TSTMATLAB, a Set of MATLAB Programs to run TRAMO/SEATS/TRACE

Víctor Gómez
Ministerio de Economía y Hacienda
Dirección Gral. de Presupuestos
Subdirección Gral. de Análisis y P.E.
Alberto Alcocer 2, 1-P, D-34
28046, Madrid, SPAIN.
(*VGomez@sepg.minhap.es*)

14-11-2011

Abstract

The programs Tramo, Seats (Gómez and Maravall, 1997) and Trace (Gómez, 2003) are written in Fortran and can be used for time series analysis. Tramo is a program for estimation and forecasting of regression models with possibly nonstationary (Arima) errors and any sequence of missing values. Seats is a program for estimation of unobserved components in time series following the so–called Arima–model–based method. Finally, Trace is a program for the estimation of trends, cycles and even seasonally adjusted series using fixed filters that can be integrated in a model based procedure such as that of Tramo/Seats. The purpose of this manual is to explain how to run the three previous programs within the Matlab environment. This is achieved using a set of Matlab functions that pass the input information to the input files, run the Fortran programs in the operating system environment, and read the output information from the output files generated by the programs. In this way, the user is liberated from the burden of preparing the input files and looking for the appropriate files to extract the results. In addition, the user can benefit from other Matlab functions to analyze the results, plot the different series, etc.

Contents

1 Introduction

The motivation for TSTMATLAB is to allow for the integration of the programs Tramo, Seats and Trace in a unique environment, thus facilitating its use. Up to now, if the user wanted to estimate a business cycle with Trace using a model–based procedure, he should run first the DOS version of Tramo and Seats and then run the DOS version of TRACE. This required the preparation of certain input files containing the data and several input instructions that were given using the old namelist Fortran facility. This is clearly a cumbersome procedure that deters many users from using the previous programs.

It is to be emphasized that it is the three programs, Tramo, Seats and Trace, and not just the first two, that are integrated in TSTMATLAB, thus making TSTMATLAB an efficient tool to estimate trends and cycles within a model–based procedure, as explained in the Trace manual. However, TSTMATLAB can also be used with Tramo and Seats only to perform short term analysis of time series.

The purpose of this manual is to explain how to run the programs Tramo, Seats and Trace within the Matlab environment. The solution adopted has been to write some Matlab functions that pass the input information to the input files, run the programs in the operating system environment, and read the output information from the output files generated by the programs. Thus, the programs remain unaltered and they can still be executed in the usual manner described in their manuals. Since the programs are written in Fortran and are really very fast, there is no appreciable loss of efficiency in running them from the Matlab environment in this way. However, the user is liberated from the burden of going into the manuals in detail, preparing the input files, and looking for the appropriate files to extract the results. In addition, the user can benefit from other Matlab functions to analyze the results, plot the different series, etc.

The programs Tramo, Seats and Trace, as well as the Matlab functions described in this manual should be unzipped into a directory. For example,

C:\tst

It is not necessary to run the programs from this directory, although it is simpler to do so. **It is recommended to add its path, as well as the path to its subfolders, to the Matlab paths.**

Manuals for the programas Tramo, Seats and Trace, as well as this document, can be found in the subdirectories TSMAN, TRACEMAN and TSTMATLABMAN, respectively.

2 Running Tramo/Seats (TS) in MATLAB

The Matlab function that can be used to run TS is

out = runtsmatlab(fname,path)

where the input parameters are

fname: a string containing the spec for Tramo/Seats, located in the SPECTRM subdirectory (to be described later)

path: a string containing the path where TSTMATLAB is located (for example, 'C:\tst'). This input parameter is not input if the program is run from the directory where TSTMATLAB is located.

The output parameter is

out: a structure containing the output of Tramo/Seats (to be described later)

Example:

out = runtsmatlab('ipiale_trm')

In this case the program is run from the directory where TSTMATLAB is located. If it is run from another directory, then it should be something like

out = runtsmatlab('ipiale_trm','C:\tst'),

where **'C:\tst'** is the directory where the program is located. In both cases, the string **'ipiale_trm'** is the name of a Matlab function, located in the SPECTRM subdirectory, that returns a cell–array containing the specifications for the TS programs. These functions will be described in the next section.

During the execution of the **runtsmatlab** function, MATLAB changes the current directory first to the directory where the TS programs are located and then to the SPECTRM subdirectory. Once the spec function is executed, MATLAB changes the directory again to where the TS programs are located and executes the TS programs.

The output from TS will be written in the subdirectory OUTPUT of the directory where the TS programs are located. The output from TRAMO has extension **OUT** and the output from SEATS has extension **STS**. For example, for the series **ipiale**, the outputs from TS will be **ipiale.out** and **ipiale.sts**, both located in the directory OUTPUT.

3 Input Specs for TS

The specifications for TS are located in the subdirectory SPECTRM of the directory where TSTMATLAB has been installed. These specifications are Matlab functions that generate a cell–array containing the input instructions for the TS programs. A typical input spec for TS is as follows.

```
function c = gu11na_trm
%
%function to create the cell array necessary to run TRAMO/SEATS in
%MATLAB.
%The input for TRAMO is a cell array c of size [n m]. n is the number of
%series and m is the number of fields that contain the information for
%each series
%c{i,1} contains the title of the series
%c{i,2} contains the number of observations, the initial year and the
%       initial period
%c{i,3} contains the series
%c{i,4} an optional field. It is a cell array, d, that contains the
%       namelist input and optional additional namelists reg and
%       instructions for these last namelists.
%       namelists reg, if any, and the corresponding data should be written
%       in pairs.

%series title
title='gu11na';
%initial year, initial period
inyinp='1980 1';
%data
ppath=cd;      %Current path. For example, C:\tst\spectrm
%Now, obtain path for the programs. For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrm'); ppath=ppath(1:x-1);
y=load([ppath '\series\' title '.dat']); [ny,my]=size(y);
%alternatively
% y=load('c:\tst\series\gu11na.dat'); [ny,my]=size(y);

%number of observations
nobs=num2str(ny);

c{1,1}=title;
c{1,2}=[nobs ' ' inyinp];
c{1,3}=y;

%instructions for TRAMO/SEATS
%namelist input
input='p=2 ireg=14 seats=2';
input=[input ' rmod=0 epsphi=180'];
d{1,1}=input;
```

```
%namelists reg.
d{1,2}='iuser=-1 nser=6 ilong=240';
%path for the deterministic variable
d{1,3}='series\wk6.g11';
d{1,4}='iuser=-1 nser=6 ilong=240';
d{1,5}='series\wkdec6.g11';
d{1,6}='iuser=-1 nser=1 ilong=240';
d{1,7}='series\lp.g11';
d{1,8}='iuser=-1 nser=1 ilong=228';
d{1,9}='series\school.g11';
c{1,4}=d;
```

The function should contain the appropriate instructions to create a cell–array that will be passed to the function **runtsmatlab**, previously described, or **runtstmatlab**, to be described later. The cell–array, c, has size $[n, 4]$, where n is the number of series to be treated and 4 is the number of fields containing the instructions for each series. The fields for each series are as follows.

c{i,1} a string containing the series title.

c{i,2} a string containing the number of observations, the initial year and the initial period. The three numbers should be written in this order.

c{i,3} a numerical vector containing the i–th series.

c{i,4} a cell–array of size $[1, p]$, d, where $d\{1, 1\}$ contains the input parameters for TS (the contents of namelist input in the TS manual) and the other fields, if any, contain the instructions for the regression parameters. These instructions should be given in pairs, one field for the parameters (the contents of a namelist reg in the TS manual) and another field for the actual data, or a path to the data.

It is to be noticed that sometimes the field $c\{i, 4\}$ can be empty. This may occur while handling many series if the user wants to give the same input instructions to all of them. In this case, it is possible to give the common instructions to the first series only and to set all other series instructions empty. This procedure is controlled by the parameter ITER, described in the TS manual.

An example of a spec file in which several series are input to TS with common instructions is the following.

```
function c = oipitot_trm
%
%function to create the cell array necessary to run TRAMO/SEATS in
%MATLAB.
```

```
%The input for TRAMO is a cell array c of size [n m]. n is the number of
%series and m is the number of fields that contain the information for
%each series
%c{i,1} contains the title of the series
%c{i,2} contains the number of observations, the initial year and the
%       initial period
%c{i,3} contains the series
%c{i,4} an optional field. It is a cell array, d, that contains the
%       namelist input and optional additional namelists reg and
%       instructions for these last namelists.
%       namelists reg, if any, and the corresponding data should be written
%       in pairs.

%all strings in the list should have the same length
list=['OIPIGUM'
      'OIPIGAT'
      'OIPIGBE'
      'OIPIGDE'
      'OIPIGES'
      'OIPIGFI'
      'OIPIGFR'
      'OIPIGGR'
      'OIPIGIE'
      'OIPIGIT'
      'OIPIGLU'
      'OIPIGNL'
      'OIPIGPT'
      ];
[n,m]=size(list);

%data directory. For example, C:\tst\series
ppath=cd;      %Current path. For example, C:\tst\spectrm
%Now, obtain path for the programs. For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrm'); ppath=ppath(1:x-1);
%alternatively
% ppath='C:\tst';

for i=1:n
 %series title
 title=list(i,:);
 %initial year, initial period
```

```
 inyinp='1985 1';
 %read data
 y=load([ppath '\series\' title '.dat']);  [ny,my]=size(y);
 nobs=num2str(ny); nnobs=length(nobs);     %string
 if nnobs < 5
  nobs=[nobs blanks(5-nnobs)];
 end
 c{i,1}=title;
 c{i,2}=[nobs ' ' inyinp];
 c{i,3}=y;
 %instructions for TRAMO/SEATS
 input='lam=-1 idif=3 inic=3 itrad=-1 ieast=-1 iatip=1 seats=2 iter=3';
 input=[input ' rmod=0 epsphi=180'];
 d{1,1}=input;                                  %namelist input
 c{i,4}=d;
end
```

In the following spec file example, several series are read from an EXCEL spreadsheet. The names of the series, as well as the data, are read from the spreadsheet. The data are quarterly and correspond to the official unemployment survey (EPA) carried out in the seventeen regions of Spain during the period 1995–I to 2010–III. The instructions for TS include four dummy variables intended to measure the effect of the E plan, a public spending plan implemented by the Spanish government in the year 2009 to fight against unemployment. To run TS with this example, the MATLAB expression

out = runtsmatlab('epaccaaagr_trm'),

can be executed at the MATLAB prompt in the directory where the TS programs are located. After running TS, the original and the trend–cycle series corresponding to each series can be graphed using the **seeorigtr** script file, to be described later. The title for each series is read from the structure **out** given as output by TS. The spec file is as follows.

```
function c = epaccaaagr_trm
%
%function to create the cell array necessary to run TRAMO/SEATS in
%MATLAB.
%The input for TRAMO is a cell array c of size [n m]. n is the number of
%series and m is the number of fields that contain the information for
%each series
%c{i,1} contains the title of the series
%c{i,2} contains the number of observations, the initial year and the
```

```
%       initial period
%c{i,3} contains the series
%c{i,4} an optional field. It is a cell array, d, that contains the
%       namelist input and optional additional namelists reg and
%       instructions for these last namelists.
%       namelists reg, if any, and the corresponding data should be written
%       in pairs.

c=[];
%read data from excel spreadsheet. The first column contains the dates.
hoja='agr';                        %agr, con, ind, ser
[data,header]=xlsread('c:\tst\series\epaccaa',hoja);



%all strings in the list should have the same length
%header(1,:) contains the names of the series. The first name is a blank.
list=strvcat(header(1,2:end));
[n,m]=size(list);


%data directory. For example, C:\tst\series
ppath=cd;      %Current path. For example, C:\tst\spectrm
%Now, obtain path for the programs. For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrm'); ppath=ppath(1:x-1);
%alternatively
% ppath='C:\tst';

ytotal=zeros(size(data(:,1)));

for i=1:n
 %series title
 title=list(i,:);
 %initial year, initial period
 inyinp='1995 1';
 %read data
 y=data(:,i);
 ytotal=ytotal+y;
 [ny,my]=size(y);
 nobs=num2str(ny); nnobs=length(nobs);     %string
 if nnobs < 5
```

```
  nobs=[nobs blanks(5-nnobs)];
 end
 c{i,1}=title;
 c{i,2}=[nobs ' ' inyinp];
 c{i,3}=y;
 %instructions for TRAMO/SEATS
 input='lam=-1 idif=3 inic=3 itrad=-1 ieast=-1 iatip=1 seats=2 iter=3';
 input=[input ' aio=1 mq=4 rmod=0 epsphi=180 ireg=4 '];
 d{1,1}=input;                                  %namelist input
 d{1,2}='iuser=2 nser=4 ';                      %variables para plan e
 d{1,3}='57 ao 58 ao 59 ao 60 ao ';
 c{i,4}=d;
end
%total
c{n+1,1}='Total';
c{n+1,2}=[nobs ' ' inyinp];
c{n+1,3}=ytotal;
input='lam=-1 idif=3 inic=3 itrad=-1 ieast=-1 iatip=1 seats=2 iter=3';
input=[input ' aio=1 mq=4 rmod=0 epsphi=180 ireg=4 '];
d{1,1}=input;                                  %namelist input
d{1,2}='iuser=2 nser=4 ';                      %variables para plan e
d{1,3}='57 ao 58 ao 59 ao 60 ao ';
c{n+1,4}=d;
```

The following is an example of a spec file in which the instructions for each series are different.

```
function c = examp20_trm
%
%function to create the cell array necessary to run TRAMO/SEATS in
%MATLAB.
%The input for TRAMO is a cell array c of size [n m]. n is the number of
%series and m is the number of fields that contain the information for
%each series
%c{i,1} contains the title of the series
%c{i,2} contains the number of observations, the initial year and the
%       initial period
%c{i,3} contains the series
%c{i,4} an optional field. It is a cell array, d, that contains the
%       namelist input and optional additional namelists reg and
%       instructions for these last namelists.
%       namelists reg, if any, and the corresponding data should be written
```

```
%       in pairs.


%all strings in the lists should have the same length
list1=['CAR-REGR'
       'QUAT-CAR'
      ];
[n,m]=size(list1);
list2=['1971 1'
       '1971 1'
       ];
list3=['rsa=3 iter=3'
       'mq=4 rsa=3  '
      ];

%data directory. For example, C:\tst\series
ppath=cd;      %Current path. For example, C:\tst\spectrm
%Now, obtain path for the programs. For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrm'); ppath=ppath(1:x-1);

for i=1:n
 %series title
 title=list1(i,:);
 c{i,1}=title;
 %initial year, initial period
 inyinp=list2(i,:);
 %read data
 y=load([ppath '\series\' title '.dat']);
 %transform table into column
 y=y'; y=vec(y);
% number of observations
 [ny,my]=size(y);                           %number
 nobs=num2str(ny); nnobs=length(nobs);     %string
 if nnobs < 5
  nobs=[nobs blanks(5-nnobs)];
 end
 c{i,2}=[nobs ' ' inyinp];
 c{i,3}=y;
 %instructions for TRAMO/SEATS
 input=list3(i,:);
 d{1,1}=input;                                    %namelist input
```

```
 c{i,4}=d;
end
```

4 Output Structure Returned by the Function runtsmatlab

If only one series has been input to TS, **runtsmatlab** returns a structure, **out**, with the following fields.

title a string containing the title

nziyip a $[1, 3]$ numerical array containing the number of observations, the initial year and the initial period

orig a numerical array containing the original series

model a structure containing the model information

series a structure containing the series given as output by TS in the GRAPH\SERIES directory and described in the TS manual

regoutse a structure containing the series given as output by TS in the GRAPH\REGOUTSE directory and described in the TS manual

forecast a structure containing the series given as output by TS in the GRAPH\FORECAST directory and described in the TS manual

acf a structure containing the series given as output by TS in the GRAPH\ACF directory and described in the TS manual

spectra a structure containing the series given as output by TS in the GRAPH\SPECTRA directory and described in the TS manual

filters a structure containing the series given as output by TS in the GRAPH\FILTERS directory and described in the TS manual

If more than one series have been input to TS, **runtsmatlab** returns a multiple structure, **out**, such that, for the i–th series, **out(i)** contains the following fields.

title a string containing the title

nziyip a $[1, 3]$ numerical array containing the number of observations, the initial year and the initial period

orig a numerical array containing the original series

model a structure containing the model information

lin the linearized series given as output by TS in the GRAPH\SERIES directory and described in the TS manual

tre the final trend–cycle series given as output by TS in the GRAPH\SERIES directory and described in the TS manual

sa the final seasonally adjusted series given as output by TS in the GRAPH\SERIES directory and described in the TS manual

fx the forecasts of the original series given as output by TS in the GRAPH\FORECAST directory and described in the TS manual

ftr the forecasts of the final trend–cycle series given as output by TS in the GRAPH\FORECAST directory and described in the TS manual

fsa the forecasts of the final seasonally adjusted series given as output by TS in the GRAPH\FORECAST directory and described in the TS manual

ftx the forecasts of the original series and their mse given as output by TS in the GRAPH\FORECAST directory and described in the TS manual

5 Description of the residcp function

The **residcp** function computes, for the residuals given in the output structure **out** of **runtsmatlab** or **runtstmatlab**, to be described later, the standard error, simple and partial autocorrelations and many other quantities that can be used to assess the goodness of fit of a certain ARIMA model. The general form of this function is as follows

infr = residcp(out,lag,cw,pflag)

where the input parameters are

out: an output structure given by **runtsmatlab** or **runtstmatlab**

lag: the number of lags desired for the simple and partial autocorrelations

cw: a number used to construct confidence bands, for example 1.96

pflag: 1 if graphs are desired, 0 if not

The output parameter is

infr: a structure containing the residuals and many statistics related with them. Among others, simple and partial autocorrelations of the residuals and the squared residuals, Ljung–Box statistics and their p–values, normality tests and p–value, Durbin–Watson, etc.

Function **residcp** cannot be used when more than one series has been handled by TRAMO.

6 Description of the pforcst function

The **pforcst** function plots the forecasts of the original series given in the files of the GRAPH\FORECAST subdirectory and described in the TS manual. If there is seasonality in the series, graphs of the growth rates are also produced. The general form of this function is as follows

pforcst(out,cw)

where the input parameters are

out: an output structure given by **runtsmatlab** or **runtstmatlab**

cw: a number used to construct confidence bands, for example 1.96

If several TS has been run so that several series have been handled, the appropriate call for the i–th series is

pforcst(out(i),cw)

This function has no output parameters.

7 Description of the tsplot function

The **tsplot** function can plot several series using a calendar structure. This calendar structure can be created from the output structure of **runtsmatlab** or **runtstmatlab** using the function **makedatei**. Thus, the following instruction creates the calendar structure datei.

datei = makedatei(out),

where out is the structure created by **runtsmatlab** or **runtstmatlab**. The fields of this calendar structure are: **beg_yr**, the initial year, **beg_per**, the initial period and **freq**, the frequency.

The general form of function **tsplot** is as follows

tsplot(yy,datei,vnames)

where the input parameters are

yy: a matrix whose columns contain the series to be plotted.

datei: a calendar structure

vnames: a string array containing the names of the series

For example, the sequence of Matlab commands to create a graph with the original series and the trend–cycle estimated with TS could be the following.

```
>> datei=makedatei(out);
>> vnames=strvcat('Original series','Trend-cycle');
>> tsplot([out.orig out.series.trendo],datei,vnames)
```

The **tsplot** function has no output parameters. This function, as well as some other Matlab functions to manipulate calendar structures are taken from James Lesage's Econometric Toolbox, available in the public domain.

8 Description of the seeorigtr script file

The **seeorigtr** script file can be used after running TS with several series, when one would like to see graphs of both, the original and the trend–cycle series, for each series. The MATLAB programming lines contained in the script file **seeorigtr** are the following.

```
[n,m]=size(out);
datei=makedatei(out(1));
vnames=strvcat('Original series','Trend-cycle');

i1=1;
i2=m;

for i=i1:i2
 tsplot([out(i).orig out(i).tre],datei,vnames); title(out(i).title)
 pause
end

closefig
```

The user can manipulate the previous lines if he wants to change the legends, for example. Thus, instead of "Original series" and "Trend–cycle" he can use "Original" and "Trend". Also, the variables $i1$ and $i2$ can be used to set the limits of the range of variables that the user can graph. Thus, he can set $i1 = 3$ and $i2 = 3$ to graph a single series, namely the series number three. As it stands, the user will graph all available series in the structure **out**. Finally, the **closefig** function will close all MATLAB figures open at the time once the script file has finished execution.

9 Running TRACE in MATLAB

The Matlab function that can be used to run TRACE is

out = runtracematlab(fname,path)

where the input parameters are

fname: a string containing the spec for Trace, located in the SPECTRC subdirectory (to be described later)

path: a string containing the path where TSTMATLAB is located (for example, 'C:\tst'). This input parameter is not input if the program is run from the directory where TSTMATLAB is located.

The output parameter is

out: a structure containing the output of Trace (to be described later)

Example:

out = runtracematlab('ipiale_trc')

In this case the program is run from the directory where TSTMATLAB is located. If it is run from another directory, then it should be something like

out = runtracematlab('ipiale_trc','C:\tst'),

where **'C:\tst'** is the directory where the program is located. In both cases, the string **'ipiale_trc'** is the name of a Matlab function, located in the SPECTRC subdirectory, that returns a cell–array containing the instructions for the Trace program. These functions will be described in the next section.

During the execution of the **runtracematlab** function, MATLAB changes the current directory first to the directory where the Trace program is located and then to the SPECTRC subdirectory. Once the spec function is executed, MATLAB changes the directory again to where the Trace program is located and executes the Trace program.

The output from TRACE will be written in the subdirectory OUTPUT of the directory where the TRAMO, SEATS and TRACE programs are located and has extension TRC. For example, for the series **ipiale**, the output is **ipiale.trc**.

Each time **runtracematlab** is executed Trace creates a file with the name

SERIE.SPA

This file contains the paths for the files where the input and filtered series are copied. These files are located in the GNUPLOT subdirectory of the directory where TSTMATLAB is installed. They have extension DAT and the data are in two columns. The first one is the input series and the second one is the output series. When several series are treated, if the first one corresponds to the reference cycle, the list of paths contained in the SERIE.SPA file is used by the function **spectran** to perform an analysis in the frequency and time domains of the other estimated cycles with respect to the estimated reference cycle. The **spectran** function will be described later.

10 Input Specs for TRACE

The specifications for Trace are located in the subdirectory SPECTRC of the directory where TSTMATLAB has been installed. These specifications are Matlab functions that generate a cell–array containing the input instructions for the Trace program. A typical input spec for Trace is as follows.

```
function e = ipiale_trc
%
%function to create the cell array necessary to run TRACE in
%MATLAB.
%the input for TRACE is a cell array e of size [n 4]. n is the number of
%series and 4 is the number of fields that contain the information for
%each series
%e{i,1} contains the title of the series
%e{i,2} contains the number of observations, the initial year, the
%       initial period and control number, see pp. 24 and 25 of TRACE
%       manual
%e{i,3} contains the series path or the series itself
%e{i,4} contains the namelist input


name='ipitrend';                                    %series name for TRACE
st='336 1950 1 3';     %NZ NYER NPER NFILE on page 24 of the TRACE manual
```

```
ppath=cd;      %Current path. For example, C:\tst\spectrc
%Now, obtain path for the programs (ppath). For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrc'); ppath=ppath(1:x-1);
%alternatively
% ppath='C:\tst';

%output file from SEATS. In this example, TRACE will read some information
%from this file about the model followed by the trend component
otpfile='ipiale';
%the following is the path for the SEATS output. For example,
%C:\tst\output\ipiale.sts
otpfile=[otpfile '.sts'];
path=['path=''' ppath '\output\' otpfile ''''];
%alternatively
%path='path=''C:\tst\output\ipiale.sts''';

spec='dd(1)=.1 dd(2)=.01 xp=.02 xp2=.08 xs=.15'; %filter specifications
freq='12';                                       %series frequency
%do not forget to add the 'mfrec' parameter in the following
input=['filter=3 method=2 read=''y'' comp=''trend'' graph=1 mfrec=' freq];

e{1,1}=name;
e{1,2}=st;
%path for the data, according to NFILE above. In this case, the data will
%be read from the GRAPH directory because NFILE=3.
e{1,3}=[' ' ppath '\graph'];

input=[input ' ' path];
input=[input ' ' spec];
e{1,4}=input;
```

The function should contain the appropriate instructions to create a cell–array that will be passed to the function **runtracematlab** or **runtstmatlab**, to be described later. The cell–array, e, has size $[n, 4]$, where n is the number of series to be treated and 4 is the number of fields containing the instructions for each series. The fields for each series are as follows.

e{i,1} a string containing the series title.

e{i,2} a string containing the number of observations, the initial year, the initial period and a control number for the data, described in the TRACE manual on pages 24 and 25 and later in this manual for completeness. The four numbers should be written in this order.

e{i,3} depending on the control number of the previous field, a numerical vector containing the i–th series or a path to a file containing the series. This is described in the TRACE manual on pages 24 and 25 and later in this manual for completeness.

e{i,4} a string containing the input parameters for TRACE (the contents of namelist input in the TRACE manual).

The following points should be noticed.

- Sometimes the field $e\{i, 4\}$ can be empty. This may occur while handling many series if the user wants to give the same input instructions to all of them. In this case, it is possible to give the common instructions to the first series only and to set all other series instructions empty. This procedure is controlled by the parameter ITER, described in the TRACE manual.

- The input parameter MQ is used only if the filter is applied with a model, and it refers to this model. The parameter MFREC is used in the tables given by the output file. **MFREC is recommended to be always input when the input series is not annual** because by default TRACE assumes that the input series is annual.

- When the control number (the fourth number in e{i,2}) is 3, the path given in e{i,3} should be that of the GRAPH subdirectory. In this case, TRACE automatically reads the appropriate estimated component produced by SEATS to which the filter should be applied. Usually, this component is the trend (read = 'y', comp = 'trend' in the input instructions), but it can also be the original series (read = 'y', comp = 'aggregate'). If the control number is 2, TRACE reads the file in the GRAPH subdirectory whose path is given in e{i,3}. The control number 3 can only be used with a single series while 2 should be used with several series. If the control number is zero, the series can be in any file and its format should be ASCII, not the format of the files produced by TS. Finally, if the control number is one, the data are passed directly to e{i,3} and no path is given.

An example of a spec file in which several series are input to TRACE with common instructions is the following.

```
function e = oipitot_trc
%
%function to create the cell array necessary to run TRACE in
```

```
%MATLAB.
%the input for TRACE is a cell array e of size [n 4]. n is the number of
%series and 4 is the number of fields that contain the information for
%each series
%e{i,1} contains the title of the series
%e{i,2} contains the number of observations, the initial year, the
%       initial period and control number, see pp. 24 and 25 of TRACE
%       manual
%e{i,3} contains the series path or the series itself
%e{i,4} contains the namelist input

ppath=cd;      %Current path. For example, C:\tst\spectrc
%Now, obtain path for the programs (ppath). For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrc'); ppath=ppath(1:x-1);

%all strings in the list should have the same length
list=['OIPIGUM'
      'OIPIGAT'
      'OIPIGBE'
      'OIPIGDE'
      'OIPIGES'
      'OIPIGFI'
      'OIPIGFR'
      'OIPIGGR'
      'OIPIGIE'
      'OIPIGIT'
      'OIPIGLU'
      'OIPIGNL'
      'OIPIGPT'
      ];
[n,m]=size(list);
freq='12';                                              %series frequency
%do not forget to add the 'mfrec' parameter in the following
input1=['filter=3 method=2 read=''y'' comp=''trend'' graph=1 mfrec=' freq];
iter='iter=3';                                     %several series in the file
input1=[input1 ' ' iter];
spec='dd(1)=.1 dd(2)=.01 xp=.02 xp2=.08 xs=.15'; %filter specifications
st='215 1985 1 2';

for i=1:n
 e{i,1}=['C' list(i,:)];                              %series name for TRACE
```

```
 e{i,2}=st;
 e{i,3}=[' ' ppath '\graph\series\' list(i,:) '.tre']; %path for the data
 otpfile=list(i,:);                               %output file from SEATS
 otpfile=[otpfile '.sts'];
 path=['path=''' ppath '\output\' otpfile ''''];
 input=[input1 ' ' path];  input=[input ' ' spec ];
 e{i,4}=input;
end
```

The following is an example of a spec file in which the instructions for each series are different.

```
function e = examp8_trc
%
%function to create the cell array necessary to run TRACE in
%MATLAB.
%the input for TRACE is a cell array e of size [n 4]. n is the number of
%series and 4 is the number of fields that contain the information for
%each series
%e{i,1} contains the title of the series
%e{i,2} contains the number of observations, the initial year, the
%       initial period and control number, see pp. 24 and 25 of TRACE
%       manual
%e{i,3} contains the series path or the series itself
%e{i,4} contains the namelist input

ppath=cd;      %Current path. For example, C:\tst\spectrc
%Now, obtain path for the programs (ppath). For example, C:\tst
ppath=lower(ppath); x=strfind(ppath,'\spectrc'); ppath=ppath(1:x-1);
%alternatively
% ppath='C:\tst';

%all strings in the lists should have the same length
list1=['GERMANUN'
       'TRDUSGNP'
       'AIRLINE '
      ];
[n,m]=size(list1);

list2=['116 1965 1 0'
       '140 1965 1 0'
       '144 1965 1 0'
```

```
       ];
%do not forget to add the 'mfrec' parameter in the following
list31=['FILTER=7  METHOD=2 MQ=4 LAM=1 P=1 Q=0 PHI(1)=-.52311 '
        'FILTER=1  METHOD=1      LAM=1 XC=.1 DI=2 GRAPH=1      '
        'FILTER=6  METHOD=2 MQ=12 LAM=0 TH(1)=-.4 BTH(1)=-.6   '
      ];
list32=['BTH(1)=-.38534 T1=.985 TS=-.4 MS=4 DD(1)=.1 DD(2)=.1 XP=.0625 '
        'GRAPH=1 MFREC=4 ITER=3                                        '
        'XC=.125 DI=4 T1=-.4 TS=-.6 MS=12 GRAPH=1 MFREC=12 ITER=3      '
      ];
list33=['XP2=.3 XS=.4 GRAPH=1 MFREC=4 ITER=3'
        '                                   '
        '                                   '
      ];


for i=1:n
 e{i,1}=deblank(list1(i,:));                            %series name for TRACE
 e{i,2}=list2(i,:);
 e{i,3}=[' ' ppath '\series\' deblank(list1(i,:))];   %path for the data
 e{i,4}=[list31(i,:) list32(i,:) deblank(list33(i,:))]; %input instructions
end
```

Note that in the previous example we have used the Matlab function deblank to trim the trailing blanks in the names of the series. If the names do not have the same length, as in this example, and the blanks are not trimmed, there may be problems with the paths created by the function **runtracematlab** and passed on to the TS programs.

11 Output Structure Returned by the Function runtracematlab

If only one series has been input to Trace, **runtracematlab** returns a structure, **out**, with the following fields.

trace.title a string containing the title

trace.nziyip a $[1, 4]$ numerical array containing the number of observations, the initial year, the initial period and the control number.

trace.inser a numerical array containing the input series, i.e. the series to which the filter is applied.

trace.outser a numerical array containing the output series, i.e. the series obtained after applying the filter.

If more than one series have been input to Trace, **runtracematlab** returns a multiple structure, **out**, such that, for the i– series, **out(i)** contains the same four fields than those just described for a single series.

12 Running TS and TRACE Simultaneously in MATLAB

The Matlab function that can be used to run TS and Trace is

out = runtstmatlab(fname1,fname2,path)

where the input parameters are

fname1: a string containing the spec for TS, located in the SPECTRM subdirectory

fname2: a string containing the spec for Trace, located in the SPECTRC subdirectory

path: a string containing the path where TSTMATLAB is located (for example, 'C:\tst'). This input parameter is not input if the program is run from the directory where TSTMATLAB is located.

The output parameter is

out: a structure containing the output of TS and Trace as described earlier for the functions **runtsmatlab** and **runtracematlab**.

Example:

out = runtstmatlab('ipiale_trm','ipiale_trc')

In this case the program is run from the directory where TSTMATLAB is located. If it is run from another directory, then it should be something like

out = runtstmatlab('ipiale_trm','ipiale_trc','C:\tst')

In both cases, the strings **'ipiale_trm'** and **'ipiale_trc'** are the names of two Matlab function, located in the SPECTRM and SPECTRC subdirectories respectively, that return two cell–arrays containing the instructions for the TS and Trace programs. These functions were described in the earlier section.

During the execution of the **runtstmatlab** function, MATLAB changes the current directory first to the directory where the TS and Trace programs are located, then to the SPECTRM subdirectory. Once the spec function for TS is executed, MATLAB changes the directory again to where the TS and Trace programs are located and executes the TS programs. After that, a similar pattern is followed to run the Trace program, using the SPECTRC instead of the SPECTRM subdirectory.

13 Designing Filters With TRACE in MATLAB

The Matlab function that can be used to design a filter with TRACE is

runtrcfildes(fname,path)

where the input parameters are

fname: a string containing the spec for Trace, located in the SPECTRC subdirectory

path: a string containing the path where TSTMATLAB is located (for example, 'C:\tst'). This input parameter is not input if the program is run from the directory where TSTMATLAB is located.

The function produces a graph displaying the filter characteristics.

Example:

runtrcfildes('bpq1fildes_trc')

In this case the program is run from the directory where TSTMATLAB is located. If it is run from another directory, then it should be something like

runtrcfildes('bpq1fildes_trc','C:\tst')

In both cases, the string **'bpq1fildes_trc'** is the name of a Matlab function, located in the SPECTRC subdirectory, that returns a cell–array containing the instructions for the Trace program. These functions were described in an earlier section.

During the execution of the **runtrcfildes** function, MATLAB changes the current directory first to the directory where the Trace program is located and then to the SPECTRC subdirectory. Once the spec function is executed, MATLAB changes the directory again to where the Trace program is located and executes the Trace program.

The spec for **bpq1fildes_trc** is as follows.

```
function e = bpq1fildes_trc
%
%function to create the cell array necessary to run TRACE in
%MATLAB.
%the input for TRACE is a cell array e of size [n 4]. n is the number of
%series and 4 is the number of fields that contain the information for
%each series
%e{i,1} contains the title of the series
%e{i,2} contains the number of observations, the initial year, the
%       initial period and control
%e{i,3} contains the series path
%e{i,4} contains the namelist input

%filter specifications
spec='dd(1)=.30 dd(2)=.30 xp=.0625 xp2=.33333 xs=.40';
filtertype='3';                                    %filter type
name='bpq1';                                        %series name for TRACE


%the remaining lines should not be modified
input=['filter=' filtertype ' method=0 graph=1 '];
input=[input ' ' spec];
e{1,1}=name;
e{1,2}='15 1 1 1';
e{1,3}='0 0 0 0 0 0 0 0 0 0 0 0 0 0 0';          %fictitious data
e{1,4}=input;
```

14 Description of the spectran function

The **spectran** function computes the spectrum, coherence, phase delay, gain and cross correlations between a reference cycle and other cycles. The general form of this function is as follows

spectran(win,corlag,graph,per,lpath,ppath)

where the input parameters are

win: a number to select the kind of window used to smooth the periodogram. The options are

win=0 no window is applied (nonsmoothed periodogram)

win=1 the Blackman-Tukey window is applied win=2 the Parzen window is applied

win=3 the Tukey-Hanning window is applied

corlag: the number of cross correlations desired

graph: 1 if graphs are desired, 0 if not

per: the series frequency, 12 for monthly, 4 for quarterly, etc.

lpath: path to the file SERIE.SPA. The first path in SERIE.SPA should contain the reference cycle. The files that contain the cycles have each two columns, the first one with the original series and the other with the filtered series (the cycle).

ppath: path to the directory where SERIE.SPA is located. The output will be written for each cycle different from the reference cycle in two files in a subdirectory of the previous directory called SPECTAN. The first file, with the extension SPA, contains six columns with the frequencies used, coherence, phase delay, gain, periodogram for the reference cycle and periodogram for the other cycle. The second file, with the extension, COR, contains the cross correlations between the reference cycle and the other cycle.

Example:

Assume that the current directory is C:\tst, where TSTMATLAB has been installed. First, we run TRAMO, SEATS and TRACE using the command

out = runtstmatlab('oipitot_trm','oipitot_trc')

Then, we set the parameter values

win=2; corlag=16; graph=1; per=12; lpath='SERIE.SPA';
ppath='C:\tst';

and finally run the **spectran** function

spectran(win,corlag,graph,per,lpath,ppath)

15 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. (1999), "Three Equivalent Methods for Filtering Finite Nonstationary Time Series", *Journal of Business and Economic Statistics*, 17, 109–116.

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

Gómez, V. (2003), "Program Trace, Instructions for the User", *Mimeo*, Dirección General de Análisis y P.P., Ministry of Finance.

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.