Detection of environmental changes. Description of the TrendSpotter software

TECHNICAL REPORT · JUNE 2004

1 AUTHOR:

Hans Visser
PBL Netherlands Environmental Assessment …

Memorandum 007/2004 IMP

Detection of Environmental Changes

description of the TrendSpotter software

Hans Visser

December, 2 2004

This investigation has been performed by order and for the account of RIVM, within the framework of project S/550002/01/TO, Uncertainties, Transparency and Communication: Tools for Uncertainty Analysis.

RIVM, Postbus 1, 3720 BA Bilthoven, telefoon: 030 - 274 91 11; fax: 030 - 274 29 71

Abstract

Are there significant trends in temperatures and precipitation over the past hundred years? And do these series show some cyclic behaviour corresponding to sun spot numbers? Or, can we detect significant downward trends in concentrations of Particulate Matter? And what is the role of meteorological conditions? Are observed trends due to reduced emissions?

In this Memorandum we describe a generic statistical tool dealing with these type of questions. The technique for analysing environmental time series is based on *Structural Time Series Analysis and the Kalman filter*. These techniques are well-known in fields as Econometrics and Signal processing and Control, but are relatively unknown in Environmental research. Structural Time-Series models can be seen as a modular 'toolkit': we can estimate trends, cycles and the influence of explanatory variables (also called 'regressors' or 'predictors'). Also combinations of these components can be chosen. Moreover, confidence limits are given for all estimation results.

The associated software is called *TrendSpotter* and has been made available for both UNIX and PC. Early versions of TrendSpotter were developed at KEMA, under the name *KALFIMAC*. This Memorandum gives the details for running the PC version of the software.

Contents

Summary

Research into environmental issues yields a wide variety of data, the analysis of which can raise a number of questions, including:

- Do the data represent a trend, and is any increase or decrease statistically significant?
- Can cyclic signals be detected, and if so, what do they look like?
- What is the effect on the data of such external variables as meteorological conditions and the state of the economy?
- What should be done about missing data?
- How can forecasts be generated, and how reliable are they?

Answers to these questions can be found by modelling the relevant physical, chemical, biological, or meteorological relationships (white box modelling), or by means of statistical data-focused models (black box modelling). Intermediate forms may also be used (grey box modelling).

This Memorandum describes a generic statistical technique that seeks to provide answers to the above questions. The method is based on *structural time series models* and the *Kalman filter*. The method has been implemented in a software package, TrendSpotter, a version of which has recently become available for PC use.

The TrendSpotter software for the analysis of environmental measuring sequences started development as KALFIMAC at the KEMA institute. The package was purchased by RIVM in 1996, and with the permission of KEMA is currently being improved and modified for RIVM use (PC implementation and use of S-PLUS).

The method offers a number of unique features that are highly relevant to environmental research. Three of these features are:

- Trend estimates with one or more points of inflection along the time axis, with full availability of uncertainty information;
- Cycle estimates, with the form of the cycle being free to evolve in time;
- Estimates of weighting factors for explanatory variables, the weighting factors being free to change in time.

Essential in these features is that the method rather than the modeller indicates whether the trend displays points of inflection or whether the shape of a cycle remains constant in time.

This Memorandum provides details for running the TrendSpotter software. It describes the format of the input and output files using a detailed example. In addition, it describes how analyses and graphs are generated using S-PLUS. Finally, it describes the contents of the TrendSpotter installation CD-ROM. In addition, an example is given showing how flexible trends are estimated complete with uncertainty information. Mathematical details of the Kalman filter are summarized in an Appendix.

1. Introduction

This Memorandum provides a detailed description of the TrendSpotter software, the purpose of which is to structure one or more measuring sequences in time. The time series method used is that of structural time series models. This type of model was developed within the discipline of Econometrics. A number of structural time series models have been described in Visser (2003, Appendix A). Once a model is selected, estimates are obtained using the Kalman filter. Mathematical formulae are also included in Visser (2003, Appendix A). A recent application for use on weather extremes over the past century was given by Tang (2003) and Visser (2004b).

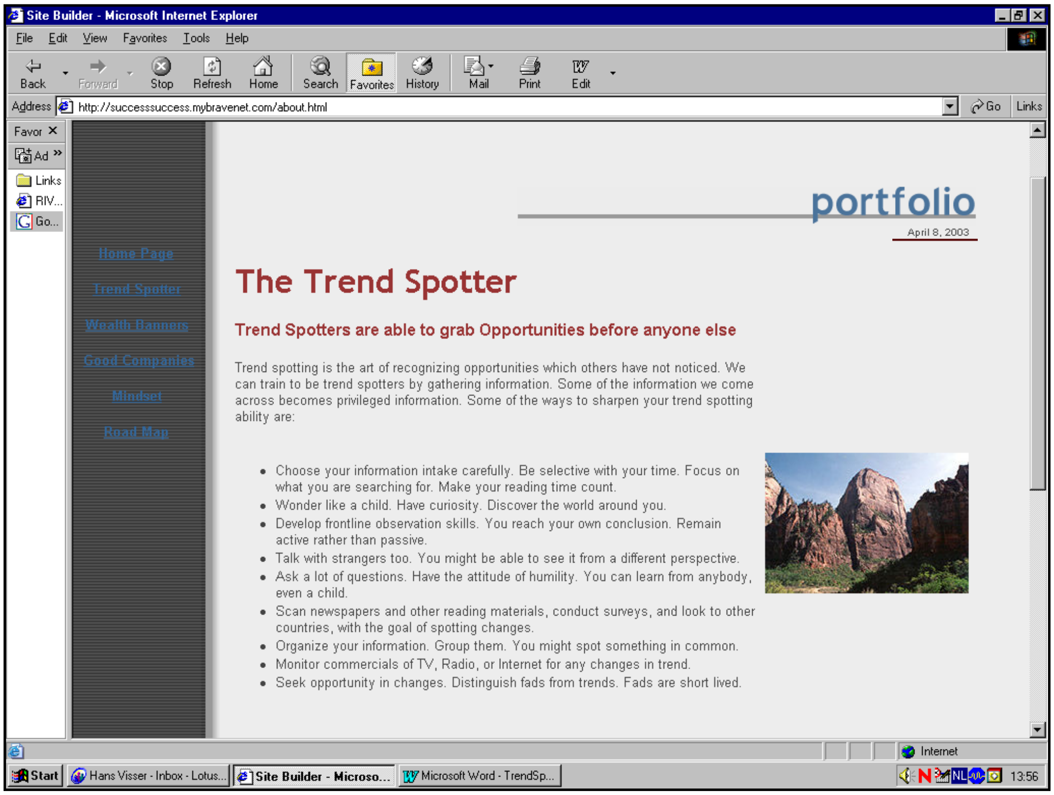
The development history of the TrendSpotter software is as follows. Initial development was carried out in 1984 by H. Visser and J. Molenaar (T.U. Eindhoven). The first version for general use was programmed by M.A.C. Mettes (HTS Arnhem) under the name of KALFIMAC, in which the programmer's initials form the last three letters of the package name. A second release was prepared by R. Leene (HTS Arnhem), who added a number of options, and created a "turbo version" for the optimization routine. This increased the execution speed of the software by 2 to 10 times. The third release was created by M. Habets (HTS Heerlen) and included the migration of the software from the KEMA mainframe computer, a UNIVAC 1100, to an APOLLO network environment. For a description of release 3.1, see Visser, Habets and Leene (1990). W.C.A. Maas and K. Friso added two routines to the APOLLO version to select variables in the context of regression models with time-dependent parameters. This resulted in release number 4.0.

The transition from release 4.0 to 5.0 involved three changes. First of all, the package was migrated from APOLLO UNIX to HPUX. Secondly, the SIMPLEPLOT plotting package was replaced with the UNIGRAPH interactive plotting package. And thirdly, ARIMA models were added to the trend estimate options. The modifications were carried out by A. Binzer of the Danish Technical University at Lyngby.

The current release, which is version 6.0 dating from May 2002, has been specially adapted for RIVM use (RIVM purchased the package from KEMA in 1996). Redundant statements from previous versions have been removed. Partly as a result of this, the options file has become more compact and easier to read. Also, the selection of regressors within the context of time-dependent weighting factors was *excluded* from version 6.0, as the selection procedures had not evolved sufficiently.

As the latest step in the development of the software, a PC version of KALFIMAC, named TrendSpotter, was developed by A. Beusen (IMP, RIVM).

The development of release 6.0 for UNIX and release 1.0 of the PC version took place as part of the Tools sub-project for Uncertainty Analysis (S/550002/01/TO).



The initial version of TrendSpotter was developed on a UNIVAC 1100 system under the name of KALFIMAC. This origin is reflected by the batch-like format of the input and output files. The latest development is a migration to the PC platform. The PC software goes under the name of TrendSpotter. By the way, unlike the Trend Spotter shown here, our TrendSpotter is not a portfolio package. They do have one thing in common, though: grabbing chances.

2. Working with TrendSpotter

This chapter provides a short description of how to use the TrendSpotter software. It assumes that all the relevant files for running the software are located in the *d:\TrendSpotter\run* base directory. The directory contains the executable and DLL files for running TrendSpotter, as well as an executable for tracking calculation errors in the software (a debug version).

2.1 Input and output

The software is started by double-clicking the **TrendSpotter.exe** executable. TrendSpotter will ask for the path to a file, *parset.inp*, which should contain the file specifications of the input and output files. By default, TrendSpotter will expect to find the *parset.inp* file in the directory containing the executable. After double clicking the executable, two consecutive screens are presented as shown in **Figures 1 and 2**.

The contents of the ***parset.inp*** file look like this :

```
PARAMETERS
INPUT DATA                :    data.km
INPUT OPTIONS             :    optie.km
LOGFILE                   :    logfile
OUTPUT PLOTTING DATA      :    writeall.out
OUTPUT COMPUTATIONS       :    uitvoer
```

If the path name of any of the above files is missing, the **d:*TrendSpotter**run*** base path is automatically assumed. If one or more file names are missing, the default file names listed above will be used.

For the temperature example which is used throughout this report the *parset.inp* file is:

```
PARAMETERS
INPUT DATA                :  D:\TrendSpotter Installation\Temperature.dat
INPUT OPTIONS             :  D:\TrendSpotter Installation\TemperatureT.opt
LOGFILE                   :  D:\TrendSpotter Installation\logfile
OUTPUT PLOTTING DATA      :  D:\TrendSpotter Installation\writeall.out
OUPUT COMPUTATIONS        :  D:\TrendSpotter Installation\Calculations.out
```

Here, the data, options and executables are place under the directory

d:\TrendSpotter Installation\

In the above example, the model specifications are in the options file, *optie.km*. The data can be found in *data.km*. These are the default files. The *optie.km* file contains all the options for the model to be estimated. A description of this file is included in section 3.3. The *data.km* file is an ASCII file containing the data in columns (obligatory fixed format). The columns are all right-aligned and the time always increases vertically from top to bottom, in equidistant steps. Instead of times, the first column may contain an ascending index. The data file must always contain a column containing a time stamp or index sequence.

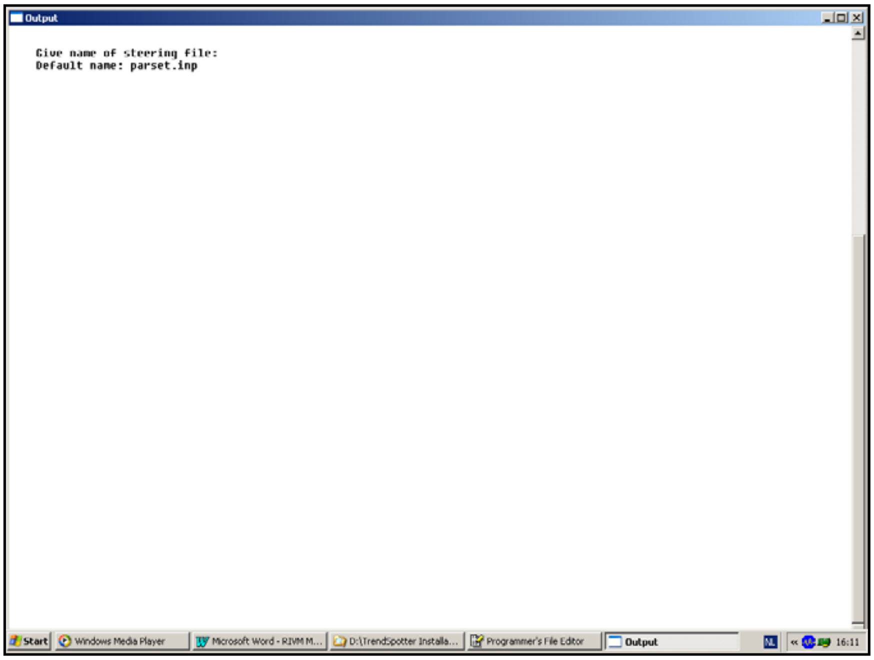


Figure 1 Opening screen after double clicking the *TrendSpotter.exe* file. Press 'Enter' to continue.

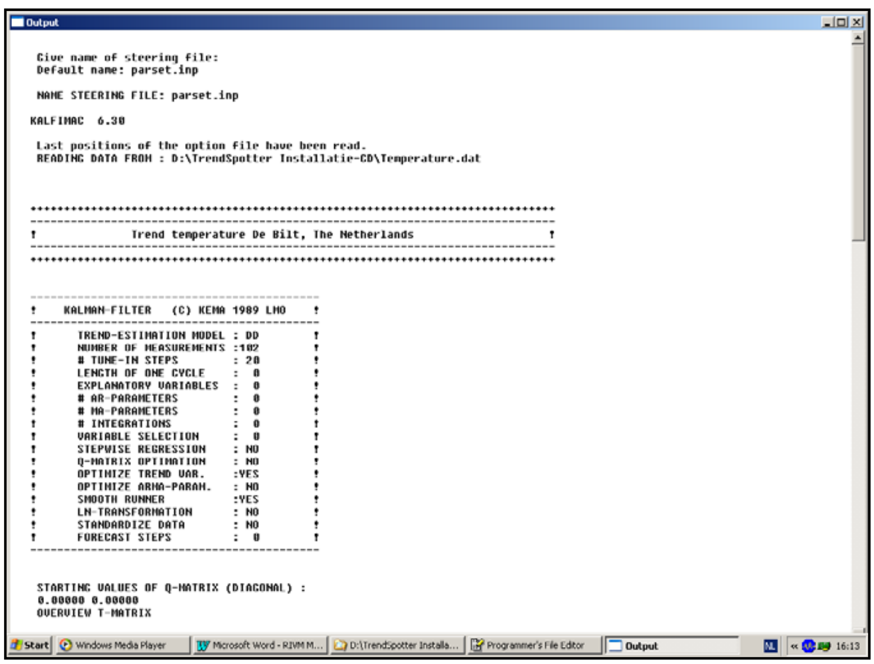


Figure 2 Upper part of the screen with calculation results. Results are also placed in D:\TrendSpotter Installation\Calculations.out

TrendSpotter creates output at three levels. Firstly, the output computations are added to the file specified for "OUTPUT COMPUTATIONS". This file contains all the relevant tables and statistical key figures. Note that unless the file name is changed, the output of any previous run will be overwritten when TrendSpotter is started. The default file name is *uitvoer*. The output will be discussed in detail in section 3.1.

Secondly, a file is created for the purpose of making graphics in S-PLUS. The data required by S-PLUS are entered in columns in an ASCII file, the default name of which is *writeall.out*. The contents of this file will be discussed in detail in section 3.2.

Thirdly, a file named *logfile* is created, which contains FORTRAN messages relating to the progress of TrendSpotter. This file can be useful for troubleshooting if the program aborts.

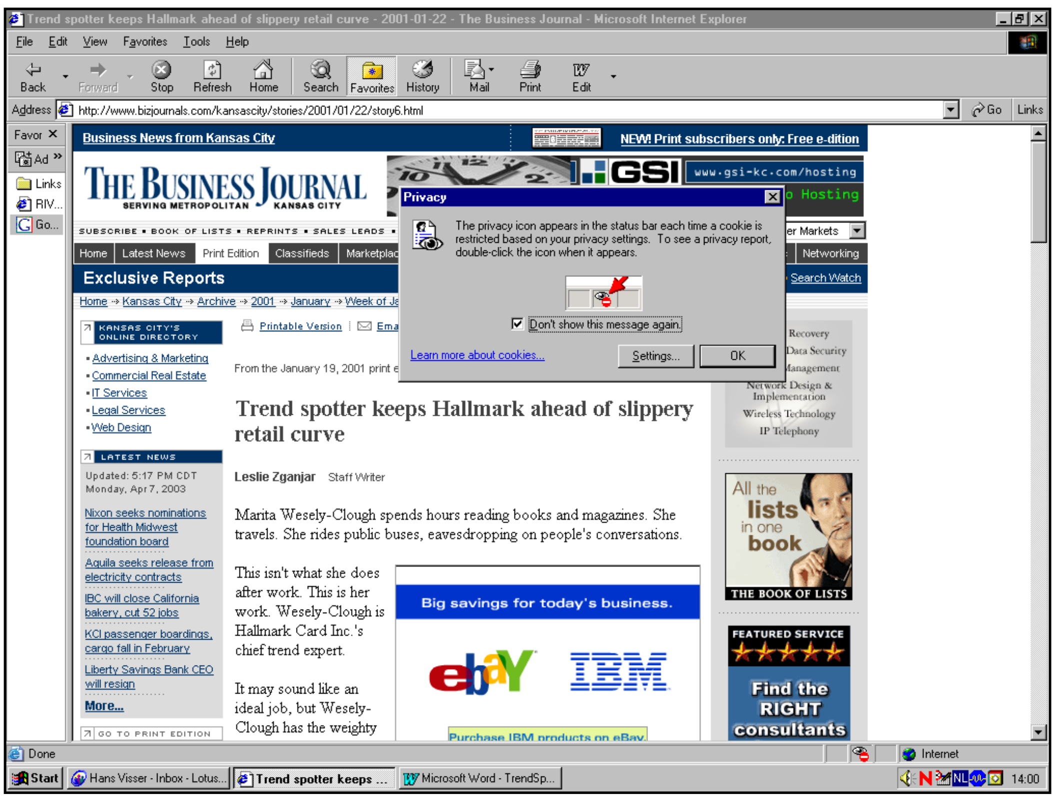
2.2 Plots and statistics

In principle, graphical output based on the computational results can be prepared using any available plotting software. All the relevant components, such as trend, trend-plus-cycle, and weighting factor for explanatory variables, each with its own standard deviation, are listed in the file specified after "OUTPUT PLOTTING DATA" (the default file name is *writeall.out*). Since it is convenient to carry out statistical tests on the standardised residuals of the model, S-PLUS was selected as the plotting package. The S-PLUS package is highly suitable both for graphical presentation of the results and for statistical analysis of the estimated results. For a general introduction to S-PLUS within RIVM, see Dekkers (2001), and for statistical methods in environmental research using S-PLUS, see Millard and Neerschal (2001).

The S-PLUS package contains a plotting and analysis routine in the form of a script, called PlotTrendSpotter. This script auto-selects the components making up the estimated time series model, and prepares the relevant graphs. These graphs are displayed in the Kalman screen, which can be pulled to the foreground by clicking the Window button on the S-PLUS menu bar. A second screen contains the graphs relating to tests on the original data, such as the Range-Mean plot. The display includes graphs calculated using the residuals of the model (in particular, the standardised innovations). These test concern normality, cycles, and the dependence of consecutive residuals.

The S-PLUS package includes easy-to-use features to smarten up the graphics for various publication purposes (see the S-PLUS manuals, the Help function, or simply use trial and error). The script is started by opening either the *PlotTrendSpotterTCE* or *PlotTrendSpotterTCE* script (select File => Open), and by pressing [F10] on the keyboard (= start execution of script). Selected parts of the script may be executed by blocking the desired number of lines and then pressing [F10]. The *PlotTrendSpotterTCE* script is included in Appendix A.

As a by-product, the script creates a data frame, writeall, based on the *writeall.out* file. The data frame can be used for additional statistics, for example to evaluate the preliminary performance of the Kalman filter by means of the MAD (mean of absolute deviation) criterium. Additional residuals testing can be performed on the stinnov variable (i.e. the vector, writeall$stinnov). By default, three graphical tests are performed on the residuals: a test to see whether the residuals follow a normal distribution (the so-called probability plot), a test to see whether consecutive residuals are statistically independent, and a test to see whether any cycles are present in the residuals.



The output of Trendspotter is saved in the writeall.out ASCII file, which forms the basis for each graphical visualisation. A standard visualisation can be obtained using the ***PlotTrendSpotterTCE*** *S-PLUS script. By the way, unlike Trend spotter shown above, TrendSpotter is not a postcard information system, although nice graphs could be send to colleagues by post or E-mail.*

In addition, S-PLUS is useful for preparing data (e.g. simulation examples), or converting data (e.g. obtaining monthly values based on daily values of a variable, y_t). From S-PLUS, the data frame can be written to an ASCII file.

Let us assume that the data frame is called *PM10concentrations*, and that its contents are to be written to a file named *PM10.dat* in the d*:/TrendSpotter Installation/* directory. To do so, type the following commands in the S-PLUS command window:

```
sink(file="d:/TrendSpotter Installation/PM10.dat")
PM10concentraties
sink( )
```

2.3 Options file format

The TrendSpotter options are selected by means of an options file. Selection flags for Yes or No are represented by the values 1 (one) and 0 (zero), respectively. The number spacing is irrelevant, since the file is read in a free format. Also, there is no distinction made between integers and real numbers, i.e. 1 is considered to be the same as 1.00.

The format of the options file is as follows:

1. Title: ------------------------...
   The title is a general description of the data or the specific run (maximum length 48 characters).

2. Selection of three model components: - - -
   This step contains three integer values representing trend selection, cycle selection, and regression model selection, respectively. The trend option may be any one of the following:
   0 = no trend model
   1 = Local Linear (LL) trend model
   2 = Doubly Differenced (DD) trend model, also denoted as Integrated Random Walk (IRW) model
   3 = Stochastic Level (SL) model, also denoted as local level model
   The value for the cycle is 0 (zero), i.e. no cycle, or the quantity, S, the period length.
   The value for the regression model is either 0 (zero), i.e. no explanatory variables, or the exact number of explanatory variables required.

   Note: when selecting explanatory variables and a cycle in a single model, no values may be missing from the explanatory variables.

3. Noise variances: - - - - ...
   The noise variances determine the flexibility of the various components. A value of 0.0 represents the most inflexible setting (see model (2)), whereas the highest level of flexibility is represented by 1.0 (the maximum value offered by the package). The order of the variances is trend-cycle-regression model. The LL model offers two noise variances, and the DD and SL trend model each offer only one. Any cycle will always have one noise variance, and each explanatory variable in the model has its own noise variance.

4. Adjustment period: -
   This is the adjustment period of the Kalman filter. For most applications, a value between 10 and 20 will suffice.

5. Noise variance optimization: -
The value of this flag can be 0, 1, 2, 3, 4, 5, 6, or 7. The table below lists the significance of each flag value.

Optimization flag	Trend	Explanatory variables or cycle model	ARIMA trend model
0	-	-	-
1	-	x	-
2	x	-	-
3	x	x	-
4	-	-	x
5	-	x	x
6	x	-	x
7	x	x	x

6. Ln transformation: -
This flag can be either 0, i.e. no transformation, or 1 to indicate that a ln transformation is to be used.

Note 1: The values in the plot output are not automatically reverse-transformed. A reverse-transformation could be added to the *TrendSpotterTCE* S-PLUS script.

Note 2: If the series, y_t contains negative values, and a log transformation is required, a constant must be added to the data to ensure that no negative values remain. This cannot be done within the TrendSpotter software.

7. Filtering or smoothing: -
This flag indicates filtering (0) or smoothing (1). For filtering, the plot output displays the $a_{t/t-1}$ estimators, and for smoothing, it displays the $a_{t/N}$ estimators.

8. Weighting factors and standardization: --
Two flags are required. The first flag refers to the weighting factors plot. If set to 0, all weighting factors from a regression model will be plotted with 2-σ confidence limits. If set to 1, the weighting factor will be multiplied by the corresponding explanatory variable, and the product will be plotted.
The second flag can be 0, 1, or 2. A value of 0 indicates no standardization of the variables from the model, whereas a value of 1 indicates standardization of all explanatory variables from the regression model. If a value of 2 is specified, the dependent variable y_t will also be standardized. If regression is not applicable, dummy values should be entered.

9. Missing values: - -
The first flag indicates whether any values are missing (1) or not (0).
The second number indicates the missing code. This number is required even if the first flag is set to 0.

Note 1: Please bear in mind that the missing code value will be included in the plot, which can result in stretching of the y axis scale! In the *TrendSpotterTCE* script this can be prevented by specifying the code for the missing readings.

Note 2: This option cannot be selected through the missing value ranges options (see nos. 10 and 11).

10. Missing value ranges: -
This option is used to specify the number of ranges with missing values. A flag set to 0 indicates no missing value ranges, 1 indicates a single range, and 2 indicates two ranges.
If no missing value ranges have been selected, go to step 12.

11. Missing value ranges limits: - -
(- -)
Specifies the start and end times of each missing range (one or two lines).

12. Differential: -
This line contains a flag to indicate whether a plot of the first differential of the LL or DD trend is to be included (1) or not (0) (this flag has no effect for other trends).

13. Explanatory variable plot titles: --------...
(--------...)
Each plot will be given its own title, each of which should be specified on a separate line.

14. Number of readings (number of records in data file): -

15. Forecast options: - - -
This line contains three numbers:

- The number of forecasts to be included (0 = none);
- For a model with explanatory variables, three possible situations exist. In the first situation, no information may be available about the values of y_t and the explanatory variables $x_{i,t}$ for the forecast period. In the second situation, although y_t may not be known and may therefore have to be forecast, values for $x_{i,t}$ may be available for the forecast period. In the third situation, both y_t and $x_{i,t}$ may be available for the forecast period. The latter situation occurs if model validation is required. This involves pretending that y_t data are missing from a significant location in the data sequence in order to have these forecast. Comparison of real and forecast y_t values provides a good impression of the forecasting capability of the estimated model.
- A flag can be set to indicate whether any additional data should be read for the explanatory variables and y_t in the forecast period (second and third situations). These values will be used to generate improved forecasts, but the y_t values will of course not be used; these occur only in the forecasts table. Of course, if the first option is set to 0, this flag has no effect.

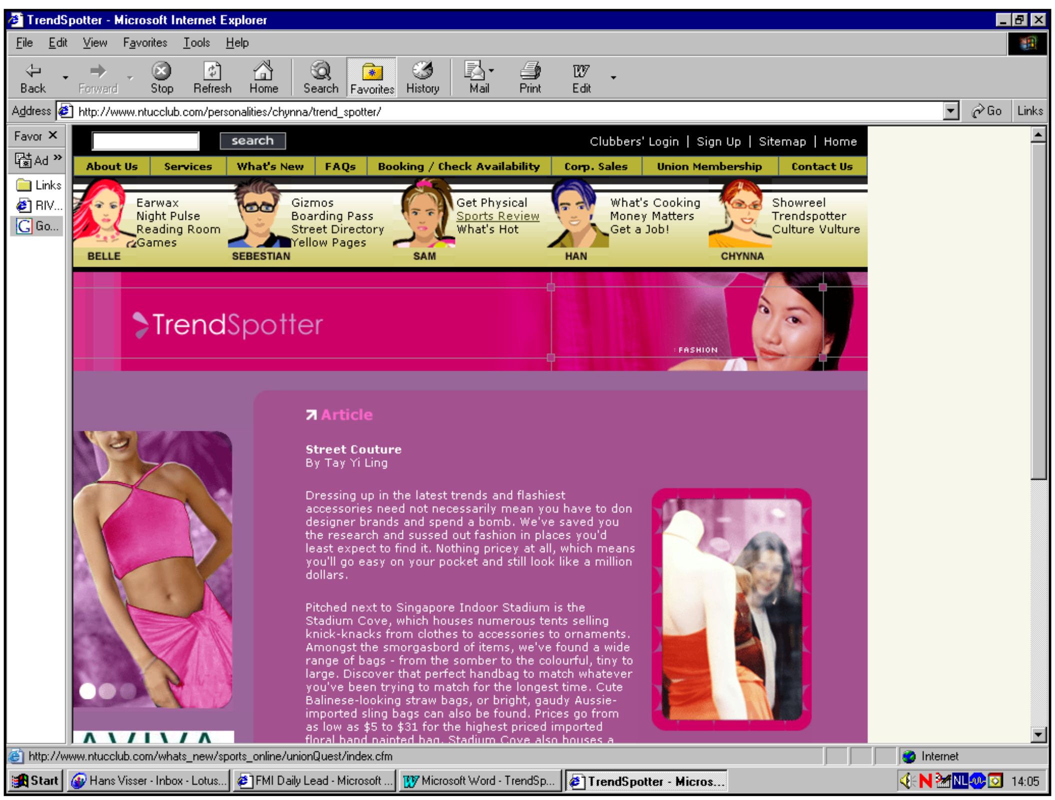
Note1: another way of forecasting can be performed by extending the data set with L future y_t values having the missing code. Now, the software will treat these 'future measurements' as 'very unreliable', and will give L estimates for the prediction interval of interest. Extra advantage: confidence limits are shown in the graphs generated with the S-PLUS script.
Note 2: the method in Note 1 generates predictions for the trend μ_t. The other method generates predictions for the measurements y_t (which are much wider).

16. Data file format: -----....
This specifies the FORTRAN format to be used for reading the data file (fixed format). The format must be specified between single quotation marks, e.g. '(3F6.2)' or '(F4.0.20X,F6.3)'.

17. Data file column positions: - - ...
(- - - ...)
Since the significance of the columns in the data file can be chosen at will, this setting must be included to indicate the contents type of each column. Each of the columns used in the data file is given an ascending number ranging from 1 up to and including M+2 (time, y_t, and M explanatory variables). In this step, the column numbers are listed so that the first number points to the time, the second to y_t, and the next M numbers to the explanatory variables in the required order The normal order in a data file is t; y_t; $x_{1,t}$; ...; $x_{M,t}$. This results in a line containing the integers 1 2 3 ... M+2. However, if for example y_t is included last, the line becomes: 1 M+2 2 3 4 ... M+1.

Note: The maximum number count per line is 10. If additional numbers are required, continue on a new line.

This completes the settings steps.



Structural time series models have not become fashionable in environmental research. The main reason for this is the lack of software. In addition to TrendSpotter there is only one commercial package available, Stamp (see Appendix D for references). By the way, unlike the TrendSpotter shown here, our TrendSpotter is nothing to do with fashion; its focus is on stripping measurements.

The TrendSpotter software uses a number of default settings when necessary. The current default settings are:

- maximum number of readings: 2000
- maximum number of explanatory variables: 3
- maximum number of forecasts: 10
- maximum length of a cycle: 25

2.4 Example of an options file

This section gives an example of an options file. Let us assume that we intend to estimate the DD trend (IRW model) in annual averaged temperatures for the 1901-2002 period (N = 102). See Visser (2003) for details of this example. These measurements are for main observatorium in De Bilt.

The options file contents will be as follows:

```
Trend temperature, De Bilt
2 0 0
0.0
20
2
0
1
0 0
1 -1.0
0
1
102
0 0 0
'(3x,F5.0,59X,F10.6)'
1 2
```

3. Description of TrendSpotter output

This chapter briefly discusses the significance of the major output files produced by TrendSpotter. The default file names for these files are *uitvoer* and *writeall.out*.

3.1 Tables

3.1.1 The *uitvoer* file

The significance of the columns in the first large table in the *uitvoer* file is as follows:

- T: time
- Y(T): the readings for y_t, or in the event of a log transformation, $\log(y_t)$
- RESIDUAL: residuals, i.e. (y_t - total model)
- FT: multiplied by the measuring noise variance, σ. This gives the variance of a single-step forecast error or innovation, ν_t . For filtering see formula (A.30), for smoothing see formula (A.35), both in Appendix D.
- MU(T): estimated trend
- BETA(T): estimated trend, but delayed by one time step, i.e. MU(T-1)
- DELTA 1: weighting factor for the first explanatory variable
- DELTA 2: weighting factor for the second explanatory variable

The significance of the columns in the second large table in the *uitvoer* file is as follows:

- time: time
- measured: the original readings for y_t
- model: total estimated model, or in case of a log transformation of y_t, exp(model)
- trend: estimated trend, μ_t
- 1: weighting factor for the first explanatory variable
- 2: weighting factor for the second explanatory variable

The significance of the columns in the third large table in the *uitvoer* file is as follows:

- T: time measured: the original readings for y_t
- model: total estimated model, or in the event of a log transformation of y_t, exp(model)
- TREND: standard deviation for the estimated trend, μ_t
- PARAMETER 1: standard deviation for the estimate of the first weighting factor, $\alpha_{1,t}$
- PARAMETER 1: standard deviation for the estimate of the second weighting factor, $\alpha_{2,t}$.

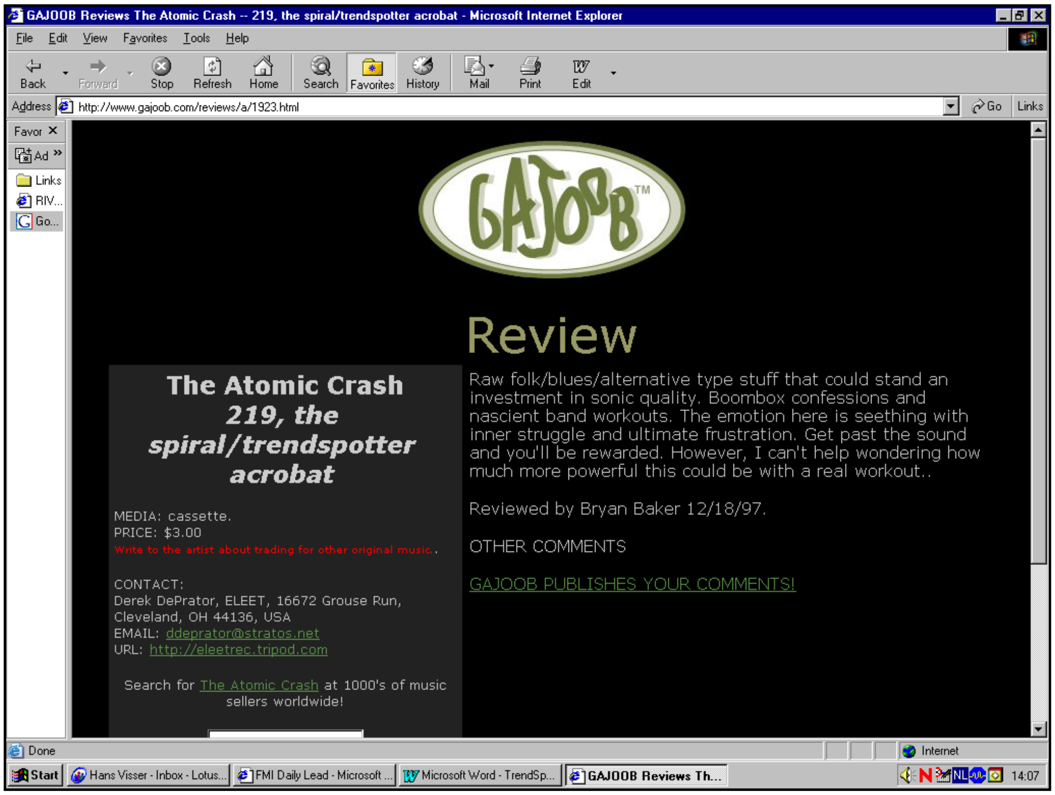
3.1.2 The *writeall.out* file

The significance of the columns in the *writeall.out* file is as follows:

- time: time
- measured: the original readings for y_t
- model: the total model, e.g. trend plus cycle plus effect of explanatory variable
- residual: residuals, i.e. (y_t - total model)
- stinnov: standardized innovations of the model
- trend: estimated trend
- sdtrend: standard deviation for the trend estimate (1-σ limit)
- mutNN: the (μ_N - μ_t) difference, with N the final time step
- SDmutNN: standard deviation for the (μ_N - μ_t) difference (1-σ limit)
- increment: first differential of the estimated trend, i.e. $\mu_t - \mu_{t-1}$
- sdinc: standard deviation for $\mu_t - \mu_{t-1}$ (1-σ limit)
- cycle: estimated cycle (varies about the mean value of 0.0)
- sdcycle: standard deviation for the cycle estimate (1-σ limit)
- exp1: weighting factor for the first explanatory variable
- sdexp1: standard deviation for this weighting factor (1-σ limit)
- expval1: the value of the explanatory variable, $x_{1,t}$.

  Note: If $x_{1,t}$ is standardized by TrendSpotter, this contains the standardized $x_{1,t}$ values.

- Ditto for any remaining explanatory variables.



The writeall.out ASCII file contains two tables of estimate results, as well as additional information about the optimization of unknown parameters and the explanatory power of the estimated model. By the way, unlike the trendspotter shown above, TrendSpotter is not a music information system, although it estimates may experienced as good music.

3.1.3 Effect of explanatory variables

When using regression models, it can be useful to know to what extent each of the explanatory variables contributes to the explanation of the noise surrounding the estimated trend. In other words, the variance, S of the (measurement minus estimated trend) sequence. It is also useful to know how much an estimated cycle contributes to the explanation of the model. The effect of an estimated cycle and explanatory variables is listed in a table in the uitvoer file, directly after the first large table.

The table lists variances per explanatory variable, i.e. the variance Si2 of (measurement - trend - ai,t * xi,t). This variance expresses the extent to which the fast variations around the trend can be explained by adding the explanatory variable, xi. The percentage of explained variance per variable is also listed in the table, and is defined as 100 * (S2 - Si2)/S2.

In addition, the table lists what all the explanatory variables together explain (S_{tot}^2):

$$S_{tot}^2 = var\left(y_t - \mu_t - \sum_{i=1}^{M} \alpha_{i,t} * x_{i,t} \right)$$

By way of further explanation, if a regression model is estimated with a trend, care must be taken if one or more explanatory variables also contain trends, since in that case, the Kalman filter will not know whether to explain the trend in y_t from the main trend or from the trend (or trends) in the explanatory variable (or variables).

The options file can be used to select a log transformation of the y_t readings. This can be useful if the measuring noise distribution is log normal rather than normal. In such cases, it is useful to be aware of the actual model being estimated. Based on the following model:

$y_t' = \ln(y_t) = \mu_t + a_t * x_t + \text{noise}$,

we find

$y_t = \exp(\mu_t) * \exp(\alpha_t * x_{1,t}) * \exp(\text{noise})$

A second model may result if, in addition to y_t, one or more explanatory variables also are log transformed. Since these transformations cannot be performed within TrendSpotter, they must be executed by some other means (e.g. S-PLUS) if they are required, and the results fed to TrendSpotter.

For example, assuming the following model:

$y_t' = \ln(y_t) = \mu_t + \alpha_t * \ln(x_t) + \text{noise}$,

we find

$y_t = e^{\mu t}\ x_t^{\alpha t}\ e^{\text{noise}}$

Note: *All values in the tables from the uitvoer file represent non-retransformed values. In other words, μ_t is given, but not exp(μ_t). The re-transformation can be performed using the S-PLUS script TrendSpotterTCE .*

3.2 Example

We will further examine the example with average annual temperatures. The options file has been given in Section 2.5. (See also section 6.2 in Visser, 2003). Appendix B lists the *uitvoer* file of this example, and Appendix C lists the *writeall.out* file.

The estimation results are shown in **Figures 3 and 4**. See also the example in Visser (2004a).

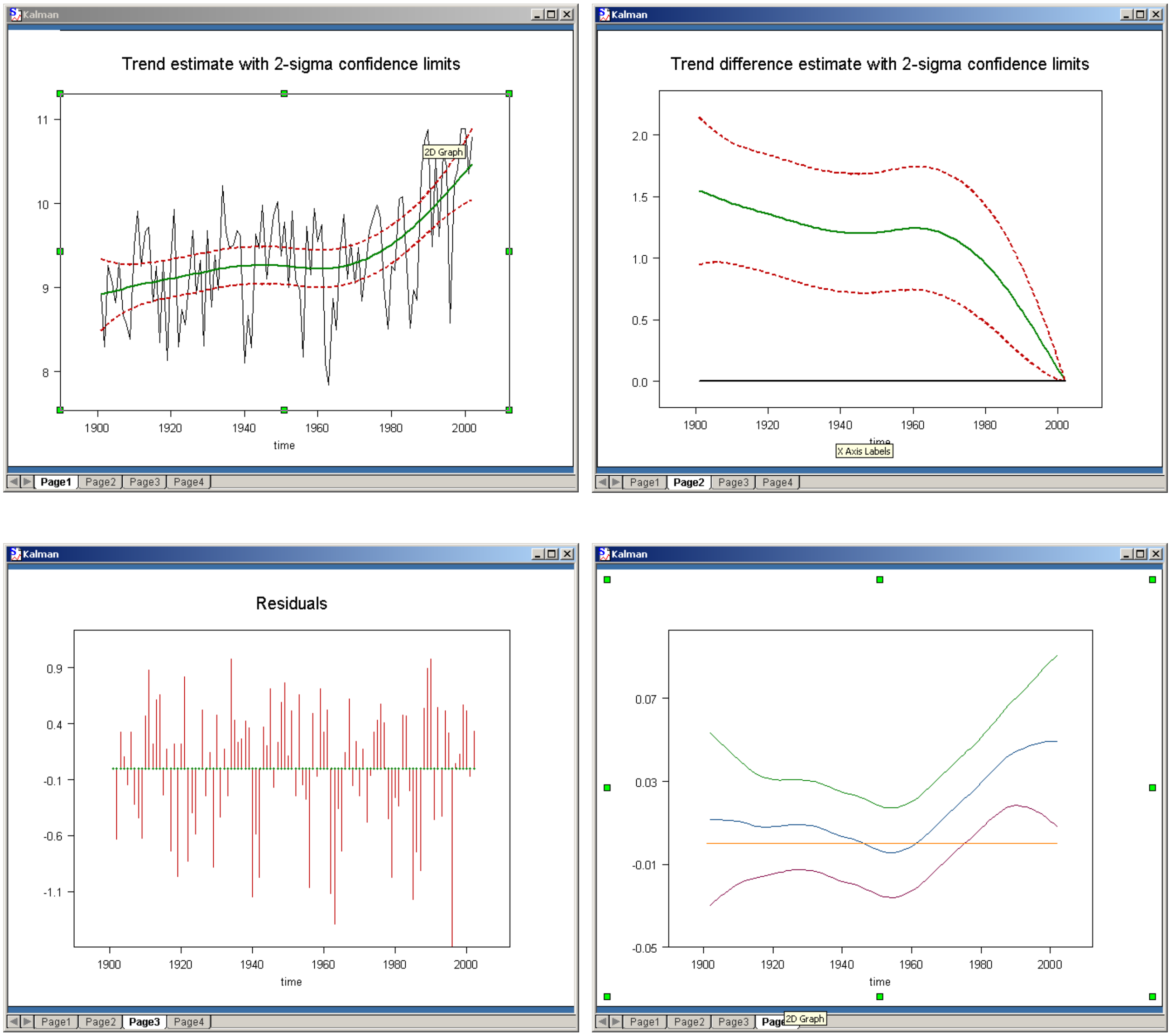


Figure 3 Standard output from the S-PLUS script *PlotTrendSpotterTCE*. The sheet is called 'Kalman'. The estimates are for the example *TemperatureT.opt* and *Temperature.dat* (annual averaged air temperatures for station De Bilt. Period: 1901-2002, N=102). Clockwise: (i) data with trend μ_t and corresponding 95% confidence limits, (ii) trend difference μ_{2002} - μ_t with corresponding 95% confidence limits, (iii) trend difference μ_t - μ_{t-1} with corresponding 95% confidence limits, and (iv) residuals.

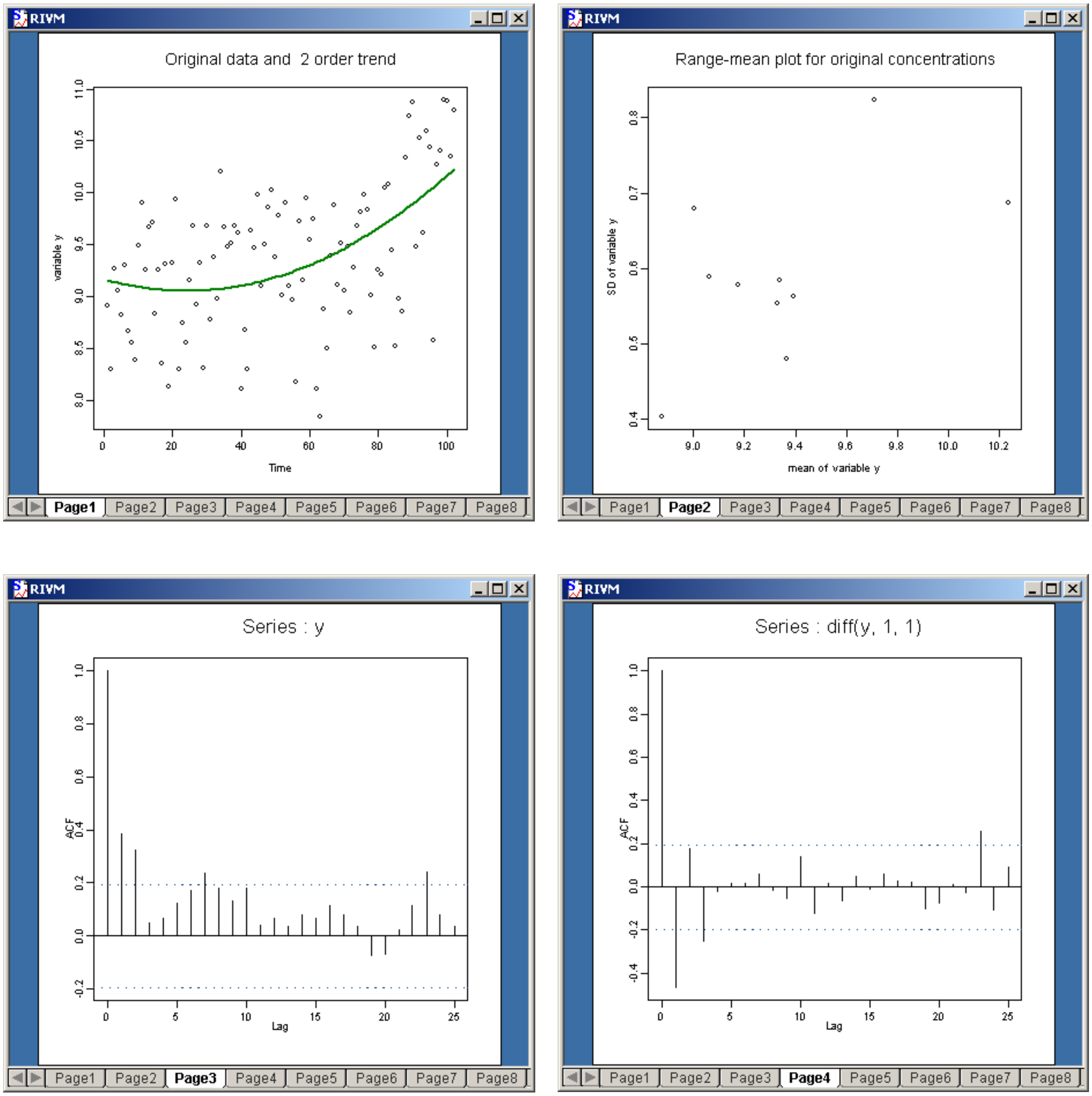


Figure 4 Diagnostic graphs based on the original measurements y_t.

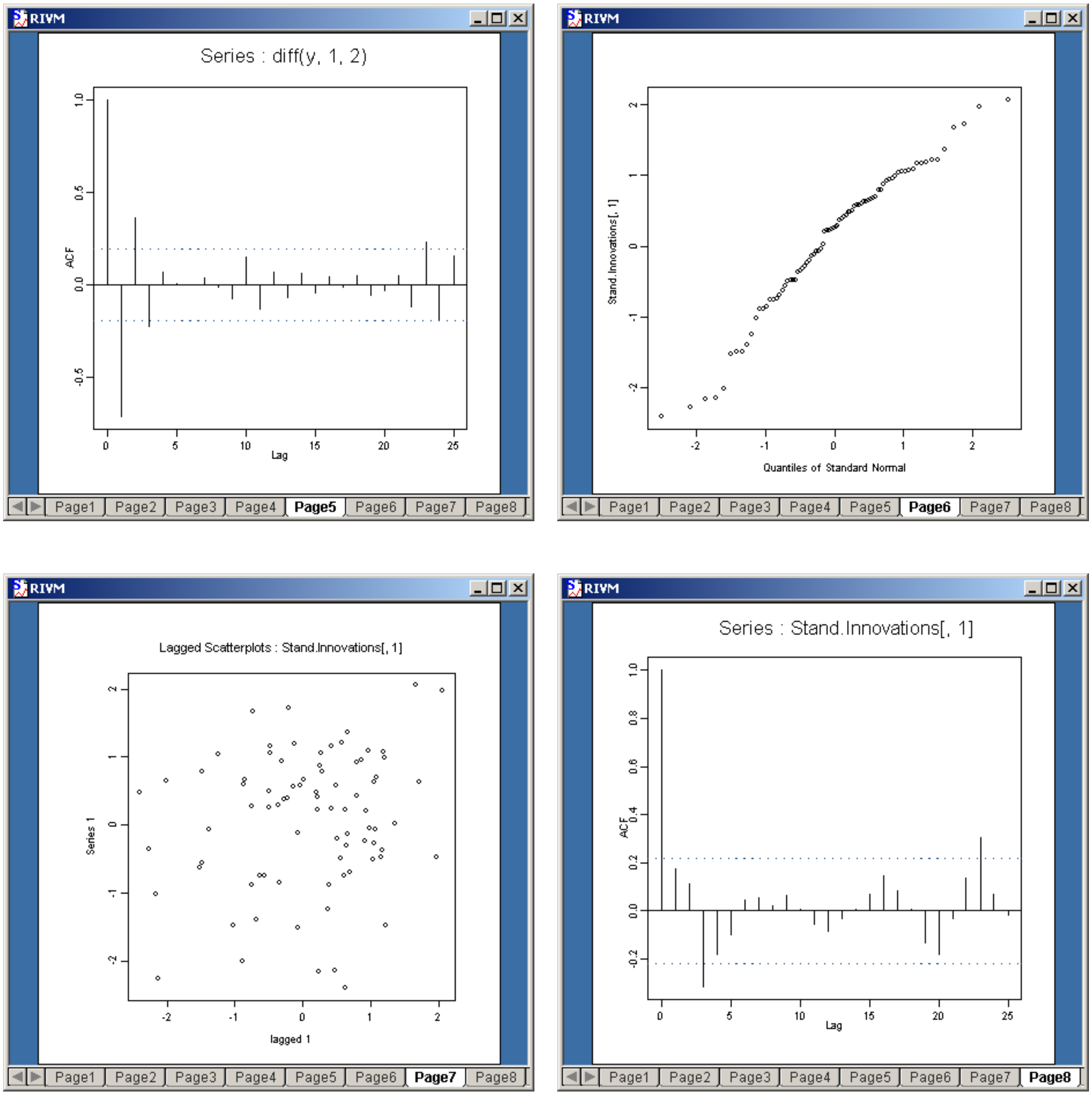


Figure 4 Continued. Clockwise: Autocorrelation function on doubly differenced y_t, normality plot for the standardized innovations (please see equation D.11 from Appendix D), idem for autocorrelation function, and lagged scatterplot for the standardized innovations.

4. Validation

Testing and validating structural time series models is a subject in itself, and will therefore not be discussed in detail in this context. An in-depth description can be found in the standard work by A.C. Harvey (1989, Chapter 5). A good recent reference work is Durbin and Koopman (2001). The following is a brief discussion of a number of issues.

A prerequisite for the basis of a well-defined structural time series model is that the noise processes from the model are white-noise processes, i.e. each noise process consists of uncorrelated readings with a constant variance over time (homoscedasticity). In addition, it is very helpful (though not essential) if the noise processes show a normal distribution. Under these conditions, the standardized single-step-forecasts e_t (i.e. the innovations ν_t from equation (D.11)) also show a normal distribution and mutual independence with a constant variance of 1.0. In the formula, ν_t represents the forecast error $y_t - y_{t/t-1}$, in which the forecast $y_{t/t-1}$ is determined using all the available information up to and including the time, t-1.

Tests on the e_t sequence are not performed by TrendSpotter, but must be calculated using S-PLUS on the *writeall.out* file, or directly from the *writeall* data frame created in S-PLUS if graphs were plotted from the estimate results. Examples of simple statements are:

- qqnorm(writeall$standinnov)
  This function results in a normality plot for the standardized innovations;
- acf(writeall$standinnov,Nlag)
  This function provides an autocorrelation function (ACF), calculated for the standardized innovations. The number of lags is Nlag;
- lag.plot(writeall$standinnov)
  This function results in a scatterplot of e_t against e_{t-1}. The plot gives an impression of the correlation between successive residuals.

The above functions (and others) can also be accessed using mouse clicks: Statistics => Time Series for example gives access to Autocorrelations. General statistical data such as skewness or kurtosis may be obtained through Statistics => Data Summaries =>

Summary Statistics. The required information can be obtained by clicking the Statistics screen of S-PLUS.

A much more extensive set of tests can be accessed after linking the **Envstats library**. With the beta 2.0 version linked, the following option becomes available: EnvironmentalStats =>

EDA, or Hypothesis Tests. The additional options are too numerous to discuss them here in any detail.

For valuable tests on residuals, refer to Harvey (1989, pp. 256-258). Harvey also includes a number of residuals tests that can be used to analyse the misspecification of the model used (pp. 258 - 260).

The goodness of fit of a time series model usually is tested with the prediction error variance, which, provided the time series is long enough, equals the sum of the squares of the innovations after the start-up effect of the Kalman filter has subsided (make a plot of the filtered innovations over time). For details, refer to Harvey (1989, pp. 263-264). Another option is to take the sum of the absolute values of the innovations (pp. 265-266).

For the purpose of comparing the forecasting powers of various structural models, either the above-mentioned prediction error variance may be used, or the maximized log likelihood value. Harvey offers additional options on pp. 269-270.


Trendspotter: Sinatra-café wordt in

JEANETTE ROTTEVEEL

AMSTERDAM - Het modernste bijbaantje voor jongeren: coolhunting. Na schooltijd de nieuwste trends signaleren onder leeftijdgenoten. Het is weer wat anders dan achter de kassa zitten of vakkenvullen. In Amerika is de jacht op hypes al een rage. Communicatiebureau Young Works in Amsterdam is een van de eerste Nederlandse bedrijven die zich van coolhunters bedienen.

Miranda Draaijer (21) speurt twee keer in de maand het centrum van Amsterdam af op jacht naar nieuwe trends. Haar specialiteiten zijn mode en uitgaan. "Te gek," vindt ze het bijbaantje. " Ik had mijn ogen er altijd al voor open staan. Ik let automatisch op kleding en gedrag van jongeren."

De studente media-informatiemanagement aan de Hogeschool van Amsterdam is een van de trendspotters die Young Works in bijna alle hoeken van Nederland inschakelen. Het communicatiebedrijf meent zo een beeld te krijgen van jongeren van achttien tot 24 jaar. Opdrachtgevers van het bedrijf zijn tot nog toe non-profitorganisaties. Een recente opdracht kwam van de provincie Groningen, die wilde weten wat jongeren van het bestuur vinden.

Via internet houden de trendspotters contact. Draaijer heeft nogal uiteenlopende missies. De ene keer bekijkt ze wat voor sieraden de jeugd draagt, de andere keer neemt ze broeken in het vizier. Wat haar opvalt, noteert ze in een boekje en speelt ze door aan haar opdrachtgever. Laatste opdracht die

Validation of the estimated model is important in order to gain confidence in the model and its forecasts. By the way, unlike the Trendspotter shown above, our TrendSpotter is not into cool hunting. What it does do is hunt for the latest environmental trends.

5. Installation

The TrendSpotter installation CD contains the files listed in the screendump shown in **Figure 5**. A short description of the purpose of each file follows below.

- The software is started by double-clicking *TrendSpotter.exe* (see section 2.1). An extended version for debugging purposes is included as TrendSpotter_debug.exe. The *salflibc.dll* file is needed to run the executable files.
- The input and output file names are listed in the *parset.inp* ASCII file (see section 2.1). The default behaviour of the program is to send output to the *uitvoer* and *writeall.out* ASCII files. Any FORTRAN error messages are sent to the *logfile* file.
- The *TrendSpotterTCE* and *TrendSpotterTE* files (MS-Word format) contain the scripts to be loaded into S-PLUS (block the text and copy with [Ctrl]+[C], then use [Ctrl]+[V] to paste the text into a newly opened script file within the S-PLUS environment). The former of these files is used to estimate trends (T), trends plus cycle (TC), and trends plus cycle plus up to 3 explanatory variables (TCE). The latter file is used to estimate trends plus explanatory variables (TE).

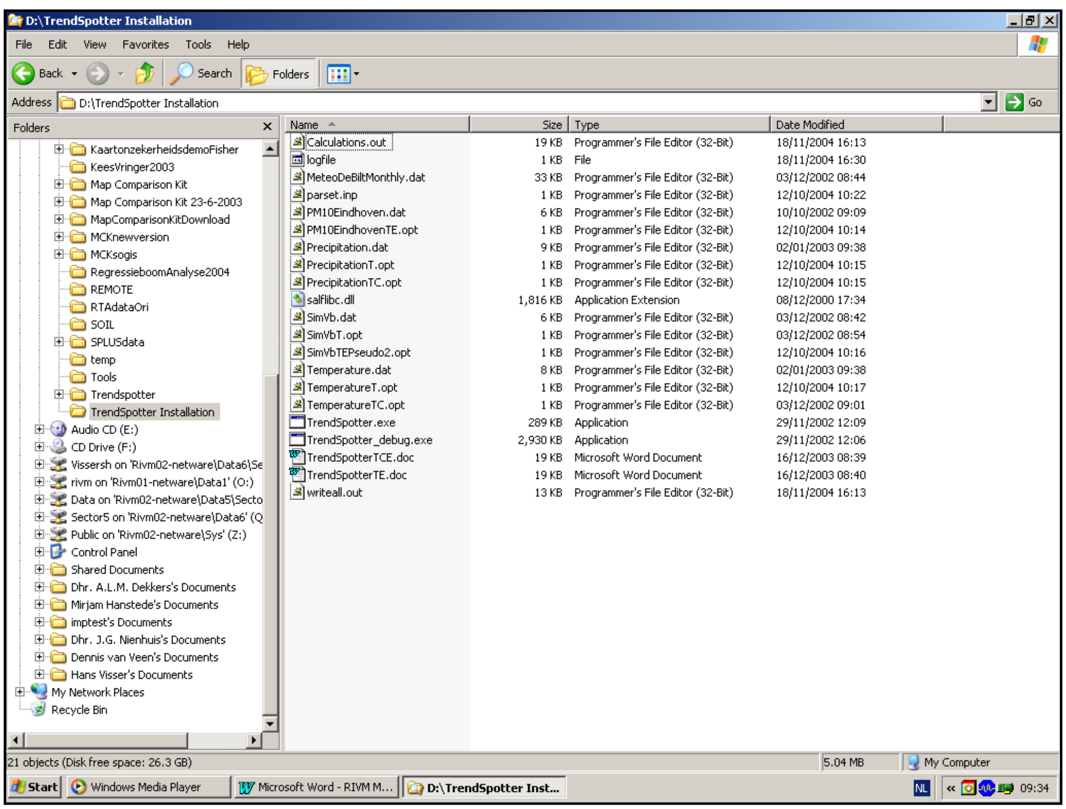
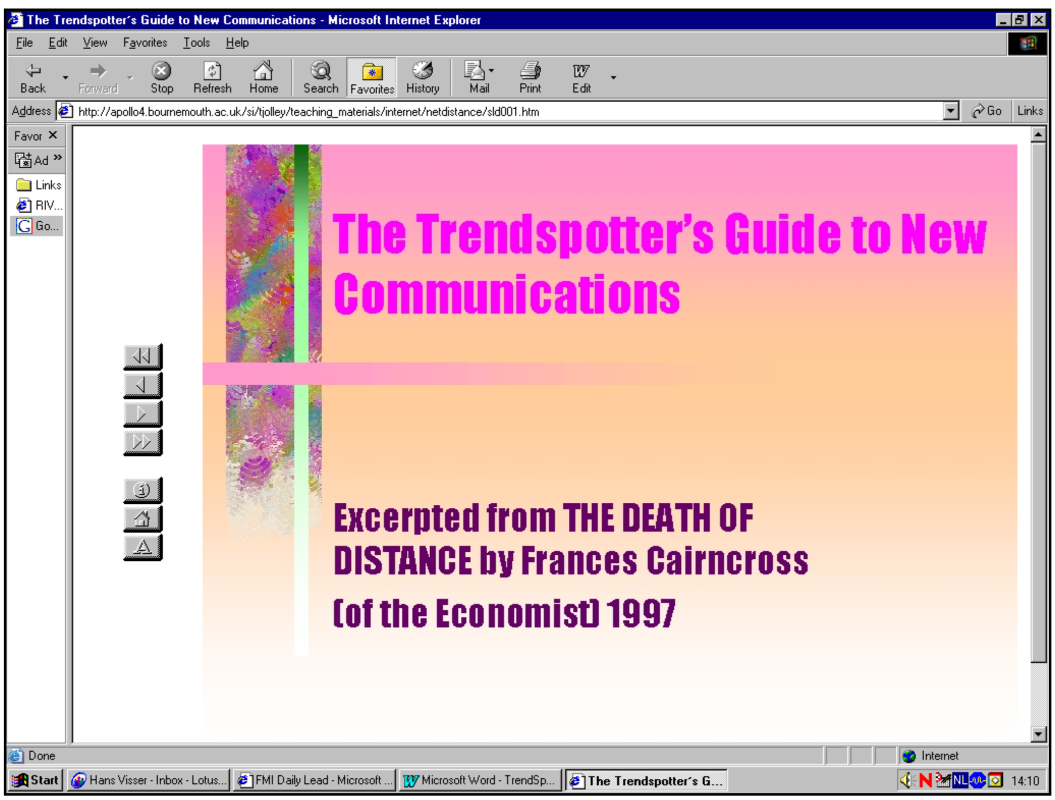


Figure 5 Files contained in the Installation CD.

The Installation CD includes a number of sample data and options files. These sample files form part of the applications given by Visser (2003, in Dutch). The references point to the sections in this report:

- SimVbT.opt (section 5.3) & SimVb.dat (section 5.2).
- SimVbTEPseudo2.opt (section 5.5) & SimVb.dat (section 5.2).
- TemperatureT.opt (section 6.2) & Temperature.dat (section 6.1). See also Appendices C and D.
- TemperatureTC.opt (section 6.4) & MeteoDeBiltMonthly.dat (section 6.1).
- PrecipitationT.opt (section 6.3) & Precipitation.dat (section 6.1).
- PrecipitationTC.opt (section 6.4) & MeteoDeBiltMonthly.dat (section 6.1).
- PM10EindhovenTE.opt (section 7.2) & PM10Eindhoven.dat (section 7.1).

In the notation'T' stands for Trend estimataion, TC for Trend plus Cyclic component, and 'TE' for Trend plus one or more explanatory variables.



The files supplied on the installation CD can be used to 'replay' the examples given in Visser (2003). This is a good way of preparing for the analysis of one's own data. By the way, unlike the Trendspotter shown here, our TrendSpotter is not a communication guide. However, detecting trends in environmental measuring sequences does play a major role in the interaction between science and policy.

Literature

Buringh, E., Opperhuizen, A. (editors), 2002. On health risks of ambient PM in the Netherlands: Executive Summary. RIVM report 650010 033. Bilthoven, the Netherlands.

Cleveland, R.W., Grosse, E., 1991. Computational methods for local regression. Statistics and Computing 1, 47-62.

De Jong, P., 1988. The likelihood for a state space model. Biometrika 75, 165-169.

Dekkers, A.L.M., 2001. S-PLUS voor het RIVM. Krachtig statistisch software gereedschap. RIVM-rapport 422 516 001.

Durbin, J. and Koopman, J., 2001. Time series analysis by state space methods. Oxford Statistical Science Series 24, Oxford University Press.

Gilbert, R.O., 1987. Statistical methods for environmental pollution monitoring. Van Nostrand Reinhold Company, New York.

Harvey, A.C., 1984. A unified view of statistical forecasting procedures. Journal of Forecasting 3, 245-275.

Harvey, A.C., 1989. Forecasting, structural time series models and the Kalman filter. Cambridge University Press, Cambridge.

Hess, A., Iyer, H., Malm, W., 2001. Linear trend analysis: a comparison of methods. Atmospheric Environment 35, 5211-5222.

Kitagawa, 1981. A nonstationary time series model and its fitting by a recursive filter. Journal of Time Series Analysis 2(2), 103-116.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Trans. ASME D, 82, 95-108.

Libiseller, C., Grimvall, A., 2002. Performance of partial Mann-Kendall tests for trend detection in the presence of covariates. Environmetrics 13, 71-84.

Millard, S.P., Neerchal, N.K., 2001. Environmental Statistics with S-PLUS. CRC press Boca Raton, Florida.

Koopman, S.J., Ooms, M., 2003. Time series modelling of daily tax revenues. Statistica Neerlandica 57(4), 439-469.

Tang, J., 2003. Veranderingen in klimaatextremen en weergerelateerde indicatoren in Nederland. Trends en onzekerheden in historische klimaatdata (1901-2003). RIVM-KNMI-TUDelft stagerapport.

van den Brakel, J., Visser, H., 1996. The influence of environmental conditions on tree-ring series of Norway Spruce for different canopy and vitality classes. Forest Science 42(2), 206-219.

van der Wal, J.T., Janssen, L.H.J.M., 2000. Analysis of spatial and temporal variations of PM_{10} concentrations in the Netherlands using Kalman filtering. Atmospheric Environment 34, 3675-3687.

van Oldenborgh, G.J., van Ulden, A., 2003. On the relationship between global warming, local warming in the Netherlands and changes in circulation in the 20th century. Int. J. of Climatology 23, 14, 1711-1724.

Visser, H., 1994. Regression models with time-varying parameters: applications in the environmental sciences. Proefschrift Universiteit van Amsterdam.

Visser, H., 2003. Detectie van milieuveranderingen. Een toepassing van Structurele Tijdreeksmodellen en het Kalmanfilter. RIVM-rapport 550002002.

Visser, H., Habets, M., Leene, R., 1990. KALFIMAC, a software package to analyse time series with trend, cycli and explanatory variables. Release 3.1. KEMA report 50385-MOF 89-3282.

Visser, H., Molenaar, J., 1995. Trend estimation and regression analysis in climatological time series: an application of structural time series models and the Kalman filter. Journal of Climate 8(5), 969-979.

Visser, H., Noordijk, H., 2002. Correcting air pollution time series for meteorological variability. RIVM report 722601007, Bilthoven, The Netherlands.

Visser, H., 2004a. Estimation and detection of flexible trends. Atmospheric Environment 38(25), 4135-4145.

Visser, H., 2004b. The significance of climate change in the Netherlands. An analysis of historical (1901-2003) and future trends (2004-2020) in weather conditions, weather extremes and weather-related impacts. RIVM report 550002 007 (in press).

Young, P.C., Lane, K., Ng, C.N., Palmer, D., 1991. Recursive forecasting, smoothing and seasonal adjustment of nonstationary environmental data. Journal of Forecasting 10, 57-89.

Appendix A S-PLUS-script TrendSpotterTCE

```
###############################################################################
#
# This script generates plots for output from TrendSpotter.
# The software package TrendSetter runs on a normal PC.
# Files can be changed or imported from directory d:\\TrendSpotter Installation\\
#
# Data are imported from the file 'd:\\TrendSpotter Installation\\writeall.out',
# and depending on the presence of trend, cycle and explanatory variables, plots
# are generated by S-PLUS.
#
# Date: 15 January, 2004
# Programmer: Hans Visser (IMP)
#
###############################################################################
#
# Step 1 Import data from file 'TrendSetter/run/writeall.out' into dataframe
   'writeall'.
#
menuSelectData(data.source = "Import File", existing.name = "writeall")

guiEval(remove('writeall',where=1), 'writeall', removal=T)

import.data(FileName = "D:\\TrendSpotter Installation\\writeall.out",
   FileType = "ASCII",
   ColNames = "",
   Format = "",
   TargetStartCol = "1",
   DataFrame = "writeall",
   NameRow = "",
   StartCol = "1",
   EndCol = "END",
   StartRow = "5",
   EndRow = "END",
   Delimiters = "",
   SeparateDelimiters = F,
   PageNumber = "1",
   RowNameCol = "",
   StringsAsFactors = "Auto",
   VLabelAsNumber = F,
   DatesAsDoubles = F,
   Filter = "",
   OdbcConnection = "",
   OdbcSqlQuery = "")

if(sum(names(writeall) == "trend") == 1) {
  writeall$Trendmin <- writeall$trend - 2*writeall$sdtrend
  writeall$Trendmax <- writeall$trend + 2*writeall$sdtrend }

if(sum(names(writeall) == "trend") == 1) {
  writeall$mutNNmin <- writeall$mutNN - 2*writeall$SDmutNN
  writeall$mutNNmax <- writeall$mutNN + 2*writeall$SDmutNN }

if(sum(names(writeall) == "trend") == 1 & sum(names(writeall) == "cycle") == 1) {
   writeall$Trendcyc <- writeall$trend + writeall$cycle }

if(sum(names(writeall) == "cycle") == 1) {

  writeall$Cyclemin <- writeall$cycle - 2*writeall$sdcycle
  writeall$Cyclemax <- writeall$cycle + 2*writeall$sdcycle }

if(sum(names(writeall) == "exp1") == 1) {
  writeall$Exp1min <- writeall$exp1 - 2*writeall$sdexp1
  writeall$Exp1max <- writeall$exp1 + 2*writeall$sdexp1
  writeall$Inflexp1 <- writeall$exp1 * writeall$expval1
  writeall$Inflexp1min <- writeall$Exp1min * writeall$expval1
```

```
  writeall$Inflexp1max <- writeall$Exp1max * writeall$expval1 }

if(sum(names(writeall) == "exp2") == 1) {
  writeall$Exp2min <- writeall$exp2 - 2*writeall$sdexp2
  writeall$Exp2max <- writeall$exp2 + 2*writeall$sdexp2
  writeall$Inflexp2 <- writeall$exp2 * writeall$expval2
  writeall$Inflexp2min <- writeall$Exp2min * writeall$expval2
  writeall$Inflexp2max <- writeall$Exp2max * writeall$expval2 }

if(sum(names(writeall) == "exp3") == 1) {
  writeall$Exp3min <- writeall$exp3 - 2*writeall$sdexp3
  writeall$Exp3max <- writeall$exp3 + 2*writeall$sdexp3
  writeall$Inflexp3 <- writeall$exp3 * writeall$expval3
  writeall$Inflexp3min <- writeall$Exp3min * writeall$expval3
  writeall$Inflexp3max <- writeall$Exp3max * writeall$expval3 }


if(sum(names(writeall) == "increment") == 1) {
  writeall$sdinc[1] <- NA
  writeall$increment[1] <- NA
  writeall$Incmin <- writeall$increment - 2*writeall$sdinc
  writeall$Incmax <- writeall$increment + 2*writeall$sdinc }

zero <- rep(0.0,length(writeall$time))
writeall$zero <- zero

writeall[1:4,]
#
# Missing values have code -1. These values are set to NA.
writeall[writeall == -1.00] <- NA
writeall
#
###############################################################################
#
# Step 2. Here we make 2 plots from which we can judge if a transformation on
#         the original concentrations is needed. If points in the Range-mean plot
#         show an increasing tendency, this points to a log-trafo or
#         to a division by an estimated polynomial trend.
#         N.B.: in the first line the dataframe 'aa.dat' with y en x values
#               is copied to 'x.dat'. We only use the column with variable
#               JJJJ and the y-variable. First 'JJJJ', then 'yvar'.
#
x.dat <- data.frame(writeall$measured)
len <- length(x.dat[,1])
len
nperiod <- trunc(len/10)
index <- rep(1,10)
for (i in 2:nperiod)
{index <- c(index,rep(i,10))}
index <- c(index,rep(nperiod,len - nperiod*10))
x.dat <- data.frame(cbind(1:len,x.dat[,1],index))
names(x.dat)[1:3] <- c("time","variable y","index")
x.dat
#
Polyorder <- 2
x.lm <- lm(x.dat[,2] ~ poly(x.dat[,1], Polyorder), na.action = na.exclude)

trend <- predict(x.lm)
plot(x.dat[,1], x.dat[,2], xlab= "Time", ylab= "variable y")
lines(x.dat[,1], trend, col = 4, lwd = 3)
title(paste("Original data and ", Polyorder, "order trend"))
#
x.mean <- aggregate.data.frame(x.dat[,2],x.dat[,3],mean,na.rm=T)
x.mean
x.var <- aggregate.data.frame(x.dat[,2],x.dat[,3],var,na.method="omit")
x.var
x.sd <- sqrt(x.var$x)
x.ave <- x.mean$x
plot(x.ave,x.sd, xlab= "mean of variable y", ylab= "SD of variable y")
```

```
title("Range-mean plot for original concentrations")
###############################################################################
#
# Step 3: tests for the choice of a proper trendmodel.
# If the data are stationary for the undifferenced y's: no trend!
#
y <- writeall$measured
y <- na.omit(y)
y
acf(y,25)
acf(diff(y,1,1),25)
acf(diff(y,1,2),25)
#
###############################################################################
#
# Step 4: Tests on standardized innovations
#
z <- data.frame(writeall$stinnov)
Stand.Innovations <- na.omit(z)
Stand.Innovations
qqnorm(Stand.Innovations[,1])
lag.plot(Stand.Innovations[,1])
acf(Stand.Innovations[,1],25)
#
###############################################################################
#
# Step 5: graphs of all relevant variables.
#
#
#
if(sum(names(writeall) == "trend") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, measured, trend,
    Trendmin, Trendmax",
  GraphSheet= "Kalman", Page= "New")

guiModify( "LinePlot", Name = "Kalman$1$2",
    LineColor = "User4",
    LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$1$3",
    LineStyle = "Med Dash",
    LineColor = "User8",
    LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$1$4",
    LineStyle = "Med Dash",
    LineColor = "User8",
    LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$1$1",
    LineColor = "User1")

guiCreate( "MainTitle", Name = "Kalman$1$1",
    Title = "Trend estimate with 2-sigma confidence limits",
    FillColor = "Transparent")
}
#
###############################################################################
#
#
if(sum(names(writeall) == "trend") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, mutNN, mutNNmin,
    mutNNmax,zero",
  GraphSheet= "Kalman", Page= "New")

guiModify( "LinePlot", Name = "Kalman$2$1",
    LineColor = "User4",
```

```
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$2$2",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")


guiModify( "LinePlot", Name = "Kalman$2$3",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$2$4",
   LineColor = "User1",
   LineWeight = "2")

guiCreate( "MainTitle", Name = "Kalman$2$1",
   Title = "Trend difference estimate with 2-sigma confidence limits",
   FillColor = "Transparent")
}
#
##############################################################################
#
guiPlot( PlotType = "Y Zero Density", DataSet = "writeall", Columns = "time,
   residual, zero",
         GraphSheet= "Kalman", Page= "New")

guiModify( "LinePlot", Name = "Kalman$3$2",
   LineColor = "User4",
   LineWeight = "3")

guiModify( "LinePlot", Name = "Kalman$3$1",
   LineColor = "User8")

guiCreate( "MainTitle", Name = "Kalman$3$1",
   Title = "Residuals",
   FillColor = "Transparent")
#
##############################################################################
#
if(sum(names(writeall) == "trend") == 1 & (sum(names(writeall) == "exp1") == 1 |
   sum(names(writeall) == "cycle") == 1) )   {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, measured, model,
   trend",
   GraphSheet= "Kalman")

guiModify( "LinePlot", Name = "Kalman$4$1",


    SymbolStyle = "X",
   SymbolColor = "User1")

guiModify( "LinePlot", Name = "Kalman$4$1",
   LineColor = "User1")

guiModify( "LinePlot", Name = "Kalman$4$2",
   LineColor = "User8")

guiModify( "LinePlot", Name = "Kalman$4$3",
   LineWeight = "2")

guiCreate( "MainTitle", Name = "Kalman$4$1",
   Title = "Measurements, model estimates and trend\
",
   FillColor = "Transparent")

}
```

```
############################################################################
#

if(sum(names(writeall) == "cycle") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, cycle, Cyclemin,
   Cyclemax",
       GraphSheet= "Kalman", Page="New")

guiCreate( "MainTitle", Name = "Kalman$5$1",
   Title = "Cyclic component with 2-sigma conf. limits",
   FillColor = "Transparent")

guiModify( "LinePlot", Name = "Kalman$5$1",
   LineColor = "User4",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$5$3",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$5$2",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")
}

############################################################################
#
if(sum(names(writeall) == "exp1") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, exp1, Exp1min,
   Exp1max",
       GraphSheet= "Kalman", Page="New")

guiCreate( "MainTitle", Name = "Kalman$6$1",
   Title = "Weight for explanatory variable 1, with 2-sigma conf. limits",
   FillColor = "Transparent")

guiModify( "LinePlot", Name = "Kalman$6$1",
   LineColor = "User1",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$6$3",
   LineStyle = "Med Dash",


   LineColor = "Black",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$6$2",
   LineStyle = "Med Dash",
   LineColor = "User1",
   LineWeight = "2")

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, Inflexp1,
   Inflexp1min, Inflexp1max",
  GraphSheet= "Kalman", Page="New")

guiModify( "LinePlot", Name = "Kalman$7$2",
   LineStyle = "Short Dash",
   LineColor = "User4",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$7$1",
   LineColor = "User1",
   LineWeight = "2")
```

```
guiModify( "LinePlot", Name = "Kalman$7$3",
   LineStyle = "Short Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$7$3",
   LineStyle = "Solid")

guiModify( "LinePlot", Name = "Kalman$7$2",
   LineStyle = "Solid")

guiModify( "LinePlot", Name = "Kalman$7$1",
   LineWeight = "3")

guiCreate( "MainTitle", Name = "Kalman$7$1",
   Title = "Response to expl. variable 1 with 2-sigma conf. limits",
   FillColor = "Transparent")

}

##############################################################################
#
if(sum(names(writeall) == "exp2") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, exp2, Exp2min,
   Exp2max",
      GraphSheet= "Kalman", Page="New")

guiCreate( "MainTitle", Name = "Kalman$8$1",
   Title = "Weight for explanatory variable 2, with 2-sigma conf. limits",
   FillColor = "Transparent")

guiModify( "LinePlot", Name = "Kalman$8$1",
   LineColor = "User1",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$8$3",
   LineStyle = "Med Dash",
   LineColor = "Black",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$8$2",
   LineStyle = "Med Dash",
   LineColor = "User1",
   LineWeight = "2")

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, Inflexp2,
   Inflexp2min, Inflexp2max",
   GraphSheet= "Kalman", Page="New")

guiModify( "LinePlot", Name = "Kalman$9$2",
   LineStyle = "Short Dash",
   LineColor = "User4",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$9$1",
   LineColor = "User1",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$9$3",
   LineStyle = "Short Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$9$3",
   LineStyle = "Solid")

guiModify( "LinePlot", Name = "Kalman$9$2",
   LineStyle = "Solid")
```

```
guiModify( "LinePlot", Name = "Kalman$9$1",
   LineWeight = "3")

guiCreate( "MainTitle", Name = "Kalman$9$1",
   Title = "Response to expl. variable 2 with 2-sigma conf. limits",
   FillColor = "Transparent")

}
#
##############################################################################
#
if(sum(names(writeall) == "exp3") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, exp3, Exp3min,
   Exp3max",
       GraphSheet= "Kalman", Page="New")

guiCreate( "MainTitle", Name = "Kalman$10$1",
   Title = "Weight for explanatory variable 3, with 2-sigma conf. limits",
   FillColor = "Transparent")

guiModify( "LinePlot", Name = "Kalman$10$1",
   LineColor = "User1",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$10$3",
   LineStyle = "Med Dash",
   LineColor = "Black",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$10$2",
   LineStyle = "Med Dash",
   LineColor = "User1",
   LineWeight = "2")

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, Inflexp3,
   Inflexp3min, Inflexp3max",
   GraphSheet= "Kalman", Page="New")

guiModify( "LinePlot", Name = "Kalman$11$2",
   LineStyle = "Short Dash",
   LineColor = "User4",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$11$1",
   LineColor = "User1",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$11$3",
   LineStyle = "Short Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$11$3",
   LineStyle = "Solid")

guiModify( "LinePlot", Name = "Kalman$11$2",
   LineStyle = "Solid")

guiModify( "LinePlot", Name = "Kalman$11$1",
   LineWeight = "3")

guiCreate( "MainTitle", Name = "Kalman$11$1",
   Title = "Response to expl. variable 3 with 2-sigma conf. limits",
   FillColor = "Transparent")

}
```

```
#
#
##############################################################################
#
if(sum(names(writeall) == "increment") == 1) {

guiPlot( PlotType = "Line", DataSet = "writeall", Columns = "time, increment,
   Incmin, Incmax,zero",
  GraphSheet= "Kalman", Page= "New")

guiModify( "LinePlot", Name = "Kalman$12$1",
   LineColor = "User4",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$12$2",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")

guiModify( "LinePlot", Name = "Kalman$12$3",
   LineStyle = "Med Dash",
   LineColor = "User8",
   LineWeight = "2")

guiCreate( "MainTitle", Name = "Kalman$12$1",
   Title = "Differenced trend estimate with 2-sigma confidence limits",
   FillColor = "Transparent")

guiModify( "LinePlot", Name = "Kalman$12$4",
   LineColor = "User1",
   LineWeight = "2")

}
```

Appendix B File 'uitvoer' for temp. 1901-2002

```
GEBRUIKTE BESTANDEN:
 INPUT DATA          : Temperature.dat
 INPUT OPTIONS        : TemperatureT.opt

 TrendSpotter IS COMPILED ON PC (SALFORD)


TrendSpotter PC version 1.3, RIVM, November 2002


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----------------------------------------------------------------------------
!            Trend temperature De Bilt                                      !
-----------------------------------------------------------------------------
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


-------------------------------------------
!    KALMAN-FILTER   (C) KEMA 1989 LMO    !
-------------------------------------------
!     TREND-ESTIMATION MODEL : DD        !
!     NUMBER OF MEASUREMENTS :102        !
!     # TUNE-IN STEPS        : 20        !
!     LENGTH OF ONE CYCLE    :  0        !
!     EXPLANATORY VARIABLES  :  0        !
!     # AR-PARAMETERS        :  0        !
!     # MA-PARAMETERS        :  0        !
!     # INTEGRATIONS         :  0        !
!     VARIABLE SELECTION     :  0        !
!     STEPWISE REGRESSION    : NO        !
!     Q-MATRIX OPTIMATION    : NO        !
!     OPTIMIZE TREND VAR.    :YES        !
!     OPTIMIZE ARMA-PARAM.   : NO        !
!     SMOOTH RUNNER          :YES        !
!     LN-TRANSFORMATION      : NO        !
!     STANDARDIZE DATA       : NO        !
!     FORECAST STEPS         :  0        !
-------------------------------------------


 STARTING VALUES OF Q-MATRIX (DIAGONAL) :
 0.00000 0.00000
 OVERVIEW T-MATRIX
  2.00       -1.00
  1.00       0.000E+00

 TREND VARIANCE OPTIMIZED


-------------------------------------------
!       STATE AFTER OPTIMALIZATION       !
-------------------------------------------
! FCN-EVALUATIONS             :  234      !
! SPECIFIED SIGNIFICANCE      :    2      !
! ESTIMATED SIGNIFICANCE IN Q :     7.14  !
! VALUE OF NSRCH              :    7      !
! VALUE LOGLC IN OPTIMUM      : -71.16336 !
! ERROR MESSAGE IER           :    0      !
-------------------------------------------
ESTIMATED VARIANCE (SCALE FACTOR) : 0.3635
OPTIMIZED Q-MATRIX :
0.9190E-04   0.0000E+00
```

HYPER PARAMETERS : 0.9190E-04

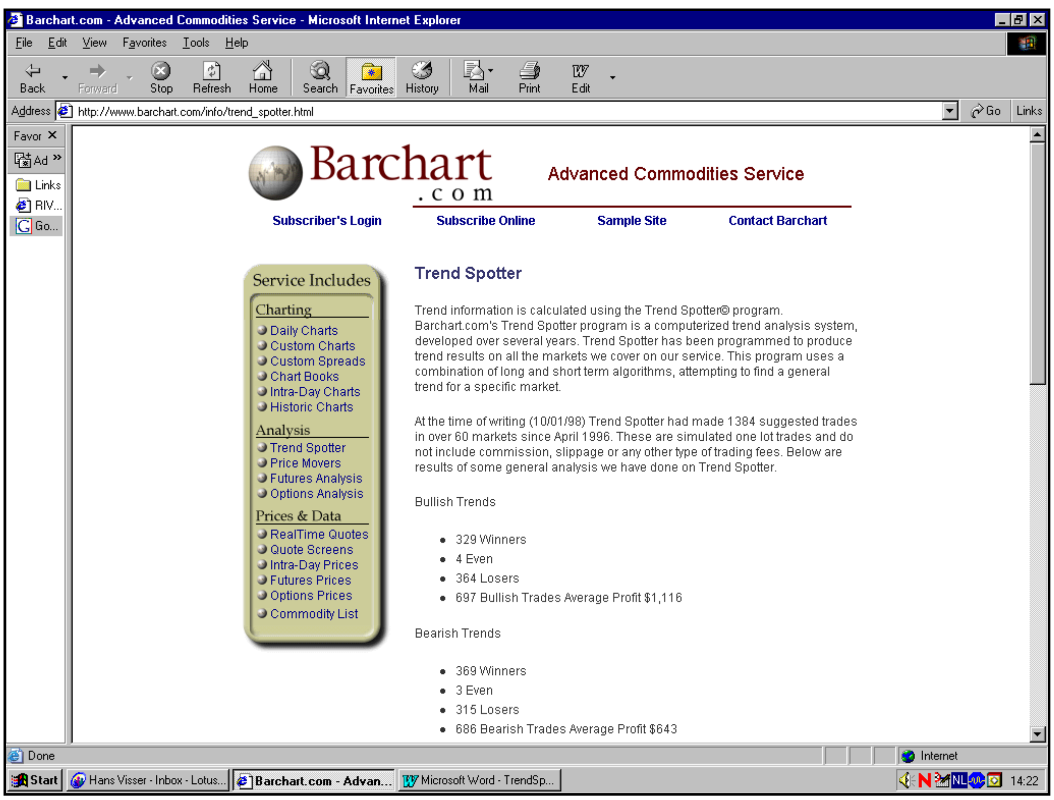
P(T/T)=

0.129 0.120
0.120 0.113

THE ESTIMATION OF SIGMA^2, BASED ON 82 INNOVATIONS =0.36354
LOG(L) AND THE CONCENTRATED LOG(LC)=-80.770 AND -71.163

RD-SQUARE : 0.20816 . Yardstick is the SL model.
MEAN OF PREDICTION ERRORS AFTER TRANSIENT PERIOD: 0.41968

NB: Values for RD**2 and mean of pr. errors are only reliable with no missing values!



TrendSpotter is just as the Trend Spotter program above, a "computerized trend analysis systeem". However, the area of application of both packages is quite different.

OVERVIEW TABLE

				STATEVECTOR (VECTOR ALFA(T))	
T	Y(T)	RESIDUAL	FT	MU(T)	BETA(T)
1901.00	8.91	-0.01	1.13	8.918	8.907
1902.00	8.30	-0.63	1.11	8.930	8.918
1903.00	9.27	0.32	1.10	8.942	8.930
1904.00	9.06	0.11	1.09	8.953	8.942
1905.00	8.82	-0.15	1.08	8.965	8.953
1906.00	9.30	0.32	1.07	8.976	8.965
1907.00	8.67	-0.32	1.06	8.987	8.976
1908.00	8.56	-0.44	1.05	8.999	8.987
1909.00	8.38	-0.63	1.05	9.010	8.999
1910.00	9.49	0.47	1.05	9.020	9.010
1911.00	9.91	0.88	1.04	9.031	9.020
1912.00	9.26	0.22	1.04	9.040	9.031
1913.00	9.67	0.62	1.04	9.050	9.040
1914.00	9.72	0.66	1.04	9.059	9.050
1915.00	8.83	-0.23	1.04	9.067	9.059
1916.00	9.25	0.17	1.04	9.075	9.067
1917.00	8.35	-0.73	1.04	9.084	9.075
1918.00	9.31	0.22	1.04	9.092	9.084
1919.00	8.13	-0.97	1.04	9.100	9.092
1920.00	9.32	0.22	1.04	9.108	9.100
1921.00	9.93	0.82	1.04	9.116	9.108
1922.00	8.30	-0.82	1.04	9.125	9.116
1923.00	8.74	-0.39	1.04	9.133	9.125
1924.00	8.56	-0.58	1.04	9.142	9.133
1925.00	9.15	0.00	1.04	9.151	9.142
1926.00	9.68	0.52	1.04	9.160	9.151
1927.00	8.93	-0.24	1.04	9.169	9.160
1928.00	9.32	0.15	1.04	9.178	9.169
1929.00	8.31	-0.88	1.04	9.187	9.178
1930.00	9.68	0.48	1.04	9.196	9.187
1931.00	8.78	-0.43	1.04	9.205	9.196
1932.00	9.38	0.17	1.04	9.213	9.205
1933.00	8.97	-0.25	1.04	9.221	9.213
1934.00	10.21	0.98	1.04	9.228	9.221
1935.00	9.67	0.43	1.04	9.235	9.228
1936.00	9.47	0.23	1.04	9.241	9.235
1937.00	9.51	0.26	1.04	9.246	9.241
1938.00	9.68	0.42	1.04	9.250	9.246
1939.00	9.62	0.36	1.04	9.254	9.250
1940.00	8.11	-1.15	1.04	9.257	9.254
1941.00	8.68	-0.59	1.04	9.260	9.257
1942.00	8.29	-0.97	1.04	9.263	9.260
1943.00	9.63	0.37	1.03	9.265	9.263
1944.00	9.47	0.20	1.03	9.266	9.265
1945.00	9.98	0.72	1.03	9.267	9.266
1946.00	9.10	-0.17	1.03	9.267	9.267
1947.00	9.50	0.23	1.03	9.267	9.267
1948.00	9.86	0.59	1.03	9.265	9.267
1949.00	10.03	0.76	1.03	9.263	9.265
1950.00	9.38	0.11	1.03	9.260	9.263
1951.00	9.78	0.52	1.03	9.257	9.260
1952.00	9.01	-0.24	1.03	9.253	9.257
1953.00	9.91	0.66	1.03	9.248	9.253
1954.00	9.10	-0.14	1.03	9.244	9.248
1955.00	8.97	-0.27	1.03	9.240	9.244
1956.00	8.18	-1.06	1.03	9.236	9.240
1957.00	9.72	0.49	1.03	9.232	9.236
1958.00	9.16	-0.07	1.03	9.229	9.232
1959.00	9.94	0.71	1.03	9.227	9.229
1960.00	9.55	0.32	1.03	9.226	9.227
1961.00	9.75	0.52	1.04	9.225	9.226
1962.00	8.11	-1.12	1.04	9.226	9.225

1963.00	7.84	-1.39	1.04	9.228	9.226
1964.00	8.88	-0.36	1.04	9.232	9.228
1965.00	8.50	-0.74	1.04	9.238	9.232
1966.00	9.39	0.15	1.04	9.245	9.238
1967.00	9.88	0.62	1.04	9.254	9.245
1968.00	9.11	-0.16	1.04	9.265	9.254
1969.00	9.52	0.24	1.04	9.277	9.265
1970.00	9.05	-0.24	1.04	9.290	9.277
1971.00	9.47	0.17	1.04	9.305	9.290
1972.00	8.84	-0.48	1.04	9.322	9.305
1973.00	9.28	-0.07	1.04	9.340	9.322
1974.00	9.68	0.32	1.04	9.360	9.340
1975.00	9.82	0.44	1.04	9.382	9.360
1976.00	9.98	0.58	1.04	9.404	9.382
1977.00	9.83	0.40	1.04	9.429	9.404
1978.00	9.01	-0.45	1.04	9.455	9.429
1979.00	8.51	-0.97	1.04	9.482	9.455
1980.00	9.25	-0.26	1.04	9.512	9.482
1981.00	9.21	-0.33	1.04	9.543	9.512
1982.00	10.05	0.47	1.04	9.575	9.543
1983.00	10.08	0.47	1.04	9.610	9.575
1984.00	9.45	-0.20	1.04	9.646	9.610
1985.00	8.52	-1.17	1.04	9.684	9.646
1986.00	8.97	-0.75	1.04	9.723	9.684
1987.00	8.85	-0.91	1.04	9.764	9.723
1988.00	10.34	0.54	1.04	9.806	9.764
1989.00	10.74	0.89	1.04	9.850	9.806
1990.00	10.88	0.98	1.04	9.894	9.850
1991.00	9.48	-0.46	1.04	9.939	9.894
1992.00	10.53	0.55	1.04	9.985	9.939
1993.00	9.61	-0.42	1.05	10.031	9.985
1994.00	10.59	0.51	1.05	10.078	10.031
1995.00	10.44	0.32	1.05	10.126	10.078
1996.00	8.57	-1.60	1.06	10.174	10.126
1997.00	10.27	0.04	1.07	10.222	10.174
1998.00	10.40	0.13	1.08	10.271	10.222
1999.00	10.89	0.57	1.09	10.320	10.271
2000.00	10.88	0.51	1.10	10.370	10.320
2001.00	10.35	-0.07	1.11	10.419	10.370
2002.00	10.80	0.33	1.13	10.469	10.419

TrendSpotter PC version 1.3, RIVM, November 2002

VAR. OF DATA AROUND TREND:0.33398

ESTIMATION RESULTS

time	measured	model	trend	inc mu(t)
1901.00	8.908	8.918	8.918	0.000
1902.00	8.300	8.930	8.930	0.012
1903.00	9.267	8.942	8.942	0.012
1904.00	9.058	8.953	8.953	0.012
1905.00	8.817	8.965	8.965	0.012
1906.00	9.300	8.976	8.976	0.011
1907.00	8.667	8.987	8.987	0.011
1908.00	8.558	8.999	8.999	0.011
1909.00	8.383	9.010	9.010	0.011
1910.00	9.492	9.020	9.020	0.011
1911.00	9.908	9.031	9.031	0.010
1912.00	9.258	9.040	9.040	0.010
1913.00	9.667	9.050	9.050	0.009
1914.00	9.717	9.059	9.059	0.009
1915.00	8.833	9.067	9.067	0.009
1916.00	9.250	9.075	9.075	0.008
1917.00	8.350	9.084	9.084	0.008
1918.00	9.308	9.092	9.092	0.008
1919.00	8.133	9.100	9.100	0.008
1920.00	9.325	9.108	9.108	0.008
1921.00	9.933	9.116	9.116	0.008
1922.00	8.300	9.125	9.125	0.008
1923.00	8.742	9.133	9.133	0.009
1924.00	8.558	9.142	9.142	0.009
1925.00	9.150	9.151	9.151	0.009
1926.00	9.683	9.160	9.160	0.009
1927.00	8.925	9.169	9.169	0.009
1928.00	9.325	9.178	9.178	0.009
1929.00	8.308	9.187	9.187	0.009
1930.00	9.675	9.196	9.196	0.009
1931.00	8.775	9.205	9.205	0.009
1932.00	9.383	9.213	9.213	0.008
1933.00	8.975	9.221	9.221	0.008
1934.00	10.208	9.228	9.228	0.007
1935.00	9.667	9.235	9.235	0.007
1936.00	9.475	9.241	9.241	0.006
1937.00	9.508	9.246	9.246	0.005
1938.00	9.675	9.250	9.250	0.004
1939.00	9.617	9.254	9.254	0.004
1940.00	8.108	9.257	9.257	0.003
1941.00	8.675	9.260	9.260	0.003
1942.00	8.292	9.263	9.263	0.002
1943.00	9.633	9.265	9.265	0.002
1944.00	9.467	9.266	9.266	0.001
1945.00	9.983	9.267	9.267	0.001
1946.00	9.100	9.267	9.267	0.000
1947.00	9.500	9.267	9.267	-0.001
1948.00	9.858	9.265	9.265	-0.001
1949.00	10.025	9.263	9.263	-0.002
1950.00	9.375	9.260	9.260	-0.003
1951.00	9.775	9.257	9.257	-0.003
1952.00	9.008	9.253	9.253	-0.004
1953.00	9.908	9.248	9.248	-0.004
1954.00	9.100	9.244	9.244	-0.004
1955.00	8.967	9.240	9.240	-0.004
1956.00	8.175	9.236	9.236	-0.004
1957.00	9.725	9.232	9.232	-0.003
1958.00	9.158	9.229	9.229	-0.003
1959.00	9.942	9.227	9.227	-0.002
1960.00	9.550	9.226	9.226	-0.001
1961.00	9.750	9.225	9.225	0.000
1962.00	8.108	9.226	9.226	0.001
1963.00	7.842	9.228	9.228	0.002

1964.00	8.875	9.232	9.232	0.004
1965.00	8.500	9.238	9.238	0.006
1966.00	9.392	9.245	9.245	0.007
1967.00	9.875	9.254	9.254	0.009
1968.00	9.108	9.265	9.265	0.010
1969.00	9.517	9.277	9.277	0.012
1970.00	9.050	9.290	9.290	0.014
1971.00	9.475	9.305	9.305	0.015
1972.00	8.842	9.322	9.322	0.017
1973.00	9.275	9.340	9.340	0.018
1974.00	9.683	9.360	9.360	0.020
1975.00	9.817	9.382	9.382	0.021
1976.00	9.983	9.404	9.404	0.023
1977.00	9.833	9.429	9.429	0.024
1978.00	9.008	9.455	9.455	0.026
1979.00	8.508	9.482	9.482	0.028
1980.00	9.250	9.512	9.512	0.029
1981.00	9.208	9.543	9.543	0.031
1982.00	10.050	9.575	9.575	0.033
1983.00	10.083	9.610	9.610	0.034
1984.00	9.450	9.646	9.646	0.036
1985.00	8.517	9.684	9.684	0.038
1986.00	8.975	9.723	9.723	0.039
1987.00	8.850	9.764	9.764	0.041
1988.00	10.342	9.806	9.806	0.042
1989.00	10.742	9.850	9.850	0.043
1990.00	10.875	9.894	9.894	0.044
1991.00	9.483	9.939	9.939	0.045
1992.00	10.533	9.985	9.985	0.046
1993.00	9.608	10.031	10.031	0.046
1994.00	10.592	10.078	10.078	0.047
1995.00	10.442	10.126	10.126	0.048
1996.00	8.575	10.174	10.174	0.048
1997.00	10.267	10.222	10.222	0.049
1998.00	10.400	10.271	10.271	0.049
1999.00	10.892	10.320	10.320	0.049
2000.00	10.883	10.370	10.370	0.049
2001.00	10.350	10.419	10.419	0.049
2002.00	10.800	10.469	10.469	0.049

TABEL OF STANDARD DEVIATIONS
STANDARD DEVIATIONS

T	TREND	INC MU(T)
1901.00	0.2165	-
1902.00	0.2021	0.0212
1903.00	0.1889	0.0204
1904.00	0.1768	0.0196
1905.00	0.1660	0.0188
1906.00	0.1565	0.0181
1907.00	0.1482	0.0173
1908.00	0.1410	0.0166
1909.00	0.1350	0.0159
1910.00	0.1300	0.0153
1911.00	0.1260	0.0147
1912.00	0.1228	0.0142
1913.00	0.1204	0.0137
1914.00	0.1185	0.0132
1915.00	0.1172	0.0128
1916.00	0.1162	0.0125
1917.00	0.1155	0.0122
1918.00	0.1151	0.0120
1919.00	0.1149	0.0118
1920.00	0.1147	0.0116
1921.00	0.1147	0.0115
1922.00	0.1146	0.0113
1923.00	0.1146	0.0113
1924.00	0.1146	0.0112
1925.00	0.1146	0.0111
1926.00	0.1146	0.0111
1927.00	0.1146	0.0111
1928.00	0.1145	0.0111
1929.00	0.1144	0.0110
1930.00	0.1143	0.0110
1931.00	0.1142	0.0110
1932.00	0.1141	0.0110
1933.00	0.1140	0.0110
1934.00	0.1138	0.0110
1935.00	0.1137	0.0110
1936.00	0.1135	0.0110
1937.00	0.1134	0.0110
1938.00	0.1133	0.0110
1939.00	0.1132	0.0110
1940.00	0.1131	0.0110
1941.00	0.1129	0.0110
1942.00	0.1129	0.0110
1943.00	0.1128	0.0110
1944.00	0.1127	0.0110
1945.00	0.1126	0.0110
1946.00	0.1126	0.0110
1947.00	0.1125	0.0110
1948.00	0.1125	0.0110
1949.00	0.1125	0.0110
1950.00	0.1124	0.0110
1951.00	0.1124	0.0110
1952.00	0.1124	0.0110
1953.00	0.1124	0.0110
1954.00	0.1125	0.0110
1955.00	0.1125	0.0110
1956.00	0.1125	0.0110
1957.00	0.1126	0.0110
1958.00	0.1126	0.0110
1959.00	0.1127	0.0110
1960.00	0.1128	0.0110
1961.00	0.1129	0.0110
1962.00	0.1130	0.0110

1963.00	0.1131	0.0110
1964.00	0.1132	0.0110
1965.00	0.1133	0.0110
1966.00	0.1134	0.0110
1967.00	0.1136	0.0110
1968.00	0.1137	0.0110
1969.00	0.1138	0.0110
1970.00	0.1140	0.0110
1971.00	0.1141	0.0110
1972.00	0.1142	0.0110
1973.00	0.1143	0.0110
1974.00	0.1144	0.0110
1975.00	0.1145	0.0110
1976.00	0.1146	0.0111
1977.00	0.1146	0.0111
1978.00	0.1146	0.0111
1979.00	0.1146	0.0111
1980.00	0.1146	0.0112
1981.00	0.1146	0.0113
1982.00	0.1147	0.0113
1983.00	0.1147	0.0115
1984.00	0.1149	0.0116
1985.00	0.1151	0.0118
1986.00	0.1156	0.0120
1987.00	0.1162	0.0122
1988.00	0.1172	0.0125
1989.00	0.1185	0.0129
1990.00	0.1204	0.0132
1991.00	0.1229	0.0137
1992.00	0.1261	0.0142
1993.00	0.1301	0.0147
1994.00	0.1351	0.0153
1995.00	0.1412	0.0159
1996.00	0.1483	0.0166
1997.00	0.1567	0.0173
1998.00	0.1663	0.0181
1999.00	0.1771	0.0188
2000.00	0.1891	0.0196
2001.00	0.2024	0.0204
2002.00	0.2169	0.0212

TrendSpotter PC version 1.3, RIVM, November 2002

Appendix C File ‘writeall.out’ for temp. 1901-2002

GEBRUIKTE BESTANDEN:
INPUT DATA : Temperature.dat
INPUT OPTIONS : TemperatureT.opt

time	measured	model	residual	stinnov	trend	sdtrend	mutNN	SDmutNN	increment	sdinc
1901.000	8.908	8.918	-0.010	-1.000	8.918	0.217	1.550	0.306	0.0000	0.1379
1902.000	8.300	8.930	-0.630	-1.000	8.930	0.202	1.538	0.296	0.0117	0.0212
1903.000	9.267	8.942	0.325	-1.000	8.942	0.189	1.527	0.287	0.0117	0.0204
1904.000	9.058	8.953	0.105	-1.000	8.953	0.177	1.515	0.280	0.0116	0.0196
1905.000	8.817	8.965	-0.148	-1.000	8.965	0.166	1.504	0.273	0.0115	0.0188
1906.000	9.300	8.976	0.324	-1.000	8.976	0.157	1.492	0.267	0.0114	0.0181
1907.000	8.667	8.987	-0.321	-1.000	8.987	0.148	1.481	0.262	0.0113	0.0173
1908.000	8.558	8.999	-0.440	-1.000	8.999	0.141	1.470	0.258	0.0111	0.0166
1909.000	8.383	9.010	-0.626	-1.000	9.010	0.135	1.459	0.255	0.0109	0.0159
1910.000	9.492	9.020	0.471	-1.000	9.020	0.130	1.448	0.253	0.0107	0.0153
1911.000	9.908	9.031	0.878	-1.000	9.031	0.126	1.438	0.250	0.0103	0.0147
1912.000	9.258	9.040	0.218	-1.000	9.040	0.123	1.428	0.249	0.0098	0.0142
1913.000	9.667	9.050	0.617	-1.000	9.050	0.120	1.419	0.248	0.0094	0.0137
1914.000	9.717	9.059	0.658	-1.000	9.059	0.119	1.410	0.247	0.0089	0.0132
1915.000	8.833	9.067	-0.234	-1.000	9.067	0.117	1.401	0.246	0.0085	0.0128
1916.000	9.250	9.075	0.175	-1.000	9.075	0.116	1.393	0.246	0.0083	0.0125
1917.000	8.350	9.084	-0.734	-1.000	9.084	0.116	1.385	0.245	0.0081	0.0122
1918.000	9.308	9.092	0.217	-1.000	9.092	0.115	1.377	0.245	0.0081	0.0120
1919.000	8.133	9.100	-0.966	-1.000	9.100	0.115	1.369	0.245	0.0081	0.0118
1920.000	9.325	9.108	0.217	-1.000	9.108	0.115	1.361	0.245	0.0082	0.0116
1921.000	9.933	9.116	0.817	1.224	9.116	0.115	1.352	0.245	0.0083	0.0115
1922.000	8.300	9.125	-0.825	-1.484	9.125	0.115	1.344	0.245	0.0084	0.0113
1923.000	8.742	9.133	-0.392	-0.568	9.133	0.115	1.335	0.245	0.0086	0.0113
1924.000	8.558	9.142	-0.584	-0.752	9.142	0.115	1.326	0.245	0.0088	0.0112
1925.000	9.150	9.151	-0.001	0.278	9.151	0.115	1.317	0.245	0.0090	0.0111
1926.000	9.683	9.160	0.523	1.050	9.160	0.115	1.308	0.245	0.0091	0.0111
1927.000	8.925	9.169	-0.244	-0.277	9.169	0.115	1.299	0.245	0.0091	0.0111
1928.000	9.325	9.178	0.147	0.372	9.178	0.115	1.290	0.245	0.0091	0.0111
1929.000	8.308	9.187	-0.879	-1.249	9.187	0.114	1.281	0.245	0.0090	0.0110
1930.000	9.675	9.196	0.479	1.034	9.196	0.114	1.272	0.245	0.0088	0.0110
1931.000	8.775	9.205	-0.430	-0.503	9.205	0.114	1.264	0.245	0.0086	0.0110
1932.000	9.383	9.213	0.170	0.503	9.213	0.114	1.256	0.245	0.0083	0.0110
1933.000	8.975	9.221	-0.246	-0.202	9.221	0.114	1.248	0.245	0.0078	0.0110
1934.000	10.208	9.228	0.980	1.723	9.228	0.114	1.241	0.245	0.0073	0.0110
1935.000	9.667	9.235	0.432	0.633	9.235	0.114	1.234	0.245	0.0066	0.0110
1936.000	9.475	9.241	0.234	0.226	9.241	0.114	1.228	0.245	0.0059	0.0110
1937.000	9.508	9.246	0.263	0.219	9.246	0.113	1.223	0.245	0.0052	0.0110
1938.000	9.675	9.250	0.425	0.418	9.250	0.113	1.218	0.245	0.0045	0.0110
1939.000	9.617	9.254	0.363	0.239	9.254	0.113	1.215	0.245	0.0039	0.0110
1940.000	8.108	9.257	-1.149	-2.160	9.257	0.113	1.211	0.246	0.0033	0.0110
1941.000	8.675	9.260	-0.585	-1.017	9.260	0.113	1.208	0.246	0.0029	0.0110
1942.000	8.292	9.263	-0.971	-1.484	9.263	0.113	1.206	0.246	0.0025	0.0110
1943.000	9.633	9.265	0.369	0.791	9.265	0.113	1.204	0.246	0.0021	0.0110
1944.000	9.467	9.266	0.200	0.429	9.266	0.113	1.202	0.247	0.0015	0.0110
1945.000	9.983	9.267	0.716	1.167	9.267	0.113	1.201	0.247	0.0008	0.0110
1946.000	9.100	9.267	-0.167	-0.366	9.267	0.113	1.201	0.247	0.0001	0.0110
1947.000	9.500	9.267	0.233	0.288	9.267	0.113	1.202	0.248	-0.0007	0.0110
1948.000	9.858	9.265	0.593	0.791	9.265	0.112	1.203	0.248	-0.0014	0.0110
1949.000	10.025	9.263	0.762	0.924	9.263	0.112	1.206	0.248	-0.0022	0.0110
1950.000	9.375	9.260	0.115	-0.231	9.260	0.112	1.208	0.249	-0.0029	0.0110
1951.000	9.775	9.257	0.518	0.390	9.257	0.112	1.212	0.249	-0.0035	0.0110
1952.000	9.008	9.253	-0.244	-0.879	9.253	0.112	1.216	0.250	-0.0040	0.0110
1953.000	9.908	9.248	0.660	0.604	9.248	0.112	1.220	0.250	-0.0042	0.0110
1954.000	9.100	9.244	-0.144	-0.754	9.244	0.112	1.225	0.251	-0.0044	0.0110
1955.000	8.967	9.240	-0.273	-0.885	9.240	0.112	1.229	0.252	-0.0043	0.0110
1956.000	8.175	9.236	-1.061	-2.010	9.236	0.113	1.233	0.252	-0.0040	0.0110
1957.000	9.725	9.232	0.493	0.653	9.232	0.113	1.236	0.253	-0.0035	0.0110
1958.000	9.158	9.229	-0.071	-0.311	9.229	0.113	1.239	0.253	-0.0029	0.0110
1959.000	9.942	9.227	0.715	0.942	9.227	0.113	1.241	0.254	-0.0022	0.0110

1960.000	9.550	9.226	0.324	0.206	9.226	0.113	1.243	0.254	-0.0014	0.0110
1961.000	9.750	9.225	0.525	0.479	9.225	0.113	1.243	0.255	-0.0004	0.0110
1962.000	8.108	9.226	-1.118	-2.137	9.226	0.113	1.242	0.255	0.0008	0.0110
1963.000	7.842	9.228	-1.387	-2.268	9.228	0.113	1.240	0.256	0.0023	0.0110
964.000	8.875	9.232	-0.357	-0.350	9.232	0.113	1.236	0.256	0.0039	0.0110
1965.000	8.500	9.238	-0.738	-0.856	9.238	0.113	1.231	0.257	0.0056	0.0110
1966.000	9.392	9.245	0.147	0.671	9.245	0.113	1.223	0.257	0.0073	0.0110
1967.000	9.875	9.254	0.621	1.362	9.254	0.114	1.214	0.257	0.0089	0.0110
1968.000	9.108	9.265	-0.156	0.018	9.265	0.114	1.204	0.257	0.0105	0.0110
1969.000	9.517	9.277	0.240	0.665	9.277	0.114	1.192	0.257	0.0120	0.0110
1970.000	9.050	9.290	-0.240	-0.131	9.290	0.114	1.178	0.257	0.0136	0.0110
1971.000	9.475	9.305	0.170	0.557	9.305	0.114	1.163	0.257	0.0152	0.0110
1972.000	8.842	9.322	-0.480	-0.487	9.322	0.114	1.146	0.256	0.0167	0.0110
1973.000	9.275	9.340	-0.065	0.259	9.340	0.114	1.128	0.256	0.0183	0.0110
1974.000	9.683	9.360	0.323	0.868	9.360	0.114	1.108	0.255	0.0198	0.0110
1975.000	9.817	9.382	0.435	0.964	9.382	0.115	1.087	0.254	0.0213	0.0110
1976.000	9.983	9.404	0.579	1.091	9.404	0.115	1.064	0.252	0.0228	0.0111
1977.000	9.833	9.429	0.405	0.702	9.429	0.115	1.040	0.251	0.0243	0.0111
1978.000	9.008	9.455	-0.446	-0.688	9.455	0.115	1.014	0.249	0.0259	0.0111
1979.000	8.508	9.482	-0.974	-1.389	9.482	0.115	0.986	0.246	0.0276	0.0111
1980.000	9.250	9.512	-0.262	-0.065	9.512	0.115	0.957	0.243	0.0294	0.0112
1981.000	9.208	9.543	-0.334	-0.124	9.543	0.115	0.926	0.240	0.0311	0.0113
1982.000	10.050	9.575	0.475	1.193	9.575	0.115	0.893	0.236	0.0328	0.0113
1983.000	10.083	9.610	0.473	1.077	9.610	0.115	0.859	0.232	0.0345	0.0115
1984.000	9.450	9.646	-0.196	-0.064	9.646	0.115	0.822	0.227	0.0361	0.0116
1985.000	8.517	9.684	-1.167	-1.522	9.684	0.115	0.785	0.222	0.0377	0.0118
1986.000	8.975	9.723	-0.748	-0.624	9.723	0.116	0.745	0.216	0.0394	0.0120
1987.000	8.850	9.764	-0.914	-0.739	9.764	0.116	0.704	0.209	0.0409	0.0122
1988.000	10.342	9.806	0.535	1.669	9.806	0.117	0.662	0.201	0.0423	0.0125
1989.000	10.742	9.850	0.892	2.061	9.850	0.119	0.619	0.193	0.0434	0.0129
1990.000	10.875	9.894	0.981	1.972	9.894	0.120	0.575	0.184	0.0443	0.0132
1991.000	9.483	9.939	-0.456	-0.484	9.939	0.123	0.530	0.174	0.0450	0.0137
1992.000	10.533	9.985	0.549	1.161	9.985	0.126	0.484	0.163	0.0457	0.0142
1993.000	9.608	10.031	-0.423	-0.473	10.031	0.130	0.437	0.152	0.0464	0.0147
1994.000	10.592	10.078	0.514	1.061	10.078	0.135	0.390	0.139	0.0470	0.0153
1995.000	10.442	10.126	0.316	0.633	10.126	0.141	0.343	0.125	0.0475	0.0159
1996.000	8.575	10.174	-1.599	-2.401	10.174	0.148	0.295	0.111	0.0480	0.0166
1997.000	10.267	10.222	0.044	0.485	10.222	0.157	0.246	0.095	0.0486	0.0173
1998.000	10.400	10.271	0.129	0.582	10.271	0.166	0.197	0.078	0.0490	0.0181
1999.000	10.892	10.320	0.571	1.216	10.320	0.177	0.148	0.060	0.0492	0.0188
2000.000	10.883	10.370	0.514	0.983	10.370	0.189	0.099	0.041	0.0493	0.0196
2001.000	10.350	10.419	-0.069	-0.041	10.419	0.202	0.049	0.021	0.0494	0.0204
2002.000	10.800	10.469	0.331	0.589	10.469	0.217	0.000	0.000	0.0494	0.0212

Appendix D THE KALMAN FILTER

D.1 PRELIMINARIES

The Kalman filter is designed to deal with state-space models. It is appropriate for a range of univariate and multivariate time-series models (Harvey, 1989). Although the formulation of the Kalman filter may be stated more generally, only the univariate case is given here. Thus, only scalar measurements y_t are considered. The description of the filter is not given in detail in the main text. Therefore, the Kalman filter will be given in this Appendix.

In literature on filter theory it is sometimes ambiguous which ensemble of equations defines the 'Kalman filter'. In this thesis the Kalman filter will be defined by the filtering, smoothing and post-sample prediction equations defined in Sections D.2, D.3 and D.4. Equations additional to the filter are given in Sections D.5 and D.6. The following notation is introduced.

The state-space model considered here consists of a *measurement equation*

$$y_t = c_t{}'\alpha_t + \xi_t \qquad \text{(D.1)}$$

and a *transition equation*

$$\alpha_t = T\,\alpha_{t-1} + \eta_t \qquad \text{(D.2)}$$

where the prime stands for transpose. Both equations suffice for all models mentioned.. It contains the M×1 state vector α_t, the *a priori* known M×M matrix T and M×1 vector c_t. The processes η_t and ξ_t are, respectively, an M×1 vector of disturbances and a scalar disturbance term which are distributed independently of each other. It is assumed that the noise processes ξ_t and η_t are distributed independently of each other. Each process is normally and independently distributed, i.e. $\xi_t \sim NID(0, \sigma^2 h_t)$ and $\eta_t \sim NID(0, \sigma^2 Q)$, with σ^2 a scalar and Q an M×M diagonal matrix. The inclusion of the constant term σ^2 in the variance of η_t is arbitrary, but the reason for formulating the model in this way will become apparent in Section D.5. Let the vector $a_{t/s}$ denote the minimum mean square estimator (MMSE) or 'optimal' estimator of α_t conditional upon all the information up to and including time s

$$a_{t/s} = E\left\{\alpha_t \,|\, y_1, \ldots, y_s\right\} \qquad \text{(D.3)}$$

The symbol $|$ stands for 'conditional upon'. The fact that $a_{t/s}$ is optimal means that the Euclidian distance between estimator $a_{t/s}$ and α_t is minimal, i.e. the conditional error variance

$$E\left\{\| a_{t/s} - \alpha_t \|^2 \quad | y_1, \dots, y_s \right\} \qquad \text{(D.4)}$$

is minimized over all possible estimators which are a function of the data $\{y_1, ..., y_s\}$. For more details, see Anderson and Moore (1979, Ch. 5). The mean square error (MSE) matrix, i.e. the covariance matrix of $a_{t/s} - \alpha_t$, is denoted by

$$\sigma^2 P_{t/s} = E\left\{(a_{t/s} - \alpha_t)(a_{t/s} - \alpha_t)' | y_1, \dots, y_s \right\} \qquad \text{(D.5)}$$

Also the covariance between two different state vectors $a_{s/N}$ and $a_{t/N}$ with $s \le t \le N$, may be of importance. This matrix is denoted by

$$\sigma^2 P_{s,t/N} = E\left\{(a_{s/N} - \alpha_s)(a_{t/N} - \alpha_t)' | y_1, \dots, y_N \right\} \qquad \text{(D.6)}$$

The case t = s is referred to as filtering, the case s > t as smoothing and s < t as prediction.

D.2 FILTERING

Given the estimates $a_{t-1/t-1}$ and $P_{t-1/t-1}$ at t-1, we have the following recursive formulae

$$a_{t/t-1} = T\, a_{t-1/t-1} \qquad \text{(D.7)}$$

$$P_{t/t-1} = T\; P_{t-1/t-1} T' + Q \qquad \text{(D.8)}$$

Equations (D.7) and (D.8) are known as the *prediction equations*. The one-step-ahead prediction error, or innovation, is defined as

$$n_{t/t-1} = y_t - c_t{}' a_{t/t-1} \qquad \text{(D.9)}$$

It can be shown that this quantity is normally and independently distributed, i.e. $n_{t/t-1} \sim NID(0, \sigma^2 f_{t/t-1})$. The term $f_{t/t-1}$ follows from

$$f_{t/t-1} = c_t{}' P_{t/t-1} c_t + h_t \qquad \text{(D.10)}$$

The standardized innovations

$$\tilde{n}_{t/t-1} = \frac{n_{t/t-1}}{\sqrt{\sigma^2 f_{t/t-1}}} \qquad \text{(D.11)}$$

follow from (D.9). They are independently distributed and have a standard normal distribution or $\tilde{n}_{t/t-1} \sim NID(0,1)$. Therefore, they may be used for diagnostic checking (Jarque and Bera 1980 and Harvey 1989, Ch. 5).

In terms of the innovations, the filtered quantities are obtained from (D.7) and (D.8) using the following filter equations:

$$a_{t/t} = a_{t/t-1} + P_{t/t-1} c_t \, n_{t/t-1} \, f^{-1}_{t/t-1} \qquad \text{(D.12)}$$

$$P_{t/t} = P_{t/t-1} - P_{t/t-1} c_t \, c_t' \, P_{t/t-1} \, f^{-1}_{t/t-1} \qquad \text{(D.13)}$$

Equations (D.12) and (D.13) are known as the *updating equations*.

To start the filter iterations, the values of $a_{0/0}$ and $P_{0/0}$ have to be known. In general, no prior information is available at this point. One method, followed in this thesis, is to start the filter taking $a_{0/0}$ arbitrary and $P_{0/0} = \kappa I$, with I the M×M identity matrix and κ a large but finite number. Now the filter will generate its own starting values in, say, N_s iterations. This method is similar to using a diffuse or noninformative prior within a Bayesian framework. Another method is described by Ansley and Kohn (1985), who have shown that the Kalman filter with a diffuse initial state may lead to numerical difficulties under specific conditions. Other methods of initializing the filter were proposed recently (De Jong 1991 and Bell and Hillmer 1991).

D.3 SMOOTHING

Once $a_{t/t}$ has been estimated for t = 1,....,N, these estimates can be smoothed by using the additional information contained in the data y_s, s > t. Three types of smoothers exist, viz. the optimal fixed-point smoother, the optimal fixed-lag smoother and the optimal fixed-interval smoother. Because all applications in this thesis are 'off-line', only the fixed-interval smoother $a_{t/N}$, with $t \leq N$ and N the sample size, is of interest. For a general overview of the different smoothers, see Anderson and Moore (1979, Ch. 7).

The fixed-interval smoother works backwards, according to

$$a_{t/N} = a_{t/t} + P_t^*(a_{t+1/N} - T\, a_{t/t}) \quad \text{(D.14)}$$

$$P_{t/N} = P_{t/t} + P_t^*(P_{t+1/N} - P_{t+1/t}) P_t^{*\prime} \quad \text{(D.15)}$$

$$P_t^* = P_{t/t}\, T'\, P_{t+1/t}^{-1} \quad \text{(D.16)}$$

with t = N-1,....,1. Unlike the filtered estimates, the smoothed quantities are less sensitive to transient phenomena, induced by inaccurate starting values of the filter scheme in the first N_s iterations. Thus, smoothed rather than filtered estimates can be used for practical applications if $t \le N_s$.

If the filter is initiated with the diffuse initial state and Q is set to zero, the Kalman filter estimates $a_{t/N}$ and $P_{t/N}$ become time-invariant and equal the ordinary least squares (OLS) estimates. The OLS fitting procedure is thus a special case of the Kalman filter (Otter 1978). The smoothed innovations or *residuals* are defined by

$$n_{t/N} = y_t - c_t'\, a_{t/N} \quad \text{(D.17)}$$

The residuals are normally distributed, i.e. $n_{t/N} \sim N(0, \sigma^2 f_{t/N})$, with $f_{t/N}$ given by

$$f_{t/N} = c_t'\, P_{t/N}\, c_t + h_t \quad \text{(D.18)}$$

These residuals are not necessarily stochastically independent.

D.4 POST-SAMPLE PREDICTION

If post-sample predictions have to be made L steps ahead, the prediction equations are used repeatedly without the updating equations. Thus, the MMSE of α_{N+L}, made at time t = N, is given by

$$\begin{aligned} a_{N+L/N} &= T\, a_{N+L-1/N} \\ &= T^L a_{N/N} \end{aligned} \quad \text{(D.19)}$$

Matrix $P_{N+L/N}$ is computed iteratively from

$$P_{N+i/N} = T\, P_{N+i-1/N}\, T' + Q\,,\ i = 1,....,L \quad \text{(D.20)}$$

The MMSE of the future observationy $_{N+L}$ is given by

$$\tilde{y}_{N+L} = c_{N-L}' \, a_{N+L/N} \tag{D.21}$$

The MSE of y_{N+L}, i.e. the variance of $\tilde{y}_{N+L}$ - y_{N+L}, follows from

$$MSE(\tilde{y}_{N+L}) = (c_{N'+L} P_{N+L/N} c_{N+L} + h_{N+L})\sigma^2 \tag{D.22}$$

D.5 MAXIMUM LIKELIHOOD ESTIMATION

The log-likelihood function of the filtering procedure is based on the normality of the noise processes ξ_t and η_t. It is given by means of the so-called *prediction-error decomposition*

$$\log(L) = -\frac{1}{2}\left[(N - N_s)\log(2\pi\sigma^2) + \sum_{N_s+1}^{N}\left(\log(f_{t/t-1}) + \frac{1}{\sigma^2}\frac{n_{t/t-1}^2}{f_{t/t-1}}\right)\right] \tag{D.23}$$

The unknown parameters σ^2 and Q can be estimated by maximizing this function. The matrix Q appears not explicitly in (D.23), but it influences $f_{t/t-1}$ through the relations (D.8) and (D.10).

The explicit dependence of log(L) on σ^2 yields the maximum likelihood (ML) estimator σ_{ML}^2:

$$\hat{\sigma}_{ML}^2 = \frac{1}{(N - N_s)}\sum_{t=N_s+1}^{N}\frac{n_{t/t-1}^2}{f_{t/t-1}} \tag{D.24}$$

Through substitution of this estimator into (D.23) and multiplication of the result by -2 we get the concentrated log-likelihood function L_c

$$\log(L_c) = \sum_{t=N_s+1}^{N}\log(\hat{\sigma}_{ML}^2 f_{t/t-1}) \tag{D.25}$$

Equation (D.25) shows that ML optimization leads to minimization of the sum of innovation variances. The ML estimate of Q is found by minimizing $\log(L_c)$. This involves the evaluation of the prediction and updating equations outlined in Section D.2 for a number of Q values.

D.6 TESTING OF SIGNIFICANCE

Significance testing may be performed on elements of the state vector α_t. Two tests are of practical importance. First, we may test if an element i of α_t equals a fixed level k. Second, we may test if a certain element i of α_t has evolved over a time interval (t - s). Now the null hypothesis $\alpha_t^i = \alpha_s^i$ can be tested against alternatives, in which $0 \le s \le t \le N$. Statistics will be derived for both situations.

Let $a_{t/N}^i$ denote element i of $a_{t/N}$ and let $P_{t/N}^{i,i}$ denote element (i,i) of $P_{t/N}$. Furthermore, it is assumed that variance σ^2 is estimated by maximum likelihood following equation (D.24). The null hypothesis $\alpha_t^i = k$ can be tested using the statistic

$$g_t = \frac{a^i_{t/N} - k}{\hat{\sigma}_{ML}\sqrt{P^{i,i}_{t/N}}} \qquad \text{(D.26)}$$

The statistic g_t is t-distributed with $N\text{-}N_s$ degrees of freedom under the null hypothesis. This result follows from rewriting (D.26):

$$g_t = \frac{a^i_{t/N} - k}{\sigma\sqrt{P^{i,i}_{t/N}}} \Bigg\rangle \sqrt{\frac{1}{(N - N_s)}\frac{(N - N_s)\,\hat{\sigma}^2_{ML}}{\sigma^2}} \quad \Bigg\langle \qquad \text{(D.27)}$$

The first term on the right-hand side has a standard normal distribution, while the quantity $(N\text{-}N_s)_{ML}{}^2/\sigma^2$ is χ^2-distributed with $N\text{-}N_s$ degrees of freedom:

$$\begin{aligned} \frac{(N - N_s)\,\hat{\sigma}^2_{ML}}{\sigma^2} &= \sum_{t=N_s+1}^{N} \frac{n^2_{t/t-1}}{\sigma^2 f_{t/t-1}} \\ &= \sum_{t=N_s+1}^{N} \tilde{n}^2_{t/t-1} \approx \chi^2(N - N_s) \end{aligned} \qquad \text{(D.28)}$$

Now g_t is t-distributed by definition.

The second test concerns the comparison of two levels $a_{s/N}^i$ and $a_{t/N}^i$ $(1 \le s \le t \le N)$. To test the null hypothesis $\alpha_s^i = \alpha_t^i$ against alternatives, the following statistic g_t may be appropriate

$$g_t = \frac{a^i_{t/N} - a^i_{s/N}}{\hat{\sigma}_{ML}\sqrt{P^{i,i}_{s/N} + P^{i,i}_{t/N} - 2\,P^{i,i}_{s,t/N}}} \qquad \text{(D.29)}$$

Again statistic g_t has a t-distribution under H_0 with $N\text{-}N_s$ degrees of freedom.

This result follows by rewriting (D.29) as

$$g_t = \frac{a^i_{t/N} - a^i_{s/N}}{\sqrt{\sigma^2\left(P^{i,i}_{s/N} + P^{i,i}_{t/N} - 2\,P^{i,i}_{s,t/N}\right)}} \Big/ \sqrt{\frac{1}{(N-N_s)}\frac{(N-N_s)\ \sigma^{\wedge 2}}{\sigma^2}} \qquad \text{(D.30)}$$

The MSE of ($a_{t/N}^i$ - $a_{s/N}^i$) is derived from

$$MSE\,(\,a_{t/N} - a_{s/N}\,) = \sigma^2\,(\,P^2_{t/N} + P^2_{s/N} - 2\,P^2_{s,t/N}\,) \qquad \text{(D.31)}$$

Therefore, the first term on the right-hand side of (D.30) has a standard normal distribution. The MSE matrix $\sigma^2 P_{s,t/N}$ is defined in (D.6). De Jong and Mackinnon (1988) show that $P_{s,t/N}$ is given by

$$P_{s,t/N} = P^*_s\,P_{s+1,t/N} \qquad \text{(D.32)}$$

In the case where s = t, we have $P_{t,t/N} = P_{t/N}$, $1 \le t \le N$. The matrix P_s^* is given by equation (D.16).

D.7 REFERENCES

Anderson, B.D.O. and J.B. Moore, 1979. Optimal Filtering. *Prentice-Hall, Inc. Englewood Cliffs, New Jersey*.

Bell, W. and S. Hillmer, 1991. Initializing the Kalman filter for nonstationary time series models. *J. of Time Series*, 12(4): 283-300.

De Jong, P. and M.J. Mackinnon, 1988. Covariances for smoothed estimates in state space models. *Biometrika*, 75(3): 601-602.

De Jong, P., 1991. The diffuse Kalman filter. *The Annals of Statistics*, 19(2): 1073- 1083.

Harvey, A.C., 1989. Forecasting, structural time series models and the Kalman filter. *Cambridge University Press, Cambridge*.

Jarque, C.M. and A.K. Bera, 1980. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6: 255-259.

Otter, P.W., 1978. The discrete Kalman filter applied to linear regression models: statistical considerations and an application. *Statistica Neerlandica*, 32: 41-56.