A Knowledge Database for System Identification

Larsson, Jan Eric; Persson, Per

1987

Document Version:
Publisher's PDF, also known as Version of record

CODEN: LUTFD2/(TFRT–7342)/1–80/(1987)

A Knowledge Database for System Identification

Jan Eric Larsson & Per Persson

Department of Automatic Control
Lund Institute of Technology
March 1987

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Technical report

Date of issue
March 1987

Document Number
CODEN: LUTFD2/(TFRT–7342)/1–80/(1987)

Author(s)
Jan Eric Larsson
Per Persson

Supervisor

Sponsoring organisation
The National Swedish Board for Technical Development, (STU), contract no. 85–3042

Title and subtitle
A Knowledge Database for System Identification

Abstract

This report presents an example of a script and a rule database for system identification, using the identification package Idpac together with the help system **(ihs)**. Two examples of terminal sessions with the system are also given.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
80

Recipient's notes

Security classification

A Knowledge Database for System Identification

Conv, plot, trend, nl, resid...

Department of Automatic Control
Lund Institute of Technology
Box 118
S–221 00 LUND
Sweden

Published 1987
Printed in Sweden

A Knowledge Database for System Identification

Jan Eric Larsson
Per Persson

Department of Automatic Control
Lund Institute of Technology
March 1987

Contents

1

Introduction

System identification demands both skill and experience, and the validity of the results strongly depends on the user's knowledge. There is a large theory that must be taken into account, and one will also have to learn to use some tools, i.e., computer programs. For this reason a help facility containing knowledge about system identification *and* a specific program is needed. This report describes the knowledge database of `(ihs)`, an expert system interface for the interactive identification package "Idpac." The expert system works as an interface to Idpac, as an intelligent front-end. The user's impression is that he runs with a help system that has knowledge not only of Idpac, but also of system identification.

The expert interface is totally *non-invasive*. This means that the user must ask for help if he wants it. If he does not, he will not notice that the help system is there. The help system is *goal related*, which means that it has knowledge about the methods used in system identification, and the different command sequences that should be used. It works with the *command spy* strategy, i.e., it uses the previous command history to give different help in different situations, and to guess what the next sensible command should be.

In order for the help system to work, it must have a knowledge database. This database consists of *scripts* and *rules*, and building it is a central part of the implementation of the system. This report is the main documentation of the knowledge database. The entire collection of scripts and rules is shown. Two example runs are also given. The data of the first example have been produced with the interactive simulation program Simnon. The data of the second problem was measured during an experiment with the cargo ship Atlantic Song.

The expert interface has been described in Larsson and Persson [1987 a]. The system is further documented and the source code given in Larsson and Persson [1987 b]. This report describes the full database. For readings on system identification, see Cox [1958], Eykhoff [1974, 1981], Fedorov [1972], Ljung and Söderström [1983], Åström and Eykhoff [1971], and Åström [1980]. Idpac and Simnon were developed at the Department of Automatic Control, Lund Institute of Technology, Wieslander [1979 a, b, c, 1980], and Åström [1983 a,b, 1985]. More about Idpac can be found in Gustavsson and Nilsson [1979] and Gustavsson [1979].

2

System Identification

This chapter will discuss the way of performing system identification which is implemented in the script listed in Chapter 3 and the rules listed in Chapter 4. Listings of sessions with Idpac and `(ihs)` using this script and these rules are found in Chapter 5 and 6.

The scripts contain procedural knowledge, i.e., knowledge of what commands to use, and sensible orders in which they may be used. The rule base contains diagnostic knowledge about identification, signal analysis, motivations and explanations for using different commands, comments, and help to be displayed in special situations.

The Script

The script takes care of several things which are necessary in doing a sensible identification of a system, given measured input and output signals.

It is assumed that the measured data is available in two separate ASCII text files. These files are first converted to Idpac's internal binary format, and the binary files are examined with the command **STAT**. This command displays some statistics of a data file, e.g., the mean value, the variance, and the length of the file. The files are then plotted to see if the signals are reasonable. Now it is possible for the user to change data points which are obviously wrong with the command **PLMAG**. The script guides the user through the subcommands of the command **PLMAG**.

When this is done, both files are plotted in the same diagram to get a feeling for the interaction between the signals. If the data files have been sampled with too high a frequency there is now a possibility to resample the signals internally in Idpac with the command **PICK**.

If the files are long enough they may be cut in two parts; one for identification and the other for cross validation. If the user intends to make a cross validation, it is assumed that he will use the first half of the signals for the first part of the cross validation and the second half for the second part of the cross validation. A natural choice is to cut the signals in two equal halves, but if the system is affected by some unknown disturbance it may be wise to cut out pieces of the signals which are not affected by it. After the files are cut their trends are removed with the command **TREND**, and after this the files are plotted again to see that the new signals look reasonable.

Before the actual parameter estimation the user has the option to carry out some tests on the signals. He may compute the coherence between the input and output signal, the autospectrum of the input signal, and display the result in a Bode plot. The autospectrum is computed to see if the input signal contains sufficient energy for excitation of the system. These computations are done in order to get an estimation of a frequency range in which the model will be reliable. It is recommended that the coherence between input and output signal should not be less than 0.7 in the frequency range in which a purely deterministic and linear model will be reliable. The user can also prewhite the input and output signals and compute the cross correlation of these to detect time delays and feedback loops. If a time

delay is detected it is possible to slide the signals relative to each other with the macro **SLID**, and then once again compute the prewhitened signals and the cross correlation.

After this, parameters of models of different orders are estimated and the residuals are examined. The expert system keeps information of the models' AIC values and continuously computes the minimum value and keeps track of the corresponding model.

A successful identification should give "white" residuals, i.e., residuals with zero autocorrelation except for $\tau = 0$. It can also be useful to calculate the cross correlation between the residuals and the input. A successful identification should give zero cross correlation for positive lags. The presence of feedback in the experiment is seen from correlation at negative lags. The actual residuals are also plotted in order to detect outliers.

It is also recommended to compute the frequency responses of the estimated models and display them in a Bode plot. When the curves coincide well in regions with high coherence, this is a sign that the order of the transfer function is sufficiently high. The script handles computation and plotting of frequency responses of models relating input signal to output signal and from noise to output signal.

When a sufficiently high model order, according to the AIC test and the Bode plots, has been found it is time for the cross validation. The cross validation consists of two parts. The first part starts by removing the trends of the complete input and output signal. After that the residuals of these signals are computed using the models estimated earlier and the loss function of the residual files is computed using parts of the residual files *not* used in the parameter estimation of the models. In the second part of the cross validation the second half of the signals are cut out, and trends are removed from these signals. The rule system suggests the use of the same order of trend polynomial as used on the first half. Then models of the same orders as in the first part of the cross validation are estimated on the second half. Residuals and loss functions are computed. The loss functions are computed using the last 80% of the residual files in order to avoid initial transients.

During the cross validation, the rule system asks the user for values of the loss function for the different models, and the minimum values and their corresponding models are stored as facts. Actually, the rule system also computes a minimum set, containing the minimum model and all models with loss functions less than $minimum\_value + 0.05 * |minimum\_value|$. Using these minimum sets the rule system chooses the best model. For details, see the rules!

After the parameter estimation, some tests are carried out on the model chosen. If a covariance matrix has been stored during the parameter estimation, it is possible to compute a random distribution set of models based on these statistical measures, and compute step responses from these models. A plot with a number of step responses is shown as a result of the macro **RANDSTEP**, and the user is asked if it looks reasonable.

Another possibility is to compute a random distribution of the frequency response of the different models, and plot them in a Bode diagram to see if they coincide. This is done with the macro **RANDTF**.

The third, and perhaps best, test is to compute the output signal from the model using the input signal and then plot it together with the real output signal, and see if they look reasonably similar.

At last the chosen model is listed and if the leading B-coefficients are very small the user can now fix them to zero and make a new parameter estimation. This part of the script could have been be much more elaborated.

Much of the motivations presented in this section is available in rule form, and these rules are trigged from the script during the session.

The Rules

Some rules are useful in all scripts. These might for example be rules for generating output, automatic documentation, etc. Therefore a list of global rules was introduced. These rules are added to all scripts at startup time.

Throughout the session with the system the user gets a lot of information from the expert system, and is also occasionally asked questions by the system. Questions are asked when the expert system needs information which is only displayed on the screen, or when it needs information which is a result of the users judgement, e.g., "In which region does the plot look reasonable?"

Rules that ask the user are placed in the beginning of the rule list, and facts triggering a question from the script always begin with the word **question**. The fact **(question enable)** must be present for the system to be allowed to ask the user. This fact is placed in the fact database when the command **think** is given, or when the user enters the beginner mode.

Rules that are trigged from the script are trigged by facts of the type **(note ...)** and **(suggest ...)**. Most of the rules are of this type, and their only action is to display a message on the screen.

Some rules are not trigged directly by a fact from the script, e.g., rules that compute the minimum values of the loss functions and the minimum sets.

As it is now the rule base contains only a small part of all knowledge needed in identification. A lot of rules must be added to take care of all special cases that may arise. E.g., currently the system assumes that the user wants do to a cross validation if he cuts the signals. However, it may be the case that he cuts them in such a way that it is impossible to carry out a parameter estimation on the remaining parts after the first **CUT**. Much more of this safety net remains to be implemented in the rules.

3

The Script

This chapter consists of a listing of the script used in chapter 5 and 6.

```
(setq scripts '(

  (ml
    (script ml
      ("ML"
       "The ML (Maximum Likelihood) estimation script computes and"
       "verifies a model of a transfer function that could have"
       "produced the output (given) from the input (also given).")

      (
       (kscall (system-list 1 ()))
       (kscall (bode-plot-list-ba ()))
       (kscall (bode-plot-list-ca ()))
       (kscall (resid-file-list ()))
       (kscall (xval-1-resid-file-list ()))
       (kscall (xval-2-resid-file-list ()))
       (kscall (question a priori knowledge))
;
; Convert the input signal from ASCII to binary format, and examine it.
;
       (kscall (suggest take care of insignal))
       (command conv (outfile INSIGNAL-T) (globfile INSIGNALDATA))
       (or
        ((command stat (infile INSIGNAL-T))
         (kscall (note interpret stat))
         (kscall (question length of (parameter INSIGNAL-T))))
        ((kscall (fact length (parameter INSIGNAL-T) unknown))))
       (command plot (infile INSIGNAL-T))

       (kscall (suggest modify outliers in insignal))
       (or ((scriptmacro plmag-macro (in INSIGNAL-T))) ())
;
; Convert the output signal from ASCII to binary format, and examine it.
;
       (kscall (suggest take care of outsignal))
       (command conv (outfile OUTSIGNAL-T) (globfile OUTSIGNALDATA))
       (or
        ((command stat (infile OUTSIGNAL-T))
         (kscall (note interpret stat))
         (kscall (question length of (parameter OUTSIGNAL-T))))
        ((kscall (fact length (parameter OUTSIGNAL-T) unknown))))
       (command plot (infile OUTSIGNAL-T))

       (kscall (suggest modify outliers in outsignal))
       (or ((scriptmacro plmag-macro (in OUTSIGNAL-T))) ())
;
; Plot the input signal and output signal in the same plot.
;
       (kscall (note select two interesting regions))
       (scriptmacro double-plot-macro (in INSIGNAL-T OUTSIGNAL-T)(out JUNK OF))
;
; If necessary sample the given signals before identification.
;
       (kscall (suggest sample in and outsignals
                 (parameter INSIGNAL-T) (parameter OUTSIGNAL-T)))
       (or
        ((repeat
          ((command pick (outfile INSIGNAL-TP) (infile INSIGNAL-T)
                         (number SAMP-NR1))
```

```
          (kscall (fact sampled (parameter INSIGNAL-TP)
                        (parameter INSIGNAL-T) (parameter SAMP-NR1)))
          (command pick (outfile OUTSIGNAL-TP) (infile OUTSIGNAL-T)
                        (number SAMP-NR2))
          (kscall (fact sampled (parameter OUTSIGNAL-TP)
                        (parameter OUTSIGNAL-T) (parameter SAMP-NR2)))
          (kscall (note pick rates must be equal (parameter SAMP-NR1)
                                                 (parameter SAMP-NR2)))
          (or
           ()
           ((scriptmacro double-plot-macro
                  (in INSIGNAL-T OUTSIGNAL-T) (out JUNK OF)))))))
       ((assign INSIGNAL-TP INSIGNAL-T)
        (assign OUTSIGNAL-TP OUTSIGNAL-T)))
;
; Cut out one piece in the signals (if the series are long enough) for
; estimation.
;
      (kscall (note two interesting regions))
      (or
       ((kscall (note cut insignal of first half (parameter INSIGNAL-TP)))
        (command cut
         (outfile INSI-1-T) (infile INSIGNAL-TP) (number F1) (number L1))
        (kscall (fact cut before estimation))
        (kscall (fact cut-from 1 (parameter INSI-1-T) (parameter INSIGNAL-TP)
                      (parameter F1) (parameter L1)))
        (kscall (note cut outsignal of first half (parameter OUTSIGNAL-TP)))
        (kscall (note earlier cut (parameter INSI-1-T)))
        (command cut
         (outfile OUTSI-1-T) (infile OUTSIGNAL-TP) (number F2) (number L2))
        (kscall (fact cut-from 1 (parameter OUTSI-1-T)(parameter OUTSIGNAL-TP)
                      (parameter F2) (parameter L2))))
       ((assign INSI-1-T INSIGNAL-TP)
        (assign OUTSI-1-T OUTSIGNAL-TP)))
;
; Remove trends from the cutted signals.
;
      (kscall (note remove trends from (parameter INSI-1-T)))
      (repeat
       ((command trend (outfile INSI-1-OK) (infile INSI-1-T) (number TI-1))
        (kscall (fact trends-removed (parameter INSI-1-OK)
                      (parameter INSI-1-T) (parameter TI-1)))
        (kscall (note check no trends in (parameter INSI-1-OK)))
        (command plot (infile INSI-1-OK))
        (kscall (suggest remove more trends (parameter TI-1)))))

      (kscall (note remove trends from (parameter OUTSI-1-T)))
      (repeat
       ((command trend (outfile OUTSI-1-OK) (infile OUTSI-1-T) (number TI-2))
        (kscall (fact trends-removed (parameter OUTSI-1-OK)
                      (parameter OUTSI-1-T) (parameter TI-2)))
        (kscall (note check no trends in (parameter OUTSI-1-OK)))
        (command plot (infile OUTSI-1-OK))
        (kscall  (suggest remove more trends (parameter TI-2)))))
;
; Compute the coherence and autospectrum.
; Prewhite the output and input and try to detect time delays.
; Slide the signals if necessary.
;
      (or
       ((repeat
         ((or
           ((kscall (note do coh))
            (command coh (outfile COHF) (infile INSI-1-OK) (infile OUTSI-1-OK))
            (kscall (question results of coherence test (parameter INSI-1-OK)
                                                        (parameter OUTSI-1-OK)))
            (kscall (note do aspec))
            (command aspec (outfile ASP) (infile INSI-1-OK))
            (command bode (infile ASP))
            (kscall (question results aspec insignal (parameter INSI-1-OK))))
           ((kscall (note do ccoff))
            (command prewhite (outfile PREWI) (outfile PREWO)
                              (infile INSI-1-OK) (infile OUTSI-1-OK))
            (command ccoff (outfile CCOFF) (infile PREWI) (infile PREWO))
            (kscall (question number of steps to slide (parameter INSI-1-OK)
                                                       (parameter OUTSI-1-OK))))
           ((kscall (note slide correct number of steps (parameter INSI-1-OK)
```

```
                                                    (parameter OUTSI-1-OK)))
            (repeat
             ((command slid (outfile INSI-1) (outfile OUTSI-1)
                            (infile INSI-1-OK) (infile OUTSI-1-OK))
              (command prewhite (outfile PREWI) (outfile PREWO)
                                (infile INSI-1) (infile OUTSI-1))
              (command ccoff (outfile CCOFF) (infile PREWI) (infile PREWO))
              (kscall (question number of steps to slide (parameter INSI-1)
                                                         (parameter OUTSI-1)))
             (kscall (note slide correct number of steps (parameter INSI-1-OK)
                                                  (parameter OUTSI-1-OK))))))))
       ((assign INSI-1 INSI-1-OK)
        (assign OUTSI-1 OUTSI-1-OK)))
;
; Estimate the parameters, look at the residuals, compute transfer function
; and plot it in a Bode diagram.
;
      (kscall (note start parameter estimation))
      (repeat
       ((command mlid
         (outfile SYST) (infile INSI-1) (infile OUTSI-1) (number ORDER))
        (kscall
         (fact identified-from 1 (parameter SYST) (parameter OUTSI-1)
               (parameter INSI-1) (parameter ORDER)))
        (kscall (question ml identification result (parameter SYST)))
        (kscall (note analyze model (parameter SYST)))

        (or
         ((command residu (outfile RES) (infile SYST) (infile INSI-1)
                                                      (infile OUTSI-1))
          (kscall (fact resfile (parameter RES) (parameter SYST)))
          (kscall (note white residuals))
          (or ((command page)) ())
          (command kill)
          (command plot (infile RES))
          (kscall (note interpret resid plot)))
         ())

        (or
         ((repeat
           ((or
             ((scriptmacro plmag-macro (in INSI-1)))
             ((scriptmacro plmag-macro (in OUTSI-1)))))))
         ((kscall (note transferfunction B / A (parameter SYST)))
          (command sptrf (outfile TRF-1) (infile SYST))
          (kscall (fact frequency-response (parameter TRF-1)
                                           (parameter SYST)))
          (kscall (note interpret bode plot ba))
          (kscall (note coherence test (parameter INSI-1)
                                       (parameter OUTSI-1)))
          (command bode (infile TRF-1))

          (or
           ()
           ((kscall (note transferfunction C / A (parameter SYST)))
            (command sptrf (outfile TRF-2) (infile SYST))
            (kscall (fact noise-response (parameter TRF-2) (parameter SYST)))
            (kscall (note interpret bode plot ca))
            (kscall (note coherence test (parameter INSI-1)
                                         (parameter OUTSI-1)))
            (command bode (infile TRF-2)))))
         ())
        (kscall (suggest cross validation or mlid))))
;
; If the signals have been cutted, compute the residuals of the second
; half with a model estimated from the first half. Compute the loss
; function from the residuals of the second half.
;
      (or
       ((kscall (note cross validation))
        (kscall (note remove trends for cross validation))
        (repeat
         ((kscall (note remove trends for xval-1 (parameter INSIGNAL-TP)))
          (command trend (outfile INSIXV) (infile INSIGNAL-TP))
          (kscall (note check no trends in (parameter INSIXV)))
          (command plot (infile INSIXV))))
        (repeat
```

```
         ((kscall (note remove trends for xval-1 (parameter OUTSIGNAL-TP)))
          (command trend (outfile OUTSIXV) (infile OUTSIGNAL-TP))
          (kscall (note check no trends in (parameter OUTSIXV)))
          (command plot (infile OUTSIXV))))
        (repeat
         ((kscall (note compute cross validation residuals))
          (command residu (outfile XVRES) (globfile SYST) (infile INSIXV)
                                                          (infile OUTSIXV))
          (kscall (fact xval-1-resfile (parameter XVRES) (parameter SYST)))
          (or () ((command page)))
          (command kill)
          (kscall (note cut out second part))
          (kscall (note earlier xval-1-cut (parameter OUTSI-1-T)))
          (command cut (outfile XVCUT) (infile XVRES) (number LO) (number N))
          (kscall (fact xval-1-cut (parameter XVCUT) (parameter XVRES)
                                   (parameter LO) (parameter N)))
          (command loss (infile XVCUT))
          (kscall (question loss for (parameter XVCUT) xval-1 file))))
;
; Fit models from the second halves of the signals, and compute residuals
; with the first halves.
;
        (kscall (note xval-2 may be performed))
        (kscall (note xval-2-cut (parameter INSIGNAL-TP)))
        (command cut (outfile INSI-2-T) (infile INSIGNAL-TP)
                                   (number L-1) (number N-1))
        (kscall (fact cut-from 2 (parameter INSI-2-T) (parameter INSIGNAL-TP)
                                 (parameter L-1) (parameter N-1)))
        (kscall (note xval-2-cut (parameter INSIGNAL-TP)))
        (command cut (outfile OUTSI-2-T) (infile OUTSIGNAL-TP)
                                   (number L-2) (number N-2))
        (kscall (fact cut-from 2 (parameter OUTSI-2-T)(parameter OUTSIGNAL-TP)
                                 (parameter L-2) (parameter N-2)))
        (kscall (note trend second half as first half (parameter INSI-1-T)))
        (command trend (outfile INSI-2-OK) (infile INSI-2-T))
        (kscall (note trend second half as first half
                     (parameter OUTSI-1-T)))
        (command trend (outfile OUTSI-2-OK) (infile OUTSI-2-T))
        (assign INSI-2 INSI-2-OK)
        (assign OUTSI-2 OUTSI-2-OK)
        (kscall (note earlier orders (parameter OUTSI-1)(parameter INSI-1)))
        (repeat
         ((kscall (note possible earlier orders))
          (command mlid (outfile SYS) (infile INSI-2) (infile OUTSI-2)
                        (number ORDER))
          (kscall (note xval-2 model (parameter SYS) (parameter ORDER)))
          (kscall (fact identified-from 2 (parameter SYS) (parameter OUTSI-2)
                                          (parameter INSI-2) (parameter ORDER)))
          (command residu (outfile RES-2) (infile SYS) (infile INSI-1)
                          (infile OUTSI-1))
          (kscall (fact xval-2-resfile (parameter RES-2) (parameter SYS)))
          (or () ((command page)))
          (command kill)
          (kscall (note cut for loss computations (parameter RES-2)
                    (parameter INSI-1-T)))
          (command cut (outfile RC-2) (infile RES-2) (number LO)
                                        (number N))
          (kscall (fact xval-2-cut (parameter RC-2) (parameter RES-2)
                                   (parameter LO) (parameter N)))
          (command loss (infile RC-2))
          (kscall (question loss for (parameter RC-2) xval-2 file))))
        (kscall (note interpret result table)))
       ((assign INSIXV INSIGNAL-TP)
        (assign OUTSIXV OUTSIGNAL-TP)))
;
; If the signals have been cutted, compute models from the second half of the
; signals and compute residuals from the first half. Compute the loss function
; of the residuals.
;
      (or
       ((kscall (note create step signal for monte carlo
                        (parameter INSIGNAL-TP)))
        (command insi (outfile STP))
        (command step)
        (command x)
        (repeat
         ((command randstep (outfile STEPS) (globfile SYS) (infile STP))
```

```
          (kscall (note step response distribution of best model))
          (command randtf (outfile TFS) (infile SYS))
          (kscall (note transferfunction distribution of best model))
          (kscall (note length for deter (parameter INSIGNAL-TP)))
          (command deter (outfile DSIM) (infile SYS) (infile INSIGNAL-TP))
          (kscall (note trend before deter plot
                    (parameter DSIM) (parameter OUTSIGNAL-TP)))
          (command trend (outfile TDSIM) (infile DSIM) (number NR1))
          (command trend (outfile TOS) (infile OUTSIGNAL-TP) (number NR2))
          (kscall (note plot simulated and real outsignal
                        (parameter TDSIM) (parameter TOS)))
          (or
           ((command plot (infile TDSIM) (symbol OF)))
           ((command plot (infile TOS) (symbol DF))))
          (kscall (note compare with deter (parameter TDSIM)
                                           (parameter TOS)))))))
;
; Try to fix parameters
;
      (or
       ()
       ((kscall (note try to fix a parameter))
        (repeat
         ((command list (globfile SYST))))
        (repeat*
         ((command mlidsc (outfile FIXSYS) (infile INSI-1) (infile OUTSI-1))
          (repeat
            ((command fix)))
          (command resume)
          (kscall (question fixed aic (parameter FIXSYS)))
          (command residu (outfile FIXRES) (infile FIXSYS) (infile INSI-1)
                                                           (infile OUTSI-1))
          (kscall (note white residuals))
          (or () ((command page)))
          (command kill)
          (kscall (note try to fix another parameter (parameter FIXSYS)))))))

      (kscall (note check consistency))
      (kscall (note common sense validation))
      (kscall (note choose model))

      (command stop) )))))
```

4

The Rules

This chapter consists of a listing of the rules used in chapter 5 and 6.

```
; This is the rule databases of the system.

(setq global-rules '(

  ((note double plot -y -z)
   -->
   (fact message ("Plot the files" -y "and" -z "with one PLOT command."
                  "GIVE THE COMMAND PLOT " -y "/" -z
                  ". There is not full support for the command PLOT,"
                  "so you will not be queried for all parameters.")))

  ((check other half -h1 -h2 -oh)
    test
    -oh
    (not (or (equal -h1 -oh) (equal -h2 -oh)))
    -->
   (fact message ("You should have plotted the two halves, wise guy!")))

  ((prepare-document)
    -->
    (fact document-text ()))

  ((document -x)
   (document-text -y)
    -->
    (setq -y (append1 -y -x))
    (fact document-text -y))

  ((write-document -filename)
   (document-text -text)
    -->
    (let
      ((tmp-prt (fileopen -filename "w")))
      (mapc
        '(lambda (x)
           (princ (list-to-string x) tmp-prt)
           (terpr tmp-prt))
        -text)
      (close tmp-prt)))

  ((patom -x)
   -->
   (patom -x) (terpr))

  ((message -x)
   -->
   (remove 1)
   (<- self ':save-yaps-info -x))

  ((note start plmag)
   -->
   (fact message ("If you want a strange block length, use BLOCK."
                  "In any case you must start with PLBEG.")))

  ((note kill plmag -file)
   -->
   (fact message ("You left the command PLMAG working on the file"
                  -file "with kill. You have not altered the file.")))
))
```

```
(setq rules '(

(ml

; ---ASK-------------------------------------------------------------------------

 ((question enable)
  (question a priori knowledge)
  -->
  (cond
    ((equal
      (<- uinterf ':prompt
        '("Do you have any a priori knowledge of the system") 'symbol) 'y)
     (fact a priori knowledge exists))))

 ((question enable)
  (a priori knowledge exists)
  -->
  (let ((lo) (hi))
    (setq lo
      (<- uinterf ':prompt
        '("What is your guess about the lower limit of model order") 'number))
    (setq hi
      (<- uinterf ':prompt
        '("What is your guess about the upper limit of model order") 'number))
    (fact ~(append1 '(guessed lower model order limit) lo))
    (fact ~(append1 '(guessed upper model order limit) hi))))

 ((question enable)
  (a priori knowledge exists)
  -->
  (cond
    ((equal
      (<- uinterf ':prompt
        '("Is the system identified in a feedback loop?") 'symbol) 'y)
     (fact identified in feedback))))

 ((question enable)
  (a priori knowledge exists)
  -->
  (cond
    ((equal
      (<- uinterf ':prompt
        '("Can the system be viewed as a linear system?") 'symbol) 'y)
     (fact system may be linear))))

 ((question enable)
  (a priori knowledge exists)
  -->
  (cond
    ((equal
      (<- uinterf ':prompt
        '("Is the purpose of the identification controller design")
        'symbol)
      'y)
     (fact identification for controller design))))

 ((question enable)
  (question results of coherence test -f1 -f2)
  -->
  (let ((x) (y))
    (setq x
      (<- uinterf ':prompt
          '("Lower limit of interval where the coherence is"
            "greater than ~0.7") 'number))
    (setq y
      (<- uinterf ':prompt
          '("Upper limit of interval where the coherence is"
            "greater than ~0.7") 'number))
    (fact fact coherence-limits -f1 -f2 ~x ~y)))

 ((question enable)
  (question results aspec insignal -f1)
  -->
  (let ((x) (y))
    (setq x
```

```
    (<- uinterf ':prompt
        '("Lower limit of interval where the spectral density is"
          "high") 'number))
  (setq y
    (<- uinterf ':prompt
        '("Upper limit of interval where the spectral density is"
          "high") 'number))
  (fact fact aspec-limits -f1 ^x ^y)))

((question enable)
 (question number of steps to slide -f1 -f2)
 -->
 (let ((steps))
   (setq steps
     (<- uinterf ':prompt
         '("What is the x-coordinate for the zero-crossing")
         'number))
   (fact fact slided -f1 -f2 ^steps)))

((~ (question enable))
 (question ml identification result -file)
 -->
 (let ((filename (concat -file ".t")) (lossfunc) (aic))
   (cond
     ((probef filename)
       (setq file (infile filename))
       (do
         ((expr (ratom file) (ratom file)))
         ((equal expr 'AIC)
           (setq aic (ratom file))
           (close file))
         (cond
           ((and
             (equal expr 'LOSS)
             (equal (ratom file) 'FUNCTION))
             (setq lossfunc (ratom file))))
         (do
           ()
           ((and (> (tyipeek file) 64) (< (tyipeek file) 91)))
           (tyi file)))
       (fact fact mlid-data -file ^aic ^lossfunc)))))

((question enable)
 (question ml identification result -file)
 -->
 (let ((filename (concat -file ".t")) (lossfunc) (aic))
   (cond
     ((probef filename)
       (setq file (infile filename))
       (do
         ((expr (ratom file) (ratom file)))
         ((equal expr 'AIC)
           (setq aic (ratom file))
           (close file))
         (cond
           ((and
             (equal expr 'LOSS)
             (equal (ratom file) 'FUNCTION))
             (setq lossfunc (ratom file))))
         (do
           ()
           ((and (> (tyipeek file) 64) (< (tyipeek file) 91)))
           (tyi file))))
     (t
       (setq aic (<- uinterf ':prompt '("What is the AIC") 'number))
       (setq lossfunc nil)))
  (fact fact mlid-data -file ^aic ^lossfunc)))

((question enable)
 (question loss function -syst)
 -->
 (let ((loss-func))
   (setq loss-func
     (<- uinterf ':prompt
       '("What is the value of the loss function") 'number))
   (fact fact xval-1-loss-func -syst ^loss-func)))
```

```
  ((question enable)
   (question length of -fil)
   -->
   (let ((x))
     (setq x
       (<- uinterf ':prompt
         '("What is the the length of the file" ,-fil) 'number))
     (fact fact length -fil ^x)))

  ((question enable)
   (question results of ccoff -fil)
   -->
   (let ((direct-term) (feed-back))
     (setq direct-term
       (cond
        ((equal (<- uinterf ':prompt
          '("Is there any correlation in zero") 'symbol) 'y)
          t)
         (t 'f)))
     (setq feed-back
       (cond
        ((equal (<- uinterf ':prompt
          '("Is there any correlation for negative values") 'symbol) 'y)
          t)
         (t 'f)))
     (fact fact feed-back -fil ^feed-back)
     (fact fact direct-term -fil ^direct-term)))

  ((question enable)
   (question loss for -resf -type file)
   -->
   (let ((loss))
     (setq loss
        (<- uinterf ':prompt '("What is the value of"
                               "the loss function for" ,-resf) 'number))
     (fact fact -type has loss -resf ^loss)))

  ((question enable)
   (question fixed aic -syst)
   -->
   (let ((aic))
     (setq aic
       (<- uinterf ':prompt
         '("What is the value of the AIC") 'number))
     (fact fact fixed-aic -syst ^aic)))

; ------------------------------------------------------------------------

  ((fact length -fil -len)
   (fact length -fil unknown)
   test
   (numberp -len)
   -->
   (remove 2))

  ((fact length -fil1 -len)
   (fact sampled -fil2 -fil1 -sr)
   test
   (numberp -len)
   -->
   (fact fact length -fil2 ^(fix (divide -len -sr))))

; ---NOTE-----------------------------------------------------------------------

  ((fact length -f1 -l1)
   (fact length -f2 -l2)
   test
   (and
     (not (equal -f1 -f2))
     (not (equal -l1 -l2)))
   -->
   (cond
     ((not (numberp -l1))
       (fact message ("The length of the file" -f1 "is unknown. This may"
                      "lead to trouble later on.")))
     ((and (numberp -l1) (numberp -l2))
       (fact message ("The files" -f1 "and" -f2 "are not equally long."
```

```
                    "You must CUT one or both before using an identification"
                    "algorithm.")))))

((note interpret stat)
 -->
 (fact message ("If maximum or minimum deviate very much from mean,"
                "this may indicate outliers.")))

((note pick rates must be equal -r1 -r2)
 test
 (not (equal -r1 -r2))
 -->
 (fact message ("You must pick the insignal and outsignal with the same"
                "frequency. This will not work. PICK once again.")))

((note select two interesting regions)
 -->
 (fact message ("I suggest that you try to select two different regions"
                "that are free of strange disturbances and other things"
                "that might not be part of the real system behavior."
                "This can only be done if the file is long enough, i.e.,"
                "if there is any noise, some one hundred points.")))

((note two interesting regions)
 -->
 (fact message ("Select the two regions. If you can't find any"
                "good choice, just take the first and second half of"
                "the data files. You must CUT out two regions if"
                "you want to do cross validation later.")))

((note cut insignal of first half -x)
 (fact length -x -len)
 -->
 (cond
   ((and (numberp -len) (> -len 50))
     (fact message ("Use CUT to cut out the first half of the input signal"
                    "in the file" -x ". Use first record = 1 and"
                    "number of records =" ^(fix (times -len 0.50)) ".")))
   ((and (numberp -len) (< -len 50))
     (fact message ("The file is probably too short to use for a"
                    "meaningful cross validation.")))
   ((equal -len 'unknown)
     (fact message ("Use CUT to cut out the first half of the input signal"
                    "in the file" -x "if the file is not too short.")))))

((note cut insignal of second half -x)
 (- cut-from 1 .-)
 -->
 (fact message ("Use CUT to cut out the second half of the input signal"
                "in the file" -x ".")))

((note cut outsignal of first half -x)
 -->
 (fact message ("Use CUT to cut out the first half of the output signal"
                "in the file" -x ".")))

((note cut outsignal of second half -x)
 (- cut-from 1 .-)
 -->
 (fact message ("Use CUT to cut out the second half of the output signal"
                "in the file" -x ".")))

((note xval-1-cut -file1)
 (fact cut-from 1 - -file1 -start -length)
 -->
 (fact message ("The first part of the file you cut out started at"
                -start "and had length" -length ".")))

((note earlier cut -file)
 (fact cut-from - -file -x -start -length)
 -->
 (fact message ("The last file you CUT was" -x ", the starting column"
                -start "and the record length" -length ".")))

((note earlier xval-1-cut -)
 (fact xval-1-cut - -file -lo -n)
 -->
```

```
 (remove 1)
 (fact message ("The last file you CUT was" -file ", the starting column"
                -lo "and the record length" -n "."))

((note earlier xval-1-cut -file)
 (fact cut-from 1 -file -x -start -length)
 (~(fact xval-1-cut - - - -))
 -->
 (remove 1)
 (fact message ("The last file you CUT was" -file ", the starting column"
                -start "and the record length" -length "."))

((note cut out second part)
 -->
 (fact message ("Cut out a part of the residuals file. Choose a part"
                "that was not used in the parameter estimation."))

((note remove trends from -x)
 -->
 (fact message ("Use the TREND command to remove any"
                "trends in" -x "."))

((note remove trends from second half -x)
 (fact cut-from 1 -x -x1 - -)
 (fact cut-from 1 -x2 -x1 - -)
 (fact trends-removed - -x2 -order)
 test
 (not (equal -x -x2))
 -->
 (remove 1)
 (fact message
  ("On the other half (" -x2 ") you removed trends of order" -order "."))

((note check no trends in -x)
 -->
 (fact message ("Make sure that there are no trends left in" -x "by PLOT.")))

((fact trends-removed -x trends-removed -y -order)
 test
 (> -order 0)
 -->
 (fact message ("You removed trends of an order higher than 0. This"
                "may suggest that the process was not in stationarity."
                "The model may not be valid for control purposes.")))

((note do coh)
 -->
 (fact message ("Check that it is possible to get reasonable results"
                "by making a coherence test. The coherence must be larger"
                "than ~ 0.7 if you are going to use the results in a"
                "parameter estimation and you use one input signal"
                "and expect a deterministic linear model. Use number of"
                "lags = 20 - 25% of the number of data points.")))

((note do aspec)
 -->
 (fact message ("Check that the input has excited the system sufficiently"
                "by examining the input spectrum with ASPEC.")))

((fact coherence-limits -f1 -f2 -locohlim -hicohlim)
 (fact aspec-limits -f1 -loasplim -hiasplim)
 -->
 (fact fact limits -f1 -f2 ^(cond ((< -locohlim -loasplim) -loasplim)
                                  (t -locohlim))
                           ^(cond ((> -hicohlim -hiasplim) -hiasplim)
                                  (t -hicohlim))))

((note do ccoff)
 -->
 (fact message ("Check for direct terms or time delays by making a"
                "cross correlation test with CCOFF on PREWHITEned"
                "inputs and outputs (use the macro PREWHITE). The cross"
                "correlation between input and output should pass"
                "through the origin.")))

((note slide correct number of steps -f1 -f2)
 (fact slided -f1 -f2 -steps)
```

```
 test
 (not (equal -steps 0))
 -->
 (fact message ("Slide the files" -f1 "and" -f2 "so that the correlation"
                "becomes visibly larger than zero in the point 1, i.e.,"
                "use SLID with" -steps "."))

((note start parameter estimation)
 -->
 (fact message ("Parameter estimation may be started. Use the MLID"
                "command to fit models of INCREASING order. Look at the"
                "parameters and the estimated uncertainties. A minimal"
                "AIC is a sign that the model order is high enough.")))

((fact resfile -res -)
 (resid-file-list -lis)
 test
 (not (member -res -lis))
 -->
 (remove 2)
 (fact resid-file-list ^(append1 -lis -res)))

((fact identified-from -nr -sys - - -order)
 (system-list -nr -syslis)
 test
 (not (assoc -sys -syslis))
 -->
 (remove 2)
 (fact system-list -nr ^(append1 -syslis (list -sys -order))))

((resid-file-list -res)
 (fact res-file -f)
 test
 (not (member -f -res))
 -->
 (remove 2)
 (fact res-file ^(cons -f -res)))

((note analyze model -this)
 (fact mlid-data -file -aic1 -)
 (fact mlid-data -this -aic2 -)
 (fact identified-from 1 -file - - -order1)
 (fact identified-from 1 -this - - -order2)
 test
 (and
   (> -aic2 -aic1)
   (> -order2 -order1))
 -->
 (remove 1)
 (fact message ("The model order seems to be high enough, according to"
                "the AIC-test.")))

((note white residuals)
 -->
 (remove 1)
 (fact message ("If the order is high enough, the residuals will be white,"
                "i.e., their autocorrelation close to zero except at"
                "tau = zero. Note that small residuals due to quantization"
                "will not be white.")))

((note interpret resid plot)
 -->
 (remove 1)
 (fact message ("Large single residuals may be caused by measurement errors."
                "If there are any, change corresponding points in the"
                "input and output signals with PLMAG.")))

((note transferfunction B / A -syst)
 -->
 (fact message ("Use SPTRF ... <" -syst "B / A  to produce a frequency"
                "response file from" -syst ".")))

((fact frequency-response -file -syst)
 (bode-plot-list-ba -ba)
 test
 (not (member -file -ba))
 -->
```

```
 (remove 2)
 (fact bode-plot-list-ba ^(append1 -ba -file)))

((note interpret bode plot ba)
 (bode-plot-list-ba -ba)
 -->
 (remove 1)
 (fact message ("Plot the frequency responses for all systems in"
                "one diagram. Use the command BODE" ^(reverse -ba) ".")))

((note transferfunction C / A -syst)
 -->
 (fact message ("Use SPTRF ... <" -syst "C / A to produce a frequency"
                "response file from" -syst ", for the noise influence.")))

((fact noise-response -file -syst)
 (bode-plot-list-ca -ca)
 test
 (not (member -file -ca))
 -->
 (remove 2)
 (fact bode-plot-list-ca ^(append1 -ca -file)))

((note interpret bode plot ca)
 (bode-plot-list-ca -ca)
 -->
 (remove 1)
 (fact message ("Plot the frequency responses for the noise influence of"
                "all systems in one diagram. Use the command BODE"
                ^(reverse -ca) ".")))

((note compare with deter -simulated -real)
 -->
 (fact message ("The simulated signal," -simulated "should agree well with"
                "the real output from the system," -real ". If not, try"
                "another model.")))

((note coherence test -insi -outsi)
 (fact limits -insi -outsi -low -high)
 -->
 (fact message ("The 0.7 limits of the insignal and outsignal"
                "which you have identified from" "(" -insi -outsi ")"
                "are" -low "to" -high ". This is the interesting region in"
                "the Bode diagram.")))

((suggest cross validation or mlid)
 (~(- cut-from 1 .-))
  -->
 (remove 1)
 (fact message ("You may perform another estimation with a higher"
                "model order.")))

((suggest cross validation or mlid)
 (- cut-from 1 .-)
 -->
 (remove 1)
 (fact message ("You may perform another estimation with a higher"
                "model order, or you may start a cross validation"
                "with the part of the signal you have not used for"
                "estimation.")))

((note cross validation)
 (~ (fact cut before estimation))
 -->
 (fact message ("You can not cross validate because two data sets"
                "are not available.")))

((note cross validation)
 (fact cut before estimation)
 (system-list 1 -syslis)
 test
 (> (length -syslis) 1)
 -->
 (remove 1)
 (fact message ("The following system files are available for"
                "cross validation: "
                ^(mapcar 'car -syslis) ". The cross validation"
```

```
                "is started with the command TREND, where trends are"
                "removed from the entire files and loss functions are"
                "computed for the second half.")))

 ((note cross validation)
  -->
  (remove 1))

 ((note remove trends for xval-1 -fil)
  (fact cut-from 1 -x -fil - -)
  (fact trends-removed - -x -order)
  (system-list 1 -syslis)
  test
  (> (length -syslis) 1)
  -->
  (remove 1)
  (fact message ("Earlier you removed trends of order" -order "on a file"
                "cut from" -fil ". It is advisable that you do the same"
                "now.")))

 ((note remove trends for xval-1 -fil)
  (fact cut-from 1 - - - -)
  (system-list 1 -syslis)
  test
  (> (length -syslis) 1)
  -->
  (remove 1) ; was earlier commented
  (fact message ("Remove the trends from the complete insignal"
                "and outsignal, for cross validation.")))

 ((note remove trends for xval-1 -fil)
   -->
  (remove 1))

 ((note compute cross validation residuals)
  (xval-1-resid-file-list -res)
  (system-list 1 -sys)
  -->
  (let ((systems))
    (do
      ((resids -res (cdr resids))
       (syst   (mapcar 'car -sys)))
      ((null resids)
        (setq systems syst))
      (setq syst (wegnehmen (car resids) syst)))
  (remove 1)
  (cond
    (systems
      (fact message
        ("Compute residuals using the complete input and output"
         "signals and the"
         ^(cond ((> (length systems) 1) "systems") (t "system"))
         ^systems "."))))))

; Find minimal AIC

 ((fact mlid-data -syst -aic -)
  (~  (aic-minimum - -))
  -->
  (fact aic-minimum -syst -aic))

 ((aic-minimum -syst -aic)
  (fact identified-from 1 -syst - - -order)
  (~ (aic-minimum-set -))
  -->
  (fact aic-minimum-set ((-syst -order -aic))))

 ((aic-minimum-set -set)
  (aic-minimum -syst1 -aic1)
  (fact mlid-data -syst2 -aic2 -)
  (fact identified-from 1 -syst2 - - -order2)
  test
  (and
   (< -aic2 -aic1)
   (not (member (list -syst2 -order2 -aic2) -set)))
  -->
  (remove 1 2)
```

```
   (fact aic-minimum -syst2 -aic2)
   (let
    ((result nil))
    (do
      ((tmp -set (cdr tmp))
       (res ()))
      ((null tmp) (setq result res))
      (cond
        ((< (caddar tmp) (plus -aic2 (times 0.05 (abs -aic2))))
          (setq res (append1 res (car tmp))))))
    (setq result (cons (list -syst2 -order2 -aic2) result))
    (fact aic-minimum-set ^result)))

  ((aic-minimum-set -set)
   (aic-minimum -syst1 -aic1)
   (fact mlid-data -syst2 -aic2 -)
   (fact identified-from 1 -syst2 - - -order2)
   test
   (and
    (>= -aic2 -aic1)
    (not (member (list -syst2 -order2 -aic2) -set))
    (< -aic2 (plus -aic1 (times 0.05 (abs -aic1)))))
   -->
   (remove 1)
   (fact aic-minimum-set ^(append1 -set (list -syst2 -order2 -aic2))))

; Find minimal loss function of xval-1

 ((fact xval-1-resfile -resf -sys)
  (xval-1-resid-file-list -list)
  test
  (not (member -sys -list))
  -->
  (remove 2)
  (fact xval-1-resid-file-list ^(append1 -list -sys)))

 ((fact xval-1 has loss -resfc -loss)
  (fact xval-1-resfile -resf -syst)
  (fact xval-1-cut -resfc -resf - -)
  (~ (xval-1-minimum - -))
  -->
  (fact xval-1-minimum -syst -loss))

  ((xval-1-minimum -syst -loss)
   (fact identified-from 1 -syst - - -order)
   (~ (xval-1-minimum-set -))
   -->
   (fact xval-1-minimum-set ((-syst -order -loss))))

  ((xval-1-minimum-set -set)
   (xval-1-minimum -syst1 -loss1)
   (fact xval-1 has loss -resfc -loss2)
   (fact xval-1-resfile -resf -syst2)
   (fact xval-1-cut -resfc -resf - -)
   (fact identified-from 1 -syst2 - - -order2)
   test
   (and
    (< -loss2 -loss1)
    (not (member (list -syst2 -order2 -loss2) -set)))
   -->
   (remove 1 2)
   (fact xval-1-minimum -syst2 -loss2)
   (let
    ((result nil))
    (do
      ((tmp -set (cdr tmp))
       (res ()))
      ((null tmp) (setq result res))
      (cond
        ((< (caddar tmp) (plus -loss2 (times 0.05 (abs -loss2))))
          (setq res (append1 res (car tmp))))))
    (setq result (cons (list -syst2 -order2 -loss2) result))
    (fact xval-1-minimum-set ^result)))

  ((xval-1-minimum-set -set)
   (xval-1-minimum -syst1 -loss1)
   (fact xval-1 has loss -resfc -loss2)
```

```
  (fact xval-1-resfile -resf -syst2)
  (fact xval-1-cut -resfc -resf - -)
  (fact identified-from 1 -syst2 - - -order2)
  test
  (and
   (>= -loss2 -loss1)
   (not (member (list -syst2 -order2 -loss2) -set)))
   (< -loss2 (plus -loss1 (times 0.05 (abs -loss1))))
  -->
  (remove 1)
  (fact xval-1-minimum-set ^(append1 -set (list -syst2 -order2 -loss2))))

; Find minimal loss function of xval-2

 ((fact xval-2-resfile -resf -sys)
  (xval-2-resid-file-list -list)
  test
  (not (member -sys -list))
  -->
  (remove 2)
  (fact xval-2-resid-file-list ^(append1 -list -sys)))

 ((fact xval-2 has loss -resfc -loss)
  (fact xval-2-resfile -resf -syst)
  (fact xval-2-cut -resfc -resf - -)
  (~ (xval-2-minimum - -))
  -->
  (fact xval-2-minimum -syst -loss))

  ((xval-2-minimum -syst -loss)
   (fact identified-from 2 -syst - - -order)
   (~ (xval-2-minimum-set -))
   -->
   (fact xval-2-minimum-set ((-syst -order -loss))))

  ((xval-2-minimum-set -set)
   (xval-2-minimum -syst1 -loss1)
   (fact xval-2 has loss -resfc -loss2)
   (fact xval-2-resfile -resf -syst2)
   (fact xval-2-cut -resfc -resf - -)
   (fact identified-from 2 -syst2 - - -order2)
   test
   (and
    (< -loss2 -loss1)
    (not (member (list -syst2 -order2 -loss2) -set)))
   -->
   (remove 1 2)
   (fact xval-2-minimum -syst2 -loss2)
   (let
    ((result nil))
    (do
      ((tmp -set (cdr tmp))
       (res ()))
      ((null tmp) (setq result res))
      (cond
        ((< (caddar tmp) (plus -loss2 (times 0.05 (abs -loss2))))
          (setq res (append1 res (car tmp))))))
    (setq result (cons (list -syst2 -order2 -loss2) result))
    (fact xval-2-minimum-set ^result)))

  ((xval-2-minimum-set -set)
   (xval-2-minimum -syst1 -loss1)
   (fact xval-2 has loss -resfc -loss2)
   (fact xval-2-resfile -resf -syst2)
   (fact xval-2-cut -resfc -resf - -)
   (fact identified-from 2 -syst2 - - -order2)
   test
   (and
    (>= -loss2 -loss1)
    (not (member (list -syst2 -order2 -loss2) -set)))
    (< -loss2 (plus -loss1 (times 0.05 (abs -loss1))))
   -->
   (remove 1)
   (fact xval-2-minimum-set ^(append1 -set (list -syst2 -order2 -loss2))))

  ((note xval-2-cut -)
   (number of original -nr1)
```

```
 (number of xval-1   -nr2)
 test
 (not (equal -nr1 -nr2))
 -->
 (remove 1))

((note xval-2-cut -fil)
 (fact cut-from 2 - fil -start -no)
 -->
 (remove 1)
 (fact message ("The last cut in the second part of the cross validatiion"
                "was start at" -start "and number of records" -no ".")))

((note xval-2-cut -fil)
 (fact cut-from 1 - -fil -start -no)
 -->
 (fact message ("Start the second part of the cross validation by cutting"
                "out a part of the file" -fil "you have not identified on"
                "earlier. For the first estimation you used" -no "points"
                "starting at" -start ". The idea is to estimate models"
                "on two halves and to compute the loss functions of"
                "the residuals using the other halves.")))

((fact xval-1 has loss -resf -loss)              ; Trick rule
 -->
 (fact fact xval-1 has-loss -resf -loss))

((fact xval-2 has loss -resf -loss)              ; Trick rule
 -->
 (fact fact xval-2 has-loss -resf -loss))

((fact mlid-data -syst - -loss)                  ; Trick rule
 -->
 (fact fact original has-loss -syst -loss))

((fact -type has-loss -resf -loss)
 (~ (number of -type -))
 -->
 (remove 1)
 (fact number of -type 1))

((fact -type has-loss -resf -loss)
 (number of -type -nr)
 -->
 (remove 1)
 (remove 2)
 (fact number of -type ^(+ -nr 1)))

((number of original -nr1)
 (number of xval-1 -nr1)
 -->
 (fact fact xval-1 performed))

((fact xval-2 may be performed)
 (fact xval-1 performed)
 -->
 (fact message ("You should now compute models of the second half"
                "and compute loss functions using the first half.")))

((note earlier orders - -)
 -->
 (fact message ("Now run MLID on a part of the signal not earlier used.")))

((note earlier orders - -)
 (~ (order-list -))
 (system-list 1 -list)
 -->
 (fact order-list ^(mapcar 'cadr -list)))

((note possible earlier orders)
 (order-list -list)
 (note xval-2 model - -order)
 test
 (member -order -list)
 -->
 (remove 1 2)
 (fact order-list ^(wegnehmen -order -list)))
```

```
((note possible earlier orders)
 (order-list -list)
 test
 (not (null -list))
 -->
 (fact message ("Models of order" -list "should be handled.")))

((note cut for loss computations -file1 -file2)
 (fact cut-from 1 -file2 - -st -len)
 -->
 (fact message ("To eliminate initial transients"
                "CUT out the last 80 percent of the file" -file1 ", i.e.,"
                "starting column =" ^(fix (times 0.20 -len))
                "and record length =" ^(fix (times 0.80 -len)) ".")))
;
; Decide which model to choose
;

((note interpret result table)
 (number of xval-2 -nr)
 (number of xval-1 -nr)
 (xval-2-minimum-set -s1)
 (xval-1-minimum-set -s2)
 -->
 (let
   ((xval-2-set (sort -s1 '(lambda (x y) (< (cadr x) (cadr y)))))
    (xval-1-set (sort -s2 '(lambda (x y) (< (cadr x) (cadr y))))))
   (do
     ((a xval-2-set)
      (x xval-1-set))
     ((cond
       ((or (null a) (null x))
         (fact message ("Since the two cross validations do not"
                        "coincide the results are not trustworthy."))
         t)
       ((equal (cadar a) (cadar x))
         (fact xval-1 and xval-2 coincide ^(caar a) ^(cadar a))
         t)))
     (cond
       ((> (cadar a) (cadar x)) (setq x (cdr x)))
       ((> (cadar x) (cadar a)) (setq a (cdr a)))))))

((xval-1 and xval-2 coincide -mod -ord)
 (aic-minimum-set -set)
 -->
 (let ((ss (sort -set '(lambda (x y) (< (cadr x) (cadr y)))))
       (res ()))
   (do
     ((aic-set ss (cdr aic-set)))
     ((or (null aic-set) (equal (cadar aic-set) -ord))
       (setq res (cadar aic-set))))
   (cond
     (res
       (fact fact best model -mod -ord)
       (fact message ("The model" -mod "is probably the best possible"
                      "since the loss functions and the AIC have"
                      "a common minimum. The order of this model is"
                      -ord ".")))
     (t
       (fact fact best model -mod -ord)
       (fact message ("Use the model" -mod ". It is probably the best"
                      "model although the AIC does not agree"
                      "with the loss functions."))))))

((note interpret result table)
 (number of xval-2 -nr1)
 (number of xval-1 -nr2)
 test
 (not (equal -nr1 -nr2))
 -->
 (remove 1))

((note create step signal for monte carlo -file)
 (fact cut before estimation)
 (fact length -file -len)
 (number of xval-2 -nr)
```

```
 (number of xval-1 -nr)
 test
 (numberp -len)
 -->
 (remove 1)
 (fact message ("Use INSI, STEP, and X to create a step insignal"
                "for use in a Monte Carlo simulation of step responses."
                "Use INSI to create a step input of length" -len ".")))

((note create step signal for monte carlo -file)
 (number of original -nr)
 (~ (cut before estimation))
 -->
 (remove 1)
 (fact message ("Use INSI, STEP, and X to create a step insignal"
                "for use in a Monte Carlo simulation of step responses.")))

((note create step signal for monte carlo -file)
 -->
 (remove 1))

((note step response distribution of best model)
 -->
 (fact message ("There should not be too much deviation in the step"
                "responses. If there is, the model may not be useful."
                "However, if the signals are short, the static gain"
                "may have some variation.")))

((note transferfunction distribution of best model)
 (fact limits -f1 -f2 -lo -hi)
 -->
 (fact message ("If the transfer functions do not coincide in the"
                "interval" -lo "to" -hi "there may be something wrong.")))

((note transferfunction distribution of best model)
 (~(fact coherence-limits -f1 -f2 -lo -hi))
 -->
 (fact message ("If the transfer functions do not coincide in the"
                "medium frequencies there may be something wrong.")))

((note try to fix a parameter)
 -->
 (fact message ("If there are any parameters, whose values are as small"
                "as their variances, try to set them to zero, with the"
                "FIX subcommand in MLIDSC. First look at the models with"
                "LIST (t) .")))

((note try to fix another parameter -syst1)
 (fact fixed-aic -syst1 -aic1)
 (fact best model -syst2 -)
 (fact mlid-data -syst2 -aic2 -)
 test
 (> -aic2 -aic1)
 -->
 (remove 1)
 (fact message ("It worked fine to fix one parameter. If there is another"
                "parameter which is as small as its variance, try to FIX"
                "it too.")))

((note try to fix another parameter -syst1)
 (fact fixed-aic -syst1 -aic1)
 (fact best model -syst2 -)
 (fact mlid-data -syst2 -aic2 -)
 test
 (< -aic2 -aic1)
 -->
 (remove 1)
 (fact message ("The last FIX didn't work. Do not use this model.")))

((note common sense validation)
 (fact best model -model -order)
 test
 (> -order 4)
 -->
 (fact message ("You have fitted a model with the order" -order "."
                "Maybe you should be suspicious about such a high"
                "model order.")))
```

```
((note check consistency)
 (fact best model - -order)
 (guessed lower model order limit -lowguess)
 test
 (< -order -lowguess)
 -->
 (fact message ("You have fitted a model with"
                "a lower order than your a priori guess.")))

((note check consistency)
 (fact best model - -order)
 (guessed upper model order limit -upguess)
 test
 (> -order -upguess)
 -->
 (fact message ("You have fitted a model with"
                "a higher order than your a priori guess.")))

((note cross validation)
 (- cut-from 1 .-)
 -->
 (fact message ("If you want to perform a cross validation you"
                "should pick out another interesting region of the"
                "signals and use it for a new fitting procedure.")))

((note choose model)
 (aic-minimum -syst -)
 (~(resvar-minimum - -))
 (~(xval-1-minimum - -))
 -->
 (fact message ("Probably the best model is that in the file" -syst
                ", according to the AIC.")))

((note xval-1 may be performed)
 (fact xval-1 performed)
 (~ (fact cut before estimation))
 -->
 (fact message ("You can not cross validate because two data sets"
                "are not available.")))

((note xval-1 may be performed)
 (fact xval-1 performed)
 (fact cut before estimation)
 (number of original -nr1)
 (number of xval-1 -nr1)
 -->
 (remove 1)
 (fact message ("The following system files are available for"
                "cross validation: " -syslis ". The cross validation"
                "is started with the command CUT, where parts of the"
                "signals not earlier estimated on are cut out.")))

((note trend second half as first half -fil)
 (fact trends-removed -res -fil -order)
 -->
 (fact message ("You removed trends of order" -order "from file" -fil ".")))

((note trend before deter plot -file1 -file2)
 -->
 (fact message ("To remove irrelevant biases in the plots,"
"remove trends of order 0 from" -file1 "and" -file2 ".")))

((note plot simulated and real outsignal -f1 -f2)
 -->
 (fact message ("Plot the simulated and real output signal in the"
                "same diagram. Give the command PLOT" -f1 -f2 ".")))

((note length for deter -file)
 (fact length -file -len)
 -->
 (fact message ("The length of" -file "is" -len ". You must simulate"
                -len "points in DETER.")))

; ---SUGGEST-------------------------------------------------------------------

((suggest take care of insignal)
```

```
    -->
    (fact message ("This script assumes that the input signal is"
                   "available in ASCII format, use CONV"
                   "to convert it to binary format."
                   "After this, look at it with PLOT.")))

   ((suggest take care of outsignal)
    -->
    (fact message ("This script assumes that the output signal is"
                   "available in ASCII format, use CONV"
                   "to convert it to binary format."
                   "After this, look at it with PLOT.")))

   ((suggest modify outliers in insignal)
    -->
    (fact message ("If there are any obviously crazy points in the insignal"
                   "you should consider avoiding them or changing them with"
                   "PLMAG. But beware! You might be making a fool"
                   "of yourself.")))

   ((suggest modify outliers in outsignal)
    -->
    (fact message ("If there are any outliers in the outsignal, they must"
                   "be removed. The best way is if you can use parts of"
                   "the signals where there are no outliers. Otherwise use"
                   "PLMAG to change the values of the outlier points.")))

   ((suggest sample in and outsignals -in -out)
    -->
    (fact message ("When identifying it is not recommended that you sample the"
                   "system too fast. 3 - 5 samples per time constant is"
                   "a good choise. If necessary use the command PICK to"
                   "sample the signals" -in "and" -out "."
                   "If the input signal is a PRBS and it is not synchronized"
                   "with the sampling period, it may be dangerous"
                   "to sample it.")))

    ((suggest cross validation or mlid)
     (- cut-from 1 .-)
     -->
     (fact message ("Either continue with the parameter estimation using"
                    "another model order or start the cross validation.")))

    ((suggest cross validation or mlid)
     (fact identified-from 1 -syst -insi -outsi -order)
     (fact mlid-data -syst -aic -loss)
     -->
     (fact message ("You have identified the model" -syst "with order" -order
                    "aic =" -aic "and loss function =" -loss ".")))

   ((suggest remove more trends -last)
    -->
    (fact message ("If there still are trends in the signal give the"
                   "TREND command with a higher order polynomial."
                   "The last trend polynomial you removed was of order"
                   -last ", a higher order may be appropriate.")))

; ---Other rules-----------------------------------------------------------------

   ((fact length -file -len)
    (~ (nplx advice given))
    test
    (numberp -len)
    (not (equal -len 100))
    -->
    (fact nplx advice given)
    (fact message ("Set appropriate plot width with LET NPLX.=" -len ".")))

)))
```

5

The Simnon Example

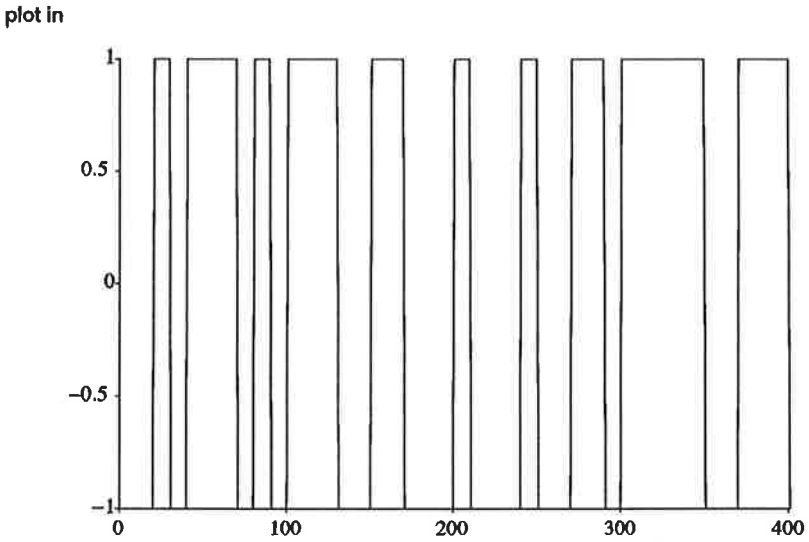
To get a simple example of an identification, an input and output signal was generated with the simulation program Simnon, Elmqvist [1975]. The model was a second order linear system with transfer function

$$G(s) = \frac{\omega^2}{s^2 + 2\omega\zeta s + \omega^2}$$

where $\omega = 0.2$ and $\zeta = 0.7$. The system was sampled with 1.0 Hz. White noise, generated with the Simnon system **noise1**, was added to the output signal from the model. The input signal was generated from a shift register with linear feedback.

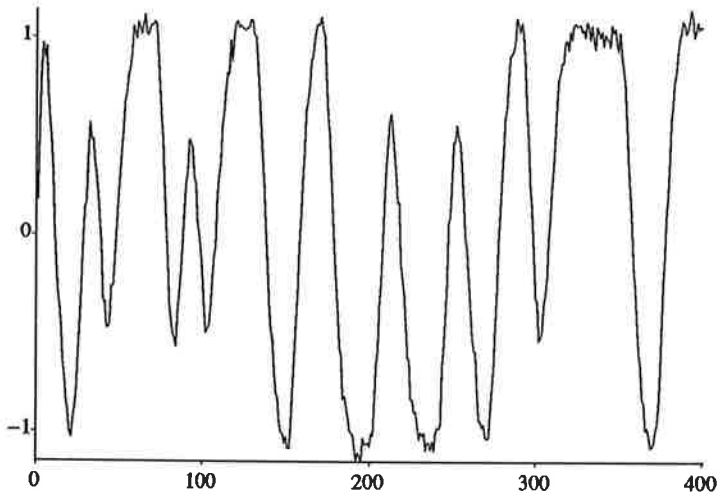
Here follows a listing of a terminal session with Idpac and **(ihs)**, where an attempt is made to identify a discrete model.

```
$ IDPAC
IDPAC    V7A


Copyright (c) Department of Automatic Control
Lund Institute of Technology, Lund, SWEDEN 1986
All Rights Reserved


IHS> think
Do you have any a priori knowledge of the system? > n
ML: This script assumes that the input signal is available in ASCII
format, use CONV to convert it to binary format. After this, look at
it with PLOT.
ML: conv
IHS> conv
conv ... < ... ...
floating point data outfile? >  in
conv in < ... ...
ASCII indata file? > in
conv in < in ...
number of columns? > 1
ML:
ML: stat, plot
IHS> stat

stat in

SUM      =   39.0000000
MEAN     =  9.725685418E-02
VARIANCE =  0.990541101
ST.DEV.  =  0.995259285
MINIMUM  =  -1.00000000     AT ROW      1
MAXIMUM  =   1.00000000     AT ROW     21
LENGTH   =    401

What is the the length of the file in? > 401
ML: If maximum or minimum deviate very much from mean, this may indicate
outliers. Set appropriate plot width with LET NPLX.= 401 .
ML: plot
IHS> let nplx. = 401
ML:
ML: plot
IHS> plot
```



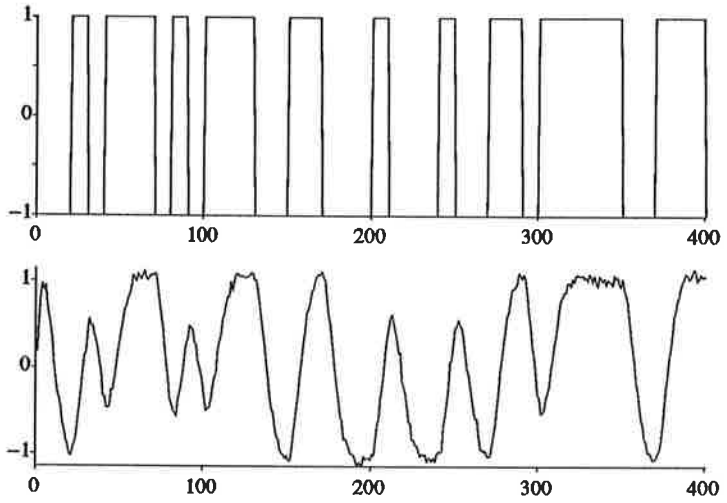

```
ML: If there are any obviously crazy points in the insignal you should
consider avoiding them or changing them with PLMAG. But beware! You
might be making a fool of yourself. This script assumes that the
output signal is available in ASCII format, use CONV to convert it to
binary format.  After this, look at it with PLOT.
ML: plmag, conv
IHS> conv out out 1
ML: The length of the file out is unknown. This may lead to trouble later on.
ML: stat, plot
IHS> stat

stat out

SUM      =   47.0085716
MEAN     =  0.117228359
VARIANCE =  0.565077960
ST.DEV.  =  0.751716673
MINIMUM  = -1.15236998     AT ROW    195
MAXIMUM  =  1.14131999     AT ROW    394
LENGTH   =    401

What is the the length of the file out? > 401
ML: If maximum or minimum deviate very much from mean, this may
indicate outliers.
ML: plot
IHS> plot
```

plot out



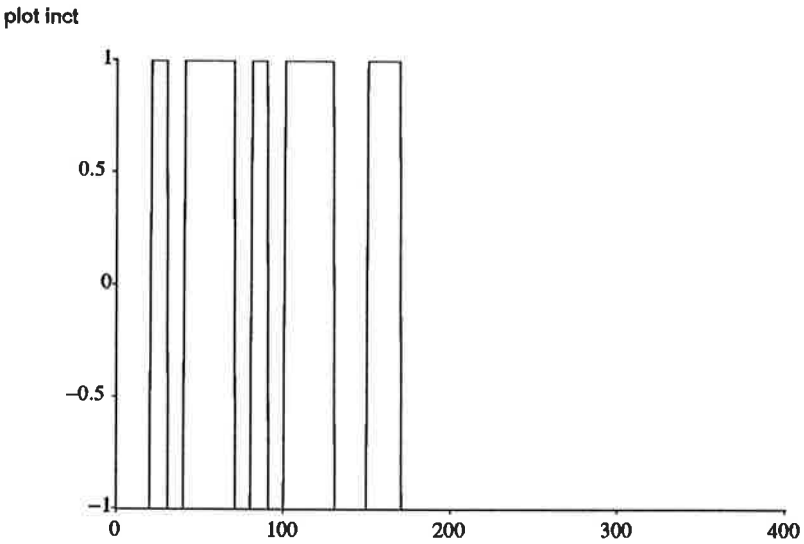

```
ML: If there are any outliers in the outsignal, they must be removed.
The best way is if you can use parts of the signals where there are no
outliers. Otherwise use PLMAG to change the values of the outlier
points. I suggest that you try to select two different regions that
are free of strange disturbances and other things that might not be
part of the real system behaviour. This can only be done if the file is
long enough, i.e., if there is any noise, some one hundred points.
   Plot the files in and out with one PLOT command. GIVE THE COMMAND
PLOT in / out . There is not full support for the command PLOT, so you
will not be queried for all parameters.
ML: plmag, plot
IHS> plot in / out
```

plot in / out




```
ML: When identifying it is not recommended that you sample the system
too fast.3 - 5 samples per time constant is a good choice. If
necessary use the command PICK to sample the signals in and out . If
the input signal is a PRBS signal and it is not synchronized with the
sampling period, it may be dangerous to sample it. Select the two
regions. If you can't find any good choice, just take the first and
second half of the data files. You must CUT out two regions if you
want to do cross validation later.
   Use CUT to cut out the first half of the input signal in the file
in . Use first record = 1 and number of records = 200 . Use the TREND
command to remove any trends in in .
ML: pick, cut, trend
IHS> cut
```

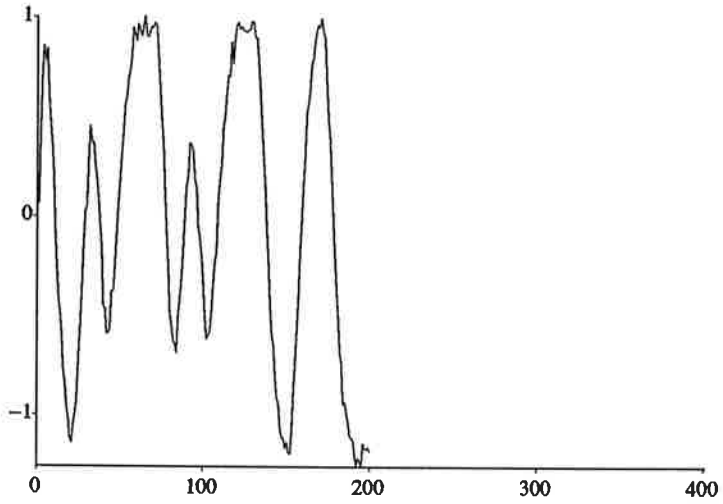
```
cut ... < in ... ...
resultfile? > inc
cut inc < in ... ...
number of first record? > 1
cut inc < in 1 ...
number of records? > 200
ML: The last file you CUT was in , the starting column 1 and the
record length 200 . Use CUT to cut out the first half of the output
signal in the file out .
ML: cut
IHS> cut outc 1 200
ML: Use the TREND command to remove any trends in inc .
ML: trend
IHS> trend
trend ...< inc ...
trend free outfile? > inct
trend inct < inc ...
trend polynomial order? > 0
ML: Make sure that there are no trends left in inct by PLOT.
ML: plot
IHS> plot
```



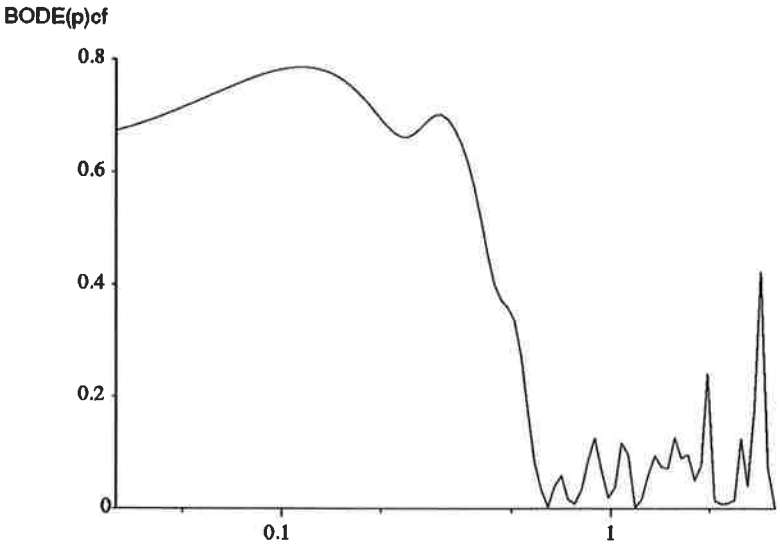

```
ML: If there still are trends in the signal give the TREND command
with a higher order polynomial. The last trend polynomial you removed
was of order 0 , a higher order may be appropriate. Use the TREND
command to remove any trends in outc .
ML: trend
IHS> trend
trend ... < outc ...
trend free outfile? > outct
trend outct < outc ...
trend polynomial order? > 0
ML: Make sure that there are no trends left in outct by PLOT.
ML: plot
```

```
IHS> plot
```

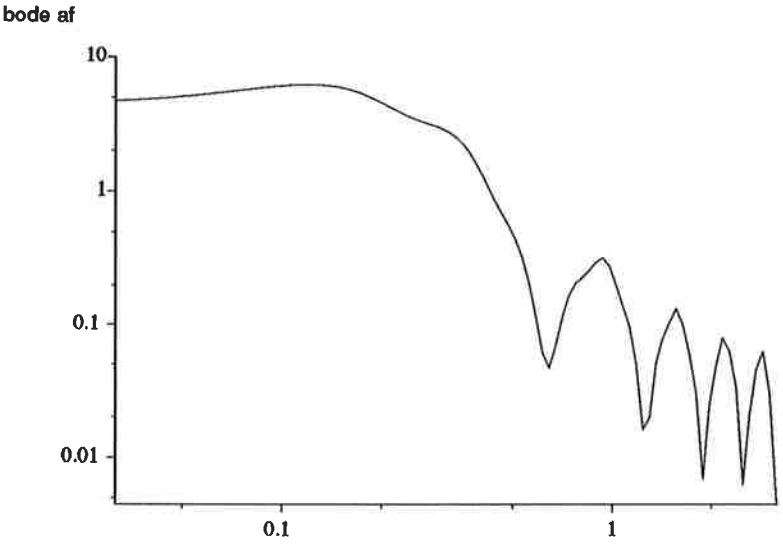
plot outct




```
ML: If there still are trends in the signal give the TREND command
with a higher order polynomial. The last trend polynomial you removed
was of order 0 , a higher order may be appropriate. Check that it is
possible to get reasonable results by making a coherence test. The
coherence must be larger than ~ 0.7 if you are going to use the
results in a parameter estimation and you use one input signal and
expect a deterministic linear model. Use number of lags = 20 - 25% of
the number of data points. Check for direct terms or time delays by
making a cross correlation test with CCOFF on PREWHITEned inputs and
outputs (use the macro PREWHITE). The cross correlation between input
and output should pass through the origin. Parameter estimation may be
started. Use the MLID command to fit models of INCREASING order. Look
at the parameters and the estimated uncertainties. A minimal AIC is a
sign that the model order is high enough.
ML: trend, coh, prewhite, slid, mlid
IHS> coh
coh ... < inct outct ...
coherence outfile? > cf
coh cf < inct outct ...
number of lags? > 50
```



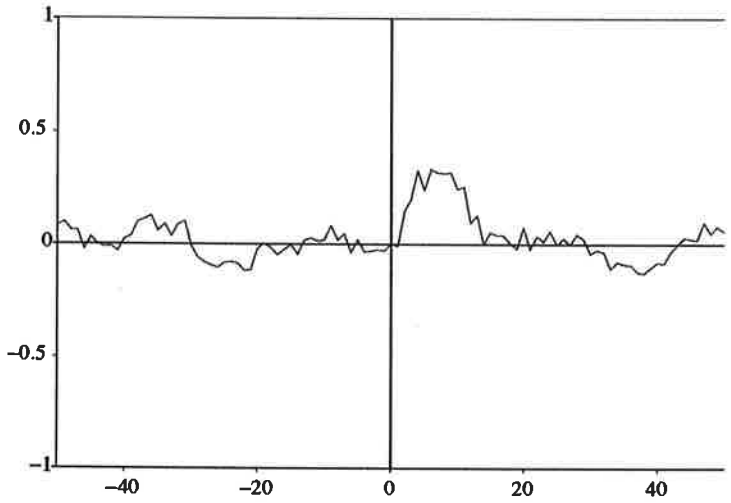

```
Lower limit of interval where the coherence is greater than ~0.7? > 0.0
Upper limit of interval where the coherence is greater than ~0.7? > 0.7
ML: Check that the input has excited the system sufficiently by
examining the input spectrum with ASPEC.
ML: aspec
IHS> asp
aspec ... < inct ...
frequency response outfile? > af
aspec af < inct ...
number of lags? > 50
ML:
ML: bode
IHS> bode
```




```
Lower limit of interval where the spectral density is high? > ???
Upper limit of interval where the spectral density is high? > ???
ML: Check that it is possible to get reasonable results by making a
coherence test. The coherence must be larger than ~ 0.7 if you are
going to use the results in a parameter estimation and you use one
input signal and expect a deterministic linear model. Use number of
lags = 20 - 25% of the number of data points.  Check for direct terms
or time delays by making a cross correlation test with CCOFF on
PREWHITEned inputs and outputs (use the macro PREWHITE). The cross
correlation between input and output should pass through the origin.
```

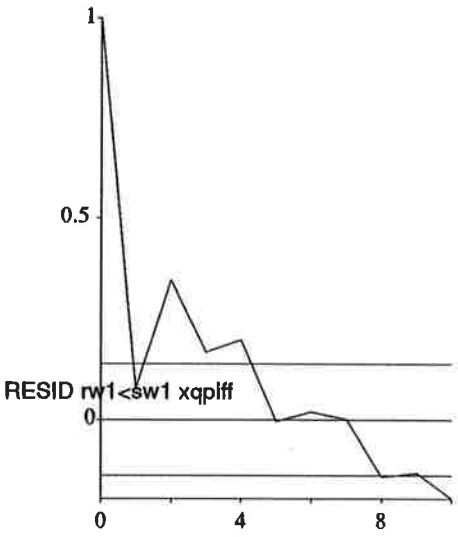
```
Parameter estimation may be started. Use the MLID command to fit
models of INCREASING order. Look at the parameters and the estimated
uncertainties. A minimal AIC is a sign that the model order is high
enough.
ML: coh, prewhite, slid, mlid
IHS> prew
prewhite ... ... < inct outct
the input signal after prewhitening? > p1
prewhite p1 ... < inct outct
the output signal after prewhitening? > p2
ML:
ML: ccoff
IHS> cco
ccoff ... < p1 p2 ...
cross correlation outfile? > ccf
ccoff ccf < p1 p2 ...
number of lags? > 50
```

PLOT(nm)(101)zqp(3)<ccf zqp(1 4)(hp)zqp(2)

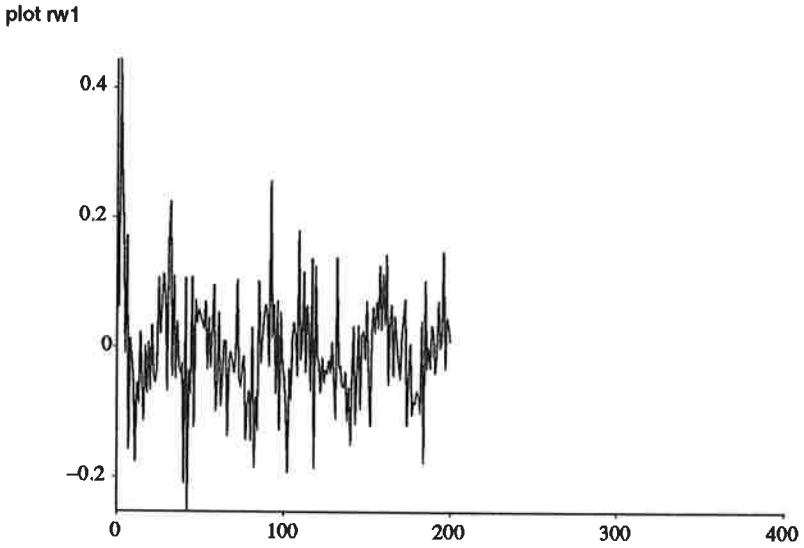



```
What is the x-coordinate for the zero-crossing? > 0
ML: Check that it is possible to get reasonable results by making a
coherence test. The coherence must be larger than ~ 0.7 if you are
going to use the result s in a parameter estimation and you use one
input signal and expect a deterministic linear model. Use number of
lags = 20 - 25% of the number of data points. Check for direct terms
or time delays by making a cross correlation test with CCOFF on
PREWHITEned inputs and outputs (use the macro PREWHITE). The cross
correlation between input and output should pass through the origin.
Parameter estimation may be started. Use the MLID command to fit
models of INCREASING order. Look at the parameters and the estimated
uncertainties. A minima l AIC is a sign that the model order is high
enough.
ML: coh, prewhite, slid, mlid
```

```
IHS> mlid
mlid ... < inct outct ...
system outfile? > sw1
mlid sw1 < inct outct ...
model order? > 1

ML(SC)sw1<xqpiff 1 ext

NO SMALLER LOSS FUNCTION FOUND EVEN IF
MAX RELATIVE COEFF. CORRECTION =   1.8E-04
****************************************

A1            -0.961317     +-  1.135788E-02
B1             7.982602E-02 +-  8.088452E-03
C1             0.252962     +-  5.571383E-02

LAMBDA         8.946531E-02 +-  4.473266E-03
LOSS FUNCTION  0.800404
AIC            -391.986

ML: You have identified the model sw1 with order 1 aic = -391.99 and
loss function = 0.80 . Use INSI, STEP, and X to create a step insignal
for use in a Monte Carlo simulation of step responses. Use SPTRF ... <
sw1 B / A to produce a frequency response file from sw1 . You may
perform another estimation with a higher model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu
residu ... < sw1 inct outct
residuals outfile? > rw1
```

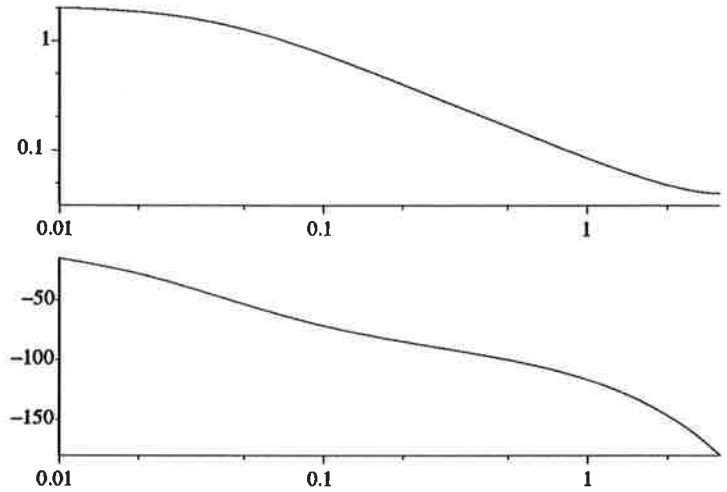



```
VARIANCE OF THE RESIDUALS:  0.185466

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:    20

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11
TEST QUANTITY:   32.7673
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   230.672
DEGREES OF FREEDOM:    17

ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```



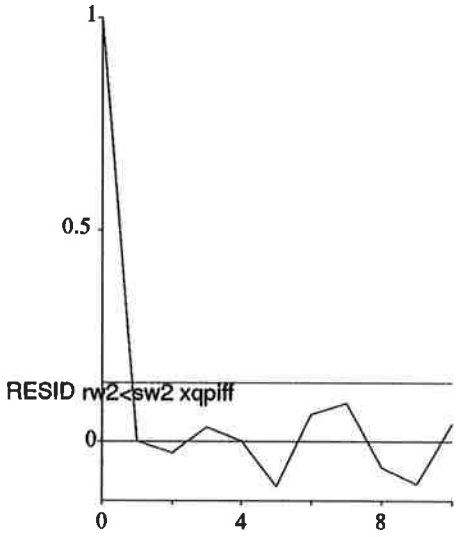

```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG. Use SPTRF ... < sw1 B / A to produce a frequency
response file from sw1 .  You have identified the model sw1 with order
1 aic = -391.99 and loss function = 0.80 . You may perform another
estimation with a higher model order. Use INSI, STEP, and X to create
a step insignal for use in a Monte Carlo simulation of step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf
sptrf ... < sw1 ... / ...
frequency response outfile? > fw1
sptrf fw1 < sw1 ... / ...
numerator polynomial type? > b
sptrf fw1 < sw1 b / ...
denominator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fw1 . The 0.7 limits of the insignal and outsignal
which you have identified from ( inct outct ) are 0 to 0.60 . This is
the interesting region in the Bode diagram.
ML: bode
```

```
IHS> bode fw1
```

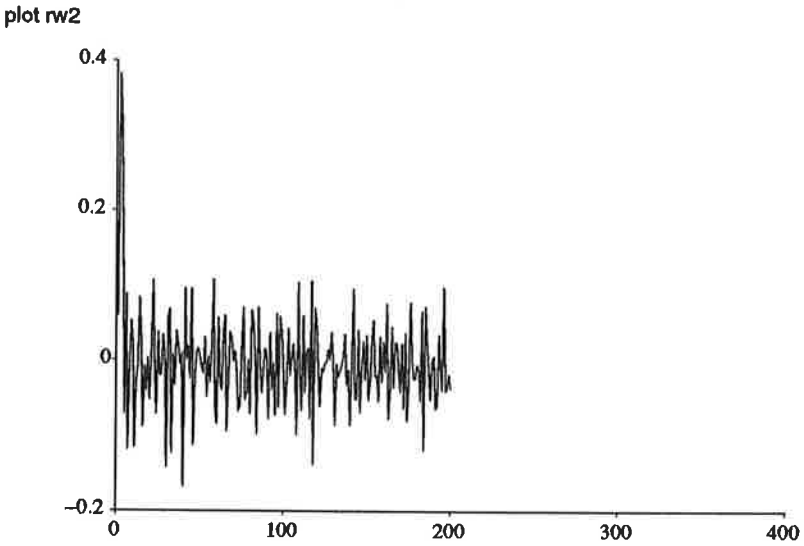



```
ML: You have identified the model sw1 with order 1 aic = -391.99 and
loss function = 0.80 . You may perform another estimation with a
higher model order. Use INSI, STEP, and X to create a step insignal
for use in a Monte Carlo simulation of step responses. Use SPTRF ... <
sw1 C / A to produce a frequency response file from sw1 , for the
noise influence.
ML: mlid, trend, insi, sptrf
IHS> mlid sw2 2

ML(SC)sw2<xqpiff 2 ext

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1              -1.70447      +-  4.135112E-02
A2              0.741546      +-  3.938660E-02
B1              4.107059E-03  +-  9.048525E-03
B2              3.158994E-02  +-  1.124726E-02
C1             -0.751562      +-  8.089002E-02
C2              0.230852      +-  7.023533E-02

LAMBDA          6.496051E-02  +-  3.248025E-03
LOSS FUNCTION   0.421987
AIC             -514.015

ML: The following system files are available for cross validation: sw1
sw2 . The cross validation is started with the command TREND, where
trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0 on
a file cut from in . It is advisable that you do the same now. You
have identified the model sw2 with order 2 aic = -514.02 and loss
function = 0.42 . Use INSI, STEP, and X to create a step insignal for
use in a Monte Carlo simulation of step responses. Use SPTRF ... < sw2
B / to produce a frequency response file from sw2 . You have
identified the model sw1 with order 1 aic = -391.99 and loss function
= 0.80 . You may perform another estimation with a higher model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu rw2
```

```
VARIANCE OF THE RESIDUALS:  4.207507E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   113

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   8.05457
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   25.0273
DEGREES OF FREEDOM:    17
```

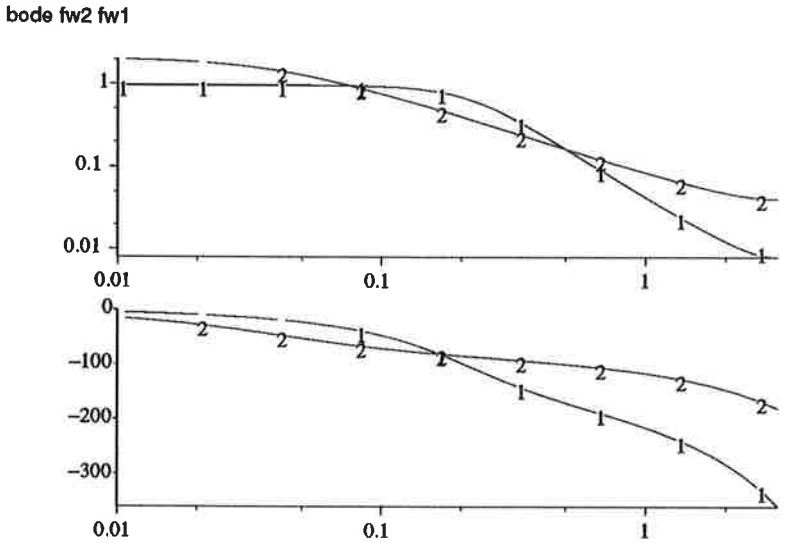



```
ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```



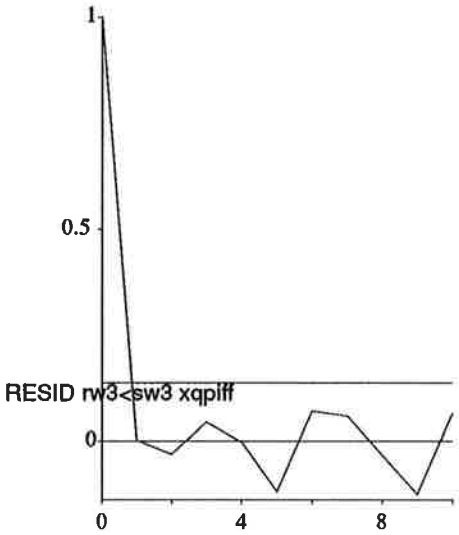

```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG. Use SPTRF ... < sw2 B / A to produce a frequency
response file from sw2 .  You have identified the model sw2 with order
2 aic = -514.02 and loss function = 0.42 . You have identified the
model sw1 with order 1 aic = -391.99 and loss function = 0.80 . You
may perform another estimation with a higher model order.  The
following system files are available for cross validation: sw1 sw2 .
The cross validation is started with the command TREND, where trends
are removed from the entire files and loss functions are computed for
the second half.  Earlier you removed trends of order 0 on a file cut
from in . It is advisable that you do the same now. Use INSI, STEP,
and X to create a step insignal for use in a Monte Carlo simulation of
step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf
sptrf ... < sw2 ... / ...
```

```
frequency response outfile? > fw2
sptrf fw2 < sw2 ... / ...
numerator polynomial type? > b
sptrf fw2 < sw2 b / ...
denominator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fw2 fw1 . The 0.7 limits of the insignal and
outsignal which you have identified from ( inct outct ) are 0 to 0.60.
This is the interesting region in the Bode diagram.
ML: bode
IHS> bode fw2 fw1
```

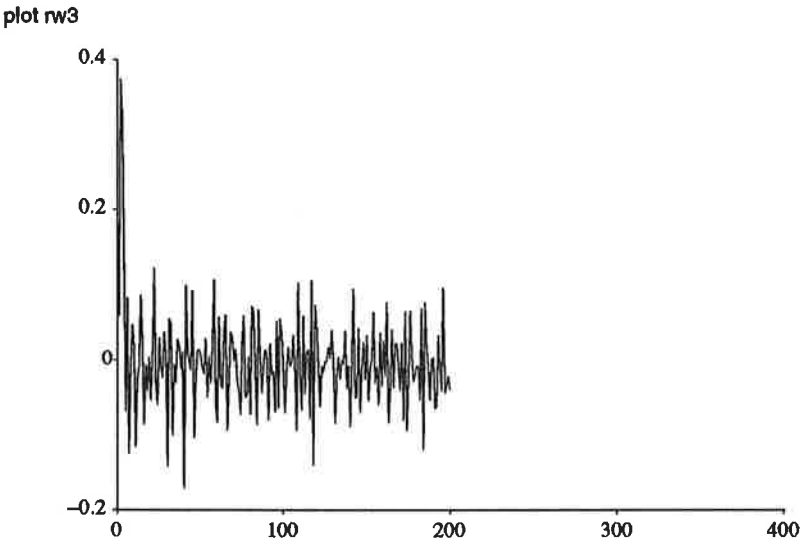



```
ML: You have identified the model sw2 with order 2 aic = -514.02 and
loss function = 0.42 . You have identified the model sw1 with order 1
aic = -391.99 and loss function = 0.80 . You may perform another
estimation with a higher model order. The following system files are
available for cross validation: sw1 sw2 . The cross validation is
started with the command TREND, where trends are removed from the
entire files and loss functions are computed for the second half.
Earlier you removed trends of order 0 on a file cut from in . It is
advisable that you do the same now.
  Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses. Use SPTRF ... < sw2 C / A to
produce a frequency response file from sw2 , for the noise influence.
ML: mlid, trend, insi, sptrf
IHS> mlid sw3 3
```

```
ML(SC)sw3<xqpiff 3 ext

NO SMALLER LOSS FUNCTION FOUND EVEN IF
MAX RELATIVE COEFF. CORRECTION =   1.8E-04
*****************************************

A1             -0.981773      +-  0.185429
A2             -0.520170      +-  0.304403
A3              0.563633      +-  0.133374
B1             -3.954903E-03  +-  1.002183E-02
B2              5.544465E-02  +-  1.342333E-02
B3              6.364993E-03  +-  1.583971E-02
C1             -1.591260E-02  +-  0.192974
C2             -0.357500      +-  0.134005
C3              0.169912      +-  7.120723E-02

LAMBDA          6.446893E-02  +-  3.223446E-03
LOSS FUNCTION   0.415624
AIC             -511.053

ML: The following system files are available for cross validation: sw1
sw2 sw3. The cross validation is started with the command TREND, where
trends are removed from the entire files and loss functions are
computed for the second half.  Earlier you removed trends of order 0
on a file cut from in . It is advisable that you do the same now. The
model order seems to be high enough, according to the AIC-test. You
have identified the model sw3 with order 3 aic = -511.05 and loss
function = 0.42 . Use INSI, STEP, and X to create a step insignal for
use in a Monte Carlo simul ation of step responses.
   Use SPTRF ... < sw3 B / A to produce a frequency response file from
sw3 . You have identified the model sw2 with order 2 aic = -514.02
and loss function = 0.42 . You have identified the model sw1 with
order 1 aic = -391.99 and loss function = 0.80 . You may perform
another estimation with a higher model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu rw3

VARIANCE OF THE RESIDUALS:  4.144936E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   113

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   9.49451
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   21.0533
DEGREES OF FREEDOM:    17
```

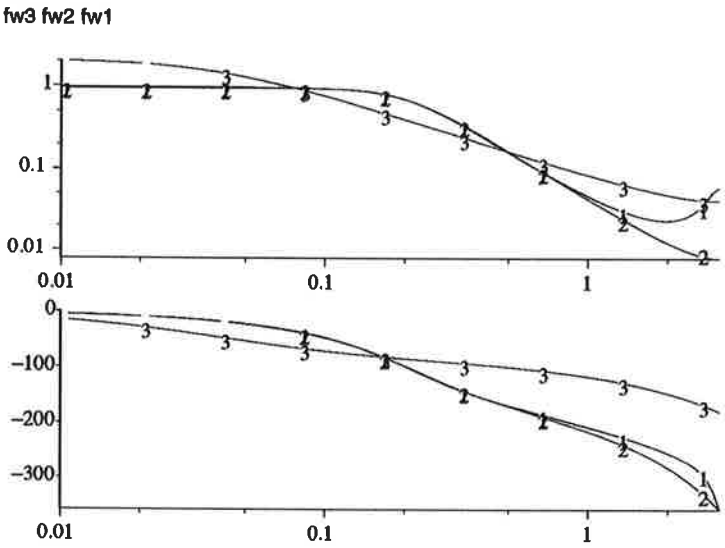



```
ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```



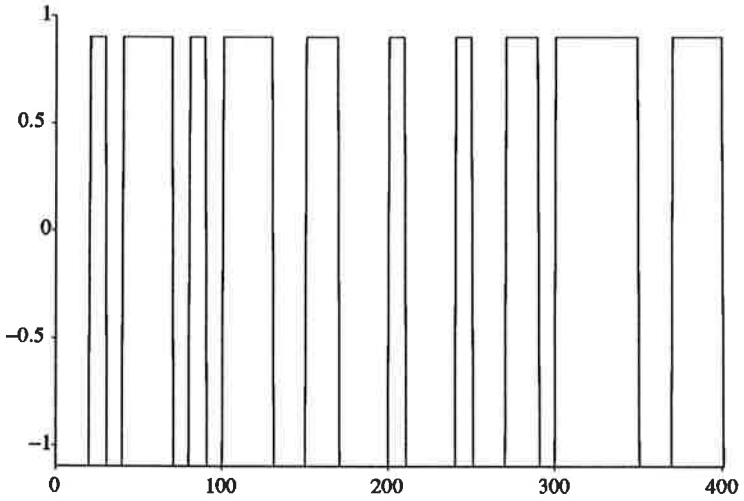

```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG. Use SPTRF ... < sw3 B / A to produce a frequency
response file from sw3 . You have identified the model sw3 with order 3
aic = -511.05 and loss function = 0.42 .  You have identified the
model sw2 with order 2 aic = -514.02 and loss function = 0.42 . You
have identified the model sw1 with order 1 aic = -391.99 and loss
function = 0.80 . You may perform another estimation with a higher
model order.  The following system files are available for cross
validation: sw1 sw2 sw3 .  The cross validation is started with the
command TREND, where trends are remove d from the entire files and
loss functions are computed for the second half.  Earlier you removed
trends of order 0 on a file cut from in . It is advisable that you do
the same now. Use INSI, STEP, and X to create a step insignal for use
in a Monte Carlo simulation of step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf
sptrf ... < sw3 ... / ...
frequency response outfile? > fw3
sptrf fw3 < sw3 ... / ...
numerator polynomial type? > b
sptrf fw3 < sw3 b / ...
denominator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fw3 fw2 fw1 . The 0.7 limits of the insignal and
outsignal which you have identified from ( inct outct ) are 0 to 0.60.
This is the interesting region in the Bode diagram.
```

```
ML: bode
IHS> bode fw3 fw2 fw1
```



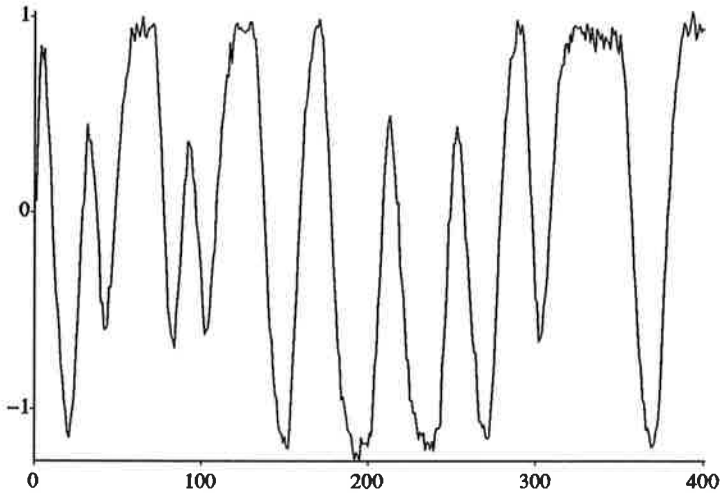

```
ML: You have identified the model sw3 with order 3 aic = -511.05 and
loss function = 0.42 . You have identified the model sw2 with order 2
aic = -514.02 and loss function = 0.42 .You have identified the model
sw1 with order 1 aic = -391.99 and loss function = 0.80 . You may
perform another estimation with a higher model order. The following
system files are available for cross validation: sw1 sw2 sw3 . The
cross validation is started with the command TREND, where trends are
removed from the entire files and loss functions are computed for the
second half. Earlier you removed trends of order 0 on a file cut from
in . It is advisable that you do the same now.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses. Use SPTRF ... < sw3 C / A to
produce a frequency response file from sw3 , for the noise influence.
ML: mlid, trend, insi, sptrf
IHS> trend
trend ... < in ...
trend free outfile? > int
trend int < in ...
trend polynomial order? > 0
ML: Make sure that there are no trends left in int by PLOT.
ML: plot
IHS> plot
```

plot int




```
ML: Earlier you removed trends of order 0 on a file cut from in . It
is advisable that you do the same now. Earlier you removed trends of
order 0 on a file cut from out . It is advisable that you do the same
now.
ML: trend
IHS> trend 0
out in Which file? > out
trend ... < out 0
trend free outfile? > outt
ML: Make sure that there are no trends left in outt by PLOT.
ML: plot
IHS> plot
```

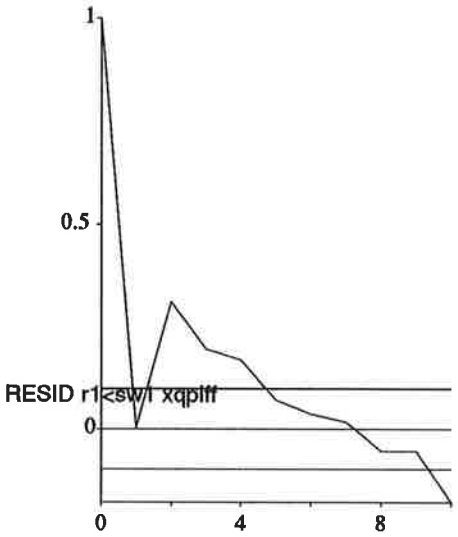
plot outt




```
ML: Earlier you removed trends of order 0 on a file cut from out . It
is advisable that you do the same now. Compute residuals using the
complete input and output signals and the systems sw1 sw2 sw3 .
ML: trend, residu
```

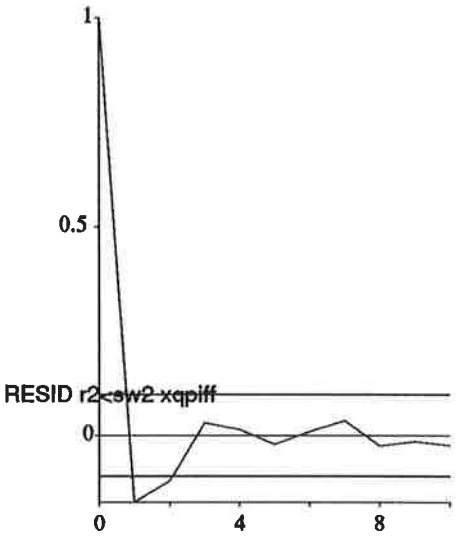
```
IHS> resid
residu ... < ... int outt
residuals outfile? > r1
residu r1 < ... int outt
system infile? > sw1
```

```
VARIANCE OF THE RESIDUALS:  7.339432E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   205

5 PERCENT TOLERANCE LIMITS:   180   220

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   82.8000
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   23.9825
DEGREES OF FREEDOM:    25
```



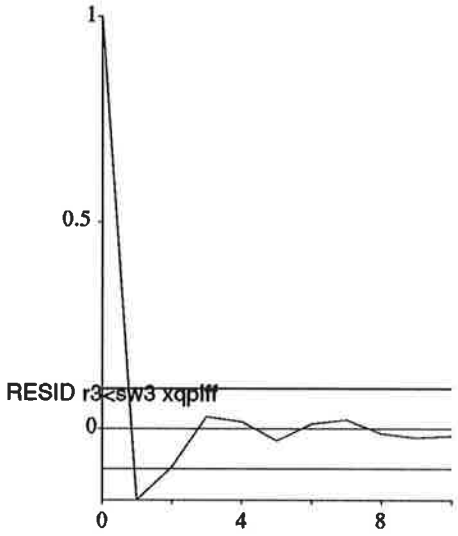

```
ML:
ML: kill, page
IHS> kill
ML: Cut out a part of the residuals file. Choose a part that was not
used in the parameter estimation. The last file you CUT was outc , the
starting column 1 and the record length 200 .
ML: cut
IHS> cut 200 200
cut ... < r1 200 200
resultfile? > r1c
ML:
ML: loss
IHS> loss
The value of the loss function for r1c is 1.3439

What is the value of the loss function for r1c? >  1.3439
ML: Compute residuals using the complete input and output signals and
the systems sw2 sw3 .
```

```
ML: residu, cut
IHS> residu r2 sw2


VARIANCE OF THE RESIDUALS:  3.668312E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   242

5 PERCENT TOLERANCE LIMITS:   180   220

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   16.8056
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   16.8844
DEGREES OF FREEDOM:    25
```




```
ML:
ML: kill, page
IHS> kill
ML: Cut out a part of the residuals file. Choose a part that was not
used in the parameter estimation. The last file you CUT was r1 , the
starting column 200 and the record length 200 .
ML: cut
IHS> cut 200 200
cut ... < r2 200 200
resultfile? > r2c
ML:
ML: loss
IHS> loss
The value of the loss function for r2c is 0.636485

What is the value of the loss function for r2c? > 0.636485
ML: Compute residuals using the complete input and output signals and
the system sw3 .
```

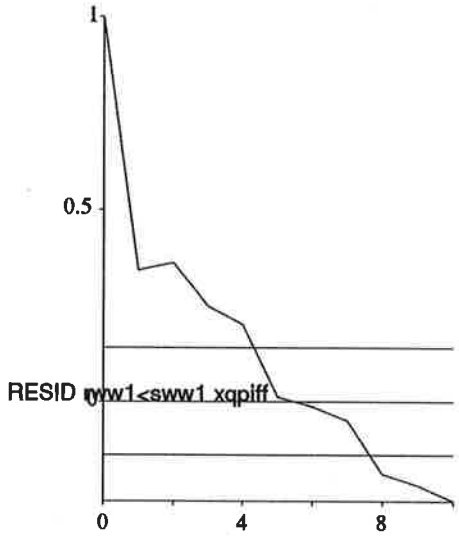
```
ML: residu, cut
IHS> residu r3 sw3

VARIANCE OF THE RESIDUALS:  3.700057E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   242

5 PERCENT TOLERANCE LIMITS:   180   220

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   17.5446
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   19.7303
DEGREES OF FREEDOM:    25
```




```
ML:
ML: kill, page
IHS> kill
ML: Cut out a part of the residuals file. Choose a part that was not
used in the parameter estimation. The last file you CUT was r2 , the
starting column 200 and the record length 200 .
ML: cut
IHS> cut r3c 200 200
ML:
ML: loss
IHS> loss
The value of the loss function for r3c is 0.661432

What is the value of the loss function for r3c? > 0.661432
ML: Start the second part of the cross validation by cutting out a
part of the file in you have not identified on earlier. For the first
estimation you used 200 points starting at 1 . The idea is to estimate
models on two halves and to compute the loss functions of the
residuals using the other halves.
ML: residu, cut
IHS> cut
cut ... < in ... ...
resultfile? > incc
cut incc < in ... ...
number of first record? > 200
cut incc < in 200 ...
number of records? > 200
ML: Start the second part of the cross validation by cutting out a
part of the file in you have not identified on earlier. For the first
estimation you used 200 points starting at 1 . The idea is to estimate
models on two halves and to compute the loss functions of the
residuals using the other halves.
ML: cut
IHS> cut
cut ... < out ... ...
resultfile? > outcc
cut outcc < out ... ...
number of first record? > 200
cut outcc < out 200 ...
number of records? > 200
ML: You removed trends of order 0 from file inc .
ML: trend
IHS> trend 0
trend ... < incc 0
trend free outfile? > incct
ML: You removed trends of order 0 from file outc .
ML: trend
IHS> trend
```

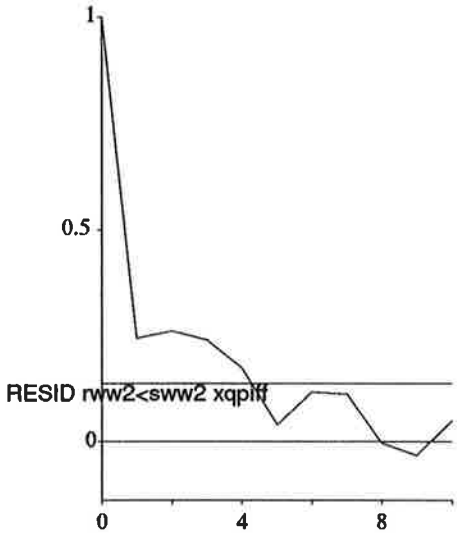
```
trend ... < outcc ...
trend free outfile? > outcct
trend outcct < outcc ...
trend polynomial order? > 0
ML: Models of order 1 2 3 should be handled. Now run MLID on a part of
the signal not earlier used.
ML: mlid
IHS> mlid sww1 1

ML(SC)sww1<xqpiff 1 ext

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1             -0.940986     +-  1.194224E-02
B1              9.185015E-02 +-  9.569808E-03
C1              3.645095E-02 +-  6.171606E-02

LAMBDA          0.117085     +-  5.854249E-03
LOSS FUNCTION   1.37089
AIC            -284.367

ML:
ML: residu
IHS> residu
residu ... < sww1 inct outct
residuals outfile? > rww1
```

```
VARIANCE OF THE RESIDUALS:  8.777311E-03

NUMBER OF CHANGES OF SIG
OF THE RESIDUALS:    88

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   100.325
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   27.7820
DEGREES OF FREEDOM:    17
```



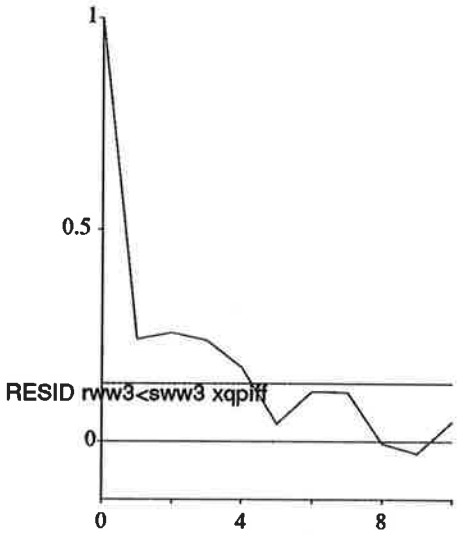

```
ML:
ML: kill, page
IHS> kill
ML: To eliminate initial transients CUT out the last 80 percent of the
file rww1 , i.e., starting column = 40 and record length = 160 .
ML: cut
IHS> cut 40 160
cut ... < rww1 40 160
resultfile? > rww1c
ML:
ML: loss
IHS> loss

The value of the loss function for rww1c is 1.00453
What is the value of the loss function for rww1c? > 1.00453
ML: Models of order 2 3 should be handled. Use INSI, STEP, and X to
create a step insignal for use in a Monte Carlo simulation of step
responses.
ML: mlid, insi
IHS> mlid sww2 2
```

```
ML(SC)sww2<xqpiff 2 ext

NO SMALLER LOSS FUNCTION FOUND EVEN IF
MAX RELATIVE COEFF. CORRECTION =   1.8E-04
*****************************************

A1              -1.71997      +-  4.604993E-02
A2              0.755345      +-  4.209116E-02
B1              1.583395E-02  +-  1.397298E-02
B2              1.904755E-02  +-  1.772897E-02
C1             -0.917399      +-  8.752497E-02
C2              0.140656      +-  7.145921E-02

LAMBDA          9.961389E-02  +-  4.980695E-03
LOSS FUNCTION   0.992293
AIC             -343.006

ML:
ML: residu
IHS> residu rww2
```




```
VARIANCE OF THE RESIDUALS:  4.865638E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   103

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   49.3308
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   47.0183
DEGREES OF FREEDOM:    17

ML:
ML: kill, page
IHS> kill
ML: To eliminate initial transients CUT out the last 80 percent of the
file rww2 , i.e., starting column = 40 and record length = 160 .
ML: cut
IHS> cut rww2c 40 160
ML:
ML: loss
IHS> loss

The value of the loss function for rww2c is 0.352096
What is the value of the loss function for rww2c? > 0.352096
```

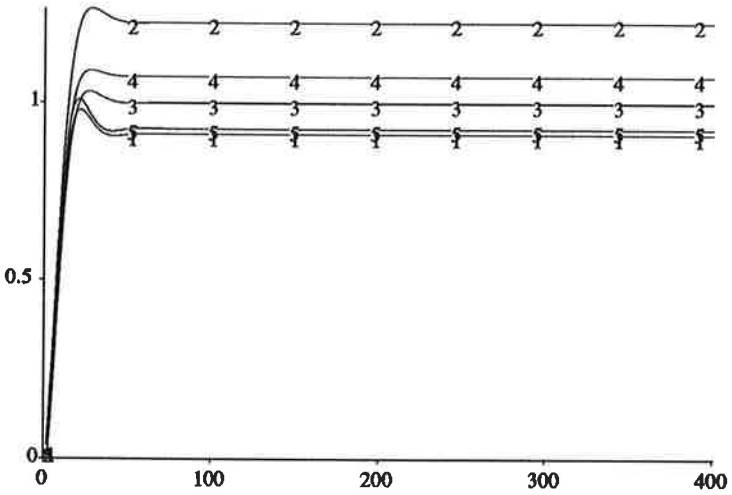
```
ML: Models of order 3 should be handled. Use INSI, STEP, and X to
create a step insignal for use in a Monte Carlo simulation of step
responses.
ML: mlid, insi
IHS> mlid sww3 3

ML(SC)sww3<xqpiff 3

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1              -1.41280
A2              0.232001
A3              0.227611
B1              1.788649E-02
B2              1.829106E-02
B3              1.011040E-02
C1             -0.611359
C2             -0.129527
C3              3.636916E-02

LAMBDA          9.960403E-02 +-  4.980202E-03
LOSS FUNCTION   0.992096
AIC             -337.046

<X
WARNING: THE COVARIANCE MATRIX CANNOT BE COMPUTED

ML:
ML: residu
IHS> residu
residu ... < sww3 inct outct
residuals outfile? > rww3
```




```
VARIANCE OF THE RESIDUALS:  4.881374E-03

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:   103

5 PERCENT TOLERANCE LIMITS:    85   113

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   49.1999
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   45.4999
DEGREES OF FREEDOM:    17

ML:
ML: kill, page
IHS> kill
ML: To eliminate initial transients CUT out the last 80 percent of the
file rww3 , i.e., starting column = 40 and record length = 160 .
ML: cut
IHS> cut rww3c 40 160
ML:
ML: loss
IHS> loss

The value of the loss function for rww3c is 0.354615
What is the value of the loss function for rww3c? > 0.354615
ML: The model sww2 is probably the best possible since the loss
functions and the AIC have a common minimum. The order of this model
is 2 . Use INSI, STEP, and X to create a step insignal for use in a
Monte Carlo simulation of step responses. Use INSI to create a step
```

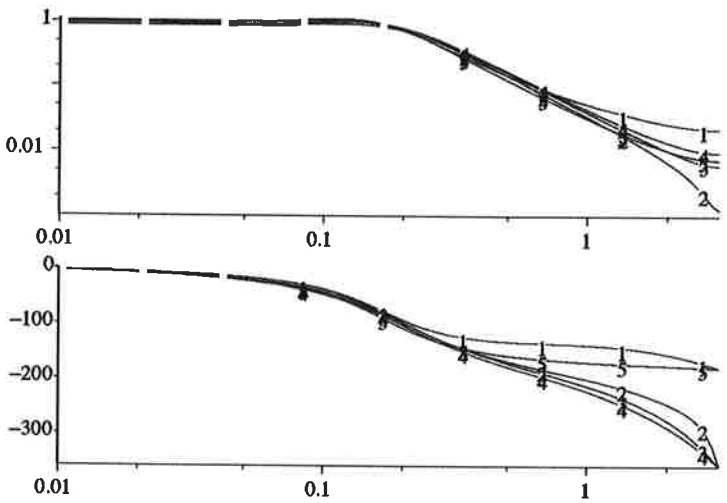
```
input of length 401 .
ML: mlid, insi
IHS> insi
insi ... ...
outfile? > so
insi so ...
number of points? > 401
ML:
ML: step
IHS step
ML:
ML: x
IHS> x
ML:
ML: randstep
IHS> randstep
randstep ... < ... so ...
outfile? > rf
randstep rf < ... so ...
system file? > sww2
randstep rf < sww2 so ...
number of step simulations? > 5
```

PLOT(401)rf



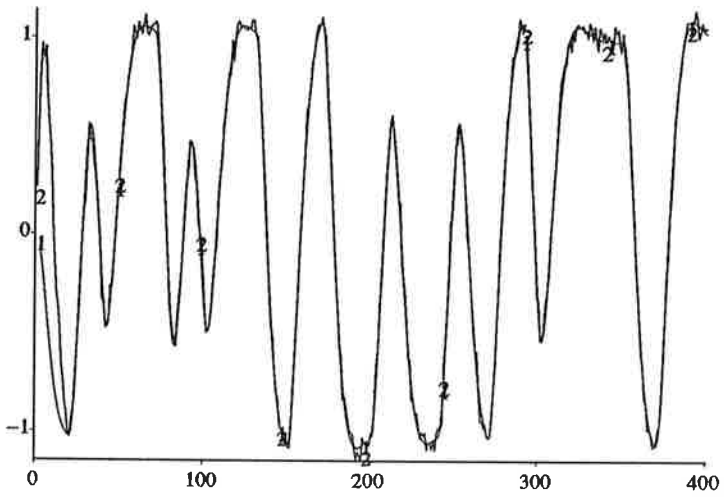
```
ML: There should not be too much deviation in the step responses. If
there is, the model may not be useful. However, if the signals are
short, the static gain may have some variation.
ML: randtf
IHS> randtf
randtf ... < sww2 ...
data outfile? > rtf
randtf rtf < sww2 ...
number of simulations? >5
```

BODE rtf




```
ML: If the transfer functions do not coincide in the interval 0 to
0.60 there may be something wrong.  ML: deter
IHS> deter
deter ... < sww2 in ...
outfile? > d2
deter d2 < sww2 in ...
number of points? > 401
ML: Plot the simulated and real output signal in the same diagram.
Give the command PLOT d2 out .
ML: plot
IHS> plot d2 out
```

plot d2 out




```
ML: The simulated signal, d2 should agree well with the real output
from the system, out . If not, try another model. If there are any
parameters, whose values are as small as their variances, try to set
```

```
them to zero, with the FIX subcommand in MLIDSC.
ML: randstep, stop, mlidsc
IHS> stop
$
```

6

The Atlantic Song Example

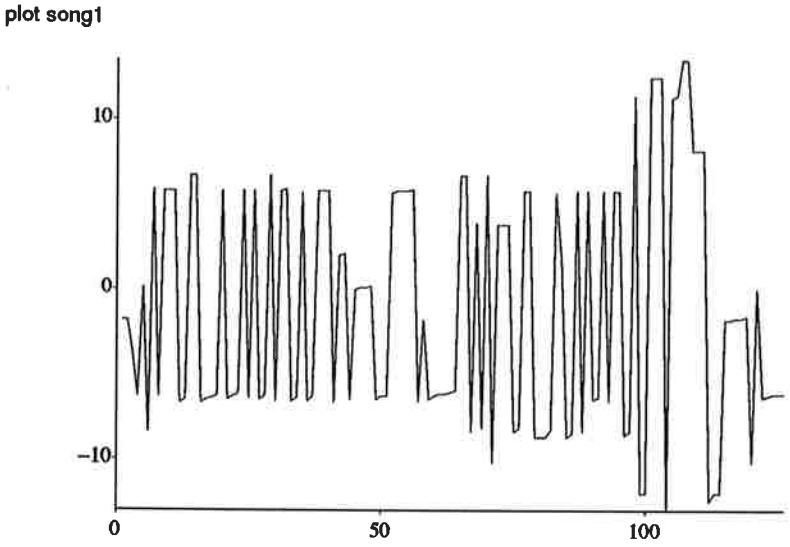
In this example, the expert interface is used to identify a model for data measured during the Atlantic Song experiment. The Atlantic Song is a freighter of the Wallenius lines. It is 197 meters long, weighs 15 000 tons, and has a maximum speed of 21 knots. In the experiment, the rudder angle was perturbed and the yaw angle measured. The experiment was made on Sunday, December 21th, 1969, off the west coast of Denmark, at lat. N. 54°17′, long. E. 4°51′. The course was 217°, and the wind about 8 Beaufort, i.e., 17–20 m/s, fresh gale. The wave height was estimated to 3.5–4.0 m. The sight was poor due to heavy snow fall. The ship had a luffing tendency and a wind gust forced a port yaw. The impact of the waves induced several sudden and violent starboard yaws. The experiment lasted about half an hour. In the beginning the speed was 18.5 knots, but it sank to 18 knots towards the end of the experiment.

The experiment was carried out by two students, Mr. Ekwall and Mr. Edvardsson. Mr. Ekwall was standing on the bridge, together with the captain, Mr. Tärnsjö, the second mate, Mr. Håkansson, and the helmsman, Mr. Brand. Every 15th second, Mr. Ekwall read the yaw angle from the gyro compass, ordered a rudder change to be done by the helmsman, and a rudder angle reading to be done by Mr. Edvardsson. The rudder angle is used as the input, and the yaw angle as the output of the parameter estimation. A more detailed description of the experiment is given in Åström and Kjällström [1972, 1973].

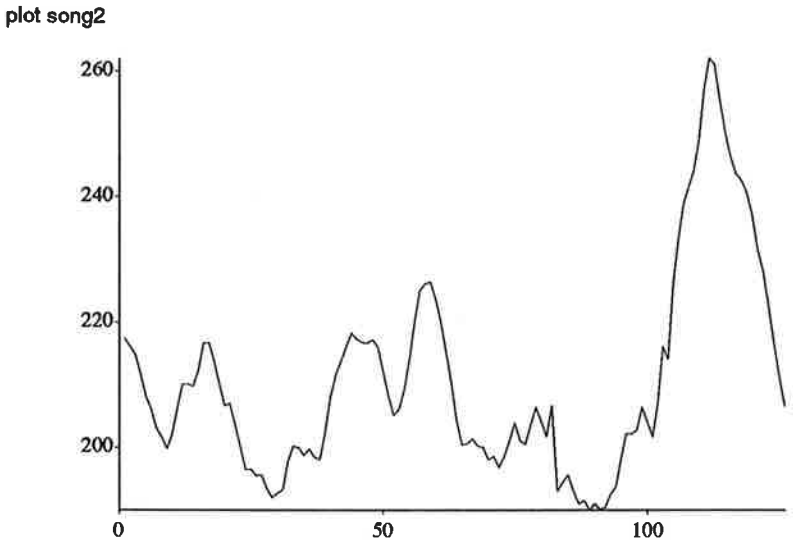
Here follows a listing of a terminal session with Idpac and `(ihs)`, where an attempt is made to identify a discrete model.

```
$ IDPAC
IDPAC    V7A


Copyright (c) Department of Automatic Control
Lund Institute of Technology, Lund, SWEDEN 1986
All Rights Reserved


IHS> think
Do you have any a priori knowledge of the system? > n
ML: This script assumes that the input signal is available in ASCII
format, use CONV to convert it to binary format. After this, look at
it with PLOT.
ML: conv
IHS> conv
conv ... < ... ...
floating point data outfile? > song1
conv song1 < ... ...
ASCII indata file? > song1
conv song1 < song1 ...
number of columns? > 1
ML:
ML: stat, plot
IHS> stat

stat song1

SUM      =  -143.699997
MEAN     =  -1.14047611
VARIANCE =   49.3786011
ST.DEV.  =   7.02699089
MINIMUM  =  -13.0000000      AT ROW    104
MAXIMUM  =   13.5000000      AT ROW    107
LENGTH   =    126

What is the the length of the file song1? > 126
ML: If maximum or minimum deviate very much from mean, this may
indicate outliers. Set appropriate plot width with LET NPLX.= 126 .
ML: plot
IHS> let nplx. = 126
```

```
ML:
ML: plot
IHS> plot
```



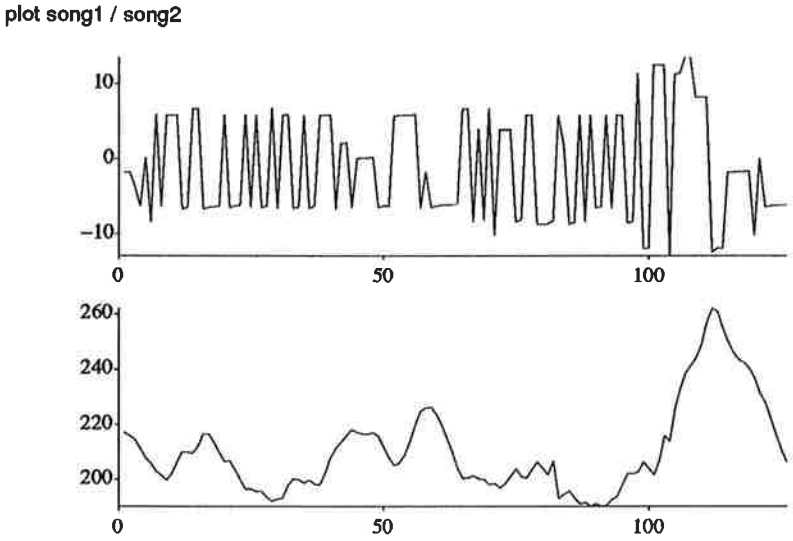

```
ML: If there are any obviously crazy points in the insignal you should
consider avoiding them or changing them with PLMAG. But beware! You
might be making a fool of yourself.
   This script assumes that the output signal is available in ASCII
format, use CONV to convert it to binary format. After this, look at
it with PLOT.
ML: plmag, conv
IHS> conv song2 song2 1
ML: The length of the file song2 is unknown. This may lead to trouble later on.
ML: stat, plot
IHS> stat

stat song2

SUM      =   26552.8008
MEAN     =   210.736511
VARIANCE =   266.270721
ST.DEV.  =   16.3178043
MINIMUM  =   190.000000     AT ROW     89
MAXIMUM  =   262.000000     AT ROW    112
LENGTH   =    126

What is the the length of the file song2? > 126
ML: If maximum or minimum deviate very much from mean, this may
indicate outliers.
ML: plot
IHS> plot
```



ML: If there are any outliers in the outsignal, they must be removed. The best way is if you can use parts of the signals where there are no outliers. Otherwise use PLMAG to change the values of the outlier points. I suggest that you try to select two different regions that are free of strange disturbances and other things that might not be part of the real system behavior. This can only be done if the file is long enough, i.e., if there is any noise, some one hundred points.

Plot the files song1 and song2 with one PLOT command. GIVE THE COMMAND PLOT song1 / song2 . There is not full support for the command PLOT, so you will not be queried for all parameters.

```
ML: plmag, plot
IHS> plot song1 / song2
```



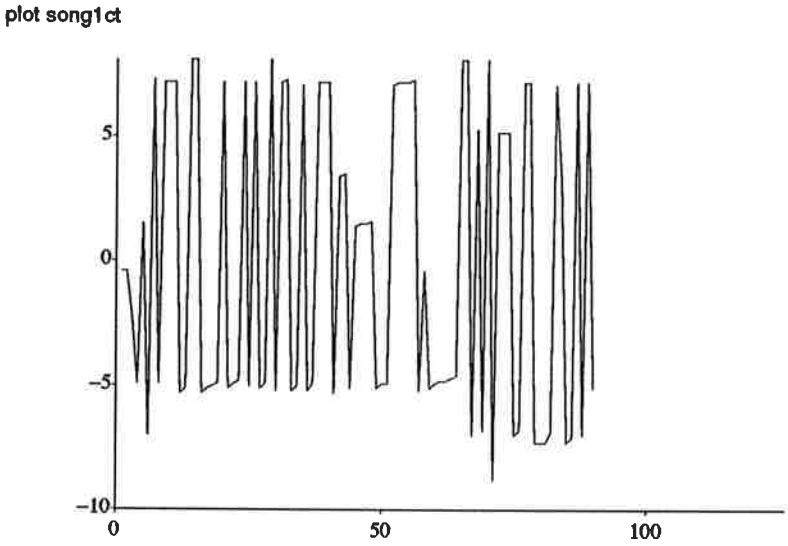
ML: When identifying it is not recommended that you sample the system too fast. 3 - 5 samples per time constant is a good choise. If necessary use the command PICK to sample the signals song1 and song2 . If the input signal is a PRBS and it is not synchronized with the sampling period, it may be dangerous to sample it.

Select the two regions. If you can't find any good choice, just take the first and second half of the data files. You must CUT out two regions if you want to do cross validation later.

Use CUT to cut out the first half of the input signal in the file song1 . Use first record = 1 and number of records = 63 . Use the TREND command to remove any trends in song1 .

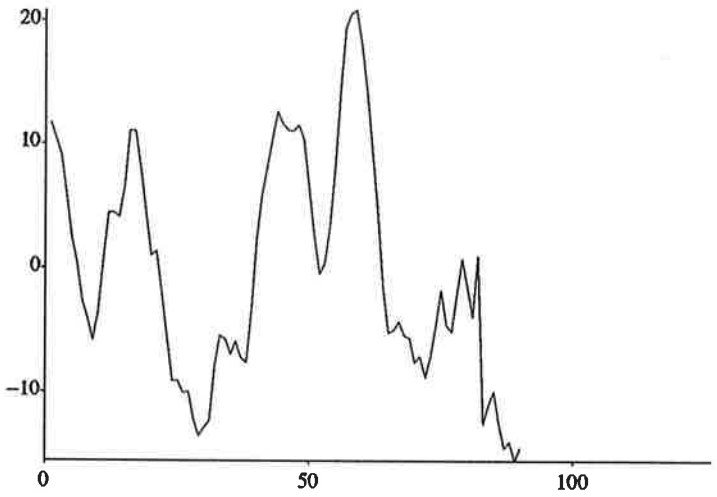
```
ML: pick, cut, trend
IHS> cut 1 90
```

```
cut ... < song1 1 90
resultfile? > song1c
ML: The last file you CUT was song1 , the starting column 1 and the
record length 90 . Use CUT to cut out the first half of the output
signal in the file song2 .
ML: cut
IHS> cut song2c 1 90
ML: Use the TREND command to remove any trends in song1c .
ML: trend
IHS> trend 0
trend ...  < song1c 0
trend free outfile? > song1ct
ML: Make sure that there are no trends left in song1ct by PLOT.
ML: plot
IHS> plot
```



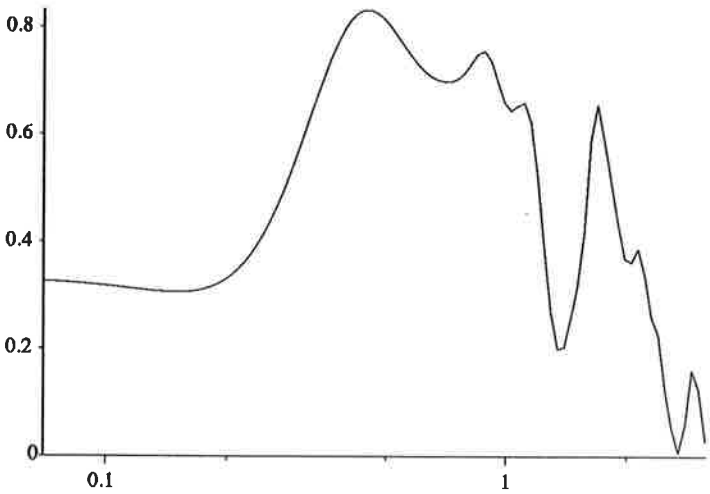

```
ML: If there still are trends in the signal give the TREND command
with a higher order polynomial. The last trend polynomial you removed
was of order 0 , a higher order may be appropriate. Use the TREND
command to remove any trends in song2c .
ML: trend
IHS> trend 0
song2c song1c Which file? > song2c
trend ... < song2c 0
trend free outfile? > song2ct
ML: Make sure that there are no trends left in song2ct by PLOT.
ML: plot
IHS> plot
```

plot song2ct



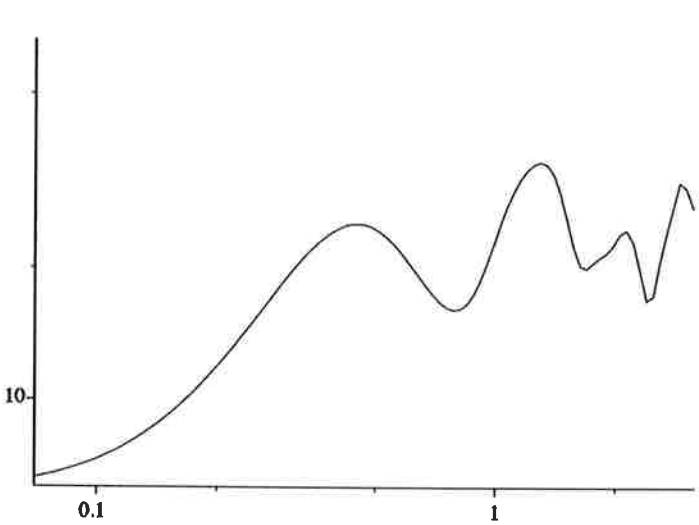

```
ML: If there still are trends in the signal give the TREND command
with a higher order polynomial. The last trend polynomial you removed
was of order 0 , a higher order may be appropriate.
   Check that it is possible to get reasonable results by making a
coherence test. The coherence must be larger than ~ 0.7 if you are
going to use the results in a parameter estimation and you use one
input signal and expect a deterministic linear model. Use number of
lags = 20 - 25% of the number of data points.
   Check for direct terms or time delays by making a cross correlation
test with CCOFF on PREWHITEned inputs and outputs (use the macro
PREWHITE). The cross correlation between input and output should pass
through the origin.
   Parameter estimation may be started. Use the MLID command to fit
models of INCREASING order. Look at the parameters and the estimated
uncertainties. A minimal AIC is a sign that the model order is high
enough.
ML: trend, coh, prewhite, slid, mlid
IHS> coh
coh ... < song1ct song2ct ...
coherence outfile? > cf
coh cf < song1ct song2ct ...
number of lags? > 25
```

BODE(p)cf




```
Lower limit of interval where the coherence is greater than ~0.7 ? > 0.5
Upper limit of interval where the coherence is greater than ~0.7 ? > 1.0
ML: Check that the input has excited the system sufficiently by
```

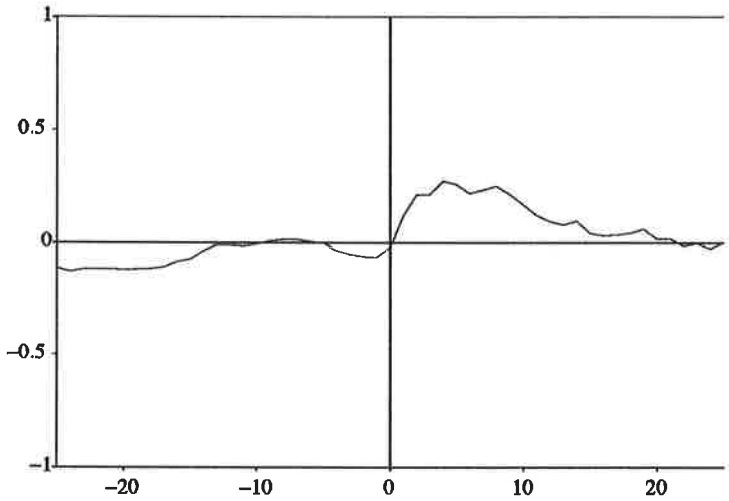
```
examining the input spectrum with ASPEC.
ML: aspec
IHS> aspec
aspec ... < songict ...
frequency response outfile? > af
aspec af < songict ...
number of lags? > 25
ML:
ML: bode
IHS> bode
```




```
Lower limit of interval where the spectral density is high? > 0.5
Upper limit of interval where the spectral density is high? > 2.0
ML: Check that it is possible to get reasonable results by making a
coherence test. The coherence must be larger than ~ 0.7 if you are
going to use the results in a parameter estimation and you use one
input signal and expect a deterministic linear model. Use number of
lags = 20 - 25% of the number of data points.
```

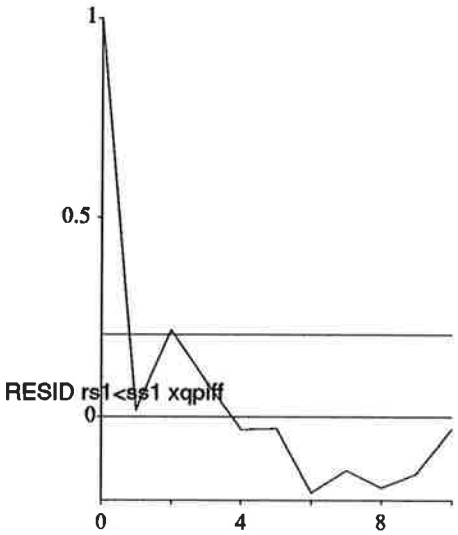
```
   Check for direct terms or time delays by making a cross correlation
test with CCOFF on PREWHITEned inputs and outputs (use the macro
PREWHITE). The cross correlation between input and output should pass
through the origin.
   Parameter estimation may be started. Use the MLID command to fit
models of INCREASING order. Look at the parameters and the estimated
uncertainties. A minimal AIC is a sign that the model order is high
enough.
ML: coh, prewhite, slid, mlid
IHS> prewh p1 p2
ML:
ML: ccoff
IHS> ccoff
ccoff ... < p1 p2 ...
cross correlation outfile? > ccf
ccoff ccf < p1 p2 ...
number of lags? > 25
```

PLOT(nm)(51)zqp(3)<ccf zqp(1 4)(hp)zqp(2)

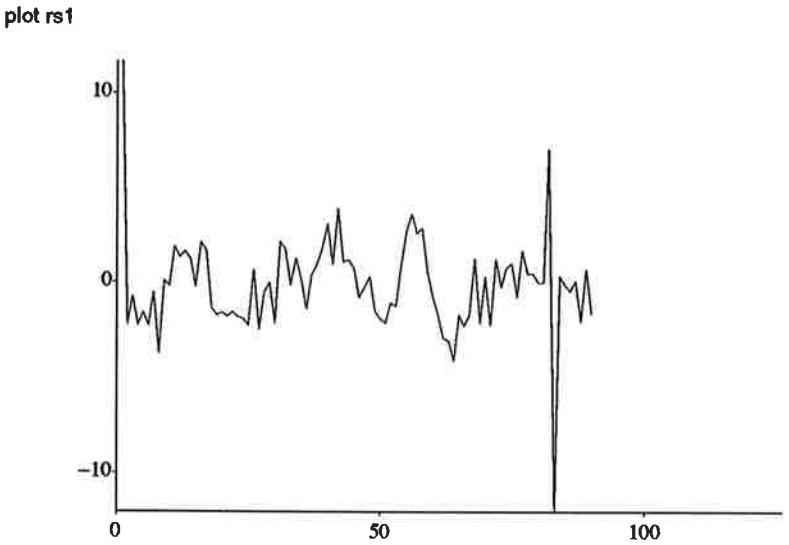



```
What is the x-coordinate for the zero-crossing? > 0
ML: Check that it is possible to get reasonable results by making a
coherence test. The coherence must be larger than ~ 0.7 if you are
going to use the results in a parameter estimation and you use one
input signal and expect a deterministic linear model. Use number of
lags = 20 - 25% of the number of data points.
   Check for direct terms or time delays by making a cross correlation
test with CCOFF on PREWHITEned inputs and outputs (use the macro
PREWHITE). The cross correlation between input and output should pass
through the origin.
   Parameter estimation may be started. Use the MLID command to fit
models of INCREASING order. Look at the parameters and the estimated
uncertainties. A minimal AIC is a sign that the model order is high
enough.
ML: coh, prewhite, slid, mlid
IHS> mlid ss1 1
```

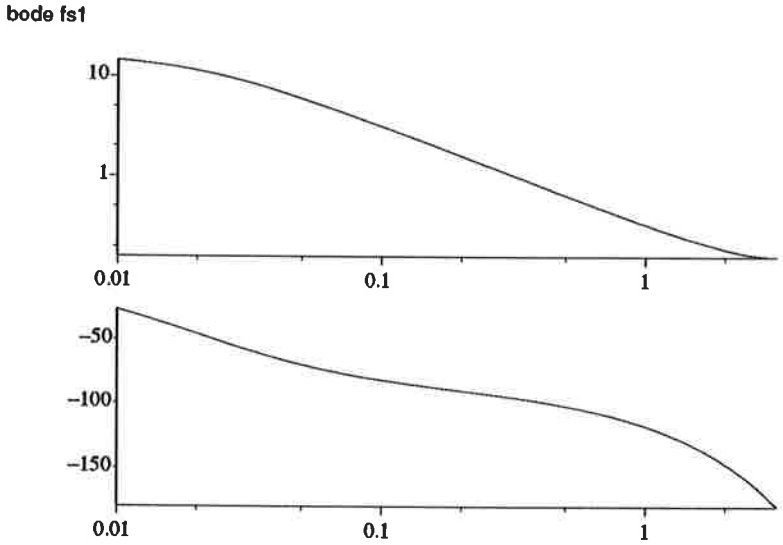
```
ML(SC)ss1<xqpiff 1 ext

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1              -0.980296      +-  3.511994E-02
B1               0.316539      +-  5.112291E-02
C1               0.107419      +-  0.101487

LAMBDA           2.54943       +-  0.190023
LOSS FUNCTION    292.482
AIC              429.866

ML: You have identified the model ss1 with order 1 aic = 429.87 and
loss function = 292.48 . Use INSI, STEP, and X to create a step
insignal for use in a Monte Carlo simulation of step responses.
   Use SPTRF ... < ss1 B / A to produce a frequency response file from
ss1 . You may perform another estimation with a higher model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu
residu ... < ss1 song1ct song2ct
residuals outfile? > rs1
```




```
VARIANCE OF THE RESIDUALS:   6.47762

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:    37

5 PERCENT TOLERANCE LIMITS:    35    53

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   14.6295
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   36.4799
DEGREES OF FREEDOM:    17

ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```

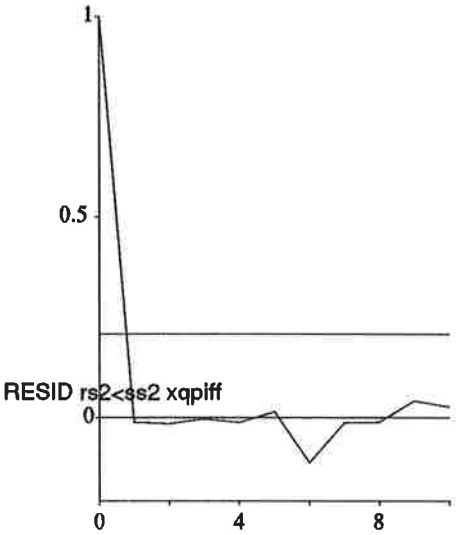



```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG.
   Use SPTRF ... < ss1 B / A to produce a frequency response file from
ss1 . You have identified the model ss1 with order 1 aic = 429.87 and
loss function = 292.48 .
   You may perform another estimation with a higher model order.
Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf fs1
sptrf fs1 < ss1 ... / ...
numerator polynomial type? > b
sptrf fs1 < ss1 b / ...
denominator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fs1 . The 0.7 limits of the insignal and outsignal
which you have identified from ( song1ct song2ct ) are 0.50 to 1.00 .
This is the interesting region in the Bode diagram.
ML: bode
IHS> bode fs1
```



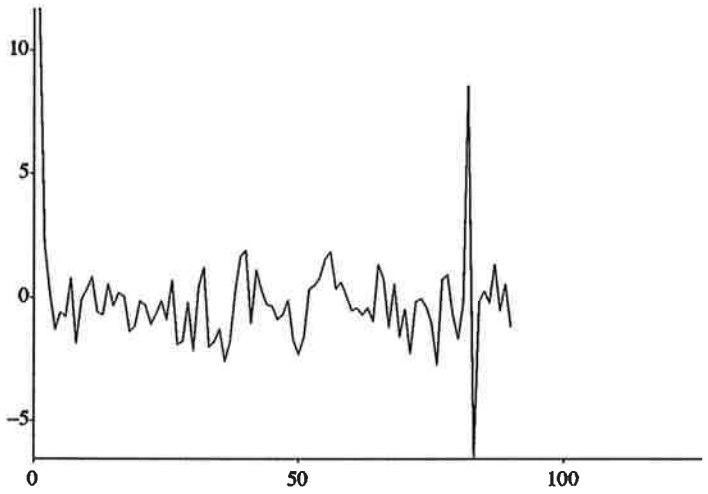

```
ML: You have identified the model ss1 with order 1 aic = 429.87 and
loss function = 292.48 . You may perform another estimation with a
higher model order.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.
   Use SPTRF ... < ss1 C / A to produce a frequency response file from
ss1 , for the noise influence.
ML: mlid, trend, insi, sptrf
IHS> mlid ss2 2

ML(SC)ss2<xqpiff 2 ext

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1              -1.75515      +-  5.400475E-02
A2              0.780311      +-  5.314677E-02
B1              0.386883      +-  3.848762E-02
B2             -0.119482      +-  5.575246E-02
C1              -1.03639      +-  0.123959
C2              0.180254      +-  0.111139

LAMBDA           2.01169      +-  0.149942
LOSS FUNCTION    182.110
AIC              393.224

ML: The following system files are available for cross validation: ss1
ss2 . The cross validation is started with the command TREND, where
trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0 on
a file cut from song1 . It is advisable that you do the same now.
   You have identified the model ss2 with order 2 aic = 393.22 and
loss function = 182.11 .
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.  Use SPTRF ... < ss2 B / A to
produce a frequency response file from ss2 .  You have identified the
model ss1 with order 1 aic = 429.87 and loss function = 292.48 .  You
may perform another estimation with a higher model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu rs2
```

```
VARIANCE OF THE RESIDUALS:   4.01865

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:    33

5 PERCENT TOLERANCE LIMITS:    36    53

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:   1.44666
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:   37.6809
DEGREES OF FREEDOM:    17
```



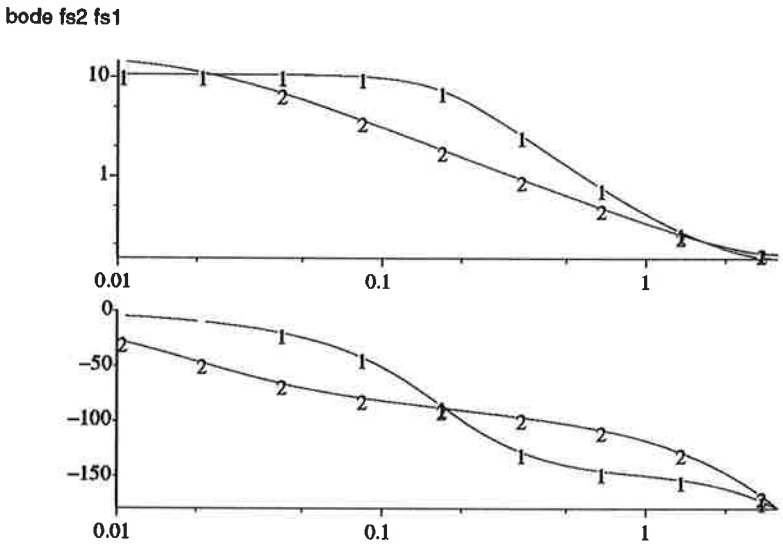

```
ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```

plot rs2




```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG. Use SPTRF ... < ss2 B / A to produce a frequency
response file from ss2 .
```

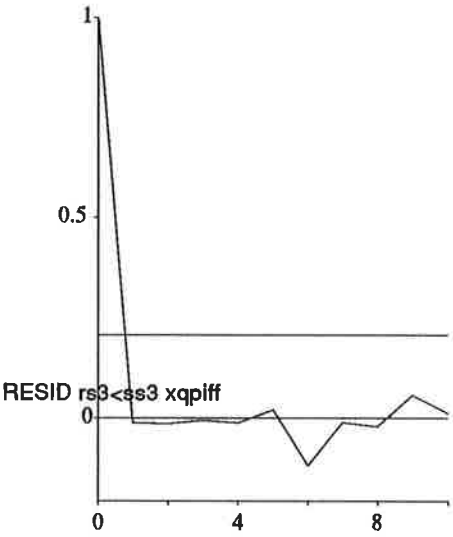
```
   You have identified the model ss2 with order 2 aic = 393.22 and
loss function = 182.11 .  You have identified the model ss1 with order
1 aic = 429.87 and loss function = 292.48 .  You may perform another
estimation with a higher model order.
   The following system files are available for cross validation: ss1
ss2 . The cross validation is started with the command TREND, where
trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0 on
a file cut from song1 . It is advisable that you do the same now.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf fs2
sptrf fs2 < ss2 ... / ...
numerator polynomial type? > b
sptrf fs2 < ss2 b / ...
denominator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fs2 fs1 . The 0.7 limits of the insignal and
outsignal which you have identified from ( song1ct song2ct ) are 0.50
to 1.00 . This is the interesting region in the Bode diagram.
ML: bode
IHS> bode fs2 fs1
```



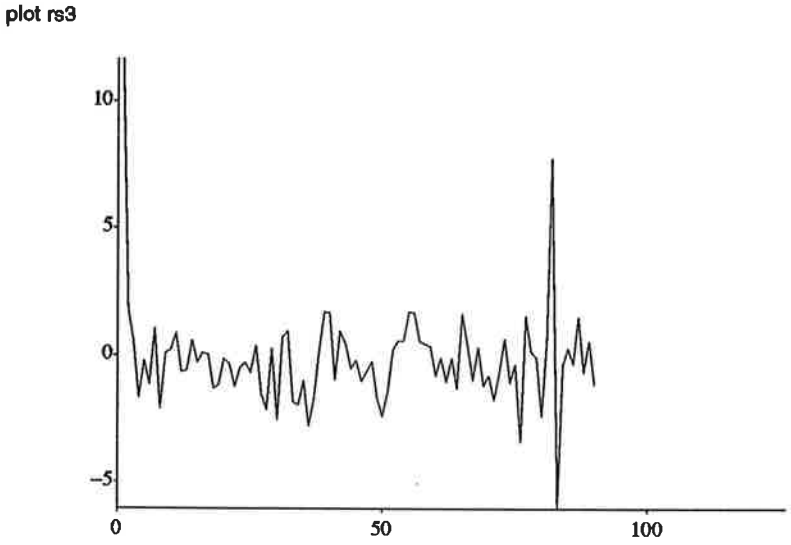

```
ML: You have identified the model ss2 with order 2 aic = 393.22 and
loss function = 182.11 . You have identified the model ss1 with order
1 aic = 429.87 and loss function = 292.48 . You may perform another
estimation with a higher model order.
   The following system files are available for cross validation: ss1
ss2 . The cross validation is started with the command TREND, where
trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0 on
a file cut from song1 . It is advisable that you do the same now.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.  Use SPTRF ... < ss2 C / A to
produce a frequency response file from ss2 , for the noise influence.
```

```
ML: mlid, trend, insi, sptrf
IHS> mlid ss3
mlid ss3 < song1ct song2ct ...
model order? > 3

ML(SC)ss3<xqpiff 3 ext

CONVERGENCE (DV/V<  1.8E-06)
****************************

A1             -0.813560      +-  0.100163
A2             -0.885024      +-  0.139128
A3              0.747067      +-  7.474906E-02
B1              0.383115      +-  3.796156E-02
B2              0.258054      +-  4.592171E-02
B3             -0.133009      +-  5.544883E-02
C1             -6.537110E-02  +-  0.146130
C2             -0.848254      +-  7.189097E-02
C3              0.188249      +-  0.114563

LAMBDA          1.98437       +-  0.147906
LOSS FUNCTION   177.197
AIC             396.763

ML: The following system files are available for cross validation: ss1
ss2 ss3 . The cross validation is started with the command TREND,
where trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0 on
a file cut from song1 . It is advisable that you do the same now.
   The model order seems to be high enough, according to the AIC-test.
You have identified the model ss3 with order 3 aic = 396.76 and loss
function = 177.20 .
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.  Use SPTRF ...  < ss3 B / A to
produce a frequency response file from ss3 .  You have identified the
model ss2 with order 2 aic = 393.22 and loss function = 182.11 .  You
have identified the model ss1 with order 1 aic = 429.87 and loss
function = 292.48 .  You may perform another estimation with a higher
model order.
ML: residu, plmag, sptrf, mlid, trend, insi
IHS> residu rs3
```

```
VARIANCE OF THE RESIDUALS:    3.91070

NUMBER OF CHANGES OF SIGN
OF THE RESIDUALS:    37

5 PERCENT TOLERANCE LIMITS:    35    53

TEST OF INDEPENDENCE OF THE RESIDUALS

E(RES(T)*RES(T+TAU)) FOR: 0<TAU<   11

TEST QUANTITY:    1.73278
DEGREES OF FREEDOM:    10

TEST OF NORMALITY

TEST QUANTITY:    33.9155
DEGREES OF FREEDOM:    17
```

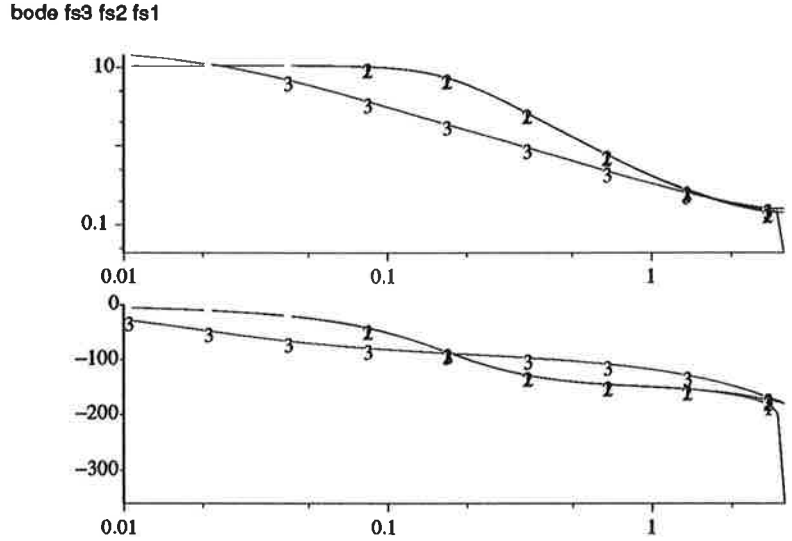



```
ML: If the order is high enough, the residuals will be white, i.e.,
their autocorrelation close to zero except at tau = zero. Note that
small residuals due to quantization will not be white.
ML: page, kill
IHS> kill
ML:
ML: plot
IHS> plot
```



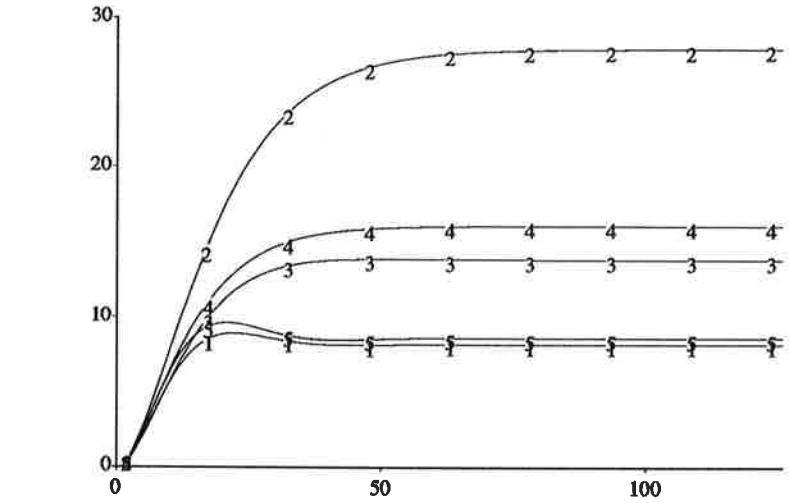

```
ML: Large single residuals may be caused by measurement errors. If
there are any, change corresponding points in the input and output
signals with PLMAG.
   Use SPTRF ... < ss3 B / A to produce a frequency response file from
ss3 . You have identified the model ss3 with order 3 aic = 396.76 and
loss function = 177.20 . You have identified the model ss2 with order
2 aic = 393.22 and loss function = 182.11 . You have identified the
model ss1 with order 1 aic = 429.87 and loss function = 292.48 . You
may perform another estimation with a higher model order.
   The following system files are available for cross validation: ss1
ss2 ss3 . The cross validation is started with the command TREND,
where trends are removed from the entire files and loss functions are
computed for the second half. Earlier you removed trends of order 0
on a file cut from song1 . It is advisable that you do the same now.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
```

```
Carlo simulation of step responses.
ML: plmag, sptrf, mlid, trend, insi
IHS> sptrf fs3
sptrf fs3 < ss3 ... / ...
numerator polynomial type? > b
sptrf fs3 < ss3 b / ...
numerator polynomial type? > a
ML: Plot the frequency responses for all systems in one diagram. Use
the command BODE fs3 fs2 fs1 . The 0.7 limits of the insignal and
outsignal which you have identified from ( song1ct song2ct ) are 0.50
to 1.00 . This is the interesting region in the Bode diagram.
ML: bode
IHS> bode fs3 fs2 fs1
```



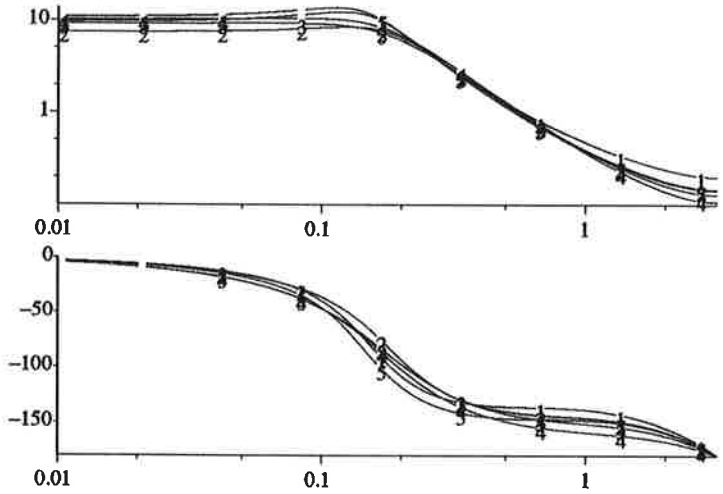

```
ML: You have identified the model ss3 with order 3 aic = 396.76 and
loss function = 177.20 . You have identified the model ss2 with order
2 aic = 393.22 and loss function = 182.11 .  You have identified the
model ss1 with order 1 aic = 429.87 and loss function = 292.48 .
You may perform another estimation with a higher model order.
   The following system files are available for cross validation: ss1
ss2 ss3 . The cross validation is started with the command TREND,
where trends are removed from the entire files and loss functions are
computed for the second half.  Earlier you removed trends of order 0
on a file cut from song1 . It is advisable that you do the same now.
   Use INSI, STEP, and X to create a step insignal for use in a Monte
Carlo simulation of step responses.  Use SPTRF ... < ss3 C / A to
produce a frequency response file from ss3 , for the noise influence.
ML: mlid, trend, insi, sptrf
IHS> insi
insi ... ...
outfile? > inf
insi inf ...
number of points? > 126
```

```
ML:
ML: step
IHS> step
ML:
ML: x
IHS> x
ML:
ML: randstep
IHS> randstep
randstep ... < ... ... ...
outfile? > rf
randstep rf < ... ... ...
system file? > ss2
randstep rf < ss2 ... ...
insignal? > inf
randstep rf < ss2 inf ...
number of step simulations? > 5
```



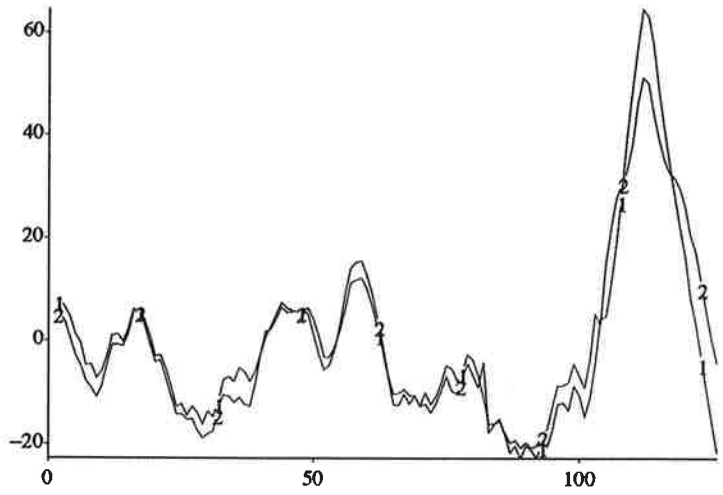

```
ML: There should not be too much deviation in the step responses. If
there is, the model may not be useful. However, if the signals are
short, the static gain may have some variation.
ML: randtf
IHS> randtf
randtf ... < ss2 ...
data outfile? > rtf
randtf rtf < ss2 ...
number of simulations? > 5
```

BODE rtf




```
ML: If the transfer functions do not coincide in the interval 0.50 to
1.00 there may be something wrong. The length of song1 is 126 . You
must simulate 126 points in DETER.
ML: deter
IHS> deter
deter ... < ss2 song1 ...
outfile? > det
deter det < ss2 ...
number of points? > 126
ML: To remove irrelevant biases in the plots, remove trends of order 0
from det and song2 .
ML: trend
IHS> trend 0
trend ... < det 0
trend free outfile? > dett
ML:
ML: trend
IHS> trend 0
trend ... < song2 0
trend free outfile? > song2t
ML: Plot the simulated and real output signal in the same diagram.
Give the command PLOT dett song2t .
ML: plot
IHS> plot dett song2t
```

plot dett song2t



```
ML: The simulated signal, dett should agree well with the real output
from the system, song2t . If not, try another model. Probably the best
model is that in the file ss2 , according to the AIC.
   If there are any parameters, whose values are as small as their
variances, try to set them to zero, with the FIX subcommand in MLIDSC.
First look at the models with LIST (t) .
ML: randstep, stop, list
IHS> list (t) ss2

list (t) ss2

 BEGIN

 "ML(SC)ss2<xqpiff 2 ext

 DISCRETE MISO TRANSFER FUNCTION

 SAMPLE INTERVAL 1. S

 LAMBDA 2.01169 +- 0.149942

 LOSS FUNCTION 182.11

 AIC 393.224

 APOLYNOMIAL
 1. - 1.75515Q^-1 + 0.780311Q^-2

 BPOLYNOMIAL
 Q^-1 * (0.386883 - 0.119482Q^-1 )

 CPOLYNOMIAL
 1. - 1.03639Q^-1 + 0.180254Q^-2
```

```
 COVARIANCE MATRIX
     2.91651E-3   -2.85586E-3   -3.99745E-4    2.26029E-3     3.2624E-3
    -2.85586E-3    2.82458E-3    3.68248E-4   -2.19292E-3   -3.18995E-3
    -3.99745E-4    3.68248E-4     1.4813E-3   -1.41244E-3   -1.62502E-3
     2.26029E-3   -2.19292E-3   -1.41244E-3    3.10834E-3    3.39194E-3
      3.2624E-3   -3.18995E-3   -1.62502E-3    3.39194E-3    1.53658E-2
    -1.26164E-3    1.28896E-3    1.47141E-3   -1.9363E-3   -1.18852E-2

    -1.26164E-3
     1.28896E-3
     1.47141E-3
    -1.9363E-3
    -1.18852E-2
      1.2352E-2

 END

ML:
ML: list, mlidsc, stop
IHS> stop
$
```

7

References

COX, D. R. (1958): *Planning of Experiments*, John Wiley & Sons, New York.

ELMQVIST, H. (1975): "SIMNON, an Interactive Simulation Program for Nonlinear Systems," Technical report, TFRT–3091, Department of Automatic Control, Lund Institute of Technology, Lund.

EYKHOFF, P. (1974): *System Identification, Parameter and State Estimation*, John Wiley & Sons, London.

EYKHOFF, P. (1981): *Trends and Progress in System Identification*, Pergamon Press, Oxford.

FEDOROV, V. V. (1972): *Theory of Optimal Experiments*, Academic Press, New York.

GUSTAVSSON, I. and A. B. NILSSON (1979): "Övningar för Idpac," Technical report, TFRT–7169, Department of Automatic Control, Lund Institute of Technology, Lund.

GUSTAVSSON, I. (1979): "Några macros för Idpac," Technical report, TFRT–7170, Department of Automatic Control, Lund Institute of Technology, Lund.

LARSSON, J. E. and P. PERSSON (1987 a): *An Expert Interface for Idpac*, Licentiate thesis, TFRT–3184, Department of Automatic Control, Lund Institute of Technology, Lund.

LARSSON, J. E. and P. PERSSON (1987 b): "The (ihs) Reference Manual," Technical report, TFRT–7341, Department of Automatic Control, Lund Institute of Technology, Lund.

LJUNG, L. and T. SÖDERSTRÖM (1983): *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Massachusetts.

WIESLANDER, J. (1979 a): *Interaction in Computer-Aided Analysis and Design of Control Systems*, Doctorial Dissertation, TFRT–1019, Department of Automatic Control, Lund Institute of Technology, Lund.

WIESLANDER, J. (1979 b): "Design Principles for Computer-Aided Design Software," *Preprints of the IFAC Symposium on CAD of Control Systems*, Zürich.

WIESLANDER, J. (1979 c): "Idpac User's Guide," TFRT–7605, Department of Automatic Control, Lund Institute of Technology, Lund.

WIESLANDER, J. (1980): "Idpac Commands—User's Guide," TFRT–3157, Department of Automatic Control, Lund Institute of Technology, Lund.

ÅSTRÖM, K. J. and P. EYKHOFF (1971): "System Identification—A Survey," *Automatica*, **7**, 123–162.

ÅSTRÖM, K. J. and C. G. KJÄLLSTRÖM (1972): "Identification and Modelling of Ship Dynamics," Technical report, TFRT–7202, Department of Automatic Control, Lund Institute of Technology, Lund.

ÅSTRÖM, K. J. and C. G. KJÄLLSTRÖM (1973): "Application of System Identification Techniques to the Determination of Ship Dynamics," *Proceedings of the 3rd IFAC Symposium on Identification and System Parameter Estimation*, The Hague/Delft, The Netherlands, p. 415–424.

ÅSTRÖM, K. J. (1980): "Maximum Likelihood and Prediction Error Methods," *Automatica*, **16**, 551–574.

ÅSTRÖM, K. J. (1983 a): "Computer-Aided Modeling, Analysis and Design of Control Systems—A Perspective," *IEEE Control Systems Magazine*, No. 2, May 1983, 4–16.

ÅSTRÖM, K. J. (1983 b): "Modeling and Simulation Techniques," *Agard Lecture Series*, No. 128.

ÅSTRÖM, K. J. (1985): "Computer-Aided Tools for Control System Design," in Jamshidi, M. and C. J. Herget (Eds.): *Computer-Aided Control Systems Engineering*, North-Holland, Amsterdam.