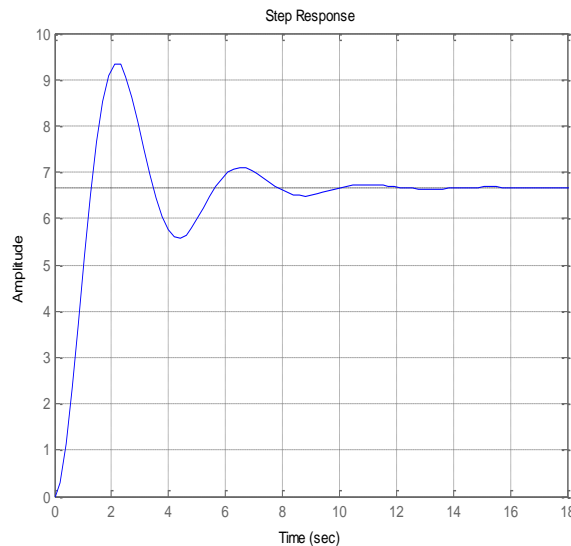
$$\omega_n = \sqrt{2.25} \leftrightarrow \omega_n = 1.5 \text{ rad/sec}$$

$$2 * \zeta * \omega_n = 0.83 \leftrightarrow \zeta = \frac{0.83}{2 * 1.5} \leftrightarrow \zeta = 0.276$$

$$\omega_d = \omega_n * \sqrt{1 - \zeta^2} = 1.5 * \sqrt{1 - 0.076} \leftrightarrow \omega_d = 1.438 \text{ rad/sec}$$

Η βηματική απόκριση του συστήματος σχεδιάζεται με τις παρακάτω εντολές:
`G=tf([15],[1 0.83 2.25]);`

`step(G);grid on;`



Άσκηση 8: Η συνάρτηση μεταφοράς ανοιχτού βρόγχου συστήματος με μοναδιαία ανάδραση δίνεται από τη σχέση:

$$G(s) = \frac{\kappa}{s * (T * s + 1)}$$

- i) Με ποιον παράγοντα πρέπει να πολλαπλασιαστεί η σταθερά ενίσχυσης κ ώστε ο συντελεστής απόσβεσης να αυξηθεί από το 0.2 στο 0.8
- ii) Με ποιο παράγοντα πρέπει να πολλαπλασιαστεί η σταθερά χρόνου T ώστε ο συντελεστής απόσβεσης να μειωθεί από 0.6 σε 0.3
- iii) Να αποδείξετε ότι $\frac{4 * T \kappa_1 - 1}{4 * T \kappa_2 - 1} \approx 10$ όπου κ_1 και κ_2 οι τιμές του κ για ποσοστό υπερύψωσης 60% και 20%.

Η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου με μοναδιαία ανάδραση είναι:

$$F_{(s)} = \frac{G_{(s)}}{1 + G_{(s)}} = \frac{\frac{\kappa}{s * (T * s + 1)}}{1 + \frac{\kappa}{s * (T * s + 1)}} = \frac{\kappa}{T * s^2 + s + \kappa} \leftrightarrow F_{(s)} = \frac{\frac{\kappa}{T}}{s^2 + \frac{s}{T} + \kappa}$$

Το σύστημα είναι 2^{ης} τάξης με $\omega_n = \sqrt{\frac{\kappa}{T}}$ και $\zeta = \frac{1}{2 * T * \omega_n}$.

$$i) \begin{cases} \zeta_1 = \frac{1}{2 * T * \omega_n} \\ \zeta_2 = \frac{1}{2 * T * \omega_n} \end{cases} \leftrightarrow \begin{cases} \zeta_1 = \frac{1}{2 * T * \sqrt{\frac{\kappa_1}{T}}} \\ \zeta_2 = \frac{1}{2 * T * \sqrt{\frac{\kappa_2}{T}}} \end{cases} \leftrightarrow \frac{\zeta_1}{\zeta_2} = \frac{\sqrt{\kappa_2}}{\sqrt{\kappa_1}} \leftrightarrow \frac{\sqrt{\kappa_2}}{\sqrt{\kappa_1}} = \frac{0.2}{0.8} = \frac{1}{4} \leftrightarrow$$

$$\leftrightarrow \frac{\kappa_2}{\kappa_1} = \frac{1}{16} \leftrightarrow \kappa_2 = 16 * \kappa_1$$

$$ii) \begin{cases} \zeta_1 = \frac{1}{2 * T_1 * \omega_n} \\ \zeta_2 = \frac{1}{2 * T_2 * \omega_n} \end{cases} \leftrightarrow \begin{cases} \zeta_1 = \frac{1}{2 * T_1 * \sqrt{\frac{\kappa}{T_1}}} \\ \zeta_2 = \frac{1}{2 * T_2 * \sqrt{\frac{\kappa}{T_2}}} \end{cases} \leftrightarrow \frac{\zeta_1}{\zeta_2} = \frac{\sqrt{T_2}}{\sqrt{T_1}} \leftrightarrow \frac{\sqrt{T_2}}{\sqrt{T_1}} = \frac{0.6}{0.3} = 2 \leftrightarrow$$

$$\leftrightarrow \frac{T_2}{T_1} = 4 \leftrightarrow T_2 = 4 * T_1$$

$$iii) \begin{cases} M_{p_1} \% = e^{-\pi * \zeta_1 / \sqrt{1 - \zeta_1^2}} * 100 \\ M_{p_2} \% = e^{-\pi * \zeta_2 / \sqrt{1 - \zeta_2^2}} * 100 \end{cases} \leftrightarrow \begin{cases} 60 = e^{-\pi * \zeta_1 / \sqrt{1 - \zeta_1^2}} * 100 \\ 20 = e^{-\pi * \zeta_2 / \sqrt{1 - \zeta_2^2}} * 100 \end{cases} \leftrightarrow$$

$$\leftrightarrow \begin{cases} 60 = e^{-\pi * \zeta_1 / \sqrt{1 - \zeta_1^2}} * 100 \\ 20 = e^{-\pi * \zeta_1 / \sqrt{1 - \zeta_2^2}} * 100 \end{cases} \leftrightarrow \begin{cases} -0.51 = -\pi * \zeta_1 / \sqrt{1 - \zeta_1^2} \\ -1.61 = -\pi * \zeta_2 / \sqrt{1 - \zeta_2^2} \end{cases} \leftrightarrow$$

$$\leftrightarrow \begin{cases} 0.51 = \zeta_1 / \sqrt{1 - \zeta_1^2} \\ 1.61 = \zeta_2 / \sqrt{1 - \zeta_2^2} \end{cases} \leftrightarrow \frac{\zeta_2 * \sqrt{1 - \zeta_1^2}}{\zeta_1 * \sqrt{1 - \zeta_2^2}} = 3.156 \leftrightarrow$$

$$\leftrightarrow \frac{\sqrt{\kappa_1} * \sqrt{1 - 4 * \kappa_1 * T}}{\sqrt{\kappa_2} * \sqrt{1 - 4 * \kappa_2 * T}} = 3.156 \leftrightarrow \frac{4 * \kappa_1 * T - 1}{4 * \kappa_2 * T - 1} \approx 9.96$$

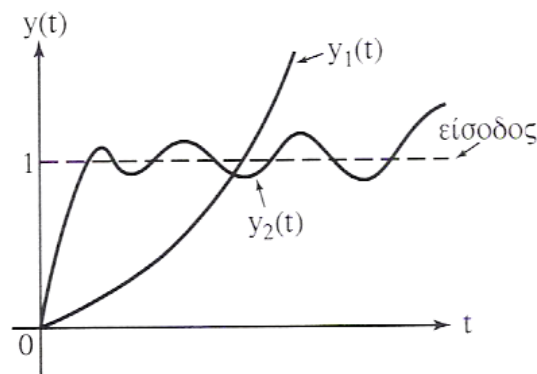
7. Ευστάθεια Συστημάτων Αυτόματου Ελέγχου

Ένα από τα χαρακτηριστικά των συστημάτων αυτόματου ελέγχου που διαδραματίζουν το βασικότερο ρόλο στη συμπεριφορά τους είναι η ευστάθεια, γι' αυτό κατά τη σχεδίαση ενός συστήματος αυτόματου ελέγχου επιδιώκεται πρώτα και πάνω από όλα η εξασφάλιση της ευστάθειας. Μετά την εξασφάλιση της ευστάθειας του συστήματος επιδιώκεται η ικανοποίηση άλλων απαιτήσεων σχεδίασης όπως η ταχύτητα, η ακρίβεια απόκρισης, το εύρος ζώνης, το σφάλμα στη μόνιμη κατάσταση κ.λπ.

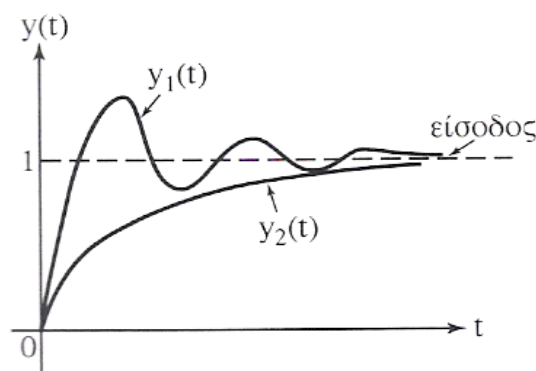
Ένα σύστημα είναι ευσταθές αν για πεπερασμένη είσοδο παρουσιάζει πεπερασμένη απόκριση. Αντιθέτως ένα σύστημα ονομάζεται ασταθές αν έστω και για μια πεπερασμένη είσοδο δεν παρουσιάζει πεπερασμένη απόκριση.

Ένα ασταθές σύστημα κλειστού βρόγχου δεν παρουσιάζει πρακτικά ενδιαφέρον γιατί η έξοδος ενός ασταθούς συστήματος αυξάνει θεωρητικά προς το άπειρο σε αντίθεση με ένα ευσταθές σύστημα κλειστού βρόγχου όπου η έξοδος του βρίσκεται μέσα σε επιτρεπτά όρια. Για το λόγο αυτό αναζητούμε μεθόδους οι οποίες μας βοηθούν να αναλύουμε και να σχεδιάζουμε ευσταθή συστήματα.

Η απόκριση ενός ασταθούς συστήματος είναι της μορφής:



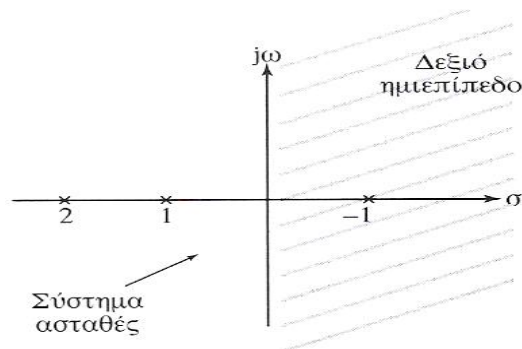
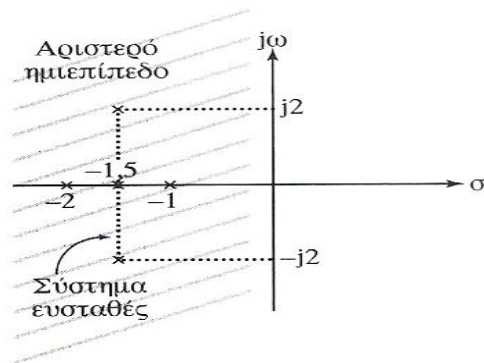
Η απόκριση ενός ευσταθούς συστήματος είναι της μορφής:

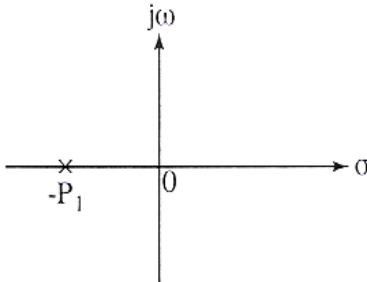
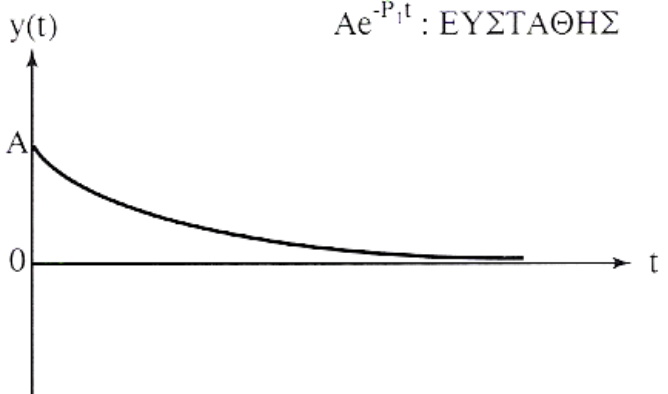
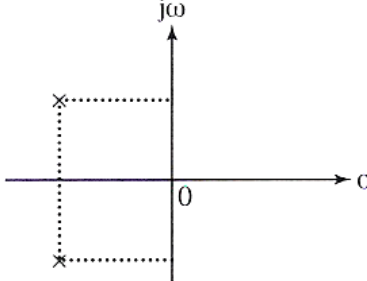
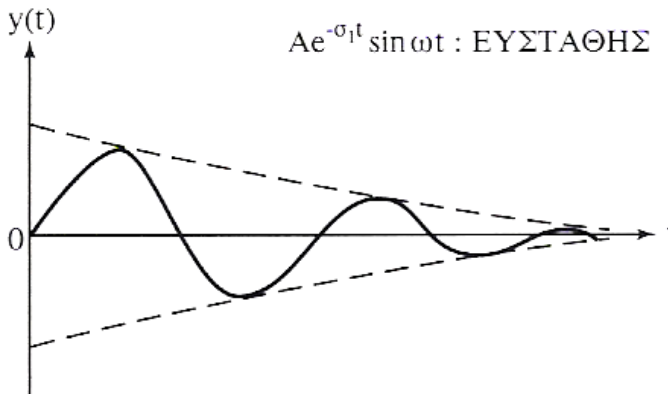
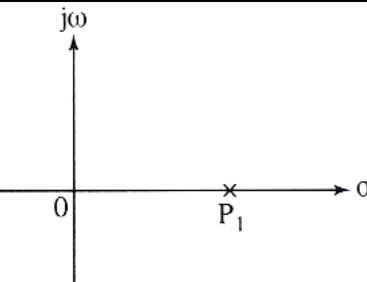
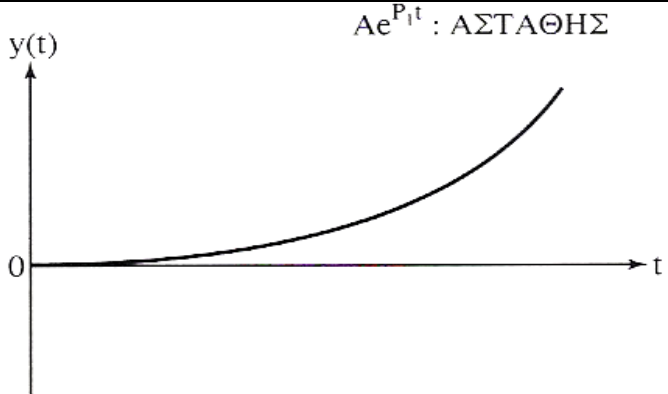
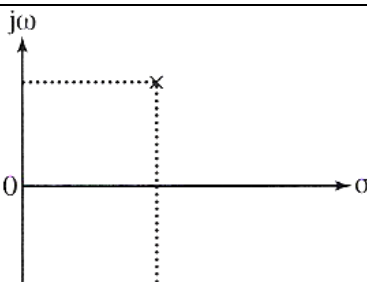
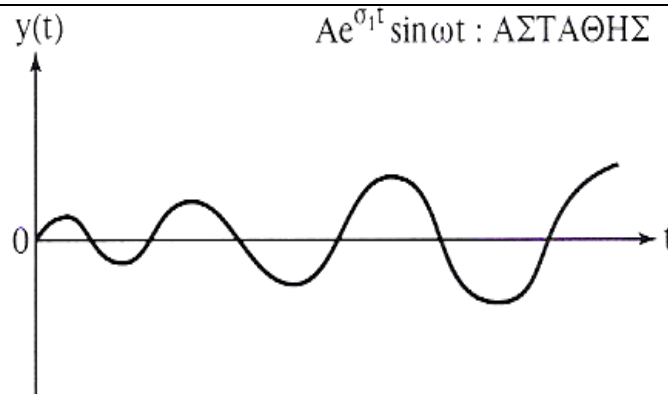


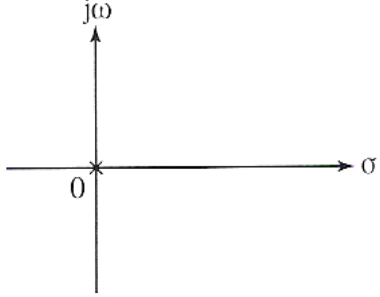
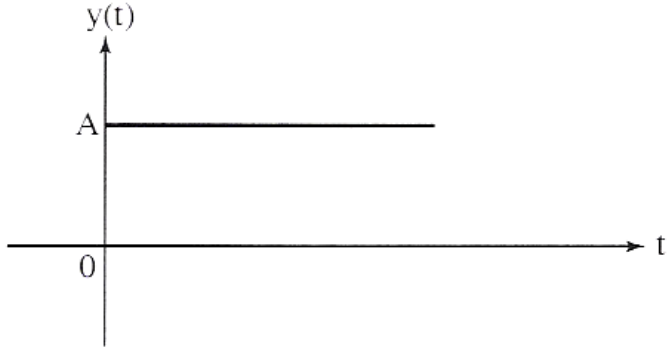
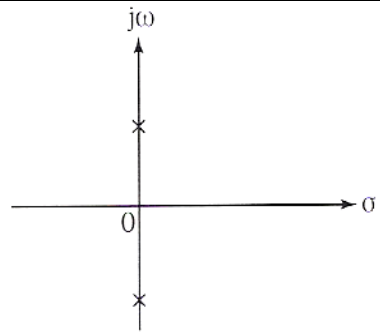
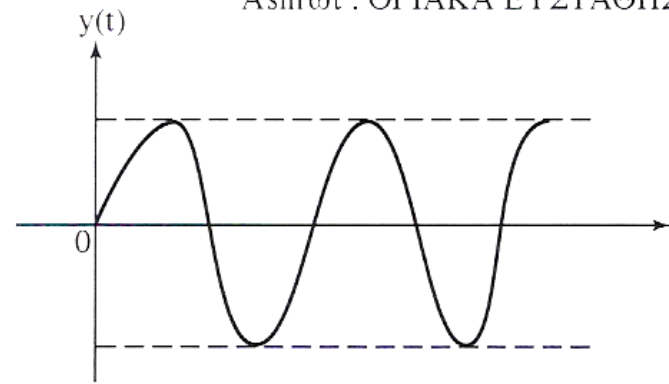
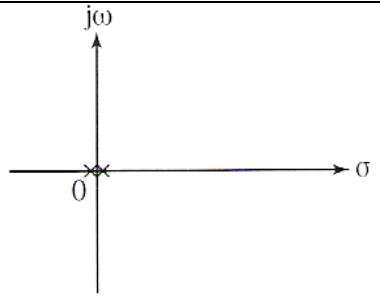
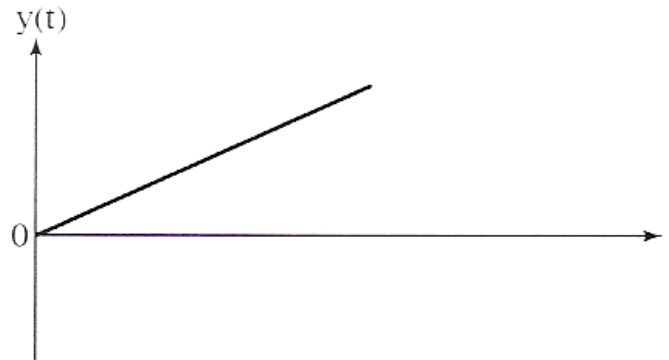
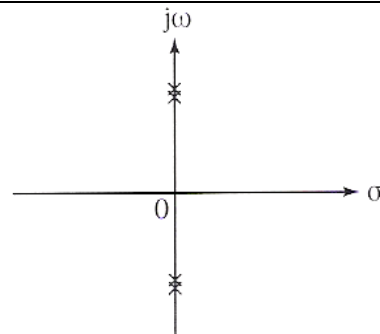
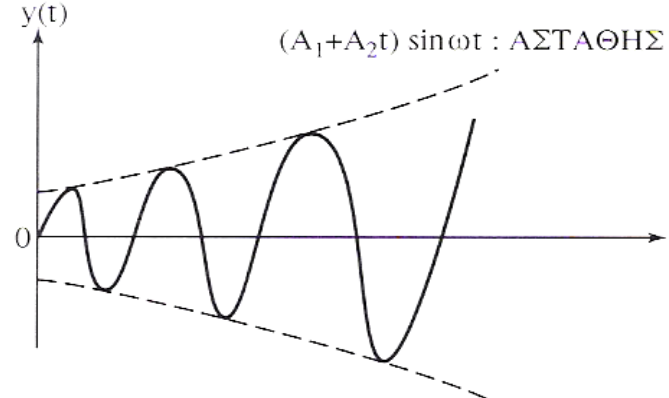
Για ένα σύστημα κλειστού βρόχου με συνάρτηση μεταφοράς ανοιχτού βρόγχου $G(s)$ και συνάρτηση μεταφοράς της ανάδρασης $H(s)$ τότε το πολυώνυμο του παρανομαστή της συνάρτησης μεταφοράς του συστήματος $\frac{G(s)}{1+G(s)*H(s)}$, το $1 + G(s) * H(s)$ ονομάζεται χαρακτηριστικό πολυώνυμο ενώ η εξίσωση $1 + G(s) * H(s) = 0$ ονομάζεται χαρακτηριστική εξίσωση της συνάρτησης μεταφοράς κλειστού βρόγχου.

Για την κατηγορία των γραμμικών μη χρονικά μεταβαλλόμενων συστημάτων που θα μελετήσουμε η ευστάθεια τους συνδέεται με τη θέση των ριζών της χαρακτηριστικής εξίσωσης στο μιγαδικό επίπεδο. Σε αυτή τη περίπτωση ένα σύστημα είναι ευσταθές αν όλες οι ρίζες του χαρακτηριστικού πολυωνύμου βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο. Αν έστω και μια ρίζα του χαρακτηριστικού πολυωνύμου βρίσκεται στο δεξιό μιγαδικό ημιεπίπεδο το σύστημα είναι ασταθές.

Ένα σύστημα είναι ευσταθές όταν όλοι οι πόλοι του βρίσκονται στο αριστερό μιγαδικό ημιεπίπεδο. Στα παρακάτω διαγράμματα παρουσιάζεται η σχέση των πόλων και της ευστάθειας του συστήματος.



Χρονικές αποκρίσεις για διάφορες θέσεις πόλων στο μιγαδικό επίπεδο	
Θέση πόλων	Χρονική απόκριση
 <p>Αρνητικός πραγματικός πόλος</p>	$y(t) = Ae^{-P_1 t} : \text{ΕΥΣΤΑΘΗΣ}$ 
 <p>Μιγαδικοί συζυγείς πόλοι με αρνητικό πραγματικό μέρος</p>	$y(t) = Ae^{-\sigma t} \sin \omega t : \text{ΕΥΣΤΑΘΗΣ}$ 
 <p>Θετικός πραγματικός πόλος</p>	$y(t) = Ae^{P_1 t} : \text{ΑΣΤΑΘΗΣ}$ 
 <p>Μιγαδικοί συζυγείς πόλοι με θετικό πραγματικό μέρος</p>	$y(t) = Ae^{\sigma t} \sin \omega t : \text{ΑΣΤΑΘΗΣ}$ 

 <p>Απλός πόλος στο μηδέν</p>	<p>A : ΟΡΙΑΚΑ ΕΥΣΤΑΘΗΣ</p> 
 <p>Συζυγείς φανταστικοί πόλοι</p>	<p>$A \sin \omega t$: ΟΡΙΑΚΑ ΕΥΣΤΑΘΗΣ</p> 
 <p>Διπλός πόλος στο μηδέν</p>	<p>$A t$: ΑΣΤΑΘΗΣ</p> 
 <p>Διπλοί συζυγείς φανταστικοί πόλοι</p>	<p>$(A_1 + A_2 t) \sin \omega t$: ΑΣΤΑΘΗΣ</p> 

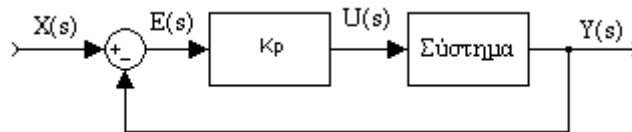
8. Ελεγκτές Συστημάτων Αυτόματου Ελέγχου

Στα συστήματα αυτόματου ελέγχου χρησιμοποιούμε συνδεσμολογίες ελεγκτών οι οποίοι αποσκοπούν στη βελτίωση των χαρακτηριστικών του συστήματος που μελετάμε. Με τη χρήση ελεγκτών μπορούμε να πετύχουμε καλύτερες επιδόσεις του συστήματος δηλαδή να γίνει το σύστημα πιο γρήγορο, πιο ακριβές, πιο ευσταθές, να μην επηρεάζεται από τυχόν διαταραχές κ.λπ.

Η μέθοδος με την οποία οι ελεγκτές πετυχαίνουν το στόχο τους είναι η εισαγωγή πόλων ή μηδενικών ή και των δύο στη συνάρτηση μεταφοράς του συστήματος ανοιχτού βρόγχου του οποίου τη συμπεριφορά θέλουμε να βελτιώσουμε. Υπάρχουν διάφορα είδη ελεγκτών τα οποία στην πραγματικότητα αποτελούν συνδυασμό των τριών βασικών ελεγκτών. Η πιο ευρύτητα χρησιμοποιημένη μονάδα ελεγκτή είναι ο ελεγκτής PID ο οποίος ονομάζεται και ελεγκτής τριών όρων γιατί στη ουσία αποτελεί συνδυασμό το τριών βασικών ελεγκτών. Ο ελεγκτής PID περιλαμβάνει έναν αναλογικό, ένα διαφορικό και ένα ολοκληρωτικό όρο. Υπάρχουν όμως και ελεγκτές που περιέχουν τους δύο από τους τρεις όρους.

8.1 Αναλογικός (P) ελεγκτής

Ο αναλογικός ελεγκτής (Proportional Control) δίνει στην έξοδο του σήμα ανάλογο του σφάλματος που δέχεται στην είσοδο και στην ουσία πρόκειται για μια διάταξη ενίσχυσης του σήματος σφάλματος με κέρδος K_p . Ο αναλογικός ελεγκτής έχει την μορφή που παρουσιάζεται στο παρακάτω σχήμα.



Είναι φανερό από το σχήμα ότι η συνάρτηση μεταφοράς του ελεγκτή είναι:

$$P(s) = K_p$$

Ο αναλογικός ελεγκτής αυξάνει την ταχύτητα απόκρισης του συστήματος αλλά πιθανόν να δημιουργεί σφάλμα μόνιμης κατάστασης αναλόγως τον τύπο του συστήματος, ενώ για μηδενικό σφάλμα όπως είναι λογικό θα έχει και μηδενική έξοδο.

Σε περίπτωση που το σύστημα είναι 1^{ης} τάξης με συνάρτηση μεταφοράς $G(s) = \frac{1}{s+A}$ τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου με P ελεγκτή είναι:

$$G_{c(s)} = \frac{\frac{K_p}{s+A}}{1 + \frac{K_p}{s+A}} \leftrightarrow G_{c(s)} = \frac{K_p}{s+A+K_p}$$

Επομένως αν διεγείραμε το σύστημα με σήμα εισόδου μοναδιαία βηματική θα βρίσκαμε το σφάλμα μόνιμης κατάστασης από τη σχέση:

$$e_{ss} = 1 - \lim_{s \rightarrow 0} \left(s * G_{c(s)} * \frac{1}{s} \right) = 1 - \lim_{s \rightarrow 0} \frac{K_p}{s+A+K_p} \leftrightarrow e_{ss} = 1 - \frac{K_p}{A+K_p}$$

Συμπεραίνουμε ότι το σφάλμα μόνιμης κατάστασης μειώνεται όσο αυξάνουμε το κέρδος του αναλογικού ελεγκτή, όμως στη πράξη δεν γίνεται να αυξήσουμε απεριόριστα το κέρδος.

Στη περίπτωση τώρα που το σύστημα είναι 2^{ης} τάξης με συνάρτηση μεταφοράς $G(s) = \frac{1}{s^2+s*A+B}$ τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου με P ελεγκτή είναι:

$$G_{c(s)} = \frac{\frac{K_p}{s^2+s*A+B}}{1 + \frac{K_p}{s^2+s*A+B}} \leftrightarrow G_{c(s)} = \frac{K_p}{s^2+s*A+B+K_p}$$

Ομοίως αν διεγείραμε το σύστημα με σήμα εισόδου μοναδιαία βηματική θα βρίσκαμε το σφάλμα μόνιμης κατάστασης από τη σχέση:

$$e_{ss} = 1 - \lim_{s \rightarrow 0} \left(s * G_{c(s)} * \frac{1}{s} \right) = 1 - \lim_{s \rightarrow 0} \frac{K_p}{s^2+s*A+B+K_p} \leftrightarrow e_{ss} = 1 - \frac{K_p}{B+K_p}$$

Επομένως και στο σύστημα 2^{ης} τάξης η χρήση P ελεγκτή μειώνει το σφάλμα όσο αυξάνεται το κέρδος του ελεγκτή και αυξάνεται και η φυσική συχνότητα $\omega_n (= \sqrt{B+K_p})$ ενώ χωρίς τον ελεγκτή $\omega_n = \sqrt{B}$ του συστήματος κάτι που προσδίδει στο σύστημα ταχύτερη απόκριση. Στο σύστημα με αναλογικό ελεγκτή το A ισούται με $A = 2 * \zeta * \omega_n$ ομοίως με το σύστημα χωρίς ελεγκτή και παραμένει σταθερό με τη διαφορά όμως ότι αυξάνεται η φυσική συχνότητα του συστήματος ενώ μειώνεται ο συντελεστής απόσβεσης ζ του συστήματος κάτι που έχει ως αποτέλεσμα την αύξηση της υπερύψωσης στη βηματική απόκριση του συστήματος.

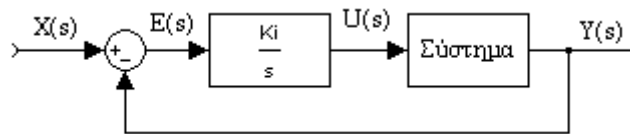
Ένας αναλογικός ελεγκτής αντιδρά άμεσα στην ύπαρξη σφάλματος επιτυγχάνει μείωση του σφάλματος μόνιμης κατάστασης όμως αδυνατεί να το εξαλείψει τελείως αφού για μέγιστη μείωση του σφάλματος χρειάζονται μη ρεαλιστικές υψηλές τιμές κέρδους, αυξάνει την ταχύτητα απόκρισης του συστήματος αλλά δημιουργεί σημαντική υπερύψωση και ταλαντώσεις με την αύξηση του κέρδους K_p .

Για να μπορούμε να χρησιμοποιούμε πιο εύκολα τον αναλογικό ελεγκτή σε διάφορες εφαρμογές ή ασκήσεις θα δημιουργήσουμε μια συνάρτηση η οποία θα μας επιστρέφει την απόκριση ενός υπό ελέγχου συστήματος. Για το λόγο αυτό δημιουργήσαμε το παρακάτω m-file.


```
function Gc=Pcontrol(H,A,Kp)%H:το σύστημα ανοιχτού βρόχου, A:η βαθμίδα ανάδρασης ή το αισθητήριο
P=tf(Kp,1);%Συνάρτηση μεταφοράς του P-ελεγκτή
Gc=feedback(series(P,H),A);%Συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου
```

8.2 Ολοκληρωτικός (I) ελεγκτής

Ο ολοκληρωτικός ελεγκτής (Integral Control) πήρε αυτή την ονομασία επειδή δίνει στην έξοδο του σήμα ανάλογο του ολοκληρώματος του σφάλματος που δέχεται στην είσοδο. Ο ολοκληρωτικός ελεγκτής περιέχει και μια παράμετρο την K_i η οποία έχει μονάδα μέτρησης το sec^{-1} και ονομάζεται συντελεστής ολοκλήρωσης. Ο ολοκληρωτικός ελεγκτής έχει την μορφή που παρουσιάζεται στο παρακάτω σχήμα.



Ο ολοκληρωτικός ελεγκτής εισάγει στο σύστημα ένα πόλο και είναι φανερό από το σχήμα ότι η συνάρτηση μεταφοράς του ελεγκτή είναι:

$$I(s) = \frac{K_i}{s}$$

Ο ολοκληρωτικός ελεγκτής για βηματική διέγερση απαιτεί κάποιο χρονικό διάστημα για να «πιάσει» την είσοδο δηλαδή η έξοδος του έχει μορφή αναρρίχησης και όταν το σφάλμα μηδενίζεται η έξοδος του ελεγκτή παραμένει σταθερή.

Σε περίπτωση που το σύστημα είναι 1^{ης} τάξης με συνάρτηση μεταφοράς $G(s) = \frac{1}{s+A}$ τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου με I ελεγκτή είναι:

$$G_{c(s)} = \frac{\frac{K_i}{s * (s + A)}}{1 + \frac{K_i}{s * (s + A)}} \leftrightarrow G_{c(s)} = \frac{K_i}{s * (s + A) + K_i} \leftrightarrow G_{c(s)} = \frac{K_i}{s^2 + A * s + K_i}$$

Επομένως αν διεγείραμε το σύστημα με σήμα εισόδου μοναδιαία βηματική θα βρίσκαμε το σφάλμα μόνιμης κατάστασης από τη σχέση:

$$e_{ss} = 1 - \lim_{s \rightarrow 0} \left(s * G_{c(s)} * \frac{1}{s} \right) = 1 - \lim_{s \rightarrow 0} \frac{K_i}{s^2 + A * s + K_i} \leftrightarrow e_{ss} = 1 - \frac{K_i}{K_i} \leftrightarrow e_{ss} = 0$$

Άρα το σφάλμα μόνιμης κατάστασης μηδενίζεται με τη χρήση ολοκληρωτικού ελεγκτή.

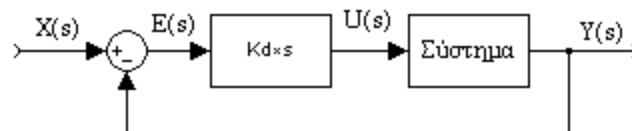
Παρατηρούμε ότι το σύστημα με τη χρήση του ολοκληρωτικού ελεγκτή από 1^{ης} τάξης μετατράπηκε σε 2^{ης} τάξης όπου η τιμή του συντελεστή ολοκλήρωσης είναι ίδια με της φυσικής συχνότητας του συστήματος. Όσο αυξάνεται ο συντελεστής ολοκλήρωσης αυξάνεται η φυσική συχνότητα του συστήματος άρα έχουμε ταχύτερη απόκριση του συστήματος και μειώνεται ο συντελεστής απόσβεσης, αφού το γινόμενο $2 * \zeta * \omega_n$ παραμένει σταθερό και

ισο με A , με συνέπεια την αύξηση της υπερύψωσης στη βηματική απόκριση του συστήματος.

Ένας ολοκληρωτικός ελεγκτής εξαλείφει το σφάλμα μόνιμης κατάστασης για σταθερή είσοδο αναφοράς και επιπλέον αντιμετωπίζει με επιτυχία αποκλίσεις από πιθανό διαταραχές του συστήματος.

8.3 Διαφορικός (D) ελεγκτής

Ο διαφορικός ελεγκτής (Differential Control) όταν το σήμα του σφάλματος στην είσοδο του έχει τη μορφή βηματικής διέγερσης τότε η έξοδος του ελεγκτή είναι η κρουστική συνάρτηση με θεωρητικά άπειρο πλάτος για $t = 0$. Όταν το σφάλμα είναι σταθερό τότε η έξοδος του διαφορικού ελεγκτή είναι μηδέν.



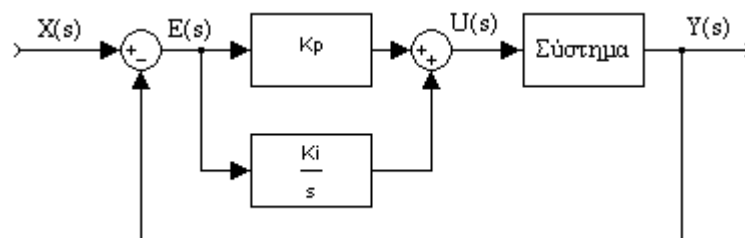
Ο διαφορικός ελεγκτής εισάγει στο σύστημα ένα μηδενικό και η συνάρτηση μεταφοράς του ελεγκτή είναι:

$$D(s) = K_d * s$$

Ο διαφορικός ελεγκτής περιορίζει το σφάλμα στη μόνιμη κατάσταση κατά το στάδιο της μεταβατικής απόκρισης των συστημάτων όμως στη πράξη δεν χρησιμοποιείται ποτέ αποκλειστικά διαφορικός ελεγκτής.

8.4 Αναλογικός – Ολοκληρωτικός (PI) ελεγκτής

Ο αναλογικός – ολοκληρωτικός ελεγκτής συνδυάζει αθροιστικά τον αναλογικό και τον ολοκληρωτικό έλεγχο. Η συνδεσμολογία του αναλογικού – ολοκληρωτικού ελέγχου είναι η συνδεσμολογία του παρακάτω σχήματος.



Ο αναλογικός – ολοκληρωτικός ελεγκτής εισάγει ένα πόλο στο μηδέν και ένα μηδενικό στο $-\frac{K_i}{K_p}$ συμπέρασμα που προκύπτει από τη συνάρτηση μεταφοράς του ελεγκτή.

$$PI(s) = K_p + \frac{K_i}{s} = \frac{s * K_p + K_i}{s} \leftrightarrow PI(s) = K_p * \frac{s + \frac{K_i}{K_p}}{s}$$

Η χρήση του όρου ολοκλήρωσης αποσκοπεί στην εξάλειψη του σφάλματος στη μόνιμη κατάσταση και μεταβάλλει το DC-κέρδος του συστήματος ενώ ο αναλογικός όρος αυξάνει τη ταχύτητα απόκρισης και βελτιώνει την ευστάθεια του συστήματος.

Αν το σύστημα που ελέγχουμε είναι πρώτης τάξης με συνάρτηση μεταφοράς $G(s) = \frac{1}{s+A}$ τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου με PI ελεγκτή είναι:

$$G_{c(s)} = \frac{\frac{s * K_p + K_i}{s} * \frac{1}{s + A}}{1 + \frac{s * K_p + K_i}{s} * \frac{1}{s + A}} \leftrightarrow G_{c(s)} = \frac{\frac{s * K_p + K_i}{s * (s + A)}}{\frac{s * (s + A) + s * K_p + K_i}{s * (s + A)}} \leftrightarrow$$

$$\leftrightarrow G_{c(s)} = \frac{s * K_p + K_i}{s^2 + A * s + s * K_p + K_i} \leftrightarrow G_{c(s)} = \frac{s * K_p + K_i}{s^2 + (A + K_p) * s + K_i}$$

Το σύστημα γίνεται 2^{ης} τάξης με:

$$\omega_n = K_i$$

και

$$2 * \zeta * \omega_n = A + K_p$$

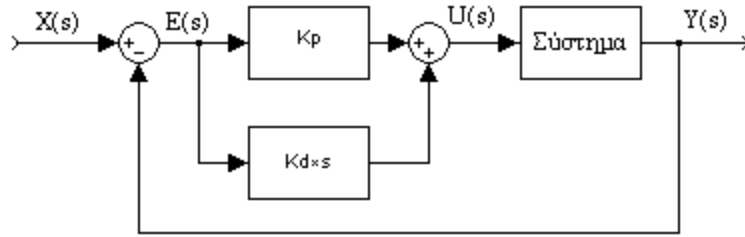
Άρα όταν χρησιμοποιούμε τον PI ελεγκτή με την αύξηση του ολοκληρωτικού όρου K_i αυξάνεται η φυσική συχνότητα ω_n του συστήματος και με την αύξηση του αναλογικού συντελεστή K_p αυξάνεται το γινόμενο $2 * \zeta * \omega_n$ αλλά χωρίς να μεταβάλλεται η ω_n οπότε αυξάνεται ο συντελεστής απόσβεσης του συστήματος με αποτέλεσμα τη μείωση της υπερύψωσης. Η ακριβής επίδραση του PI ελεγκτή στην μεταβατική απόκριση του συστήματος εξαρτάται από μηδενικό $s = -\frac{K_i}{K_p}$ που εισάγει ο ελεγκτής στο σύστημα. Τέλος με την εισαγωγή του όρου ολοκλήρωσης το σύστημα για βηματική είσοδο παρουσιάζει μηδενικό σφάλμα μόνιμης κατάστασης.

Το m-file του PI ελεγκτή είναι το εξής:

```
function Gc=PIcontrol(H,A,Kp,Ki)
PI=tf([Kp Ki],[1 0]);%Συνάρτηση μεταφοράς του PI-ελεγκτή
Gc=feedback(series(PI,H),A);
```

8.5 Αναλογικός – Διαφορικός (PD) ελεγκτής

Ο αναλογικός – διαφορικός ελεγκτής συνδυάζει αθροιστικά τον αναλογικό και το διαφορικό έλεγχο. Η συνδεσμολογία του αναλογικού – διαφορικού ελέγχου είναι η συνδεσμολογία του παρακάτω σχήματος.



Ο αναλογικός – διαφορικός εισάγει ένα πόλο στο $-\frac{K_p}{K_d}$ συμπέρασμα που προκύπτει από τη συνάρτηση μεταφοράς του ελεγκτή:

$$PD(s) = K_p + s * K_d \leftrightarrow PD(s) = s + \frac{K_p}{K_d}$$

Ο PD ελεγκτής αυξάνει την απόσβεση του συστήματος και περιορίζει την υπερύψωση όμως δεν μειώνει άμεσα το σφάλμα μόνιμης κατάστασης. Η χρήση του διαφορικού όρου προκαλεί την αυξημένη απόσβεση του συστήματος που επιτρέπει την αύξηση του αναλογικού κέρδους που έχει ως αποτέλεσμα τη βελτίωση του σφάλματος μόνιμης κατάστασης και την αύξηση της ταχύτητας απόκρισης του συστήματος. Στην πράξη ο PD έλεγχος εξαιτίας του διαφορικού όρου ενισχύει το θόρυβο ο οποίος μπορεί να εμφανιστεί στην απόκριση του συστήματος, επίσης είναι πιθανό να δημιουργηθούν προβλήματα στην περίπτωση που υπάρχουν απότομες αλλαγές στο σήμα αναφοράς γιατί έχουν ως συνέπεια ο διαφορικός όρος να παράγει έξοδο κρουστικής μορφής.

Αν το σύστημα που ελέγχουμε είναι 2^{ns} τάξης με συνάρτηση μεταφοράς $G(s) = \frac{1}{s^2 + A*s + B}$ τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόχου με PD ελεγκτή είναι:

$$\begin{aligned} G_{c(s)} &= \frac{(K_p + s * K_d) * \frac{1}{s^2 + A * s + B}}{1 + (K_p + s * K_d) * \frac{1}{s^2 + A * s + B}} \leftrightarrow \\ &\leftrightarrow G_{c(s)} = \frac{\frac{(K_p + s * K_d)}{s^2 + A * s + B}}{\frac{s^2 + A * s + B + K_p + s * K_d}{s^2 + A * s + B}} \leftrightarrow \\ &\leftrightarrow G_{c(s)} = \frac{K_p + s * K_d}{s^2 + (A + K_d) * s + (B + K_p)} \end{aligned}$$

Το νέο σύστημα 2^{ns} τάξης έχει τα εξής χαρακτηριστικά:

$$\omega_n = B + K_p$$

και

$$2 * \zeta * \omega_n = A + K_d$$

Άρα η αύξηση του αναλογικού κέρδους K_p οδηγεί σε αύξηση της φυσικής συχνότητας του συστήματος επομένως και σε ταχύτερη απόκριση του συστήματος. Η αύξηση του διαφορικού συντελεστή K_d οδηγεί σε αύξηση του γινομένου $2 * \zeta * \omega_n$ με αμετάβλητη τη φυσική συχνότητα άρα αυξάνεται ο

συντελεστής απόσβεσης του συστήματος που οδηγεί στη μείωση της υπερύψωσης της απόκρισης του συστήματος.

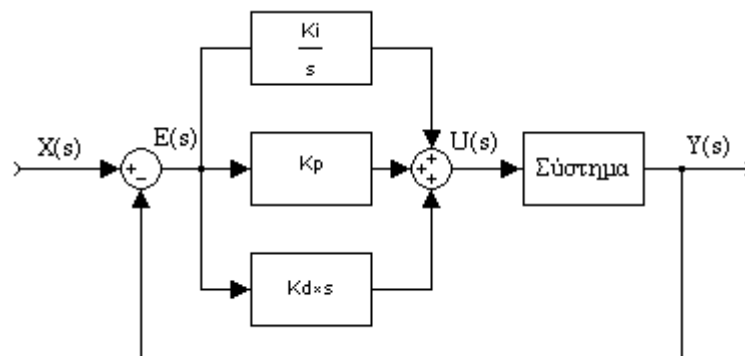
Το σφάλμα μόνιμης κατάστασης μειώνεται όσο αυξάνουμε το αναλογικό κέρδος ενώ δεν επηρεάζεται από τη μεταβολή του διαφορικού συντελεστή.

Το m-file του PD ελεγκτή είναι το εξής:

```
function Gc=PDcontrol(H,A,Kp,Kd)
PD=tf([Kd Kp],1);%Συνάρτηση μεταφοράς του PD-ελεγκτή
Gc=feedback(series(PD,H),A);
```

8.6 Αναλογικός – Ολοκληρωτικός – Διαφορικός (PID) ελεγκτής

Η δημοτικότητα του ελεγκτή PID οφείλεται κατά ένα μέρος στην πολύ καλή συμπεριφορά του έναντι μιας αρκετά ευρείας κλίμακας λειτουργικών συνθηκών και κατά ένα άλλο μέρος στη σχετική του απλότητα η οποία επιτρέπει έναν εύκολο και αξιόπιστο χειρισμό. Ο PID έλεγχος συνδυάζει αθροιστικά την επίδραση των τριών βασικών όρων ελέγχου. Η συνδεσμολογία του PID ελέγχου παρουσιάζεται στο παρακάτω σχήμα.



Ο PID ελεγκτής εισάγει στο υπό έλεγχο σύστημα ανοιχτού βρόχου ένα πόλο στο μηδέν και δύο μηδενικά. Η συνάρτηση μεταφοράς του ελεγκτή είναι η εξής:

$$PID_{(s)} = K_p + s * K_d + \frac{K_i}{s} \leftrightarrow PID_{(s)} = \frac{K_d * s^2 + K_p * s + K_i}{s}$$

Πολλές διεργασίες βιομηχανικού τύπου ελέγχονται με την βοήθεια μονάδων ελεγκτών PID. Ο PID ελεγκτής έχει την καλύτερη συνολικά απόδοση καθώς το σύστημα εμφανίζει ταχεία απόκριση, μηδενική υπερύψωση και μηδενικό σφάλμα μόνιμης κατάστασης. Για να χρησιμοποιήσουμε ένα τέτοιου είδους ελεγκτή θα πρέπει να σχεδιαστεί με κατάλληλη επιλογή των τριών παραμέτρων του αναλογικού κέρδους, του ολοκληρωτικού συντελεστή και του διαφορικού συντελεστή έτσι ώστε το σύστημα κλειστού βρόχου να έχει τα πλεονεκτήματα όλων των ελεγκτών που συνδυάζονται στον PID ε-

λεγκτή. Στη ουσία το πιο δύσκολο στη σχεδίαση ενός PID ελεγκτή είναι να δοθούν οι κατάλληλες τιμές στα K_p , K_d και K_i έτσι ώστε το ένα μην επιδρά αρνητικά στη λειτουργία του άλλου.

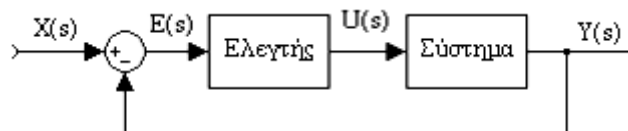
Το m-file του PID ελεγκτή είναι το εξής:

```
function Gc=PIDcontrol(H,A,Kp,Kd,Ki)
PID=tf([Kd Kp Ki],[1 0]);%Συνάρτηση μεταφοράς του PID-ελεγκτή
Gc=feedback(series(PID,H),A);
```

8.7 Ασκήσεις Πράξης ελεγκτών Συστημάτων Αυτόματου Ελέγχου

Άσκηση 1: Δίνεται σύστημα ανοιχτού βρόχου με συνάρτηση μεταφοράς $H(s) = \frac{1}{s^2+s+1}$ του οποίου επιθυμούμε να βελτιώσουμε την απόκριση. Να χρησιμοποιηθούν οι ελεγκτές P-ελεγκτής ($K_p=10$), PD-ελεγκτής ($K_p=10$, $K_d=1$) και PID-ελεγκτής ($K_p=100$, $K_d=100$ και $K_i=1$) και να δοθεί κοινό γράφημα με την απόκριση του συστήματος με και χωρίς έλεγχο για κάθε ελεγκτή ξεχωριστά.

Το δομικό διάγραμμα του συστήματος κλειστού βρόχου θα είναι της μορφής του παρακάτω σχήματος.



Για τον P ελεγκτή

Δίνουμε τις παρακάτω εντολές στο MatLab

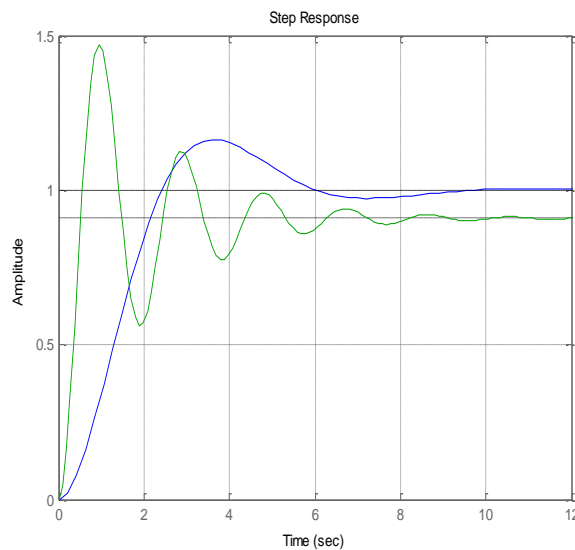
```
H=tf(1,[1 1 1]);
Kp=10;
Gc=Pcontrol(H,1,Kp);
step(H,Gc);grid on;
```

Στο παρακάτω διάγραμμα παρατηρούμε ότι με τη χρήση του P-ελεγκτή το σύστημα έγινε πιο γρήγορο αλλά με σημαντικό ποσοστό υπερύψωσης και ταλαντώσεων και μικρό σφάλμα μόνιμης κατάστασης.

Μπορούμε χρησιμοποιώντας τα χαρακτηριστικά του διαγράμματος (αριστερό κλικ στο διάγραμμα απόκρισης Characteristics) να πάρουμε τις εξής πληροφορίες.

Για το H σύστημα, χρόνος ανύψωσης 1.64 sec, χρόνος αποκατάστασης 8.08 sec και 16.3% ποσοστό υπερύψωσης.

Για το σύστημα με τον έλεγχο, χρόνος ανύψωσης 0.355 sec, χρόνος αποκατάστασης 7.91 sec και 61.9% ποσοστό υπερύψωσης.

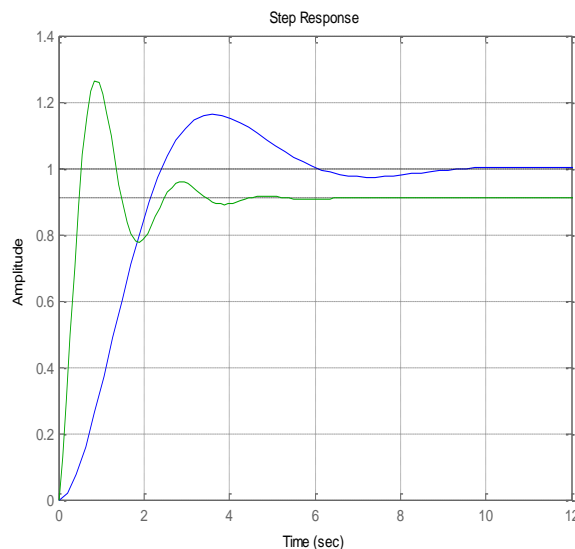


Για τον PD ελεγκτή

Δίνουμε τις παρακάτω εντολές στο MatLab

```
H=tf(1,[1 1 1]);
Kp=10;Kd=1;
Gc=PDcontrol(H,1,Kp,Kd);
step(H,Gc);grid on;
```

Στο παρακάτω διάγραμμα παρατηρούμε ότι με τη χρήση του PD-ελεγκτή το σύστημα έγινε πιο γρήγορο και με σημαντικό μικρότερο ποσοστό υπερύψωσης και ταλαντώσεων σε σχέση με αυτό του P-ελεγκτή.



Μπορούμε χρησιμοποιώντας τα χαρακτηριστικά του διαγράμματος να πάρουμε τις εξής πληροφορίες.

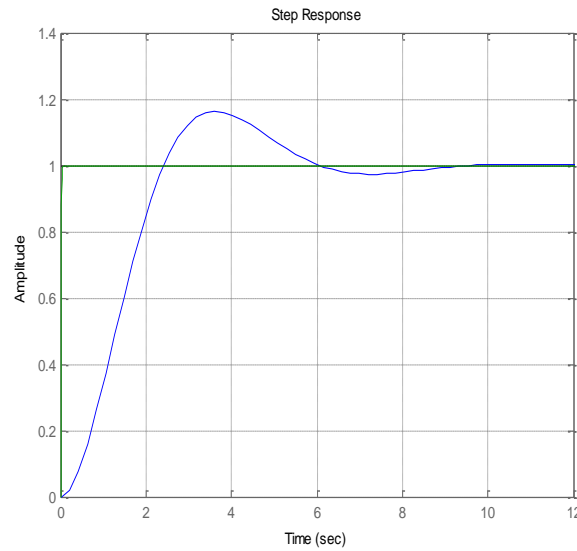
Για το σύστημα με τον έλεγχο, χρόνος ανύψωσης 0.371 sec, χρόνος αποκατάστασης 3.29 sec και 39.1% ποσοστό υπερύψωσης.

Για τον PID ελεγκτή

Δίνουμε τις παρακάτω εντολές στο MatLab

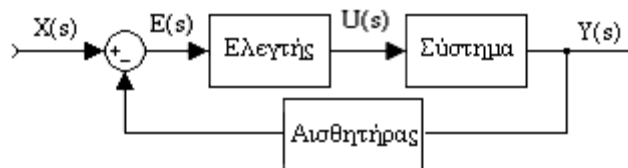
```
H=tf(1,[1 1 1]);
Kp=100;Kd=100;Ki=10;
Gc=PIDcontrol(H,1,Kp,Kd,Ki);
step(H,Gc);grid on;
```

Στο παρακάτω διάγραμμα παρατηρούμε ότι με τη χρήση του PID -ελεγκτή το σύστημα απέκτησε σχεδόν τέλεια απόκριση.



Για το σύστημα με τον έλεγχο, χρόνος ανύψωσης 0.0223 sec, χρόνος αποκατάστασης 0.00403 sec και μηδενικό ποσοστό υπερύψωσης και μηδενικό σφάλμα μόνιμης κατάστασης.

Άσκηση 2: Δίνεται το παρακάτω δομικό διάγραμμα συστήματος κλειστού βρόγχου.



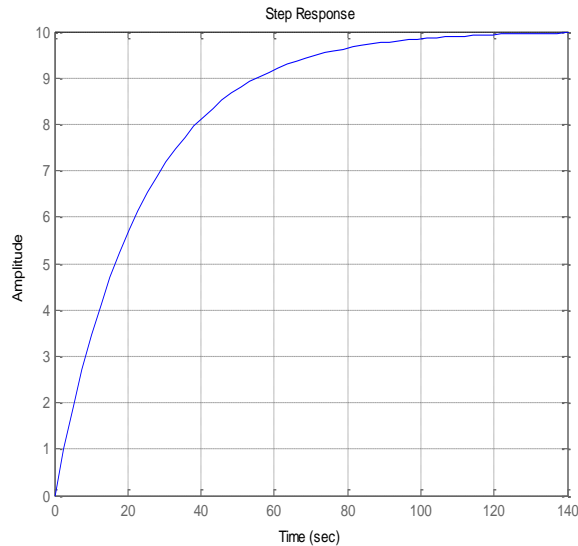
Το παραπάνω σύστημα χρησιμοποιείται για έλεγχο ταχύτητας για αυτό το λόγο ο αισθητήρας λαμβάνει σήμα γωνιακής ταχύτητας και επιστρέφει σήμα τάσης αφού το σήμα ελέγχου είναι τάση. Το σύστημα ελέγχεται από σήμα τάσης 0 έως 50 Volt και οι μέγιστες στροφές που έχουμε στην έξοδο είναι 1500 rad οπότε ο αισθητήρας που θα χρησιμοποιήσουμε έχει λόγο $\frac{1}{30}$.

Επιθυμούμε να βελτιώσουμε το σύστημα με τη χρήση P-ελεγκτή με διαφορετικό αναλογικό κέρδος κάθε φορά ($K_p=10, 100$ και 1000), το σύστημα είναι 1^{ης} τάξης με συνάρτηση μεταφοράς $H(s) = \frac{10}{24s+1}$.

Από τη συνάρτηση μεταφοράς του συστήματος συμπεραίνουμε ότι έχει σταθερά ενίσχυσης 10 και σταθερά χρόνου 24.

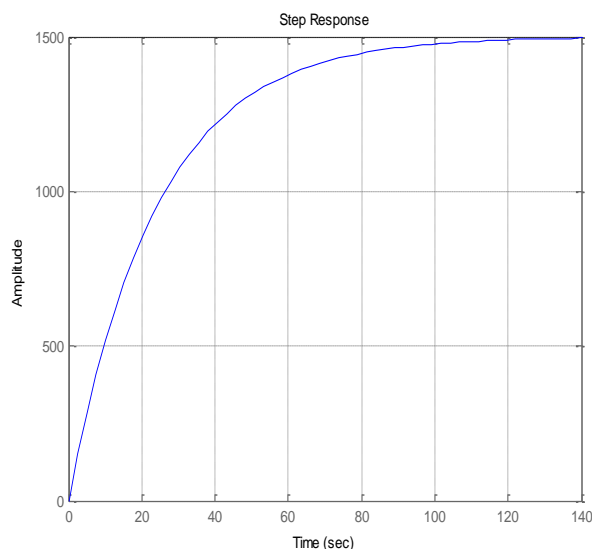
Η απόκριση του συστήματος για βηματική είσοδο είναι:

```
H=tf([10],[24 1]);
step(H);grid on;
```



Για να πλησιάσει τα 1500 rad που θέλουμε εμείς στην έξοδο το ανοιχτού βρόγχου σύστημα θα χρειαζόταν ενίσχυση ίση με $\frac{1500}{10} = 150$, άρα η απόκριση του συστήματος ανοιχτού βρόγχου με την ενίσχυση είναι:

```
step(150*H);grid on;
```



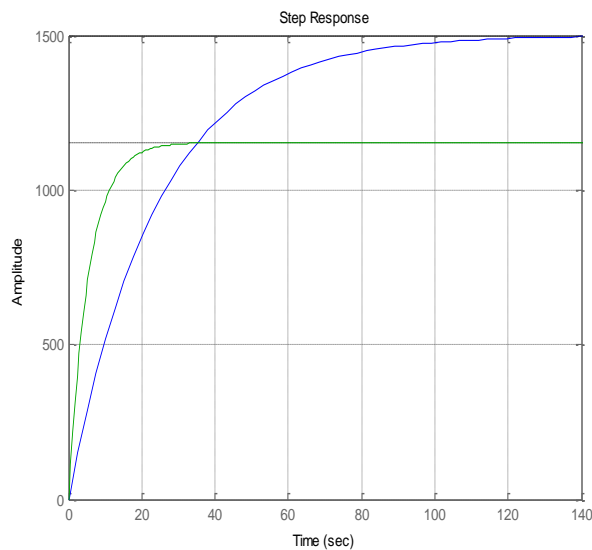
Για τον P ελεγκτή με $K_p=10$

Δίνουμε τις παρακάτω εντολές με τις οποίες υπολογίζουμε τη σταθερά κέρδους και τη σταθερά χρόνου του συστήματος με τον ελεγκτή και συγκρίνουμε τις αποκρίσεις των συστημάτων με και χωρίς ελεγκτή (την απόκριση του

υπό έλεγχο συστήματος την πολλαπλασιάζουμε με 50 που είναι η τάση εισόδου).

```
H=tf([10],[24 1]);
Kp=10;
Gc=Pcontrol(H,1/30,Kp);
Ks=dcgain(Gc)
Ks =
    23.0769
Ts=-1/pole(Gc)
Ts =
    5.5385
step(150*H,50*Gc);grid on;
```

Η νέα σταθερά κέρδους είναι 23.0769 και η σταθερά χρόνου είναι 5.5385.

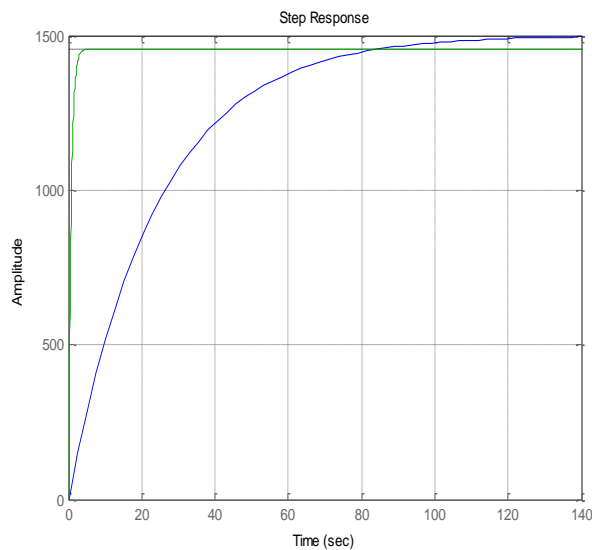


Επιτυγχάνουμε γρηγορότερη απόκριση του συστήματος αλλά εισάγουμε και σφάλμα μόνιμης κατάστασης κάτι που δεν είναι επιθυμητό.

Για τον P ελεγκτή με $K_0=100$

```
Kp=100;
Gc=Pcontrol(H,1/30,Kp);
Ks=dcgain(Gc)
Ks =
    29.1262
Ts=-1/pole(Gc)
Ts =
    0.6990
step(150*H,50*Gc);grid on;
```

Η νέα σταθερά κέρδους είναι 29.1262 και η σταθερά χρόνου είναι 0.6990.



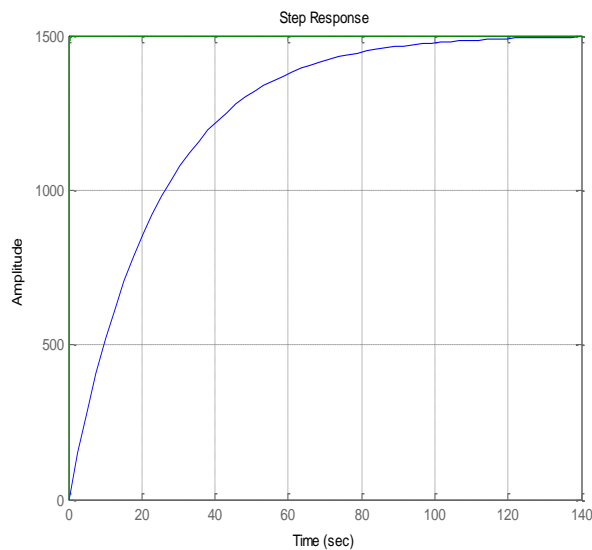
Με την αύξηση του αναλογικού κέρδους του ελεγκτή παρατηρούμε ότι η απόκριση γίνεται ακόμα πιο γρήγορη και καταφέραμε να μειώσουμε το σφάλμα μόνιμης κατάστασης που είχαμε εισάγει στο σύστημα.

Για τον P ελεγκτή με $K_p=1000$

```
Kp=1000;
Gc=Pcontrol(H, 1/30, Kp);
Ks=dcgain(Gc)
Ks =
    29.9103
Ts=-1/pole(Gc)
Ts =
    0.0718
step(150*H, 50*Gc); grid on;
```

Η νέα σταθερά κέρδους είναι 29.9103 και η σταθερά χρόνου είναι 0.0718. Παρατηρούμε από το παρακάτω διάγραμμα της απόκρισης του συστήματος ότι για τιμή αναλογικού κέρδους ίση με 1000 η ταχύτητα της απόκρισης είναι εξαιρετική και το σφάλμα μόνιμης κατάστασης του συστήματος είναι μηδενικό.

Συμπεραίνουμε ότι η σωστή επιλογή των συντελεστών ενός ελεγκτή είναι αυτό που θα μας εξασφαλίσει τη βελτίωση του συστήματος που θέλουμε να υποβάλουμε σε έλεγχο.



Άσκηση 3: Δίνεται παρακάτω η συνάρτηση μεταφοράς ενός συστήματος,

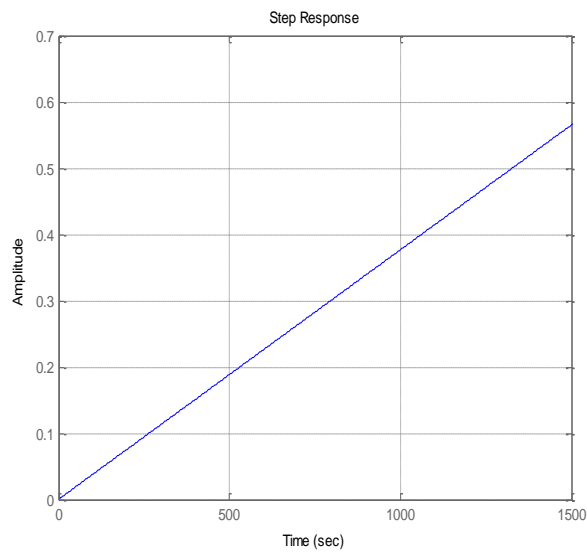
$$H(s) = \frac{4}{4 * s^5 + 92 * s^4 + 1045 * s^3 + 5550 * s^2 + 10625 * s}$$

- i) Να βρεθεί ο χρόνος που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής τιμής της για βηματική είσοδο.
- ii) Να βρεθεί ο χρόνος που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής τιμής της για βηματική είσοδο με τη χρήση P-ελεγκτή με $K_p=1$.
- iii) Να βρεθούν οι παράμετροι ενός PD-ελεγκτή έτσι ώστε να επιτυγχάνει τη μείωση του χρόνου που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής τιμής της κατά 6000 φορές και ταυτόχρονα η υπερύψωση της απόκρισης να μην υπερβαίνει το ποσοστό του 10%.

i) Δίνουμε τις παρακάτω εντολές στο MatLab.

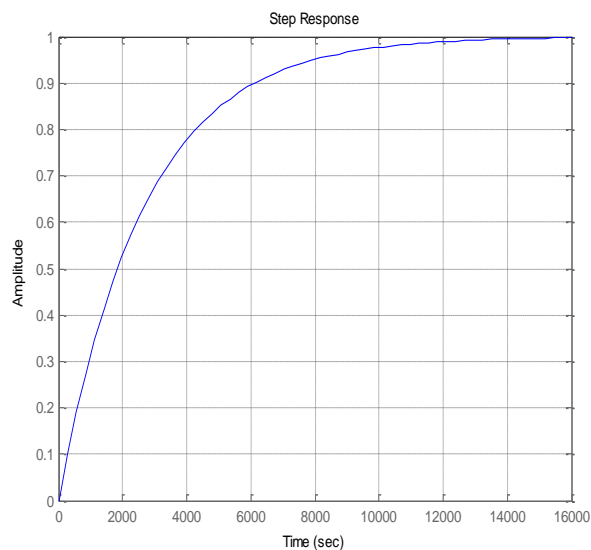
```
H=tf([4],[4 92 1045 5550 10625 0]);
step(H);grid on;
```

Από το παρακάτω διάγραμμα της απόκρισης του συστήματος δεν μπορούμε να βρούμε το χρόνο που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής τιμής της αφού το σύστημα δεν σταθεροποιείται.



ii) Για τη χρήση του αναλογικού ελεγκτή δίνουμε τις παρακάτω εντολές στο MatLab.

```
H=tf([4],[4 92 1045 5550 10625 0]);
Kp=1;
Gc=Pcontrol(H,1,Kp);
step(Gc);grid on;
```

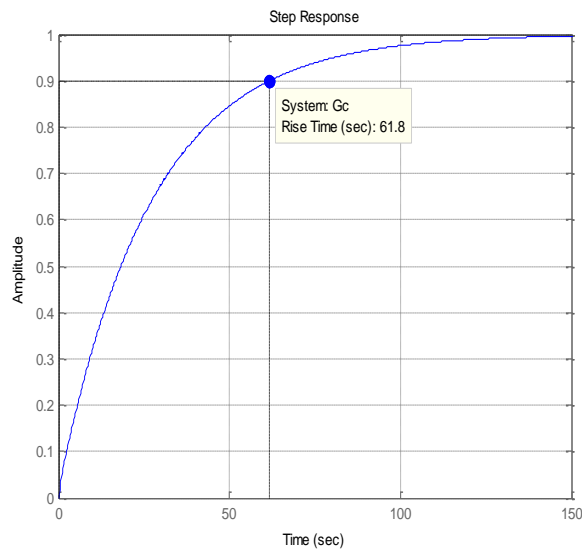


Ο χρόνος που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής της τιμής είναι 6120 sec (αριστερό κλικ στο διάγραμμα απόκρισης Characteristics → Rise time με κριτήριο από 0-90%).

iii) Για να βρούμε τις τιμές των παραμέτρων του PD-ελεγκτή που θα ικανοποιούν τις απαιτήσεις θα πειραματιστούμε με διάφορες τιμές

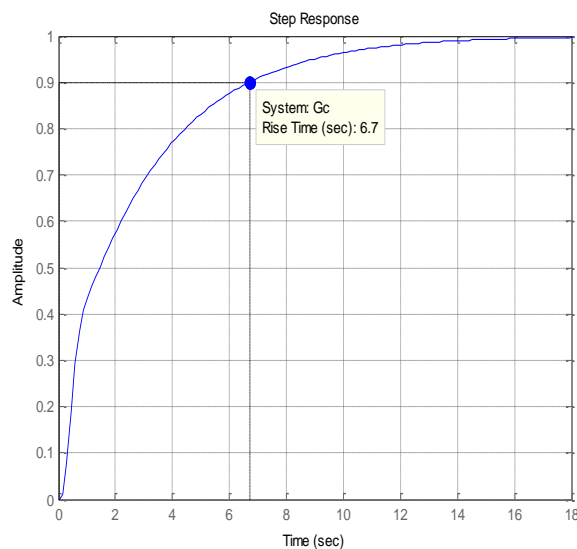
των K_p και K_d . Επιθυμούμε rise time μικρότερο των $\frac{6120}{6000} = 1.02 \text{ sec}$ και ποσοστό υπερύψωσης κάτω από το 10%.

```
Kp=100;Kd=100;
Gc=PDcontrol(H,1,Kp,Kd);
step(Gc)
```



Το σύστημα μπορεί για τις παραπάνω τιμές των παραμέτρων του PD-ελεγκτή να μην παρουσιάζει υπερύψωση αλλά το rise time (0-90%) είναι πολύ μεγαλύτερο από ότι επιθυμούμε.

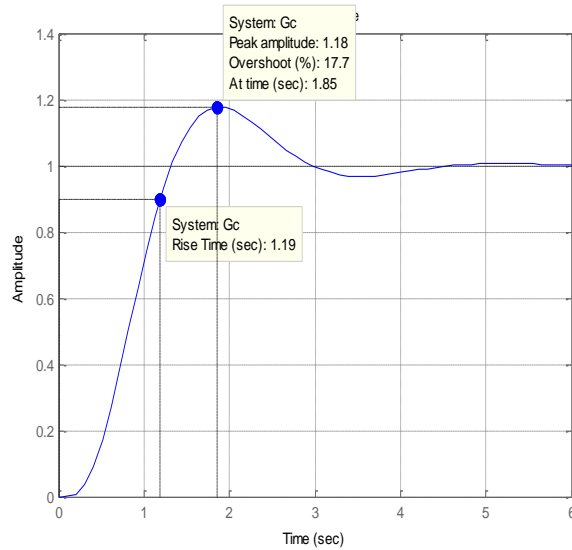
```
Kp=1000;Kd=1000;
Gc=PDcontrol(H,1,Kp,Kd);
step(Gc);grid on;
```



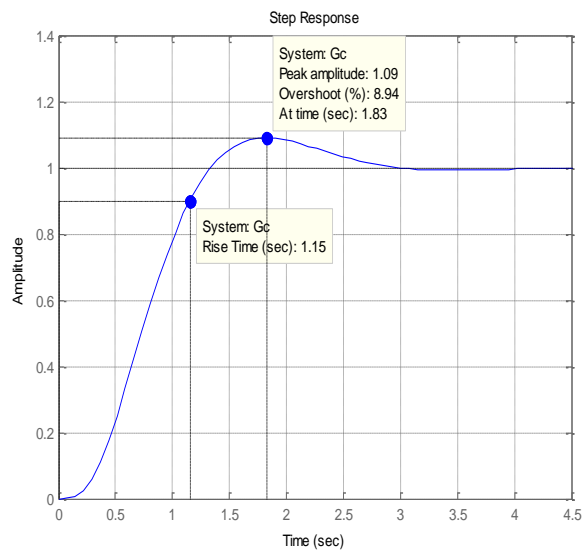
Σε αυτή τη περίπτωση μειώσαμε το rise time (0-90%) αλλά ακόμα απέχει από τα επιθυμητά επίπεδα.

8. Ελεγκτές Συστημάτων Αυτόματου Ελέγχου

```
Kp=4000;Kd=100;  
Gc=PDcontrol(H,1,Kp,Kd);  
step(Gc);grid on
```

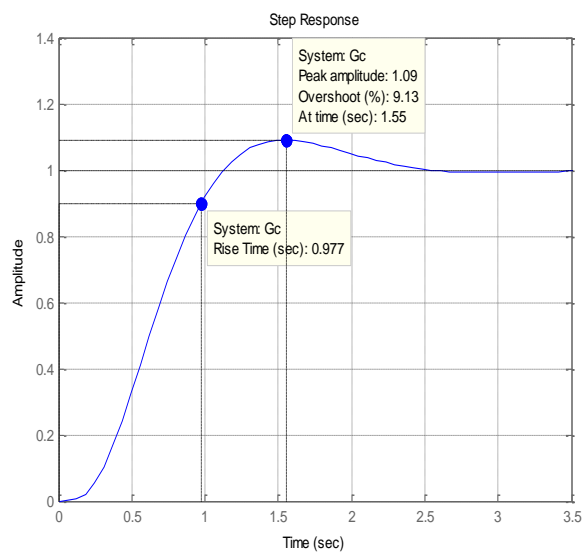


```
Kp=4000;Kd=500;  
Gc=PDcontrol(H,1,Kp,Kd);  
step(Gc);grid on
```



Μετά από αρκετές πειραματικές μετρήσεις καταλήγουμε,

```
Kp=4700;Kd=850;  
Gc=PDcontrol(H,1,Kp,Kd);  
step(Gc);grid on
```

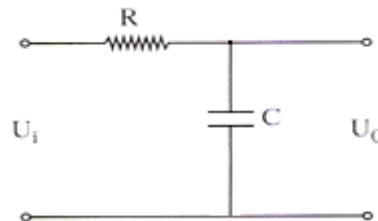


Με τιμές παραμέτρων $K_p=4700$ και $K_d=850$ επιτυγχάνουμε χρόνο που απαιτείται για να φτάσει η απόκριση του συστήματος στο 90% της τελικής τιμής της 0.977 sec και ποσοστό υπερύψωσης 9.13%.

9. Ασκήσεις Πράξης Συστημάτων Αυτόματου Ελέγχου

9.1 Φίλτρα

Άσκηση 1: Να βρεθεί η συνάρτηση μεταφοράς κλειστού βρόγχου του παρακάτω RC φίλτρου και να δοθεί το διάγραμμα βαθμίδων στο Simulink.



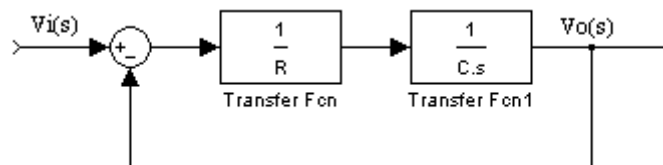
Ισχύουν οι εξισώσεις,

$$\begin{cases} V_{i(s)} = R * I_{(s)} \\ V_{o(s)} = \frac{I_{(s)}}{C * s} \end{cases} \leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{I_{(s)}}{R * I_{(s)}} \leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{1}{R * C * s}$$

Άρα η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου θα είναι:

$$F(s) = \frac{1}{R * C * s} \leftrightarrow F(s) = \frac{1}{R * C * s + 1}$$

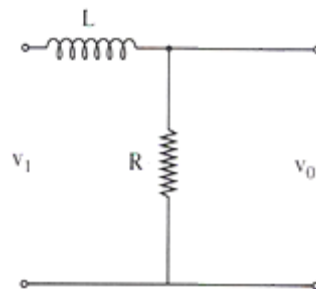
Το διάγραμμα βαθμίδων είναι:



Για να βρούμε εύκολα τη συνάρτηση μεταφοράς για RC φίλτρα για διαφορετικές τιμές αντίστασης και πυκνωτή δημιουργήσαμε το παρακάτω m-file το οποίο μας επιστρέφει τη συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου του φίλτρου.

```
function f=RCfilter(R,C)
h1=tf(1,R);
h2=tf(1,[C 0]);
h=series(h1,h2);
f=feedback(h,1);
```

Άσκηση 2: Να βρεθεί η συνάρτηση μεταφοράς κλειστού βρόγχου του παρακάτω LRφίλτρου και να δοθεί το διάγραμμα βαθμίδων στο Simulink.



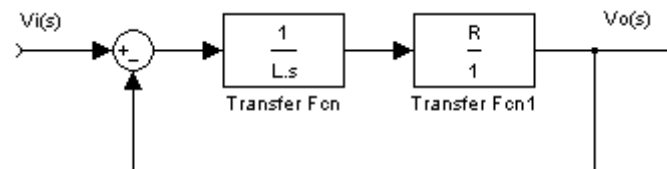
Ισχύουν οι εξισώσεις,

$$\begin{cases} V_{i(s)} = L * s * I(s) \\ V_{o(s)} = R * I(s) \end{cases} \leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R * I(s)}{L * s * I(s)} \leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R}{L * s}$$

Άρα η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου θα είναι:

$$F(s) = \frac{\frac{R}{L * s}}{1 + \frac{R}{L * s}} \leftrightarrow F(s) = \frac{R}{L * s + R} \leftrightarrow F(s) = \frac{1}{\frac{L}{R} * s + 1}$$

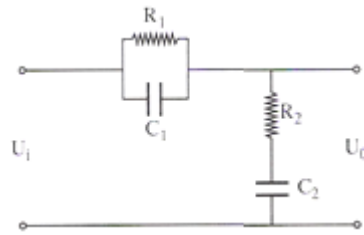
Το διάγραμμα βαθμίδων είναι:



Για να βρούμε εύκολα τη συνάρτηση μεταφοράς για RL φίλτρα για διαφορετικές τιμές πηνίου και αντίστασης δημιουργήσαμε το παρακάτω m-file το οποίο μας επιστρέφει τη συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου του φίλτρου.

```
function f=LRfilter(L,R)
h1=tf(1,[L 0]);
h2=tf(R,1);
h=series(h1,h2);
f=feedback(h,1);
```

Άσκηση 3: Να βρεθεί η συνάρτηση μεταφοράς κλειστού βρόγχου του παρακάτω φίλτρου και να δοθεί το διάγραμμα βαθμίδων στο Simulink.



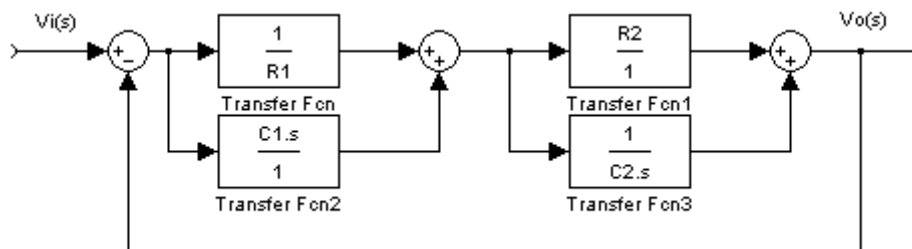
Ισχύουν οι εξισώσεις,

$$\begin{cases} V_{i(s)} = \frac{R_1}{R_1 * C_1 * s + 1} * I(s) \\ V_{o(s)} = \frac{R_2 * C_2 * s + 1}{C_2 * s} * I(s) \end{cases} \Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2 * C_2 * s + 1}{R_1 * C_1 * s + 1} \Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{(R_2 * C_2 * s + 1) * (R_1 * C_1 * s + 1)}{R_1 * C_2 * s}$$

Άρα η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου θα είναι:

$$F(s) = \frac{(R_1 * R_2 * C_1 * C_2) * s^2 + (R_1 * C_1 + R_2 * C_2) * s + 1}{(R_1 * R_2 * C_1 * C_2) * s^2 + (R_1 * C_1 + R_2 * C_2 + R_1 * C_2) * s + 1}$$

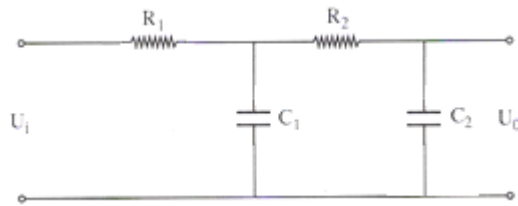
Το διάγραμμα βαθμίδων είναι:



Για να βρίσκουμε εύκολα τη συνάρτηση μεταφοράς για RCRC φίλτρα της παραπάνω μορφής για διαφορετικές τιμές πηνίου και αντίστασης δημιουργήσαμε το παρακάτω m-file το οποίο μας επιστρέφει τη συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου του φίλτρου.

```
function f=RCRC1filter(R1,C1,R2,C2)
h1=tf(1,R1);h2=tf([C1 0],1);
g1=tf(R2,1);g2=tf(1,[C2 0]);
h=parallel(h1,h2);
g=parallel(g1,g2);
F=series(h,g);
f=feedback(F,1);
```

Άσκηση 4: Να βρεθεί η συνάρτηση μεταφοράς κλειστού βρόγχου του παρακάτω φίλτρου.



Ισχύουν οι εξισώσεις,

$$\begin{cases} V_{i(s)} = \frac{R_1 * C_1 * s + 1}{C_1 * s} * I_{(s)} \\ V_{o(s)} = \frac{R_2 * C_2 * s + 1}{C_2 * s} * I_{(s)} \end{cases} \Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2 * C_2 * s + 1}{R_1 * C_1 * s + 1} * \frac{C_1 * s}{C_2 * s} \Leftrightarrow$$

$$\Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{(R_2 * C_2 * s + 1) * C_1 * s}{(R_1 * C_1 * s + 1) * C_2 * s} \Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2 * C_1 * C_2 * s + C_1}{R_1 * C_1 * C_2 * s + C_2} \Leftrightarrow$$

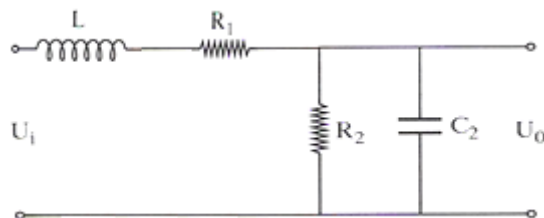
Άρα η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου θα είναι:

$$F(s) = \frac{R_2 * C_1 * C_2 * s + C_1}{(R_1 + R_2) * C_1 * C_2 * s + (C_1 + C_2)}$$

Για να βρούμε εύκολα τη συνάρτηση μεταφοράς για RCRC φίλτρα της παραπάνω μορφής για διαφορετικές τιμές πηνίου και αντίστασης δημιουργήσαμε το παρακάτω m-file το οποίο μας επιστρέφει τη συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου του φίλτρου.

```
function f=RCRC2filter(R1,C1,R2,C2)
h1=tf(1,R1);h2=tf(1,[C1 0]);g2=tf([C2 0],1);
F1=RCfilter(R2,C2);
F2=feedback(series(F1,h2),g2);
f=feedback(series(h1,F2),inv(F1));
```

Άσκηση 5: Να βρεθεί η συνάρτηση μεταφοράς κλειστού βρόγχου του παρακάτω φίλτρου



Ισχύουν οι εξισώσεις,

$$\begin{cases} V_{i(s)} = (L * s + R_1) * I_{(s)} \\ V_{o(s)} = \frac{R_2}{R_2 * C_2 * s + 1} * I_{(s)} \end{cases} \Leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2}{(L * s + R_1) * I_{(s)}} * \frac{1}{R_2 * C_2 * s + 1} \Leftrightarrow$$

$$\leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2}{(L * s + R_1) * (R_2 * C_2 * s + 1)} \leftrightarrow$$

$$\leftrightarrow \frac{V_{o(s)}}{V_{i(s)}} = \frac{R_2}{(R_2 * C_2 * L) * s^2 + (R_1 * R_2 * C_2 + L) * s + R_1}$$

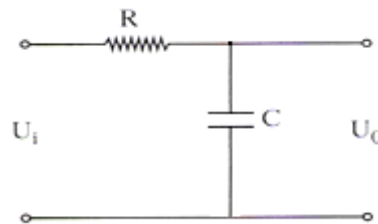
Άρα η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου θα είναι:

$$F(s) = \frac{R_2}{(R_2 * C_2 * L) * s^2 + (R_1 * R_2 * C_2 + L) * s + (R_1 + R_2)}$$

Για να βρούμε εύκολα τη συνάρτηση μεταφοράς για RLRC φίλτρα της παραπάνω μορφής για διαφορετικές τιμές πηνίου και αντίστασης δημιουργήσαμε το παρακάτω m-file το οποίο μας επιστρέφει τη συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου του φίλτρου.

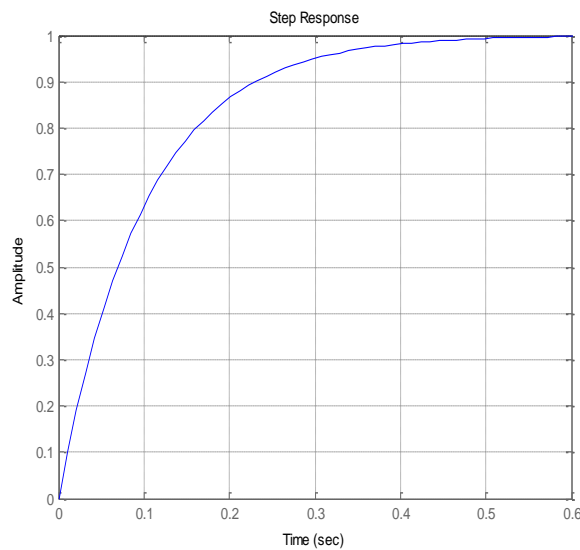
```
function f=RLRCfilter(R1,L,R2,C)
h1=tf(1,[L R1]);
h2=tf([R2],[C*R2 1]);
h=series(h1,h2);
f=feedback(h,1);
```

Άσκηση 6: Να βρεθεί η βηματική απόκριση του παρακάτω φίλτρου για $R=10$ Ω και $C=0.01$ F και στη συνέχεια να επαληθευτεί μέσω του Simulink.

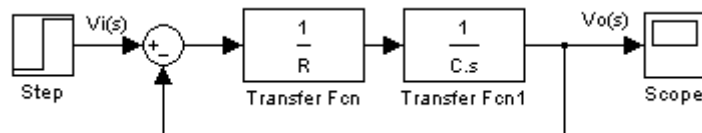


Δίνουμε τις παρακάτω εντολές,

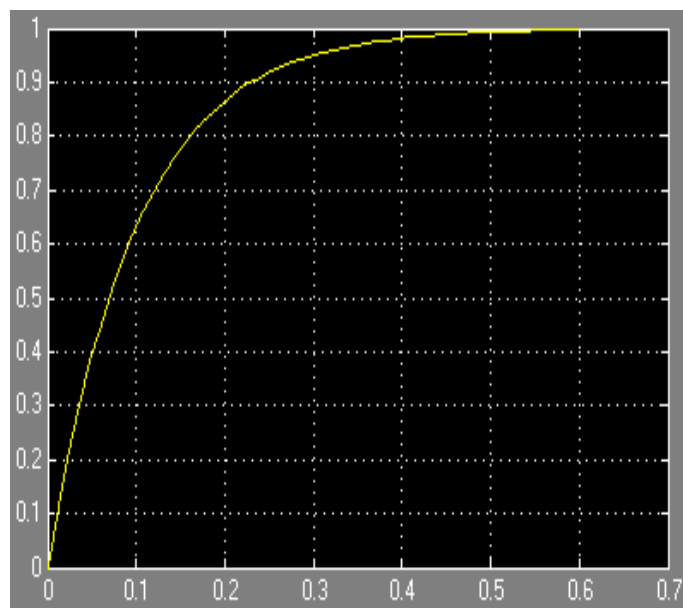
```
R=10;C=0.01;
F=RCfilter(R,C);
step(F);grid on;
```



Φτιάχνουμε το παρακάτω δομικό διάγραμμα στο Simulink.

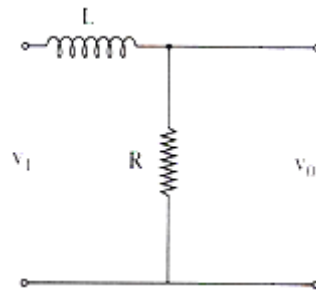


Τη βαθμίδα Step την επιλέγουμε από τη βιβλιοθήκη Source του Simulink ενώ το Scope από το Sinks. Το Step το χρησιμοποιούμε ώστε να εισάγουμε στο σύστημα βηματική είσοδο και μπορούμε να εισάγουμε τις παραμέτρους που επιθυμούμε με διπλό κλικ πάνω στη βαθμίδα, στη περίπτωση της άσκησης θέτουμε step time=0 και final value=1. Το Scope στην ουσία είναι ένας παλμογράφος στον οποίο θα βλέπουμε το σήμα εξόδου του συστήματος. Ρυθμίζουμε χρόνο μοντέλου στα 0.6 sec (ίδιος με τη βηματική του MatLab) και τρέχουμε το μοντέλο και μας δίνει την παρακάτω απόκριση στον παλμογράφο.



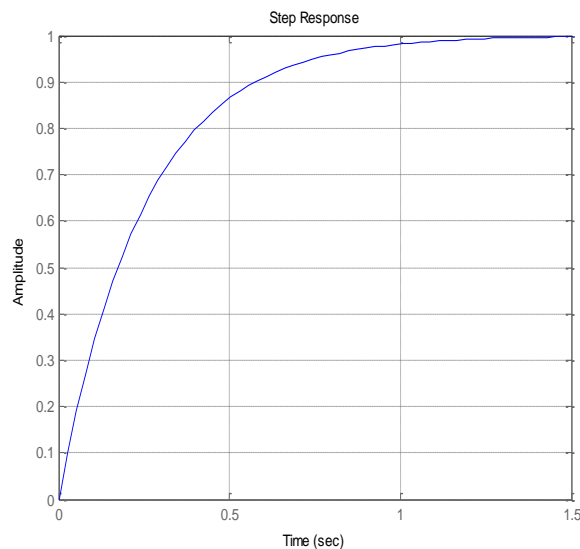
Επομένως η απόκριση από το μοντέλο του Simulink επαληθεύει την απόκριση που υπολογίσαμε μέσω της συνάρτησης μεταφοράς στο MatLab.

Άσκηση 7: Να βρεθεί η βηματική απόκριση του παρακάτω φίλτρου για $R=10\ \Omega$ και $H=2.5\ H$ και στη συνέχεια να επαληθευτεί μέσω του Simulink.

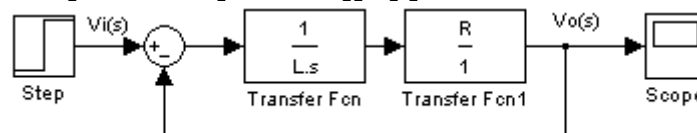


Δίνουμε τις παρακάτω εντολές,

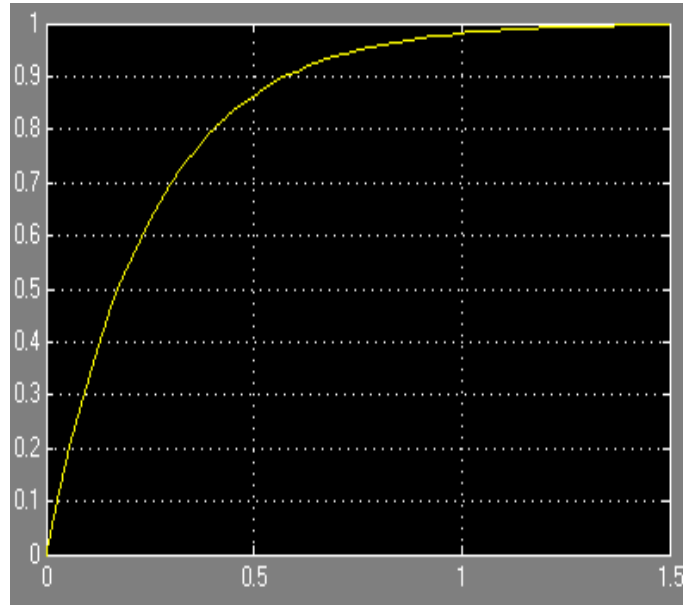
```
R=10;L=2.5;
F=LRfilter(L,R);
step(F);grid on;
```



Φτιάχνουμε το παρακάτω δομικό διάγραμμα στο Simulink.

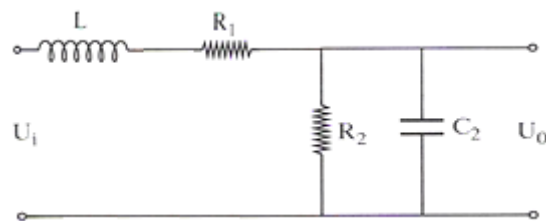


Στο Step εισάγουμε τις παραμέτρους step time=0 και final value=1. Ρυθμίζουμε χρόνο μοντέλου στα 1.5 sec (ίδιος με τη βηματική του MatLab) και τρέχουμε το μοντέλο και μας δίνει την παρακάτω απόκριση στον παλμογράφο.

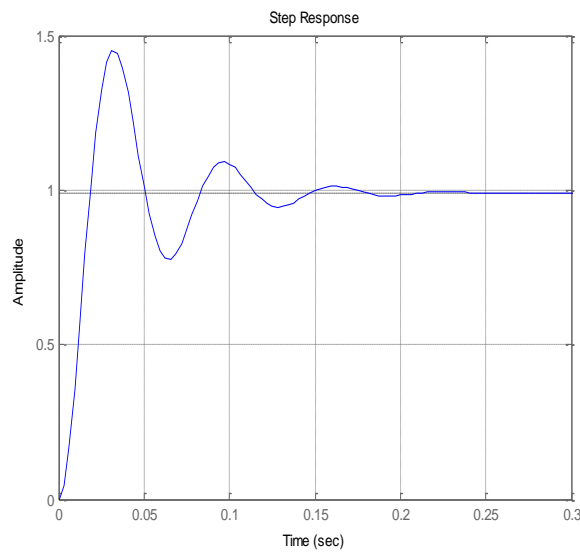


Επομένως η απόκριση από το μοντέλο του Simulink επαληθεύει την απόκριση που υπολογίσαμε μέσω της συνάρτησης μεταφοράς στο MatLab.

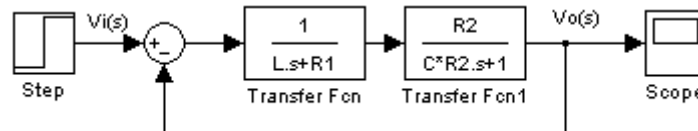
Άσκηση 8: Να βρεθεί η βηματική απόκριση του παρακάτω φίλτρου για $R_1=5 \Omega$, $R_2=550 \Omega$, $L=2.5 \text{ H}$ και $C=4 \cdot 10^{-5} \text{ F}$ και στη συνέχεια να επαληθευτεί μέσω του Simulink.



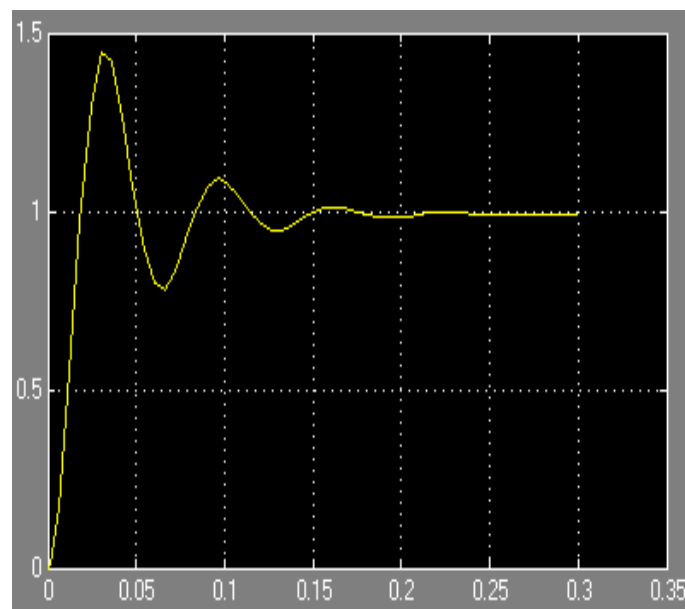
Δίνουμε τις παρακάτω εντολές,
 $R_1=5; L=2.5; C=4 \cdot 10^{-5}; R_2=550;$
 $F=RLRCfilter(R_1, L, R_2, C);$
 $step(F); grid \text{ on};$



Φτιάχνουμε το παρακάτω δομικό διάγραμμα στο Simulink.

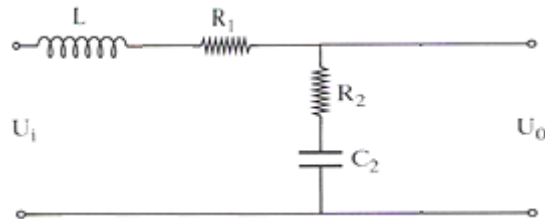


Στο Step εισάγουμε τις παραμέτρους step time=0 και final value=1. Ρυθμίζουμε χρόνο μοντέλου στα 0.3 sec (ίδιος με τη βηματική του MatLab) και τρέχουμε το μοντέλο και μας δίνει την παρακάτω απόκριση στον παλμογράφο.

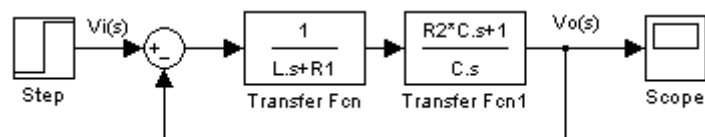


Και σε αυτή τη περίπτωση η απόκριση από το μοντέλο του Simulink επαληθεύει την απόκριση που υπολογίσαμε μέσω της συνάρτησης μεταφοράς στο MatLab.

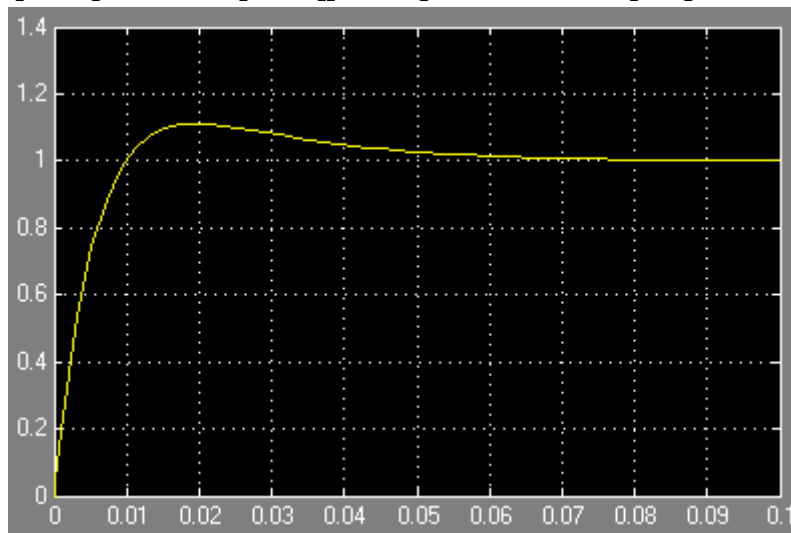
Άσκηση 9: Να βρεθεί το δομικό διάγραμμα του παρακάτω φίλτρου για $R_1=5 \Omega$, $R_2=550 \Omega$, $L=2.5 \text{ H}$ και $C=4 \cdot 10^{-5} \text{ F}$ και στη συνέχεια να δοθεί η βηματική απόκριση μέσω του Simulink.



Το διάγραμμα βαθμίδων στο Simulink είναι το εξής,



Η απόκριση του μοντέλου για βηματική είσοδο είναι η παρακάτω.



Για το παραπάνω κύκλωμα δημιουργήσαμε το παρακάτω m-file.

```
function h=RLRC2filter(R1,L,R2,C)
z1=tf([L R1],1);
z2=tf([C*R2 1],[C 0]);
h=feedback(series(inv(z1),z2),1);
```

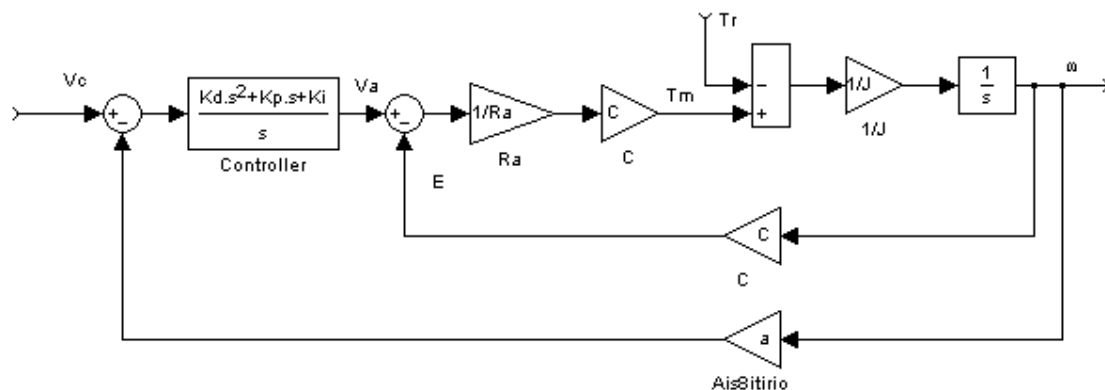
9.2 DC Κινητήρας

Ο κινητήρας συνεχούς ρεύματος μετατρέπει την ηλεκτρική ενέργεια σε μηχανική ενέργεια. Ένα μεγάλο μέρος της ροπής που εμφανίζεται στον άξονα του κινητήρα χρησιμοποιείται για την κίνηση ενός εξωτερικού μηχανικού φορτίου.

Οι κινητήρες συνεχούς ρεύματος διαθέτουν ιδιότητες, όπως υψηλή ροπή, δυνατότητα ελέγχου της περιστροφικής ταχύτητας, καλή απόκριση ταχύτητας και ροπής, που τους καθιστούν ευρύτατα χρήσιμους σε αρκετές εφαρμογές ελέγχου.

Για τη μελέτη του κινητήρα συνεχούς ρεύματος δημιουργήσαμε ένα μοντέλο Simulink και ένα αρχείο m-file.

Το μοντέλο Simulink του κινητήρα συνεχούς ρεύματος.



Τα χαρακτηριστικά που χρησιμοποιήσαμε για το μοντέλο είναι τα εξής:

- ✚ V_c είναι η τάση ελέγχου.
- ✚ ω είναι η υπό έλεγχο γωνιακή ταχύτητα κινητήρα.
- ✚ Aisditirio είναι ο αισθητήρας που μετατρέπει την γωνιακή ταχύτητα σε τάση.
- ✚ R_a είναι η αντίσταση τυμπάνου του κινητήρα.
- ✚ I_a είναι το ρεύμα τυμπάνου του κινητήρα.
- ✚ V_a είναι η τάση τυμπάνου του κινητήρα.
- ✚ Controller είναι ο ελεγκτής ο οποίος υπολογίζει την τάση τυμπάνου V_a λαμβάνοντας υπόψη πληροφορίες από το σφάλμα $e = (V_c - V_\omega)$, μπορεί να είναι οποιοδήποτε είδος από τους ελεγκτές που έχουμε αναφέρει.
- ✚ C η χαρακτηριστική σταθερά του κινητήρα
- ✚ T_m η κινητήρια ροπή.
- ✚ T_r η ανθισταμένη ροπή.
- ✚ J η ολική ροπή αδραειας ανοιγμένη στον άξονα του κινητήρα.

Το σύστημα ανοικτού βρόχου του κινητήρα έχει δυο εισόδους. Την τάση τυμπάνου και την ανθισταμένη ροπή. Η πρώτη ονομάζεται είσοδος ελέγχου και η δεύτερη είσοδος διαταραχής. Η σχεδίαση του συστήματος αυτομάτου ελέγχου στοχεύει στο να διατηρούνται σταθερές οι στροφές του κινητήρα όταν μεταβάλλεται η ροπή φορτίου.

Τα χαρακτηριστικά που επιθυμούμε να πετύχουμε για τις συναρτήσεις μεταφοράς 1^{ης} τάξης είναι δυο η ταχύτητα και η ακρίβεια της απόκρισης του συστήματος. Η αύξηση της ταχύτητας της απόκρισης του συστήματος κλειστού βρόχου επιτυγχάνεται με τη μείωση της σταθερά χρόνου του συστήματος κλειστού βρόχου του κινητήρα, ενώ όπως είναι λογικό η απόκριση του συστήματος γίνεται ακριβέστερη όταν το σφάλμα μόνιμης κατάστασης έχει τιμή όσο πιο κοντά στο μηδέν γίνεται.

Το m-file του κινητήρα συνεχούς ρεύματος είναι το εξής:

```
function [H G]=DCMotor(R,C,J)
%W(s)=[H G]*[V;T]
%W: Angular velocity of the shaft
%V: Input Voltage
%T: Load Torque
%*****
%R: Armature resistance
%C: Electromechanical characteristic (E=CW, M=CI)
%E: counter electromotive force
%I: Armature current
%M: Motor torque
%J: Total Inertia on the shaft
h1=tf(1,R);h2=tf(C,1);h3=tf(1,[J 0]);
f1=series(h1,h2);f2=series(f1,h3);
H=feedback(f2,h2);
G=-feedback(h3,series(f1,h2));
```

Παρατηρούμε ότι το m-file μας επιστρέφει δύο συναρτήσεις μεταφοράς, η συνάρτηση μεταφοράς $H(s)$ μας δίνει το λόγο εξόδου και εισόδου του κινητήρα όταν έξοδος είναι η γωνιακή ταχύτητα του κινητήρα και είσοδος η τάση ενώ η συνάρτηση μεταφοράς $G(s)$ μας δίνει το λόγο εξόδου και εισόδου του κινητήρα όταν έξοδος είναι η γωνιακή ταχύτητα αλλά είσοδο τη ροπή του φορτίου.

Η μελέτη στο μοντέλο Simulink γίνεται αρχικά για το σύστημα ανοικτού βρόχου του κινητήρα χωρίς τη χρήση ελεγκτή και του αισθητήρα έτσι ώστε να δούμε τη συμπεριφορά του συστήματος και να επιλέξουμε στη συνέχεια το κατάλληλο αισθητήρα, που θα ορίζει τη τάση ελέγχου ανάλογα με την επιθυμητή τιμή εξόδου, και τον κατάλληλο ελεγκτή που θα βελτιώσει το σύστημα.

Η συνάρτηση μεταφοράς του κινητήρα είναι 1^{ης} τάξης άρα έχει τη μορφή $\frac{K_{\sigma}}{T*s+1}$. Το K_{σ} που είναι η σταθερά κέρδους του κινητήρα υπολογίζεται από το λόγο της τελικής τιμής της εξόδου προς την τιμή της εισόδου του κι-

νητήρα. Π.χ. αν δίνουμε είσοδο στον κινητήρα 10 volt και στο διάγραμμα απόκρισης η τελική τιμή της (όπου σταθεροποιείται) ισούται με 15.8 rad/sec τότε η σταθερά κέρδους θα ισούται με $K_o=1.58$. Για να βρούμε τη σταθερά χρόνου του συστήματος από το διάγραμμα απόκρισης θα βρούμε τη χρονική στιγμή που αντιστοιχεί στο 63.2% της τελικής τιμής της εξόδου. Για την ίδια έξοδο του προηγούμενου παραδείγματος η σταθερά χρόνου θα είναι η χρονική στιγμή κατά την οποία η απόκριση του συστήματος έχει την τιμή $T = \frac{15.8 \cdot 63.2}{100} = 9.9856 \text{ sec.}$

Τους παραπάνω υπολογισμούς μπορούμε να τους επαληθεύσουμε και μέσω του MatLab με τη χρήση του m-file το οποίο μας επιστρέφει απευθείας τη συνάρτηση μεταφοράς του κινητήρα.

Κατά τη διάρκεια της μελέτης του μοντέλου του κινητήρα θα εισάγουμε και φορτίο ως διαταραχή ώστε να βλέπουμε τις μεταβολές στη συμπεριφορά του κινητήρα. Το φορτίο θα το εισάγουμε με χρονική καθυστέρηση τουλάχιστον 10 φορές μεγαλύτερη από τη σταθερά χρόνου του κινητήρα και αυτό για να μπορέσουμε να δούμε τα φαινόμενα που δημιουργούνται δηλαδή να δώσουμε χρόνο στον κινητήρα να φτάσει στην τελική τιμή στην έξοδο και να αντιδράσει το σύστημα στην εισαγωγή του φορτίου.

Είναι πιθανό ένα φορτίο να διαταράξει τη λειτουργία του συστήματος για αυτό το λόγο υπάρχει ο ελεγκτής ο οποίος με τις κατάλληλες τιμές στις παραμέτρους τους καθιστά τον κινητήρα ικανό στο να αντέξει τα φορτία και να έχει ταχύ απόκριση.

Άσκηση 1: Δίνεται κινητήρας συνεχούς ρεύματος με τα εξής χαρακτηριστικά $R=0.5 \Omega$, $C=0.95 \text{ F}$ και $J=12$ και σήμα εισόδου 120 volt. Μετά από 3 λεπτά εισάγουμε ως διαταραχή φορτίο 50 Nt*m, να βρεθεί η απόκριση του συστήματος σε χρονικό ορίζοντα 600 sec.

Αρχικά με τη χρήση του m-file θα βρούμε τις συναρτήσεις μεταφοράς του κινητήρα, δίνουμε τις παρακάτω εντολές.

```
[H G]=DCmotor(0.5,0.95,12)
```

```
Transfer function:
```

```
0.95
```

```
-----  
6 s + 0.9025
```

```
Transfer function:
```

```
-0.5
```

```
-----  
6 s + 0.9025
```

Όπως αναφέραμε μας επιστρέφει δύο συναρτήσεις μεταφοράς πρώτα την $H(s) = \frac{0.95}{6*s+0.9025}$ και δεύτερη την $G(s) = \frac{-0.5}{6*s+0.9025}$. Ορίζουμε το χρονικό ορίζοντα που επιθυμούμε και αφού η $H(s)$ αφορά είσοδο τάσης με τη χρήση της εντολής *lsim* έχουμε:

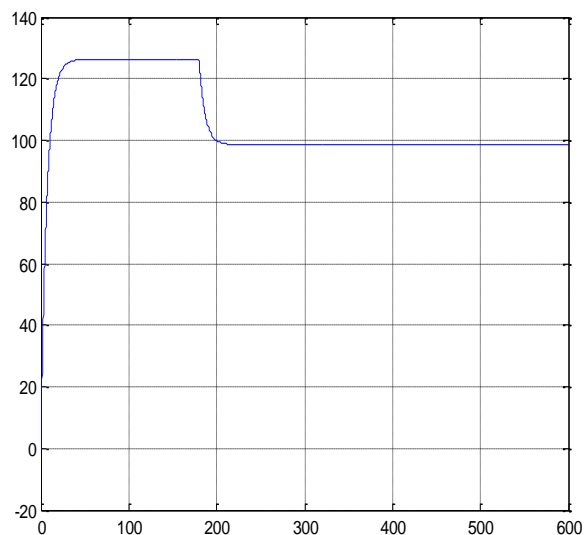
```
time=0:0.1:600;
w1=lsim(H,120*unistep(time,0),time);
```

Ο λόγος που πολλαπλασιάζουμε την *unistep* με 120 είναι για να δηλώσουμε στο παλμό ύψος 120 όσο είναι και η τάση εισόδου. Η $G(s)$ αφορά είσοδο ροπής φορτίου οπότε:

```
w2=lsim(G,50*unistep(time,180),time);
```

Σε αυτή τη περίπτωση πολλαπλασιάσαμε τη *unistep* με 50 γιατί τόσο είναι το φορτίο που εισάγουμε στον κινητήρα, και το 180 αντιστοιχεί στα 3 λεπτά ($3*60=180$ sec) που μεσολαβούν από την εκκίνηση του κινητήρα έως την εισαγωγή του φορτίου. Μέχρι στιγμής έχουμε υπολογίσει τη γωνιακή ταχύτητα για τάση εισόδου και για είσοδο φορτίου, για να βρούμε την απόκριση του κινητήρα θα προσθέσουμε τα δύο παραπάνω αποτελέσματα, οπότε η απόκριση του κινητήρα δίνεται από τις εντολές:

```
w=w1+w2;
plot(time,w);grid on;
```



Παρατηρούμε την πτώση στροφών εξαιτίας της εισαγωγής του φορτίου.

Άσκηση 2: Δίνεται κινητήρας συνεχούς ρεύματος με τα εξής χαρακτηριστικά $R=0.99 \Omega$, $C=0.71 \text{ F}$ και $J=13.3$ και σήμα εισόδου 120 volt. Να βρεθεί η σταθερά χρόνου του συστήματος, η σταθερά κέρδους για τάση εισόδου αλλά και η σταθερά κέρδους για είσοδο ροπή φορτίου.

Αρχικά βρίσκουμε τις συναρτήσεις μεταφοράς του κινητήρα.

```
[H G]=DCmotor(0.99,0.71,13.3)
```

```
Transfer function:
```

```
0.71
```

```
-----  
13.17 s + 0.5041
```

```
Transfer function:
```

```
-0.99
```

```
-----  
13.17 s + 0.5041
```

Άρα $H(s) = \frac{0.71}{13.17s+0.5041}$ και $G(s) = \frac{-0.99}{13.17s+0.5041}$ οπότε τη σταθερά χρόνου του συστήματος μπορούμε να την υπολογίσουμε $\frac{13.17}{0.5041} = 26.125 \text{ sec}$ και την επαληθεύουμε από την εντολή:

```
T=-1/pole(H)
```

```
T =
```

```
26.1198
```

Η διαφορά που παρατηρείται οφείλεται στο γεγονός ότι το MatLab κρατάει στη μνήμη του περισσότερα ψηφία από ότι εμφανίζει στα αποτελέσματα. Ο υπολογισμός της σταθερά κέρδους και για τις δύο εισόδους μπορεί να βρεθεί αν γνωρίζουμε τις τελικές τιμές των αποκρίσεων, επειδή εμείς δεν έχουμε όλα τα στοιχεία για τον υπολογισμό μας βοηθά το MatLab με τις παρακάτω εντολές:

```
Ks1=dcgain(H)
```

```
Ks1 =
```

```
1.4085
```

```
Ks2=dcgain(G)
```

```
Ks2 =
```

```
-1.9639
```

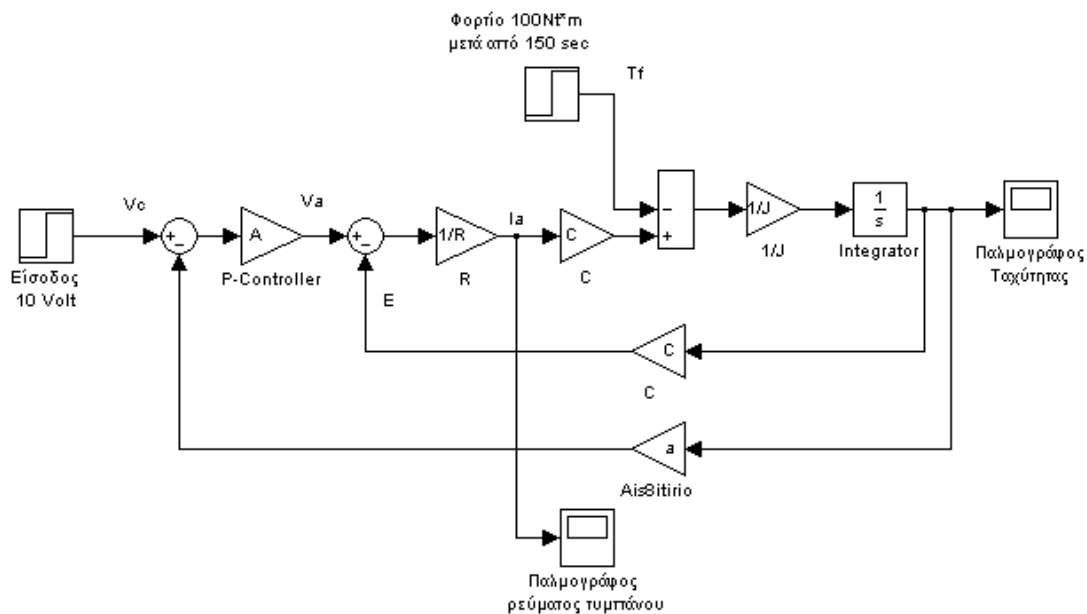
Οπότε η σταθερά κέρδους για είσοδο τάσης είναι 1.4085 ενώ η σταθερά κέρδους για είσοδο ροπή φορτίου ισούται με -1.9639.

Άσκηση 3: Δίνεται κινητήρας συνεχούς ρεύματος με τα εξής χαρακτηριστικά $R=2 \Omega$, $C=0.61 \text{ F}$ και $J=12$.

i) Ο κινητήρας χρησιμοποιείται σε ένα σύστημα ελέγχου ταχύτητας με αισθητήριο με χαρακτηριστική $a=0.1$. Ποιο πρέπει να είναι το αναλογικό κέρδος ενός P-ελεγκτή ώστε η πτώση των στροφών από την εν κενό λειτουργία σε αντίσταση φορτίου $100 \text{ Nt}\cdot\text{m}$ να μην υπερβαίνει το 5%. Να δοθεί διάγραμμα του ρεύματος τυμπάνου από την εν κενό λειτουργία σε πλήρες φορτίο.

ii) Ο κινητήρας χρησιμοποιείται από ένα σύστημα ελέγχου θέσης με το ίδιο αισθητήριο και ένα μειωτήρα στροφών $1/300$. Με αναλογικό κέρδος 1000 να βρεθεί ο χρόνος αποκατάστασης (με κριτήριο 2%) και να βρεθεί η χαρακτηριστική του αισθητηρίου ώστε ο χρόνος αποκατάστασης να μην μεταβληθεί αισθητά και παράλληλα το σύστημα να παρουσιάζει υπερακόντιση 10% (μεταβάλλεται και το αναλογικό κέρδος).

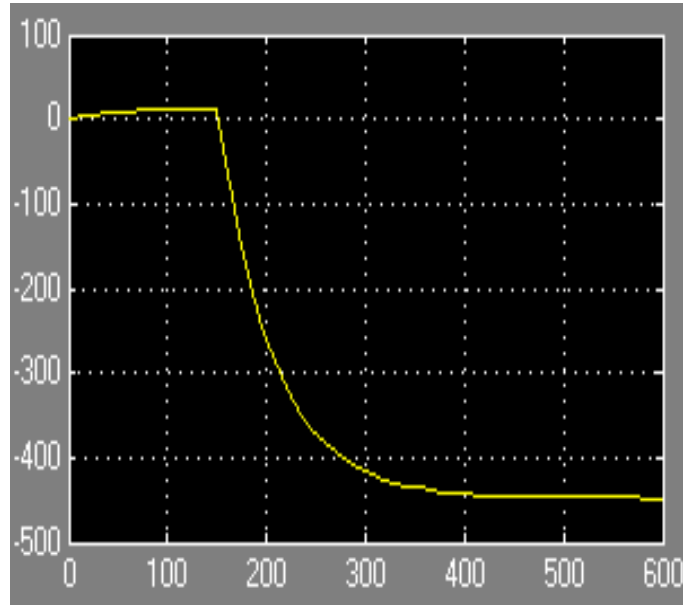
i) Θα χρησιμοποιήσουμε το παρακάτω δομικό διάγραμμα σε μοντέλο του Simulink.



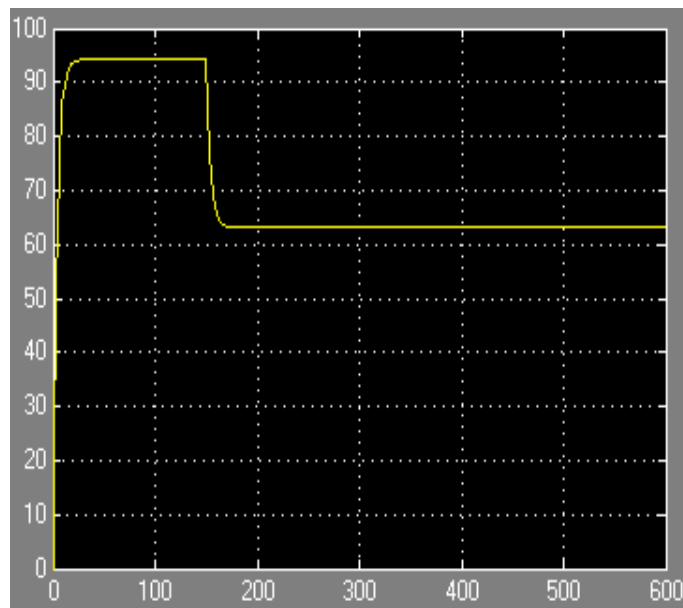
Δίνουμε τις εξής εντολές στο MatLab και παίρνουμε την παρακάτω απόκριση.

```
R=2; C=0.61; J=12;
a=0.1; Kp=1;
```

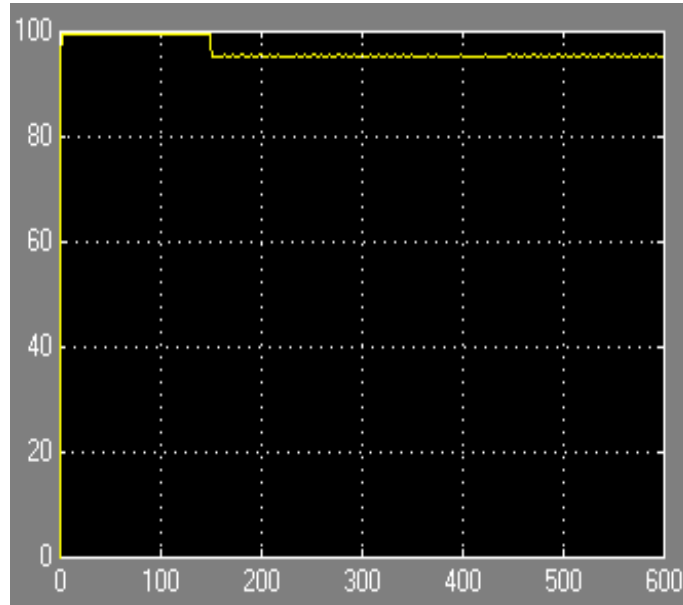
Παρατηρούμε από το παρακάτω διάγραμμα ότι μετά την εισαγωγή του φορτίου ο κινητήρας «γονατίζει» οπότε θα χρειαστούμε αύξηση του αναλογικού κέρδους ώστε να βελτιώσουμε το σύστημα.



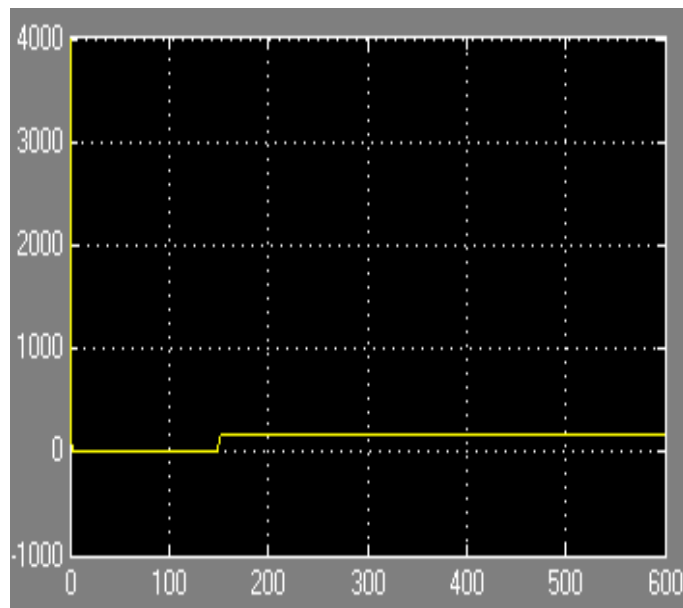
Για K_p ίσο με 100 το σύστημα παρουσιάζει βελτίωση αλλά η πτώση των στροφών από την εν κενό λειτουργία σε αντίσταση φορτίου 100 Nt*m υπερβαίνει το 5%.



Επαναλαμβάνουμε μετρήσεις έως η πτώση στροφών να μην υπερβαίνει το 5%. Για K_p ίσο με 800 η πτώση των στροφών από την εν κενό λειτουργία σε αντίσταση φορτίου 100 Nt*m του συστήματος δεν υπερβαίνει το 5%. (μέγιστη τιμή δίχως 99.3 και μετά την εισαγωγή φορτίου 95.3)



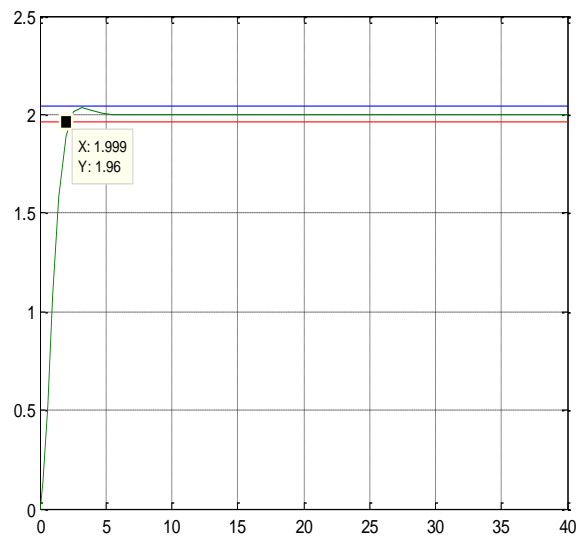
Το διάγραμμα του ρεύματος τυμπάνου από την εν κενό λειτουργία σε πλήρες φορτίο είναι το εξής:



Παρατηρούμε ότι ο κινητήρας χρειάζεται πολύ υψηλό ρεύμα εκκίνησης.

Προσθέσαμε δύο σήματα βηματικής τα οποία οριοθετούν το $\pm 2\%$ του χρόνου αποκατάστασης, ένα ρολόι ώστε να αποθηκευτεί ο χρόνος τον οποίον όπως και τα τρία σήματα στην έξοδο με την βαθμίδα *To Workspace* (στις επιλογές της βαθμίδας για το χρόνο ορίζουμε *variable name: t* και στην επιλογή *Save format* επιλέγουμε *Array* και για την έξοδο *variable name: y* και ομοίως στην επιλογή *Save format* επιλέγουμε *Array*) έχουμε τη δυνατότητα να τα εκμεταλλευτούμε στο περιβάλλον του MatLab. Δίνουμε τις παρακάτω εντολές και έχουμε:

```
plot(t,y);grid on;
```

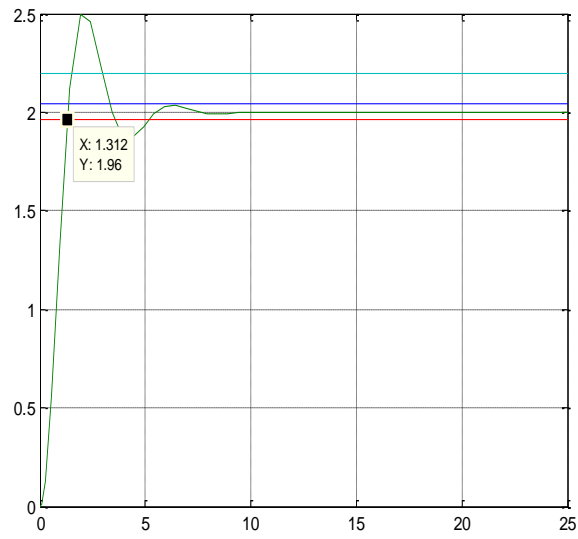


Ο χρόνος αποκατάστασης με κριτήριο 2% ισούται με 1.99 sec.

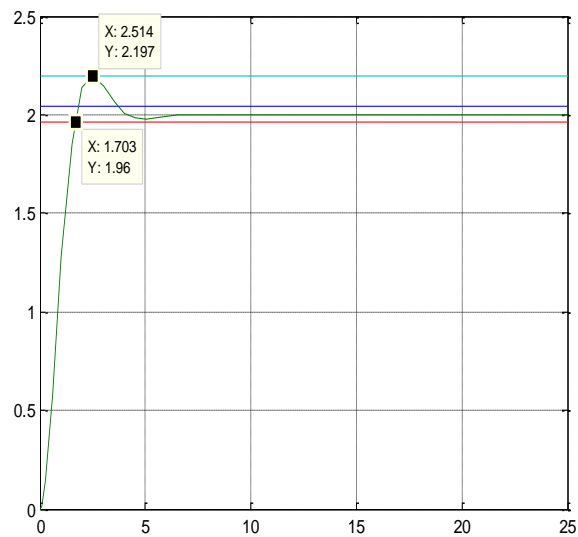
Με τον ίδιο τρόπο αλλά σε αυτή τη περίπτωση προσθέτουμε ένα σήμα βηματικής το οποίο οριοθετεί το 10% της μέγιστης υπερύψωσης στην τιμή 2.2 μεταβάλλουμε τις τιμές του K_p και του a μέχρι να πετύχουμε 10% μέγιστη υπερύψωση.

```
Kp=1000;a=0.05;  
plot(t,y);grid on;
```

Βλέπουμε ότι η απόκριση του συστήματος παρουσιάζει πάνω από 10 % μέγιστη υπερύψωση, οπότε οι τιμές των παραμέτρων δεν ήταν οι κατάλληλες.



Μετά από πειραματικές τιμές καταλήγουμε στο συμπέρασμα ότι οι τιμές που πληρούν τις προϋποθέσεις είναι $K_r=960$ και $\alpha=0.77$.



Βιβλιογραφία

1. Richard C. Dorf – Robert H. Bishop, «Σύγχρονα Συστήματα Αυτόματου Ελέγχου», 9^η έκδοση, εκδόσεις Τζιόλα.
2. Αναστασία Ν. Βελώνη, «Συστήματα Αυτόματου Ελέγχου - Λυμένες Ασκήσεις», εκδόσεις Παπασωτηρίου.
3. Σταύρος Βολογιαννίδης, «Συστήματα Αυτόματου Ελέγχου Θεωρία και Εφαρμογές».
4. Γιώργος Γεωργίου – Χρίστος Ξενοφώντος, «Εισαγωγή στη MATLAB».
5. Αριστοτέλης Γιαννακουδάκης, «Άσκηση Πράξης 1–4».
6. Μανώλης Καβουσιανός, «Συστήματα Αυτόματου Ελέγχου».
7. Δημήτρης Καλλιγερόπουλος, «Αναλογική Εξομοίωση Συστημάτων».
8. Νίκος Καραμπετάκης, «Εισαγωγή στα Συστήματα Αυτόματου Ελέγχου».
9. Παρασκευάς Ν. Παρασκευοπούλου, «Εισαγωγή στον Αυτόματο Έλεγχο, Τόμος Α : Θεωρία», Πρώτη Έκδοση Αθήνα 2001.
10. Μιχάλης Σφακιωτάκης, «Συστήματα Αυτόματου Ελέγχου II – Διαλέξεις Θεωρίας», Δεκέμβριος 2009.
11. Γεώργιος Φ. Φραγκούλης, «Εισαγωγή στο MATLAB – SIMULINK».