Program

TRACE

(**Tr**end **A**nd **C**ycle **E**stimation)

Instructions for the User

(Beta Version: May 2003)

by

Víctor Gómez
Dirección General de Presupuestos
Subdirección General de Análisis y P.P.
Ministerio de Hacienda
Alberto Alcocer 2, 1-P, D-34
28046 Madrid
Tel.: +34–1–583.54.39
Email: vgomez@sepg.minhac.es

Abstract

A brief summary and the user instructions are presented for the program TRACE ("**Tr**end **A**nd **C**ycle **E**stimation").

TRACE is a program for the estimation of trends, cycles and even seasonally adjusted series. The filters used by the program are fixed ARMA filters of three types. First, low–pass filters to estimate trends which are the two–sided forms of the well known Butterworth filters of electrical engineering, like the Hodrick–Prescott filter. Second, band–pass filters based on Butterworth filters to estimate cycles. Third, trend-cycle filters which are the finite versions of the Wiener–Kolmogorov filters corresponding to the trend in a canonical decomposition of an airline model. Assigning appropriate values to the parameters, these last filters can also be used for seasonal adjustment.

An important feature of the program is that, when a model for the input series is available, it can be used to improve the performance of the filter at both ends of the series. Thus, the program is an alternative to the recent X12-ARIMA program of the U.S. Bureau of the Census. However, the ARMA filters used by TRACE, while capable of closely approximating an extremely wide range of gain functions, require many less parameters, forecasts and backcasts than the MA filters used by X12-ARIMA.

The main purpose of TRACE is the estimation of trends, with various degrees of smoothness, and business cycles. To this end, it is strongly recommended that the program be always used as a second step in a two–step procedure. The first step in this procedure should consist of the estimation of the trend–cycle component by means of the application of a model–based method, like the one implemented in programs TRAMO and SEATS[1]. In this way, undesirable effects, like the generation of spurious cycles, are avoided.

The program is structured so as to be used both for in–depth analysis of a few series or for automatic routine applications to a large number of series.

[1] TRAMO and SEATS are two programs for time series analysis developed by Víctor Gómez and Agustín Maravall, which are freely available at the Internet address **http://www.bde.es**

ACKNOWLEDGEMENTS

I wish to express my gratitude to Agustín Maravall for teaching me so many things about time series analysis in general and signal extraction in particular.

I also want to thank Paco Melis for introducing me to the wonderful world of digital filters. Many of his ideas have deeply influenced me.

Contents

1 Linear Time Invariant Filters

Given an infinite time series $\{z_t\}$, a linear time invariant filter that produces a new time series $\{y_t\}$ is a transformation of the form

$$y_t = \sum_{j=-\infty}^{\infty} h_j z_{t-j},$$

where $\sum_{j=-\infty}^{\infty} |h_j| < \infty$. In terms of the backward shift operator, defined by $Bz_t = z_{t-1}$, the filter can be expressed more compactly as

$$y_t = H(B)z_t = \sum_{j=-\infty}^{\infty} h_j B^j z_t, \tag{1}$$

where $H(B) = \sum_{j=-\infty}^{\infty} h_j B^j$ and $B^j = B(B^{j-1})$. The filter is said to be symmetric if $h_{-j} = h_j$ for all $j = 1, 2, \ldots$, in which case $H(B)$ can be written as

$$H(B) = h_0 + \sum_{j=1}^{\infty} h_j (B^j + B^{-j}).$$

The filter is called causal or backward–looking if $h_j = 0$ for $j < 0$. In that case, the filter output depends only on the present and past of the input series.

Every stationary process $\{z_t\}$ has a representation in terms of periodic stochastic components that resembles the representation of a periodic function as a Fourier series. This is the so called spectral representation, which mathematically is expressed as

$$z_t = \int_{-\pi}^{\pi} e^{itx} dY(x). \tag{2}$$

Here, $\{Y(x)\}$ is a stochastic process with independent increments, continuous on the right, and $i = \sqrt{-1}$. Heuristically,the previous integral can be interpreted as the limit in mean squared of sums of harmonics with stochastic coefficients. To see the effect of the linear filter (1) on the spectral representation (2), consider the following simple example. Let z_t be an elementary complex exponential function

$$z_t = e^{itx} = \cos(tx) + i\sin(tx).$$

Then, if this is passed through the filter, it becomes

$$\begin{aligned} y_t &= H(B)e^{itx} = \sum_{j=-\infty}^{\infty} h_j e^{i(t-j)x} \\ &= \left\{ \sum_{j=-\infty}^{\infty} h_j e^{-ijx} \right\} e^{itx}. \end{aligned}$$

Thus, the effects of the filter are summarized by the complex–valued function

$$\hat{H}(x) = \sum_{j=-\infty}^{\infty} e^{-ijx} h_j, \tag{3}$$

which is the Fourier transform of the sequence $(\ldots, h_{-1}, h_0, h_1, \ldots)$. Note that $\hat{H}(x)$ is obtained by replacing B in $H(B)$ with e^{-ix}. In the general case, it can be shown that the spectral representation of the transformed series y_t is

$$y_t = \int_{-\pi}^{\pi} \hat{H}(x) e^{itx} dY(x). \tag{4}$$

The function $\hat{H}(x)$ is usually called the frequency response function of the filter. Like in the previous example, equation (4) shows intuitively the effect of the filter $H(B)$ on the input series $\{z_t\}$. The effect is twofold on the harmonics at frequency $x \in [-\pi, \pi]$. First, the amplitudes are multiplied by the modulus $G(x) = |\hat{H}(x)|$ of the complex number $\hat{H}(x)$ and second, there is a shift effect measured by the argument $\phi(x)$ of $\hat{H}(x)$. The functions $G(x)$ and $\phi(x)$ are usually called gain and phase function of the filter, respectively. Note that if the filter is symmetric, there is no phase effect, because the number $\hat{H}(x)$ is a real number due to the cancellation of the sine functions in the expression (3). This feature makes symmetric filters desirable in all practical applications.

If the phase function $\phi(x)$ is divided by the frequency x, the phase delay function $\tau(x) = \phi(x)/x$ is obtained, which is a measure of the time delay experienced by the input z_t in passing through the filter.

Given the autocovariances $\gamma(j) = \mathrm{E}(z_{t+j} z_t)$ of a stationary process $\{z_t\}$ which follows an ARMA model, the spectrum of the process $f(x)$ is defined as the Fourier transform of the covariance sequence $(\ldots, \gamma(-1), \gamma(0), \gamma(1), \ldots)$, that is

$$f(x) = \frac{1}{2\pi} \sum_{j=-\infty}^{\infty} \gamma(j) e^{-ijx},$$

assuming $\sum_{j=-\infty}^{\infty} |\gamma(j)| < \infty$. For example, for a white noise series $\{a_t\}$ with $\mathrm{Var}(a_t) = \sigma_a^2$, the spectrum is $f(x) = \sigma_a^2/2\pi$. The autocovariances $\gamma(j)$ are given by the inverse Fourier transform of $f(x)$

$$\gamma(j) = \int_{-\pi}^{\pi} e^{itx} f(x). \tag{5}$$

If $f_z(x)$ is the spectrum of the series $\{z_t\}$ in (1), then the spectrum $f_y(x)$ of the filtered series $\{y_t\}$ is given by the formula

$$f_y(x) = |\hat{H}(x)|^2 f_z(x), \quad x \in [0, \pi].$$

As an application of this formula, suppose that the series $\{z_t\}$ follows the ARMA model $\phi(B)z_t = \theta(B)a_t$, where $\text{Var}(a_t) = \sigma_a^2$. Then, the series $\{z_t\}$ can be seen as the result of applying the filter $H(B) = \theta(B)/\phi(B)$ to the white noise series $\{a_t\}$. Therefore, the spectrum is

$$f(x) = \frac{1}{2\pi}\frac{\theta(e^{-ix})\theta(e^{ix})}{\phi(e^{-ix})\phi(e^{ix})}\sigma_a^2 = \frac{1}{2\pi}\frac{\mid \theta(e^{-ix}) \mid^2}{\mid \phi(e^{-ix}) \mid^2}\sigma_a^2.$$

Note that if $j = 0$ en (5), we obtain the variance of the process expressed as an integral of the spectrum. This can be seen as how the stochastic components associated with the different frequencies contribute to the variance of the series.

Traditionally, filters have been designed in terms of its desired effect on a certain band of frequencies. So, a low-pass filter is a filter which, ideally, has a gain function equal to one for the frequencies near to zero and a value of zero for all the other frequencies. More generally, a band-pass filter for the band $[x_1, x_2]$ is a filter such that the gain function $G(x) = 1$ for $x \in [x_1, x_2]$ and is zero otherwise. Note that frequency and period are related by the well known formula $x = 2\pi/T$, where x is the frequency and T is the period.

1.1 Arma Filters

Arma (auto-regressive moving-average) filters are also called I.I.R. (infinite impulse response) filters in electrical engineering. Letting z_t be the input of Arma filter, the output y_t is obtained recursively by means of an expression of the form

$$y_t + b_1 y_{t-1} + \cdots + b_n y_{t-n} = a_0 z_t + a_1 z_{t-1} + \cdots + a_m z_{t-m}.$$

Using the backshift operator B, the previous expression can be written more concisely as $y_t = H(B)z_t$, where

$$H(B) = \frac{a_0 + a_1 B + \cdots + a_m B^m}{1 + b_1 B + \cdots + b_n B^n}. \tag{6}$$

It is assumed that the polynomial $1 + b_1 B + \cdots + b_n B^n$ in the backshift operator has all its roots outside the unit circle. This last condition ensures that the function $H(B)$ can be developed as a convergent series. That is, there exist coefficients h_j, $j = 0, 1, \ldots$ such that the equality $H(B) = \sum_{j=0}^{\infty} h_j B^j$ holds (hence the name I.I.R. filters) and the series is convergent.

If the degree of the numerator in (6) is zero, we get a pure autoregressive (Ar) filter, and if the degree of the denominator is zero, the filter is a pure moving-average (Ma) filter. These last filters are also called F.I.R. (finite impulse response) filters in electrical engineering.

ARMA filters are causal filters which have a non–zero phase effect because they are not symmetric. Besides, it can be shown that ARMA filters that are not pure MA filters have a non–linear phase function, which means that the phase delay is not constant. On the contrary, MA filters have a linear phase function.

An obvious way to obtain a symmetric filter from an ARMA filter $H(B)$ is to form the product $H(F)H(B)$, where F is the forward shift operator, $Fz_t = z_{t+1}$. These two–sided forms of ARMA filters are the ones used by program TRACE.

As a simple example, consider the filter $H(B) = k/(1 + bB)$. Then, its two–sided form is the filter $H(F)H(B) = k^2/(1 + b^2 + b(B + F))$.

1.2 The Airline Trend–cycle Filter

In this section, we consider the so–called canonical model–based decomposition. This model–based method constructs models for the components from the model specified for the aggregate series. The basic references are Cleveland and Tiao (1976), Box, Hillmer and Tiao (1978), Burman (1980), Hillmer and Tiao (1982), Bell and Hillmer (1984), and Maravall and Pierce (1987).

Suppose that the observed series $\{z_t\}$ follows the invertible ARIMA model

$$\phi_z(B)z_t = \theta_z(B)a_t, \tag{7}$$

where $\phi_z(B)$ is a polynomial in B which may have unit roots, and $\{a_t\}$ is an i.i.d. sequence of $N(0, \sigma_a^2)$ random variables. Suppose further that the usual assumption is made of a decomposition into orthogonal unobserved components

$$z_t = p_t + s_t + w_t,$$

where p_t is the trend–cycle, s_t is the seasonal and w_t is the irregular component. The decomposition can in fact be additive or multiplicative. Since the latter can be transformed into the former by taking logarithms, we will suppose in what follows that the decomposition is additive.

It is assumed that the components also follow ARIMA models, which are determined by the autoregressive polynomial $\phi_z(B)$ and, in particular, its unit root part. This is because each unit root in $\phi_z(B)$ induces an infinite peak in the pseudospectrum of the series. For example, suppose a quarterly series with a seasonal difference $\nabla_4 = (1 - B)(1 + B + B^2 + B^3)$ in $\phi_z(B)$. Then, the factor $1 - B$ induces an infinite peak at the zero frequency and the other factor $1 + B + B^2 + B^3$ induces infinite peaks at the seasonal frequencies.

However, the assignment of the different roots in $\phi_z(B)$ is not enough to identify models for the components. The identification is achieved by imposing certain restrictions on the orders of the moving average parts and the so–called "canonical decomposition". This last decomposition consists of eliminating from p_t and s_t as much white noise as possible and assigning it to the irregular w_t. In this way, the variance of this last component is maximized and, on the contrary, the trend–cycle and seasonal components are made as stable as possible.

When the model (7) is the airline model

$$\nabla\nabla_s z_t = (1+\theta B)(1+\Theta B^s)a_t, \tag{8}$$

where $\nabla = 1-B$, $\nabla_s = 1-B^s$ and s is the number of observations per year, the application of this method yields the following models for the components

$$\begin{aligned} \nabla^2 p_t &= (1+\alpha B)(1+B)b_t \\ S(B)s_t &= \theta_s(B)c_t, \end{aligned} \tag{9}$$

where $S(B) = 1+B+\cdots+B^{s-1}$ and $\theta_s(B)$ is a non–invertible moving average polynomial of order $s-1$. See Maravall (1995). Note that $\nabla_s = \nabla S(B)$.

When the series is stationary and a complete realization $\{\ldots, z_{t-1}, z_t, z_{t+1}, \ldots\}$ of the process is known, the Wiener–Kolmogorov filters to optimally estimate the components are given by the ratio of the covariance generation function of the component and that of the series. By the results of Bell (1984), the Wiener–Kolmogorov filter can also be applied in the non–stationary case when an infinite realization is known. Gómez (1999) has shown that for the finite non–stationary situation Wiener–Kolmogorov filtering, Kalman filtering, and penalized least squares smoothing are equivalent.

Because in this manual we focus our interest on the trend–cycle component, suppose that the model for p_t given by the canonical decomposition is

$$\phi_p(B)p_t = \theta_p(B)b_t.$$

Then, the estimator $\hat{p}_t$ of p_t based on the infinite sample is $\hat{p}_t = H(B)H(F)z_t$, where $H(B)H(F)$ is the Wiener–Kolmogorov filter,

$$H(B) = \frac{\sigma_b}{\sigma_a}\frac{\theta_p(B)\phi_z(B)}{\phi_p(B)\theta_z(B)},$$

and $\sigma_b^2 = \mathrm{Var}(b_t)$.

In the case of the airline model (8), since p_t follows tha model (9), after some manipulations, it is obtained that

$$H(B) = \frac{\sigma_b}{\sigma_a}\frac{(1+\alpha B)(1+B)S(B)}{(1+\theta B)(1+\Theta B^s)}. \tag{10}$$

The filter $H(B)H(F)$ is doubly–infinite symmetric, but convergent because the model is assumed to be invertible. Thus, this last filter can be expressed as $H(B)H(F) = h_0 + \sum_{j=1}^{\infty} h_j(B^j + F^j)$.

The parameters α and σ_b/σ_a of expression (10) can be easily computed from the formulae

$$\begin{aligned}
\frac{\sigma_b}{\sigma_a} &= \frac{(1+\theta)(1+\Theta)}{2(1+\alpha)s} \\
\alpha &= \frac{1-L-\sqrt{1-2L}}{L} \\
L &= \frac{2\theta}{(1+\theta)^2} + \frac{2\Theta s^2}{(1+\Theta)^2} - \frac{s^2+2}{6},
\end{aligned}$$

where $s = 12$ for monthly and $s = 4$ for quarterly series. The proof of this result can be seen in Gómez and Bengoechea (1998). Note that the filter values α and σ_b/σ_a depend only on the parameters θ and Θ.

1.3 Butterworth Filters and Band–pass Filters Derived From Them

We start by considering the filter proposed by Hodrick and Prescott (1980), hereafter referred to as the HP filter. It is well known (see, for example, King and Rebelo, 1989), that the HP filter admits a signal extraction interpretation. Thus, the filter is obtained as the filter that corresponds to the estimator of s_t in the signal–plus–noise model

$$z_t = s_t + n_t, \tag{11}$$

under the assumption that the signal s_t follows the model $\nabla^2 s_t = b_t$ and $\{b_t\}$ is a white noise sequence with mean zero and variance 1, independent of the white noise sequence $\{n_t\}$. Hodrick and Prescott (1980) suggested with great success the value $\lambda = \mathrm{Var}(n_t) = 1600$ when z_t is a quarterly series. Because s_t and z_t are nonstationary, under assumption A of Bell (1984), the Wiener–Kolmogorov filter can be applied to a infinite realization of z_t to obtain the minimum mean squared error estimator $\hat{s}_t$ of the signal s_t. The estimator $\hat{s}_t$ is given by an infinite symmetric filter $H_{HP}(B,F)$

$$\hat{s}_t = H_{HP}(B,F)z_t = \nu_0 z_t + \sum_{k=1}^{\infty} \nu_k(B^k + F^k)z_t. \tag{12}$$

The weights ν_t can be obtained from the signal extraction formula

$$H_{HP}(B,F) = 1/(1+\lambda(1-B)^2(1-F)^2). \tag{13}$$

The frequency response function $\hat{H}_{HP}(x)$ of the filter $H_{HP}(B,F)$ is obtained by replacing B with e^{-ix} in (13). After some manipulation, we get

$$\hat{H}_{HP}(x) = \frac{1}{1 + \left(\frac{\sin(x/2)}{\sin(x_c/2)}\right)^4}, \tag{14}$$

where x_c is the frequency that corresponds to $\hat{H}_{HP}(x) = 1/2$ and $\lambda = 1/(16\sin^4(x_c/2))$. Because (14) is a real number, it coincides with the gain function of the filter and there is no phase effect. In the case of the HP filter, $\lambda = 1600$ implies $x_c = .1583$, which corresponds to a period of 9.9 years.

1.3.1 Butterworth Filters

Expression (14) is a special case of the squared gain of a Butterworth filter based on the sine function (BFS), which is given by

$$|G(x)|^2 = \frac{1}{1 + \left(\frac{\sin(x/2)}{\sin(x_c/2)}\right)^{2d}}, \tag{15}$$

where $|G(x_c)|^2 = 1/2$. These filters are low–pass filters that depend on two parameters, d and x_c, and, if x_c is fixed, the effect of increasing d is to make the fall sharper. See figure 1. They are autoregressive filters of the form $H(B) = 1/\theta(B)$, where $\theta(B) = \theta_0 + \theta_1 B + \cdots + \theta_d B^d$ and $|G(x)|^2 = H(e^{-ix})H(e^{ix})$. It can be shown that the denominator in (13) can be factored as $\theta(B)\theta(F)$, where $\theta(B)$ is a polynomial in B of degree 2. Thus, the filter $H_{HP}(B,F)$ is a two–sided form of a BFS, because $H_{HP}(B,F) = H(B)H(F)$, where $H(B) = 1/\theta(B)$.

Butterworth filters are of two types. The first one is based on the sine function and has already been described (BFS), whereas the second one is based on the tangent function (BFT). See Otnes and Enochson (1978).

The squared gain function of a BFT is given by (15) with the "sin" replaced with "tan". Thus, the squared gain function of a BFT is

$$|G(x)|^2 = \frac{1}{1 + \left(\frac{\tan(x/2)}{\tan(x_c/2)}\right)^{2d}}, \tag{16}$$

where, like in the case of a BFS, $|G(x_c)|^2 = 1/2$. The effect of increasing d is also to make the fall sharper, as with BFS.

It is a remarkable fact that two–sided BFS and BFT can be obtained as best linear estimators, in the mean squared sense, of the signal in models like

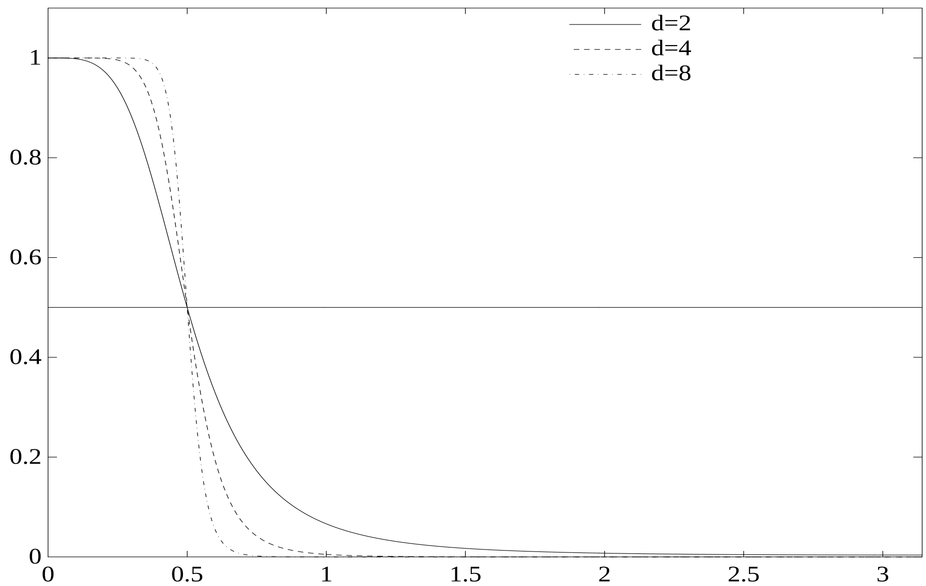

Fig1. : Squared Gain of Butterworth Filters

(11). The decomposition is given by an IMA$(d,0)$ signal plus white noise for BFS, and by an IMA(d,d) signal plus white noise for BFT; in this last case the MA polynomial is $(1+B)^d$. Thus, for example, when $d=1$, the BFS yields the "random walk plus noise" model, and the BFT its canonical version (because there is a spectral zero at frequency $x=\pi$). When $d=2$, as we saw earlier, the BFS yields the HP filter. For a proof of these results, see Gmez (1998).

Given the previous model–based interpretation of Butterworth filters, it can be shown that the Wiener–Kolmogorov filter H_{BFS} which defines a two–sided BFS is

$$H_{BFS}(B,F) = \frac{1}{1+\lambda(1-B)^d(1-F)^d} \tag{17}$$

and that $H_{BFT}(B,F)$ for a two–sided BFT is

$$H_{BFT}(B,F) = \frac{(1+B)^d(1+F)^d}{(1+B)^d(1+F)^d+\lambda(1-B)^d(1-F)^d} \tag{18}$$

where $\lambda = \sigma_n^2/\sigma_b^2$. The denominators in (17) and (18) can be factored in the form $\theta(B)\theta(F)\sigma_a^2$, where $\nabla^d z_t = \theta(b)a_t$ is the model followed by z_t in (11) and $\mathrm{Var}(a_t) = \sigma_a^2$.

1.3.2 Band–pass Filters

Band–pass filters are filters that pass only those components whose frequencies are in a pre–selected band. These filters can be obtained from low–pass filters by means of a transformation. See Oppenheim and Schaffer (1989), pp. 430–434. Let $[x_{p_1}, x_{p_2}]$, where $x_{p_1} \geq 0$ and $x_{p_1} < x_{p_2} \leq \pi$, be the pass band. Then, a suitable transformation is $z = -s(s-\alpha)/(1-\alpha s)$, where $\alpha = \cos((x_{p_2}+x_{p_1})/2)/\cos((x_{p_2}-x_{p_1})/2)$. It can be shown that $-1 < \alpha < 1$.

If we apply the previous transformation to a two–sided BFS, then the following band–pass filter $H_{pbs}(B, F)$ is obtained

$$H_{pbs}(B,F) = \frac{(1-\alpha B)^d(1-\alpha F)^d}{(1-\alpha B)^d(1-\alpha F)^d + \lambda(1-2\alpha B+B^2)^d(1-2\alpha F+F^2)^d},$$

which is the Wiener–Kolmogorov filter to estimate the signal s_t in model (11) when s_t follows the model $(1-2\alpha B+B^2)^d s_t = (1-\alpha B)^d b_t$ and $\lambda = \sigma_n^2/\sigma_b^2$. Because $-1 < \alpha < 1$, we can write $\alpha = \cos\theta$ for a certain $\theta \in [0, \pi]$, so that the polynomial $1 - 2\alpha B + B^2$ has two complex conjugate roots of unit modulus.

If the previous transformation is applied to a two–sided BFT, then the following band–pass filter $H_{pbt}(B, F)$ is obtained

$$H_{pbt}(B,F) = \frac{(1-B^2)^d(1-F^2)^d}{(1-B^2)^d(1-F^2)^d + \lambda(1-2\alpha B+B^2)^d(1-2\alpha F+F^2)^d},$$

which is the Wiener–Kolmogorov filter to estimate the signal s_t in model (11) when s_t follows the model $(1-2\alpha B+B^2)^d s_t = (1-B^2)^d b_t$ and $\lambda = \sigma_n^2/\sigma_b^2$. Note that in this case the pseudospectrum of the signal is canonical, because it is zero at both the zero and the π frequencies.

The denominators of $H_{pbs}(B, F)$ and $H_{pbt}(B, F)$ can be factored as $\theta(B) \times\theta(F)\sigma_a^2$.

1.3.3 Design of Butterworth Filters

A direct way to design BFS or BFT consists of specifying the frequency x_c where the squared gain function is equal to 1/2, or the noise to signal ratio $\lambda = \sigma_n^2/\sigma_b^2$, and the degree of the filter d in (15).

To design a two–sided BFS, let δ_1, δ_2, x_p and x_s be the specification parameters, so that the gain function $G(x)$, which is the squared of the gain function of the corresponding BFS, should verify $1-\delta_1 < G(x) \leq 1$ for $x \in [0, x_p]$ and $0 \leq G(x) < \delta_2$ for $x \in [x_s, \pi]$. Here, $[0, x_p]$ is the pass band, $[x_s, \pi]$ is the stop band, $[x_p, x_s]$ is the transition band, and δ_1 and δ_2 are the tolerances. Note that the pass band includes 0 because Butterworth filters are low–pass filters.

Because $\sin^2(x/2) = \tan^2(x/2)/(1 + \tan^2(x/2))$, we can obtain d and x_c by solving the equations

$$\begin{aligned}
1 + \left(\frac{\tan^2(x_p/2)}{1 + \tan^2(x_p/2)} \times \frac{1 + \tan^2(x_c/2)}{\tan^2(x_c/2)} \right)^d &= \frac{1}{1 - \delta_1} \\
1 + \left(\frac{\tan^2(x_s/2)}{1 + \tan^2(x_s/2)} \times \frac{1 + \tan^2(x_c/2)}{\tan^2(x_c/2)} \right)^d &= \frac{1}{\delta_2}.
\end{aligned}$$

First, d is obtained. Since d has to be an integer in (15), if the value of d obtained by solving the previous equations is not an integer, the nearest integer is selected. Then, the value of x_c is obtained which corresponds to this integer d in the earlier equations.

The equations to be solved for the design of a two–sided BFT are

$$\begin{aligned}
1 + \left(\frac{\tan(x_p/2)}{\tan(x_c/2)} \right)^{2d} &= \frac{1}{1 - \delta_1} \\
1 + \left(\frac{\tan(x_s/2)}{\tan(x_c/2)} \right)^{2d} &= \frac{1}{\delta_2}.
\end{aligned}$$

The way to proceed is like for two–sided BFS.

1.3.4 Design of Band–pass Filters

If we want to design a two–sided band–pass filter and the specifications are given by means of the parameters δ_1, δ_2, $x_{p,1}$, $x_{p,2}$, $x_{s,1}$ and $x_{s,2}$, so that the gain function $G(x)$ should verify $1 - \delta_1 < G(x) \leq 1$ for $x \in [x_{p,1}, x_{p,2}]$ and $0 \leq G(x) < \delta_2$ for $x \in [0, x_{s1}]$ and $x \in [x_{s,2}, \pi]$, we may proceed as follows. First, let $x_p = x_{p,2} - x_{p,1}$ and $x_s = x_{s,2} - x_{p,1}$ and design a low–pass filter with the specifications parameters δ_1, δ_2, x_p and x_s. Then, apply the transformation of the previous section to this low–pass filter to obtain the band–pass filter.

Note that we have not used $x_{s,1}$ in the procedure we have just described to design a band–pass filter. We have implicitly assumed that $x_{s,1}$ is the symmetrical point of $x_{s,2}$ with respect to $(x_{p,1} + x_{p,2})/2$.

1.4 Finite Versions of Arma Filters Which Admit a Model–based Interpretation

Many ARMA filters admit a model–based interpretation. For example, we have seen earlier that the two–sided forms of Butterworth filters based on the sine and the tangent function are low–pass filters which can be obtained as the Wiener–Kolmogorov filters to estimate the signal s_t in the signal–plus–noise model

(11) when s_t follows certain models. However, Wiener–Kolmogorov filters are doubly–infinite symmetric filters and a complete realization $\{\ldots, z_{-1}, z_0, z_1, \ldots,\}$ is required for their application. These filters are of the form $H(B)H(F) = h_0 + \sum_{j=1}^{\infty} h_j(B^j + F^j)$, where the series is convergent because the model for the series z_t is assumed to be invertible.

In the more real situation, in which only a finite observed series is available $z = (z_1, \ldots, z_N)'$, the usual practice consists of applying the same filter, but with the unknown observations replaced by forecasts or backcasts. That is, one computes

$$\hat{s}_t = h_0\hat{z}_t + \sum_{j=1}^{\infty} h_j(\hat{z}_{t-j} + \hat{z}_{t+j}),$$

where $\hat{z}_s$ is equal to z_s if $1 \leq s \leq N$ or else it is a forecast or backcast. To compute the estimator $\hat{s}_t$ based on the finite sample, any of the three equivalent algorithms described in Gmez (1999) can be used. The simplest of these algorithms is probably that based on Tunnicliffe Wilson's algorithm applied as described by Burman (1980).

The finite version of Arma filters so obtained is no longer time invariant. The filtered series is of the form $\hat{s}_t = \sum_{t=1}^{N} h_{jt} z_j$ and the weights h_{jt} depend on the date t as well as the lead/lag index j. The finite filter can behave very differently from the infinite filter at both ends of the series and this fact should be borne in mind.

If the model followed by the series is very different from the model which defines the filter, the behavior of the finite filter can be very poor. This is particularly true for band–pass filters, where the model implied by the filter is very seldom, if ever, encountered in practice.

1.5 Arma filters–Arima

Suppose we are interested in applying one of the filters of the previous sections, which admits a model–based interpretation, to a certain series z_t to estimate trends or business–cycles. Let the process $\{z_t\}$ follow the Arima model $\phi(B)z_t = \theta(B)a_t$. In general, this model will be different from the models implied by the filters. The question then naturally arises as to whether it is possible to use this model in order to improve the performance of the filter at both ends of the series.

An algorithm to implement the filter which uses the model followed by the series z_t can be obtained as follows. Let the filter be $H(B)H(F)$, where $H(B) = \gamma(B)/\beta(B)$. Suppose first that a complete realization of the process $\{z_t\}$ is known and let $\{x_t\}$ and $\{y_t\}$ be the filtered series $x_t = H(F)H(B)z_t$ and $y_t = H(B)z_t$, respectively. Then, using the Arima model for z_t and the definition of $H(B)$, it is easy to verify that y_t follows the model $\phi(B)\beta(B)y_t = \theta(B)\gamma(B)a_t$, where the a_t are the innovations of z_t. This, together with the fact that the series z_t also follows the backward model $\phi(F)z_t = \theta(F)v_t$, implies, after projecting onto the finite sample $z = (z_1, \ldots, z_N)'$, the relations $\phi(B)\beta(B)y_t = 0$, $t \geq N+q+a+1$, and $\phi(F)z_t = 0$, $t \leq -q$, where q is the degree of $\theta(B)$ and a is the degree of $\gamma(B)$. Let p and b be the degrees of $\phi(B)$ and $\beta(B)$. Then, the algorithm is as follows:

1. Solve the system

$$\begin{aligned} \beta(B)y_t &= \gamma(B)z_t \quad t = -q+1, \ldots, p-q \\ \phi(F)y_t &= 0 \qquad\qquad t = -q-b+1, \ldots, -q \end{aligned}$$

   where $q+a$ backcasts are needed: $\hat{z}_{-q-a+1}, \ldots, \hat{z}_0$. For $t = p-q+1, \ldots, N+q+2a$, obtain y_t from the recursion $\beta(B)y_t = \gamma(B)z_t$, where $q+2a$ forecasts are needed: $\hat{z}_{N+1}, \ldots, \hat{z}_{N+q+2a}$.

2. Solve the system

$$\begin{aligned} \beta(F)x_t &= \gamma(F)y_t \quad t = N+q+a-b-p+1, \ldots, N+q+a \\ \phi(B)\beta(B)x_t &= 0 \qquad\qquad t = N+q+a+1, \ldots, N+q+a+b \end{aligned}$$

   For $t = N+q+a-b-p, \ldots, 1$, obtain x_t from the recursion $\beta(F)s_t = \gamma(F)y_t$.

Note that the previous algorithm is similar to that proposed by Tunnicliffe Wilson, as applied by Burman (1980). If $\beta(B) = 1$, the filter is a symmetric moving average, which is factored as $\gamma(F)\gamma(B)$. In this case, it is not necessary to solve the two systems. All that is needed is to generate first y_t from $y_t = \gamma(B)z_t$ and then x_t from $x_t = \gamma(F)y_t$. To that end, a backcasts and a forecasts are required. Thus, the previous algorithm can be considered as a generalization to Arma filters of the procedure used by the program X11–Arima for finite moving average filters (see Dagum, 1980). For this reason, we call the filters given by the previous algorithm Arma filters–Arima. By using the Arima model followed by the input series as previously described, the performance of the filter at both ends of the series is improved with respect to that of the original filter, just as the X11–Arima filters are an improvement over the X11 filters.

2 Program TRACE

2.1 Brief Description of the Program

TRACE ("**T**rend **A**nd **C**ycle **E**stimation") is a program written in Fortran for PCs under MS–DOS. The program estimates trends and cycles by means of the application of two–sided ARMA filters to the input series. These filters are doubly–infinite and symmetric, and constitute a generalization of the usual finite symmetric moving average filters used, for example, in X–11. To improve the estimations at both ends of the series, a model for the input series can be used, in a way similar to that used in the popular X–11–ARIMA (see Dagum 1980).

The use of ARMA filters, or I.I.R. filters in the electrical engineering terminology, has two main advantages with respect to the usual finite moving average filters. First, because rational functions are used instead of polynomials in the backshift operator, a better approximation to an ideal gain function is possible. Second, the number of forecasts and backcasts required for the finite sample implementation of ARMA filters is much smaller than for finite moving average filters.

All of the ARMA filters used by TRACE can be obtained as Wiener–Kolmogorov filters in certain signal extraction problems. Thus, the computation of the output of ARMA filters can be made very simple by using Tunnicliffe Wilson's algorithm as described by Burman (1980). When a model for the input series is available, the algorithm of last section can be applied to improve the performance of the filter at both ends of the series.

For the estimation of trends, low–pass filters are available which can be designed in such a way that the resulting filter approximates an ideal filter as much as desired. These low–pass filters belong to the family of two–sided Butterworth filters, which can be based either on the sine (BFS) or the tangent function (BFT). The formulae for these filters are given by (17) for the BFS and (18) for the BFT.

Business cycles can be estimated by means of band–pass filters, which are obtained from tangent Butterworth filters in the manner described in the previous section. They can also be estimated as deviations from long–term trends estimated with Butterworth filters.

In TRACE, it is possible to estimate trend–cycle components by using airline model–based filters. These filters depend on two parameters, θ and Θ, which are the two moving average parameters of the airline model from which the filter is derived. The Wiener–Kolmogorov formula for these filters is given by (10).

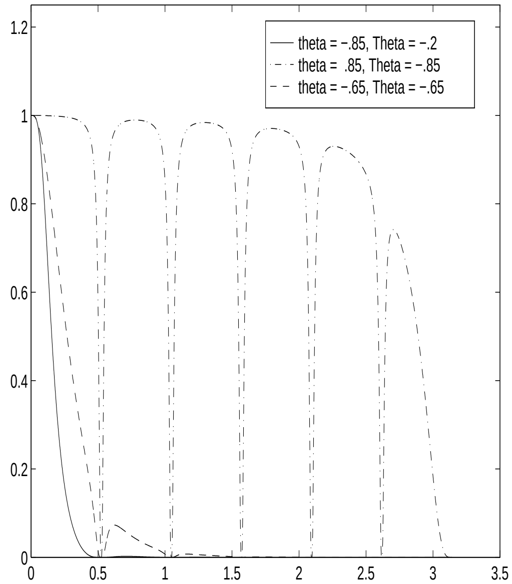

Fig. 2a) Airline trend–cycle filters: Squared gain function

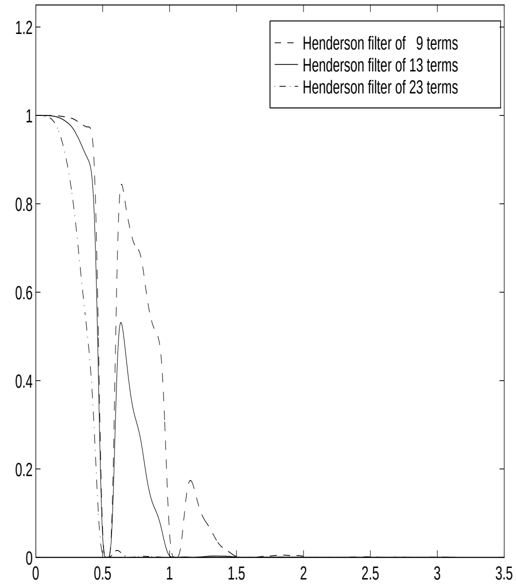

Fig. 2b) X11 Trend–cycle filters: Squared gain function

The airline trend–cycle filters are similar to the trend–cycle filters of X11, see figure 2. The parameters θ and Θ of the airline model determine the degree of stability of the trend–cycle and the seasonal components of the series which follows that model. Thus, if θ, for example, is close to -1, the trend–cycle is very stable and the gain of the trend–cycle filter is very narrow in the neighborhood of the zero frequency. The same principle applies to the seasonal component. When Θ tends to -1, the seasonal component becomes more and more deterministic and the gain of the seasonal filter gets more and more narrow around the seasonal frequencies.

To design an airline trend–cycle filter, an airline model should be first fitted to the input series. This can be done with, for example, program Tramo, mentioned earlier. Then, the estimated parameters are the values that should be given to the airline trend–cycle filter. In this way, trend–cycle (Arma) filters can be constructed which are an alternative to the trend–cycle (moving average) filters of X11. Note that the parameter estimates of the airline model give an indication of the stability of the trend–cycle and seasonal components, as mentioned earlier. These estimates can be considered as a replacement of the signal–to–noise ratio given by X11.

If one chooses for the θ parameter in the airline trend–cycle filter a value close to 1, for example $\theta = .985$, a trend–cycle filter is obtained which behaves very much like a seasonal adjustment filter. This can be verified using (10) and

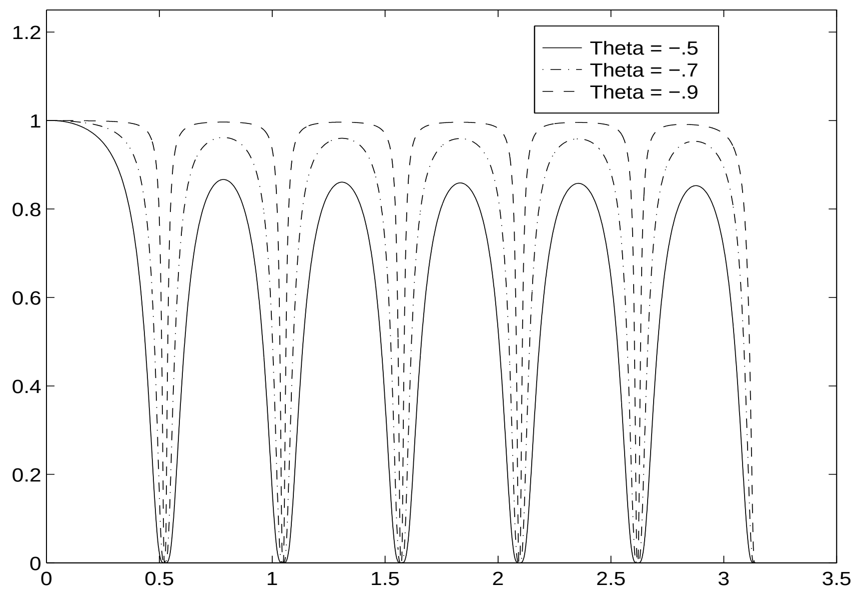

Fig. 3: Squared gain of airline filters for $\theta = .985$ and three values of Θ

the formulae given immediately after it in the last section. Fixing, for example $\theta = .985$, and letting Θ vary in the interval $[-.5, -.98]$, say, we obtain different seasonal adjustment filters. The closer Θ is to -1, the narrower the gain of the filter becomes in the neighborhood of the seasonal frequencies. In this respect, we can say that when $\theta = .985$ and Θ is close to -1, the filter behaves like a "comb" filter with respect to the seasonal frequencies. See figure 3.

To perform seasonal adjustment with Trace, a model should be first fitted to the input series. This model should be a multiplicative seasonal Arima model with $(0, 1, 1)$ modeling the seasonal part. It can be automatically identified and estimated using program Tramo, although a simple way to proceed is to specify an airline model. Then, the estimated value of Θ corresponding to the seasonal part $(0, 1, 1)$ is the value that should be given to the airline filter, together with $\theta = 0.985$, as described earlier. Since the model that defines the filter is in this case not likely to be found in practice, it is strongly recommended to apply the filter with a plausible model for the input series.

When a model for the series is not available and the filter is not a band–pass filter, the program uses the model that defines the filter and obtains the filtered series by means of signal extraction. This can be done because all of the filters considered in the program admit a model–based interpretation. The algorithm used by default is a cascade implementation of Tunnicliffe Wilson's algorithm, which is described in Gmez (1998). Also available are the Kalman filter and smoother and penalized least squares smoothing. By the results of

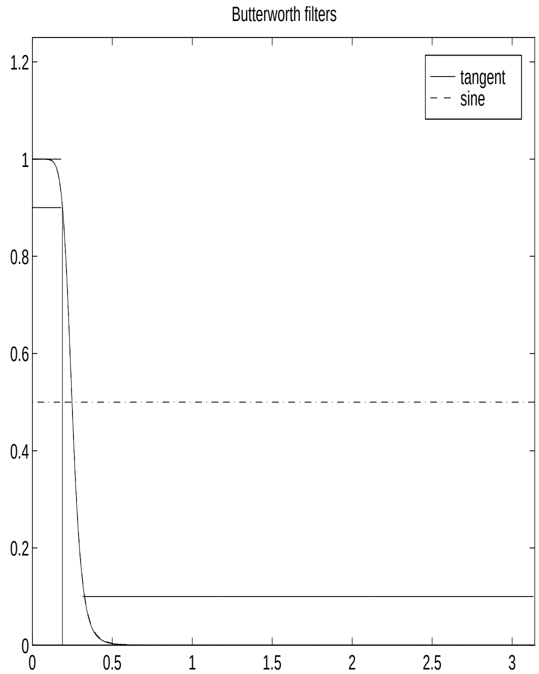

Fig. 4: Design of a low–pass filter for $\delta_1 = \delta_2 = .1$, $x_p = .06\pi$ and $x_s = .1\pi$

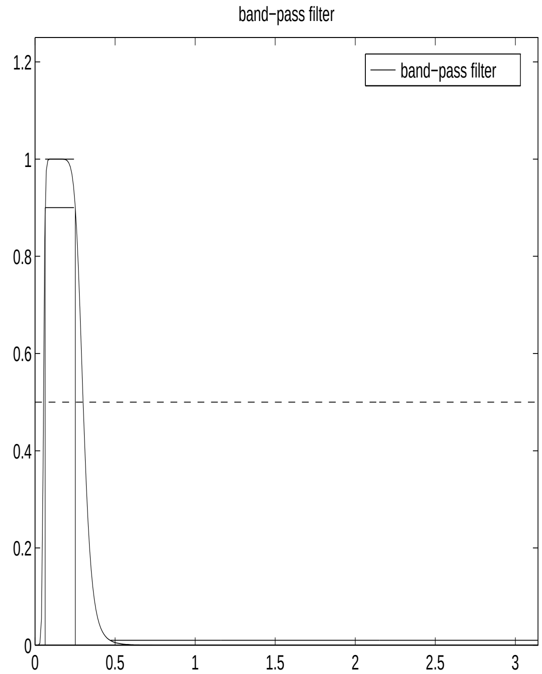

Fig. 5: Design of a band–pass filter for $\delta_1 = .1$, $\delta_2 = .01$, $x_{p,1} = .02\pi$, $x_{p,1} = .08\pi$ and $x_s = .15\pi$

Gmez (1999), the three methods should give identical results.

In the case of band–pass filters, the model which defines the filter is usually very different from the models encountered in practice and a plausible model for the input series must be given to the program.

As mentioned earlier, when an Arima model for the input series is available, the program uses the algorithm described in the last section to improve the estimation at both ends of the series.

The program allows for the design of two–sided Butterworth filters by means of the tolerance parameters, δ_1 and δ_2, the pass band $[0, x_p]$ and the stop band $[x_s, \pi]$, as described in the last section. See figure 4. They can also be designed more directly, by means of the frequency x_c where the gain function is equal to $1/2$, or the noise to signal ratio $\lambda = \sigma_n^2/\sigma_b^2$, and the degree of the filter d in (15) or (16).

The program also allows for the design of band–pass filters derived from tangent Butterworth filters. To this end, the tolerance parameters, δ_1 and δ_2, the pass band $[x_{p,1}, x_{p,2}]$, and the stop band $[x_{s,2}, \pi]$, described in the last section, must be given to the program. See figure 5. An alternative way of designing these filters consists of designing first the tangent Butterworth filter on which the band–pass filter is based, along the lines of section 1.3.4, and then giving to the program the parameters of this filter, d and x_c or λ, together with the pass band $[x_{p,1}, x_{p,2}]$.

Program TRACE allows for the application of a single filter, which can be an airline, Butterworth or band–pass filter, or a product of two filters. In the latter case, the filters are an airline filter and a low–pass or band–pass filter. The filters used by TRACE allow to mimic the X11 procedure to estimate the different components. For example, to estimate the trend–cycle component, the first filter would be the airline filter appropriate for seasonal adjustment described earlier, where $\theta = .985$, and the second filter would be a two–sided sine or tangent Butterworth filter. This last filter would be the equivalent of the Henderson filters of X11.

The program can generate arrays containing the squared gain functions and the specification lines used in the design of the different filters. It can also generate MATLAB and GNUPLOT code to obtain graphs for the designed airline, two–sided Butterworth and band–pass filters, as well as for the product filters previously described.

It has already pointed out that fixed filters can generate spurious cycles (Slutsky effect). This is particularly dangerous when one is interested in estimating long–term (smooth) trends or business cycles, which in TRACE are obtained by applying two–sided Butterworth or band–pass filters. For this reason, a two–step procedure is strongly recommended when two–sided Butterworth or band–pass filters are to be used by the program to estimate long–term trends or business cycles. The application of TRACE should always be the second step in this two–step procedure. The first step should consist of the estimation of the trend–cycle component by means of the application of a model–based method, like the one implemented in programs TRAMO and SEATS. In this way, the generation of spurious cycles is prevented because the first filter, which is the model–based filter, guarantees that if there is no power in a certain frequency band, the gain of the filter in that band will be zero. For example, if the input series is white noise, the model–based filter would be zero.

If a Butterworth or a band–pass filter is to be applied to the trend–cycle component estimated with TRAMO and SEATS in the second step of the previous two–step procedure, the model which should be specified for the input series is the model followed by the theoretical trend–cycle component given by the model–based method. This information is given by program SEATS.

2.2 Instructions for the User

2.2.1 Installation

Create a directory, for example TRACE. Decompress the file in this directory.

2.2.2 Execution of the Program

Before executing the program, the user has to enter the relevant information in the file SERIE.TRC, which is located in the directory where TRACE was installed. The structure of this file will be described later. In the directory SPECTRC, the user can find several specification files for the most often used quarterly, monthly and yearly filters.

Once the file SERIE.TRC is ready, the program is executed by typing TRACE. This command is not case sensitive.

2.2.3 Output Files

You can see the result of the program by editing or printing the file *serie-name*.TRC in the subdirectory OUTPUT. The filtered series is in the file *serie-name*.DAT in the GNUPLOT subdirectory.

2.2.4 Printing TRACE User's Manual

In the directory TRACEMAN you can find the file TRACEMAN.PDF containing the program manual.

2.3 Description of the Input Parameters

2.3.1 Filter Type

In TRACE, the filter type is controlled with the following parameter:

Parameter		Meaning	Default
FILTER	= 1	Symmetrized Butterworth filter of the sine version (BFS)	1
	= 2	Symmetrized Butterworth filter of the tangent version (BFT)	
	= 3	Band–pass filter derived from a Butterworth filter of the tangent version (BPFT)	
	= 4	Airline trend–cycle filter (ATF)	
	= 5	Product of ATF and BFS	
	= 6	Product of ATF and BFT	
	= 7	Product of ATF and BPFT	

2.3.2 Filtering Method

In TRACE, the filtering method is controlled with the following parameter:

Parameter		Meaning	Default
METHOD	= 0	No filter is applied	0
	= 1	Wiener–Kolmogorov filtering is applied; the forecasts and backcasts are obtained with the model that defines the filter	
	= 2	Wiener–Kolmogorov filtering is applied; the forecasts and backcasts are obtained with a model entered by the user	
	= 3	Kalman filtering is applied; the forecasts and backcasts are obtained with the model that defines the filter	
	= 4	Penalized least squares smoothing is applied; the forecasts and backcasts are obtained with the model that defines the filter	

2.3.3 Low–pass Filter Design

In TRACE, low–pass filters can be designed in three different ways. The first one consists of specifying the parameters d and x_c of (15) or (16). In the second one, the parameter d of (15) or (16) and the parameter λ of (17) or (18) are given to the program. The third one uses the parameters δ_1 and δ_2 of Section 1.3.3 to set the tolerance limits in the pass and the stop band, and the parameters x_p and x_s of the same Section to set the limits of the pass and the stop bands.

The following parameters are used in TRACE to design low–pass filters:

Parameter		Meaning	Default
DI	=	Parameter d in the exponent of (15) or (16); increasing d makes the fall of the filter sharper	0
XC	=	Parameter x_c in the denominator of (15) or (16); it is the frequency at which the gain of the filter is 1/2, and must be entered divided by π	0
LAMBDA	=	Parameter λ in the denominator of (17) or (18); it is the quotient σ_n^2/σ_b^2 of variances of the innovations of n_t and s_t in model (11)	0
DD	=	Array of two elements, containing the parameters δ_1 and δ_2 of Section 1.3.3 used to set the tolerance limits in the pass and the stop band	All 0
XP	=	Parameter x_p of Section 1.3.3 used to set the limit of the pass band; it is a frequency, which must be entered divided by π	0
XS	=	Parameter x_s of Section 1.3.3 used to set the limit of the stop band; it is a frequency, which must be entered divided by π	0

2.3.4 Band–pass Filter Design

In TRACE, only band–pass filters based on tangent Butterworth filters are considered. They must always be applied with a model for the series, because the model which defines the filter is usually too unreliable.

These band–pass filters can be designed in TRACE in three different ways. The first one consists of specifying the parameters d and x_c of the tangent Butterworth filter on which the band–pass filter is based, see section 1.3.4, plus the parameters $x_{p,1}$ and $x_{p,2}$ to set the pass band $[x_{p,1}, x_{p,2}]$. In the second one, the parameters d and λ of the tangent Butterworth filter on which the band–pass filter is based, together with the parameters $x_{p,1}$ and $x_{p,2}$ to set the pass band, are given to the program. The third one uses the parameters δ_1 and δ_2 of Section 1.3.4 to set the tolerance limits in the pass and the stop band, and the parameters $x_{p,1}$, $x_{p,2}$ and $x_{s,2}$ of the same Section to set the limits of the pass and the stop bands.

The following parameters are used in TRACE for designing band–pass filters:

Parameter		Meaning	Default
DI	=	Parameter d of the tangent Butterworth filter on which the band–pass filter is based	0
XC	=	Parameter x_c of the tangent Butterworth filter on which the band–pass filter is based; it is a frequency, which must be entered divided by π	0
LAMBDA	=	Parameter λ of the tangent Butterworth filter on which the band–pass filter is based	0
DD	=	Array of two elements, containing the parameters δ_1 and δ_2 of Section 1.3.4 used to set the tolerance limits in the pass and the stop band	All 0
XP	=	Parameter $x_{p,1}$ of Section 1.3.4 used to set the first limit of the pass band; it is a frequency, which must be entered divided by π	0
XP2	=	Parameter $x_{p,2}$ of Section 1.3.4 used to set the second limit of the pass band; it is a frequency, which must be entered divided by π	0
XS	=	Parameter $x_{s,2}$ of Section 1.3.4 used to set the limits of the stop bands; it is a frequency, which must be entered divided by π	0

2.3.5 Airline Trend–cycle Filter Design

The following parameters are used in TRACE for designing airline filters:

Parameter		Meaning	Default
T1	=	Parameter θ (true sign) of the regular moving average part of the airline model	0
TS	=	Parameter Θ (true sign) of the seasonal moving average part of the airline model	0
MS	=	Length of the seasonal period in the airline model	0

2.3.6 ARIMA Model for the Input Series

MQ	=	Number of seasons (12 for monthly, 6 for bimonthly, 4 for quarterly, 1 for annual, and so on).		1
LAM	= 1		No transformation of data	0
	= 0		Take logs of data	
D	=	# of non–seasonal differences		1
BD	=	# of seasonal differences		1
P	=	# of non–seasonal autoregressive terms		0
BP	=	# of seasonal autoregressive terms		0
Q	=	# of non–seasonal moving average terms		1
BQ	=	# of seasonal moving average terms		1
TH	=	*Q* values for the parameters of the regular moving average parameters.		All -.1
BTH	=	*BQ* values for the parameters of the seasonal moving average parameters.		All -.1
PHI	=	*P* values for the parameters of the regular autoregressive parameters.		All -.1
BPHI	=	*BP* values for the parameters of the seasonal autoregressive parameters.		All -.1
READ	= ‘Y’		The program reads in the output file of TRAMO or SEATS the parameters MQ, LAM, D, BD, P, BP, Q, BQ, TH, BTH, PHI, and BPHI, of the ARIMA model for the theoretical component (SEATS) or the (linearized) aggregate series (TRAMO). Used with METHOD=2	‘N’
	= ‘N’		The ARIMA model parameters are entered by the user if METHOD=2	
PATH	=	Path for the file containing the output of TRAMO or SEATS, for example: ‘C:\TRAMO\OUTPUT\AIRLINE.OUT’		
COMP	= ‘TREND’		The program reads the model for the theoretical trend–cycle component (SEATS)	
	= ‘TRANSITORY’		The program reads the model for the theoretical transitory component (SEATS)	
	= ‘AGGREGATE’		The program reads the model for the (linearized) aggregate series (TRAMO)	

2.3.7 Routine use on Many Series: the $ITER$ Parameter

The program contains a facility to handle in a single file many series, perhaps with different models each. This is controlled by the parameter $ITER$, which can take the following values

Parameter			Meaning	Default
$ITER$	=	0	One series, one model (the usual input file)	0
	=	1	several models are provided by the user to be applied by the program to the same series	
	=	2	Several series are to be treated by the program with the same model	
	=	3	Several series, each with its own specified model, are treated by the program	

When $ITER = 1$ the names of the output file are $MODEL1.OUT, \ldots,$ $MODELn.OUT$, where n is the number of models (if $ITER = 2, 3$ the usual *seriename.OUT*).

The structure of the input file *serie* if $ITER > 0$ is the following:

$ITER$ = 1 You have to append (in any format) at the end of the usual input file the number of namelists $INPUT$ (and eventually namelists REG) that you want; remember that the structure of the namelists $INPUT$ and REG is the following:
$ INPUT *parameter–name=parameter–value, . . .,*
parameter–name=parameter–value, $
$ REG *parameter–name=parameter–value, . . .,*
parameter–name=parameter–value, $

$ITER$ = 2 You have to append (in any format) at the end of the usual input file the number of series that you want, according to the following convention:

1st line:	title
2nd line:	number of observations
	starting year
	starting period
	Nfreq
3rd to n^{th} line:	observations in any format

$ITER$	=	3	You have to append (in any format) at the end of the usual input file the number of pairs series–namelist that you want; the convention for the series is the same as in $ITER = 2$, and for the namelist the same as in the case $ITER = 1$

When $ITER \neq 0$, TRACE sends the output file to the subdirectory OUTPUT.

2.3.8 Others

Parameter		Meaning	Default
GRAPH	= 0	Parameter to control the generation of data and MATLAB and GNUPLOT code for plots	0
MFREC	= 0	Parameter to control table format (the number of columns)	4

2.3.9 Minimum Number of Observations

The minimum number of observations depends on MQ, on the particular model, and on the options requested. By default, if m denotes the minimum number of observations,

– for $MQ \geq 12$, $m = 36$
– for $MQ < 12$, $m = \max(12, 4 \times MQ)$.

If the number of observations satisfies these minima, but is not enough for some additional option requested, the option is removed and its default value reset.

2.4 Input File and Examples

The input starts with the series to be treated, comprising no more than 600 observations, followed by one set of control parameters. The data can be in an external file, in which case only the path for this file need to be given, or can be written in free format before the control parameters.

To specify the set of control parameters for the series model, the NAMELIST facility is used, so that only those parameters which are not at their default values need to be set.

The series is set up as:

Card 1	TITLE (no more than 72 characters)
Card 2	NZ NYER NPER NFILE (free format)

If NFILE is 0, then

Card 3 — myfile (path for the file containing the data in ASCII format)

else if NFILE is 2

Card 3 — myfile (path for the file containing the data in the format used for graphs in TRAMO and SEATS)

else if NFILE is 3

Card 3 — mypath (path for the GRAPH directory for SEATS or TRAMO; the data will be automatically read by the program from the GRAPH directory. Used only with READ='Y'. Also, ITER should not be 1)

else if NFILE is 1

Card 3 et seq. — Z(I): I = 1, NZ (free format),

where NZ is the number of observations, NYER the start year, NPER the start period, and NFILE is an instruction to control the reading of the data. Z(.) is the array of observations. (The first nonblank characters of TITLE are used by the program to create two files, named ********.OUT and ********.LOG, in the subdirectory OUTPUT containing the output of the program.)

This is followed by namelist INPUT. The namelist starts with $INPUT and terminates with $. The parameters are entered separated by blanks.

Eight examples of input files for TRACE are provided. They illustrate some of the most relevant features of the program.

Note: the format '$\$INPUT\ldots\$$' can be replaced by others such as '$\&INPUT\ldots/$'.

EXAMPLE 1

```
SAUSGNP
140 1951 1 0
   SERIES\SAUSGNP
 $INPUT FILTER=1 METHOD=1
        LAM=1
        DI=2 LAMBDA=1600
        GRAPH=1 MFREC=4
 $
*** First example of the paper "The Use of Butterworth Filters for Trend and Cycle
Estimation in Economic Time Series", by G{\'o}mez (2003). The observations are read from
an external file (NFILE=0, the last number in the second line). A two--sided sine
Butterworth filter is applied to the seasonally adjusted series of logged USGNP
(quarterly), obtained using TRAMO/SEATS (FILTER=1). The Wiener-Kolmogorov filter
will be applied, without a model for the input series (METHOD=1). No logarithms are
taken (LAM=1). There is no seasonality in the series (MQ=1, default value). The
filter is the Hodrick and Prescott (1981) filter (DI=2 LAMBDA=1600). The program
will produce a MATLAB m--file to plot the gain function of the designed filter
(GRAPH=1). A format of four columns is given for the tables (MFREC=4).
```

EXAMPLE 2

```
TRDUSGNP
140 1951 1 0
   SERIES\TRDUSGNP
 $INPUT FILTER=3 METHOD=2
        LAM=1 D=2 Q=2 TH(1)=.0877 TH(2)=-.9123
        DD(1)=.1 DD(2)=.1 XP=.0625 XP2=.3 XS=.4
        GRAPH=1 MFREC=4
 $
*** Second example of the paper "The Use of Butterworth Filters for Trend and Cycle
Estimation in Economic Time Series", by G{\'o}mez (2003). The observations are read from
an external file (NFILE=0, the last number in the second line). A band-pass filter
is applied to the trend-cycle series of logged USGNP (quarterly), obtained using
TRAMO/SEATS (FILTER=3). The Wiener-Kolmogorov filter will be applied, with a model
for the input series (METHOD=2). The filter is applied using the model (0,2,2),
where th1=.0877 and th2=-.9123, for the theoretical trend-cycle component. The model
is taken from SEATS (D=2 Q=2 TH(1)=.0877 TH(2)=-.9123). No logarithms are taken
(LAM=1). There is no seasonality in the series (MQ=1, default value). The filter is
designed to pass all oscillations with period between 6 and 32 quarters (DD(1)=.1
DD(2)=.1 XP=.0625 XP2=.3 XS=.4) The program will produce a MATLAB m-file to plot the
gain function of the designed filter (GRAPH=1). A format of four columns is given
for the tables (MFREC=4).
```

EXAMPLE 3

```
TRDAIRLN
144 1949 1 0
   SERIES\TRDAIRLN
 $INPUT FILTER=1 METHOD=2
        LAM=1 D=2 Q=2 TH(1)=.0478 TH(2)=-.9522
        DD(1)=.1 DD(2)=.01 XP=.02 XS=.05
        GRAPH=1 MFREC=12
 $
*** Third example of the paper "The Use of Butterworth Filters for Trend and Cycle
Estimation in Economic Time Series", by G{\'o}mez (2003). The observations are read from
an external file (NFILE=0, the last number in the second line). A two--sided sine
Butterworth filter is applied to the trend-cycle series of the airline passengers
series of Box and Jenkins (1970) (in logs), obtained using TRAMO/SEATS (FILTER=1).
The Wiener-Kolmogorov filter will be applied, with a model for the input series
(METHOD=2). The filter is applied using the model (0,2,2), where th1=.0478 and
th2=-.9522, for the theoretical trend-cycle component. The model is taken from SEATS
(D=2 Q=2 TH(1)=.0478 TH(2)=-.9522). No logarithms are taken (LAM=1). There is no
seasonality in the series (MQ=1, default value). The filter is designed to pass all
oscillations with period greater than 96 months (eight years) (DD(1)=.1 DD(2)=.01
XP=.02 XS=.05). The program will produce a MATLAB m-file to plot the gain function
of the designed filter (GRAPH=1). A format of twelve columns is given for the tables
(MFREC=12).

The input file for the previous example can be simplified if we let
TRACE read the SEATS output file to find the model for the theoretical
trend-cycle component. Suppose the SEATS output file is AIRLINE.OUT.
Then, the following specification would accomplish this:
TRDAIRLN
144 1949 1 0
   SERIES\TRDAIRLN
 $INPUT FILTER=1 METHOD=2
        READ='Y' PATH='C:\SEATS\OUTPUT\AIRLINE.OUT' COMP='TREND'
        DD(1)=.1 DD(2)=.01 XP=.02 XS=.05 GRAPH=1 MFREC=12 $

If we have just run SEATS, we can even let TRACE read the trend-cycle
estimated by SEATS. To this effect, we would use the specification:
TRDAIRLN
144 1949 1 3
   C:\SEATS\GRAPH
 $INPUT FILTER=1 METHOD=2
        READ='Y' PATH='C:\SEATS\OUTPUT\AIRLINE.OUT' COMP='TREND'
        DD(1)=.1 DD(2)=.01 XP=.02 XS=.05 GRAPH=1 MFREC=12 $
```

EXAMPLE 4

```
TRDAIRLN
144 1949 1 0
   SERIES\TRDAIRLN
 $INPUT FILTER=3 METHOD=2
        LAM=1 D=2 Q=2 TH(1)=.0478 TH(2)=-.9522
        DD(1)=.1 DD(2)=.01 XP=.02 XP2=.08 XS=.15
        GRAPH=1 MFREC=12
 $
*** Fourth example of the paper "The Use of Butterworth Filters for Trend and Cycle
Estimation in Economic Time Series", by G{\'o}mez (2003). The observations are read from
an external file (NFILE=0, the last number in the second line). A band-pass filter
is applied to the trend-cycle series of the airline passengers series of Box and
Jenkins (1970) (in logs), obtained using TRAMO/SEATS (FILTER=3). The
Wiener-Kolmogorov filter will be applied, with a model for the input series
(METHOD=2). The filter is applied using the model (0,2,2), where th1=.0478 and
th2=-.9522, for the theoretical trend-cycle component. The model is taken from SEATS
(D=2 Q=2 TH(1)=.0478 TH(2)=-.9522). No logarithms are taken (LAM=1). There is no
seasonality in the series (MQ=1, default value). The filter is designed to pass all
oscillations with period between 18 and 96 months. (DD(1)=.1 DD(2)=.01 XP=.02
XP2=.08 XS=.15) The program will produce a MATLAB m-file to plot the gain function
of the designed filter (GRAPH=1). A format of twelve columns is given for the tables
(MFREC=12).
```

EXAMPLE 5

```
AIRLINE
15 1 1 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 $INPUT FILTER=4 METHOD=0
        T1=-.5 TS=-.7 MS=12
        GRAPH=1
 $
***
This is an example of an airline filter design
The observations are read from the input file (NFILE=1, the last number
in the second line).
Note that there are only 15 observations in the input series, all equal
to zero. This is because no filter is applied (METHOD=0).
The filter is an airline filter (FILTER=4).
The airline model is a monthly one with parameters -.5 and -.7 (T1=-.5
TS=-.7 MS=12).
The program will produce a MATLAB m-file to plot the gain function of
the designed filter (GRAPH=1).
```

EXAMPLE 6

```
AIRLINE     LINEAS AEREAS DE BOX-JENKINS
144   1949 1 1
112 118 132 129 121 135 148 148 136 119 104 118
115 126 141 135 125 149 170 170 158 133 114 140
145 150 178 163 172 178 199 199 184 162 146 166
171 180 193 181 183 218 230 242 209 191 172 194
196 196 236 235 229 243 264 272 237 211 180 201
204 188 235 227 234 264 302 293 259 229 203 229
242 233 267 269 270 315 364 347 312 274 237 278
284 277 317 313 318 374 413 405 355 306 271 306
315 301 356 348 355 422 465 467 404 347 305 336
340 318 362 348 363 435 491 505 404 359 310 337
360 342 406 396 420 472 548 559 463 407 362 405
417 391 419 461 472 535 622 606 508 461 390 432
 $INPUT FILTER=4 METHOD=2
  MQ=12 TH(1)=-.4 BTH(1)=-.6
  T1=.985 TS=-.6 MS=12
  GRAPH=1
  $
***
This is an example of an airline filter, designed as a seasonal
adjustment filter (T1=.985 TS=-.6 MS=12).
The observations are read from the input file (NFILE=1, the last number
in the second line).
The filter is an airline filter (FILTER=4).
The Wiener-Kolmogorov filter will be applied, with a model for the
input series (METHOD=2).
The filter is applied using the airline model (0,1,1)(0,1,1), where
th1=-.4 and th12=-.6, for the input series (TH(1)=-.4 BTH(1)=-.6).
Logarithms are taken (LAM=0, default value).
The program will produce a MATLAB m-file to plot the gain function of
the designed filter (GRAPH=1).
```

EXAMPLE 7

```
AIRLINE     LINEAS AEREAS DE BOX-JENKINS
144   1949 1 3
  C:\TRAMO\GRAPH
 $INPUT FILTER=5 METHOD=2
  READ='Y' PATH='C:\TRAMO\OUTPUT\AIRLINE.OUT' COMP='AGGREGATE'
  T1=-.4 TS=-.6 MS=12
  DD(1)=.1 DD(2)=.01 XP=.02 XS=.05
  GRAPH=1
  $
***
This is an example of a product of two filters (FILTER=5), applied to
the airline passengers series of Box and Jenkins (1970). The first
filter is an airline filter, which is the trend-cycle filter
corresponding to the estimated model (T1=-.4 TS=-.6 MS=12). The second
```

filter is a two--sided sine Butterworth filter, designed to pass all oscillations with period greater than 96 months (eight years) (DD(1)=.1 DD(2)=.01 XP=.02 XS=.05).
The observations are read from the GRAPH directory (NFILE=3, the last number in the second line, and READ='Y' in the namelist INPUT). Note that only the GRAPH directory is specified (C:\TRAMO\GRAPH, third line), but that the GRAPH directory must contain the files corresponding to the airline series.
The Wiener-Kolmogorov filter will be applied, with a model for the input series (METHOD=2).
The filter will be applied using the fitted airline model (0,1,1)(0,1,1) to the logged series, where th1=-.4 and th12=-.6. This information is contained in the TRAMO output file (PATH='C:\TRAMO\OUTPUT\AIRLINE.OUT'), which is read by TRACE (READ='Y'). The parameter COMP='AGGREGATE' tells TRACE to read the model for the aggregate series in the TRAMO output file.
The program will produce a MATLAB m-file to plot the gain function of the designed filter (GRAPH=1).

EXAMPLE 8

```
GERMANUN  GERMAN UNEMPLOYMENT SERIES
116 1965 1 0
   SERIES\GERMANUN
 $INPUT  FILTER=7  METHOD=2
   MQ=4 LAM=1 P=1 Q=0 PHI(1)=-.52311 BTH(1)=-.38534
   T1=.985 TS=-.4 MS=4
   DD(1)=.1 DD(2)=.1 XP=.0625 XP2=.3 XS=.4
   GRAPH=1 ITER=3 $
TRDUSGNP
140 1965 1 0
   SERIES\TRDUSGNP
 $INPUT  FILTER=1  METHOD=1
   MQ=4 LAM=1
   XC=.1 DI=2
   GRAPH=1 ITER=3 $
AIRLINE
144 1965 1 0
   SERIES\AIRLINE
 $INPUT FILTER=6 METHOD=2
  MQ=12 LAM=0 TH(1)=-.4 BTH(1)=-.6
  XC=.125 DI=4
  T1=-.4 TS=-.6 MS=12
  GRAPH=1 ITER=3  $
***
```

This is an example of an ITER file.
The last parameter (NFILE) in the second line tells the program whether the series is going to be read from an external file (0,2,3) or from the input file (1). In all the three series of this input file the series are read from external files which are in the SERIES directory.

1) The first series is the German unemployment series (quarterly). A product of two filters is applied (FILTER=7).
The first filter is an airline filter with parameters theta1=.985 and theta12=-.4, which is intended to perform like a seasonal adjustment filter (T1=.985 and TS=-.4; the parameter MS is used to specify the kind of airline model, MS=4 for quarterly, etc.).
The second filter is a band-pass filter, which is specified by means of the parameters DD(1)=.1 (tolerance in the pass-band), DD(2)=.1 (tolerance in the stop band), XP (first frequency for the pass-band), XP2 (second frequency for the pass-band) and XS (frequency for the stop-band).
The product filter is applied with a model for the input series (METHOD=2), obtained with the automatic model identification facility of TRAMO. The model is (1,1,0)(0,1,1), and is specified using the parameters MQ=4, P=1, Q=0, PHI(1) and BTH(1), like in TRAMO and SEATS.
The parameter GRAPH=1 tells the program to create matlab (windows version) programs to plot the gain of the different filters.
The parameter ITER=3 is used to process several files, like in TRAMO and SEATS.

2) The second series is the trend-cycle of the US GNP series (quarterly), estimated with TRAMO/SEATS. A two--sided Butterworth filter, sine version (FILTER=1) is applied. The filter is specified by means of the parameters XC (frequency where the gain is 1/2) and DI (degree of differencing in the time domain or one half of the exponent in the frequency domain).

3) The third series is the airline passengers series of Box and Jenkins. A product of two filters is applied (FILTER=6).
The first filter is an airline filter with parameters theta1=-.4 and theta12=-.6 (T1=-.4 and TS=-.6).
The second filter is a two--sided Butterworth filter, tangent version, which is specified by means of the parameters XC and DI.
The product filter is applied with a model for the input series (METHOD=2).

2.5 Identification of the data and GNUPLOT files Produced by TRACE

Description of the Files in the GNUPLOT directory that contain the data and GNUPLOT code to plot the squared gain functions.

Meaning	Name of File
GNUPLOT example file to plot three series. By changing this file appropriately the user can plot different series with the GNUPLOT program.	*gseries.gnp*
GNUPLOT file generated by TRACE to plot the squared gain function of an airline trend–cycle filter	*galfd.gnp*
Data file generated by TRACE to plot the squared gain function of an airline trend–cycle filter	*galfd.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a two–sided sine Butterworth filter	*gbfsind.gnp*
Data file generated by TRACE to plot the squared gain function of a two–sided sine Butterworth filter	*gbfsind.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a two–sided tangent Butterworth filter	*gbftand.gnp*
Data file generated by TRACE to plot the squared gain function of a two–sided tangent Butterworth filter	*gbftand.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a band–pass filter based on a tangent Butterworth filter	*gbftbpd.gnp*
Data file generated by TRACE to plot the squared gain function of a band–pass filter based on a tangent Butterworth filter	*gbftbpd.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided sine Butterworth filter	*gpalsind.gnp*
Data file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided sine Butterworth filter	*gpalsind.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided tangent Butterworth filter	*gpaltand.gnp*

Meaning	Name of File
Data file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided tangent Butterworth filter	*gpaltand.dat*
GENUPLOT file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a band–pass filter based on a tangent Butterworth filter	*gpaltbpd.gnp*
Data file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a band–pass filter based on a tangent Butterworth filter	*gpaltbpd.dat*

2.6 Identification of the MATLAB M–files Produced by TRACE

Description of the Files in the MATLAB directory that contain the MATLAB code to plot the gain functions.

Meaning	Name of File
MATLAB function to obtain the polynomials in the numerator and denominator of a trend–cycle filter of an airline model. It is needed for the m–files generated by TRACE	*ctla.m*
MATLAB function to obtain the frequency response function of an ARMA filter. It is needed for the m–files generated by TRACE	*freqsd.m*
MATLAB m–file generated by TRACE to plot the squared gain function of an airline trend–cycle filter	*galfw.m*
MATLAB m–file generated by TRACE to plot the squared gain function of a two–sided sine Butterworth filter	*gbfsinw.m*
MATLAB m–file generated by TRACE to plot the squared gain function of a band–pass filter based on a tangent Butterworth filter	*gbftbpw.m*
MATLAB m–file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided sine Butterworth filter	*gpalsinw.m*

Meaning	Name of File
MATLAB m–file generated by TRACE to plot the squared gain function of a two–sided tangent Butterworth filter	*gbftanw.m*
MATLAB m–file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a two–sided tangent Butterworth filter	*gpaltanw.m*
MATLAB m–file generated by TRACE to plot the squared gain function of a filter which is the product of an airline trend–cycle filter and a band–pass filter based on a tangent Butterworth filter	*gpaltbpw.m*

3 References

BELL, W., (1984), "Signal Extraction for Nonstationary Series", *The Annals of Statistics*, **12**, 646-664.

BELL, W.R. and HILLMER, S.C. (1984), "Issues Involved with the Seasonal Adjustment of Economic Time Series", *Journal of Business and Economic Statistics* 2, 291–320.

BOX, G.E.P., HILLMER, S.C. and TIAO, G.C. (1978), "Analysis and Modeling of Seasonal Time Series", in Zellner, A. (ed.), *Seasonal Analysis of Economic Time Series*, Washington, D.C.: U.S. Dept. of Commerce — Bureau of the Census, 309–334.

BOX, G.E.P. and JENKINS, G.M. (1970), *Time Series Analysis: Forecasting and Control*, San Francisco: Holden–Day.

BURMAN, J.P. (1980), "Seasonal Adjustment by Signal Extraction", *Journal of the Royal Statistical Society* A, 143, 321–337.

CLEVELAND, W.P. and TIAO, G.C. (1976), "Decomposition of Seasonal Time Series: A Model for the X–11 Program", *Journal of the American Statistical Association* 71, 581–587.

DAGUM, E. B., (1980), "The X11 ARIMA Seasonal Adjustment Method", Statistics Canada, Catalogue 12–564E.

GÓMEZ, V. (2001), "The Use of Butterworth Filters for Trend and Cycle Estimatin in Economic Time Series", *Journal of Business and Economic Statistics*, 19, 365–373.

GÓMEZ, V. (1999), "Three Equivalent Methods for Filtering Finite Nonstationary Time Series", *Journal of Business and Economic Statistics*, 17, 109–116.

GÓMEZ, V., and BENGOECHEA, P., (2000), "The Quarterly National Accounts Trend–Cycle Filter Versus Model–Based Filters", *Spanish Economic Review*, 2, 29–48.

GÓMEZ, V., and MARAVALL, A., (1997), "Programs TRAMO and SEATS", Instructions for the User (Beta Version: June 1997), Working Paper N. 97001, Dirección General de Análisis y P.P., Ministry of Finance.

HILLMER, S.C. and TIAO, G.C. (1982), "An ARIMA–Model Based Approach to Seasonal Adjustment", *Journal of the American Statistical Association* 77, 63–70.

HODRICK, R. J. and PRESCOTT, E. C. (1980), "Postwar U. S. business cycles: An empirical investigation", Discussion Paper No. 451, Carnegie–Mellon University.

MARAVALL, A. and PIERCE, D.A. (1987), "A Prototypical Seasonal Adjustment Model", *Journal of Time Series Analysis* 8, 177–193.

MARAVALL, A. (1995), "Unobserved Components in Economic Time Series", in Pesaran, H., Schmidt, P. and Wickens, M. (eds.), *The Handbook of Applied Econometrics*, vol. 1, Oxford: Basil Blackwell.

Oppenheim, A. V. and Schaffer, R. W., (1989), *Discrete–Time Signal Processing*, Prentice Hall, New Jersey.

Otnes, R. K. and Enochson, L., (1978), *Applied Time Series Analysis*, Vol. 1, Wiley, New York.

4 Classification of Input Parameters by Function

Filter Type

Filtering Method

Low–pass Filter Design

Band–pass Filter Design

Airline Filter Design

ARIMA Model for the Input Series

Others

Index