An Expert System for Frequency Response Analysis

Larsson, Jan Eric

1990

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

Link to publication

Citation for published version (APA):
Larsson, J. E. (1990). *An Expert System for Frequency Response Analysis*. (Technical Reports TFRT-7469). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7469)/1-24/(1990)

An Expert System for Frequency Response Analysis

Jan Eric Larsson

Department of Automatic Control
Lund Institute of Technology
November 1990

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

Document name
Technical report

Date of issue
November 1990

Document Number
CODEN: LUTFD2/(TFRT–7469)/1–24/(1990)

Supervisor

Sponsoring organisation

Author(s)
Jan Eric Larsson

Title and subtitle
An Expert System for Frequency Response Analysis

Abstract

This report describes an expert system for frequency response analysis. The system works as a guide and helps the user through a session with a frequency analyser, such as the Solartron 1250. It is written in Scheme and uses the rule-based shell MESS. A part of the report (pages 5–9) was published in the proceedings of the 1990 IFAC World Conference, Tallinn, 1990.

Key words
Expert systems, Process identification

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
24

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

An Expert System for Frequency Response Analysis

Jan Eric Larsson

Department of Automatic Control
Lund Institute of Technology
November 1990

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

Published 1990
Printed in Sweden

An Expert System for Frequency Response Analysis

Jan Eric Larsson M.Sc. L.Sc. B.A.

Department of Automatic Control, Lund Institute of Technology
Box 118, S–221 00 LUND, Sweden
Phone: +46 46 108795, Usenet: JanEric@Control.LTH.Se

Abstract

Frequency response analysis is one of the oldest and most widely used methods to determine the dynamics of a stable linear system. Though quite simple, it requires knowledge and experience of the user, in order to produce reliable results. Available equipment will perform an experiment automatically, but does not support the user in designing the experiment nor in validating the results. The expert system FREX is designed to help the user in performing the analysis. It checks whether the system is linear, finds the frequency and amplitude ranges, verifies the results, and, if errors should occur, tries to give explanations and remedies for them.

Introduction

FREX is a small expert system for frequency response analysis. It is intended to be used as an advisory system together with a frequency response analyser. The system is completely stand-alone, i.e., it runs separately from the frequency analyser, and the user handles all communication between the two. The idea is that of an expert advisor guiding the user through an experiment, with a Q/A dialog.

The system is supposed to be useful for all non-expert users, e.g., students in undergraduate identification courses, and engineers in industry.

The idea of building an expert system for running a frequency analyser was kindled during a graduate course in identification, in the fall of 1985. During this course, all participants performed a frequency response analysis experiment, and gave a small recipe for the method. The knowledge database of FREX originates partly from these recipes. Thus, the knowledge acquisition was performed in part by more than 10 persons.

The knowledge database is quite small, consisting of 65 rules. These are run using backward chaining and a Q/A dialog. The expert system was built with MESS, a very small and simple shell, Larsson (1988). Some minor changes has been made in the MESS source code.

This project has also been described in Larsson (1989).

Frequency Response Analysis

Frequency response analysis is used to find a mathematical model of a real world process with one input and one output signal. A linear model describing the gain and phase shift as functions of the input signal's frequency will be obtained. In the experiment, the process to be identified is driven by a sinusoid input signal,

$$u(t) = u_0 \sin(\omega t).$$

As the method demands the process to be linear, time invariant, and asymptotically stable, the output will become sinusoid when all transients have decayed. The output will be

$$y(t) = |G(i\omega)| u_0 \sin(\omega t + \arg G(i\omega)),$$

where G is the transfer function of the system and ω the input frequency. Normally, the phase difference, $\arg G(i\omega)$, is negative. By measuring the amplitude gain and the phase difference at several different frequencies, ω_i, it is possible to obtain, e.g., a Bode diagram.

The procedure outlined is very sensitive to noise, and it is seldom possible to use it in this simple form. However, the method can be improved by a correlation technique. The output is multiplied with $\sin(\omega t)$ and $\cos(\omega t)$, and integrated.

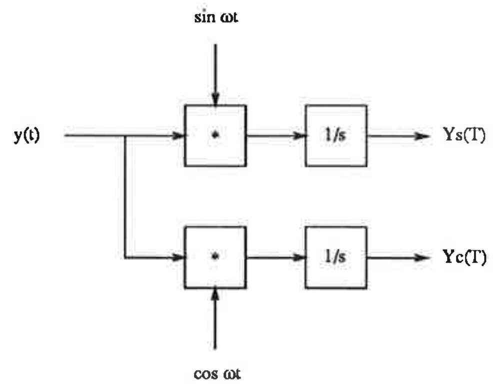

Figure 1. The correlation method setup.

This experimental setup will result in the computation of the following integral values:

$$Y_s(T) = \int_0^T y(t)\sin(\omega t)dt$$

and

$$Y_c(T) = \int_0^T y(t)\cos(\omega t)dt.$$

The integration time, T, should be a whole number of periods, i.e.,

$$T = \frac{2\pi n}{\omega}.$$

This gives

$$Y_s(T) = \frac{T}{2}u_0\text{Re}G(i\omega)$$

and

$$Y_c(T) = \frac{T}{2}u_0\text{Im}G(i\omega).$$

From Y_s and Y_c the amplitude gain and phase difference may easily be computed:

$$|G(i\omega)| = \frac{2}{Tu_0}\sqrt{Y_s^2(T) + Y_c^2(T)}$$

and

$$\arg G(i\omega) = \arctan\frac{Y_c(T)}{Y_s(T)}.$$

An intuitive way of looking at this method is to view it as a filtering of the output signal, using a band pass filter at the frequency ω, with the filter width proportional to $1/T$. The problem with the method is that it often requires long experiments. Of course, it can only be used if it is possible to disturb the process with a sinusoid input. For readings on frequency response analysis in general and this method in particular, see for example Åström (1975) or Söderström (1984). Frequency response analysis was used by the physicist Ångström as early as 1861. It enabled him to make a significant improvement in the determination of thermal diffusivity, Ångström (1861).

Commercial equipment exists for performing frequency response analysis using the method described above. In this project, a Solartron 1250 frequency analyser, Solartron Instrumentation Group (1983 a, b), was used. Before an experiment is performed, one must decide a frequency interval, decay and integration times, and a suitable input signal amplitude. Then the analyser runs experiments at a number of frequencies, covering the frequency interval, computes the integrals, the amplitude gains and the phase differences, and plots a Bode diagram. Thus, the user has to make several experiment design decisions. The expert system is mainly concerned with these decisions, and leaves the numerics to the frequency analyser.

The Knowledge Database

The knowledge database consists of rules, used by the expert system to monitor the frequency response analysis and to diagnose the results. The system performs the analysis in several stages. First, the expert system checks whether the process is stable or must be stabilized. Then it tries to establish that the process is fairly linear, at least in a certain range of amplitudes. After this, the expert system proposes tests that will find the limits of the frequency and amplitude ranges. Then the experiment is performed and several validation tests are made. The validation includes checking with a priori knowledge, verification of frequency limits, and a comparison between the process and a simulation of the model obtained. If, somewhere in this process, anything goes wrong, there are rules for explaining probable causes of the errors. The error diagnostic rules make up a large part of the database.

- The stabilization is given a simple treatment. If the process is not stable from the beginning, the system suggests a proportional regulator to stabilize it. If, when the results have been obtained, the model does not agree with the a priori knowledge, the system may guess that the experiment has led to the identification of the inverse of the controller. In this case it recommends a new experiment with better noise suppression, a different controller or a different input connection place.
- In order to assess the linearity properties of the process, the system tries several things. A visual inspection of the signals is used if possible. A dual step test is proposed, i.e., the results of two step inputs with the same amplitude, one positive and one negative, are examined. Also, several small step inputs are tried at different signal levels. All the outputs should look essentially the same in the linearity range. If there is friction in the process, this can usually be seen in the output signal. Thus, if the user knows that there is friction in the process, or the output signal is deformed, the system recommends the use of a bias in the experiment, in order to avoid friction.
- If the user knows or has an educated guess on the frequency range, the system uses

this in the experiment. If the result is not satisfactory, the error diagnosis tries to adjust the frequency limits. In cases where no guess is available, the system uses a step response experiment to get a suggestion for the cut-off frequency. The frequency limits are centered around this frequency. The system may also be designed for a certain frequency. If so, an interval around this frequency is included.

- The amplitudes of the signals used in the experiment must be larger than the noise level of the process, and, in case the process is digital, larger than the quantization level. On the other hand, the signals must be small enough not to reach linearity or other boundaries. In order to check this, the static gain of the process transfer function is measured. Should the model obtained not be satisfactory, the system checks whether the signals reaches some kind of amplitude limitation. A smaller amplitude is then recommended.
- The frequency response analysis experiment is executed using the gathered information. First a quick experiment with frequencies covering the whole frequency range is performed. A rather low amplitude is recommended. If there are points in the Bode diagram where the curves change rapidly, new experiments with a frequency range fitted rather closely around these points are suggested.
- When a result has been obtained, it must be tested and validated. For this reason the system compares the model against any a priori knowledge available. If the frequency range has only been guessed, that guess must also be checked afterwards. In this case the system suggests a new experiment with a wider frequency range. Finally, the system recommends that the model obtained is used in a simulation. In this way it is possible to make sure that the process and the model has the same behavior for some selected input signals.
- If the analysis should fail, or the results be incompatible with a priori knowledge, several errors may be diagnosed. If the process has been used in a closed loop, the inverse controller may have been identified. When the signal to noise ratio is low in the higher frequencies, the method will only produce reliable results at lower frequencies, and the upper frequency limit must be lowered. If the signals reach amplitude limitations the experiment should be rerun with a smaller amplitude; and vice versa: if the amplitude is close to the noise or quantization level, the experiment should be rerun with a larger amplitude or longer integration times. If there is backlash or hysteresis in the process, making it strongly non-linear, frequency response analysis may be unable to provide a good result. If there is friction in the process, the linearity properties may have been destroyed. In this case the experiment should be rerun with a bias if possible. Another explanation may be that the a priori knowledge is wrong. Finally, if the cut-off frequency differs quite a lot from the desired working frequency, very tight control will be needed, and it will probably be a good idea to run the experiment with a feedback to speed up the process.

Structure of the Rule Base

The knowledge database currently consists of 65 rules, running in backward chaining. The rules are split into seven groups, each of which is concerned with a different task.

- The `phase control` rule is the only one in its group, and it executes the other rule groups, one after the other.
- The `linearity` rules are concerned with checking whether the system is linear or not. If it is, they try to establish a linearity range.
- The `frequency range` rules try to find the frequency range, i.e., the lower and upper frequency limits to be used in the experiment.
- The `amplitude range` rules try to find lower and upper limits for the amplitude of the input signal. They then pick an amplitude close to the lower limit.
- The `perform experiment` rules take care of telling the user to actually run the experiment. First, an experiment covering the whole frequency range is performed, then, if there are any points of special interest, e.g., where the transfer function changes very rapidly, experiments are made, which cover these points more closely.
- The `verify` rules propose some tests to verify the results. Any a priori knowledge is checked, if the frequency range was guessed, it is increased, and a simulation is suggested.
- The `explain errors` rules are invoked if the analysis does not end with success, and certain error conditions are present. These

rules try to guess the source of the error and, if possible, suggest a remedy.

The rules are written in the MESS rule format, which should make them rather easily understood. Three examples of actual rules are shown below.

```
(rule phase-control
  (if
    (the system is stabilized)
    (the linearity range is known)
    (the frequency range is known)
    (the amplitude range is known)
    (the experiment has been performed)
    (the result is verified))
  (then
    (the frequency analysis has been performed)))
```

The phase control rule schedules the other rules, so that the different phases of a frequency response analysis are performed, one after the other. Next, here is an experiment design rule.

```
(rule stabilize-2
  (if
    (not (the system is stable))
    (* the system can be stabilized with a
       feedback loop))
  (then
    (action (use a proportional regulator
             to stabilize the system))
    (action (connect the input to the reference
             signal of the controller))
    (the system is stabilized)))
```

This rule takes care of stabilizing the system before the experiments take place. This is necessary if the process is unstable. The next rule is concerned with error diagnosis.

```
(rule explain-errors-11
  (if
    (not (the resulting transfer function is
          confirmed by the a priori knowledge))
    (the system is digital)
    (a priori knowledge about the transfer
     function is available)
    (* the transfer function is wrong close to
       the sampling frequency))
  (then
    (it is not possible to get good results close
     to the sampling frequency)
    (action (lower the upper frequency limit and
             rerun the experiment))
    (an error has been diagnosed)))
```

This last example shows a rule that tries to explain erroneous results due to sampling. The error diagnosis rules are tried in order from more specific to more general error, until a likely one is found. The expert system then reports this diagnosis as the probable cause of error.

Some Example Runs

The expert system communicates with the user through a Q/A dialog. The questions will concern the user's knowledge about the process and the results of the experiments. Here is an example of part of a typical dialog.

```
Is it true that the frequency range is known a
priori? (y/n)
n
Is it true that you have a good guess about the
limits of the frequency range? (y/n)
n
Is it true that a step response frequency range
experiment has been performed? (y/n)
n
PERFORM:            perform a step response
                    frequency range experiment
Deduction/advice:   the results of a step response
                    experiment are available
Deduction/advice:   the cutoff frequency is appr.
                    omega = 1 / rise time
Deduction/advice:   center the frequency range
                    around the cutoff frequency
Deduction/advice:   checking of the frequency
                    range is needed afterwards
Deduction/advice:   the frequency range is known
```

This example shows how the system tries to establish a frequency range for the experiment. First, it checks whether the user already knows or has an educated guess about this range. After negative answers, the system proposes a step test experiment, and uses the result to guess a reasonable frequency range. This range may be changed later on, though. The fact (checking of the frequency range is needed afterwards) actually forces the system to consider changing the range after the experiment has been performed.

The next dialog part shows the error diagnosis of a session.

```
Is it true that the resulting transfer function
is confirmed by the a priori knowledge? (y/n)
n
Is it true that the amplitude is as small as the
noise or quantization level? (y/n)
n
Is it true that the resulting transfer function
is wrong in the lower frequencies? (y/n)
n
Is it true that the resulting transfer function
is wrong in the higher frequencies? (y/n)
n
Is it true that the signal to noise ratio is low
in the higher frequencies? (y/n)
n
Deduction/advice:   the friction probably destroys
                    the experiment
PERFORM:            rerun the experiment with a
                    bias, to avoid the friction
Deduction/advice:   the signals are not linear
Deduction/advice:   an error has been diagnosed
```

Here, the experiment has already been performed, and at a certain stage the validation goes wrong. Therefore the system tries several guesses of what might be the problem. Earlier it has been established that some unknown friction is present. As no specific explanation works, the system shows good expertise, and blames the unknown friction.

The Inference Engine

The expert system shell used is the MESS system, Larsson (1988). It is a small, (192 lines of source code), Scheme system that provides the user with forward and backward chaining. The backward chaining strategy is used in FREX. Some changes in the source code of MESS have been made. All fact deductions are traced, and there are some new clauses allowed in the then parts of the rules. The input and output has been changed so that the Q/A dialog gives an impression of semi-natural language. This demands the facts to be written in appropriate syntactic versions, i.e., all facts to be asked must fit after the phrase "Is it true that...".

The MESS system has obvious strengths in that its source code is short, simple and easy to change. Its main drawback is that it does not use an effective matching strategy, which makes its execution slow for larger rule bases. As the present rule base is rather small, the speed has not been a problem, though.

The FREX system was originally implemented in Chez Scheme on a SUN 3/50, but it has also been moved to PC Scheme on an IBM PC/AT, and to MIT Scheme on a Macintosh+. The response time on the SUN is immediate, and on the AT it is a matter of a few seconds. On the Macintosh+, however, FREX is quite slow, often using close to a minute per query.

Conclusions and Further Developments

The chosen target domain, frequency response analysis using a frequency analyser, is a good, clean area for an expert system project. There are very few interactions with other areas; most of the tasks can be solved at the site of the frequency analyser. This has made it possible to produce a rather complete knowledge database, and a system that is useful for most non-experts. There exists only a few written descriptions of both the theoretical and *practical* problems of frequency response analysis, Åström (1975) is an exception, but this knowledge may now be found in this system. Thus, the project has also resulted in valuable knowledge refinement.

The system probably needs more testing before it will be robust enough for production use or inclusion in a frequency analyser. Currently, it runs completely stand-alone. Work has been done to improve the front-ends of frequency analysers, see for example Wichtel (1989), and a better way would be to embed the system in the analyser, making it an integral part of an intelligent instrument.

Acknowledgements

I am indebted to Karl Johan Åström, who gave the identification course and forced his students to try a little knowledge acquisition, and to Jan Peter Axelsson, Ola Dahl, Kjell Gustafsson, Ulf Hagberg, Mats Lilja, Michael Lundh, Bernt Nilsson, Per Persson, Anders Wallenborg, Karl-Erik Årzén, and Ann-Britt Östberg, all of whom produced written recommendations on frequency response analysis in general and running the Solartron 1250 analyser in particular. Also, I would like to thank Sven Erik Mattsson for good advice during the project.

References

Larsson, J. E., (1988): "MESS—A Minimal Expert System Shell," Technical report, TFRT–7380, Department of Automatic Control, Lund Institute of Technology, Lund.

Larsson, J. E., (1989): "Knowledge-Based Frequency Response Analysis," in Larsson, J. E., (Ed.) *Proceedings of the SAIS '89 Workshop*, Department of Automatic Control, Lund Institute of Technology, Lund.

Solartron Instrumentation Group, (1983 a): "1250 Frequency Response Analyser, Operating Manual," Solartron Instrumentation Group, Schlumberger Electronics (UK) Ltd., Farnborough, England.

Solartron Instrumentation Group, (1983 b): "1096 Data Management System," Solartron Instrumentation Group, Schlumberger Electronics (UK) Ltd., Farnborough, England.

Söderström, T., (1984): *Lecture Notes in Identification*, Automatic Control and Systems Analysis Group, Department of Technology, Uppsala University, Uppsala.

Wichtel, E., (1989): "A Control Program for a Frequency Analyser," Master thesis, TFRT–5400, Department of Automatic Control, Lund Institute of Technology, Lund.

Ångström, A. I., (1861): "Neue Methode, das Värmeleitungsvermögen der Körper zu bestimmen," Annalen der Physik und Schemie, **114**, pp. 513–530.

Åström, K. J., (1975): "Lectures on System Identification, Chapter 3, Frequency Response Analysis," Internal report, TFRT–7504, Department of Automatic Control, Lund Institute of Technology, Lund.

The Rule base

The rule base of the system is not very large, and thus, it is possible to include it in its entirety.

```
(define rules '(

(rule phase-control
  (if
    (the system is stabilized)
    (the linearity range is known)
    (the frequency range is known)
    (the amplitude range is known)
    (the experiment has been performed)
    (the result is verified))
  (then
    (the frequency analysis has been performed)))

(rule stabilize-1
  (if
    (* the system is stable))
  (then
    (the system is stabilized)))

(rule stabilize-2
  (if
    (not (the system is stable))
    (* the system can be stabilized with a feedback loop))
  (then
    (action (use a proportional regulator to stabilize the system))
    (action (connect the input to the reference signal of the controller))
    (the system is stabilized)))

(rule linearity-1
  (if
    (* the linearity range is known a priori))
  (then
    (the linearity range is known)))

(rule linearity-2
  (if
    (a visual inspection of the signals has been attempted)
    (the results of a dual step response experiment are available)
    (the results of a small steps test are available)
    (the friction of the system is known))
  (then
    (use the results of the experiments to determine a linearity range)
    (the linearity range is known)))

(rule linearity-3
  (if
    (* a visual inspection of the signals is possible))
  (then
    (remove (a visual inspection of the signals is possible))
    (action (try to find an amplitude where the signals are linear))
    (a visual inspection of the signals has been attempted)))

(rule linearity-4
  (if
    (not (a visual inspection of the signals is possible)))
  (then
    (remove (not (a visual inspection of the signals is possible)))
    (a visual inspection of the signals has been attempted)))

(rule linearity-5
  (if
    (* a dual step response experiment has been performed))
  (then
    (remove (a dual step response experiment has been performed))
    (the results of a dual step response experiment are available)))

(rule linearity-6
```

```
  (if
    (not (a dual step response experiment has been performed)))
  (then
    (remove (not (a dual step response experiment has been performed)))
    (action (perform a dual step response experiment))
    (the results of a dual step response experiment are available)))

(rule linearity-7
  (if
    (* small steps at different set points have been tested))
  (then
    (remove (small steps at different set points have been tested))
    (the results of a small steps test are available)))

(rule linearity-8
  (if
    (not (small steps at different set points have been tested)))
  (then
    (remove (not (small steps at different set points have been tested)))
    (action (perform a few small steps at different set points))
    (the results of a small steps test are available)))

(rule linearity-9
  (if
    (* there is some known coulomb friction in the system))
  (then
    (you should consider using a bias in the experiment to avoid the friction)
    (the friction of the system is known)))

(rule linearity-10
  (if
    (* there may be coulomb friction in the system))
  (then
    (the friction of the system is known)))

(rule linearity-11
  (if
    (not (there is some known coulomb friction in the system)))
  (then
    (the friction of the system is known)))

(rule frequency-range-1
  (if
    (* the frequency range is known a priori))
  (then
    (not (checking of the frequency range is needed afterwards))
    (the frequency range is known)))

(rule frequency-range-2
  (if
    (* you have a good guess about the limits of the frequency range))
  (then
    (checking of the frequency range is needed afterwards)
    (the frequency range is known)))

(rule frequency-range-3
  (if
    (the results of a step response experiment are available))
  (then
    (the cutoff frequency is approximately "omega = 1 / rise time")
    (center the frequency range around the cutoff frequency)
    (checking of the frequency range is needed afterwards)
    (the frequency range is known)))

(rule frequency-range-4
  (if
    (* a step response frequency range experiment has been performed))
  (then
    (remove (a step response frequency range experiment has been performed))
    (the results of a step response experiment are available)))
```

```
(rule frequency-range-5
  (if
    (not (a step response frequency range experiment has been performed)))
  (then
    (remove (not (a step response frequency range experiment has been performed)))
    (action (perform a step response frequency range experiment))
    (the results of a step response experiment are available)))

(rule amplitude-range-1
  (if
    (the lower amplitude limit is known)
    (the upper amplitude limit is known))
  (then
    (the amplitude range is known)))

(rule amplitude-range-2
  (if
    (* the amplitude range is known a priori))
  (then
    (the lower amplitude limit is known)
    (the upper amplitude limit is known)))

(rule amplitude-range-3
  (if
    (the noise level is known)
    (* the system is digital))
  (then
    (the amplitude should be well over the noise level)
    (the amplitude should be well over the quantization level)
    (the lower amplitude limit is known)))

(rule amplitude-range-4
  (if
    (the noise level is known)
    (not (the system is digital)))
  (then
    (the amplitude should be well over the noise level)
    (the lower amplitude limit is known)))

(rule amplitude-range-5
  (if
    (* the noise level has been measured or estimated))
  (then
    (remove (the noise level has been measured or estimated))
    (the noise level is known)))

(rule amplitude-range-6
  (if
    (not (the noise level has been measured or estimated)))
  (then
    (remove (not (the noise level has been measured or estimated)))
    (action (measure or estimate the noise level of the system))
    (the noise level is known)))

(rule amplitude-range-7
  (if
    (* there is an estimate of the gain of the transfer function))
  (then
    (remove (there is an estimate of the gain of the transfer function))
    (the amplitude will be limited by the static gain of the transfer function)
    (the upper amplitude limit is known)))

(rule amplitude-range-8
  (if
    (not (there is an estimate of the gain of the transfer function)))
  (then
    (remove (not (there is an estimate of the gain of the transfer function)))
    (action (measure the static gain of the system))
    (the amplitude will be limited by the static gain of the transfer function)
```

```
    (the upper amplitude limit is known)))

(rule perform-experiment-1
  (if
    (the cover all frequencies experiment has been performed)
    (the cover interesting frequencies experiment has been performed))
  (then
    (the experiment has been performed)))

(rule perform-experiment-2
  (if
    (* an experiment covering all frequencies has been performed))
  (then
    (remove (an experiment covering all frequencies has been performed))
    (the cover all frequencies experiment has been performed)))

(rule perform-experiment-3
  (if
    (not (an experiment covering all frequencies has been performed)))
  (then
    (remove (not (an experiment covering all frequencies has been performed)))
    (action (pick 10 frequencies covering the whole frequency range))
    (action (choose a rather small amplitude in the amplitude range))
    (action (run the experiment with increasing frequencies))
    (the cover all frequencies experiment has been performed)))

(rule perform-experiment-4
  (if
    (* an experiment covering interesting frequencies has been performed))
  (then
    (remove (an experiment covering interesting frequencies has been performed))
    (the cover interesting frequencies experiment has been performed)))

(rule perform-experiment-5
  (if
    (not (an experiment covering interesting frequencies has been performed))
    (* there are some points where the transfer function changes rapidly))
  (then
    (remove (there are some points where the transfer function changes rapidly))
    (remove (not (an experiment covering interesting frequencies has been performed)))
    (pick 10 frequencies around the interesting points)
    (choose the amplitude that worked best in the earlier runs)
    (the amplitude must not be too large at resonance frequencies)
    (action (run the experiment over the interesting frequencies))
    (the cover interesting frequencies experiment has been performed)))

(rule perform-experiment-6
  (if
    (not (an experiment covering interesting frequencies has been performed))
    (not (there are some points where the transfer function changes rapidly)))
  (then
    (remove (not (there are some points where the transfer function changes rapidly)))
    (the cover interesting frequencies experiment has been performed)))

(rule verify-1
  (if
    (the a priori knowledge has been checked)
    (the frequency range has been checked)
    (the simulation verifies the experiment))
  (then
    (the result is verified)))

(rule verify-2
  (if
    (* a priori knowledge about the transfer function is available)
    (* the resulting transfer function is confirmed by the a priori knowledge))
  (then
    (the a priori knowledge has been checked)))

(rule verify-3
```

```
  (if
    (not (a priori knowledge about the transfer function is available)))
  (then
    (the a priori knowledge has been checked)))

(rule verify-4
  (if
    (not (checking of the frequency range is needed afterwards)))
  (then
    (the frequency range has been checked)))

(rule verify-5
  (if
    (checking of the frequency range is needed afterwards)
    (the lower frequency limit has been checked)
    (the upper frequency limit has been checked))
  (then
    (the frequency range has been checked)))

(rule verify-6
  (if
    (checking of the frequency range is needed afterwards)
    (* there seems to be interesting dynamics at lower frequencies))
  (then
    (remove (there seems to be interesting dynamics at lower frequencies))
    (consider using a lower frequency limit)
    (the lower frequency limit has been checked)))

(rule verify-7
  (if
    (checking of the frequency range is needed afterwards)
    (not (there seems to be interesting dynamics at lower frequencies)))
  (then
    (remove (not (there seems to be interesting dynamics at lower frequencies)))
    (the lower frequency limit has been checked)))

(rule verify-8
  (if
    (checking of the frequency range is needed afterwards)
    (* there seems to be interesting dynamics at higher frequencies))
  (then
    (remove (there seems to be interesting dynamics at higher frequencies))
    (consider using a higher frequency limit)
    (the upper frequency limit has been checked)))

(rule verify-9
  (if
    (checking of the frequency range is needed afterwards)
    (not (there seems to be interesting dynamics at higher frequencies)))
  (then
    (remove (not (there seems to be interesting dynamics at higher frequencies)))
    (the upper frequency limit has been checked)))

(rule verify-10
  (if
    (the results of a simulation experiment are available)
    (* the simulation confirms the result of the experiment))
  (then
    (the simulation verifies the experiment)))

(rule verify-11
  (if
    (* a simulation experiment has been performed))
  (then
    (remove (a simulation experiment has been performed))
    (the results of a simulation experiment are available)))

(rule verify-12
  (if
    (not (a simulation experiment has been performed))
```

```
    (* it is possible to perform a simulation experiment))
  (then
    (remove (not (a simulation experiment has been performed)))
    (remove (it is possible to perform a simulation experiment))
    (action (run a simulation the system))
    (the results of a simulation experiment are available)))

(rule verify-13
  (if
    (not (the results of a simulation experiment are available))
    (not (it is possible to perform a simulation experiment)))
  (then
    (remove (not (a simulation experiment has been performed)))
    (remove (not (it is possible to perform a simulation experiment)))
    (the simulation verifies the experiment)))

(rule explain-errors-1
  (if
    (not (the system is stable))
    (not (the system can be stabilized with a feedback loop)))
  (then
    (it is impossible to perform a frequency response analysis)
    (an error has been diagnosed)))

(rule explain-errors-2
  (if
    (not (the system is stable))
    (the system can be stabilized with a feedback loop)
    (a priori knowledge about the transfer function is available))
  (then
    (it is probable that the inverse of the controller has been estimated)
    (action (try to suppress disturbances or use a larger amplitude))
    (action (try another controller or feed the input directly to the process input))
    (an error has been diagnosed)))

(rule explain-errors-3
  (if
    (the amplitude is too small))
  (then
    (an error has been diagnosed)))

(rule explain-errors-4
  (if
    (* the amplitude is as small as the noise or quantization level))
  (then
    (action (rerun the experiment with a larger amplitude or longer integration time))
    (the amplitude is too small)))

(rule explain-errors-5
  (if
    (the amplitude is too large))
  (then
    (an error has been diagnosed)))

(rule explain-errors-6
  (if
    (* a visual inspection of the signals is possible)
    (* the signals reaches amplitude limitations))
  (then
    (action (rerun the experiment with a smaller amplitude))
    (longer integration times may be necessary)
    (the amplitude is too large)))

(rule explain-errors-7
  (if
    (the frequency range used is too large))
  (then
    (an error has been diagnosed)))

(rule explain-errors-8
```

```
  (if
    (* the resulting transfer function is wrong in the lower frequencies))
  (then
    (the lower frequency bound is too small)
    (the frequency range used is too large)))

(rule explain-errors-9
  (if
    (* the resulting transfer function is wrong in the higher frequencies))
  (then
    (the upper frequency bound is too large)
    (the frequency range used is too large)))

(rule explain-errors-10
  (if
    (* the signal to noise ratio is low in the higher frequencies)
    (* the resulting transfer function is wrong in the higher frequencies))
  (then
    (the upper frequency bound is too large)
    (action (lower the upper frequency bound or use a larger amplitude))
    (the frequency range used is too large)))

(rule explain-errors-11
  (if
    (not (the resulting transfer function is confirmed by the a priori knowledge))
    (the system is digital)
    (a priori knowledge about the transfer function is available)
    (* the transfer function is wrong close to the sampling frequency))
  (then
    (it is not possible to get good results close to the sampling frequency)
    (action (lower the upper frequency limit and rerun the experiment))
    (an error has been diagnosed)))

(rule explain-errors-12
  (if
    (the signals are not linear))
  (then
    (an error has been diagnosed)))

(rule explain-errors-13
  (if
    (there is some known coulomb friction in the system))
  (then
    (the friction probably destroys the experiment)
    (action (rerun the experiment with a bias to avoid the friction))
    (the signals are not linear)))

(rule explain-errors-14
  (if
    (there may be coulomb friction in the system))
  (then
    (it is probable that there is some friction destroying the experiment)
    (action (rerun the experiment with a bias to avoid the friction))
    (the signals are not linear)))

(rule explain-errors-15
  (if
    (* there may be backlash or hysteresis in the system))
  (then
    (the system is non-linear)
    (it is probably not possible to use frequency response analysis)
    (the signals are not linear)))

(rule explain-errors-16
  (if
    (* the signals have been drifting slightly during the experiment))
  (then
    (the signals may have left the linearity range)
    (action (rerun the experiment with a stabilizing feedback))
    (the signals are not linear)))
```

```
(rule explain-errors-17
  (if
    (* a visual inspection of the signals is possible)
    (* the signals do not have a sinusoidal form))
  (then
    (action (rerun the experiment with a smaller amplitude))
    (longer integration times may be necessary)
    (the signals are not linear)))

(rule explain-errors-18
  (if
    (a priori knowledge about the transfer function is available)
    (* there may be some unknown disturbances))
  (then
    (the disturbances may destroy the transfer function)
    (action (try to eliminate the disturbances and rerun the experiment))
    (an error has been diagnosed)))

(rule explain-errors-19
  (if
    (a priori knowledge about the transfer function is available))
  (then
    (this is a difficult case)
    (the a priori knowledge may be wrong)
    (the model and the real system may be inconsistent)
    (an error has been diagnosed)))

))
```

Example Runs

The best way of showing what the system can do is to give a few examples. In the following dialogues, the user has typed the y's and n's, the rest is produced by the expert system. The listings were copied from terminal sessions with the `script` facility of Unix.

In the first example, very little is known from start, so all of the experiments are used.

```
Chez Scheme Version 2.0.1 Copyright (c) 1987 R. Kent Dybvig

> (load "frex.scm")
Is it true that the system is stable? (y/n)
y
Deduction/advice:  the system is stabilized
Is it true that the linearity range is known a priori? (y/n)
n
Is it true that a visual inspection of the signals is possible? (y/n)
y
PERFORM:           try to find an amplitude where the signals are linear
Deduction/advice:  a visual inspection of the signals has been attempted
Is it true that a dual step response experiment has been performed? (y/n)
n
PERFORM:           perform a dual step response experiment
Deduction/advice:  the results of a dual step response experiment are available
Is it true that small steps at different set points have been tested? (y/n)
n
PERFORM:           perform a few small steps at different set points
Deduction/advice:  the results of a small steps test are available
Is it true that there is some known coulomb friction in the system? (y/n)
n
Is it true that there may be coulomb friction in the system? (y/n)
n
Deduction/advice:  the friction of the system is known
Deduction/advice:  use the results of the experiments to determine a linearity range
Deduction/advice:  the linearity range is known
Is it true that the frequency range is known a priori? (y/n)
n
Is it true that you have a good guess about the limits of the frequency range? (y/n)
n
Is it true that a step response frequency range experiment has been performed? (y/n)
```

```
n
PERFORM:            perform a step response frequency range experiment
Deduction/advice:   the results of a step response experiment are available
Deduction/advice:   the cutoff frequency is approximately omega = 2 pi / rise time
Deduction/advice:   center the frequency range around the cutoff frequency
Deduction/advice:   checking of the frequency range is needed afterwards
Deduction/advice:   the frequency range is known
Is it true that the amplitude range is known a priori? (y/n)
n
Is it true that the noise level has been measured or estimated? (y/n)
n
PERFORM:            measure or estimate the noise level of the system
Deduction/advice:   the noise level is known
Is it true that the system is digital? (y/n)
n
Deduction/advice:   the amplitude should be well over the noise level
Deduction/advice:   the lower amplitude limit is known
Is it true that there is an estimate of the gain of the transfer function? (y/n)
n
PERFORM:            measure the static gain of the system
Deduction/advice:   the amplitude will be limited by the static gain of the transfer function
Deduction/advice:   the upper amplitude limit is known
Deduction/advice:   the amplitude range is known
Is it true that an experiment covering all frequencies has been performed? (y/n)
n
PERFORM:            pick 10 frequencies covering the whole frequency range
PERFORM:            choose a rather small amplitude in the amplitude range
PERFORM:            run the experiment with increasing frequencies
Deduction/advice:   the cover all frequencies experiment has been performed
Is it true that an experiment covering interesting frequencies has been performed? (y/n)
n
Is it true that there are some points where the transfer function changes rapidly? (y/n)
y
Deduction/advice:   pick 10 frequencies around the interesting points
Deduction/advice:   choose the amplitude that worked best in the earlier runs
Deduction/advice:   the amplitude must not be too large at resonance frequencies
PERFORM:            run the experiment over the interesting frequencies
Deduction/advice:   the cover interesting frequencies experiment has been performed
Deduction/advice:   the experiment has been performed
Is it true that a priori knowledge about the transfer function is available? (y/n)
n
Deduction/advice:   the a priori knowledge has been checked
Is it true that there seems to be interesting dynamics at lower frequencies? (y/n)
n
Deduction/advice:   the lower frequency limit has been checked
Is it true that there seems to be interesting dynamics at higher frequencies? (y/n)
n
Deduction/advice:   the upper frequency limit has been checked
Deduction/advice:   the frequency range has been checked
Is it true that a simulation experiment has been performed? (y/n)
n
Is it true that it is possible to perform a simulation experiment? (y/n)
y
PERFORM:            run a simulation the system
Deduction/advice:   the results of a simulation experiment are available
Is it true that the simulation confirms the result of the experiment? (y/n)
y
Deduction/advice:   the simulation verifies the experiment
Deduction/advice:   the result is verified
Deduction/advice:   the frequency analysis has been performed

()
```

In the second example, the user gets an unexpected result, i.e., the a priori result contradicts the experiment. The system tries several explanations before it finds a plausible one.

```
Chez Scheme Version 2.0.1 Copyright (c) 1987 R. Kent Dybvig

> (load "frex.scm")
Is it true that the system is stable? (y/n)
```

```
y
Deduction/advice:  the system is stabilized
Is it true that the linearity range is known a priori? (y/n)
n
Is it true that a visual inspection of the signals is possible? (y/n)
y
PERFORM:           try to find an amplitude where the signals are linear
Deduction/advice:  a visual inspection of the signals has been attempted
Is it true that a dual step response experiment has been performed? (y/n)
n
PERFORM:           perform a dual step response experiment
Deduction/advice:  the results of a dual step response experiment are available
Is it true that small steps at different set points have been tested? (y/n)
n
PERFORM:           perform a few small steps at different set points
Deduction/advice:  the results of a small steps test are available
Is it true that there is some known coulomb friction in the system? (y/n)
y
Deduction/advice:  you should consider using a bias in the experiment to avoid the friction
Deduction/advice:  the friction of the system is known
Deduction/advice:  use the results of the experiments to determine a linearity range
Deduction/advice:  the linearity range is known
Is it true that the frequency range is known a priori? (y/n)
n
Is it true that you have a good guess about the limits of the frequency range? (y/n)
y
Deduction/advice:  checking of the frequency range is needed afterwards
Deduction/advice:  the frequency range is known
Is it true that the amplitude range is known a priori? (y/n)
n
Is it true that the noise level has been measured or estimated? (y/n)
n
PERFORM:           measure or estimate the noise level of the system
Deduction/advice:  the noise level is known
Is it true that the system is digital? (y/n)
n
Deduction/advice:  the amplitude should be well over the noise level
Deduction/advice:  the lower amplitude limit is known
Is it true that there is an estimate of the gain of the transfer function? (y/n)
y
Deduction/advice:  the amplitude will be limited by the static gain of the transfer function
Deduction/advice:  the upper amplitude limit is known
Deduction/advice:  the amplitude range is known
Is it true that an experiment covering all frequencies has been performed? (y/n)
n
PERFORM:           pick 10 frequencies covering the whole frequency range
PERFORM:           choose a rather small amplitude in the amplitude range
PERFORM:           run the experiment with increasing frequencies
Deduction/advice:  the cover all frequencies experiment has been performed
Is it true that an experiment covering interesting frequencies has been performed? (y/n)
n
Is it true that there are some points where the transfer function changes rapidly? (y/n)
n
Deduction/advice:  the cover interesting frequencies experiment has been performed
Deduction/advice:  the experiment has been performed
Is it true that a priori knowledge about the transfer function is available? (y/n)
y
Is it true that the resulting transfer function is confirmed by the a priori knowledge? (y/n)
n
Is it true that the amplitude is as small as the noise or quantization level? (y/n)
n
Is it true that the resulting transfer function is wrong in the lower frequencies? (y/n)
n
Is it true that the resulting transfer function is wrong in the higher frequencies? (y/n)
n
Is it true that the signal to noise ratio is low in the higher frequencies? (y/n)
n
Deduction/advice:  the friction probably destroys the experiment
PERFORM:           rerun the experiment with a bias to avoid the friction
Deduction/advice:  the signals are not linear
```

```
Deduction/advice:  an error has been diagnosed

()
```

In the third example, the experiment has already been performed, and most facts are known. A simulation is performed, and the result conflicts with the result obtained earlier. Once again, a plausible explanation for the error is found.

```
Chez Scheme Version 2.0.1 Copyright (c) 1987 R. Kent Dybvig

> (load "frex.scm")
Is it true that the system is stable? (y/n)
n
Is it true that the system can be stabilized with a feedback loop? (y/n)
y
PERFORM:            use a proportional regulator to stabilize the system
PERFORM:            connect the input to the reference signal of the controller
Deduction/advice:   the system is stabilized
Is it true that the linearity range is known a priori? (y/n)
y
Deduction/advice:   the linearity range is known
Is it true that the frequency range is known a priori? (y/n)
y
Known to be false: checking of the frequency range is needed afterwards
Deduction/advice:   the frequency range is known
Is it true that the amplitude range is known a priori? (y/n)
y
Deduction/advice:   the lower amplitude limit is known
Deduction/advice:   the upper amplitude limit is known
Deduction/advice:   the amplitude range is known
Is it true that an experiment covering all frequencies has been performed? (y/n)
y
Deduction/advice:   the cover all frequencies experiment has been performed
Is it true that an experiment covering interesting frequencies has been performed? (y/n)
y
Deduction/advice:   the cover interesting frequencies experiment has been performed
Deduction/advice:   the experiment has been performed
Is it true that a priori knowledge about the transfer function is available? (y/n)
y
Is it true that the resulting transfer function is confirmed by the a priori knowledge? (y/n)
y
Deduction/advice:   the a priori knowledge has been checked
Deduction/advice:   the frequency range has been checked
Is it true that a simulation experiment has been performed? (y/n)
n
Is it true that it is possible to perform a simulation experiment? (y/n)
y
PERFORM:            run a simulation the system
Deduction/advice:   the results of a simulation experiment are available
Is it true that the simulation confirms the result of the experiment? (y/n)
n
Deduction/advice:   it is probable that the inverse of the controller has been estimated
PERFORM:            try to suppress disturbances or use a larger amplitude
PERFORM:            try another controller or feed the input directly to the process input
Deduction/advice:   an error has been diagnosed

()
```

The Inference Engine

The expert system shell used is the MESS system, Larsson (1988). It is a small Scheme system that provides the user with forward and backward chaining. Some changes to the source code has been made to fit the FREX project. As the code including changes is quite small, it is included here in its entirety.

```
;---FREX---An Expert System for Frequency Response Analysis--------------------
;
;   Author: Jan Eric Larsson
;           Department of Automatic Control
;           Lund Institute of Technology
```

```
;           Box 118, S-221 00 LUND, Sweden
;           Phone: +46 46 108795, Usenet: janeric@Control.LTH.Se
;
;---Miscellaneous stuff---------------------------------------------------------

(define (displaylist list)
  (display (car list))
  (if (cdr list)
    (begin (display " ") (displaylist (cdr list)))))

;---Rule selectors--------------------------------------------------------------

(define (getifs rule)
  (cdr (assoc 'if (cddr rule))))

(define (getthens rule)
  (cdr (assoc 'then (cddr rule))))

(define (gettests rule)
  (if (null? (assoc 'test (cddr rule)))
    nil
    (cdr (assoc 'test (cddr rule)))))

;---Fact manipulation-----------------------------------------------------------

(define (var? e)
  (and
    (symbol? e)
    (equal? (substring (symbol->string e) 0 1) "-")
    (not (equal? e '-))))

(define (fillin fact env)
  (cond
    ((null? fact) nil)
    ((and
      (var? (car fact))
      (assoc (car fact) env))
      (cons (cadr (assoc (car fact) env)) (fillin (cdr fact) env)))
    ((atom? (car fact)) (cons (car fact) (fillin (cdr fact) env)))
    (t (cons (fillin (car fact) env) (fillin (cdr fact) env)))))

(define (evalstuff fact)
  (cond
    ((null? fact) nil)
    ((equal? (car fact) '^) (cons (eval (cadr fact)) (evalstuff (cddr fact))))
    (t (cons (car fact) (evalstuff (cdr fact))))))

(define (addnewfacts fact env)
  (let ((f (evalstuff (fillin fact env))))
    (cond
      ((equal? (car f) 'remove)
        (if (member (cadr f) facts) (set! facts (remove! (cadr f) facts)) nil))
      ((equal? (car f) 'call) (eval (cadr f)) nil)
      ((equal? (car f) 'ask) (ask (cons '* (cadr f)) env))
      ((equal? (car f) 'action)
        (display "PERFORM:           ")
        (displaylist (cadr f))
        (newline)
        (set! facts (cons f facts)))
      ((equal? (car f) 'not)
        (display "Known to be false: ")
        (displaylist (cadr f))
        (newline)
        (set! facts (cons f facts)))
      (t
        (display "Deduction/advice:  ")
        (displaylist fact)
        (newline)
        (set! facts (cons f facts))))))
```

```
;---Unification-----------------------------------------------------------------

(define (unify p d env)
  (cond
    ((and (null? p) (null? d)) env)
    ((or (null? p) (null? d)) nil)
    ((or (equal? (car p) '-) (equal? (car d) '-)) (unify (cdr p) (cdr d) env))
    ((equal? (car p) '*) (unify (cdr p) d env))
    ((equal? (car p) (car d)) (unify (cdr p) (cdr d) env))
    ((not (var? (car p))) nil)
    ((null? (assoc (car p) env))
      (unify (cdr p) (cdr d) (cons (list (car p) (car d)) env)))
    ((equal? (cadr (assoc (car p) env)) (car d)) (unify (cdr p) (cdr d) env))
    (t nil)))

;---Forward chaining------------------------------------------------------------

(define (getfact fact factlist env)
  (cond
    ((equal? (car fact) '*) (getfact (cdr fact) factlist env))
    ((null? factlist) (if (equal? (car fact) 'not) (list nil env) nil))
    ((equal? (car fact) 'not)
      (if (null? (getfact (cadr fact) factlist env))
        (list (cdr factlist) env)
        nil))
    ((let ((newe (unify fact (car factlist) env)))
      (if newe (list (cdr factlist) newe) nil)))
    (t (getfact fact (cdr factlist) env))))

(define (putfact fact factlist env)
  (cond
    ((null? factlist) (addnewfacts fact env))
    ((unify fact (car factlist) env) nil)
    (t (putfact fact (cdr factlist) env))))

(define (testif ifs rule env)
  (if ifs
    (do ((e (list facts env)) (newe t) (retval nil))
      ((or (null? newe) (equal? newe '(() (())))) retval) retval)
      (set! newe (getfact (car ifs) (car e) (cadr e)))
      (if newe
        (begin
          (set! e (list (car newe) (cadr e)))
          (set! retval (testif (cdr ifs) rule (cadr newe))))))
    (checktest rule env)))

(define (checktest rule env)
  (do
    ((tests (gettests rule) (cdr tests)))
    ((or (null? tests) (not (eval (fillin (car tests) env))))
      (if tests nil (usethen rule env)))))

(define (usethen rule env)
  (do
    ((thens (getthens rule) (cdr thens)) (retval nil))
    ((null? thens) (if retval (list nil retval) nil))
    (if (putfact (car thens) facts env) (set! retval env))))

(define (findrule)
  (do
    ((rlist rules (cdr rlist)))
    ((or (null? rlist) (testif (getifs (car rlist)) (car rlist) '(())))
      (if (and trace rlist)
        (begin
          (display "Rule ")
          (display (cadar rlist))
          (display " triggered")
          (newline)))
      (if (null? rlist) nil t))))
```

```
(define (fc) (if (findrule) (fc)) 'Ok)

;---Backward chaining----------------------------------------------------------

(define (verifyif ifs rule env)
  (if ifs
    (let ((e (verify (car ifs) facts env)))
      (if e (verifyif (cdr ifs) rule (cadr e))))
    (checktest rule env)))

(define (inthens? fact thens env)
  (cond
    ((null? thens) nil)
    ((unify (car thens) fact env))
    (t (inthens? fact (cdr thens) env))))

(define (userule fact env)
  (do
    ((rlist rules (cdr rlist)) (newe nil))
    ((or (null? rlist) newe) newe)
    (set! newe (inthens? fact (getthens (car rlist)) env))
    (if newe (set! newe (verifyif (getifs (car rlist)) (car rlist) newe)))
    (if (and trace newe)
      (begin
        (display "Rule ")
        (display (cadar rlist))
        (display " triggered")
        (newline)))))

(define (ask fact env)
  (cond
    ((and
      (equal? (car fact) '*)
      (null? (getfact (list 'asked (cdr fact)) facts env)))
      (set! facts (cons (list 'asked (cdr fact)) facts))
      (let ((f (cdr (fillin fact env))))
        (display "Is it true that ")
        (displaylist (if (equal? (car f) 'not) (cadr f) f))
        (display "? (y/n)")
        (newline)
        (if (equal? (read) 'y)
          (if (equal? (car f) 'not)
            (begin (set! facts (cons (cadr f) facts)) nil)
            (begin (set! facts (cons f facts)) (list facts env)))
          (if (equal? (car f) 'not)
            (begin (set! facts (cons f facts)) (list facts env))
            (begin (set! facts (cons (list 'not f) facts)) nil)))))
    (t nil)))

(define (verify fact factlist env)
  (cond
    ((getfact fact factlist env))
    ((userule fact env))
    ((ask fact env))))

(define (bc)
  (do
    ((h hypotheses (cdr h)))
    ((or (null? h) (verify (car h) facts '((()))))
      (if (null? h)
        (display "The experiment failed without any error diagnosis"))
      (newline)))
  'Ok)

;---Main program stuff---------------------------------------------------------

(define trace nil)
(define rules nil)
(define facts nil)
(define hypotheses nil)
```

```
(define (frex)
  (set! facts nil)
  (set! hypotheses '(
    (the frequency analysis has been performed)
    (an error has been diagnosed)))
  (bc))
```

The expert system is started by loading the file `frex.scm` into the Scheme system. This file loads the inference engine and the rules, and starts the system. The inference engine resides in the file `mess.scm`, and the rules in the file `rules.scm`. This is the contents of the `frex.scm` file.

```
(load "mess.scm")
(load "rules.scm")
(frex)
```

If you want to run the system, do the following on a SUN. First, `cd /sperry/janeric/ai/frex`, then start Scheme with `scheme`, and then `(load "frex.scm")`.