



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ψηφιακή Επεξεργασία Σήματος

Ροή Σ: Σήματα, Έλεγχος, Ρομποτική

6^ο Εξάμηνο

2^η Εργαστηριακή Άσκηση

*Επεξεργασία και Κατηγοριοποίηση Σημάτων
Φυσιολογίας και Κίνησης από Φορητούς
Αισθητήρες*

Χρήστος Δημόπουλος – 031 17 037

Δημήτρης Δήμος – 031 17 165

Μέρος 1^ο – Υπολογισμός Ενέργειας Βραχέος Χρόνου και Μετ/σμού Fourier Βραχέος Χρόνου (STFT) σε σήματα κίνησης και καρδιακού ρυθμού

Σκοπός Άσκησης

Στο πρώτο μέρος της εργαστηριακής άσκησης θα χρησιμοποιήσουμε μεθόδους που μάθαμε από το πρώτο εργαστήριο, καθώς και τεχνικές επεξεργασίας σήματος και μηχανικής μάθησης, για την επεξεργασία και ανάλυση σημάτων φυσιολογίας και κίνησης, τα οποία έχουν ληφθεί με χρήση ενός έξυπνου ρολογιού Samsung Gear S3 Frontier. Ειδικότερα, πρόκειται να επεξεργαστούμε δεδομένα που έχουν ληφθεί από ένα άτομο που περπατά για διάρκεια περίπου 10 λεπτών.

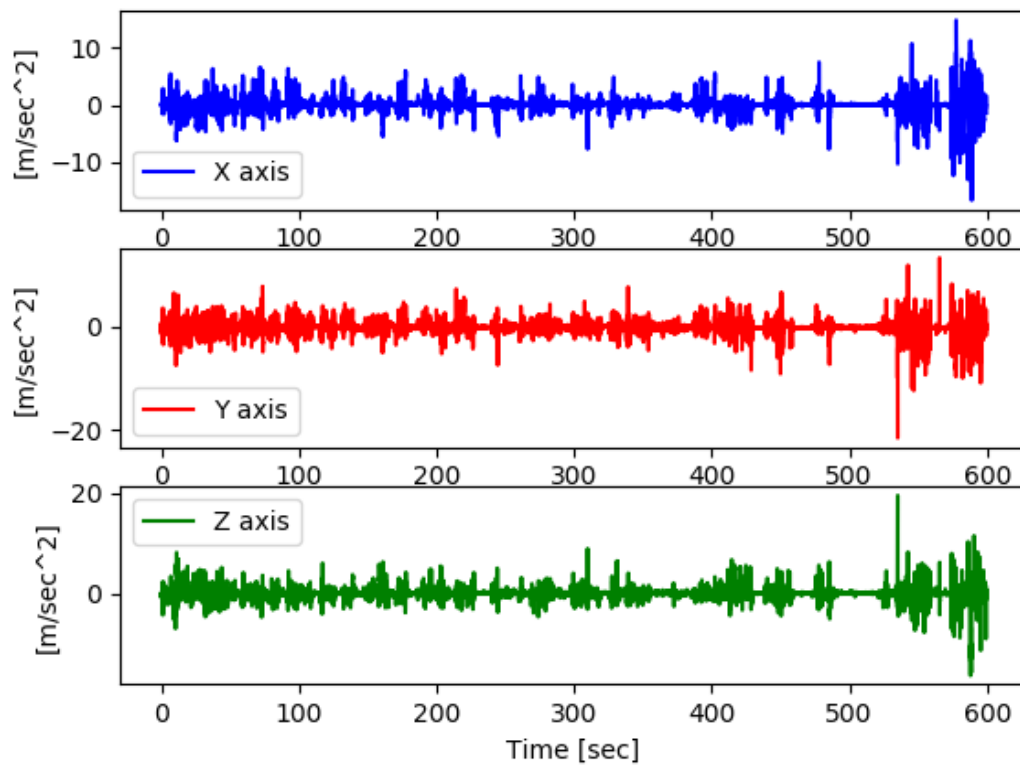
1.1 Φόρτωση αρχείου “step_00.npz”

Αρχικά, φορτώνουμε το αρχείο “step_00.npz”, με χρήση της συνάρτησης `load()` της βιβλιοθήκης `numpy`. Το αρχείο αυτό μας δίνει τα εξής σήματα:

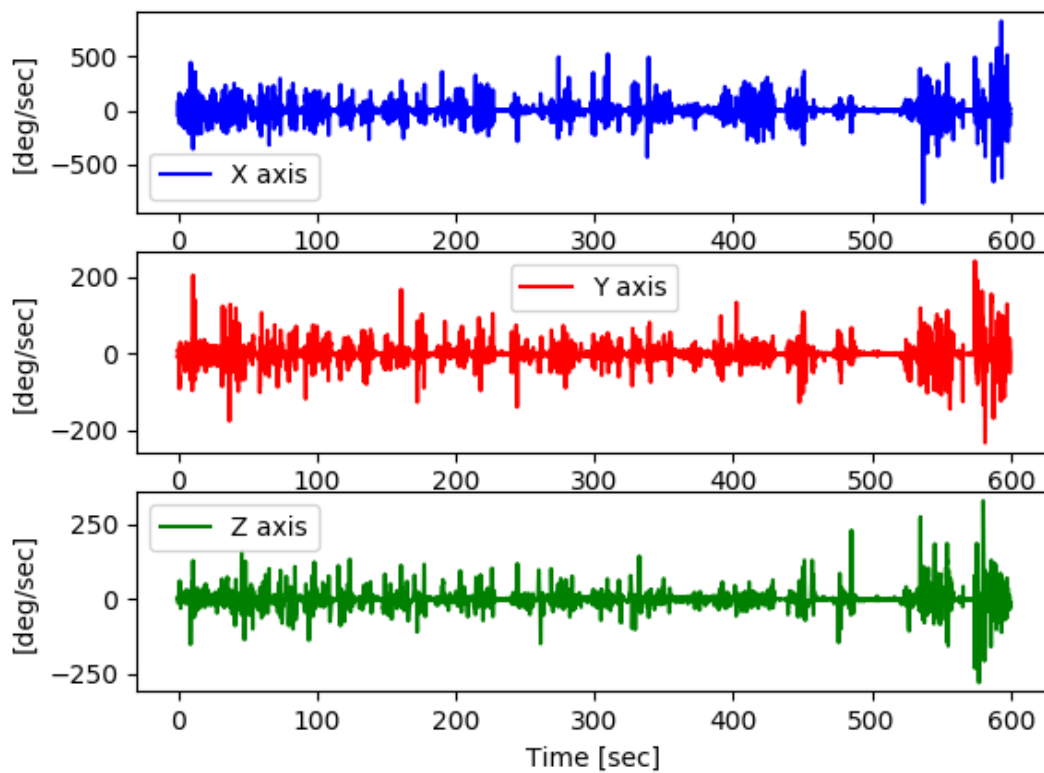
- Το πολυδιάστατο σήμα `acc` με συχνότητα δειγματοληψίας $fs1 = 20\text{Hz}$, το οποίο αντιστοιχεί στη **γραμμική επιτάχυνση** (m/s^2) που μετράει το επιταχυνσιόμετρο σε τρεις άξονες X,Y και Z (σήματα `accx`, `accy` και `accz` αντιστοίχως).
- Το πολυδιάστατο σήμα `gyr` με συχνότητα δειγματοληψίας $fs1 = 20\text{Hz}$, το οποίο αντιστοιχεί στη **γωνιακή ταχύτητα** (μοίρες/s) που μετράει το γυροσκόπιο σε τρεις άξονες X,Y και Z (σήματα `gyrx`, `gyry` και `gyrz` αντιστοίχως).
- Το μονοδιάστατο σήμα `hrm` το οποίο περιέχει τη **μεταβλητότητα καρδιακών παλμών** (heart rate variability), δηλαδή την ακολουθία των χρονικών διαστημάτων (σε ms) που μεσολαβούν από τον ένα χτύπο της καρδιάς έως τον επόμενο. Η συχνότητα δειγματοληψίας του εν λόγω σήματος είναι $fs2 = 5\text{Hz}$.

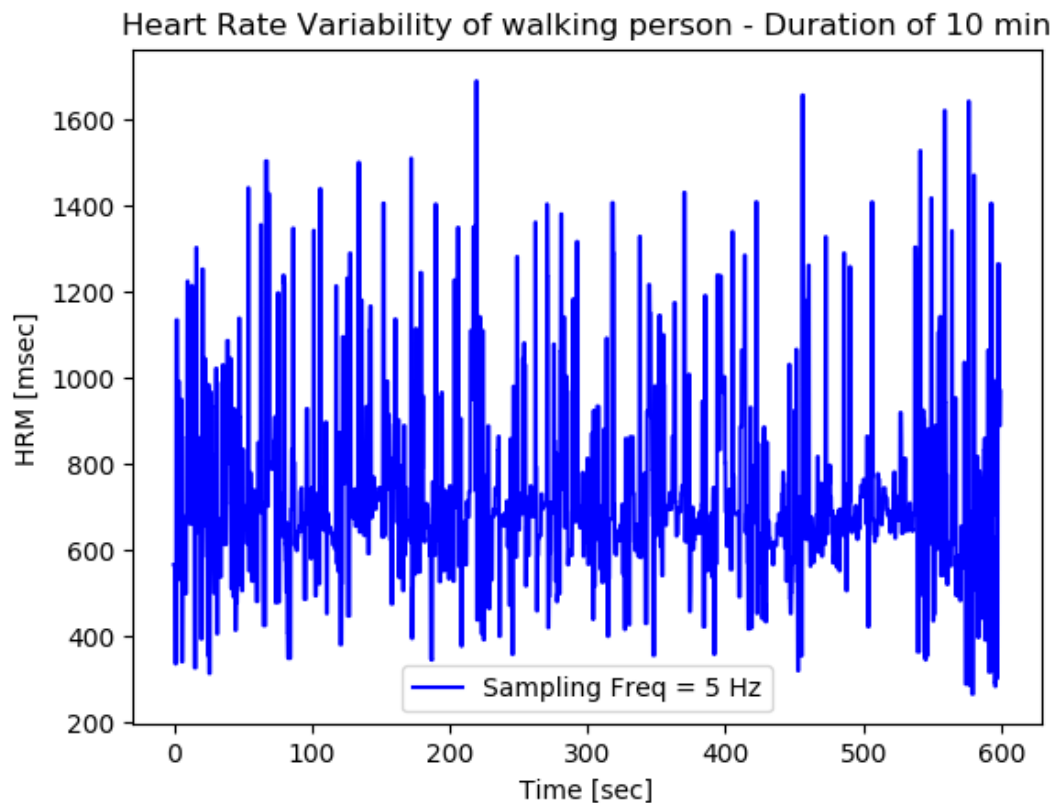
Παρακάτω, αναπαρηστούμε τις γραφικές παραστάσεις των σημάτων γραμμικής επιτάχυνσης και γωνιακής ταχύτητας- και για τους 3 άξονες- με χρήση των ρουτίνων `plot()` και `subplot()`, σε συνάρτηση με τον χρόνο, για διάρκεια περπατήματος 10 λεπτών:

Acceleration of walking person - Duration of 10 min



Angular Velocity of walking person - Duration of 10 min





1.2 Συνάρτηση υπολογισμού ενέργειας βραχέος χρόνου

Στη συνέχεια υλοποιούμε τη ρουτίνα *ste()*, η οποία δέχεται ως είσοδο ένα μονοδιάστατο σήμα και επιστρέφει την ενέργεια βραχέος χρόνου του. Ως **short-time energy** ορίζεται το μέγεθος:

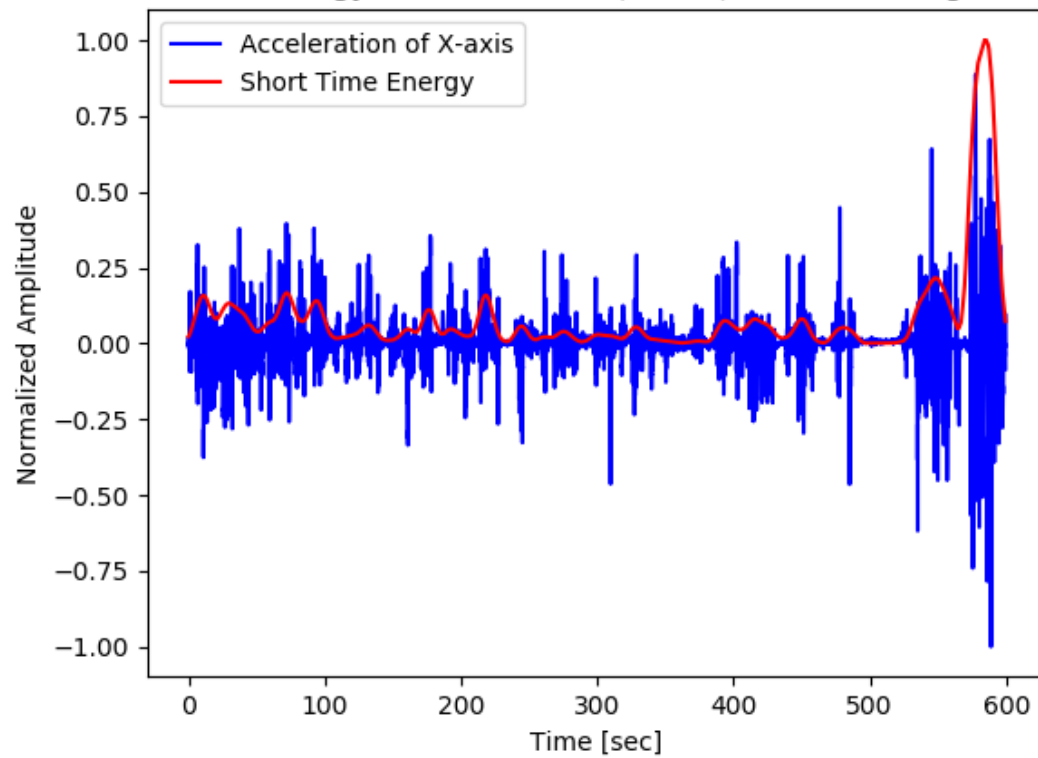
$E_n = \sum_{m=-\infty}^{\infty} [x(m)]^2 w(n-m)$	Eq. (1.1)
---	-----------

Όπου w ένα παράθυρο της επιλογής μας, συνήθως το Hamming παράθυρο.

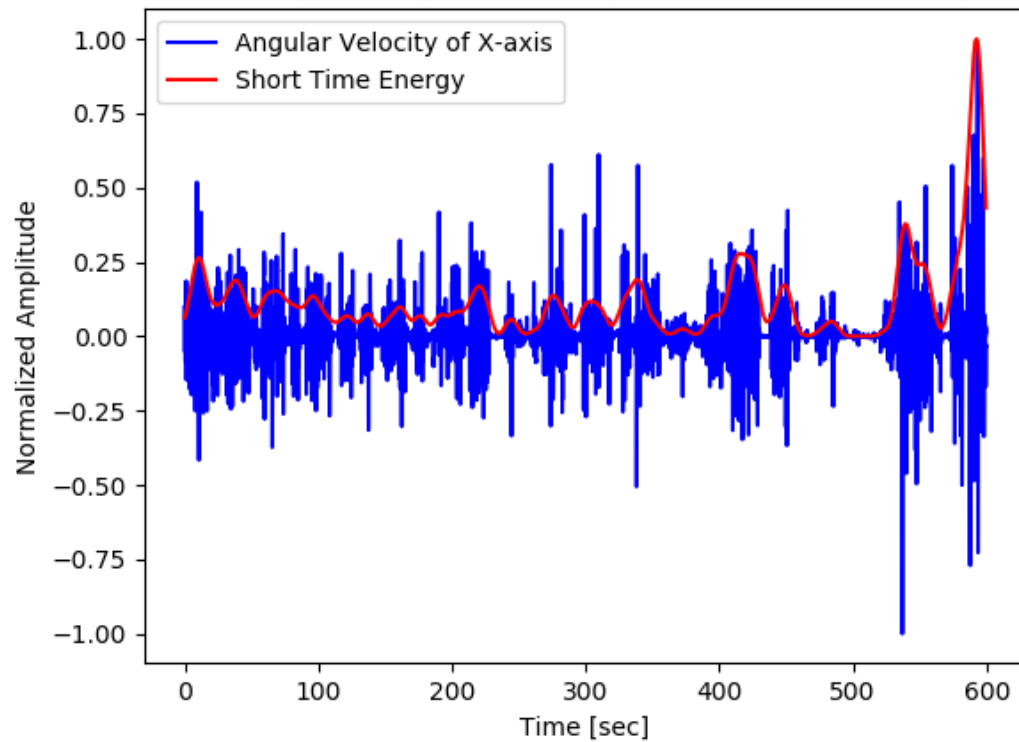
Εφαρμόζουμε την παραπάνω ρουτίνα στα σήματα που αντιστοιχούν στην γραμμική επιτάχυνση και τη γωνιακή ταχύτητα του άξονα X, καθώς και στο σήμα μεταβλητότητας καρδιακών παλμών. Χρησιμοποιούμε **παράθυρα μήκους 20sec**, δηλαδή $20 \cdot fs_1 = 400$ δειγμάτων και $20 \cdot fs_2 = 100$ δειγμάτων για τα δύο πρώτα σήματα και το τρίτο σήμα αντίστοιχα.

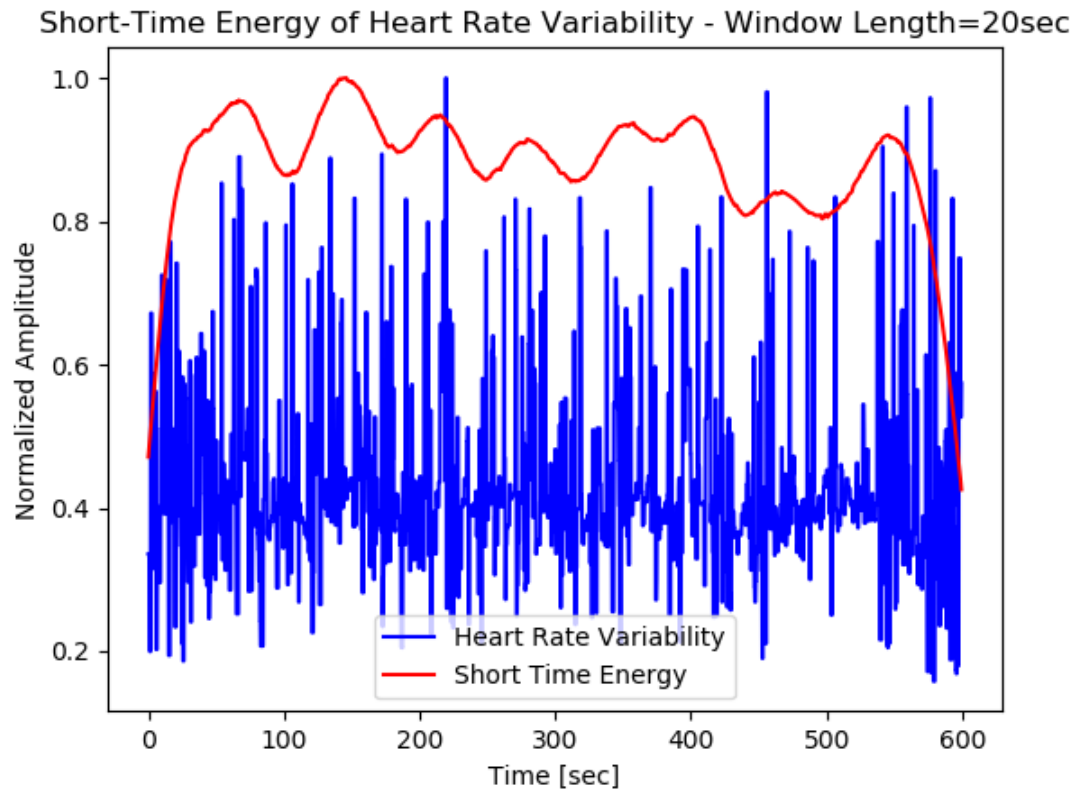
Παρατίθενται οι αντίστοιχες γραφικές παραστάσεις με τον οριζόντιο άξονα σε δευτερόλεπτο και τον κατακόρυφο άξονα να αναπαριστά τα πλάτη των μεγεθών **κανονικοποιημένα**:

Short-Time Energy of Acceleration (X-axis) - Window Length=20sec



Short-Time Energy of Angular Velocity (X-axis) - Window Length=20sec





Παρατηρούμε ότι ένα ενδεικτικό χρονικό διάστημα **ακινησίας** είναι αυτό των **500 - 520 sec**, στο οποίο η ενέργεια βραχέος χρόνου της γραμμικής επιτάχυνσης και της γωνιακής ταχύτητας είναι σχεδόν μηδενικές. Παράλληλα, η μεταβλητότητα καρδιακών παλμών στο ίδιο διάστημα παρουσιάζει σημαντική αύξηση, κάτι που είναι λογικό καθώς όταν ένα άτομο βρίσκεται σε ακινησία, η περίοδος μεταξύ δύο διαδοχικών καρδιακών χτύπων είναι μεγαλύτερη από όταν κινείται.

Αντιθέτως, στο ενδεικτικό χρονικό διάστημα των **570 – 600 sec**, παρατηρούμε ότι η ενέργεια βραχέος χρόνου των σημάτων του επιταχυνσιόμετρου και του γυροσκοπίου αυξάνεται ραγδαία, ενώ αυτή της μεταβλητότητας καρδιακών παλμών μειώνεται σημαντικά. Ως εκ τούτου, μπορούμε να συμπεράνουμε ότι πρόκειται για ένα χρονικό διάστημα **έντονης κινητικής δραστηριότητας**, κατά το οποίο τα χρονικά διαστήματα μεταξύ διαδοχικών χτύπων καρδιάς είναι μικρά, υποδηλώνοντας γρήγορο καρδιοχτύπι.

1.3 Εφαρμογή STFT Μετασχηματισμού

Στο σημείο αυτό, υπολογίζουμε τον STFT Μετασχηματισμό του σήματος του άξονα X του επιταχυνσιόμετρου, καθώς και του σήματος της μεταβλητότητας των καρδιακών παλμών, κάνοντας χρήση της ρουτίνας *stft()* της *Librosa*.

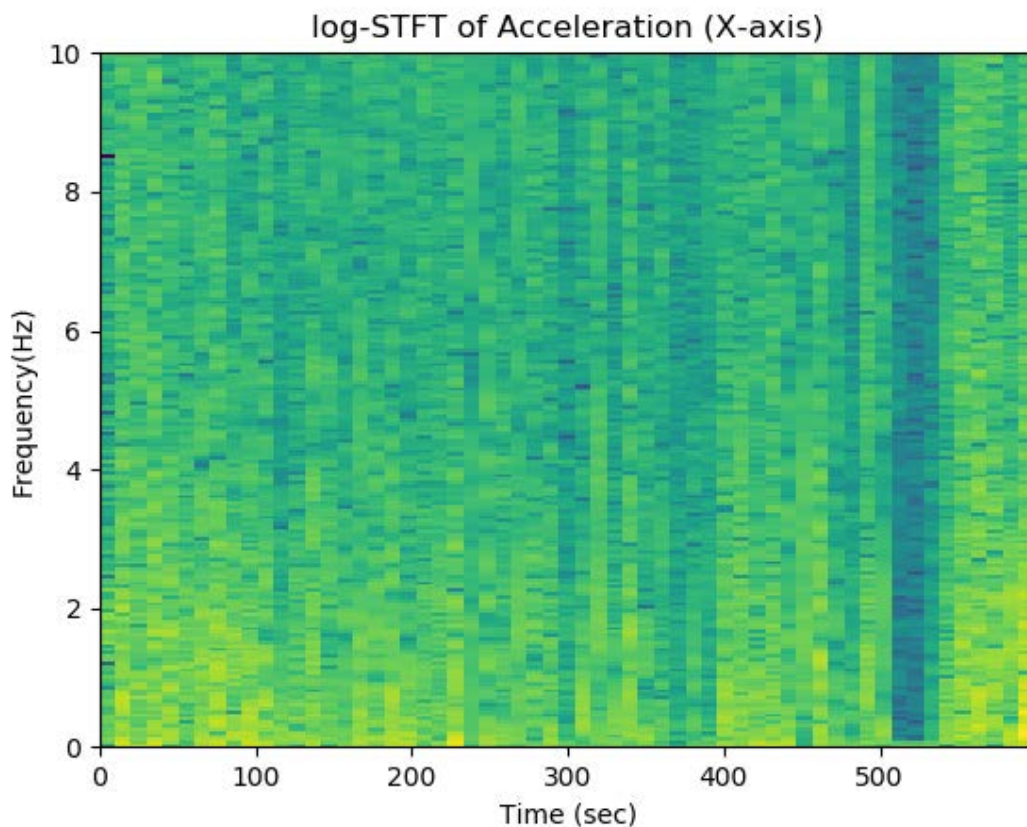
Υπενθυμίζουμε τον Μετασχηματισμό Βραχέος Χρόνου:

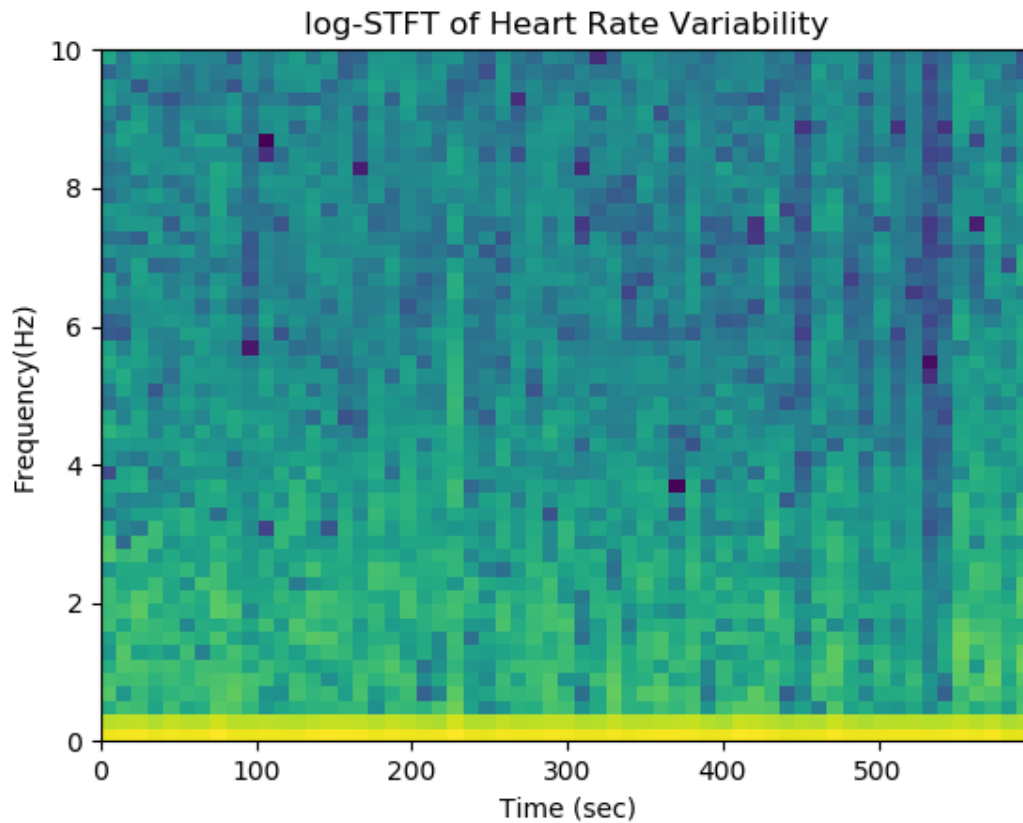
$\text{STFT}(\tau, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - \tau]e^{-j2\pi kn/M}$	Eq. (1.2)
--	-----------

Όπου M το μήκος του DFT μεγαλύτερο ή ίσο του μήκος L του παραθύρου.

Ειδικότερα, χρησιμοποιούμε παράθυρα **μήκους 20 sec** (δηλαδή $20 \cdot 20 = 400$ δειγμάτων για το πρώτο σήμα και $20 \cdot 5 = 100$ δειγμάτων για το δεύτερο) και **επικάλυψη ίση με 10 sec** (δηλαδή 200 και 50 δείγματα αντιστοίχως).

Στην προκειμένη περίπτωση υπολογίζουμε τον **λογαριθμικό** Μετασχηματισμό Βραχέος Χρόνου για τα δύο σήματα. Αναπαρηστούμε, δηλαδή, το πλάτος $20\log_{10}|\text{STFT}(\tau, f)|$ (σε dB) με τις κατάλληλες τιμές στους άξονες του χρόνου (sec) και των συχνοτήτων (Hz) για τα δύο σήματα, με χρήση της συνάρτησης *pcolormesh* της βιβλιοθήκης *matplotlib*:





1.4 Στατιστικά χαρακτηριστικά σημάτων κίνησης και φυσιολογίας

Στο συγκεκριμένο μέρος της άσκησης, αφού φορτώσουμε τα αρχεία *“sleep_03.npz”* και *“step_03.npz”*, προσδιορίζουμε την ενέργεια βραχέος χρόνου των σημάτων acc, gyr (και για τους 3 άξονες) και hrm, όπως ακριβώς πριν. Στη συνέχεια, αξιοποιούμε τις έτοιμες συνάρτησεις **mean()**, **min()**, **max()**, **std()** της βιβλιοθήκης *numpy*, προκειμένου να εξάγουμε στατιστικά χαρακτηριστικά για τις ενέργειες αυτές, τα οποία στη συνέχεια θα αποδειχθούν αντιπροσωπευτικά του αν ένα άτομο κοιμάται ή περπατάει.

Παρατίθενται τα χαρακτηριστικά αυτά σε δύο πίνακες:

File “step_03.npz”				
	mean	min	max	Std
AccX	1129.293	66.914	6097.14	974.3713
AccY	771.456	129.138	2920.08	575.153
AccZ	638.338	3.539	3228.824	542.94
GyrX	1107725.811	6452.93	4204405.4	884631.914
GyrY	207668.86	3461.81	1117712.53	181270.08
GyrZ	226901.246	848.361	1231543.19	255810.963
HRM	127058450	47334247.59	162997740.06	23332682.29

File "sleep_03.npz"				
	mean	min	max	Std
AccX	3.89655	0.4888	47.5844	5.9184
AccY	1.2788	0.1607	14.3963	2.4585
AccZ	4.0606	0.1395	72.3764	8.2128
GyrX	1390.686	3.676	22990.332	3666.790
GyrY	115.231	3.602	1429.249	261.142
GyrZ	132.745	1.7827	1975.555	348.729
HRM	220535443.921	111114503.16	246519344.77	20526628.148

Παρατηρώντας καλύτερα τους δύο παραπάνω πίνακες, μπορούμε να εξάγουμε ορισμένα συμπεράσματα αναφορικά με τη στατιστική συμπεριφορά των σημάτων κίνησης και φυσιολογίας ενός ατόμου που περπατάει ή κοιμάται:

- i. Στον πρώτο πίνακα που αντιστοιχεί σε **κινούμενο άτομο**, βλέπουμε ότι τα στατιστικά χαρακτηριστικά των ενεργειών των σημάτων κίνησης που μετράει το επιταχυνσιόμετρο και το γυροσκόπιο λαμβάνουν αρκετά υψηλές τιμές, με αυτό που ξεχωρίζει να είναι τα σήματα accx και gyrx (κάτι που μπορεί να υποδηλώνει ευθεία κίνηση). Αντιθέτως, τα στατιστικά χαρακτηριστικά της ενέργειας της καρδιακής μεταβλητότητας λαμβάνουν τιμές σημαντικά μικρότερες τις αντίστοιχες τιμές του δεύτερου πίνακα.
- ii. Από την άλλη μεριά, στον δεύτερο πίνακα που αντιστοιχεί σε **άτομο που κοιμάται**, βλέπουμε ότι τα στατιστικά χαρακτηριστικά που αντιστοιχούν στις ενέργειες σημάτων κίνησης acc και gyr λαμβάνουν πολύ πιο χαμηλές τιμές, κάτι που συνεπάγεται ακινησία. Αντιθέτως, οι στατιστικές τιμές της ενέργειας του σήματος heart rate variability είναι αισθητά πολύ μεγαλύτερες, κάτι που ερμηνεύεται ως χαλαρή λειτουργία της καρδιάς.

Συνεπώς, απλώς και μόνο παρατηρώντας τη στατιστική συμπεριφορά των παραπάνω σημάτων κίνησης και καρδιακής μεταβλητότητας, μπορούμε με σχετική ευκολία να συμπεράνουμε αν ο φορέας του έξυπνου ρολογιού κοιμάται ή εμπλέκεται σε κάποια φυσική δραστηριότητα. Το σήμα εκείνο, μάλιστα, η ενέργεια του οποίου φαίνεται να διαφέρει εμφανέστατα ανάμεσα σε ένα άτομο που κοιμάται και σε ένα που περπατά, είναι αυτό της μεταβλητότητας καρδιακών παλμών.

Μέρος 2^ο – Υπολογισμός της Teager-Kaiser Ενέργειας σε πολλαπλές ζώνες συχνοτήτων με τη χρήση Gabor filterbank

Σκοπός Άσκησης

Στο δεύτερο μέρος της εργαστηριακής άσκησης εφαρμόζουμε τον ενεργειακό τελεστή Teager-Kaiser, προκειμένου να προσδιορίσουμε την ενέργεια τριών σημάτων. Για να το επιτύχουμε αυτό χρησιμοποιούμε ζωνοπερατό φιλτράρισμα των σημάτων με Gabor φίλτρα (Gabor filterbank) κάθε φορά κεντραρισμένα σε διαφορετικές συχνότητες. Τέλος, συγκρίνουμε την Teager ενέργεια με την ενέργεια βραχέος χρόνου, όπως αυτή υπολογίστηκε στο μέρος 1.

2.1 Φόρτωση αρχείου “step_00.npz”

Φορτώνουμε πάλι το αρχείο “step_00.npz” του Μέρους 1. Από αυτό απομονώνουμε τα σήματα *accx*, *gyrx* και *hrm*, πάνω στα οποία θα εφαρμόσουμε στη συνέχεια τις συναρτήσεις που θα υλοποιήσουμε.

2.2 Υλοποίηση Μη-γραμμικού Ενεργειακού Τελεστή Teager – Kaiser (TEO)

Υλοποιούμε τη συνάρτηση *teo()*, η οποία θα εφαρμόζει τον Teager – Kaiser ενεργειακό τελεστή σε ένα διακριτό σήμα εισόδου $x[n]$ που λαμβάνει ως όρισμα. Για σήματα διακριτού χρόνου, ο ενεργειακός τελεστής Teager ορίζεται ως:

$\Psi[x[n]] = x^2[n] - x[n-1]x[n+1]$	Eq. (2.1)
--------------------------------------	-----------

Παρακάτω φαίνεται η εν λόγω συνάρτηση σε *python*, η οποία υπολογίζει τον ενεργειακό τελεστή Teager του σήματος *x*, επαναλαμβάνοντας τις τιμές της στα όρια του σήματος:

#2.2 Defining *teo()* function

```
def teo(x):  
    y = np.zeros(np.size(x))  
    for i in range(1,np.size(x)-1):  
        y[i] = (x[i])**2 - (x[i-1])*x[i+1]  
    y[0] = y[1]  
    y[np.size(x)-1] = y[np.size(x)-2]  
    return y
```

2.3 Υλοποίηση Gabor Filterbank

Στη συνέχεια, υλοποιούμε μια συνάρτηση για ζωνοπερατό φιλτράρισμα του σήματος με Gabor φίλτρα (**Gabor Filterbank**). Η κρουστική απόκριση του φίλτρου στον διακριτό χρόνο έχει την εξής μορφή:

$h[n] = \exp[-b^2 n^2] \cos[\Omega_c n]$	Eq. (2.2)
--	-----------

Όπου:

- $\Omega_c = \omega_c / f_s$ είναι η κεντρική συχνότητα του φίλτρου
- f_s είναι η συχνότητα δειγματοληψίας
- $b = a / f_s$, όπου η παράμετρος a ελέγχει το εύρος της απόκρισης συχνότητας
- $-N \leq n \leq N$

Προκειμένου να αντιμετωπίσουμε το πρόβλημα εύρεσης της κατάλληλης κεντρικής συχνότητας ω_c του φίλτρου, ώστε να απομονωθεί η συνιστώσα εκείνη που περιέχει τη χρήσιμη πληροφορία, δημιουργούμε μια **συστοιχία φίλτρων**, τα οποία θα καλύπτουν όλο το πεδίο συχνοτήτων.

Πιο συγκεκριμένα, επιλέγουμε συνολικά **K = 25** Gabor φίλτρα και ορίζουμε την παράμετρο $a = f_s / (2K)$, την ελάχιστη συχνότητα φίλτρων ίση με $f_{c,\min} = a/2$ και τη μέγιστη ίση με $f_{c,\max} = (f_s - a)/2$. Η απόσταση μεταξύ των φίλτρων επιλέγουμε να ορίζεται γραμμικά με βήμα ίσο με $\text{step} = (f_{c,\max} - f_{c,\min}) / K$.

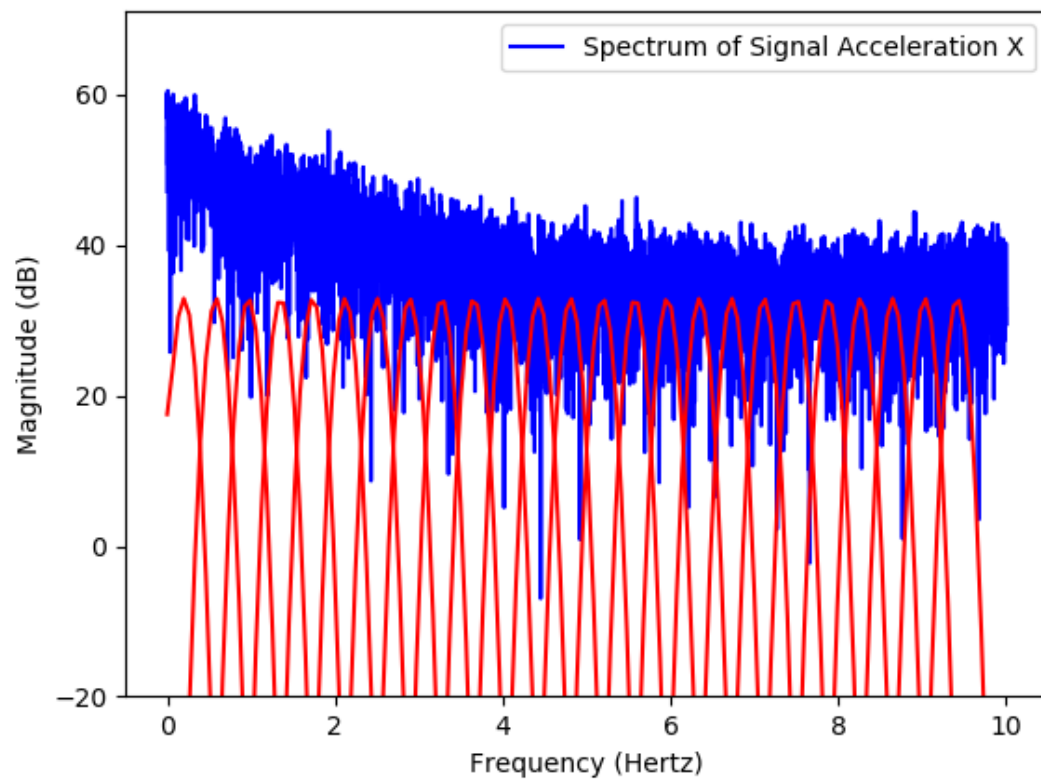
Παρακάτω φαίνεται η υλοποίηση της εν λόγω συνάρτησης *gaborfilt(x, fc, a, fs)* σε γλώσσα προγραμματισμού *python*:

#2.3 Filter Bank

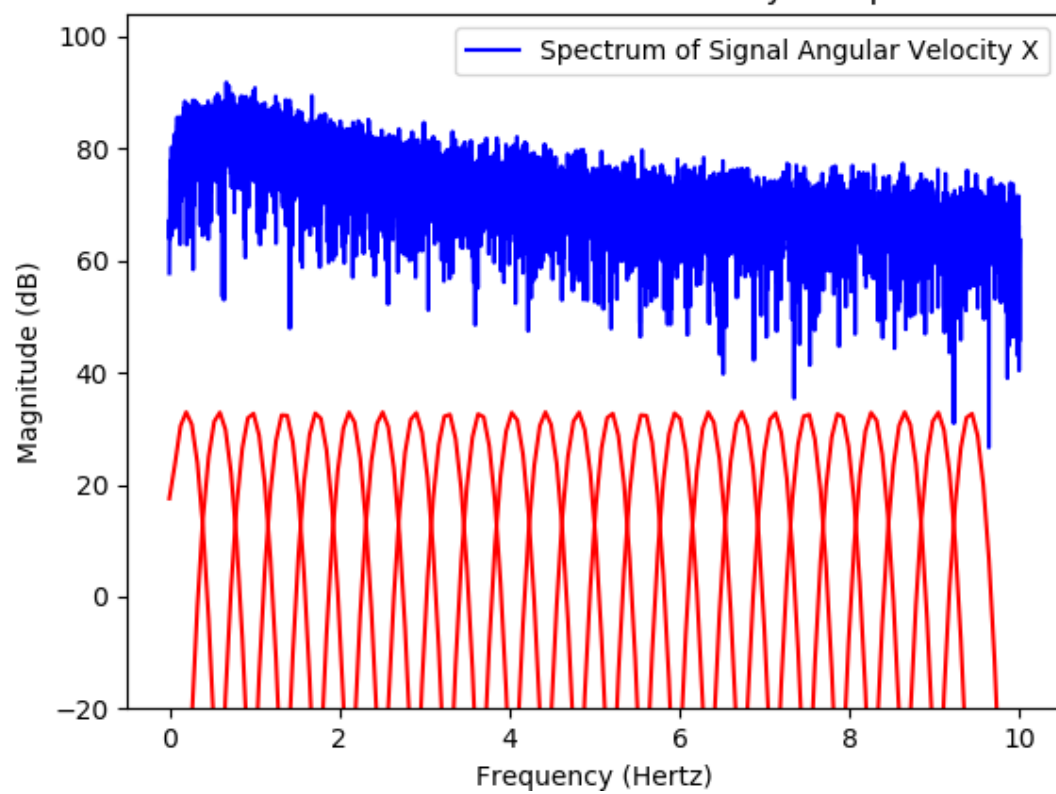
```
def gaborfilt(x, fc, a, fs):  
    b = a/fs  
    N = (3/b)+1  
    n = np.arange(-N,N+1,1)  
    h = np.exp(-(b**2)*(n**2))*np.cos((2*np.pi*fc/fs)*n)  
    out = sp.signal.convolve(x,h, mode="same")  
    #out = signal.lfilter(h,1,x) alternatively  
    return out
```

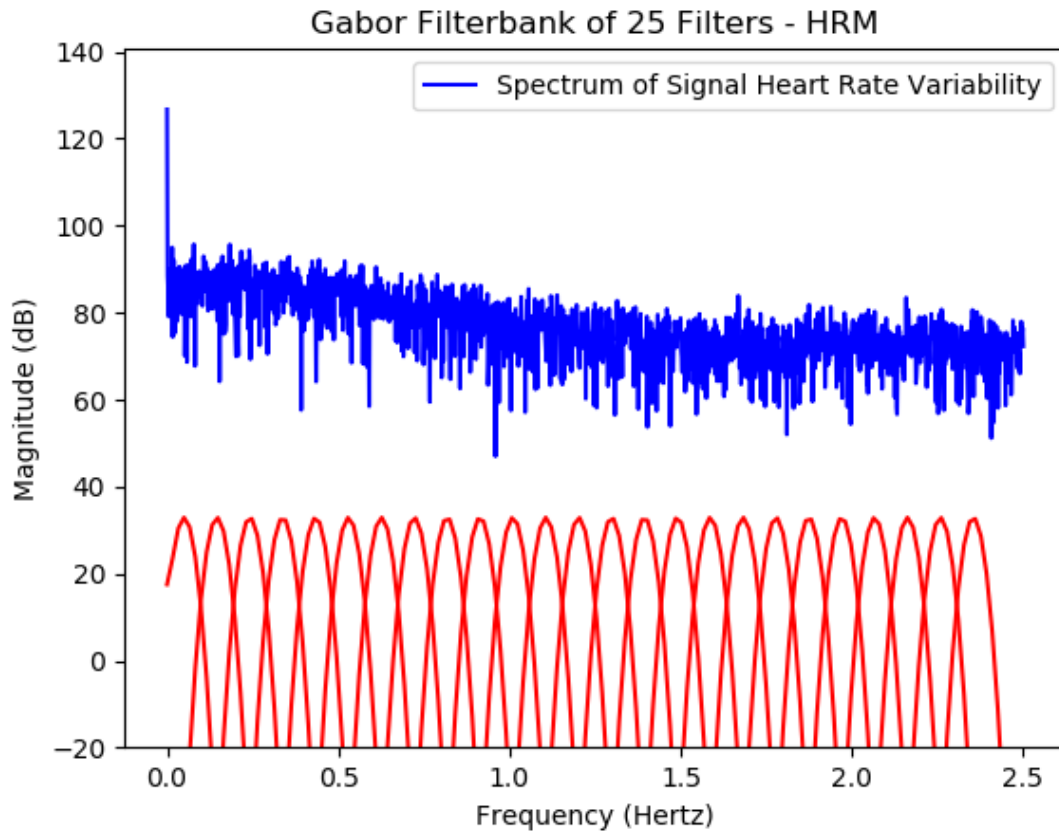
Στη συνέχεια σχεδιάζουμε τη συστοιχία φίλτρων που δημιουργήσαμε στο πεδίο της συχνότητας σε κοινό διάγραμμα με το φάσμα των σημάτων, *accx*, *gyrx* και *hrm*, σε λογαριθμική κλίμακα. Ο οριζόντιος άξονας του διαγράμματος εκφράζει συχνότητες σε Hertz, ενώ ο κατακόρυφος dB. Ειδικότερα:

Gabor Filterbank of 25 Filters - Acceleration X



Gabor Filterbank of 25 Filters - Gyroscope X





2.3 Ομαλοποιητικό Βαθυπερατό Φιλτράρισμα

Ακολουθως, υλοποιούμε μια συνάρτηση που ορίζει ένα ομαλοποιητικό βαθυπερατό φιλτράρισμα (**binomial filter**), το οποίο θα εφαρμόζεται στην έξοδο του ενεργειακού τελεστή, προκειμένου να απομονωθεί όσο είναι δυνατό το επιθυμητό μέρος της εξόδου από το σφάλμα και τις υψίσυχνες συνιστώσες της Teager ενέργειας. Η κρουστική απόκριση του binomial φίλτρου δίνεται από τον τύπο:

$$h_{\text{bin}}[n] = 0.25\delta[n-1] + 0.5\delta[n] + 0.25\delta[n+1]$$

Eq. (2.3)

Παρατίθεται η συνάρτηση *smooth()*, η οποία εφαρμόζει το παραπάνω φίλτρο σε ένα σήμα εισόδου *x*, υλοποιημένη σε γλώσσα προγραμματισμού *python*:

#2.4 Smoothing Binomial Lowpass Filter

```
def smooth(x):
    h = np.array([0.25, 0.5, 0.25])
    return sp.signal.lfilter(h,[1],x)
```

2.5 Παραθύρωση των σημάτων με παράθυρα Hamming

Προτού ξεκινήσουμε τη διαδικασία υπολογισμού της Teager ενέργειας, παραθυρώνουμε με παράθυρα Hamming το αρχικό σήμα $x[n]$ σε πλαίσια ανάλυσης διάρκειας $t_{win} = 20 \text{ sec}$ με βήμα $t_{shift} = 5 \text{ sec}$.

Επομένως, για τα σήματα *accx* και *gyrx*, όπου έχουμε συχνότητα δειγματοληψίας $fs1 = 20 \text{ Hz}$, τα παράθυρα θα είναι μήκους $winlen = t_{win}fs1 = 400 \text{ δειγμάτων}$ με βήμα $winshift = t_{shift}fs1 = 100 \text{ δείγματα}$, ενώ για το σήμα *hrm*, δειγματοληπτούμενο με συχνότητα $fs2 = 5 \text{ Hz}$, τα αντίστοιχα μεγέθη θα είναι **100 και 25 δείγματα**.

Συνολικά, θα προκύψουν **117 πλαίσια ανάλυσης** για κάθε σήμα, καθώς εξαιρούμε τα 3 τελευταία παράθυρα, διότι βγαίνουν εκτός ορίων του συνολικού σήματος. Σημειώνεται, επίσης, ότι για να έχουμε σήματα που αντιστοιχούν σε ακριβώς 10 λεπτά διάρκεια (600 sec), εφαρμόζουμε **padding** τεχνική στο συνολικό σήμα, προσθέτοντας 8 δείγματα στα σήματα *accx* και *gyrx*, και 2 δείγματα στο σήμα *hrm*. Ως fill value επιλέγουμε τη μέση τιμή του εκάστοτε σήματος. Ως αποτέλεσμα, τα δύο πρώτα είναι πλέον ακριβώς 12000 δειγμάτων και το τελευταίο 3000 δειγμάτων.

2.6 Φιλτράρισμα σήματος με χρήση της συστοιχίας Gabor φίλτρων

Στο συγκεκριμένο σημείο φιλτράρουμε το παραθυρωμένο σήμα σε ζώνες συχνοτήτων με χρήση των Gabor φίλτρων που έχουμε δημιουργήσει. Συγκεκριμένα, φιλτράρουμε το σήμα με την κατάλληλη σχεδιασμένη ακολουθία φίλτρων με κρουστικές αποκρίσεις h_i , $i = 1, \dots, 25$. Αυτό επιτυγχάνεται με την εφαρμογή της συνάρτησης *gaborfilt()* σε κάθε ένα εκ των 117 παραθύρων, με κάθε μια εκ των 25 διαφορετικών κεντρικών συχνοτήτων f_c .

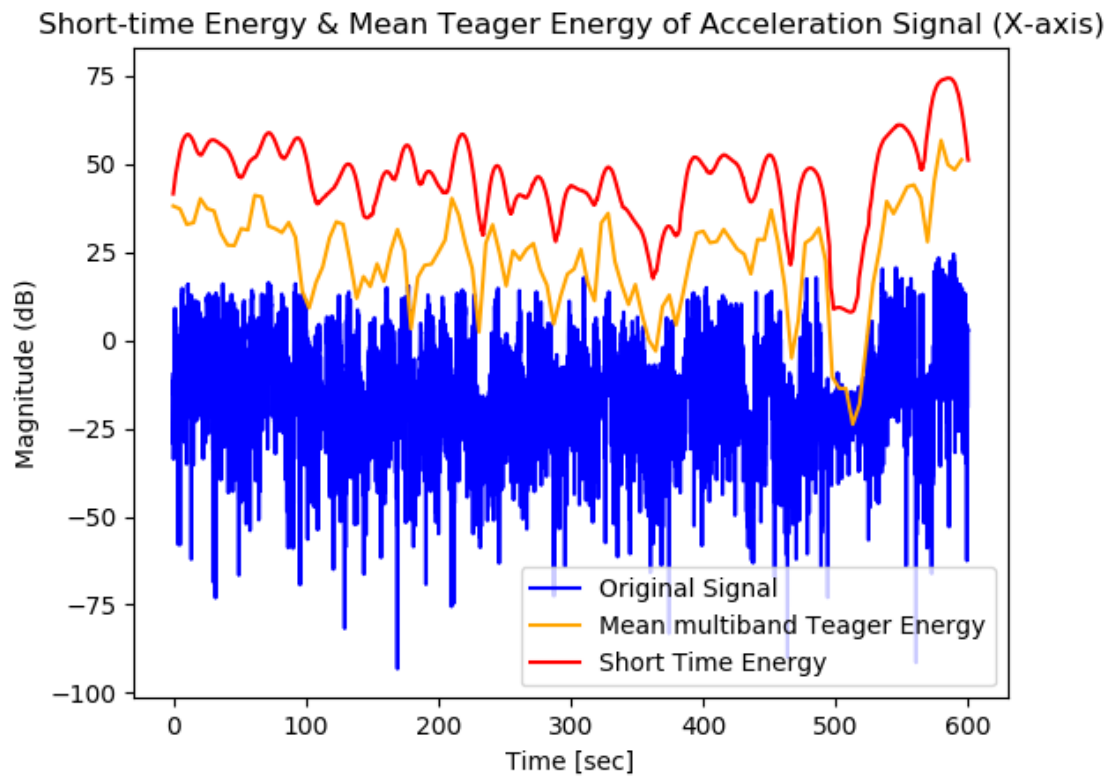
2.7 Εφαρμογή ενεργειακού τελεστή Teager

Στο τελευταίο ερώτημα του 2^{ου} μέρους προσδιορίζουμε την μέση Teager ενέργεια των τριών σημάτων μας, ακολουθώντας τα βήματα του ακόλουθου αλγόριθμου:

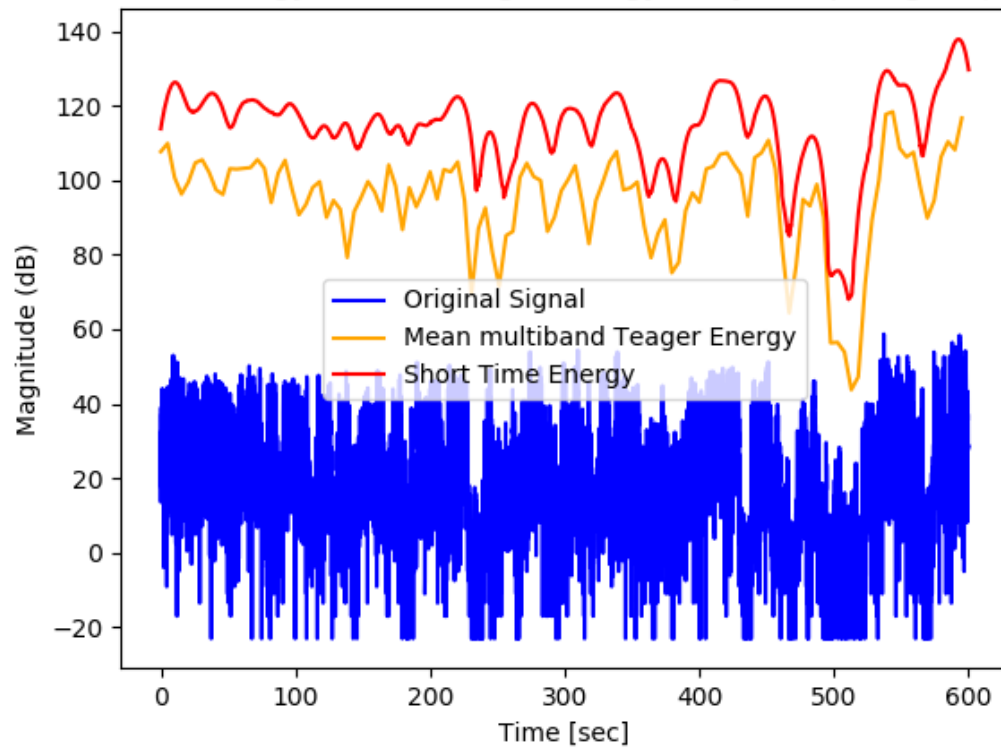
- α) Εφαρμόζουμε τον ενεργειακό τελεστή Teager στην έξοδο του κάθε φίλτρου μέσω της συνάρτησης *teo()* που ορίσαμε.
- β) Έχοντας υπολογίσει την Teager ενέργεια για κάθε φίλτρο εφαρμόζουμε δύο φορές το binomial φίλτρο, μέσω της συνάρτησης *smooth()* που έχουμε ορίσει.
- γ) Ακολουθώντας, υπολογίζουμε τη μέση τιμή της ενέργειας του κάθε φίλτρου, στο εκάστοτε παράθυρο, λαμβάνοντας υπόψη μόνο τις θετικές ενέργειες του σήματος. Αυτό επιτυγχάνεται με χρήση της συνάρτησης *mean_energy()* που ορίζουμε εμείς.
- δ) Προσδιορίζουμε τη μέγιστη τιμή ενέργειας ανά παράθυρο, καθώς και τη κεντρική συχνότητα του φίλτρου στο οποίο αυτή αντιστοιχεί με χρήση των συναρτήσεων *max* και *argmax* της βιβλιοθήκης *numpy*.

Τελικώς, λαμβάνουμε για καθένα εκ των 117 παραθύρων τη μέγιστη τιμή της μέσης ενέργειας Teager (**mean multiband Teager Energy (MTE)**). Κατά την εκτέλεση του κώδικα του αρχείου *part2.py* εκτυπώνονται για τα τρία σήματα οι 117 τιμές της MTE, καθώς και οι αντίστοιχες συχνότητες f_c για τις οποίες μεγιστοποιούνται.

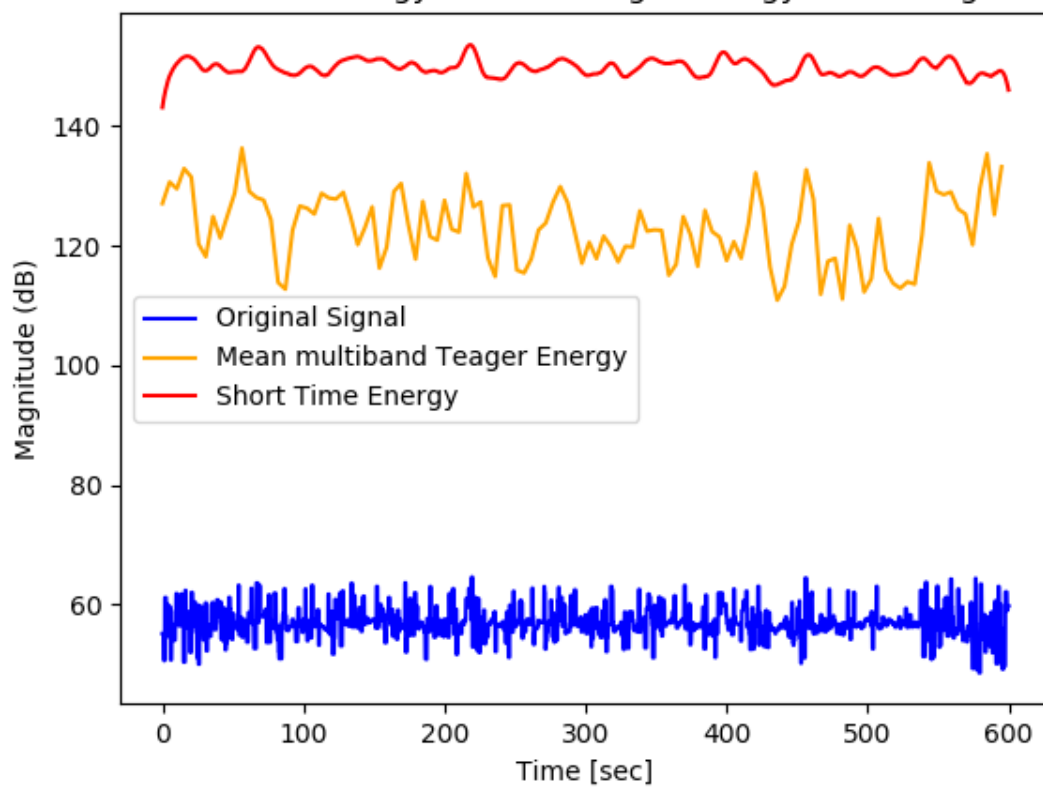
Παρακάτω φαίνονται για τα τρία σήματα της άσκησης οι γραφικές παραστάσεις της MTE σε κοινό διάγραμμα με την ενέργεια βραχέος χρόνου και το ίδιο το σήμα, ως συνάρτηση του χρόνου. Ο κατακόρυφος άξονας βαθμονομείται σε λογαριθμική κλίμακα, ώστε να εκφράζει dB:



Short-time Energy & Mean Teager Energy of Gyroscope Signal (X-axis)



Short-time Energy & Mean Teager Energy of HRM Signal



Παρατηρούμε ότι τουλάχιστον για τα δύο πρώτα σήματα κίνησης $accx$ και $gygx$, η γραφική παράσταση της ενέργειας Teager φαίνεται να ακολουθεί πιστά την περιβάλλουσα της ενέργειας βραχέος χρόνου, λαμβάνοντας απλώς χαμηλότερες τιμές. Όσον αφορά το σήμα μεταβλητότητας καρδιακών παλμών, η συνάρτηση της Teager ενέργειας φαίνεται να είναι αποτελεσματικότερος δείκτης προσδιορισμού των χρονικών διαστημάτων που παρατηρείται κίνηση ή ακίνησια (υψηλής και χαμηλής ενέργειας αντίστοιχα), σε σύγκριση με τη γραφική παράσταση της ενέργειας βραχέος χρόνου, η οποία δεν παρουσιάζει τόσο έντονες αυξομειώσεις.

Συμπερασματικά, λοιπόν, η μέση ενέργεια Teager-Kaiser μπορεί να είναι πιο δύσκολο να υπολογιστεί από την ενέργεια βραχέος χρόνου, καθώς προϋποθέτει πλαισιακή ανάλυση και εφαρμογή μιας συστοιχίας φίλτρων, ωστόσο μπορεί να αποδειχθεί ιδανικότερη για την εξαγωγή συμπερασμάτων που στηρίζονται στην παρατήρηση γραφικών παραστάσεων.

Μέρος 3^ο – Εξαγωγή χαρακτηριστικών, Ανάλυση Πρωτευουσών Συνιστωσών με PCA (Principal Component Analysis) και Συσταδοποίηση τους με τον Αλγόριθμο K-means

Σκοπός Άσκησης

Στο τρίτο μέρος αφού δημιουργήσουμε έναν *mxd* πίνακα που να περιλαμβάνει τα στατιστικά χαρακτηριστικά δύο ειδών ενέργειας, κατηγοριοποιούμε τα δεδομένα του εν λόγω πίνακα με εφαρμογή PCA. Στη συνέχεια, επιχειρούμε και μια συσταδοποίηση δεδομένων με χρήση του αλγορίθμου K-means, την οποία και αξιολογούμε ως προς την επιτυχία της.

3.1 Δημιουργία *mxd* πίνακα

Αρχικά, φορτώνουμε τα αρχεία *“sleep_m.npz”* και *“step_m.npz”* όπου $m = 0, 1, \dots, 19$ και υπολογίζουμε την ενέργεια βραχέος χρόνου και τη μέγιστη μέση Teager ενέργεια για τα σήματα *acc*, *gyr* (και για τους 3 άξονες) και *hgm*, με τρόπο παρόμοιο με αυτόν που ακολουθήσαμε στο μέρος 2^ο. Στη συνέχεια, εξάγουμε τα στατιστικά χαρακτηριστικά, με χρήση των συναρτήσεων *mean*, *min*, *max* και *std* της βιβλιοθήκης *numpy*, για τις ενέργειες βραχέος χρόνου και Teager, αντίστοιχα. Το τελικό αποτέλεσμα αποθηκεύεται σε έναν *mxd* πίνακα, όπου **m = 20** ο αριθμός των αρχείων και **d = 56** ο συνολικός αριθμός των χαρακτηριστικών των σημάτων κάθε αρχείου (4 χαρακτηριστικά x 2 είδη ενεργειών x 7 σήματα ανά αρχείο).

Σημείωση:

Επειδή η δημιουργία του *mxd* πίνακα είναι χρονοβόρα (περίπου 1 αρχείο ανά λεπτό), επιλέξαμε να την πραγματοποιήσουμε ξεχωριστά στο αρχείο *“part3_1.py”* και το αποτέλεσμα της να αποθηκεύεται με χρήση της εντολής *numpy.save* ως ένα αρχείο με όνομα *“data_mxd.npy”*. Το υπόλοιπο μέρος της 3^{ης} άσκησης υλοποιείται στο αρχείο *“part3.py”*.

3.2 Ανάλυση Πρωτευουσών Συνιστωσών

Στη συνέχεια της άσκησης, εφαρμόζουμε τη μέθοδο Ανάλυσης Πρωτευουσών Συνιστωσών (Principal Component Analysis – PCA), προκειμένου να μειώσουμε τη διαστατικότητα του *mxd* πίνακα σε πρωτεύουσες συνιστώσες, τέτοιες, ώστε:

- Να είναι ασυσχέτιστες/ορθογώνιες μεταξύ τους
- Διατηρώντας τις $p < d$ εξ αυτών, να παίρνουμε ένα ελάχιστο μέσο τετραγωνικό σφάλμα.

Για την υλοποίηση της PCA εφαρμόζουμε τη μέθοδο SVD (Singular Value Decomposition) στον πίνακα εισόδου X μεγέθους *mxd*, ο οποίος περιέχει τα αρχικά διανύσματα εισόδου ως γραμμές.

Η εν λόγω μέθοδος αναλύει τον πίνακα εισόδου X σε γινόμενο τριών πινάκων ως εξής:

$X = UV^T$	Eq. (3.1)
------------	-----------

Όπου:

- $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ και $r = (X) \leq \min(m, d) = 40$
- Η στήλη vk του πίνακα V είναι η k -οστή πρωτεύουσα κατεύθυνση ενώ η αντίστοιχη τιμή $\lambda_k = \sigma_k^2/m$ περιγράφει το ποσοστό της διακύμανσης των αρχικών δεδομένων που εξηγεί η συνιστώσα $y_k = vk^T x$.
- Ο πίνακας πρωτευουσών συνιστωσών δίνεται από τη σχέση $Y = XV$.

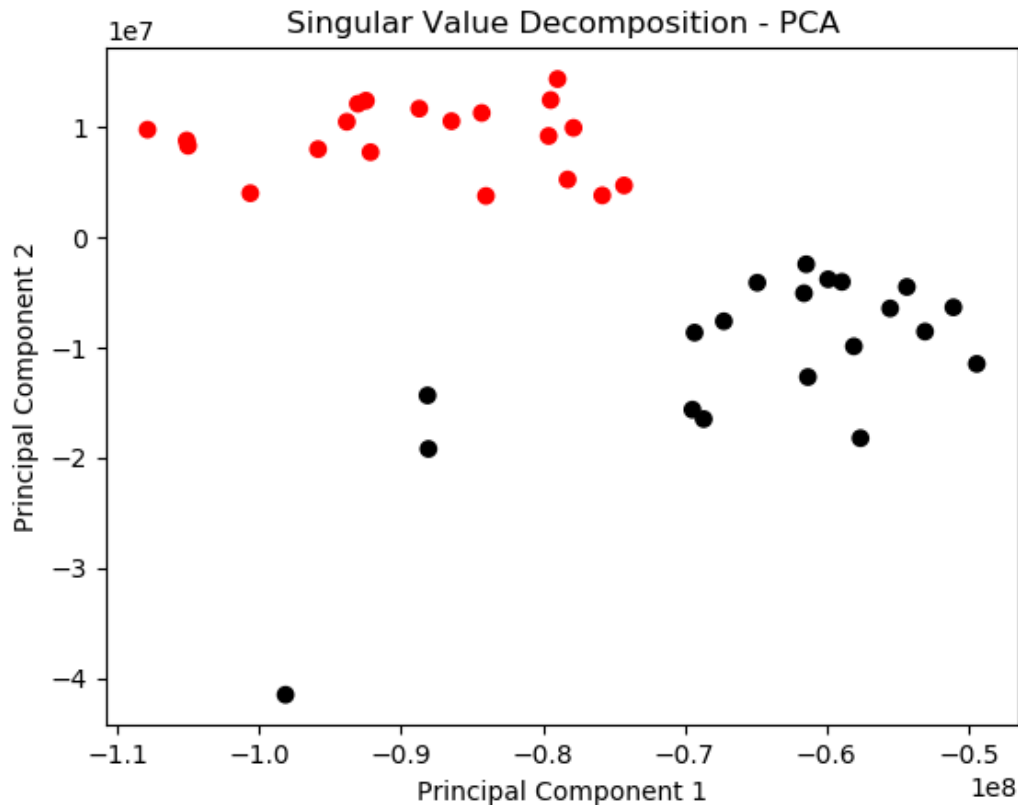
Υλοποιούμε τον παραπάνω αλγόριθμο με χρήση της συνάρτησης `svd()` της βιβλιοθήκης `numpy.linalg`. Παρακάτω φαίνεται το διάλυσμα των **ποσοστών της διακύμανσης** των δεδομένων που εξηγεί η κάθε συνιστώσα:

$\Lambda\% =$ [9.72390761e-01, 2.25261203e-02, 3.14049588e-03, 7.63681907e-04, 4.90698402e-04, 4.22062892e-04, 1.75283073e-04, 4.11546268e-05, 2.05893987e-05, 1.11418210e-05, 9.14712659e-06, 4.70647562e-06, 1.37724750e-06, 1.15427875e-06, 6.84932500e-07, 3.89986631e-07, 2.63165369e-07, 1.32099910e-07, 6.30380756e-08, 4.67345742e-08, 3.19301803e-08, 9.44348804e-09, 3.19679281e-09, 7.94245355e-10, 1.46746347e-10, 8.36585628e-11, 7.45933099e-11, 7.38349975e-12, 4.70275413e-12, 2.63914169e-12, 7.63249306e-13, 3.45663645e-13, 4.63145139e-14, 1.49583355e-14, 8.02895362e-15, 1.55041576e-15, 3.53337806e-16, 1.23591057e-16, 3.72944975e-17, 9.01425899e-19]

Παρατηρούμε ότι η πρώτη πρωτεύουσα συνιστώσα v_1 , εξηγεί το εντυπωσιακό ποσοστό του **97% των αρχικών δεδομένων**, ενώ παίρνοντας απλώς μόνο τις 2 πρώτες συνιστώσες, v_1 και v_2 , μπορούμε να εξηγήσουμε το **99%** των δεδομένων μας. Με αυτόν τον τρόπο η διαστατικότητα του αρχικού μας πίνακα μειώνεται αισθητά σε μόλις **$p = 2$ στήλες!**

Ο λόγος για τον οποίο επαρκούν τόσες λίγες πρωτεύουσες συνιστώσες για την επαρκή περιγραφή του $m \times d$ πίνακα εισόδου, είναι διότι σε αυτόν εμπεριέχονται στατιστικά χαρακτηριστικά από σήματα που μοιάζουν πάρα πολύ μεταξύ τους σε οργανωμένες θέσεις. Οι πρώτες 20 γραμμές, άλλωστε, του πίνακα περιέχουν στατιστικά χαρακτηριστικά από σήματα κίνησης και φυσιολογίας ενός κινούμενου ατόμου, ενώ οι 20 τελευταίες τα αντίστοιχα μεγέθη από ένα άτομο που κοιμάται. Οι θέσεις κάθε γραμμής, μάλιστα, αναφέρονται διαρκώς στο ίδιο χαρακτηριστικό, της ίδιας ενέργειας, του ίδιου σήματος, οπότε και περιέχουν παρεμφερείς τιμές. Επομένως, ο αρχικός πίνακας εισόδου είναι κάθε άλλο παρά τυχαίος.

Παρακάτω φαίνεται το διάγραμμα διασποράς (**scatter plot**) των δύο πρώτων πρωτευουσών συνιστωσών – που αντιστοιχούν και στις μεγαλύτερες διακυμάνσεις – του πίνακα Χ:



Οι πρώτες 20 κουκίδες **μαύρου** χρώματος αντιστοιχούν σε σήματα περπατήματος, ενώ οι υπόλοιπες 20 **κόκκινου** χρώματος αντιστοιχούν σε σήματα ύπνου.

3.3 Αλγόριθμος K-means

Στη συνέχεια, εφαρμόζουμε τον αλγόριθμο K-means στις δύο πρώτες πρωτεύουσες συνιστώσες του προηγούμενου ερωτήματος για τη συσταδοποίηση τους. Ακολουθούμε τα εξής βήματα:

(α) **Αρχικοποίηση**: Επιλέγουμε τυχαία κέντρα $y_i(0)$, όπου $1 \leq i \leq 2$.

(β) **Ταξινόμηση**: Αντιστοιχούμε κάθε ένα από τα 40 δείγματα με το κοντινότερο κέντρο του, έτσι ώστε να ικανοποιείται η σχέση:

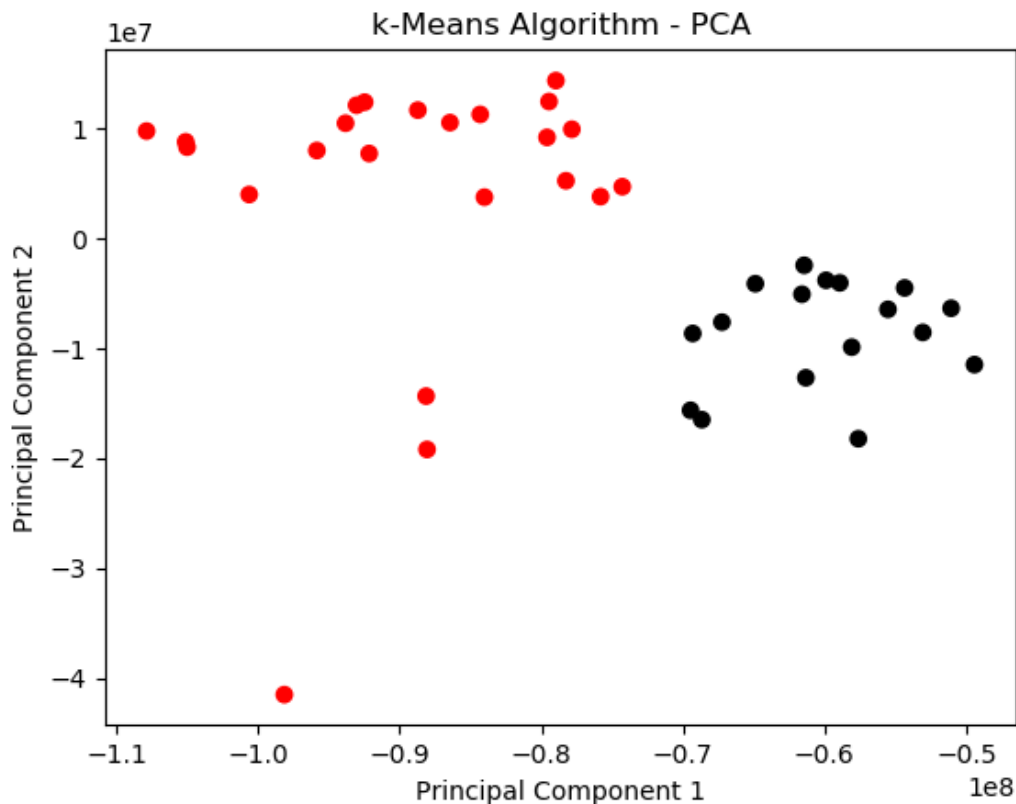
$$x \in C_i \iff \text{dist}[x, y_i(n)] \leq \text{dist}[x, y_j(n)] \quad \forall j \neq i$$

(γ) **Ανανέωση κέντρων**: Επαναυπολογίζουμε τα κέντρα κάθε συστάδας ως τον αριθμητικό μέσο των δειγμάτων που της έχουν ανατεθεί.

(δ) **Συνθήκη τερματισμού**: Ελέγχουμε αν η ολική “παραμόρφωση” $D(n) = \sum_{i=1}^m \text{dist}[x_i, y_j]$, $x \in C_j$ στο βήμα n διαφέρει από την ολική παραμόρφωση $D(n-1)$ στο βήμα $n-1$, λιγότερο

από το κατώφλι **Threshold = 10^{-7}** . Αν δεν ικανοποιείται η συνθήκη τερματισμού επανερχόμαστε στο βήμα (β).

Παρακάτω φαίνεται το διάγραμμα διασποράς (**scatter plot**) των δύο πρώτων πρωτευουσών συνιστωσών – που προκύπτει έπειτα από την υλοποίηση του αλγορίθμου K-means:



Παρατηρούμε ότι ο αλγόριθμος K-means καταφέρνει να ομαδοποιήσει αρκετά καλά τα αρχικά δεδομένα, χωρίς όμως να το πετυχαίνει στο 100%, καθώς όπως βλέπουμε **τρεις κουκίδες** που θα έπρεπε να είναι μαύρες, σύμφωνα με το scatter plot του προηγούμενου ερωτήματος, εν τέλει **κατατάσσονται λανθασμένα** στην συστάδα των κόκκινων.

3.4 Αξιολόγηση Απόδοσης της Συσταδοποίησης

Τέλος, κάνοντας χρήση της συνάρτησης `sklearn.metrics.adjusted_rand_score` υπολογίζουμε τον δείκτη **Rand Index** της συσταδοποίησης του αλγορίθμου K-means σε σύγκριση με αυτήν της PCA. Ο δείκτης αυτός μετράει την ομοιότητα μεταξύ δύο διαφορετικών ομαδοποιήσεων δεδομένων, μετρώντας όλα τα ζεύγη στα οποία συμφωνούν οι ομαδοποιήσεις, ως προς τα ζεύγη που διαφωνούν. Στην προκειμένη περίπτωση:

Random Index = 0.7153659017686895

ο οποίος είναι αρκετά κοντά στη μονάδα, υποδηλώνοντας μια αρκετά καλή συσταδοποίηση του αλγορίθμου K-means με αυτήν της PCA (ομοιότητα 72%). Προφανώς, η ομοιότητα δεν είναι τέλεια, όπως φάνηκε και από τις 3 κουκίδες λάθους χρώματος.

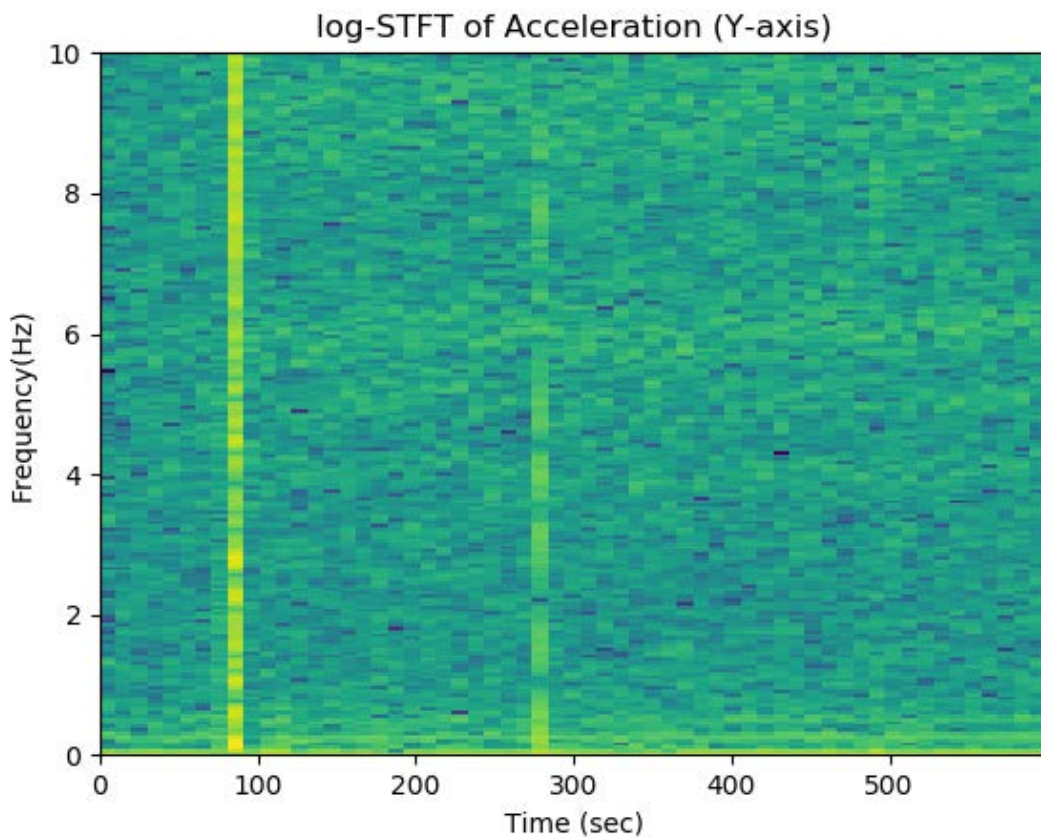
Μέρος 4^ο – Αποθορυβοποίηση σημάτων κίνησης με φίλτρα Butterworth και Wiener

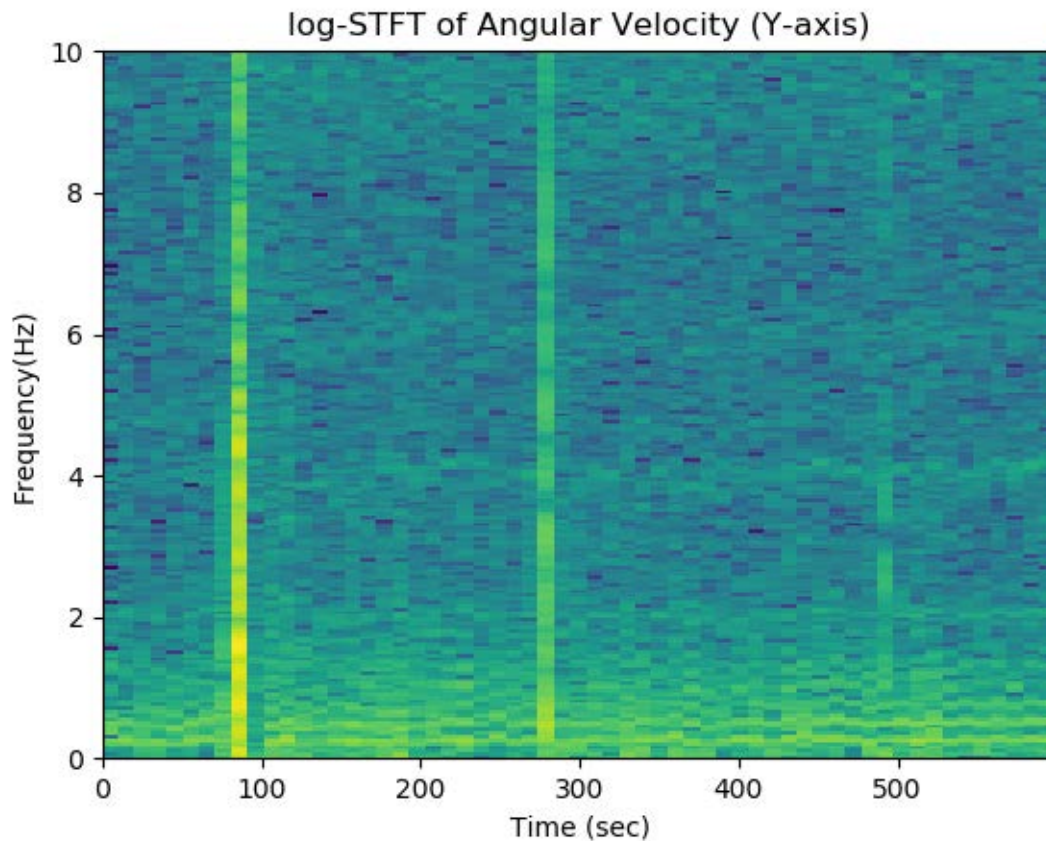
Σκοπός Άσκησης

Στο 4^ο μέρος της εργαστηριακής άσκησης εξετάζουμε την αποθορυβοποίηση δύο σημάτων με χρήση φίλτρων Butterworth και Wiener, συγκρίνοντας τις δύο αυτές μεθόδους ως προς την αποτελεσματικότητά τους να εξαλείψουν τον θόρυβο δίχως να χαθεί χρήσιμη πληροφορία.

4.1 Φόρτωση αρχείου και STFT Μετασχηματισμός

Αφού φορτώσουμε το αρχείο *“sleep_01.npz”*, υπολογίζουμε και σχεδιάζουμε σε λογαριθμική κλίμακα τον STFT Μετασχηματισμό των σημάτων *acc* και *gyr* για τον άξονα Y, χρησιμοποιώντας παράθυρα διάρκειας 20 sec και επικάλυψης ίση με 10 sec. Ειδικότερα:



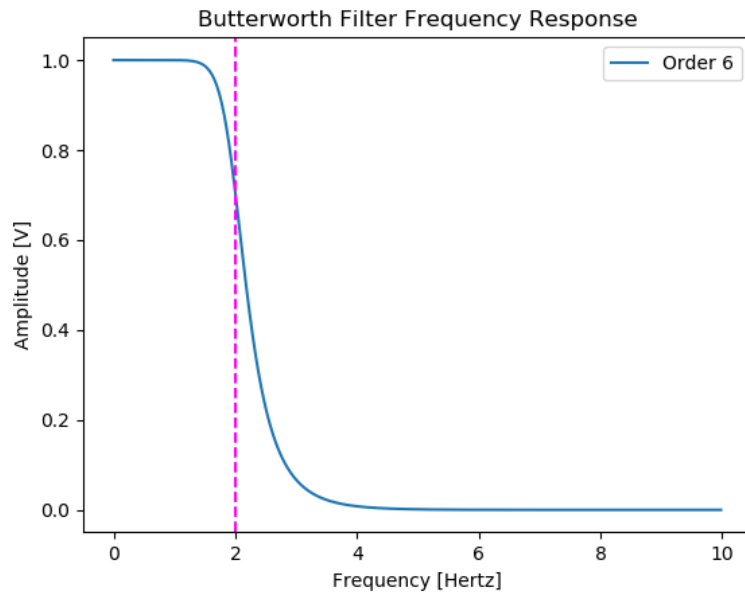


Παρατηρούμε ότι σε χρονικές περιόδους που ο χρήστης 'φαίνεται' ακίνητος, το σήμα καταλαμβάνει ένα μεγάλο μέρος του φάσματος, με 'ασθενείς' όμως τιμές. Αυτό συμβαίνει διότι εμφανίζεται **θόρυβος** μηχανικής ή ηλεκτρικής φύσεως, ο οποίος επιδρά στα MEMS (Microelectromechanical systems) που χρησιμοποιούνται από τη φορητή συσκευή.

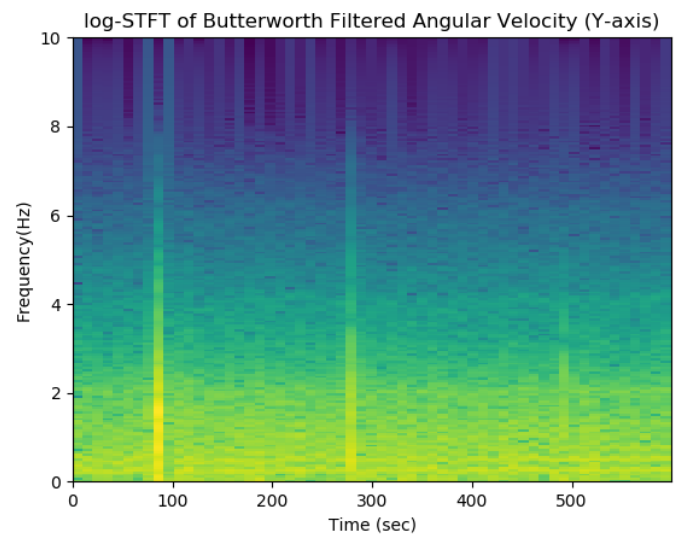
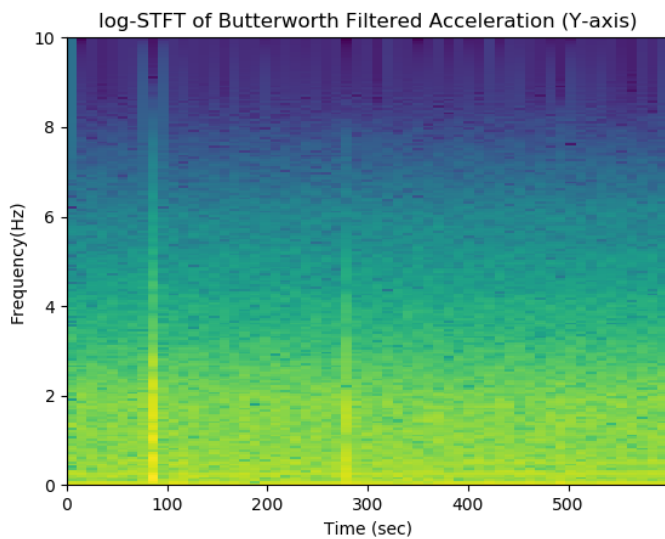
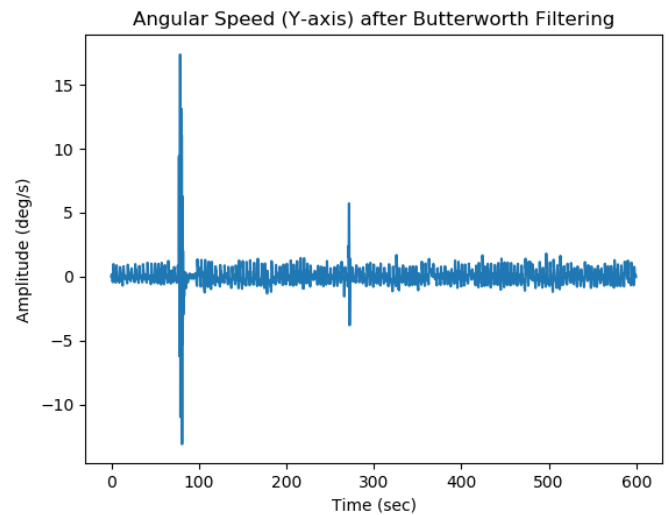
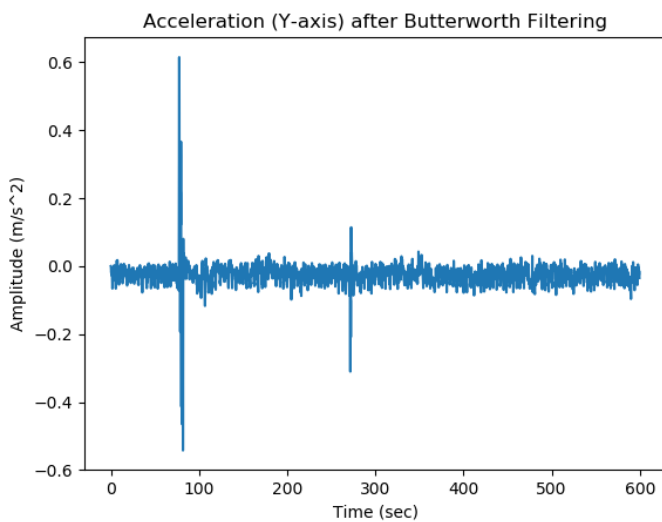
4.2 Εφαρμογή Butterworth φίλτρου

Στο ερώτημα αυτό σχεδιάζουμε ένα ψηφιακό βαθυπερατό φίλτρο Butterworth, με χρήση της συνάρτησης *butter()* της βιβλιοθήκης *scipy.signal*, προκειμένου να φιλτράρουμε τον άξονα Y των σημάτων *accx* και *gyrx*. Υποθέτουμε ότι η ελάχιστη συχνότητα που εμφανίζει ο θόρυβος είναι 2 Hz, ενώ η χρήσιμη πληροφορία του σήματος καταλαμβάνει όλο το εύρος συχνοτήτων 0 - 10 Hz.

Σχεδιάζουμε, λοιπόν, το ψηφιακό φίλτρο Butterworth **τάξης 6**, συχνότητας δειγματοληψίας **$f_s = 20 \text{ Hz}$** και συχνότητας αποκοπής **$f_{\text{cutoff}} = 2 \text{ Hz}$** , η συνάρτηση απόκρισης πλάτους του οποίου φαίνεται στο παρακάτω διάγραμμα:



Ακολουθως, φαίνονται οι γραφικές παραστάσεις των δύο φιλτραρισμένων σημάτων, καθώς και οι log-STFT τους:



Παρατηρούμε ότι ενώ η εφαρμογή του ψηφιακού Butterworth φίλτρου φαίνεται να περιορίζει σε κάποιο βαθμό τον θόρυβο του σήματος, στα διαγράμματα των log-STFT παρουσιάζεται μια εξασθένιση στο διάστημα 0 – 2Hz, γεγονός που μπορεί να οδηγήσει στο να χαθεί κάποια χρήσιμη πληροφορία φασματικών συνιστωσών. Επομένως, η επιλογή φιλτραρίσματος με Butterworth φίλτρο δεν φαντάζει ιδανική.

4.3 Εφαρμογή Wiener φίλτρου

Ακολουθως, επιχειρούμε να αποθορυβοποιήσουμε τα δύο σήματα, υλοποιώντας ένα μη-αιτιατό IIR φίλτρο Wiener για την απαλοιφή του θορύβου. Θεωρούμε ασυσχέτιστο θόρυβο με το σήμα, επομένως η συνάρτηση μεταφοράς του φίλτρου Wiener θα είναι:

$H(e^{j\omega}) = \frac{Pd(e^{j\omega})}{Pd(e^{j\omega}) + Pv(e^{j\omega})}$	Eq. (4.1)
--	-----------

Όπου:

- $Pd(e^{j\omega})$ το φάσμα ισχύος του σήματος χωρίς θόρυβο
- $Pv(e^{j\omega})$ το φάσμα ισχύος του θορύβου

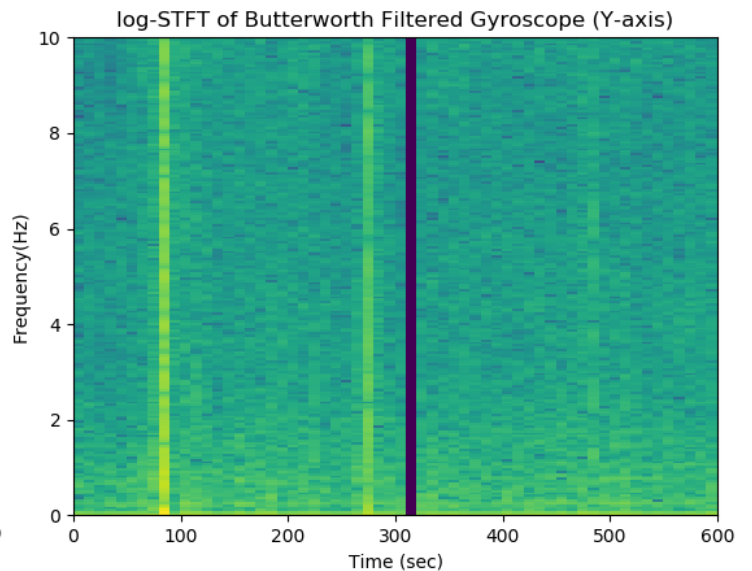
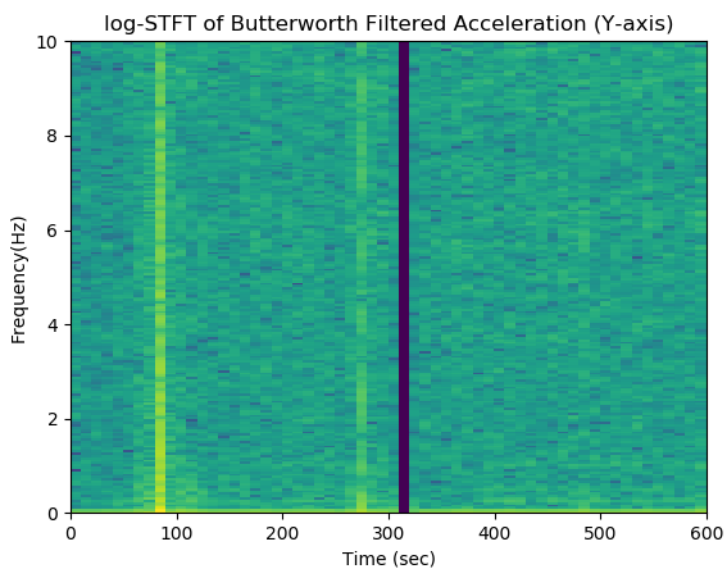
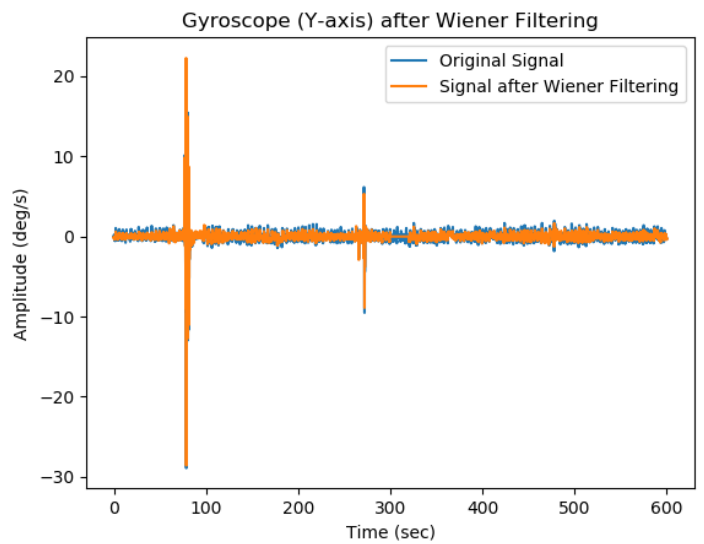
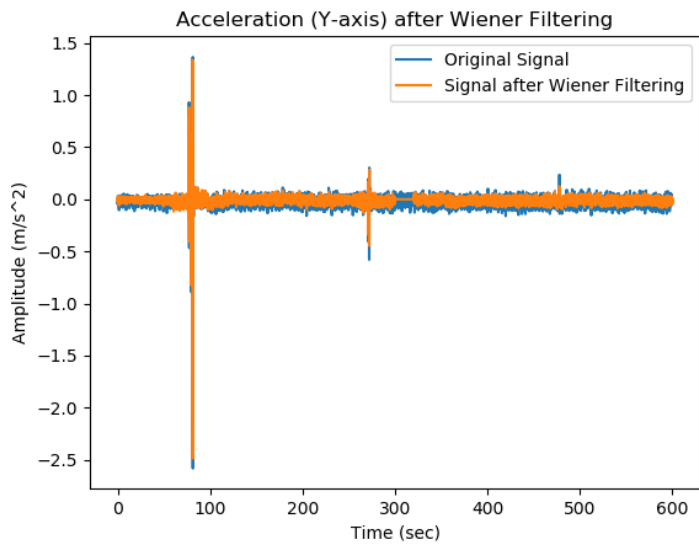
Προκειμένου να προσδιορίσουμε το φάσμα ισχύος του θορύβου, απομονώνουμε το τμήμα των σημάτων που αντιστοιχεί στη χρονική διάρκεια **300 – 550 sec** (συνολικό τμήμα $N = 400$ δειγμάτων), οπότε και ο φορέας του ρολογίου παραμένει ακίνητος. Στη συνέχεια, υπολογίζουμε στο συγκεκριμένο κομμάτι το φάσμα ισχύος του θορύβου ως:

$Pv(e^{j\omega}) = \frac{1}{N} \left \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \right ^2$	Eq. (4.2)
---	-----------

Έπειτα, προκειμένου να υπολογίσουμε το φάσμα ισχύος του σήματος χωρίς θόρυβο $Pv(e^{j\omega})$, αρχικά χωρίζουμε το συνολικό σήμα σε κινούμενα παράθυρα μήκους **L = 400**, χωρίς επικάλυψη και σε κάθε ένα εξ αυτών υπολογίζουμε το φάσμα ισχύος του θορυβώδους σήματος $Px(e^{j\omega})$. Στη συνέχεια, για κάθε παραθυρωμένο τμήμα έχουμε $Pd(e^{j\omega}) = Px(e^{j\omega}) - Pv(e^{j\omega})$ (μηδενίζοντας ωστόσο τις αρνητικές τιμές), λαμβάνοντας εν τέλει τη συνάρτηση μεταφοράς του ψηφιακού φίλτρου Wiener.

Αφού φιλτράρουμε το κάθε πλαίσιο, συνενώνουμε το τελικό αποτέλεσμα σε ένα σήμα εξόδου, **“πετώντας” το αμιγώς φανταστικό μέρος** του φιλτραρισμένου σήματος, θεωρώντας πως η πληροφορία που αυτό περιέχει είναι αμελητέα.

Παρακάτω, φαίνονται οι γραφικές παραστάσεις των δύο φιλτραρισμένων σημάτων σε κοινό διάγραμμα με τα αρχικά σήματα, καθώς και οι log-STFT τους. Σημειώνεται ότι για να αποφευχθεί η εμφάνιση της τιμής **log0** στις ζώνες συχνοτήτων που το βέλτιστο φίλτρο Wiener έχει απόκριση συχνότητας μηδενικού πλάτους, προσθέτουμε στον STFT μια μικρή θετική σταθερά (τάξης μεγέθους 10^{-4} ώστε να μην επηρεάζει το υπόλοιπο διάγραμμα). Ειδικότερα:

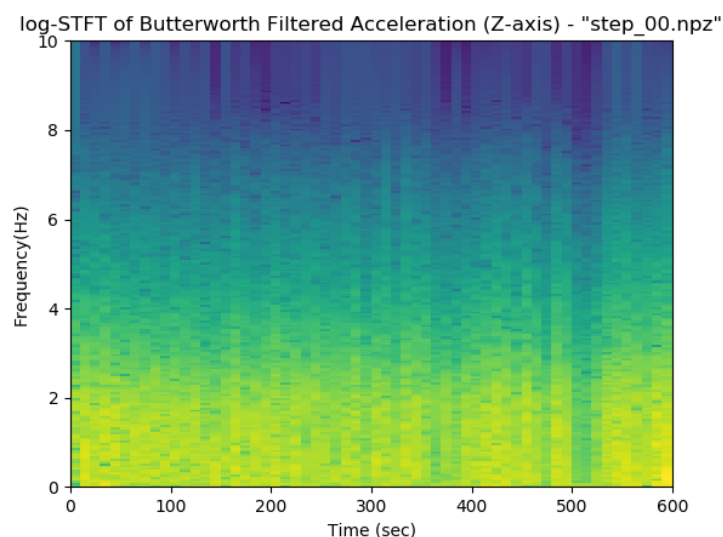
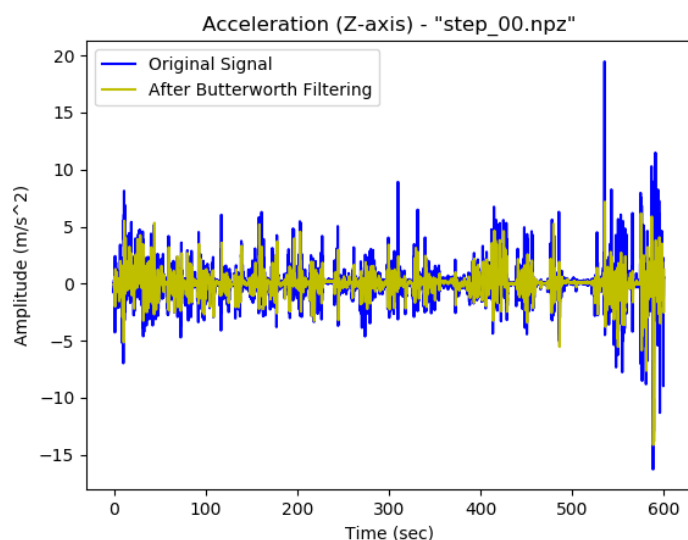
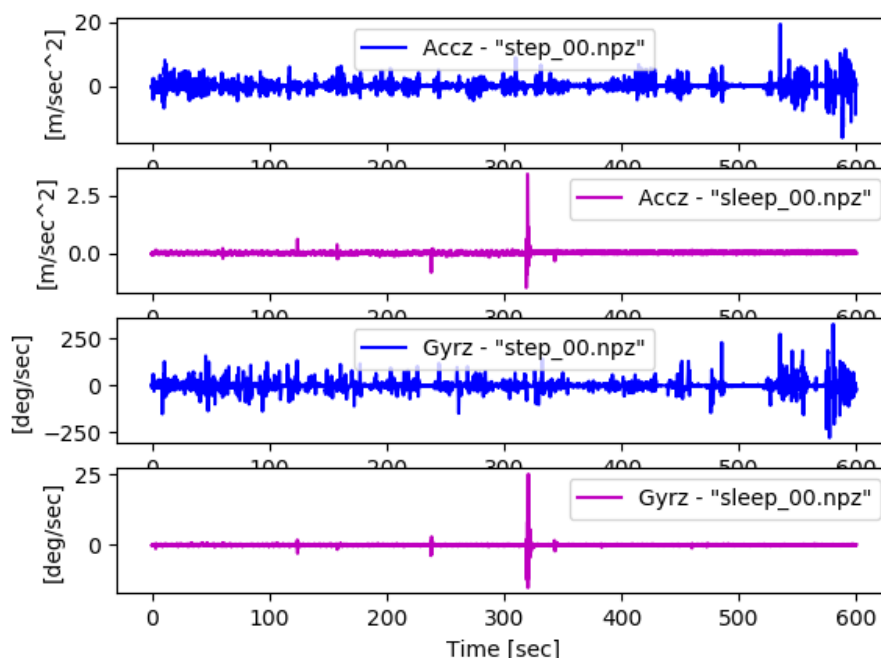


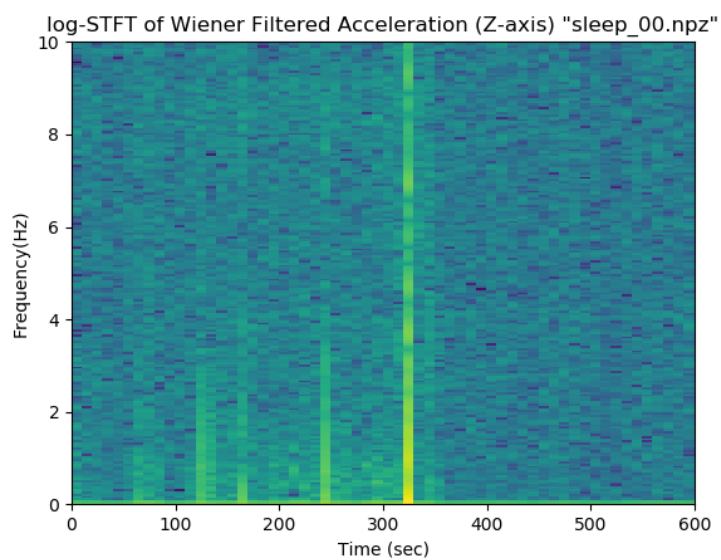
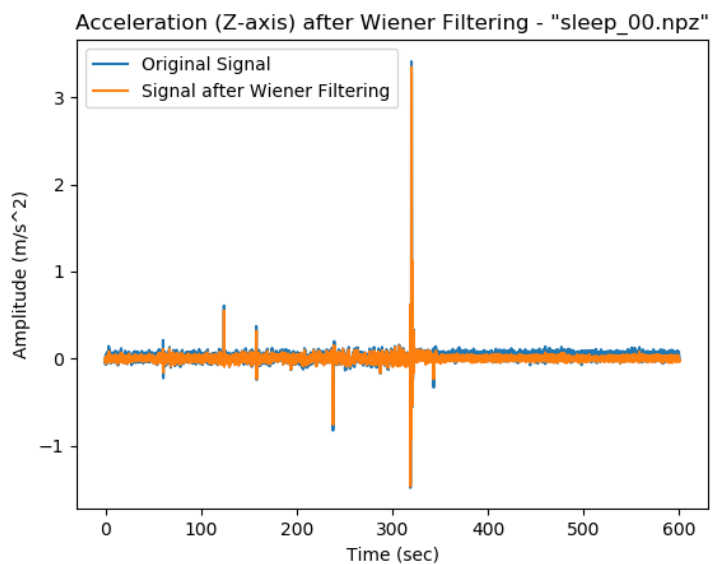
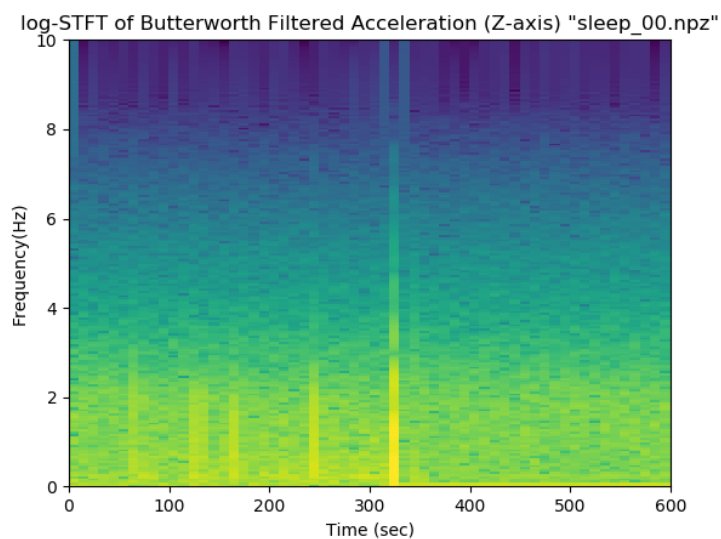
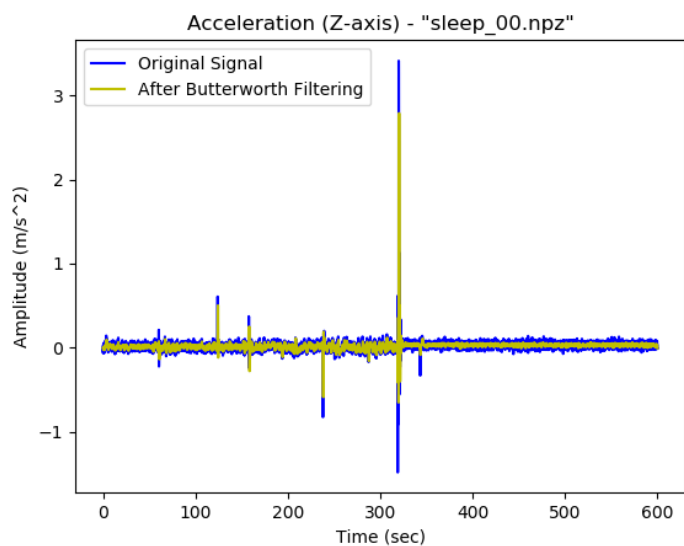
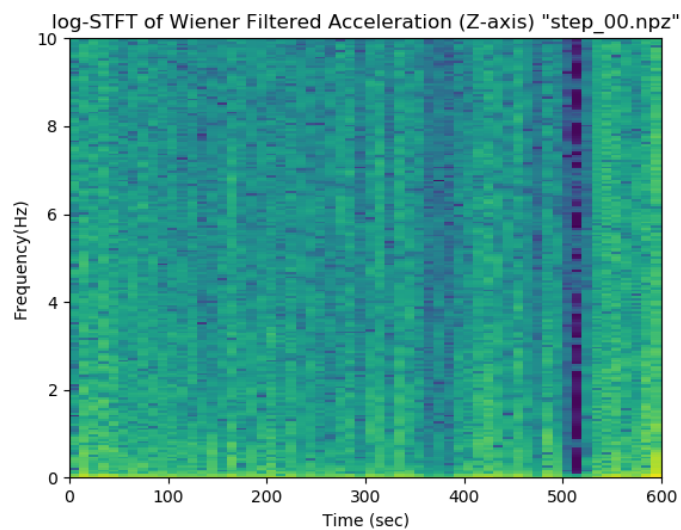
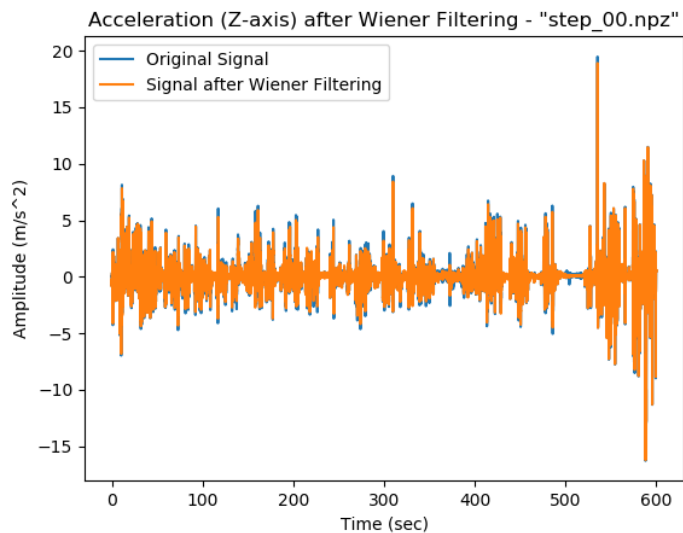
Παρατηρούμε ότι το Wiener φιλτράρισμα επιφέρει αρκετά ικανοποιητικά αποτελέσματα στην αντιμετώπιση του θορύβου, ενώ, σε αντίθεση με το Butterworth φιλτράρισμα, οι λωρίδες “κίνησης” στα διαγράμματα log-STFT είναι **πολύ πιο ευδιάκριτες**, καθώς δεν παρατηρείται η ίδια εξασθένιση στις συχνότητες εύρους 0–2Hz. Επομένως, η εφαρμογή των ψηφιακών φίλτρων Wiener φαίνεται να υπερτερεί στον εντοπισμό χρήσιμων φασματικών συνιστωσών έναντι αυτής των ψηφιακών φίλτρων Butterworth, για τα εν λόγω σήματα ύπνου.

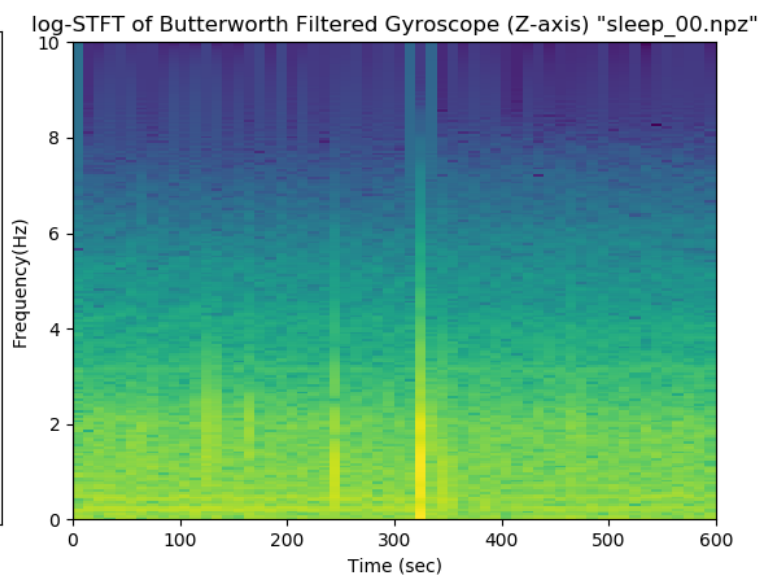
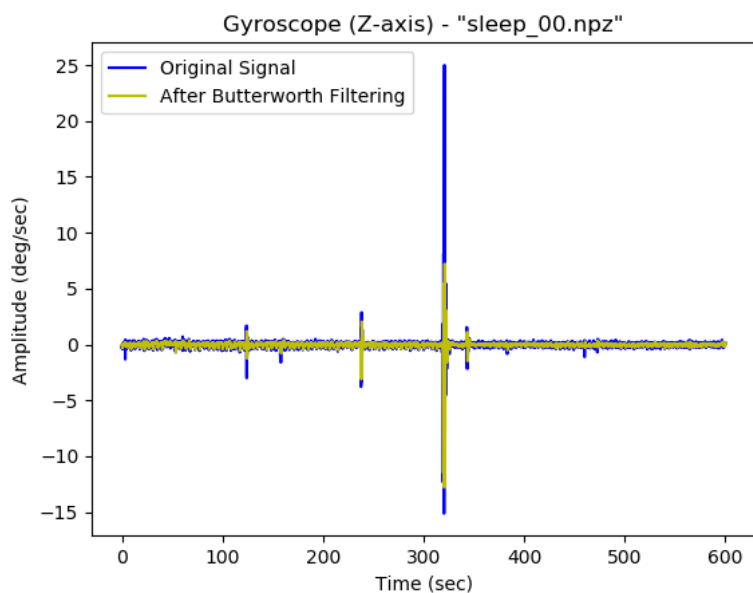
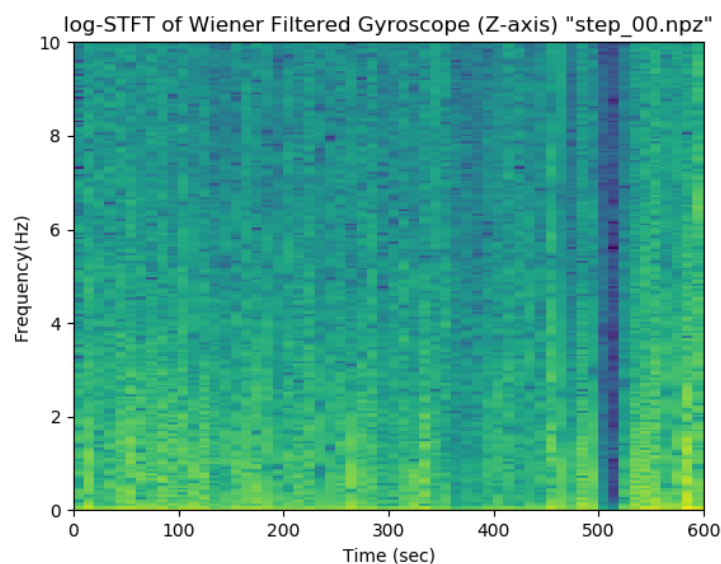
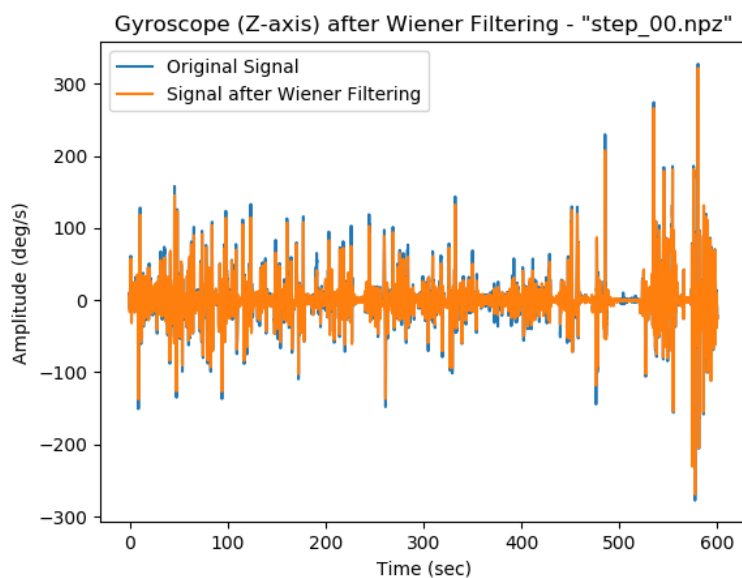
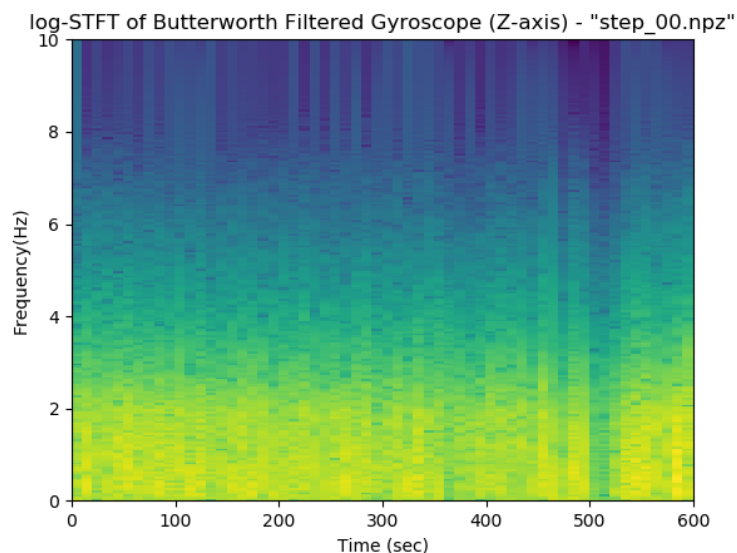
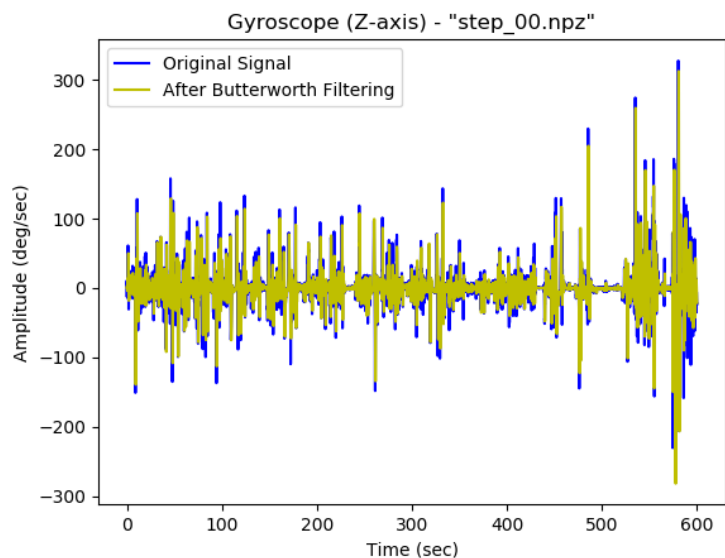
4.4 Εφαρμογή φίλτρων Wiener και Butterworth σε σήματα της επιλογής μας

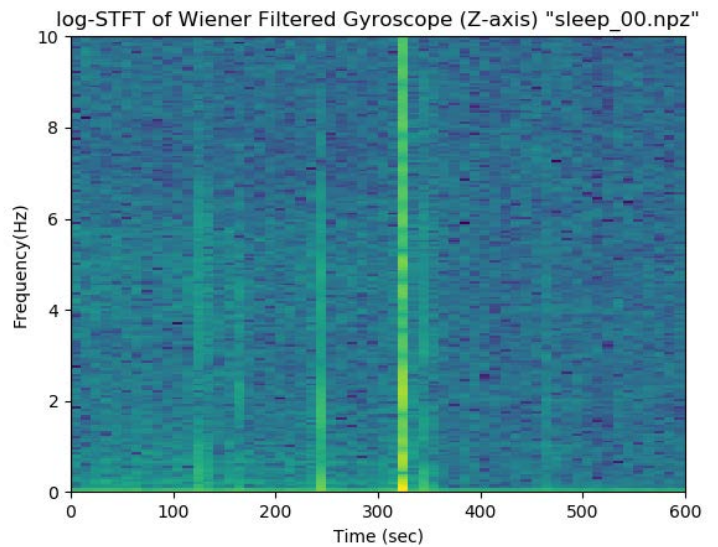
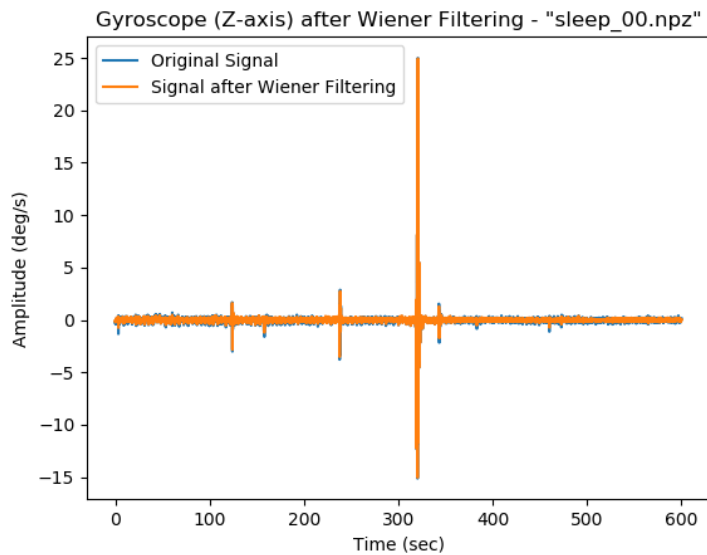
Στο τελευταίο μέρος της άσκησης, εφαρμόζουμε τα φίλτρα Butterworth και Wiener που υλοποιήσαμε σε έναν άξονα από 2 σήματα ύπνου και 2 σήματα περπατήματος της επιλογής μας. Πιο συγκεκριμένα, επιλέγουμε τα σήματα accz και gyrz των αρχείων "sleep_00.npz" και "step_00.npz":

Sleep and Step Signals of our choice









Παρατηρούμε ότι τα ψηφιακά φίλτρα Butterworth φαίνεται να αποθρομβοποιούν λίγο καλύτερα τα σήματα κίνησης από τα ψηφιακά φίλτρα Wiener. Ωστόσο, τα δεύτερα είναι καλύτερα στον προσδιορισμό των συχνοτικών λωρίδων του διαγράμματος log-STFT που αντιστοιχούν σε κίνηση – ιδίως σε σήματα ύπνου – καθώς όπως αναφέρθηκε παραπάνω τα φίλτρα Butterworth παρουσιάζουν μια εξασθένιση στις συχνότητες 0 -2Hz που καθιστά δυσδιάκριτη την συνολική φασματική εικόνα, κάτι το οποίο δεν παρατηρείται στα διαγράμματα log-STFT των Wiener φιλτραρισμένων σημάτων.

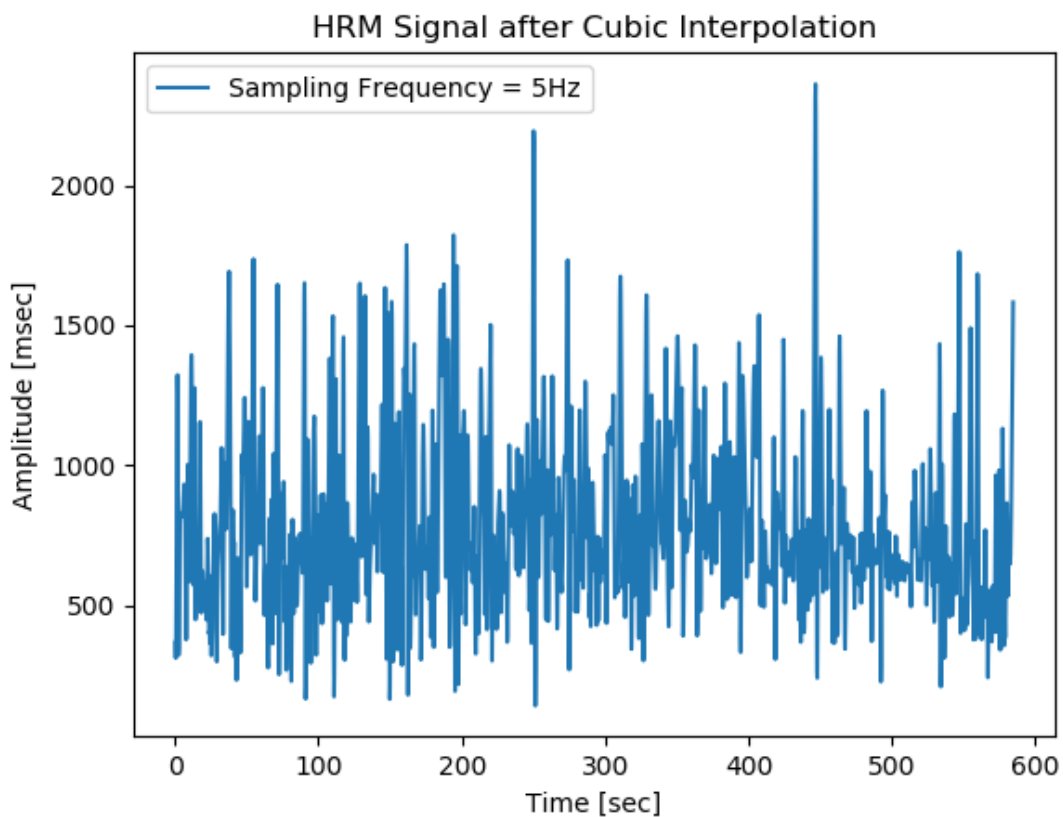
Μέρος 5^ο – Εκτίμηση του Φάσματος Ισχύος σήματος καρδιάς με Περιοδόγραμμα Schuster, Welch και Lomb-Scargle

Σκοπός Άσκησης

Στο τελευταίο μέρος της εργαστηριακής άσκησης επιχειρούμε να εκτιμήσουμε το φάσμα ισχύος με χρήση τριών διαφορετικών περιοδογραμμάτων από δύο σήματα τα οποία έχουν **μη σταθερό ρυθμό δειγματοληψίας**. Το πρώτο εξ αυτών βρίσκεται στο αρχείο `“hrm_orig.npy”` και αποτελεί ένα πραγματικό σήμα καρδιακής μεταβλητότητας, ενώ το δεύτερο δημιουργείται από εμάς, έτσι ώστε να έχει τυχαία δειγματοληψία και συγκεκριμένα peaks συχνότητων.

5.1 Φόρτωση αρχείου και cubic interpolation

Αρχικά φορτώνουμε το πραγματικό σήμα καρδιακής μεταβλητότητας που περιέχεται στο αρχείο `“hrm_orig.npy”` με χρήση της εντολής `numpy.load`. Στη συνέχεια, προσδιορίζουμε τις χρονικές στιγμές δειγματοληψίας του αρχικού σήματος καρδιακής μεταβλητότητας, με χρήση της εντολής `numpy.cumsum`, εφόσον οι τιμές αυτού υποδηλώνουν χρονικές διαφορές σε msec. Τέλος, με χρήση της εντολής `interp1d` της βιβλιοθήκης `scipy.interpolate`, εφαρμόζουμε **cubic interpolation** στο σήμα, ώστε αυτό να έχει ομοιόμορφη συχνότητα δειγματοληψίας ίση με **fs1 = 5Hz**. Ειδικότερα:

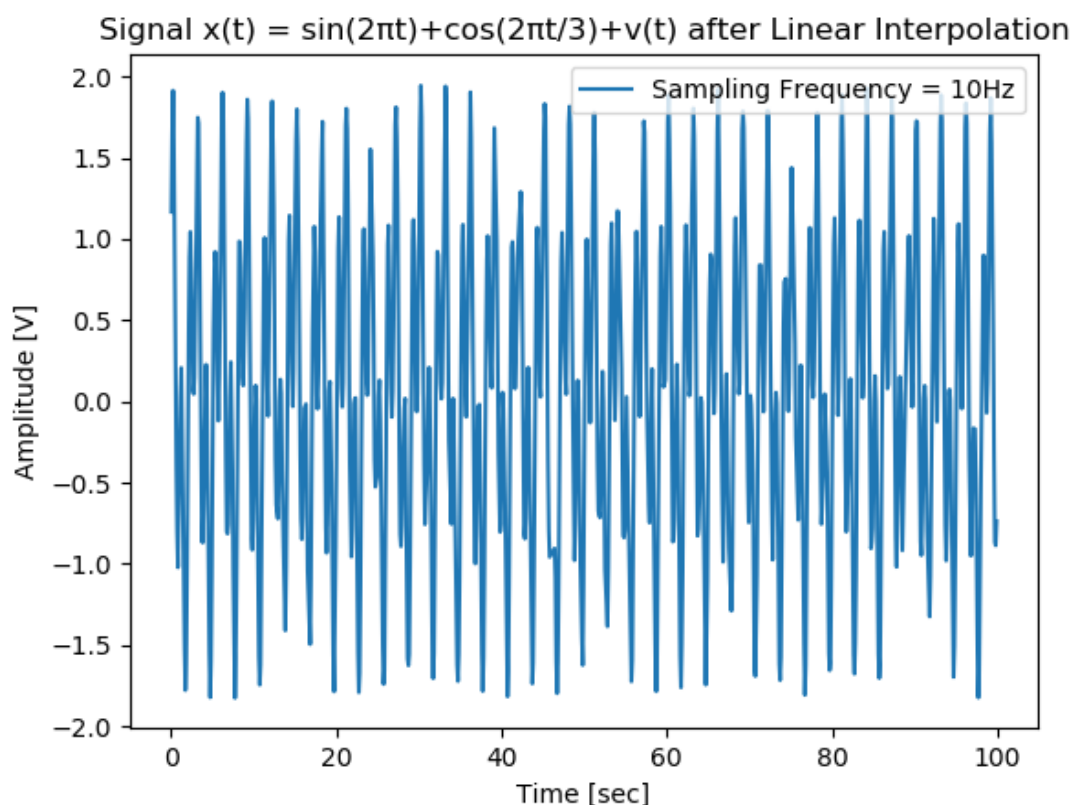


5.2 Δημιουργία σήματος $x(t)$ και linear interpolation

Ακολουθως, αφού δημιουργήσουμε ένα τυχαίο διάνυσμα μεγέθους $N = 1000$ δειγμάτων το οποίο να περιέχει τυχαίους ταξινομημένους χρόνους δειγματοληψία t_n από 0 έως 100 sec, με χρήση της εντολής `np.sort(100*np.random.rand(N), kind='mergesort')`, παίρνουμε τις τιμές του συνεχούς σήματος:

$$x(t) = \sin(2\pi t) + \cos\left(\frac{2\pi t}{3}\right) + v(t)$$

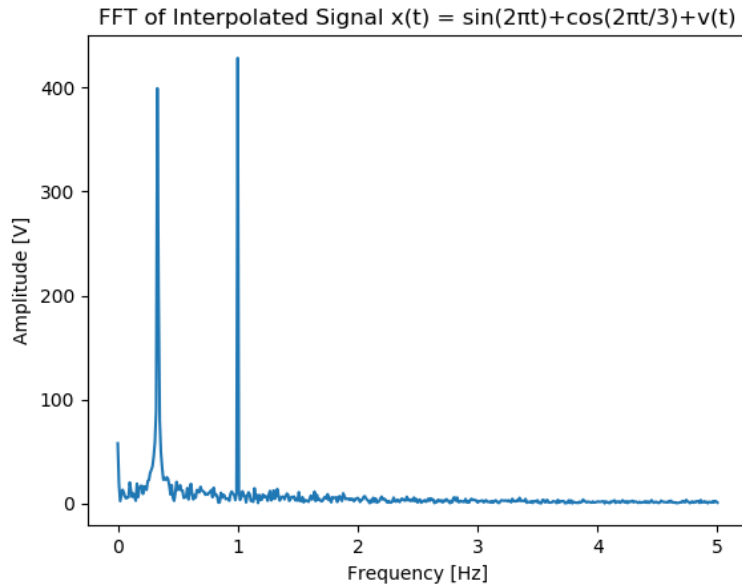
στις στιγμές αυτές t_n . Η τυχαία μεταβλητή θορύβου $v(t)$ παίρνει ομοιόμορφα τυχαίες τιμές εντός του διαστήματος $[0, 0.1]$. Στη συνέχεια, ακολουθώντας ανάλογη διαδικασία με αυτή του προηγούμενου ερωτήματος, εφαρμόζουμε **linear interpolation** στο σήμα x , ώστε αυτό να έχει ομοιόμορφο ρυθμό δειγματοληψίας ίσο με **fs2 = 10Hz**. Ειδικότερα:



Αν θέλαμε να προσδιορίσουμε τον ελάχιστο σταθερό ρυθμό δειγματοληψίας που θα μπορούσαμε να χρησιμοποιήσουμε, ώστε να καθίσταται εφικτός ο εντοπισμός των δύο κύριων συχνοτήτων του σήματος, στις θέσεις 1Hz και 1/3Hz, τότε χρειάζεται να στηριχθούμε στο **Θεώρημα Nyquist-Shannon**. Ειδικότερα, ακριβώς επειδή η χρήσιμη πληροφορία φτάνει μέχρι και την υψίσυχνη τιμή $B = 1\text{Hz}$, τότε μπορούμε να αποφύγουμε φαινόμενα **aliasing**, αν επιλέξουμε σταθερό ρυθμό δειγματοληψίας:

$$f_s > 2B \implies f_{s_{\min}} = 2\text{Hz}$$

Τα παραπάνω επαληθεύονται αν κοιτάξουμε και το διάγραμμα της απόλυτης τιμής του Μετασχηματισμού Fourier του εν λόγω σήματος:

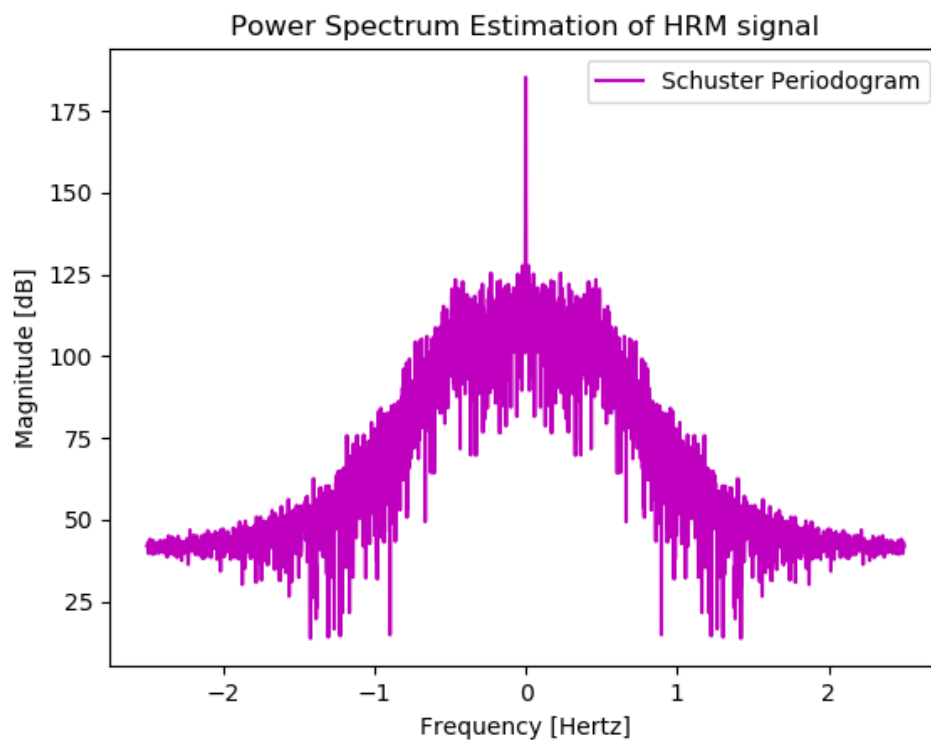


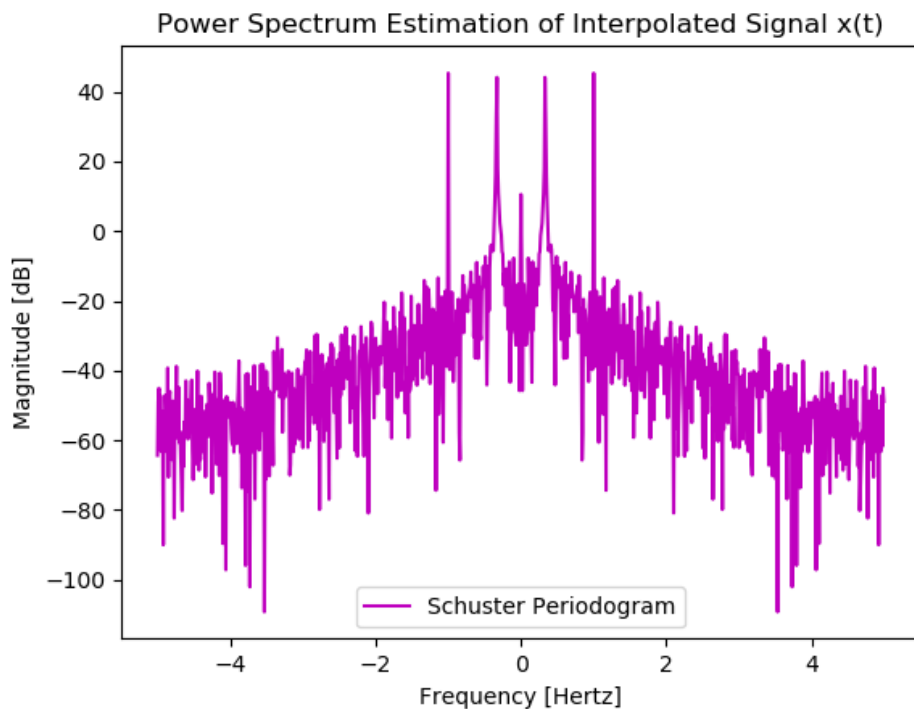
5.3 Περιοδόγραμμα Schuster

Με χρήση του DFT υλοποιούμε και σχεδιάζουμε την εκτίμηση του φάσματος ισχύος των δύο interpolated σημάτων μέσω του κλασικού περιοδογράμματος Schuster, που δίνεται από τον τύπο:

$$P_{\text{per}}(e^{j\omega}) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \right|^2 \quad \text{Eq. (5.1)}$$

Παρατίθενται οι εκτιμήσεις των φασμάτων ισχύος σε λογαριθμική κλίμακα:





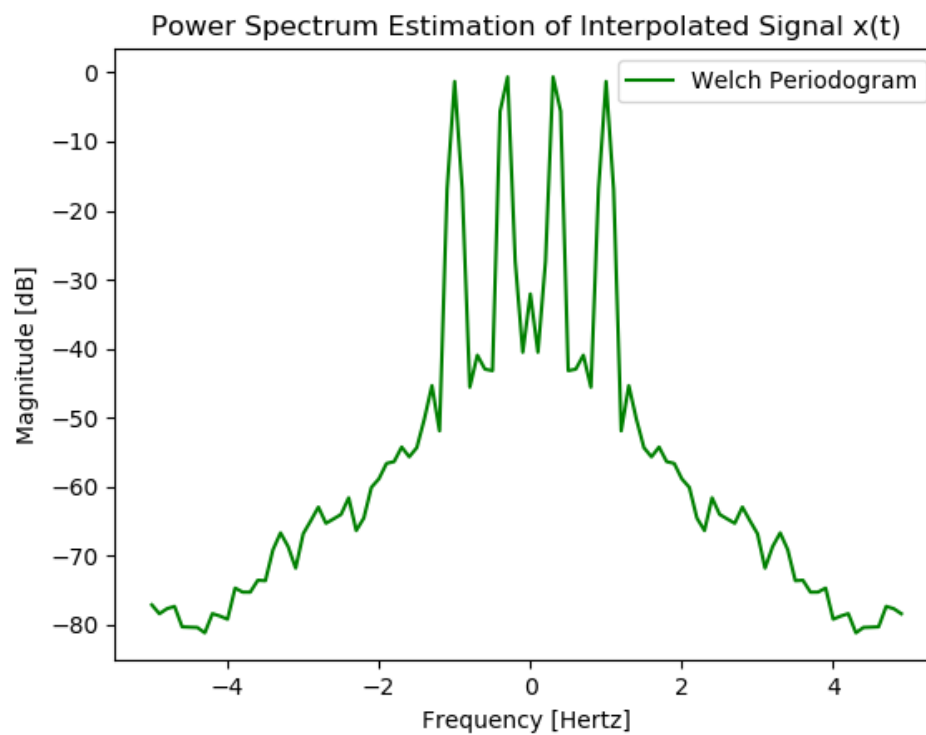
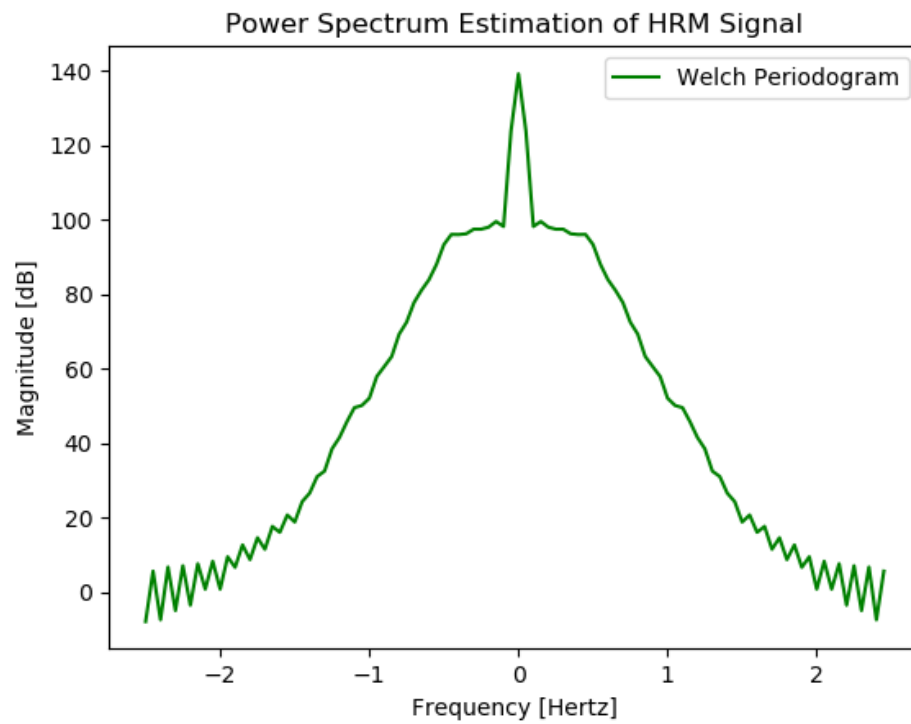
5.3 Περιοδόγραμμα Welch

Στη συνέχεια, υλοποιούμε και σχεδιάζουμε την εκτίμηση του φάσματος ισχύος των δύο interpolated σημάτων, χρησιμοποιώντας αυτήν τη φορά το περιοδόγραμμα Welch. Ακολουθούμε τον εξής αλγόριθμο:

- Χωρίζουμε το σήμα σε τμήματα μήκους $L = 100$ δειγμάτων, με επικάλυψη $D = 50$ δείγματα, τα οποία παραθυροποιούμε με παράθυρα Bartlett.
- Για κάθε ένα εξ αυτών των τμημάτων υπολογίζουμε το κλασικό περιοδόγραμμα με χρήση του τύπου (5.1).
- Στο τέλος, υπολογίζουμε τον αριθμητικό μέσο των επιμέρους περιοδογραμμάτων ως προς τον χρόνο.

Παρατηρώντας καλύτερα τις εκτιμήσεις των φασμάτων ισχύος που προκύπτουν, βλέπουμε ότι το περιοδόγραμμα Welch είναι **πιο αποτελεσματικό στην αντιμετώπιση του θορύβου** από το κλασικό περιοδόγραμμα Schuster, ωστόσο **υστερεί στο frequency resolution**, καθώς οι κύριοι λοβοί είναι μεγαλύτερου εύρους. Οι διαφορές αυτές, οφείλονται στο γεγονός ότι το περιοδόγραμμα Welch, σε αντίθεση με το κλασικό περιοδόγραμμα, χρησιμοποιεί παράθυρα, τα οποία έχουν μεγαλύτερο πλάτος στο κέντρο παρά στις άκρες, με κατάλληλο overlap.

Παρατίθενται οι εκτιμήσεις των φασμάτων ισχύος με περιοδόγραμμα Welch σε λογαριθμική κλίμακα:



5.5 Περιοδόγραμμα Lomb – Scargle

Στο τελευταίο μέρος της άσκησης επιχειρούμε να υπολογίσουμε τις εκτιμήσεις των φασμάτων ισχύος των αρχικών σημάτων, προτού εφαρμόσουμε σε αυτά κάποιο interpolation – και άρα αποφεύγοντας τον επιπλέον θόρυβο που αυτό θα εισήγαγε στο σήμα μας. Για να το επιτύχουμε αυτό, χρησιμοποιούμε το περιοδόγραμμα Lomb-Scargle, το οποίο αποτελεί μια μέθοδο εκτίμησης του φάσματος ισχύος που μπορεί να εφαρμοστεί σε δεδομένα που δεν έχουν παρατηρηθεί σε ισοδιατεταγμένα χρονικά διαστήματα (και άρα **δεν έχουν ομοιόμορφο ρυθμό δειγματοληψία**).

Το εν λόγω περιοδόγραμμα δίνεται από τον τύπο:

$P_{LS}(\Omega) = \frac{1}{2} \left\{ \frac{[\sum_{n=0}^{N-1} x[n] \cos(\Omega(tn-\tau))]^2}{\sum_{n=0}^{N-1} \cos(\Omega(tn-\tau))^2} + \frac{[\sum_{n=0}^{N-1} x[n] \sin(\Omega(tn-\tau))]^2}{\sum_{n=0}^{N-1} \sin(\Omega(tn-\tau))^2} \right\}$	Eq. (5.2)
---	-----------

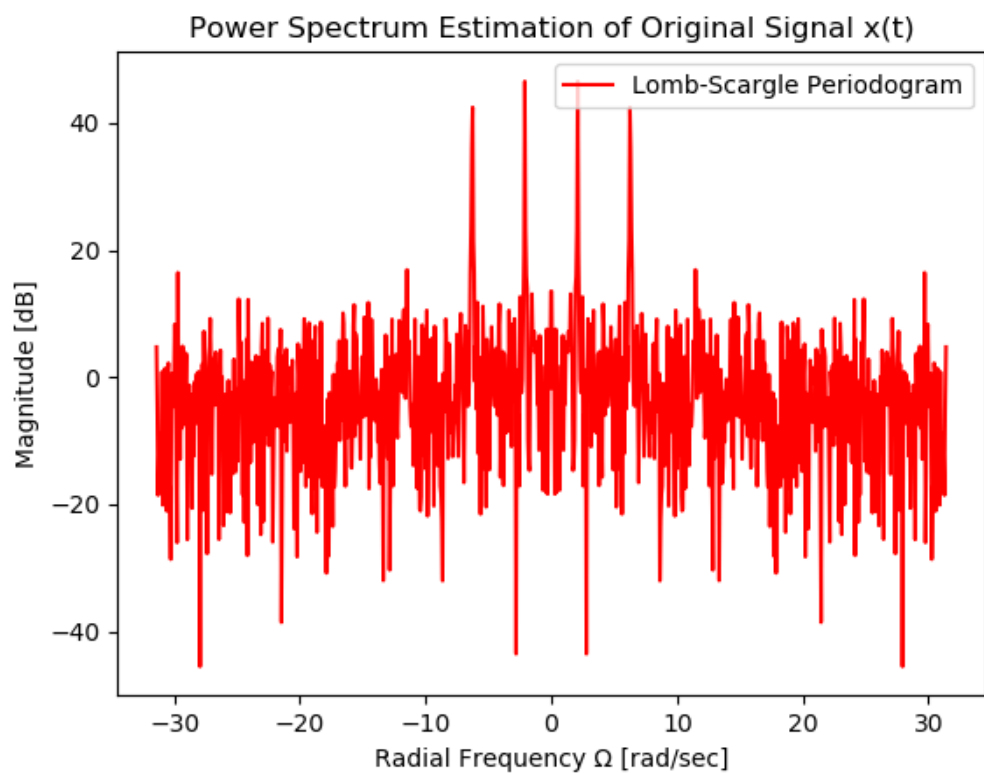
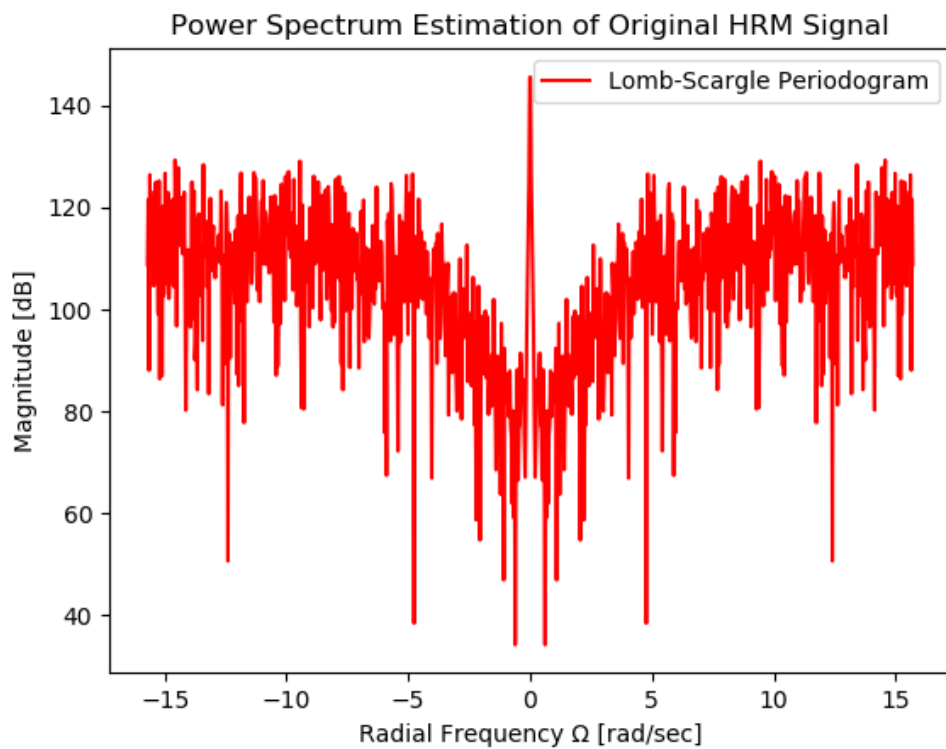
Όπου:

$\tau = \frac{1}{2\Omega} \tan^{-1} \left(\frac{\sum_{n=0}^{N-1} \sin(2\Omega tn)}{\sum_{n=0}^{N-1} \cos(2\Omega tn)} \right)$	Eq. (5.3)
---	-----------

Παρακάτω, υλοποιούμε σε γλώσσα προγραμματισμού *python* τη συνάρτηση ***lombscargle***, η οποία παίρνει ως ορίσματα το διάνυσμα που περιέχει τις χρονικές στιγμές *tn*, στις οποίες έγιναν οι παρατηρήσεις, το διάνυσμα *x[n]* το οποίο περιέχει την τιμή του σήματος τη στιγμή *tn* και τις γωνιακές συχνότητες *Ω* στις οποίες θέλουμε να υπολογίσουμε το φάσμα ισχύος, και επιστρέφει την τιμή του περιοδογράμματος LS στις ζητούμενες γωνιακές συχνότητες:

```
def lombscargle(times,sig,omega):  
    N = np.size(sig)  
    a,b=0,0  
    for i in range(N):  
        a = a + np.sin(2*omega*times[i])  
        b = b + np.cos(2*omega*times[i])  
    taf = np.arctan(a/b)/(2*omega)  
  
    A,B,C,D=0,0,0,0  
    for i in range(N):  
        A = A + sig[i]*np.cos(omega*(times[i]-taf))  
        B = B + sig[i]*np.sin(omega*(times[i]-taf))  
        C = C + np.cos(omega*(times[i]-taf))**2  
        D = D + np.sin(omega*(times[i]-taf))**2  
    Pls = ((A**2)/C+(B**2)/D)/2  
    return Pls
```

Παρατίθενται οι εκτιμήσεις των φασμάτων ισχύος με περιοδόγραμμα LS σε λογαριθμική κλίμακα των σημάτων πριν το interpolation, σε συνάρτηση με τη γωνιακή συχνότητα Ω :

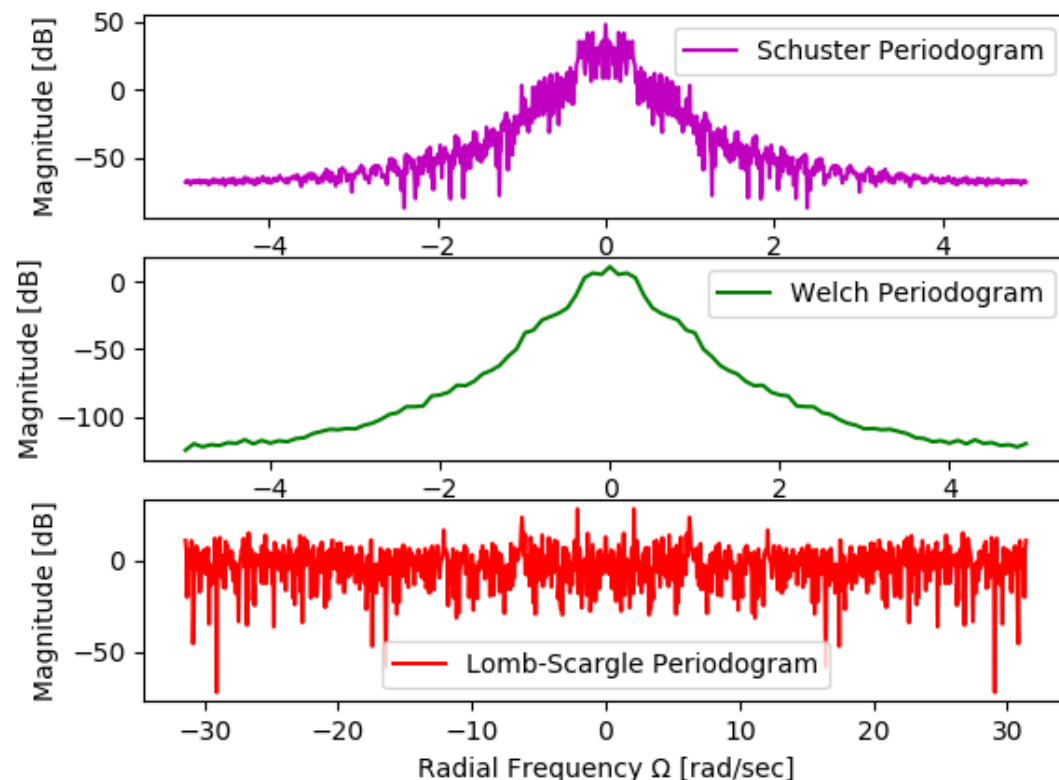


Το περιοδόγραμμα Lomb-Scargle καταφέρνει να εντοπίσει τις κύριες συχνότητες των σημάτων με **λίγοτερο θόρυβο** από το κλασικό περιοδόγραμμα Schuster, και με **καλύτερο frequency resolution** από το περιοδόγραμμα Welch. Ένα ακόμα πλεονέκτημα του εν λόγω περιοδαγραμμάτος είναι ότι όλα αυτά τα καταφέρνει δίχως να χρειάζεται να κάνουμε interpolation στα αρχικά μας σήματα, κάτι που ως διαδικασία προσθέτει επιπλέον θόρυβο.

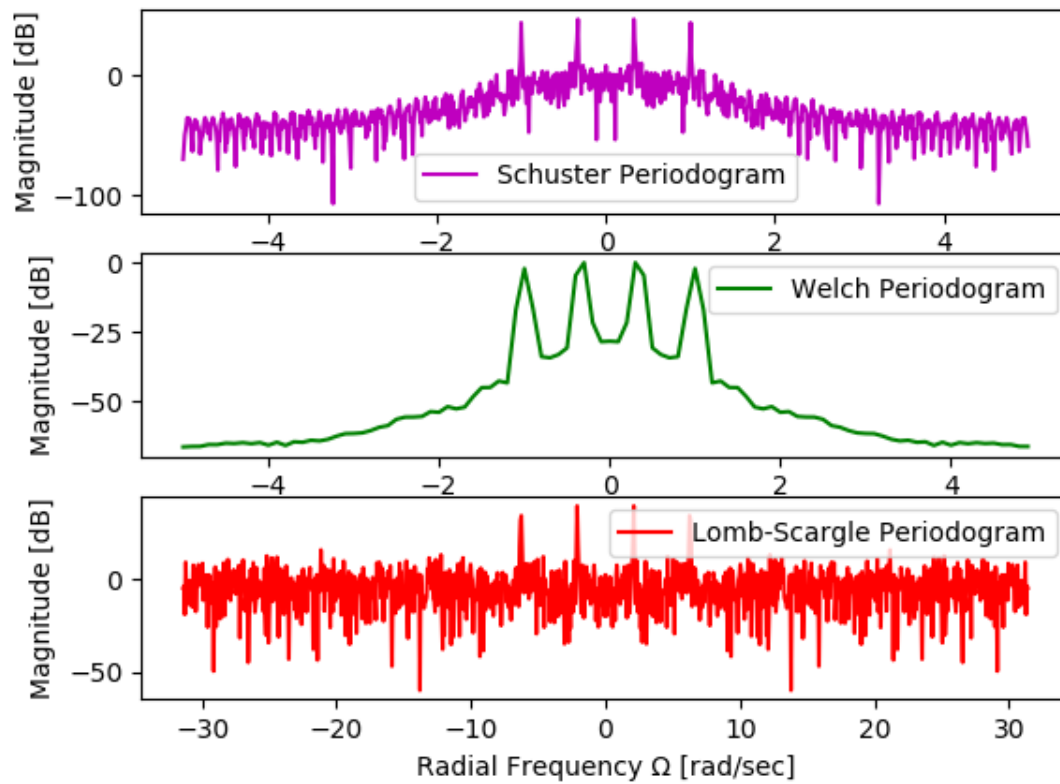
Προκειμένου να φανεί περισσότερο η υπεροχή του LS περιοδογράμματος, μελετάμε την επίδραση του μεγέθους N του σιανύσματος που δημιουργήσαμε στο ερώτημα 5.2, στην ικανότητα των τριών διαφορετικών περιοδογραμμάτων να εντοπίσουν τις κύριες συχνότητες των σημάτων. Ειδικότερα, εξετάζουμε τις τιμές $N = 100, 500, 5000, 10000$, παραθέτοντας σε κοινό διάγραμμα όλες τις εκτιμήσεις φασμάτων ισχύος του κάθε περιοδογράμματος:

(**Σημείωση:** Οι 2 πρώτες εκτιμήσεις είναι σε συνάρτηση της συχνότητας f , ενώ η Τρίτη σε συνάρτηση με τη γωνιακή συχνότητα Ω)

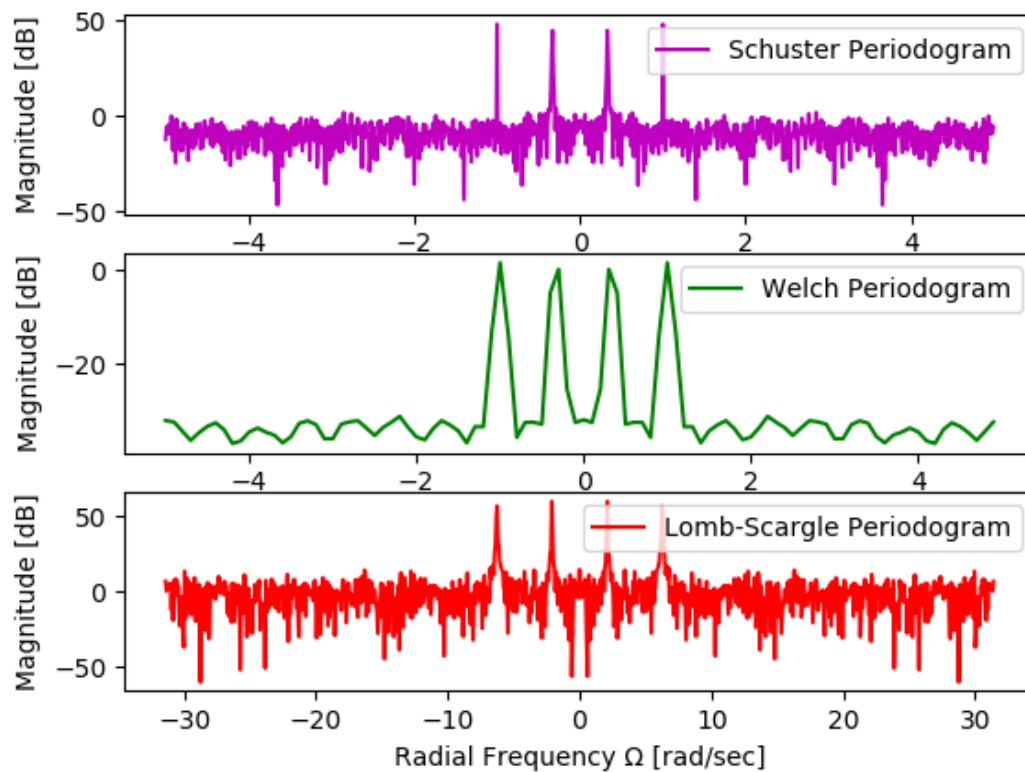
Power Spectrum Estimations of Signal $x(t)$ for $N = 100$



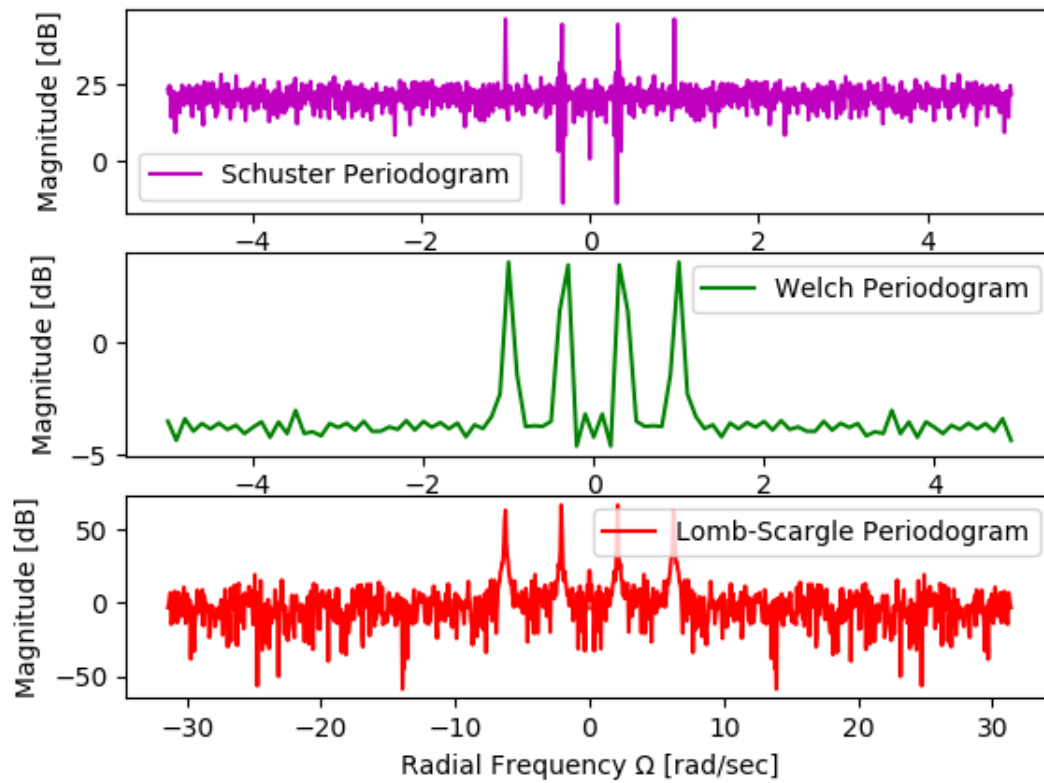
Power Spectrum Estimations of Signal $x(t)$ for $N = 500$



Power Spectrum Estimations of Signal $x(t)$ for $N = 5000$



Power Spectrum Estimations of Signal $x(t)$ for $N = 10000$



Παρατηρούμε ότι για χαμηλές τιμές του N , τα περιοδογράμματα Schuster και Welch δυσκολεύονται έως και αποτυγχάνουν να εντοπίσουν τις κύριες συχνότητες των φασμάτων των δύο σημάτων. Κάτι τέτοιο δεν συμβαίνει με το περιοδόγραμμα Lomb-Scargle, το οποίο ακόμα και για την τιμή $N = 100$, καταφέρνει να διακρίνει τις κύριες συχνότητες στην εκτίμηση του φάσματος ισχύος. Ως εκ τούτου, το τελευταίο περιοδόγραμμα ξεχωρίζει και υπερτερεί έναντι των άλλων δύο.

Βιβλιογραφία

- [1] [Ψηφιακή Επεξεργασία Σήματος](#), Alan V. Oppenheim, Ronald W. Schaffer, Εκδόσεις Φούντας, 2012
- [2] [Ψηφιακή Επεξεργασία Σήματος](#), Monson H. Hayes, Εκδόσεις Τζιόλα, 2014
- [3] [Ψηφιακή Επεξεργασία Φωνής: Θεωρία και Εφαρμογές](#), Lawrence R. Rabiner, Ronald W. Schaffer, 1η Έκδοση, Εκδόσεις Π.Χ.Πασχαλίδης, 2011

References

- [1] G. Evangelopoulos and P. Maragos. Multiband modulation energy tracking for noisy speech detection. IEEE Transactions on Audio, Speech, and Language Processing, 14:2024–2038, 2006.
- [2] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. IEEE Transactions on Multimedia, 15:1553–1568, 2013.
- [3] N. R. Lomb. Least-squares frequency analysis of unequally spaced data. Astrophysics and Space Science, 39:447–462, 1976.
- [4] P. Maragos. Summer School on Speech Signal Processing, S4P Lecture Slides: Lecture I: Nonlinear aspects of speech production: Modulations and energy operators, Interspeech2018. <http://cvsp.cs.ntua.gr/interspeech2018.shtm>, 2018.
- [5] P. Maragos, J. F. Kaiser, and T. F. Quatieri. Energy separation in signal modulations with application to speech analysis. IEEE Transactions on Signal Processing, 41:3024–3051, 1993.
- [6] J. D. Scargle. Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data. The Astrophysical Journal, 263:835–853, 1982.
- [7] R. Townsend. Fast calculation of the lomb-scargle periodogram using graphics processing units. The Astrophysical Journal Supplement Series, 191:247, 2010.
- [8] J. T. VanderPlas. Understanding the lomb–scargle periodogram. The Astrophysical Journal Supplement Series, 236:16, 2018.