

# Econ 413R: Computational Economics

## Spring Term 2013

### Time Series Data Filtering

## 1 Introduction

**Definition:** A time-series  $u_t$  is said to be stationary if the expected value, variance, and autocovariance of the series are independent of the time period  $t$ .

Many economic models are stationary, but many economic variables have readily observable growth trends. To reconcile this difference, economics often compare their stationary models with “detrended” data, or data where the growth component has been removed.

There are many ways to detrend a time series and the method used to remove the trend can seriously change the resulting series. Hence it is important to understand the theory behind time series decomposition.

A few common ways to detrend data include differencing (especially for logged data), line/polynomial fitting, HP filters, band-pass filters, and others.

We go over four common data-filtering techniques. OLS and HP are given first naively, then in depth, followed by Band-Pass filters.

## 2 OLS Filter

The goal of any filter is to decompose the time series into several series with common frequencies. Let  $y_t$  be our data at time-period  $t$ . We want to decompose the data

into growth component,  $\tau_t$ , and the cyclical component,  $c_t$

$$y_t = \tau_t + c_t \quad \text{for } t = 1, \dots, T \quad (2.1)$$

The simplest of filters involves fitting set  $\tau$  to be the “polynomial of best fit” through the time series. The “polynomial of best fit” minimizes the following expression:

$$\min_{\beta} \|y - \beta x\|$$

For a polynomial of degree 1 (a straight line), the solution is given by the well known “normal equations” given by  $\hat{\beta} = (X^T X)^{-1} X^T y$ , where

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_n \end{pmatrix}$$

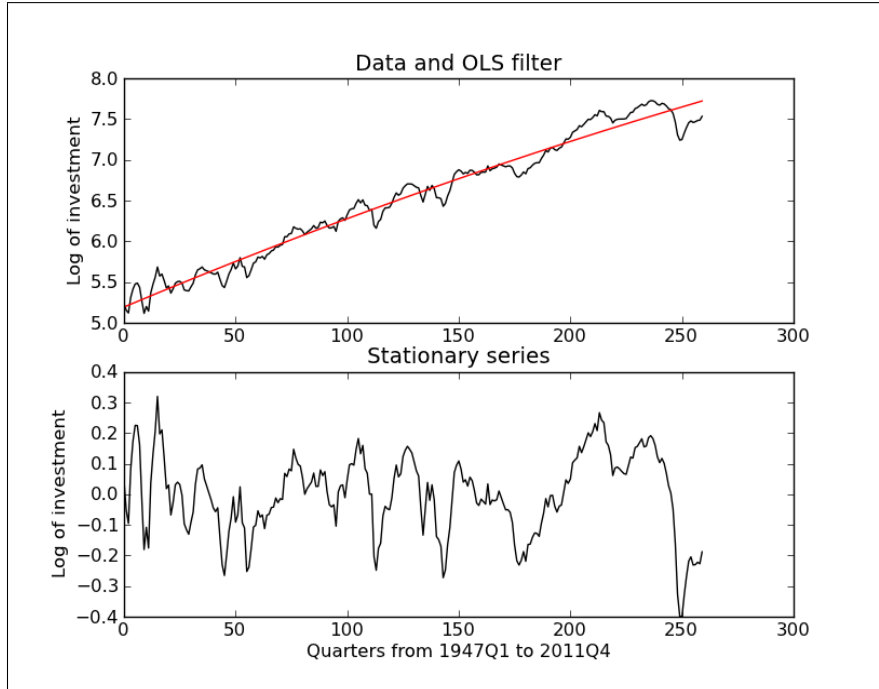
We can generalize this for a polynomial of degree  $k$  using the  $k^{th}$  degree Vandermonde matrix and the same set of normal equations.

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{pmatrix}$$

Using a quadratic polynomial filter on logged investment data gives the following trend series (depicted in red) and cyclical series (depicted in the second subpanel)

As desired, the resulting series appears to have a constant mean and variance throughout the entire series.

**Figure 1: OLS Filter (quadratic)**



### 3 HP Filter

The Hodrick-Prescott (HP) filter was popularized by [Hodrick and Prescott \(1997\)](#). It generates a line of best fit very differently from polynomial interpolation or OLS. An HP is a kind of moving average (MA( $k$ )) filter, meaning that the point in the filtered series at time  $t$  is a weighted average of the raw data over  $k$  points before/after  $t$ . As given in [2.1](#), it aims to decompose the raw data into a smooth “trend series” (sometimes referred to as the “growth series”) and a stationary “cyclical series” such that  $y_t = \tau_t + c_t$

To find the  $\tau$  series that best fits the data, we minimize the following

$$\min_{\{\tau_t\}} \left\{ \sum_{t=1}^T (y_t - \tau_t)^2 + \lambda \sum_{t=1}^T [(\tau_t - \tau_{t-1}) - (\tau_{t-1} - \tau_{t-2})]^2 \right\} \quad (3.1)$$

with  $c_t = y_t - \tau_t$  giving the deviation from the trend component (a mean zero process). The parameter  $\lambda$  penalizes changes in the trend component. A higher  $\lambda$  will result in a smoother trend component. In fact, as  $\lambda$  approaches infinity, the optimal  $\tau_t - \tau_{t-1}$

tends towards a constant  $\beta$ , so  $\tau_t = \tau_0 + \beta t$ , and the filtered series is simply the least squares solution. The proof of this is left as an exercise.

To compute the filter, we can write the first-order-conditions for 3.1, in matrix form as shown:

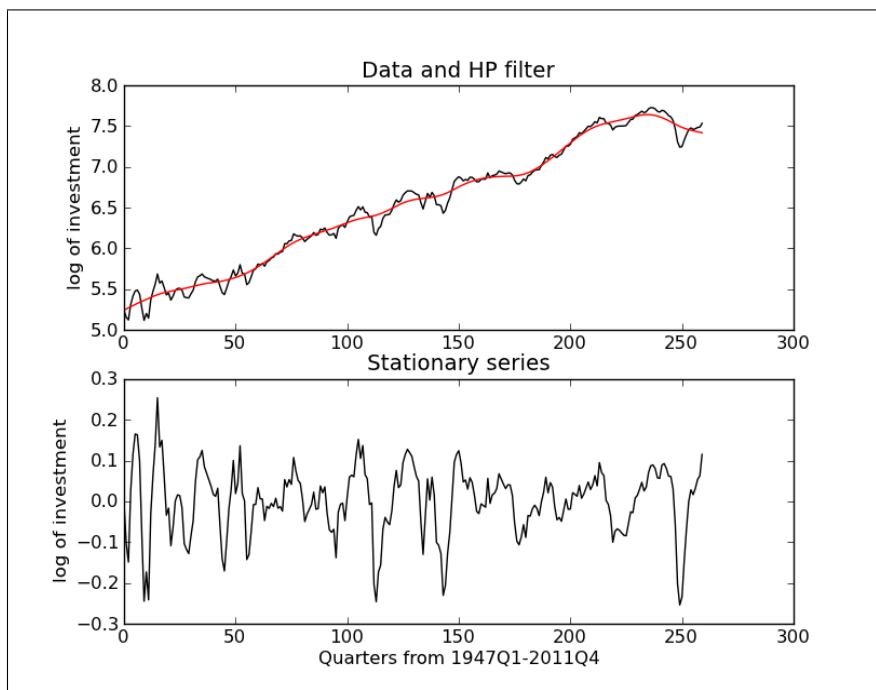
$$\Lambda T = Y$$

$$T = \Lambda^{-1} Y$$

where the  $(n, m)^{th}$  entry of  $\Lambda$  is the  $\lambda$  coefficient for the  $i^{th}$  first-order-condition found by differentiating with respect to  $\tau_i$ . The matrix  $\Lambda$  is symmetric in all cases and sparse for large values of T, which can allow us to compute  $\Lambda^{-1}$  very quickly.

An HP filter with  $\lambda = 1600$  on quarterly data will generate a filter that moves along the data more flexibly than the quadratic polynomial, as shown in the figure below.

**Figure 2: HP Filter ( $\lambda = 1600$ )**



How does the stationary series in figure 2 (HP filter) compare to the stationary series from the OLS filter in figure 1? Does the HP generate a more cyclical series

that is more or less stationary than OLS?

## 4 Spectral Analysis

Any time-series can be fitted with an appropriately high-order polynomial. Suppose we have a time series  $X = \{x_1, x_2, \dots, x_T\}$ . With  $T$  observations we can specify a polynomial of order  $T - 1$  that will pass through each point.

Similarly any time series can be fitted with a weighted sum of an appropriate number of sinusoidal wave functions of various wavelengths. Again, with  $T$  observations we can sum  $T - 1$  wave functions and the resulting time-path will pass through each point. The mapping from observations at various points in time to wave functions of various frequencies is called a Fourier transformation.

When removing trends from data it is useful to think about the effects the filter has on the various constituent sine waves. In order to do this we need to first go through the basics of spectral analysis. We will only be scratching the surface here. For more in depth discussions see...

### 4.1 Discrete Fourier Transform

Consider a sequence of (possibly complex) numbers,  $X = \{x_1, x_2, \dots, x_T\}$ . The discrete Fourier transform (DFT) of this series is defined as:

$$\hat{x}_k = \sum_{t=1}^T x_t e^{-i2\pi kt/T} \quad (4.1)$$

Each  $\hat{x}_k$  is a complex number that contains information on the amplitude and phase of a sine wave. The sum of these sine waves exactly recreates the time-series  $X$ . The frequency for  $\hat{x}_k$  is  $k/T$  cycles per sample. The amplitude of the sine wave is given by (4.2) and the phase by (4.3).

$$\frac{|\hat{x}_k|}{T} = \frac{\sqrt{Re(\hat{x}_k)^2 + Im(\hat{x}_k)^2}}{N} \quad (4.2)$$

$$\arg(\hat{x}_k) = \text{atan2}\{\text{Im}(\hat{x}_k), \text{Re}(\hat{x}_k)\} \quad (4.3)$$

The inverse discrete Fourier transform (IDFT) is:

$$x_t = \frac{1}{T} \sum_{k=1}^T y_k e^{i2\pi kt/T} \quad (4.4)$$

The DFT is often written as  $\mathcal{F}$  so that the series  $\hat{X} = \{\hat{x}_1, \hat{X}x_2, \dots, \hat{x}_T\}$  can be expressed as the DFT of  $X$  by writing  $\hat{X} = \mathcal{F}(X)$ . The IDFT, not surprisingly, is written  $X = \mathcal{F}^{-1}(\hat{X})$

$X$  is a representation of the series in the “time domain” and  $Y$  is a representation of the same series in the “frequency domain”.

## 4.2 Spectral Density

Consider an discrete infinite time series denoted  $X$ . We will define the power of  $X$  as in (4.5).

$$P \equiv \lim_{t \rightarrow \infty} \frac{1}{2T} \sum_{-T}^T x_t^2 \quad (4.5)$$

Consider now a subsample of the series running from 1 to  $T$  and denote this as  $X^T(t)$

If we define  $\omega$  as the “angular frequency” measured in radians, so that  $\omega = 2\pi k$ , then the DFT of  $X^T(t)$  can be denoted  $\hat{X}^T(\omega) = \mathcal{F}\{X^T(t)\}$ . The power spectral density (PSD), spectral density function (SDF) or simply “spectral density” of  $X$  is defined in (4.6).

$$S_x(\omega) \equiv \lim_{T \rightarrow \infty} E\{|\hat{X}^T(\omega)|^2\} \quad (4.6)$$

Since  $\omega$  is the angular frequency all the information about the spectral density is contained on the interval  $[0, 2\pi]$ .

Some useful properties of the SDF are: First,  $S_x(-\omega) = S_x(\omega)$ , the SDF is symmetric about  $\omega = 0$ . Second, the SDF describes the distribution of the variance of  $X^T$  over various frequencies.  $\text{Var}(X) = 2 \int_0^{1/2} S_x(\omega) d\omega$ .

The cross-spectral density between two time series  $X$  and  $Y$  is defined in (4.7)

$$S_{xy}(\omega) = \lim_{T \rightarrow \infty} E \left\{ |\hat{X}_T(\omega) \hat{Y}_T(\omega)| \right\}. \quad (4.7)$$

The cross-spectral density describes the distribution of the covariance of  $X^T$  and  $Y^T$  over various frequencies.  $\text{Cov}(X, Y) = 2 \int_0^{1/2} S_{xy}(\omega) d\omega$ .

### 4.3 Periodogram

In practice, we cannot get an infinite series of observations and we need an approximation or estimate of based on finite data. The classic estimate is the periodogram. The periodogram is defined by (4.8).

$$\begin{aligned} a(k) &= \frac{2}{T} \sum_{t=1}^T x_t \cos(kt) dt \\ b(k) &= \frac{2}{T} \sum_{t=1}^T x_t \sin(kt) dt \\ P_x(k) &= \sqrt{a(k)^2 + b(k)^2} \end{aligned} \quad (4.8)$$

The periodogram is normally plotted with  $\omega$  on the horizontal axis and  $P_x$  on the vertical axis.

There are a host of additional issues involved with properly estimating a periodogram. See ??? for an in depth discussion.

There are prepackaged functions in both Python and MATLAB that will generate the periodogram.

In Python there are several options. One is the `spectrum` package. The function to use is `speriodogram` . The documentation for this function is as follows:

```
speriodogram(x, NFFT=None, detrend=True, sampling=1.0,
cale_by_freq=True, window='hamming', axis=0)
Parameters:
```

```

x    an array or matrix of data samples.
NFFT length of the data before FFT is computed (zero padding)
detrend (bool)  detrend the data before computing the FFT
sampling (float)  sampling frequency of the input data.
scale_by_freq
window (str)

Returns:
2-sided PSD if complex data, 1-sided if real.
if a matrix is provided (using numpy.matrix), then a periodogram is
computed for each row.  The returned matrix has the same shape as
the input matrix.  The mean of the input data is also removed from
the data before computing the psd.

```

In MATLAB the function is `periodogram` from the signal processing toolbox. The documentation says the following:

```

    pxx = periodogram(x) returns the periodogram power spectral density (PSD)
estimate of the input signal, x, using a rectangular window.  If x is real-valued,
pxx is a one-sided PSD estimate.  If x is complex-valued, pxx is a two-sided PSD
estimate.  The number of points, nfft, in the discrete Fourier transform (DFT)
is the maximum of 256 or the next power of two greater than the signal length.

```

## 4.4 Moving Average Filters

Suppose we create a filtered series,  $y_t$ , by taking a moving average of the raw data series  $x_t$ .

$$y_t = \sum_{j=-\infty}^{\infty} b_j x_{t-j} = b(L)x_t \quad (4.9)$$



Attempting to derive the autocovariance,  $\gamma_k = E(y_t y_{t-k})$  in the time-domain results in the following

$$\sum_{j,l} b_j b_l \gamma_{k+l-j}(x),$$

which is a cumbersome convolution. By doing calculations in the spectral domain, we can use the Fourier transform of the  $b_j$  coefficients,  $\sum_j e^{-i\omega j} b_j = b(e^{i\omega})$  to derive the formula for the spectral density of  $y_t$  given the spectral density of  $x_t$ :

$$S_y(\omega) = b(e^{-i\omega})b(e^{i\omega})S_x(\omega) = |b(e^{-i\omega})|^2 S_x(\omega) \quad (4.10)$$

Note that this has the form of  $\text{Var}(y) = \text{Var}(bx) = b^2 \text{Var}(x)$ . The multivariate equivalent is

$$S_y(\omega) = B(e^{-i\omega})S_x(\omega)B(e^{i\omega})' \quad (4.11)$$

We can use this filtering formula to derive the spectral density of any ARMA series by using a white noise series as our raw data

$$x_t = \theta(L)\epsilon_t = \sum_{j=0}^{\infty} \theta_j \epsilon_{t-j}$$

Using the filtering formula, we get a formula which we can now use to find the spectral density of any ARMA process:

$$S_x(\omega) = \theta(e^{-i\omega})\theta(e^{i\omega})\sigma_\epsilon^2 \quad (4.12)$$

## 4.5 Some Examples

### 4.5.1 Simple Moving Average Filters

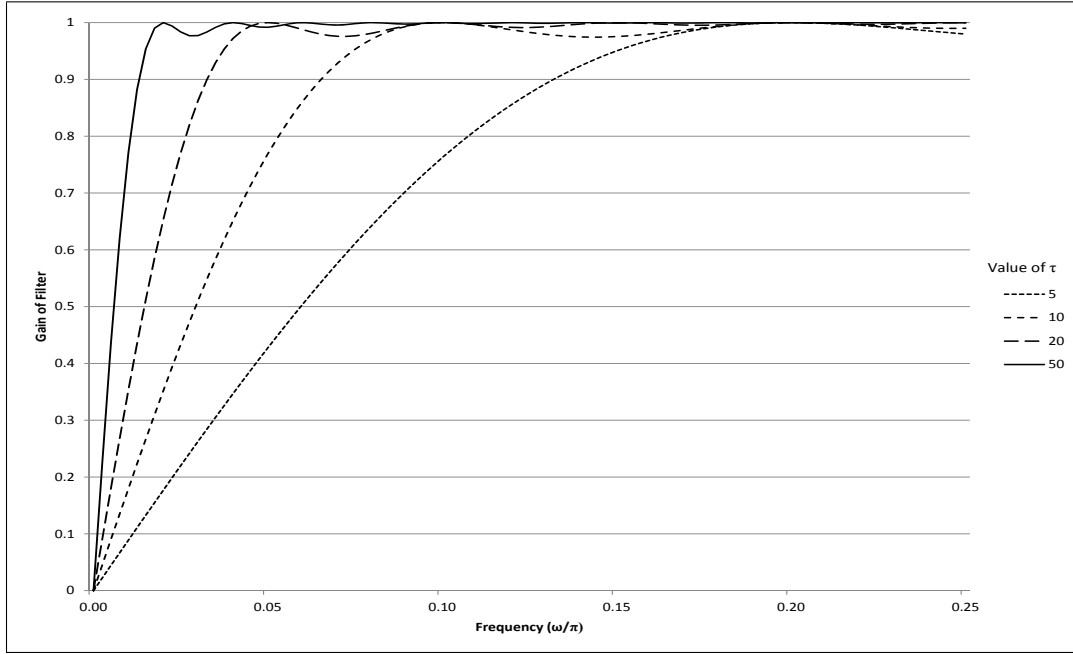
As an example, apply this to an MA(1) process,  $x_t = (1 + \theta L)\epsilon_t$ . We get

$$S_x(\omega) = (1 + \theta e^{-i\omega})(1 + \theta e^{i\omega})\sigma_\epsilon^2 = (1 + \theta(e^{i\omega} + e^{-i\omega}) + \theta^2)\sigma_\epsilon^2 = (1 + 2\theta \cos(\omega) + \theta^2)\sigma_\epsilon^2$$

Pedersen (2001) shows that the power transfer function for a moving average filter is  $|H_{MA}(\omega)|^2 = \left(\frac{\sin(\tau\omega/2)}{\tau \sin(\omega/2)}\right)^2$ , where  $\tau$  is the number of periods over which the moving-average is taken. The gain of any filter is related to the transfer function by the following formula.

$$G = \sqrt{1 - |H(\omega)|^2}. \quad (4.13)$$

**Figure 3: Gain of the Moving-Average Filter**



#### 4.5.2 Hodrick-Prescott Filter

We can also use it on an HP filter. Start by rewriting 3.1 using the lag operator  $L$  to put all  $\tau_i$  variables into the same period  $t$ . Differentiate with respect to  $\tau_t$  to get the following:

$$C(L) \equiv \frac{\lambda(1-L)^2(1-L^{-1})^2}{\lambda(1-L)^2(1-L^{-1})^2 + 1} = \frac{\lambda L^{-2}(1-L)^4}{\lambda L^{-2}(1-L)^4 + 1} \quad (4.14)$$

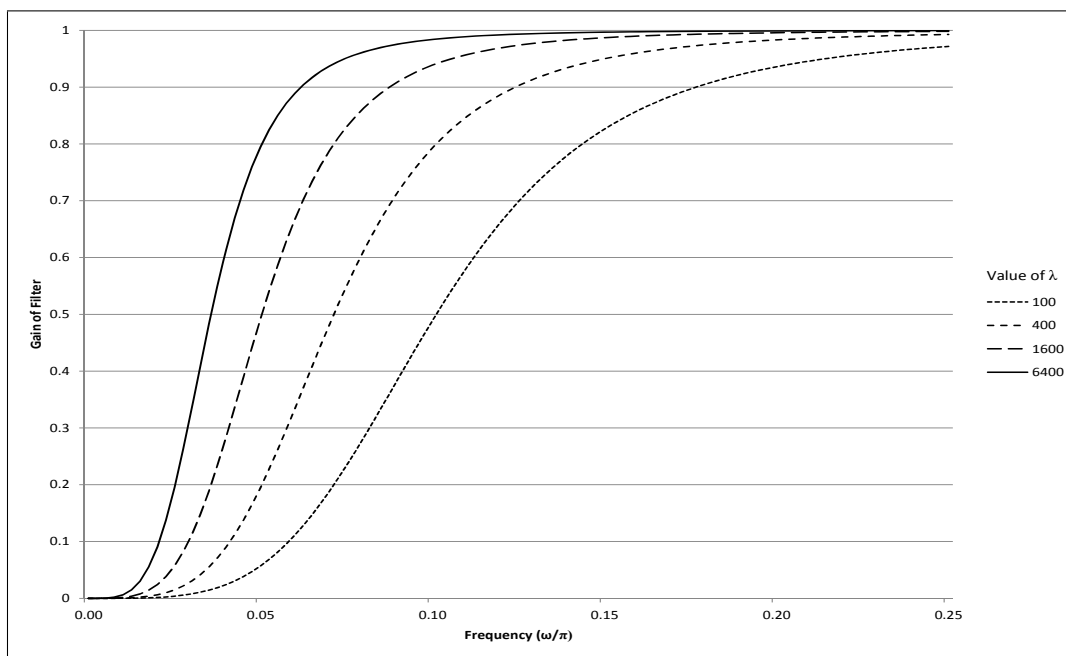
The filtering formula gives us the following:

$$|H_{HP}(\omega)|^2 = \left| \frac{\lambda(1 - e^{-i\omega})^2(1 - e^{i\omega})^2}{\lambda(1 - e^{-i\omega})^2(1 - e^{i\omega})^2 + 1} \right|^2 = \left| \frac{4\lambda[1 - \cos(\omega)]^2}{4\lambda[1 - \cos(\omega)]^2 + 1} \right|^2 \quad (4.15)$$

Plotting the gain over the domain of  $\omega \in [0, \pi)$  gives us an idea of what frequencies

are filtered out by the HP series. Figure 4 gives plots for several values of  $\lambda$ .

**Figure 4: Gain of the HP Filter**



Hodrick and Prescott (1997) argue that the appropriate value of  $\lambda$  when filtering quarterly data for business cycle frequencies is  $\lambda = 1600$ .

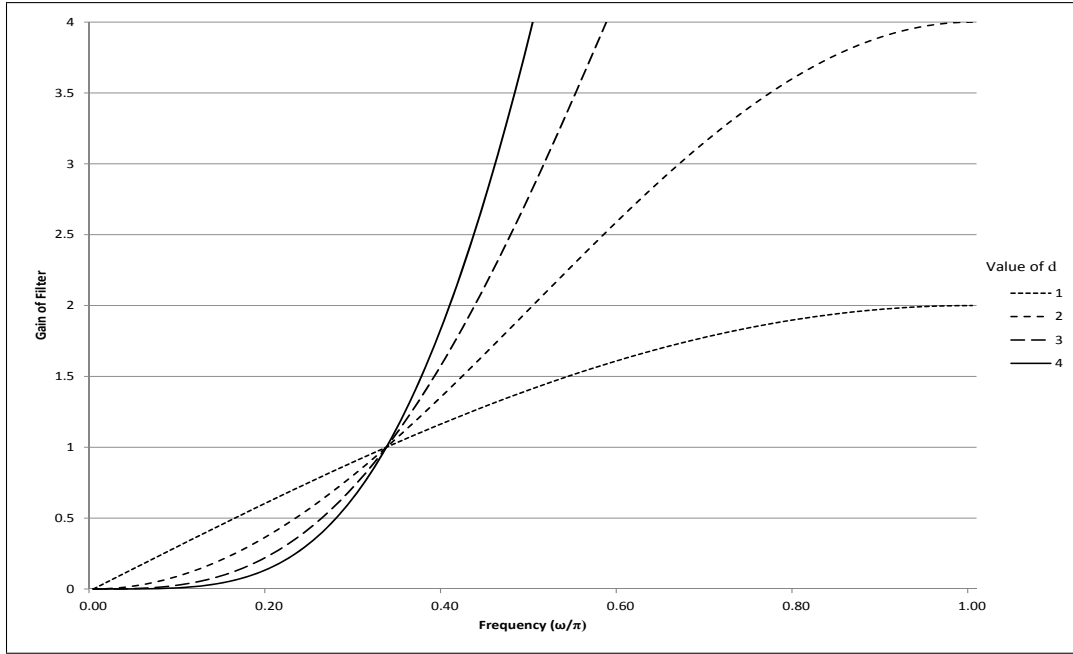
## 5 Differencing Filters

Other simple filter have been propoed and used. It is helpful to examine their effect on the spectral density of a time series. Pedersen (2001) shows the power transfer functions for some of these filters. One of these that is commonly used is the First-Difference filter. The power transfer function for a first-difference filter is  $|H_{FD}(\omega)|^2 = 2[1 - \cos(\omega)]$ . From this we can calculate the gain using (4.13).

Differencing the series  $d$  times is a  $d$ -difference filter and it's power transfer function is given by (5.1)

$$|H_{DD}(\omega)|^2 = 2^d [1 - \cos(\omega)]^d \quad (5.1)$$

Figure 5: Gain of the  $d$ -Difference Filter

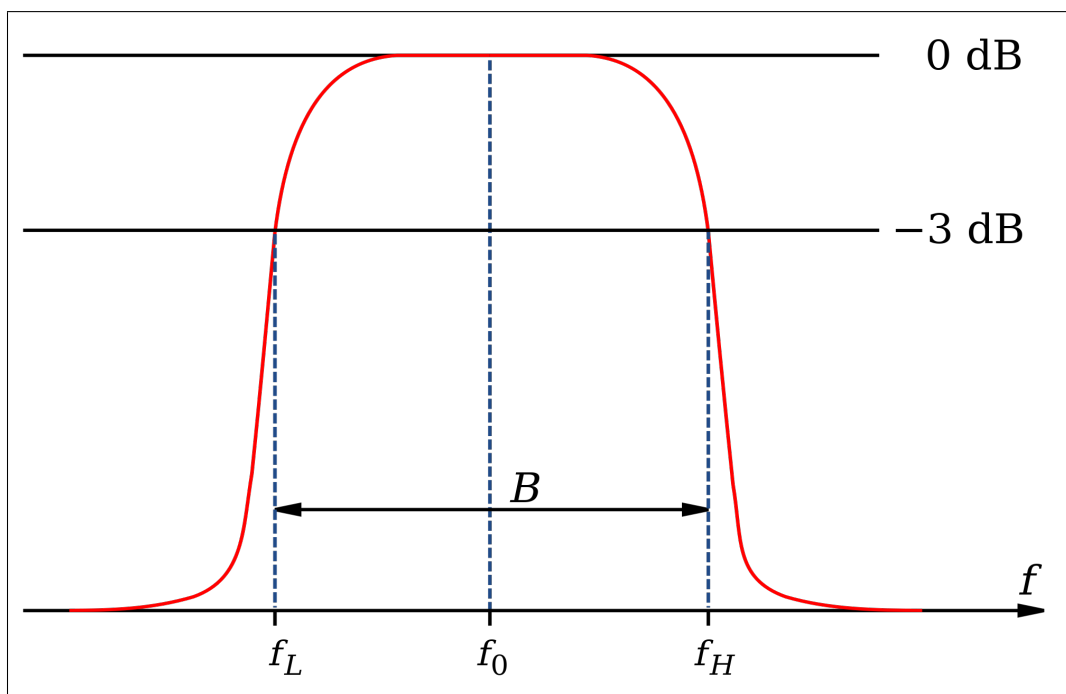


## 6 Band-Pass Filters

A band-pass filter works in a way similar to the HP filter in the way that it decomposes the data into a trend series and a cyclical series. The major difference between the band-pass filter and an HP filter is that the band-pass filter is two-sided. That is, it removes frequencies outside the chosen band that are higher than the upper cutoff and lower than the lower cutoff. By contrast, the HP filter is one-sided and removes only frequencies below a lower cutoff. As a result, the term “band-pass filter” refers to a whole family of filtering techniques.

Figure 6, which is shamelessly downloaded from Wikipedia, illustrates a typical band-pass filter, in this case for use with audio equipment. The gain of the filter is the red line. The cutoffs for the bandwidth ( $B$ ) are  $f_L$  and  $f_H$ . An ideal filter would follow the vertical dashed lines and be zero outside the bandwidth and one inside. However, with finite samples it is impossible to actually implement an ideal filter. As the red line shows, some frequencies outside the band will be passed, and some frequencies inside the band will be attenuated. A good filter is one that comes close to the step function of the ideal filter.

Figure 6: Stylized Band-Pass Filter



There are a variety of band-pass filters available for use with economic time series. The most widely used are the Baxter-King filter (Baxter and King, 1999) and the Christiano-Fitzgerald filter (Christiano and Fitzgerald, 2003). Band-pass filters specify a range of frequencies that are passed through the filter. For business cycle analysis frequencies corresponding to a band between six and thirty-two quarters are most often used.

## 7 Filter Programs

### 7.1 Python

A good source for time-series filter, including the three discussed in this chapter is the `scikits.statsmodels` package. The filters are called as `tsa.filters.hpfilter`, `tsa.filters.bkfilter` and `tsa.filters.cfilter`.

The internal documentation follows.

Hodrick-Prescott filter

```
hpfilter(X, lamb=1600)
```

Parameters:

X : array-like -- The 1d ndarray timeseries to filter of length (nobs,) or (nobs,1)

lamb : float -- The Hodrick-Prescott smoothing parameter. A value of 1600 is suggested for quarterly data. Ravn and Uhlig suggest using a value of 6.25 ( $1600/4^{**4}$ ) for annual data and 129600 ( $1600*3^{**4}$ ) for monthly data.

Returns:

cycle : array -- The estimated cycle in the data given lamb.

trend : array -- The estimated trend in the data given lamb.

Baxter-King bandpass filter

```
bkfilter(X, low=6, high=32, K=12)
```

Parameters:

X : array-like -- A 1 or 2d ndarray. If 2d, variables are assumed to be in columns.

low : float -- Minimum period for oscillations, ie., Baxter and King suggest that the Burns-Mitchell U.S. business cycle has 6 for quarterly data and 1.5 for annual data.

high : float -- Maximum period for oscillations BK suggest that the U.S. business cycle has 32 for quarterly data and 8 for annual data.

K : int -- Lead-lag length of the filter. Baxter and King propose a truncation length of 12 for quarterly data and 3 for annual data.

Returns:

Y : array -- Cyclical component of X

Christiano Fitzgerald asymmetric, random walk filter

```
cffilter(X, low=6, high=32, drift=True)
```

Parameters:

`X` : array-like -- 1 or 2d array to filter. If 2d, variables are assumed to be in columns.

`low` : float -- Minimum period of oscillations. Features below low periodicity are filtered out. Default is 6 for quarterly data, giving a 1.5 year periodicity.

`high` : float -- Maximum period of oscillations. Features above high periodicity are filtered out. Default is 32 for quarterly data, giving an 8 year periodicity.

`drift` : bool -- Whether or not to remove a trend from the data. The trend is estimated as  $\text{np.arange}(\text{nobs}) * (\text{X}[-1] - \text{X}[0]) / (\text{len}(\text{X}) - 1)$ .

Returns: `cycle` : array -- The features of 'X' between periodicities given by low and high.

`trend` : array -- The trend in the data with the cycles removed.

## 7.2 MATLAB

Time series filters in MATLAB are less centralized, but here are a few sources.

The Hodrick-Prescott filter is included in the Econometrics Toolbox. The function is `hpfiler`.

```
hpfiler(S) uses a Hodrick-Prescott filter and a default smoothing parameter of 1600 to separate the columns of S into trend and cyclical components. S is an m-by-n matrix with m samples from n time series. A plot displays each time series together with its trend (the time series with the cyclic component removed).
```

`hpfiler(S,smoothing)` applies the smoothing parameter `smoothing` to the columns of `S`. If `smoothing` is a scalar, `hpfiler` applies it to all columns. If `S` has `n` columns and `smoothing` is a conformable vector (`n`-by-1 or 1-by-`n`), `hpfiler` applies the vector components of `smoothing` to the corresponding columns of `S`.

If the smoothing parameter is 0, no smoothing takes place. As the smoothing parameter increases in value, the smoothed series becomes more linear. A smoothing parameter of `Inf` produces a linear trend component.

`T = hpfiler(...)` returns the trend components of the columns of `S` in `T`, without plotting.

`[T,C] = hpfiler(...)` returns the cyclical components of the columns of `S` in `C`, without plotting.

The Baxter-King filter may be downloaded from the Federal Reserve Bank of Atlanta. The url is <http://www.frbatlanta.org/documents/cqer/researchcq/bpf/bpassm.txt>.

```
function [fX,AA] = bpass(X,pl,pu,root,drift,ifilt,nfix,thet)
MATLAB COMMAND FOR DEFAULT FILTER: fX = bpass(X,pl,pu) Required
Inputs: X - matrix of data (T x variables) with variables
in columns pl - minimum period of oscillation of desired
component pu - maximum period of oscillation of desired component
(2<=pl<pu<infinity). Output: fX - matrix (T x nvars) containing
filtered data
```

The Christiano-Fitzgerald filter may be downloaded from the Federal Reserve Bank of Cleveland. The url is <http://clevelandfed.org/research/models/bandpass/bpassm.txt>.



```
function [fX,AA] = bpass(X,pl,pu,root,drift,ifilt,nfix,thet)
MATLAB COMMAND FOR DEFAULT FILTER: fX = bpass(X,pl,pu)
```

Required Inputs:

X - matrix of data (T x variables) with variables in columns

pl - minimum period of oscillation of desired component

pu - maximum period of oscillation of desired component

( $2 \leq pl < pu < \infty$ ).

Output:

fX - matrix (T x nvars) containing filtered data

# Exercises

## Homework 1

Spectral Density

Prove that the spectral density of  $X + Y$  is given by:

$$S_X(\omega) + S_Y(\omega) + 2\text{Re}(S_{XY}(\omega))$$

## Homework 2

Spectral Density Estimators

Find and plot the (non-smoothed) spectral density estimates for the following quarterly US time series: GDP, consumption, investment, and CPI.

## Homework 3

HP Filter

Use the HP filter to filter the time series from Exercise 2 taking  $\lambda = 1600$ . Find and plot the spectral density estimates for both the trend and cyclical parts of the filtered time series.

## Homework 4

Filter Effects on Data

Filter the data series from Exercise 2 with the first difference, HP( $\lambda = 1600$ ), and BP(6,32, $K = 8$ ) filters. Report the following moments for each data series and for each filter: mean, standard deviation, first autocorrelation, and correlation with GDP. Also plot the filtered time series comparing the bandpass filter with the first difference filter and the bandpass filter with the HP filter. Comment on what you find.

## Homework 5

Recall the minimization function for an HP filter is

$$\min_{g_t} \left\{ \sum_{t=1}^T (y_t - g_t)^2 + \lambda \sum_{t=2}^T [(g_{t+1} - g_t) - (g_t - g_{t-1})]^2 \right\} \quad (7.1)$$

- (a) Prove that as  $\lambda \rightarrow \infty$ , the filtered series  $g_t$  is linear ( $g_t = g_0 + \beta t$  for some  $\beta$ )
- (b) Let  $0 < \lambda < \infty$ . What are the first order conditions for 3.1 for  $t = 1, 2$  and the general case  $k$ ?
- (c) Using the results from (b), write a python script that finds the trended series of US investment data, GDPIC96 (available from FRED). Use the matrix equation  $Y = \Lambda \tau$ , with  $\Lambda$  containing the coefficients for  $\lambda$  from (b). The  $(m, n)^{th}$  entry corresponds to the time period  $m$ , and the row is corresponds to the  $n^{th}$  first order condition
- (d) Plot the actual data, the filtered data, and the stationary series.

## Homework 6

Using the HP filter and the GDPIC96 data from exercise 1, complete the following table for the cyclical series,  $c$ . The parameter  $\rho$  denotes the autocorrelation coefficient.

	$\lambda = 1000$	$\lambda = 1600$	$\lambda = 1800$	$\lambda = 16000$
mean(c)				
var(c)				
$\rho(c)$				

## Homework 7

Use the following filters on the log of the SP500 data series (FRED) to extract the stationary cyclical component. Report the moments listed in the table for the stationary component. Use quarterly-averages for observations. Do not truncate the Band-Pass series as you normally would (see exercise 4).

	Differencing( $y_t - y_{t-1}$ )	OLS(quadratic)	HP( $\lambda = 1600$ )	Band-Pass(6,32)
mean(c)				
var(c)				
$\rho(c)$				

## Homework 8

Repeat exercise 3, but truncate the detrended series by removing the first/last 5% of observations before finding the moments. Which series are affected the most/least?

	Differencing( $y_t - y_{t-1}$ )	OLS(quadratic)	HP( $\lambda = 1600$ )	Band-Pass(6,32)
mean(c)				
var(c)				
$\rho(c)$				

## References

- BAXTER, M., AND R. G. KING (1999): “Measuring Business Cycles: Approximate Band-Pass Filters For Economic Time Series,” *The Review of Economics and Statistics*, 81(4), 575–593.
- CHRISTIANO, L. J., AND T. J. FITZGERALD (2003): “The Band Pass Filter,” *International Economic Review*, 44(2), 435–465.
- HODRICK, R., AND E. C. PRESCOTT (1997): “Postwar U.S. Business Cycles: An Empirical Investigation,” *Journal of Money, Credit, and Banking*, 9, 1–16.
- PEDERSEN, T. M. (2001): “The Hodrick-Prescott filter, the Slutsky effect, and the distortionary effect of filters,” *Journal of Economic Dynamics and Control*, 25, 1081–1101.