Μέρος 1.1

Στο κομμάτι 1.1 φορτώνουμε την βιβλιοθήκη numpy και αποθηκεύοθμε το αρχείο που ζητείται με συναρτήσεις της βιβλιοθήκης .

In [2]:

```
import numpy as np

data = np.load('step_00.npz')
data.files
```

Out[2]:

```
['acc', 'gyr', 'hrm']
```

In [3]:

```
print(np.shape(data['acc'])) #Παρατηρώ ότι το αρχείο acc περιέχει 3 σήματα των 11992 δειγμάτων ένα για κάθε άξονα
print(np.shape(data['gyr'])) #Παρόμοια για το αρχείο του γυροσκοπείου
print(np.shape(data['hrm'])) #Το αρχείο που περιέχει το heart rate αποτελείται από 2998 δείγματα
```

```
(11992, 3)
(11992, 3)
(2998,)
```

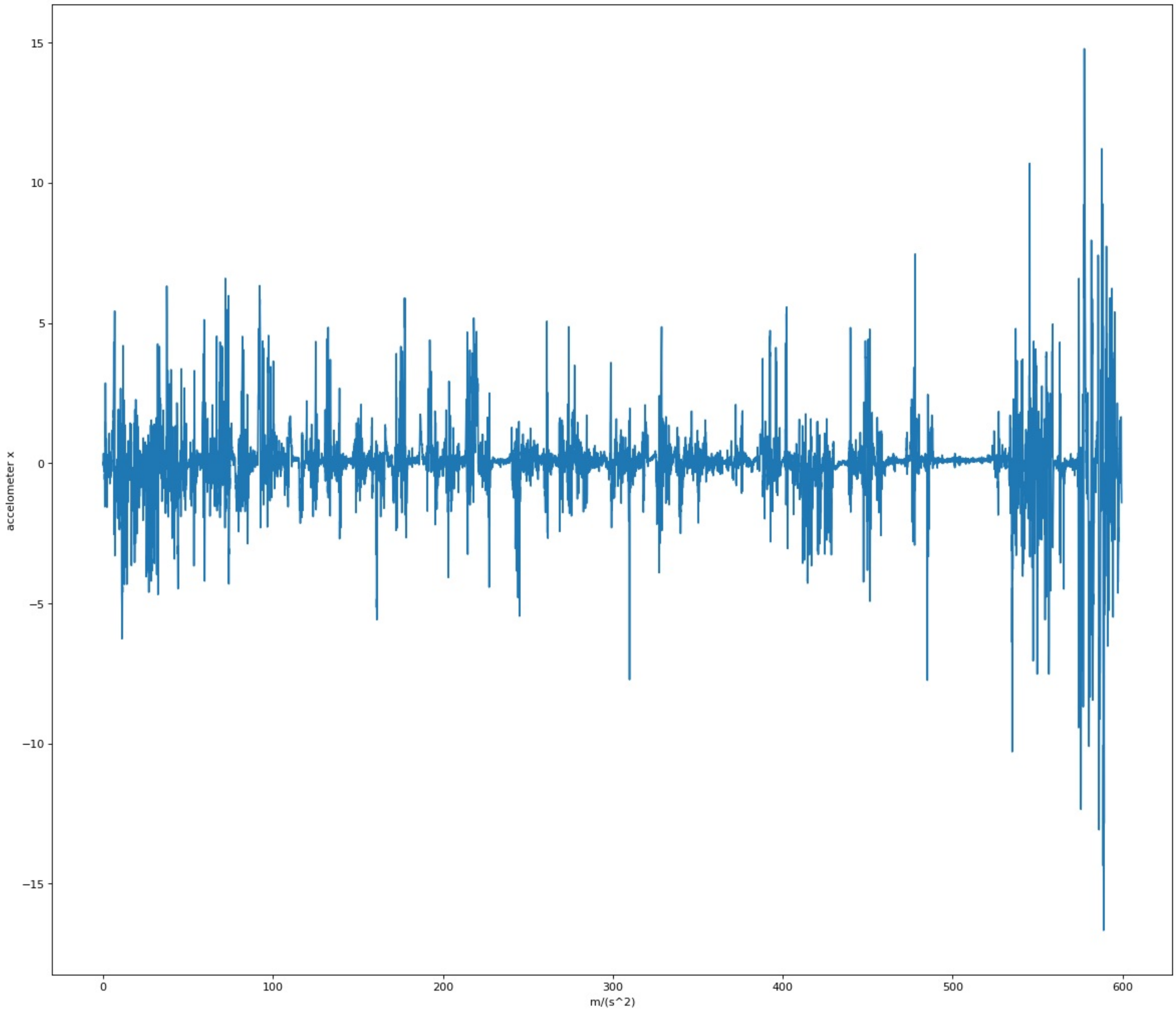
In [4]:

```
acc0=[] #Αποθηκεύω τα δείγματα του αξελερόμετρου για τον άξονα των X εδώ
acc1=[] #Για τον άξονα των Y
acc2=[] #Για τον άξονα των Z
for x in data['acc'] :
    acc0.append(x[0])
    acc1.append(x[1])
    acc2.append(x[2])

gyr0=[] #Αποθηκεύω τα δείγματα του γυροσκόπειου για τον άξονα των X εδώ
gyr1=[] #Για τον άξονα των Y
gyr2=[] #Για τον άξονα των Z
for x in data['gyr'] :
    gyr0.append(x[0])
    gyr1.append(x[1])
    gyr2.append(x[2])
```

Με το διάνυσμα t20 συμβολίζουμε τον χρόνο για ρυθμούς δειγματοληψίας 20 HZ που σημαίνει ένα δείγμα κάθε 50 ms οπότε για αυτό και το step στην κατασκευή του διανύσματος χρόνου είναι 0.05 . Επίσης φτάνουμε μέχρι την τιμή 559.6 καθώς ο αριθμός των δειγμάτων αντιστοιχεί σε τόσα δευτερόλεπτα και όχι ακριβώς σε 600 δηλαδή 10 λεπτά .

In [5]:

```
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
t20=np.arange(0.,599.6,0.05)
plt.plot(t20,acc0[:])

plt.ylabel('accelometer x')
plt.xlabel('m/(s^2)')
plt.show()
```



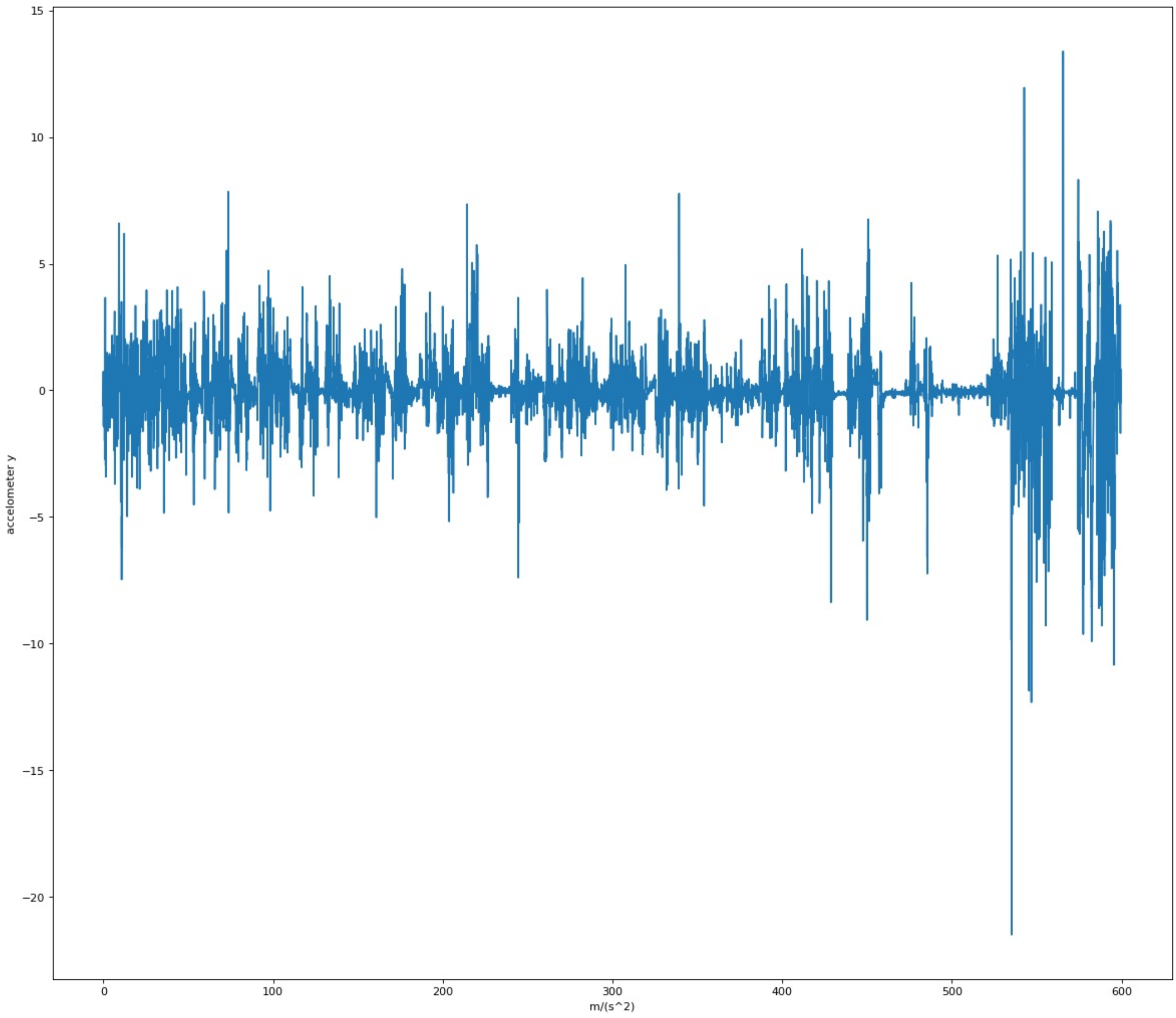
Γραφική παράσταση για το αξελερόμετρο στον άξονα των Y .

In [5]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
plt.plot(t20,acc1[:])
plt.ylabel('accelometer y')
plt.xlabel('m/(s^2)')
```

Out[5]:

```
Text(0.5, 0, 'm/(s^2)')
```

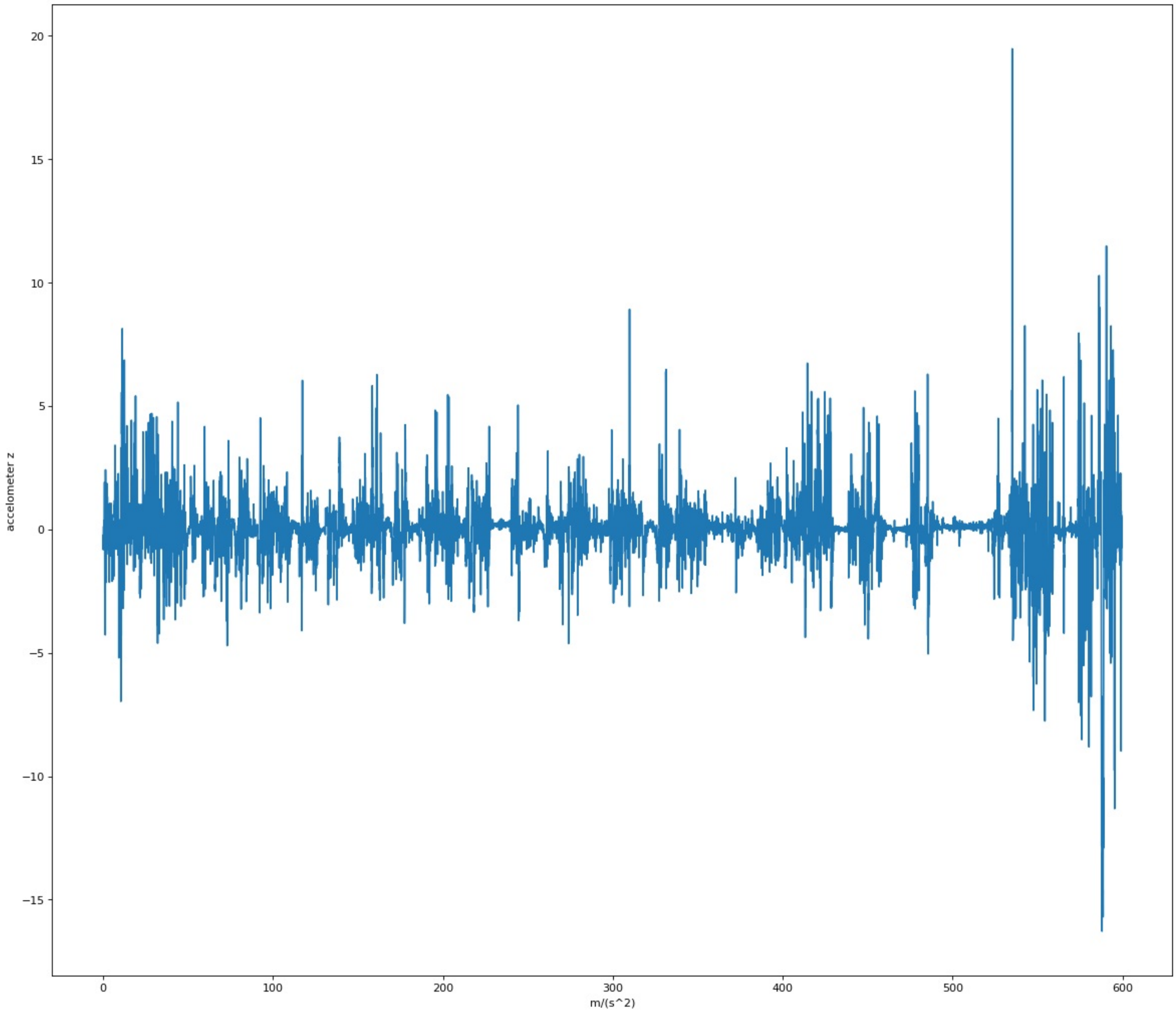


In [6]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
#γραφική του αξελερόμετρου για τον άξονα των Z
plt.plot(t20,acc2[:])
plt.ylabel('accelometer z')
plt.xlabel('m/(s^2)')
```

Out[6]:

```
Text(0.5, 0, 'm/(s^2)')
```

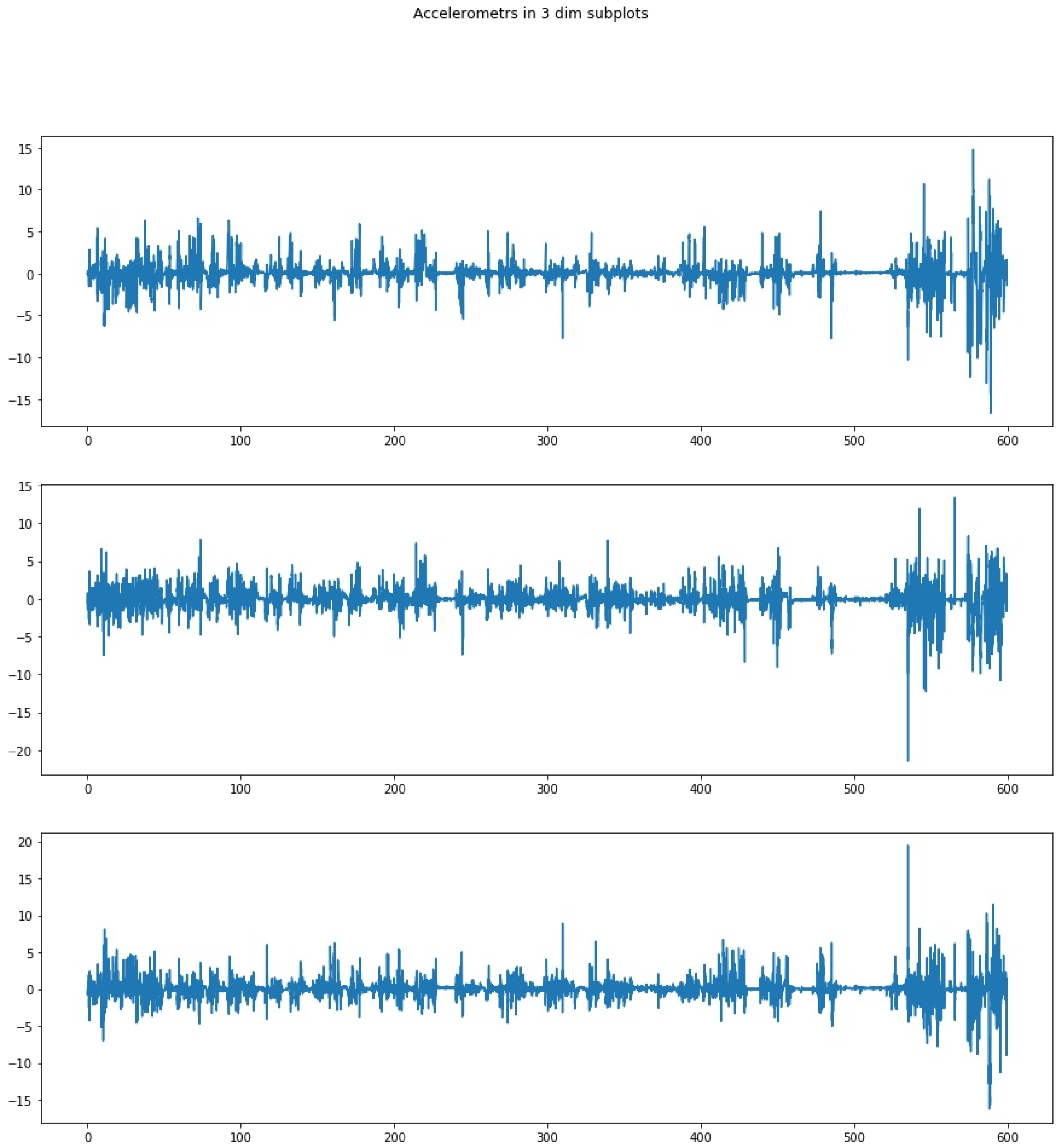


In [7]:

```
fig, axs = plt.subplots(3,figsize=(15,15)) #κοινή γραφική παράσταση και για τους 3 άξονες
fig.suptitle('Accelerometrs in 3 dim subplots')
axs[0].plot(t20, acc0)
axs[1].plot(t20, acc1)
axs[2].plot(t20, acc2)
```

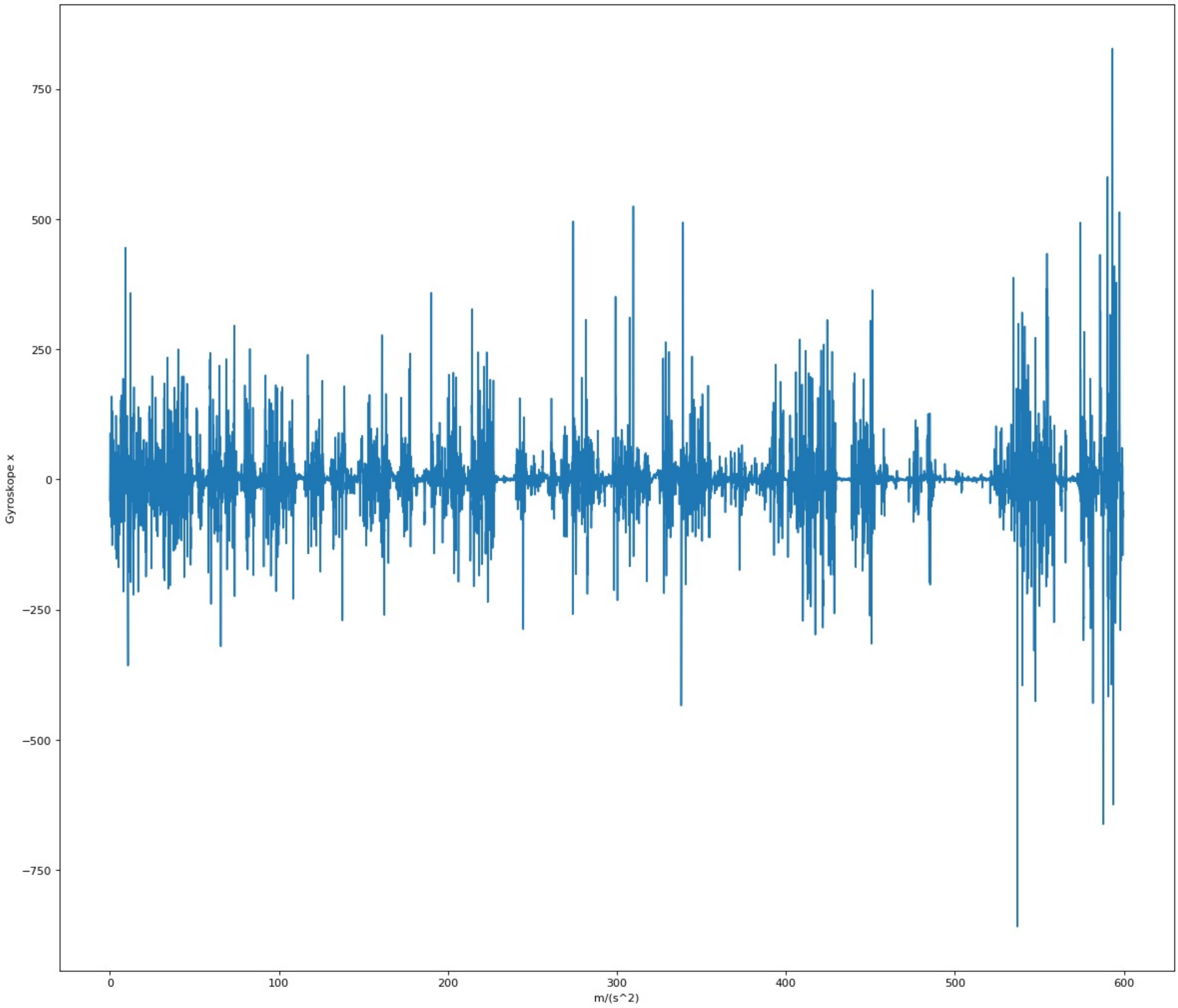
Out[7]:

```
[<matplotlib.lines.Line2D at 0x10421f810>]
```



In [8]:

```
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
t20=np.arange(0.,599.6,0.05)
plt.plot(t20,gyr0[:])
#Γραφική του γυροσκοπείου για τον άξονα των X
plt.ylabel('Gyroskope x')
plt.xlabel('m/(s^2)')
plt.show()
```

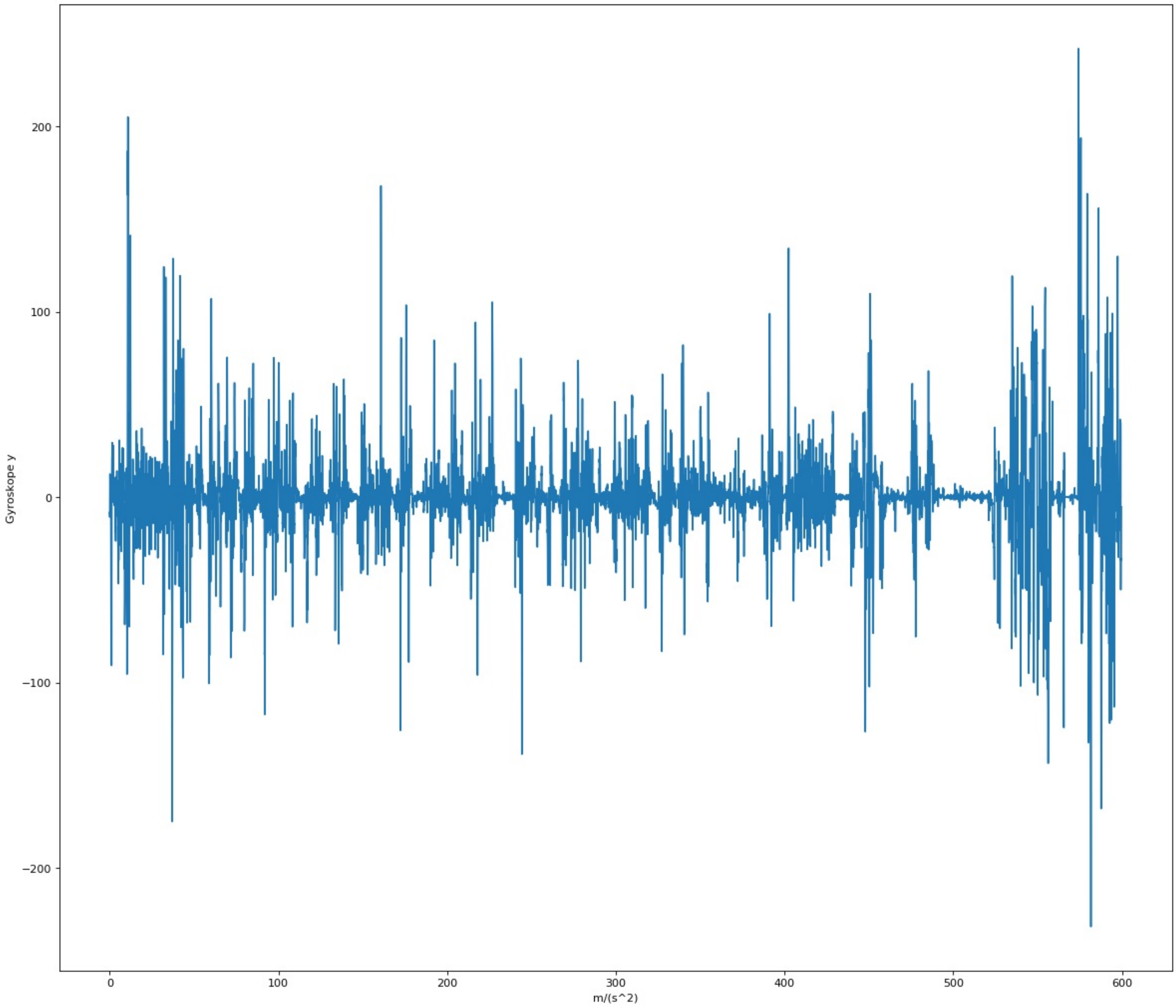


In [9]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
plt.plot(t20,gyr1[:])
#Γραφική του γυροσκοπείου για τον άξονα των Y
plt.ylabel('Gyroskope y')
plt.xlabel('m/(s^2)')
```

Out[9]:

```
Text(0.5, 0, 'm/(s^2)')
```

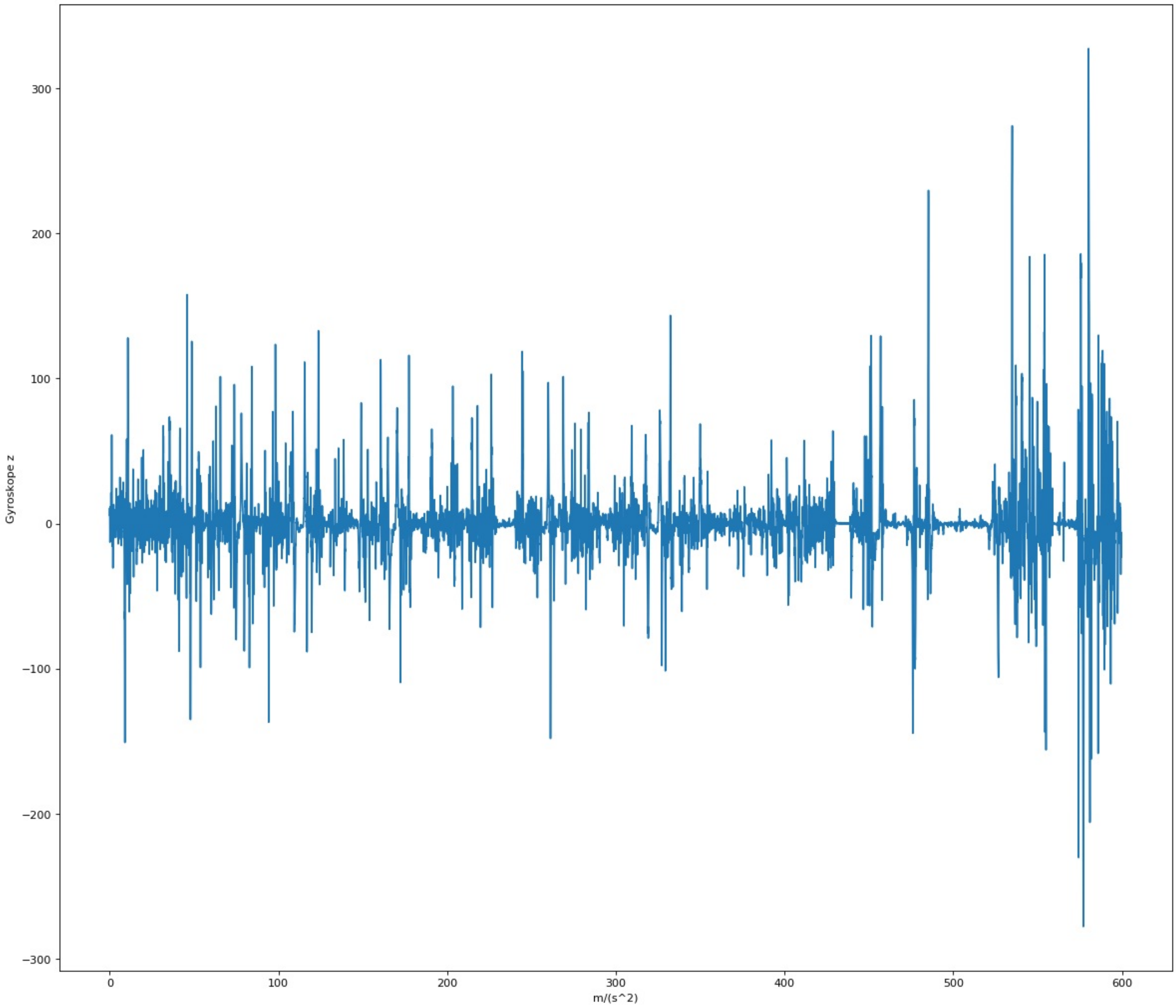


In [10]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
#Γραφική του γυροσκοπείου για τον άξονα των Ζ
plt.plot(t20,gyr2[:])
plt.ylabel('Gyroskope z')
plt.xlabel('m/(s^2)')
```

Out[10]:

```
Text(0.5, 0, 'm/(s^2)')
```

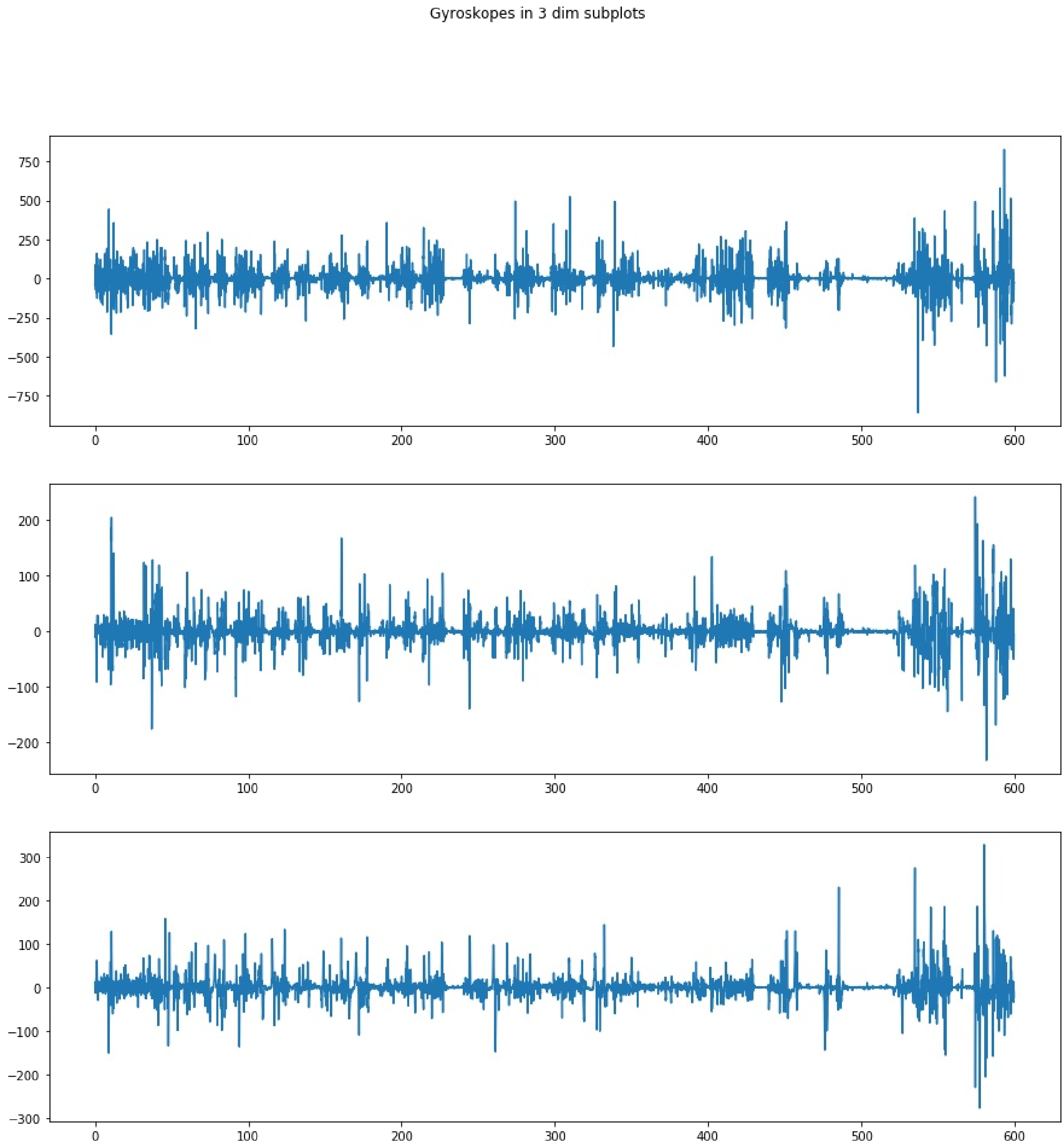


In [11]:

```
fig, axs = plt.subplots(3,figsize=(15,15))#κοινή γραφική παράσταση και για τους τρεις άξονες
fig.suptitle('Gyroskopes in 3 dim subplots')
axs[0].plot(t20, gyr0)
axs[1].plot(t20, gyr1)
axs[2].plot(t20, gyr2)
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x114e0e450>]
```



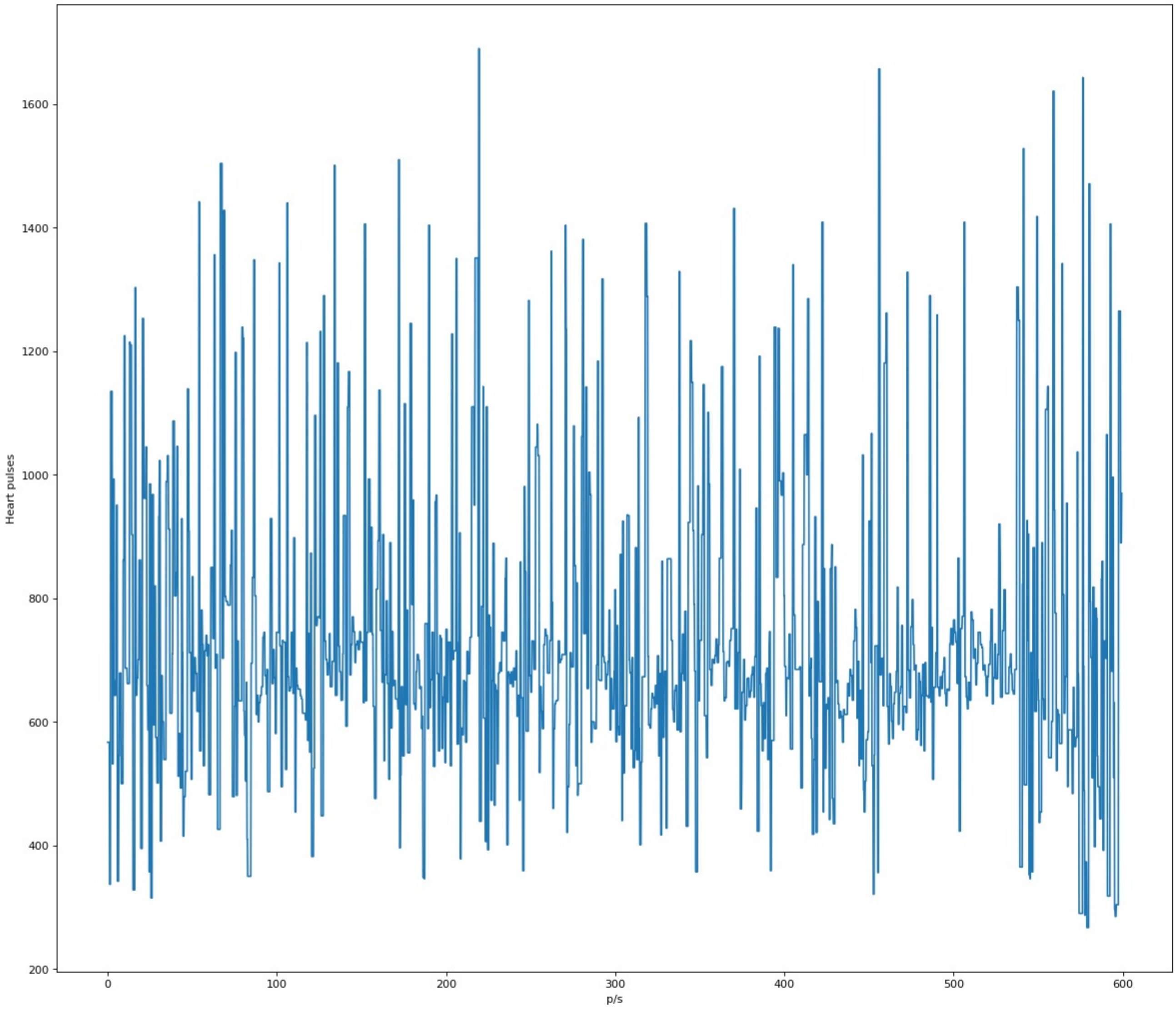
Για τους καρδιακούς παλμούς έχουμε συχνότητα 5 HZ άρα κάθε δείγμα απέχει από το επόμενο 0.2s και ο συνολικός χρόνος παραμένει ο ίδιος με πριν .

In [6]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
t5=np.arange(0.,599.6,0.2) #Γραφική παράσταση για το heart rate σήμα
plt.plot(t5,data['hrm'][:])
plt.ylabel('Heart pulses')
plt.xlabel('p/s')
```

Out[6]:

```
Text(0.5, 0, 'p/s')
```



Παρατηρούμε ότι καθώς ο χρήστης βρίσκεται σε κίνηση παρουσιάζει κάποια ένταση στην καρδιακή του συχνότητα .

In [13]:

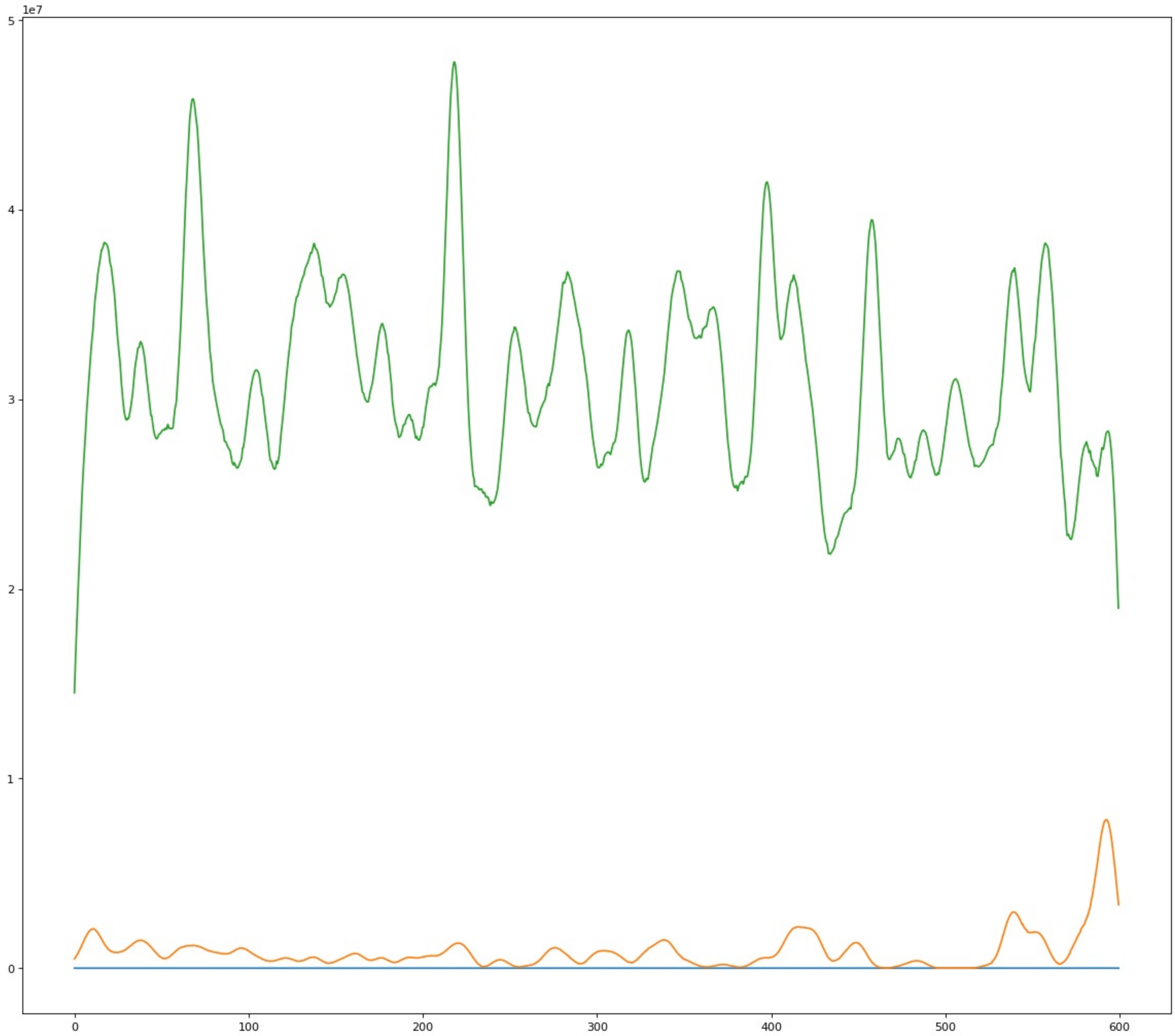
```
from scipy import signal
import scipy.signal

hamming20 = np.hamming(400) #hamming window of length 20 s for signal sampled at 20 Hz.
hamming5 = np.hamming(100) #hamming window of length 20 s for signal sampled at 5 Hz.

def square(list):
    return [i ** 2 for i in list]

def ste(x,fs): #Compute short-time energy.
    win=np.hamming(20*fs) #Declaration of a hamming window of length 20 sec
    return scipy.signal.convolve(square(np.abs(x)), win, mode="same")#Returns convolution with the hamming window of length 20s
energy1=ste(acc0,hamming20)#Short-time energy of accx signal
energy2=ste(gyr0,hamming20)#Short-time energy of accy signal
energy3=ste(data['hrm'],hamming5)#short-time energy of hrm signal
#energy_i is short time energy of every signal at x-axis
```

In [14]:

```
fig=plt.figure(figsize=(18, 16), dpi= 80, facecolor='w', edgecolor='k')
plt.plot(t20,energy1)
plt.plot(t20,energy2)
plt.plot(t5,energy3)
plt.figure()
```

Out[14]:

```
<Figure size 432x288 with 0 Axes>
```

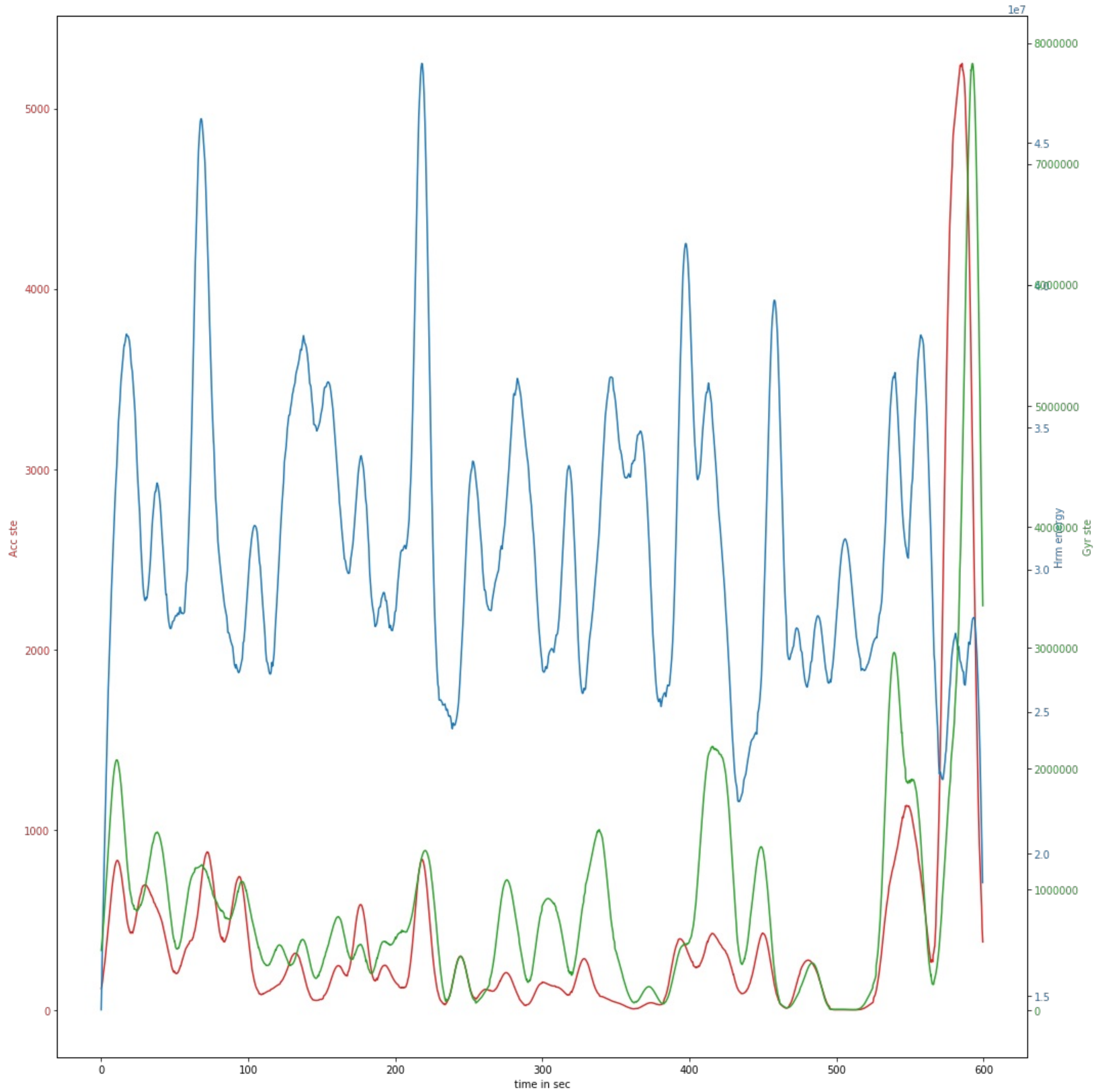



```
<Figure size 432x288 with 0 Axes>
```

Παρατηρώ ότι η ενέργεια βραχέως χρόνου για το σήμα hrm είναι πολύ μεγαλύτερη από τις άλλες δύο. Κατά την γνώμη μου αυτό οφείλεται στο γέγονος ότι ο χρήστης μπορεί παρότι βρίσκεται σε κίνηση να παρουσιάζει τόσο ένταση στις κινήσεις του ενώ η καρδιά κατά την διάρκεια κίνησης εμφανίζει πιο αυξημένη δραστηριότητα σταθερά στο χρόνο . Επίσης στα διάφορα peaks στην πορτοκαλή καμπύλη (γυροσκόπειο) εκτιμώ πως ο χρήστης είναι σε κίνηση σε σχέση με τις χρονικές στιγμές όπου είναι σχεδόν μηδενική .

In [22]:

```
fig, ax1 = plt.subplots(figsize=(15,15))
#γραφική παράσταση των 3 σημάτων accX , gyrX και hrm σε κοινούς άξονες ανάλογα βαθμονομημένους με βάση την συχνότητα
color = 'tab:red' #δειγματοληψίας κάθε σήματος
ax1.set_xlabel('time in sec')
ax1.set_ylabel('Acc ste', color=color)
ax1.plot(t20,energy1, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('Hrm energy', color=color)  # we already handled the x-label with ax1
ax2.plot(t5, energy3, color=color)
ax2.tick_params(axis='y', labelcolor=color)

ax3 = ax1.twinx()
color = 'tab:green'
ax3.set_ylabel('Gyr ste', color=color)  # we already handled the x-label with ax1
ax3.plot(t20, energy2, color=color)
ax3.tick_params(axis='y', labelcolor=color)

fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```

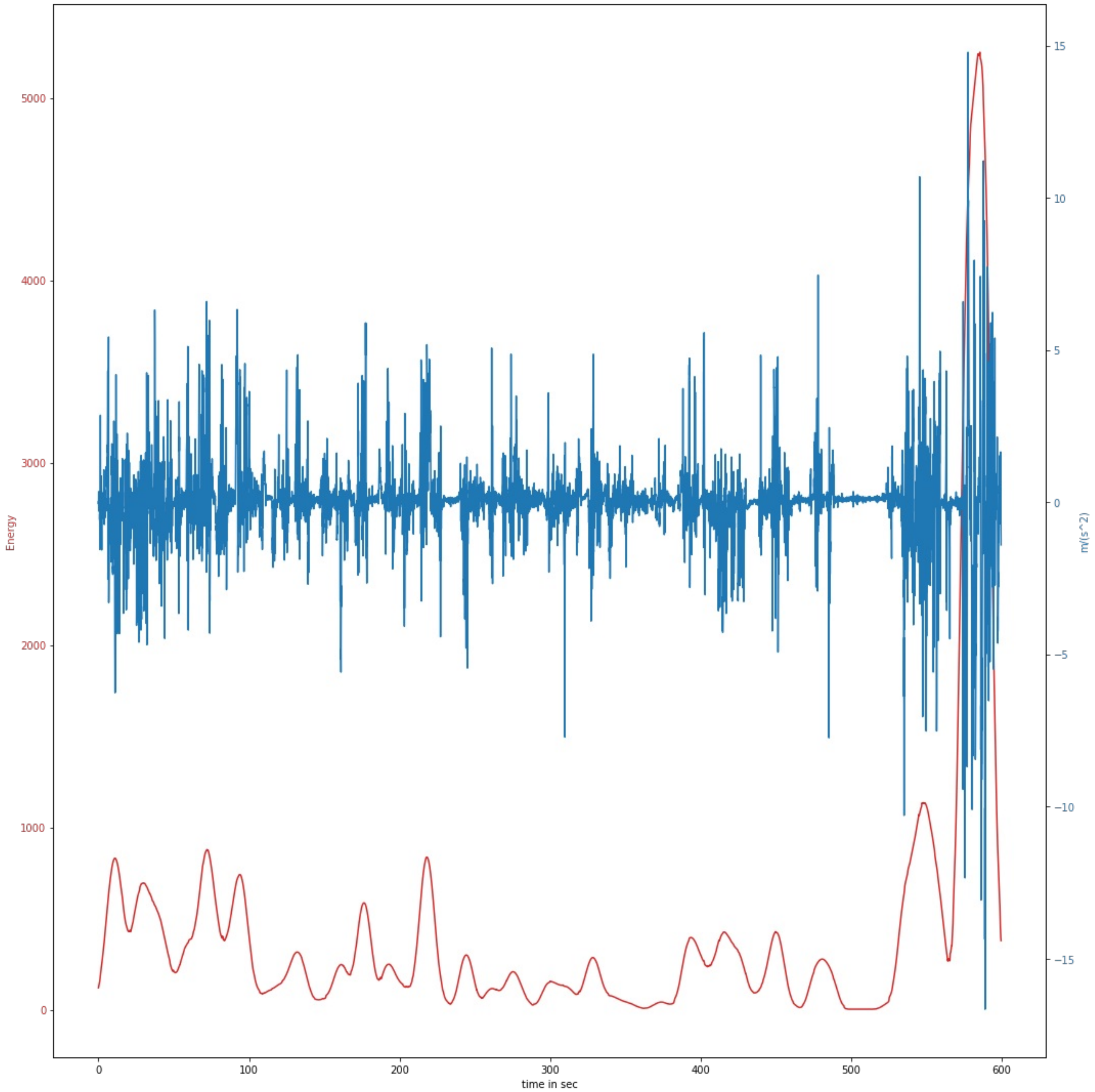


Στο παραπάνω γράφημα παρατηρώ πως στο τέλος η ενέργεια βραχέως χρόνου για το αξελερόμετρο αυξάνεται σημαντικά σε σχέση με άλλες χρονικές στιγμές άρα πιθανότητα εκείνη την χρονική στιγμή ο χρήστης ξεκίνησε να κινείται .

Στο παρακάτω γράφημα απεικονίζεται η επιτάχυνση στον άξονα Χ και η ενέργεια βραχέως χρόνου για το ίδιο σήμα βαθμονομημένα όμως σε διαφορετικούς άξονες .

In [23]:

```
fig, ax1 = plt.subplots(figsize=(15,15))

color = 'tab:red'
ax1.set_xlabel('time in sec')
ax1.set_ylabel('Energy', color=color)
ax1.plot(t20, energy1, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('m/(s^2)', color=color)  # we already handled the x-label with ax1
ax2.plot(t20, acc0, color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.show()
```



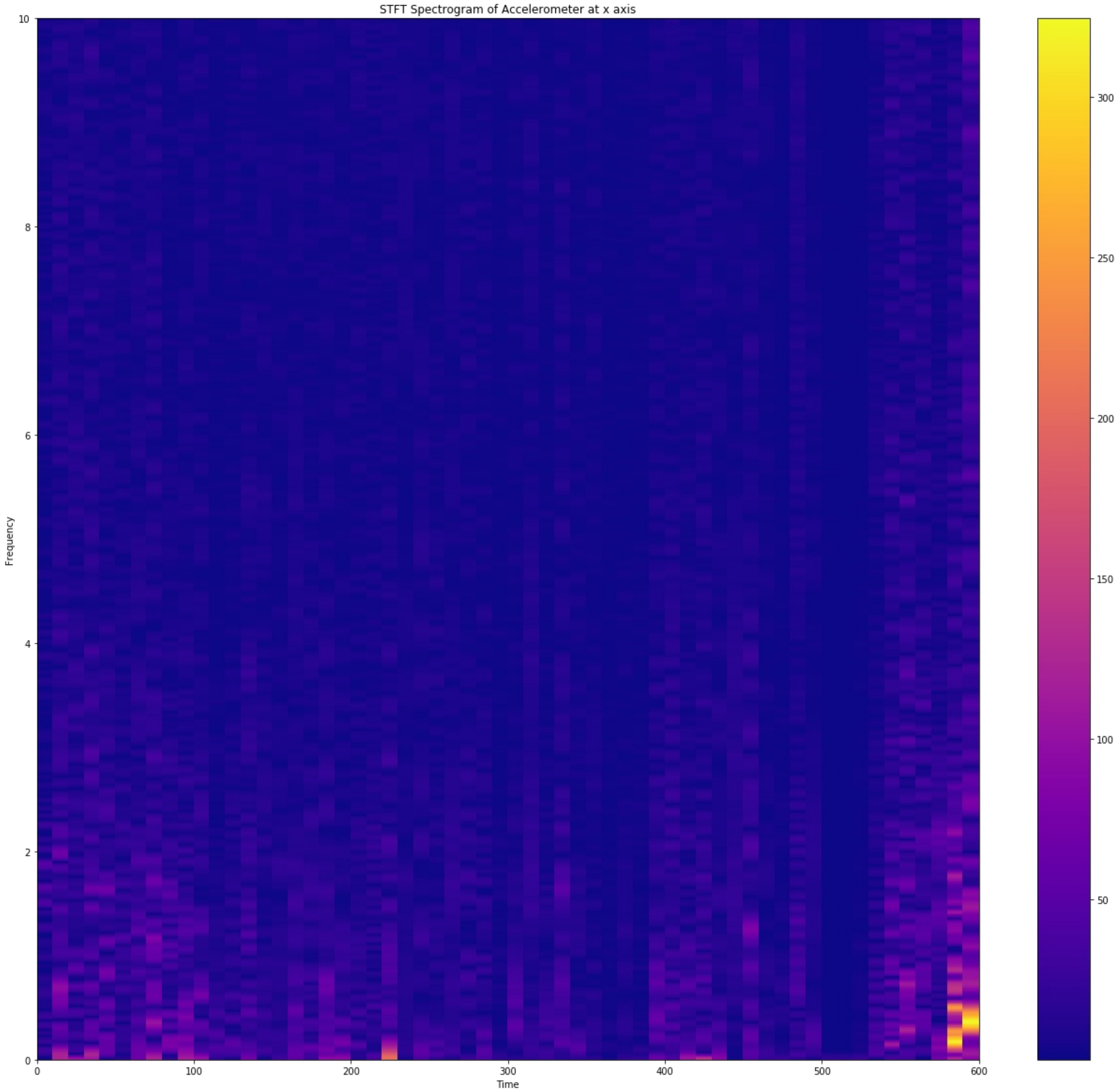
Παρατηρώ πώς το γυροσκόπειο έχει μεγάλη διαφορά σε αρκετές τάξεις μεγέθους από τα άλλα σήματα . Επίσης καθώς η ενέργεια των δύο αισθητήρων δεν κυμένεται κοντά στο μηδέν που αυτό σημαίνει πως ο χρήστης βρίσκεται σε κίνηση .

In [24]:

```
import librosa
F_acc0 = librosa.stft(np.array(acc0),hop_length=200,win_length=400)
```

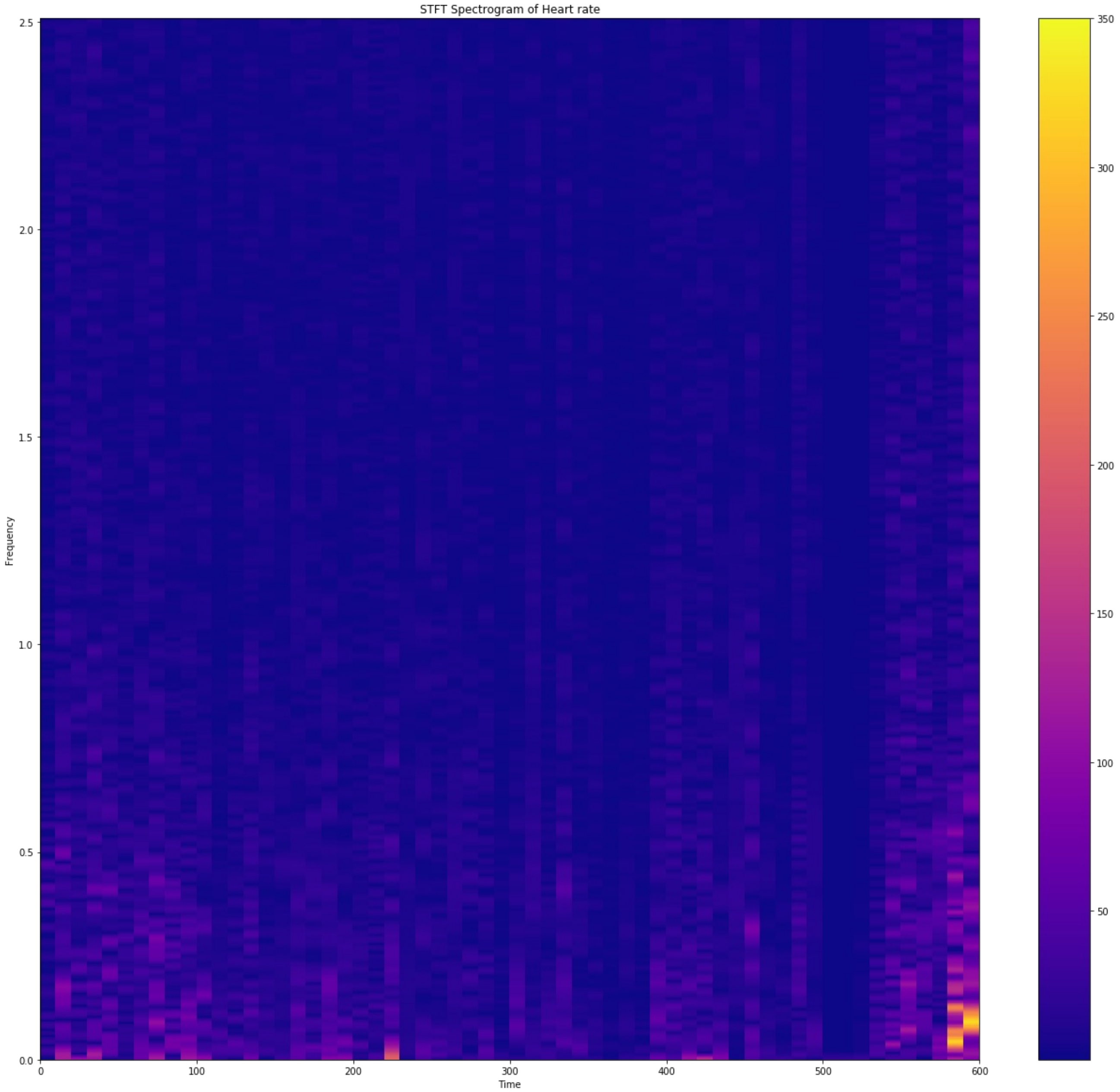
In [25]:

```
x, y = np.mgrid[0:10.01:10.01/1025, 0:600.01:10] #Το 0.1 δίπλα από το 10 και το 600 το έβαλα για να εμφανίζεται
#στην φραφική παράσταση , κανονικά το σήμα μας έπρεπε να υπολογίζεται μέχρι 10 Hz ακριβώς από τον κανόνα του
#Shannon
fig=plt.figure(figsize=(18, 16))
im=plt.pcolormesh(y,x,np.abs(F_acc0),cmap='plasma')
plt.colorbar(im)
fig.tight_layout()
plt.title('STFT Spectrogram of Accelerometer at x axis')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()
```



In [26]:

```
F_hrm = librosa.stft(np.array(data['hrm']),hop_length=50,win_length=100)
#άλλαξα κάποιες παραμέτρους για να εμφανιστούν καλύετρα τα αποτελέσματα
x, y = np.mgrid[0:2.51:2.51/1025, 0:600.01:10]
fig=plt.figure(figsize=(18, 16))
m=plt.pcolormesh(y,x,np.abs(F_acc0),cmap='plasma',vmin=10**-3,vmax=350)
plt.colorbar(m)
fig.tight_layout()
plt.title('STFT Spectrogram of Heart rate')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()
```



In [27]:

```
l=[acc0,acc1,acc2,gyr0,gyr1,gyr2,data['hrm']]
for x in l :
    print(np.mean(x))
    print(np.min(x))
    print(np.max(x))
    print(np.std(x))
```

```
0.019003001334222808
-16.662968
14.780563
1.431037324057549
-0.02170252793529018
-21.485146
13.381807
1.3863680751245282
0.03397199849899933
-16.275489999999998
19.464910999999997
1.3550099971897287
0.06398181195797184
-858.76001
827.6099849999999
65.04226522908768
0.45098399432955305
-231.770004
242.13000499999998
24.10534926705292
0.14606488817545038
-277.690002
327.390015
27.591927462649288
725.4406270847231
267.0
1690.0
236.14038302996528
```

In [ ]:

```
def features (signal): #Computation of mean , min ,max & std value of a signal
    mean=np.mean(signal)
    minimum=np.min(signal)
    maximum=np.max(signal)
    std=np.std(signal)
    return (mean,minimum,maximum,std) #returns a tuple with the characteristics
```