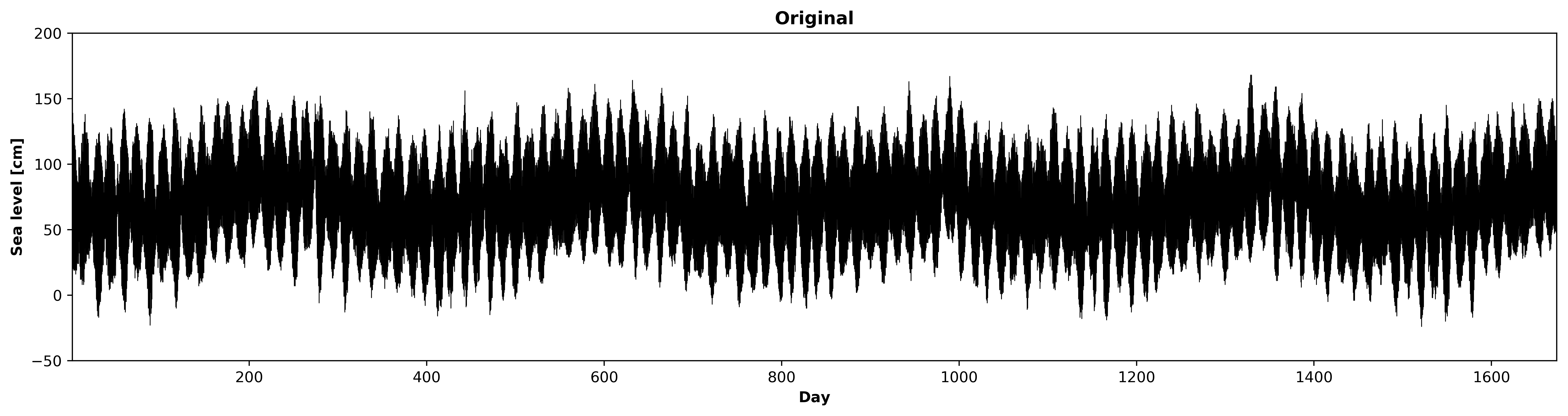
**객관적자료분석 HW#7**

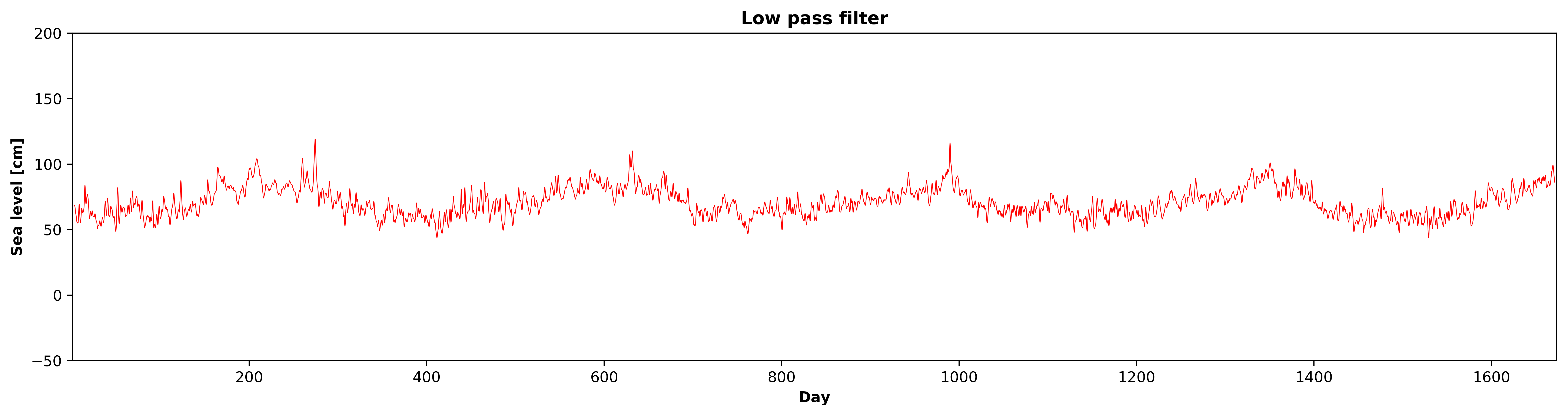
2017-29008 백관구

부산의 해수면 자료에 대해 low pass filter, high pass filter, band pass filter를 적용하고, power spectrum 분석을 통해 filter의 효과를 확인하였다. 문제에서 주어진 부산의 해수면 자료의 기존 값은 아래와 같다.



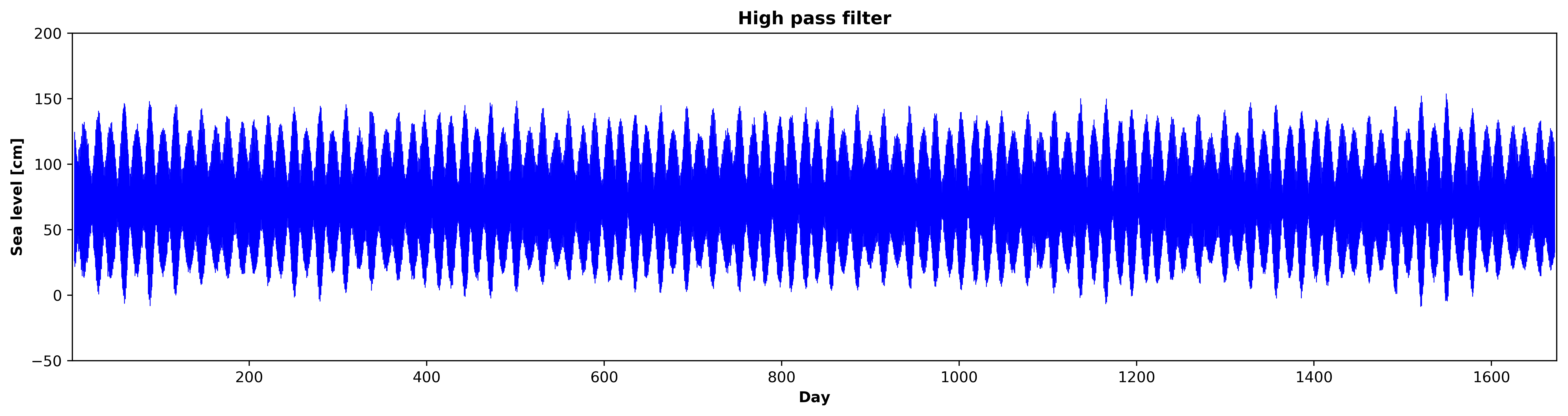
**1. Low pass filter**

Cut-off frequency가 1/36 cph인 low pass filter를 설정하기 위해서 L=0, N=54, M=2.5로 조정하였다. 이 경우 기존의 자료가 가지고 있는 36시간보다 긴 장주기 파장이 산출되었다. High pass filter와 band pass filter의 결과보다 변동폭은 작은 것으로 나타났다.

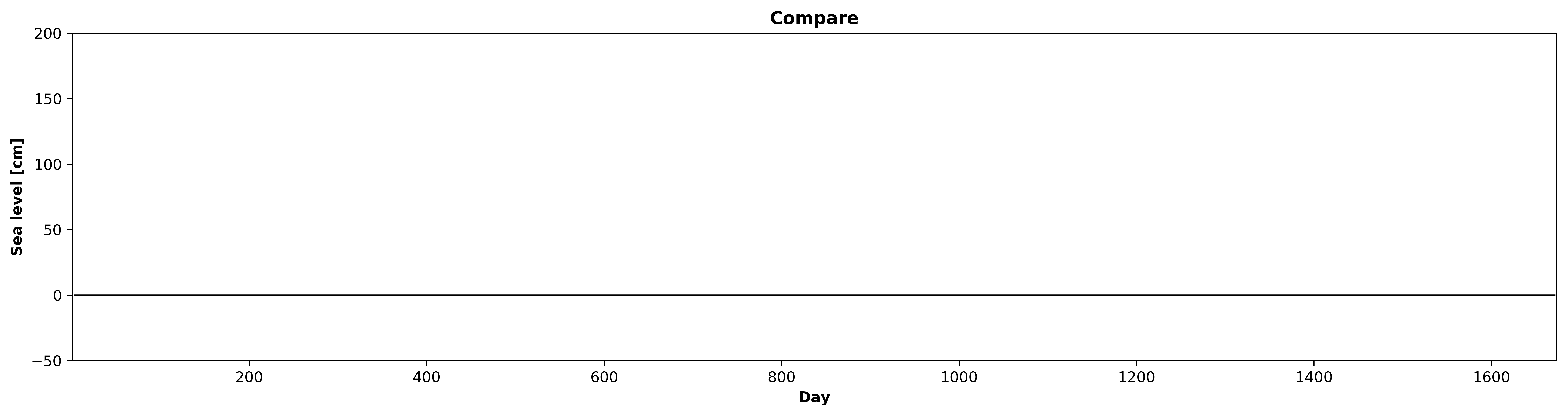


**2. High pass filter**

Cut-off frequency가 1/36 cph인 high pass filter를 설정하기 위해서 L=3.5, N=54, M=54로 조정하였다. 이 경우에는 36시간보다 짧은 단주기 파장이 산출되었다. Low pass filter 결과보다 전체적으로 진폭이 크게 나타났다.

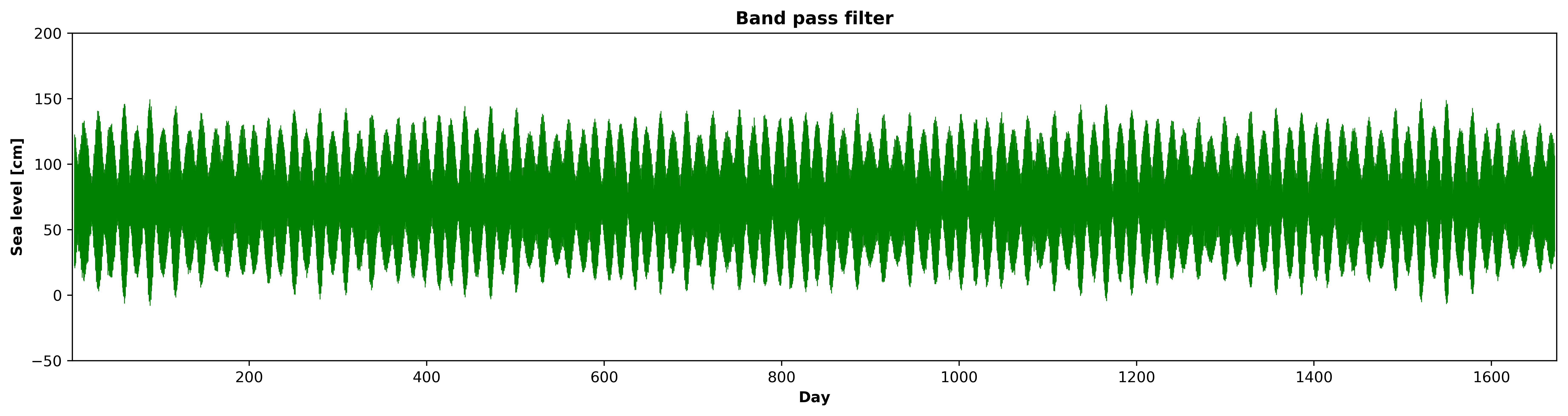


**3.** 기존의 자료에서 low pass filter의 편차값을 뺀 값과 high pass filter 값을 비교해 filter 간의 관계를 확인하였다. 그 차이를 확인했을 때 전부 0에 가까운 값을 보였다.



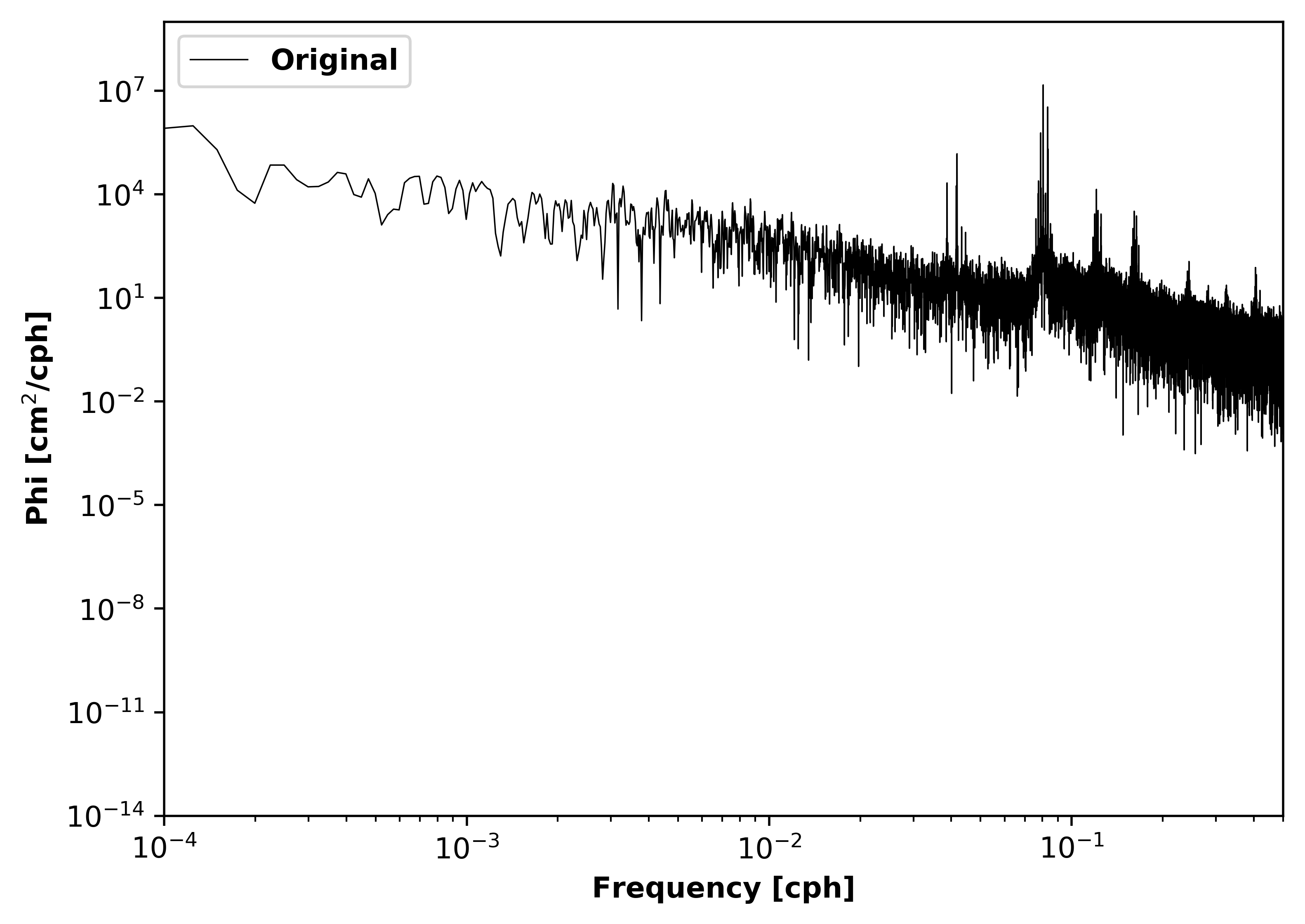
**4. Band pass filter**

1/20 cph ~ 1/10 cph의 band pass filter를 구성하기 위해 L=4.4, N=54, M=11.8로 조정하였다. 이 경우 10~20시간 주기의 파장을 분리해낼 수 있다.

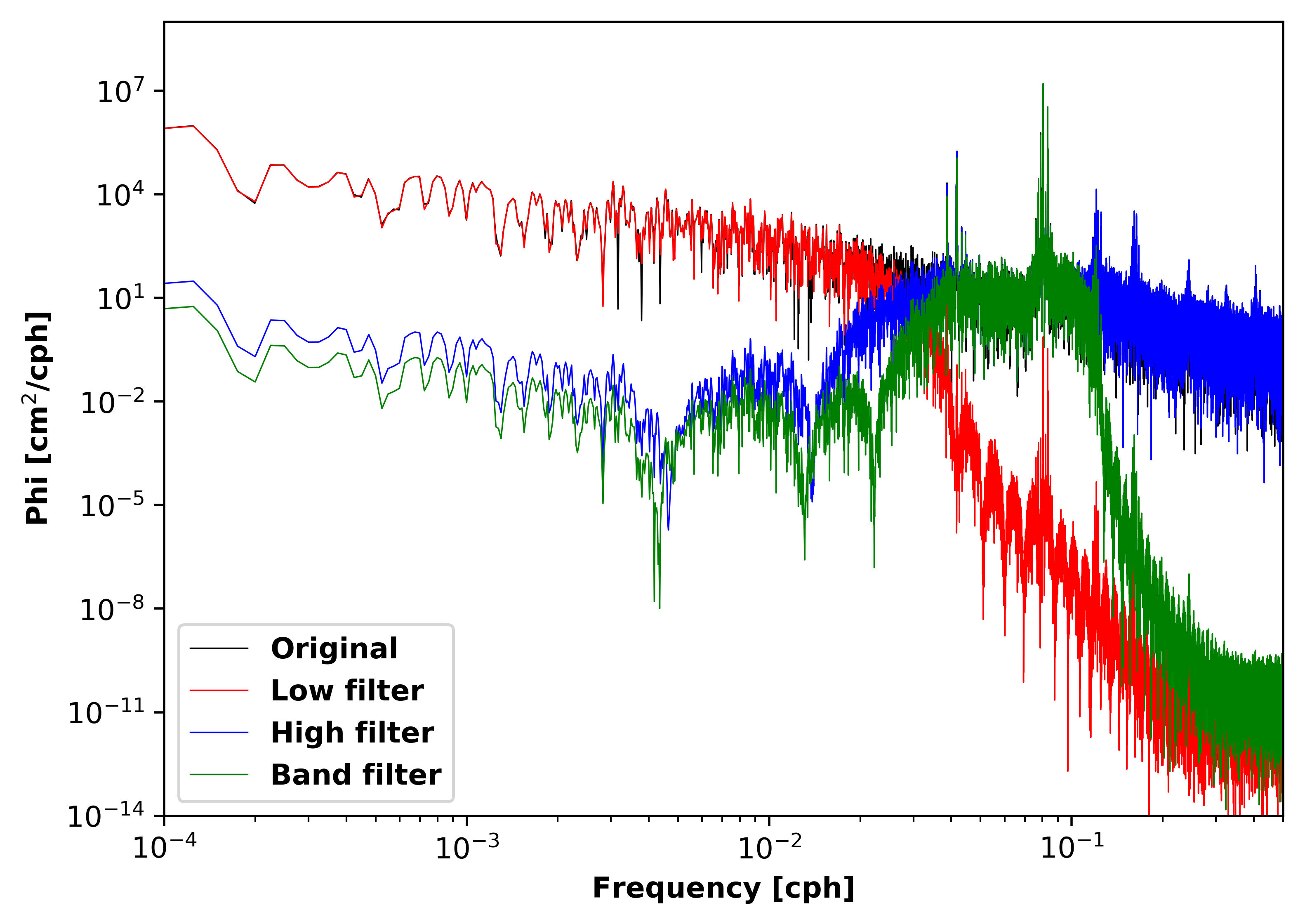


**5. Power spectrum**

위에서 구한 기존의 자료, low pass filter, high pass filter, band pass filter 자료를 power spectrum 분석에 적용하여 그 효과를 살펴보았다. Window는 parzen window를 사용하였다. 우선 기존 자료의 파워 스펙트럼 분석 결과는 아래와 같았다. 0.08 cph 부근에서 큰 에너지가 감지되었으며 이는 약 12.5 시간, 즉 반일주기에 해당한다.



각 filter를 적용해 파워 스펙트럼 분석한 결과는 아래와 같았다. 1/36 cph = 0.027 cph에서 low pass filter와 high pass filter의 파워 스펙트럼이 구분되는 것을 확인할 수 있었다. Low pass filter는 장주기의 에너지를, high pass filter는 단주기의 에너지를 잘 잡아낸 것으로 여겨진다. 1/20 cph ~ 1/10 cph의 band pass filter에 대한 파워 스펙트럼 결과는 10~20 시간 주기의 에너지를 감지하였으며, 기존의 자료와 마찬가지로 반일주기(12.5 시간)에서 가장 큰 에너지를 확인하였다.



# -\*- coding: utf-8 -\*-

"""

Created on Fri May 10 12:51:26 2019

@author: WHITE

"""

#%% IMPORT MODULES

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import periodogram

#%% RAW DATA

raw\_file = np.loadtxt("C:/Users/WHITE/Google 드라이브/SNU/2019.1/객관적자료분석/HW7/pus9802.dat")

day = raw\_file[:, 0]

raw = raw\_file[:, 1]

fig, sub = plt.subplots(figsize = (18, 4))

sub.plot(day, raw, marker = None, ls = "-", c = "k", lw = 0.5)

sub.set\_xlim(min(day), max(day))

sub.set\_ylim(-50, 200)

sub.set\_xlabel("Day", fontdict = {"weight": "bold"})

sub.set\_ylabel("Sea level [cm]", fontdict = {"weight": "bold"})

sub.set\_title("Original", fontdict = {"weight": "bold"})

plt.savefig("./original.png", dpi = 500, bbox\_inches = "tight")

#%% LOW PASS FILTER

low\_file = np.loadtxt("C:/Users/WHITE/Google 드라이브/SNU/2019.1/객관적자료분석/HW7/pus\_low.dat")

low\_day = low\_file[:, 0]

low = low\_file[:, 1]

fig, sub = plt.subplots(figsize = (18, 4))

sub.plot(low\_day, low, marker = None, ls = "-", c = "r", lw = 0.5)

sub.set\_xlim(min(day), max(day))

sub.set\_ylim(-50, 200)

sub.set\_xlabel("Day", fontdict = {"weight": "bold"})

sub.set\_ylabel("Sea level [cm]", fontdict = {"weight": "bold"})

sub.set\_title("Low pass filter", fontdict = {"weight": "bold"})

plt.savefig("./low.png", dpi = 500, bbox\_inches = "tight")

#%% HIGH PASS FILTER

high\_file = np.loadtxt("C:/Users/WHITE/Google 드라이브/SNU/2019.1/객관적자료분석/HW7/pus\_high.dat")

high\_day = high\_file[:, 0]

high = high\_file[:, 1]

fig, sub = plt.subplots(figsize = (18, 4))

sub.plot(high\_day, high, marker = None, ls = "-", c = "b", lw = 0.5)

sub.set\_xlim(min(day), max(day))

sub.set\_ylim(-50, 200)

sub.set\_xlabel("Day", fontdict = {"weight": "bold"})

sub.set\_ylabel("Sea level [cm]", fontdict = {"weight": "bold"})

sub.set\_title("High pass filter", fontdict = {"weight": "bold"})

plt.savefig("./high.png", dpi = 500, bbox\_inches = "tight")

#%% COMPARE

com1 = raw[54 : -54] - (low - np.mean(low))

com2 = high

com = com1 - com2

fig, sub = plt.subplots(figsize = (18, 4))

sub.plot(high\_day, com, marker = None, ls = "-", c = "k", lw = 1)

sub.set\_xlim(min(day), max(day))

sub.set\_ylim(-50, 200)

sub.set\_xlabel("Day", fontdict = {"weight": "bold"})

sub.set\_ylabel("Sea level [cm]", fontdict = {"weight": "bold"})

sub.set\_title("Compare", fontdict = {"weight": "bold"})

plt.savefig("./compare.png", dpi = 500, bbox\_inches = "tight")

#%% BAND PASS FILTER

band\_file = np.loadtxt("C:/Users/WHITE/Google 드라이브/SNU/2019.1/객관적자료분석/HW7/pus\_band.dat")

band\_day = band\_file[:, 0]

band = band\_file[:, 1]

fig, sub = plt.subplots(figsize = (18, 4))

sub.plot(band\_day, band, marker = None, ls = "-", c = "g", lw = 0.5)

sub.set\_xlim(min(day), max(day))

sub.set\_ylim(-50, 200)

sub.set\_xlabel("Day", fontdict = {"weight": "bold"})

sub.set\_ylabel("Sea level [cm]", fontdict = {"weight": "bold"})

sub.set\_title("Band pass filter", fontdict = {"weight": "bold"})

plt.savefig("./band.png", dpi = 500, bbox\_inches = "tight")

#%% POWER SPECTRUM

fr\_raw, ps\_raw = periodogram(raw, fs = 1, window = "parzen", scaling = "density")

fr\_low, ps\_low = periodogram(low, fs = 1, window = "parzen", scaling = "density")

fr\_high, ps\_high = periodogram(high, fs = 1, window = "parzen", scaling = "density")

fr\_band, ps\_band = periodogram(band, fs = 1, window = "parzen", scaling = "density")

fig, sub = plt.subplots(figsize = (7, 5))

sub.plot(fr\_raw[1 :], ps\_raw[1 :], c = "k", ls = "-", lw = 0.5, marker = None, label = "Original")

sub.plot(fr\_low[1 :], ps\_low[1 :], c = "r", ls = "-", lw= 0.5, marker = None, label = "Low filter")

sub.plot(fr\_high[1 :], ps\_high[1 :], c = "b", ls = "-", lw= 0.5, marker = None, label = "High filter")

sub.plot(fr\_band[1 :], ps\_band[1 :], c = "g", ls = "-", lw = 0.5, marker = None, label = "Band filter")

sub.set\_xscale("log")

sub.set\_yscale("log")

sub.set\_xlim(left = 0.0001, right = 0.5)

sub.set\_ylim(10\*\*(-14), 10\*\*9)

sub.set\_xlabel("Frequency [cph]", fontdict = {"weight": "bold"})

sub.set\_ylabel(r"Phi [cm$^2$/cph]", fontdict = {"weight": "bold"})

sub.legend(prop = {"weight": "bold"})

plt.savefig("./ps.png", dpi = 500, bbox\_inches = "tight")