

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ



ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

1^Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

«ΣΥΣΤΗΜΑΤΑ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ»

6^ο Εξάμηνο

Συγγραφείς:

Φώτιος Γκιουλέκας, Διδάσκων Π.Δ. 407/80 ΤΜΗΥΠ

Κυριάκος Βλάχος, Επίκουρος Καθηγητής ΤΜΗΥΠ

Κωνσταντίνος Μπερμπερίδης, Καθηγητής ΤΜΗΥΠ

Πάτρα-2009

ΠΡΟΛΟΓΟΣ

Στα πλαίσια του Μαθήματος του 6^{ου} εξαμήνου «Συστήματα Μετάδοσης Πληροφορίας» θα πραγματοποιηθούν δύο (2) εργαστηριακές ασκήσεις. Στόχος των ασκήσεων αυτών αποτελεί η κατανόηση των βασικών αρχών που διέπουν τα ασύρματα και οπτικά τηλεπικοινωνιακά συστήματα καθώς επίσης και η εξοικείωση των φοιτητών/φοιτητριών με τα αντίστοιχα εμπορικά λογισμικά εξομοίωσης, τα οποία χρησιμοποιούνται για την σχεδίαση και μελέτη των συστημάτων αυτών.

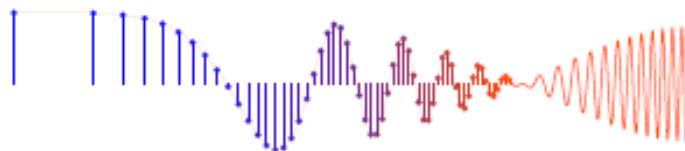
Επισημαίνεται ότι οι δύο αυτές εργαστηριακές ασκήσεις είναι προαιρετικές, βαθμολογούνται με άριστα το δύο (2) και προσμετρώνται θετικά στον τελικό βαθμό για τους φοιτητές/φοιτήτριες που θα τις παραδώσουν (π.χ. βαθμός εργαστηρίου = 1.5, βαθμός εξέτασης=5, τελικός βαθμός= 6.5).

Για την υλοποίηση των εργαστηριακών ασκήσεων οι φοιτητές/φοιτήτριες μπορούν να χρησιμοποιούν το υπολογιστικό κέντρο του τμήματος. Οι αναφορές για τις ασκήσεις αυτές είναι προσωπικές και όχι κατά ομάδες, δηλαδή ο κάθε φοιτητής/φοιτήτρια που επιθυμεί να πραγματοποιήσει τις ασκήσεις θα παραδώσει και την αντίστοιχη προσωπική αναφορά. Οι διδάσκοντες θα ενημερώσουν με σχετική ανακοίνωση για τις συγκεκριμένες ημέρες και ώρες που θα παραβρίσκονται στο υπολογιστικό κέντρο για την επίλυση αποριών.

Η πρώτη εργαστηριακή άσκηση περιλαμβάνει τη χρήση του μαθηματικού-αλγοριθμικού πακέτου MATLAB για την μελέτη των διαμορφώσεων/αποδιαμορφώσεων Πλάτους (AM) σε τηλεπικοινωνιακό πομποδέκτη.

Στα παραρτήματα Α και Β παρατίθενται αντίστοιχα το πρότυπο της αναφοράς για τις εργαστηριακές ασκήσεις και κάποιες χρήσιμες οδηγίες για προγραμματισμό στο περιβάλλον του MATLAB.

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1



Διαμόρφωση Πλάτους (AM)

1. Εισαγωγή

Η διαμόρφωση πλάτους AM (Amplitude Modulation) ανήκει στην κατηγορία των σχημάτων διαμόρφωσης αναλογικού σήματος πάνω από αναλογικό φέρον και συγκεκριμένα αφορά στη μεταβολή του πλάτους του φέροντος (carrier) συναρτήσει του σήματος πληροφορίας. Η διαμόρφωση AM διακρίνεται κυρίως σε διαμόρφωση πλάτους διπλής πλευρικής ζώνης με καταργημένο φέρον AMDSB-SC (Amplitude Modulation Double-Side Band Suppressed Carrier), Συμβατικό AM διπλής πλευρικής ζώνης (conventional AM double-sideband), AM μονής πλευρικής ζώνης (AM Single-Side Band), AM με κατάλοιπο πλευρικής ζώνης (AM Vestigial-Side Band). Σκοπός της πρώτης εργαστηριακής άσκησης είναι η κατανόηση της λειτουργίας ενός τηλεπικοινωνιακού συστήματος διαμόρφωσης που χρησιμοποιεί διαμόρφωση πλάτους AMDSB-SC και Συμβατικού AM με τη βοήθεια του MATLAB.

1.1. AMDSB-SC

Στη διαμόρφωση AMDSB-SC, το πλάτος του διαμορφωμένου σήματος είναι ανάλογο του πληροφοριακού σήματος. Η αναπαράσταση του σχήματος αυτού στο χρόνο δίνεται από την ακόλουθη εξίσωση:

$$y(t) = A_c x(t) \cos(2\pi f_c t)$$

όπου $A_c \cos(2\pi f_c t)$ είναι το φέρον σήμα συχνότητας f_c και $x(t)$ το ημιτονοειδές πληροφοριακό σήμα. Το εύρος ζώνης για τη μετάδοση του διαμορφωμένου σήματος είναι διπλάσιο του εύρους ζώνης του πληροφοριακού σήματος.

1.2. Συμβατικό AM

Η διαφορά μεταξύ της διαμόρφωσης συμβατικού AM και της AMDSB-SC έγκειται στο γεγονός ότι η πρώτη ενσωματώνει το καθαυτό φέρον. Το πληροφοριακό σήμα $x(t)$ αντικαθίσταται από τον παράγοντα $(1 + mx_n(t))$, όπου m είναι ο δείκτης διαμόρφωσης, $x_n(t)$ το κανονικοποιημένο ως προς το πλάτος του (x_0) πληροφοριακό σήμα ($x_n(t) = x(t)/x_0$). Επομένως, ο δείκτης διαμόρφωσης ορίζεται ως $m = x_0/A_c$. Στην περίπτωση αυτή το AM σήμα είναι:

$$y(t) = A_c (1 + mx_n(t)) \cos(2\pi f_c t)$$

Η παρουσία της ημιτονοειδούς συνιστώσας καθιστά το σχήμα διαμόρφωσης με συμβατικό AM λιγότερο οικονομικό όσον αφορά στην ισχύ μετάδοσης συγκρινόμενο με τη διαμόρφωση AMDSB-SC. Αντίθετα, η αποδιαμόρφωση συμβατικού AM

αποτελεί μια φθηνή λύση. Και τα δύο σχήματα διαμόρφωσης έχουν το ίδιο εύρος ζώνης μετάδοσης.

2. Προετοιμασία

2.1. Στο περιβάλλον του MATLAB, πληκτρολογήσατε τις παρακάτω γραμμές:

a. $X=[0,1,2,3,4,5]$

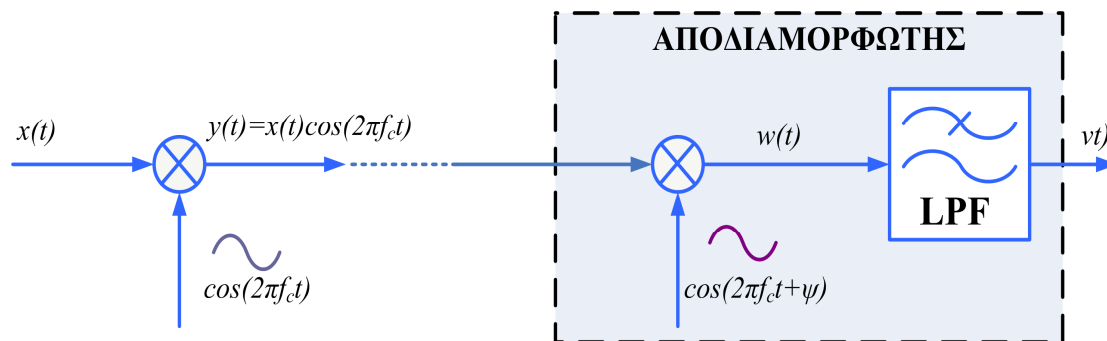
b. $Y=[1,2,3,4,5,6]$

Στη συνέχεια πολλαπλασιάστε το διάνυσμα X με το Y με δύο τρόπους. Ο πρώτος είναι με τη χρήση του τελεστή $*$ δηλαδή $(X*Y)$ ενώ ο δεύτερος με τη χρήση του τελεστή $.*$ δηλαδή $(X.*Y)$. Τι παρατηρείτε; Ποία είναι η διαφορά μεταξύ των δύο τελεστών;

2.2. Στο περιβάλλον του MATLAB, δημιουργήστε ένα διάνυσμα $t = [0:0.001:1]$. Έπειτα, παράγετε τα ημιτονοειδή σήματα $m = \cos(2*\pi*t)$ και $v = \cos(4*\pi*t)$, όπου $\pi=3.1416$. Με χρήση της εντολής `plot` εξάγατε τα γραφήματα για τα σήματα m , v και το γινόμενο τους $x=m*v$. Πληκτρολογήσατε την εντολή `figure` στην περίπτωση που θέλετε να δημιουργήσετε νέο γράφημα.

3. Εκτέλεση της άσκησης

3.1. Χρησιμοποιείστε το περιβάλλον του MATLAB για την εξομοίωση του παρακάτω τηλεπικοινωνιακού συστήματος AM δημιουργώντας το αρχείο `AMDSBSC_am.m` (όπου `am` είναι ο αριθμός μητρώου σας).



Σχήμα 1. Τηλεπικοινωνιακό σύστημα AM.

Θεωρήστε $\psi=0$ και $x(t)=\cos(2\pi 2000t)$ και χρησιμοποιήστε συχνότητα φέροντος $f_c=20\text{KHz}$. Ορίσατε το διάνυσμα χρόνου $t = [0:300]*t_s$, όπου t_s είναι το μέγεθος του βήματος της δειγματοληψίας για το οποίο ισχύει $t_s=1/f_s$. Η συχνότητα δειγματοληψίας f_s πρέπει να είναι τέτοια ώστε να ισχύει $f_s > 2(f_c + BW)$, όπου BW είναι το εύρος ζώνης του πληροφοριακού σήματος το οποίο υφίσταται διαμόρφωση. Θέσατε $f_s = 8 * f_c$. Για τη σχεδίαση του βαθυπερατού φίλτρου LPF (Low-Pass Filter) του δέκτη στο MATLAB χρησιμοποιήστε την κατηγορία των Butterworth φίλτρων τάξης n και συχνότητα αποκοπής w_o , η οποία ισούται με $f_c/f_s/2$ (όπου t_s είναι το βήμα της δειγματοληψίας που ορίσατε παραπάνω). Για την εξαγωγή των συντελεστών του φίλτρου εισάγεται στο πρόγραμμά σας την εντολή `[num,den]=butter(n, w_o)` με $n=12$. Οι μεταβλητές `num`, `den` υποδηλώνουν τους συντελεστές του αριθμητή και του παρανομαστή της ρητής συνάρτησης που περιγράφει το φίλτρο. Στη συνέχεια, με τη

χρήση της συνάρτησης *filter* φιλτράρεται το σήμα $w(t)$ μέσω το φίλτρο που έχετε σχεδιάσει ($v = filter(num, den, w)$).

Δημιουργήστε τα γραφήματα $x(t)$, $y(t)$, $w(t)$ και $v(t)$ και το αντίστοιχο μέτρο του φάσματός τους (τέσσερα διαφορετικά γραφήματα, όπου το καθένα θα αποτελείται από δύο τμήματα στα οποία θα φαίνεται το σήμα και το μέτρο του φάσματός του). Η συνάρτηση *fftshift()* είναι χρήσιμη για την απεικόνιση του *fft* με συχνότητα μηδενικής συχνότητας στο μέσο του φάσματος σε συνδυασμό με την *linspace*. Η συνάρτηση *freqz(num,den)* δείχνει την απόκριση του φίλτρου.

Χρήσιμες συναρτήσεις του MATLAB: *cos*, *fft()*, *fftshift()*, *butter*, *filter*, *abs*, *plot*, *subplot*, *figure*, *linspace*, *xlabel*, *ylabel*, *title*, *grid on*.

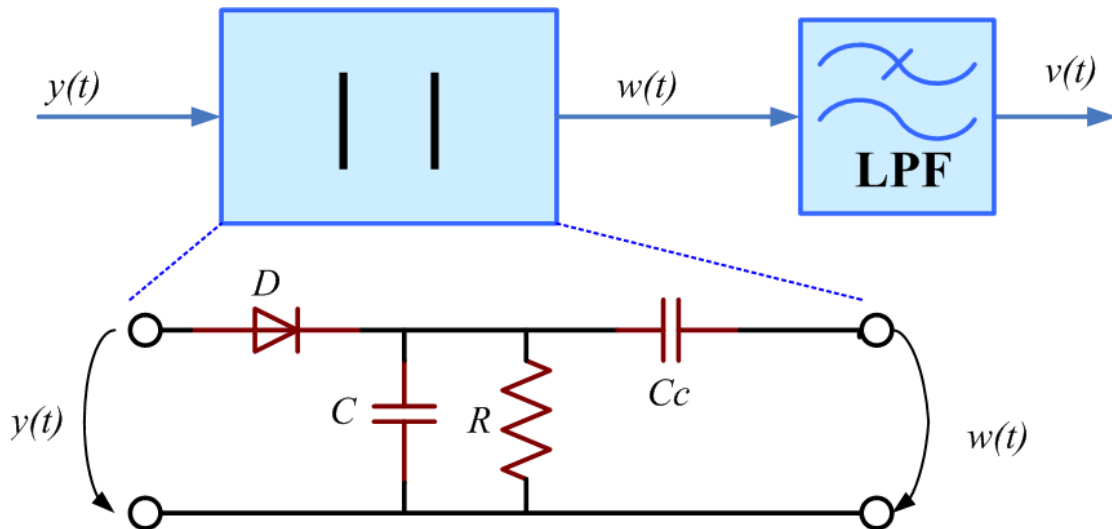
3.2. Επαναλάβετε το μέρος 3.1 της άσκησης χρησιμοποιώντας τις εξής τιμές για τη μετατόπιση της φάσης ψ στο δέκτη: $\psi = \pi/2$ και π .

Τί παρατηρείτε στην έξοδο του δέκτη; Υπάρχει κάποια διαφορά σε σχέση με το σήμα εξόδου του δέκτη στο μέρος 3.1 της άσκησης; Ποιά είναι η λύση στο πρόβλημα;

3.3. Επαναλάβετε το μέρος 3.1 της άσκησης δημιουργώντας διαμόρφωση Συμβατικού AM (δημιουργώντας το **AM_am.m** αρχείο), δηλαδή $y(t) = A_c(1 + mx_n(t))\cos(2\pi f_c t)$ θέτοντας $A_c = 4$ και $m = 0.5$ (Το πλάτος του πληροφοριακού σήματος είναι $x_0 = 1$). Τί παρατηρείτε από τις γραφικές παραστάσεις; Ποιές είναι διαφορές μεταξύ AMDSB-SC και Συμβατικού AM; Εξηγήστε την επίδραση του δείκτη διαμόρφωσης στο σύστημα με Συμβατικό AM χρησιμοποιώντας m της επιλογής σας.

3.4. Επαναλάβετε το τμήμα 3.3 της άσκησης χρησιμοποιώντας τώρα ως αποδιαμορφωτή το σύστημα που απεικονίζεται στο ακόλουθο σχήμα (δημιουργώντας το αρχείο **AMenvelope_am.m**). Πρόκειται για έναν απλό ανιχνευτή περιβάλλουσας, ο οποίος περιλαμβάνει μία δίοδο, έναν πυκνωτή και μία αντίσταση και στη συνέχεια έναν πυκνωτή σύζευξης και ένα βαθυπερατό φίλτρο (LPF). Αυτή η απλή μορφή ανιχνευτή περιβάλλουσας είναι ένα μη γραμμικό κύκλωμα με γρήγορο χρόνο φόρτισης και αργό χρόνο αποφόρτισης. Προσοχή, η σταθερά χρόνου RC θα πρέπει να είναι τέτοια ώστε η μέγιστη αρνητική κλίση της περιβάλλουσας να μην ξεπερνά τον εκθετικό χρόνο εκφόρτισης δηλαδή $1/f_c \ll RC \ll 1/f_m$ (f_m η συχνότητα του πληροφοριακού σήματος). Χρησιμοποιήστε την συνάρτηση απόλυτης τιμής (*abs*) για την πλήρη ανόρθωση (ο πυκνωτής εκφορτίζεται λόγω της αντίστασης R κατά $v_c(t) = Ee^{-t/RC}$ τη στιγμή που δεν άγει η δίοδος, θεωρήστε $t = t_s$), προσθέστε τον κατάλληλο μηχανισμό για την απομάκρυνση της συνεχής (DC) συνιστώσας από το σήμα λήψης $y(t)$ και πραγματοποιήστε την αποδιαμόρφωση αφού θέσατε πρώτα την κατάλληλη συχνότητα αποκοπής w_o για το LPF φίλτρο.

Ποίος είναι ο ρόλος του πυκνωτή σύζευξης C_c του σχήματος 2; Σε ποια κατηγορία ανήκει η AM αποδιαμόρφωση με χρήση ανιχνευτή περιβάλλουσας; Τί πρέπει να ισχύει έτσι ώστε ο αποδιαμορφωτής περιβάλλουσας να μπορεί να αναζητήσει το σήμα πληροφορίας στις μεταβολές τις περιβάλλουσας;



Σχήμα 2. Αποδιαμόρφωση βάσει ανιχνευτή περιβάλλουσας με χρήση ανορθωτικής διάταξης και βαθυπερατού φίλτρου.

Χρήσιμες συναρτήσεις του MATLAB: *abs*, *mean*, *fft()*, *fftshift()*.

3.5. Χρησιμοποιήστε τον παρακάτω κώδικα για να δημιουργήσετε ένα ηχητικό μήνυμα *ms* διάρκειας 0.8 sec, το οποίο έχει δειγματοληφθεί με συχνότητα *fsampling* = 120000Hz ($t_{\text{sampling}} = 1/\text{fsampling}$ sec) αφού πρώτα δημιουργήσετε το αρχείο **AMtransmission_am.m**.

```
fsampling = 120000;
tsampling = 1/fsampling;
tduration = 0.8;
t = 0 : tsampling : tduration; % t είναι το διάνυσμα που αντιπροσωπεύει τον χρόνο
psi = 2*pi*(100 + 200*t + 500*t.*t);
ms = real( 7.7*exp(j*psi) );
```

Μεταδώστε και λάβετε το μήνυμα *ms* χρησιμοποιώντας το AM σύστημα του τμήματος 3.4 με συχνότητα φέροντος $f_c=10\text{kHz}$, $A_c=6$ και $m=0.2$. Προσοχή στην τιμή RC που θα δώσετε.

Στη συνέχεια ακούστε το μήνυμα *ms* χρησιμοποιώντας την εντολή *sound(ms, fsampling)*. Επίσης, ακούστε το ληφθέν σήμα *v* by typing *sound(v, fsampling)*. Τί παρατηρείτε για τα δύο σήματα; Τί θα συμβεί αν χρησιμοποιείτε $m=1.2$;

Για κάθε μία από τις δύο αυτές περιπτώσεις ($m=0.2$ και $m=1.2$) δημιουργήστε δύο γραφήματα με δύο τμήματα. Στο κάθε γράφημα απεικονίστε το διαμορφωμένο σήμα *y(t)* συναρτήσει του χρόνου και της συχνότητας (μέτρο φάσματος).

Χρήσιμες συναρτήσεις του MATLAB: *sound*.

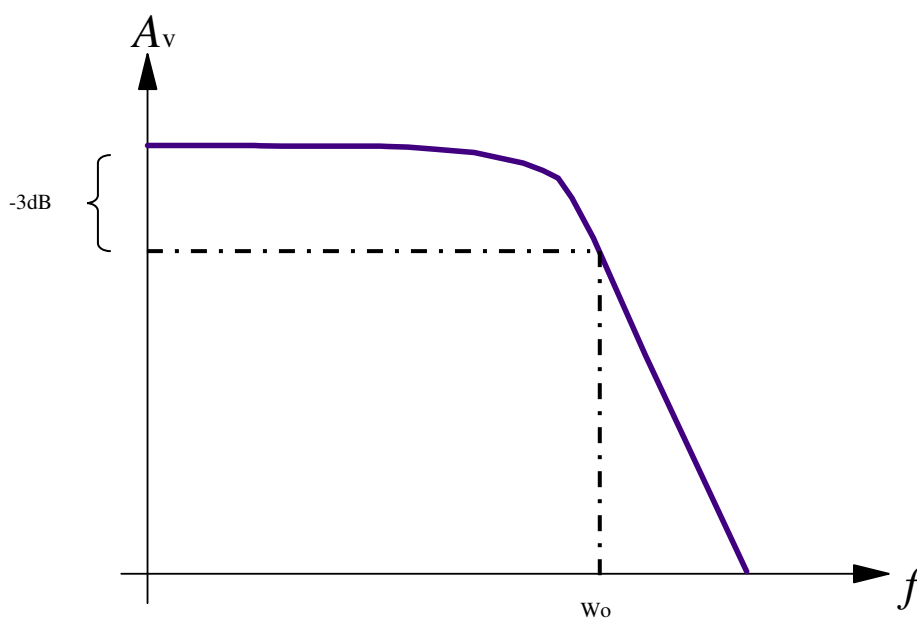
4. Γενικές σημειώσεις για φίλτρα στο MATLAB

Για την καλύτερη κατανόηση της χρήσης των φίλτρων με τη βοήθεια του Matlab παρατίθεται ο μετασχηματισμός Laplace της ρητής συνάρτησης ενός φίλτρου

$$H(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

συναρτήσει της ανεξάρτητης μεταβλητής s :

Οι συντελεστές a_n και b_n ορίζουν τους συντελεστές της συνάρτησης μεταφοράς του φίλτρου. Οι συντελεστές αυτοί χαρακτηρίζουν πλήρως την απόκριση του φίλτρου. Το Matlab, χρησιμοποιώντας συγκεκριμένη συνάρτηση σχεδίασης φίλτρου για ορισμένη συχνότητα αποκοπής, επιστρέφει τους συντελεστές αυτούς με τη μορφή δύο διανυσμάτων. Στην περίπτωση της άσκησης, χρησιμοποιήσατε το φίλτρο Butterworth, με $num=[b_m, b_{m-1}, \dots, b_0]$ και $den=[a_n, a_{n-1}, \dots, a_0]$ και το οποίο σχεδιάζετε με την εισαγωγή της εντολής $[num, den]=butter(n, fc/fs/2)$. Είναι σημαντικό να ορίσετε την τάξη n του φίλτρου. Μεγαλύτερη τάξη οδηγεί σε καλύτερη απόδοση αλλά απαιτεί περισσότερους υπολογισμούς. Η συχνότητα w_0 είναι κανονικοποιημένη ως προς $fc/fs/2$ για να ανήκει στο διάστημα $(0,1)$ όπως απαιτείται από τη συνάρτηση *butter*.



Σχήμα 3. Συνάρτηση μεταφοράς φίλτρου LPF τύπου Butterworth όπου w_0 είναι η συχνότητα αποκοπής.

Παράρτημα Α

Πρότυπο Αναφοράς

Α. Εξώφυλλο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

«ΣΥΣΤΗΜΑΤΑ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ»

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο #
«Όνομα Άσκησης»

«Όνοματεπώνυμο»
«Αριθμός Μητρώου»

«Ημερομηνία Παράδοσης»

Β. Δομή

1. Σύντομη περιγραφή του σκοπού της άσκησης.
2. Απαντήσεις στα ερωτήματα της άσκησης. Ενσωματώνετε γραφικές παραστάσεις, προσθέτετε τον κώδικα του Matlab και παραθέτετε τα αποτελέσματα και παρατηρήσεις.
3. Συμπεράσματα. Δώστε ένα σύντομο επίλογο περιγράφοντας αυτά που μάθατε πραγματοποιώντας την άσκηση.

Παρατήρηση: Μην περιγράφετε την άσκηση στην αναφορά βάσει του εντύπου που σας έχει δοθεί αλλά δώστε ακριβείς απαντήσεις στα ερωτήματα !

Παράρτημα Β

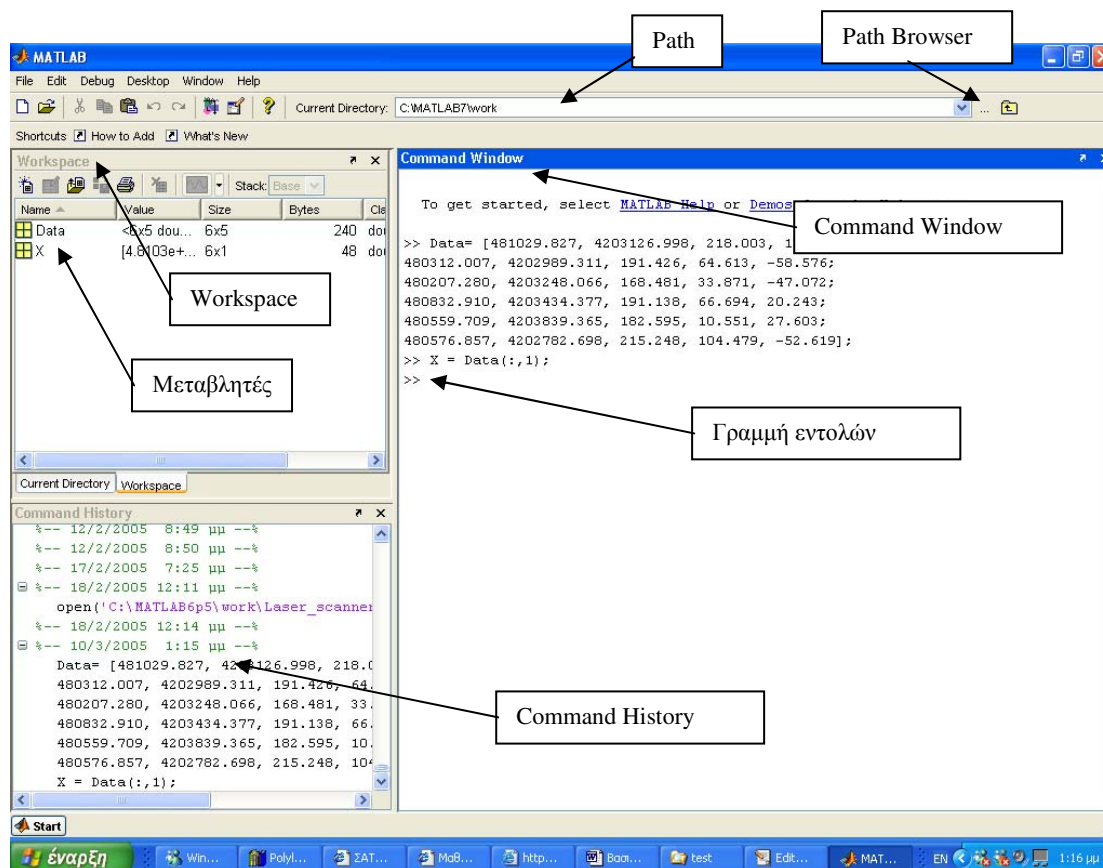
Βασικές οδηγίες για προγραμματισμό σε MATLAB (<http://www.mathworks.com>)

Εισαγωγή

Το Matlab, είναι μία γλώσσα προγραμματισμού η οποία είναι προσαρμοσμένη έτσι ώστε να διαχειρίζεται πίνακες και διανύσματα. Στη συνέχεια, σας δίνονται κάποιες βασικές εντολές τις οποίες μπορείτε να χρησιμοποιήσετε προκειμένου τα προγράμματα που δημιουργείτε να διαχειρίζονται τα δεδομένα με μορφή πινάκων έτσι ώστε η χρήση του Matlab να είναι περισσότερο αποδοτική.

1. Περιβάλλον του MATLAB

Το περιβάλλον εργασίας του Matlab περιλαμβάνει κάποια παράθυρα, τα οποία είναι: το Command window (η γραμμή εντολών), το Workspace (ο χώρος εργασίας στο οποίο αποθηκεύονται όλες οι μεταβλητές και οι πίνακες που δημιουργούνται) και το Command History (όπου φαίνονται οι εντολές που εκτελέστηκαν από όταν έγινε η εκκίνηση του προγράμματος, αλλά και εντολές που δόθηκαν κατά τη χρήση του προγράμματος τις προηγούμενες φορές). Επίσης, στο επάνω μέρος φαίνεται η γραμμή του Path που δείχνει τη διαδρομή του directory που βλέπει το Matlab, καθώς και κουμπί που οδηγεί στον Path Browser. Εκτός από τα παράθυρα αυτά υπάρχει και ο editor του Matlab, όπου γίνεται η συγγραφή του κώδικα των προγραμμάτων.



Το Matlab προσφέρει δύο εναλλακτικούς τρόπους εργασίας. Είτε μέσω του Command Window όπου ο χρήστης δίνει μεμονωμένες εντολές, μπορεί να κάνει κάποιες δοκιμές και παίρνει άμεσα κάποια αποτελέσματα, είτε μέσω scripts, προγραμμάτων δηλαδή, τα οποία ο χρήστης γράφει στο περιβάλλον του editor. Τα scripts αποτελούνται από σειρές εντολών, ακριβώς όπως και τα προγράμματα που γράφονται σε διάφορες άλλες γλώσσες προγραμματισμού, και τα οποία εκτελούνται είτε απευθείας από τον editor δίνοντας την εντολή Run είτε από το Command Window γράφοντας το όνομα του script. Τα scripts είναι τα αρχεία κώδικα που αποθηκεύονται με το extension (την κατάληξη) .m και η αποθήκευση γίνεται συνήθως στο αρχείο **work** μέσα στα directories του Matlab καθώς η τοποθεσία αυτή είναι προεπιλεγμένη. Ωστόσο ο χρήστης μπορεί να δουλέψει από οποιαδήποτε τοποθεσία εφόσον ορίσει το path στο οποίο το Matlab θα ψάχνει τα αρχεία που χρειάζεται κατά την εκτέλεση ενός προγράμματος. Ο ορισμός του path μπορεί να γίνει είτε γράφοντας το Path στην γραμμή που φαίνεται στο επάνω μέρος της οθόνης, είτε μέσω του Path Browser. Σε κάθε περίπτωση, όλα τα αρχεία που είναι απαραίτητα για τη λειτουργία ενός προγράμματος πρέπει να είναι αποθηκευμένα σε ένα και μόνο directory και το συγκεκριμένο path να έχει οριστεί στο Matlab.

2. Βασικές εντολές στο MATLAB

2.1. Η εντολή clear all

Η εντολή clear all σβήνει όλα τα περιεχόμενα του χώρου εργασίας (workspace).

2.2. Ορισμός πίνακα:

```
A= [1 2 3 4 5 6 7;  
    8 9 10 11 12 13;  
    14 15 16 17 18];
```

ή εναλλακτικά και εφόσον τα δεδομένα είναι αποθηκευμένα σε ένα αρχείο μπορούμε να κάνουμε

2.3. Διαμόρφωση της εξόδου στο Command Window

Συνήθως το Matlab εμφανίζει πολύ μεγάλους ή πολύ μικρού αριθμούς με επιστημονική μορφοποίηση (scientific format) πχ ο αριθμός 481029.827 θα εμφανιστεί ως 4.8103e+005, δηλαδή σαν δεκαδικός με 5 σημαντικά ψηφία επί κάποια δύναμη του 10, ενώ ο αριθμός που έχουμε δώσει εμείς έχει 9 σημαντικά ψηφία. Η πληροφορία αυτή διατηρείται στο Matlab. Ωστόσο ο συγκεκριμένος τρόπος αναπαράστασης δεν βοηθάει πάντα. Συνεπώς, όταν θέλουμε να βλέπουμε όλα τα σημαντικά ψηφία των αριθμών που χρησιμοποιούμε μπορούμε να δώσουμε στο Command Window την εντολή:

```
format long g
```

2.4. Προσδιορισμός του μεγέθους ενός πίνακα

Έστω ότι έχουμε εισάγει τον πίνακα Data και θέλουμε να ξέρουμε τις διαστάσεις του. Η εντολή size δίνει αυτή τη δυνατότητα:

`[r, c] = size(Data);` (Αποθηκεύει στη μεταβλητή `r` τον αριθμό των γραμμών και στην `c` τον αριθμό των στηλών)

`r = size(Data, 1);` (Αποθηκεύει στη μεταβλητή `r` τον αριθμό των γραμμών)

`c = size(Data, 2);` (Αποθηκεύει στη μεταβλητή `c` τον αριθμό των στηλών)

Η εντολή `r = size(Data,1)` είναι ισοδύναμη με την εντολή `r = length(Data)` για πίνακες, ενώ για διανύσματα, η εντολή αυτή δίνει τον αριθμό των στοιχείων.

2.5. Δυναμικός ορισμός πινάκων

Το Matlab, παρέχει κάποιες συναρτήσεις για τον ορισμό ορισμένων πινάκων.

`A = zeros(m, n);`

δημιουργεί έναν πίνακα διαστάσεων $m \times n$ με μηδενικά στοιχεία

`A = ones (m, n);`

δημιουργεί έναν πίνακα διαστάσεων $m \times n$ με μοναδιαία στοιχεία

`A = diag (n);`

δημιουργεί το διαγώνιο $n \times n$ πίνακα `I`

`A = zeros (size(Data));`

Δημιουργεί έναν πίνακα που έχει τις διαστάσεις του `Data` και μηδενικά στοιχεία

2.6. Επιλογή του στοιχείου (m,n) ενός πίνακα και αποθήκευση στη μεταβλητή d

`d= Data(m, n);`

πχ. αν θέλουμε το 3^ο στοιχείο της 2^{ης} γραμμής του πίνακα `Data` γράφουμε

`d = Data(2, 3);` (Που μας δίνει `d = 191.426`)

2.7. Επιλογή της στήλης n ενός πίνακα

`X= Data(:, n);`

πχ. αν θέλουμε την 1^η στήλη του πίνακα γράφουμε

`X = Data(:, 1);`

2.8. Επιλογή της γραμμής m ενός πίνακα

`Rm= Data (m, :);`

πχ. αν θέλουμε την 1^η γραμμή του πίνακα γράφουμε

`R1 = Data(1, :);`

2.9. Επιλογή υποπίνακα

Πχ. αν θέλουμε τις τρεις πρώτες στήλες του πίνακα `Data` δίνουμε την εντολή,

`C3 = Data (:, 1:3);`

Πχ. αν θέλουμε τις τρεις πρώτες γραμμές του πίνακα `Data` δίνουμε την εντολή,

`R3 = Data (1:3, :);`

Πχ. αν θέλουμε τα τρία πρώτα στοιχεία των τριών πρώτων γραμμών του πίνακα Data δίνουμε την εντολή,
`C33 = Data(1:3, 1:3);`

2.10. Σύνθεση πινάκων

Έστω A πίνακας ($m \times n_1$) και B πίνακας ($m \times n_2$)

`C = [A B];`

δίνει πίνακα C διαστάσεων ($m \times (n_1+n_2)$) όπου οι n_1 πρώτες στήλες είναι τα στοιχεία του πίνακα A και οι επόμενες n_2 στήλες είναι τα στοιχεία του πίνακα B

Έστω A πίνακας ($m_1 \times n$) και B πίνακας ($m_2 \times n$)

`C = [A;B];`

δίνει πίνακα C διαστάσεων ($(m_1+m_2) \times n$) όπου οι m_1 πρώτες γραμμές είναι τα στοιχεία του πίνακα A και οι επόμενες m_2 γραμμές είναι τα στοιχεία του πίνακα B

2.11. Πράξεις μεταξύ στοιχείου και πίνακα (ή μεταξύ στοιχείου και διανύσματος)

Έστω ο πίνακας Data που ορίσαμε και το στοιχείο $b=2$;

`A = Data*b;`

πολλαπλασιάζει όλα τα στοιχεία του πίνακα Data με τον αριθμό b

`A = b*Data;`

το ίδιο όπως και η προηγούμενη εντολή

`A = b + Data;`

προσθέτει σε όλα τα στοιχεία του πίνακα Data τον αριθμό b

`A = Data + b;`

το ίδιο όπως και η προηγούμενη εντολή

`A = Data - b;`

αφαιρεί από όλα τα στοιχεία του πίνακα Data τον αριθμό b

`A = b - Data;`

αφαιρεί όλα τα στοιχεία του πίνακα Data από τον αριθμό b

`A = Data/b;`

διαίρει όλα τα στοιχεία του πίνακα Data με τον αριθμό b

`A = b./Data;`

διαίρει τον αριθμό b με όλα τα στοιχεία του πίνακα Data

*(Οι εντολές αυτές χρησιμοποιούνται με τον ίδιο ακριβώς τρόπο αν αντί για πίνακα έχουμε διάνυσμα είτε στήλης πχ. το X είτε γραμμής πχ το P1)

2.12. Πράξεις μεταξύ πινάκων (ή διανυσμάτων) ίδιων διαστάσεων

Έστω δύο πίνακες ή δύο διανύσματα ίσων διαστάσεων X και Y.

πχ $X = \text{Data}(:,1)$ και $Y = \text{Data}(:,2)$ (οπότε έχουμε δύο διανύσματα στήλες)

$A = X + Y;$

προσθέτει τους πίνακες X και Y στοιχείο προς στοιχείο, δηλαδή προσθέτει στο πρώτο στοιχείο του X το πρώτο του Y κλπ

$A = X - Y;$

Αφαιρεί τους πίνακες X και Y στοιχείο προς στοιχείο, δηλαδή από το πρώτο στοιχείο του X αφαιρεί το πρώτο του Y κλπ

$A = X .* Y;$

πολλαπλασιάζει τους πίνακες X και Y στοιχείο προς στοιχείο, δηλαδή πολλαπλασιάζει το πρώτο στοιχείο του X με το πρώτο του Y κλπ

$A = X ./ Y;$

διαίρει τους πίνακες X και Y στοιχείο προς στοιχείο, δηλαδή διαίρει το πρώτο στοιχείο του X με το πρώτο του Y κλπ

2.13. Πολλαπλασιασμός πίνακα με πίνακα

Έστω ένας πίνακας A ($m \times n$) και ένας πίνακας B ($n \times k$):

$C = A * B;$

εκτελεί τον πολλαπλασιασμό μεταξύ των δύο πινάκων

2.14. Ύψωση σε δύναμη

Έστω $b = 2;$

$c = b^n;$

δίνει τη μεταβλητή b υψωμένη εις τη n

$c = X.^n;$

δίνει έναν πίνακα που περιλαμβάνει τα στοιχεία του X υψωμένα εις τη n

ΠΡΟΣΟΧΗ

Για τετραγωνικούς πίνακες η πράξη $C = A^n$, πχ $C = A^3$ ισοδυναμεί με την $C = A * A * A$

2.15. Αναστροφή πίνακα

Ο πίνακας A' είναι ο ανάστροφος του πίνακα A .

2.16. Αντιστροφή πίνακα:

Έστω ο αντιστρέψιμος τετραγωνικός πίνακας A ,

$B = \text{inv}(A);$

δίνει τον αντίστροφο του A

2.17. Τριγωνομετρικές συναρτήσεις

Οι συναρτήσεις \cos , \sin , \tan και \cot ορίζονται έτσι ώστε να δουλεύουν τόσο με στοιχεία όσο και με πίνακες. Οι γωνίες που χρησιμοποιούνται για τους υπολογισμούς υπενθυμίζεται ότι πρέπει να είναι σε ακτίνια (rad). Επίσης, σημειώνεται ότι οι αντίστροφες συναρτήσεις acos , asin , atan και acot επιστρέφουν την τιμή της γωνίας σε ακτίνια, (προσοχή στα διαστήματα στα οποία επιστρέφονται οι τιμές, καθώς χρειάζεται διερεύνηση για τον προσδιορισμό της γωνίας). Επίσης, στο Matlab ορίζεται ο αριθμός π ως pi ίσος προς 3.14159265358979

Έστω $f = \text{pi}$ (ακτίνια),
 $a = \cos(f)$; (δίνει $a = -1$)

2.18. Δομές επιλογής

Υπάρχει περίπτωση κάποιες εντολές να θέλουμε ή να πρέπει να εκτελούνται υπό συνθήκη. Για το λόγο αυτό, χρειάζεται κάποια δομή επιλογής. Η Matlab, έχει για το σκοπό αυτό τις εντολές if , else , elseif και την εντολή end προκειμένου να εντοπίζεται το τέλος του μπλοκ των εντολών που εκτελούνται υπό συνθήκη.

Με τις εντολές αυτές μπορούμε να σχηματίζουμε μπλοκ διαφόρων μορφών

<pre>if (πρόταση) end</pre>	<pre>if (πρόταση) ... else ... end</pre>	<pre>if (πρόταση) ... elseif (πρόταση) ... elseif (πρόταση) ... end</pre>	<pre>if (πρόταση) ... elseif (πρόταση) ... elseif (πρόταση) ... else ... end</pre>
(1)	(2)	(3)	(4)

- (1) Όταν ενδιαφέρει η εκτέλεση των εντολών σε μία μόνο περίπτωση (δεν καλύπτονται όλες οι περιπτώσεις)
- (2) Όταν διακρίνουμε δύο ενδεχόμενα (καλύπτονται όλες οι περιπτώσεις)
- (3) Όταν ενδιαφέρει η εκτέλεση των εντολών σε n περιπτώσεις (δεν καλύπτονται όλες οι περιπτώσεις)
- (4) Όταν διακρίνουμε n ενδεχόμενα (καλύπτονται όλες οι περιπτώσεις)

2.19. Δομή επανάληψης

Το Matlab για την πραγματοποίηση επαναλήψεων προσφέρει δύο δομές, τη while και τη for . Η δομή for προτιμάται όταν πρόκειται να πραγματοποιηθεί ένας συγκεκριμένος αριθμός επαναλήψεων, πχ όταν θέλουμε να βρούμε το γινόμενο των στοιχείων ενός διανύσματος X με n στοιχεία:

```
G=1;
for i=1:n
    G = G*X(i);
```



```
end
```

πχ. έστω ότι θέλουμε να κάνουμε το ίδιο για έναν πίνακα A m x n:

```
G=1;  
for i=1:m  
    for j=1:n  
        G = G*A(m,j);  
    end  
end
```

Πχ. έστω ότι θέλουμε να δημιουργήσουμε έναν πίνακα και να δώσουμε τιμές στα στοιχεία του. Για παράδειγμα, δίνεται ο τρόπος της δημιουργίας του n x n πίνακα Hilbert τα στοιχεία του οποίου δίνονται από τη σχέση $A(i, j) = 1/(i+j-1)$;

```
A = zeros(n,n) % Preallocate matrix  
for i = 1:n  
    for j = 1:n  
        A(i,j) = 1/(i+j-1);  
    end  
end
```

Η επαναληπτική δομή while μπορεί να χρησιμοποιηθεί όταν είναι άγνωστος ο αριθμός των επαναλήψεων που πρέπει να πραγματοποιηθούν. Η while χρησιμοποιείται πάντα σε συνδυασμό με μία πρόταση η οποία ελέγχεται. Αν η πρόταση είναι αληθής, τότε εκτελείται το μπλοκ των εντολών που περιλαμβάνονται μεταξύ της while και της end, η οποία σημειώνει το τέλος του μπλοκ.

```
while (πρόταση)
```

```
...
```

```
end
```

Η πρόταση μπορεί περιλαμβάνει για παράδειγμα τον έλεγχο ενός κριτηρίου σύγκλισης πχ while (x > 1). Αυτό σημαίνει ότι όσο η ποσότητα x είναι μεγαλύτερη του 1, θα πραγματοποιούνται επαναλήψεις. Όταν το x γίνει μικρότερο ή ίσο με το 1 τότε οι επαναλήψεις θα σταματήσουν και το πρόγραμμα θα συνεχίσει με την εκτέλεση του υπόλοιπου κώδικα. Συνήθως, για να διασφαλίζεται το ότι το πρόγραμμα δεν θα πέσει σε κύκλο άπειρων επαναλήψεων, δίνουμε μία σύνθετη πρόταση η οποία περιλαμβάνει και ένα μετρητή. Πχ.

```
c= 1;  
while ((x>1) && (c<=5))  
    ...  
    ...  
    ...  
    x =...;  
    c=c+1;  
end
```