LUND UNIVERSITY

# An Expert System Interface for Idpac

## Paper presented at Santa Barbara '85

Larsson, Jan Eric; Åström, Karl Johan

1985

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

Link to publication

*Citation for published version (APA):*
Larsson, J. E., & Åström, K. J. (1985). *An Expert System Interface for Idpac: Paper presented at Santa Barbara '85.* (Technical Reports TFRT-7308). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
2

**General rights**
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

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

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

*Document name*
Internal Report

*Date of issue*
November 1985

*Document Number*
CODEN: LUTFD2/(TFRT-7308)/1-21/(1985)

*Author(s)*
Jan Eric Larsson
Karl Johan Åström

*Supervisor*

*Sponsoring organisation*

*Title and subtitle*
An Expert System Interface for Idpac — Paper presented at Santa Barbara '85

*Abstract*

This report contains an abstract and viewgraphs for a paper presented at the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design, held in Santa Barbara, California, in March 13-15, 1985.

*Key words*
Expert Systems, Expert Interfaces, Man-machine interfaces, System Indentification

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*

*ISBN*

*Language*
English

*Number of pages*
21

*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.*

## Contents

# An Expert System Interface for Idpac

Jan Eric Larsson & Karl Johan Åström

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

Presented at the 2nd IEEE Control Systems Society Symposium on
Computer-Aided Control System Design,
Santa Barbara, California, March 13-15, 1985.

This paper deals with the design of an expert system interface for system identification using the interactive identification package "Idpac". The problem of system identification is well defined. There are several experts available in the area. Idpac and its commands are both general and well defined. The problem is large enough to be nontrivial, and at the same time not too large to be handled. Therefore this problem seems to be well fitted for an expert system solution.

The process of system identification is discussed, [1, 2]. Different matters, such as experiment planning, selection of model structure, parameter estimation methods and validation of results are mentioned, together with a brief description of the interactive program package Idpac. This program supports many useful identification methods, including spectral and correlation analysis and least squares and maximum likelihood fitting of parameters. Many services are offered, e.g. generation of signals, filtering, fast Fourier transforms, graphic plotting of data, etc. Idpac, as well as several other program packages, performing different tasks concerning automatic control, is built around the interaction module Intrac, which is a module and a language for implementing interaction. All the packages are written in Fortran. They have been developed at the Department of Automatic Control, Lund Institute of Technology, [3, 4].

The different user types discussed are the expert, the casual user, the beginner and the assistant. The knowledge and the needs of these user types are very different, and it is concluded that an expert interface should be able to handle these varying needs, and thus, be able to treat different users in different ways, [3].

An overview of the data types and representations needed in an expert interface for system identification is given. The data types are numerical data, and knowledge about many different things, such as the history, present condition and reliability of numerical data, prior knowledge about the process and the experiments performed, knowledge about Idpac and its internal states, and knowledge about the intelligent parts of the system, the knowledge base, the rules, what the system is able to do, etc.

The concept of knowledge representation frames is described, and an example of a frame system is given, [5, 6].

Production rules and techniques for administering them are illustrated by an example, where the function of a small system is described, [5]. The system is very small and primitive, and several enhancements to it are proposed. Tools such as a smart rule editor and methods for structuring the rule base are mentioned. Reasons for choosing an existing framework and the requirements are also discussed.

A short discussion of different interfaces is given. The concept of scripts, [7], is used to construct a command spy, a system that reads the user commands sent to Idpac, and tries to keep track of what the user is trying to do and how well he succeeds. The subjects of help systems, automatic documentation, command statistics and adaptivity to the user's needs are also mentioned, and this latter thing is pointed out as being quite important.

The overall structure of an expert interface for system identification is outlined, and a few different alternatives to split the implementation project into parts are also discussed. E.g. the interaction part called the command spy could be left out, or run by itself, or could the part of the system that keeps trace of the numerical data be implemented in isolation.

The demands of a framework for building an expert interface are stated. They include a powerful data structure, good interaction, flexibility, simplicity and speed. Some existing frameworks are described and found to be useful, although not perfect. The problems of designing a framework are briefly overviewed, and the conclusion is that it is probably best to experiment with one of the smaller, existing frameworks, and then either apply any changes needed to it, or pick parts from it in the process of constructing a new system.

A small set of rules for system identification has been developed and run on a test system. The development of the rules is described, and some examples of rules and runs are given. The rule base needed for running Idpac is much larger than the one implemented. At present there are 68 rules, taking care of five out of forty commands. Approximately 1000 rules are needed to cope with most of the things that Idpac currently is used for.

The paper also shows that the different parts of a knowledge based interface are rather loosely connected, so that they may be implemented and tested independently.

The main conclusion is that it indeed seems possible to develop an expert interface for Idpac, and that such a system would be a very useful asset.

References

1. Eykhoff, ***System Identification. Parameter and State Estimation.*** John Wiley & Sons, London, 1974.
2. Åström, ***Modeling and Simulation Techniques***, Agard Lecture Series No. 128, 1983.
3. Wieslander, ***Interaction in Computer Aided Analysis and Design of Control Systems***, Department of Automatic Control, Lund Institute of Technology, 1979.

4. Åström, ***Computer Aided Modeling, Analysis and Design of Control Systems – A Perspective***, Control Systems Magazine, Vol. 3, No. 2, May 1983, pp. 4-16, 1983.
5. Minsky, ***Steps Toward Artificial Intelligence***, In Feigenbaum, Feldman, (Eds.), ***Computers and Thought***, New York, 1963.
6. Winston, Horn, ***Lisp***, Addison-Wesley, Reading, Massachusetts, 1981.
7. Schank, Riesbeck, ***Inside Computer Understanding***, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981.

# An Expert System Interface for Idpac

by

Jan Eric Larsson
and
Karl Johan Åström

- Program packages

  IDPAC, SYNPAC, MODPAC, SIMNON, ...

- Interactive, command driven
- Written in FORTRAN, ~1970
- Includes graphics

# IDPAC commands

1. UTILITIES
   - CONV
   - DELET
   - EDIT
   - FHEAD
   - FORMAT
   - FTEST
   - LIST
   - MOVE
   - TURN
2. GRAPHIC OUTPUT
   - BODE
   - HCOPY
   - PLMAG
   - PLOT
3. FREQUENCY RESPONSE OPERATIONS
   - ASPEC
   - CSPEC
   - DFT
   - FROP
   - IDFT
4. TIME SERIES OPERATIONS
   - ACOF
   - CCOF
   - CONC
   - CUT
   - INSI
   - PICK
   - SCLOP
   - STAT
   - TREND
   - VCOP
5. SIMULATION AND MODEL ANALYSIS
   - DETER
   - DSIM
   - FILT
   - RANPA
   - RESID
   - SPTRF
6. IDENTIFICATION
   - LS
   - ML
   - SQR
   - STRUC

- IDPAC knows 40 commands
- Hard to use, (= arguments?)
- Hard to remember
- E.g.<br>CONV *outfile* < *infile* (1 3) 4 1

- The result strongly depends on the user's knowledge

- Remedies: the help function, manuals

- Help - gives only name and syntax

- Manuals - too brief, hard to find (physically)

## Demands on a good help system

* How to do a task

◦ When to use a command

◦ How to combine commands

◦ Intelligent defaults

◦ A goal related help function,
help on methods,
a dynamic (=fancy) help function

- Question and answer dialog
- Command style dialog

- We want something in between
- Normal command dialog, sometimes the help system comes into action
- The commmand spy
- Pop up menues?

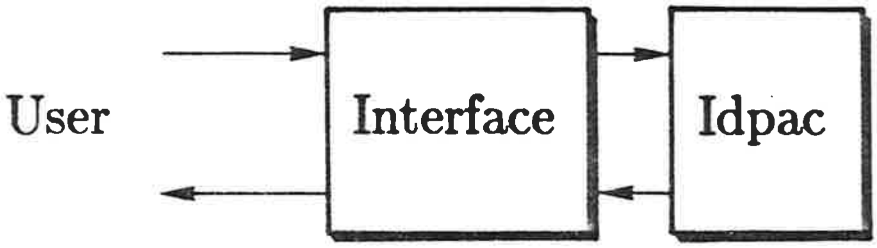
- Systems identification is a well defined problem
- Experts are available
- Idpac is well fitted for the problem

- The problem is not trivial
- The problem is not too large

- An expert system solution seems to be possible

# Building blocks

- Rule base
- Inference engine
- Script database
- Command spy
- File handler
- Interface to Idpac



User

Interface

Idpac

## Tools

- Lisp
- Prolog
- OPS5

## Data validation rules

```
(rule dataval1
    (if
        (visual inspection ok)
        (not (trends present))
        (not (outliers present))
        (histogram ok))
    (then
        (data ok)))


(rule dataval2
    (if
        (trends are present))
    (then
        (command: TREND NEWDATA < DATA ORDER)
        (forget (trends are present))
        (not (trends present))))


(rule dataval3
    (if
        (outliers are present)
        (dataset is long))
    (then
        (command: partion data)
        (forget (outliers are present))
        (not (outliers present))))


(rule dataval4
    (if
        (outliers are present)
        (not (dataset is long)))
    (then
        (command: interpolate)
        (data may be bad)
        (forget (outliers are present))
        (not (outliers present))))
```

# Sample run

```
(Is this true: (visual inspection ok) ?) yes
(Is this true: (trends are present) ?) yes

(Rule dataval2 deduces: (command:
                 TREND NEWDATA < DATA ORDER))
(Rule dataval2 deduces: (forget (trends are present)))
(Rule dataval2 deduces: (not (trends present)))

(Is this true: (outliers are present) ?) yes
(Is this true: (dataset is long) ?) no

(Rule dataval4 deduces: (command: interpolate))
(Rule dataval4 deduces: (data may be bad))
(Rule dataval4 deduces: (forget (outliers are present)))
(Rule dataval4 deduces: (not (outliers present)))

(Is this true: (histogram ok) ?) yes

(Rule dataval1 deduces: (data ok))
```

## Cross validation rules

```
(rule xval1
    (if
        (residuals available)
        (e1 and e2 uncorrelated)
        (e1 uncorrelated with u2)
        (e2 uncorrelated with u1)
        (parameters consistent with previous values))
    (then
        (form new sufficient statistic)
        (xval ok)))


(rule xval2
    (if
        (not (cross validation residuals are available)))
    (then
        (command: RESID e1 < PAR1 DATA2)
        (command: RESID e2 < PAR2 DATA1)
        (forget (not (cross validation residuals are available)))
        (residuals available)))
```

# Sample run

```
(Is this true: (cross validation residuals are available) ?) no
(Rule xval6 deduces: (command: RESID e1 < PAR1 DATA2))
(Rule xval6 deduces: (command: RESID e2 < PAR2 DATA1))
(Rule xval6 deduces: (forget (not (cross validation residuals
                                   are available))))
(Rule xval6 deduces: (residuals available))
(Is this true: (e1 and e2 uncorrelated) ?) yes
(Is this true: (e1 uncorrelated with u2) ?) yes
(Is this true: (e2 uncorrelated with u1) ?) yes
(Is this true: (parameters consistent with previous values) ?) yes
(Rule xval1 deduces: (form new sufficient statistic))
(Rule xval1 deduces: (xval ok))
```

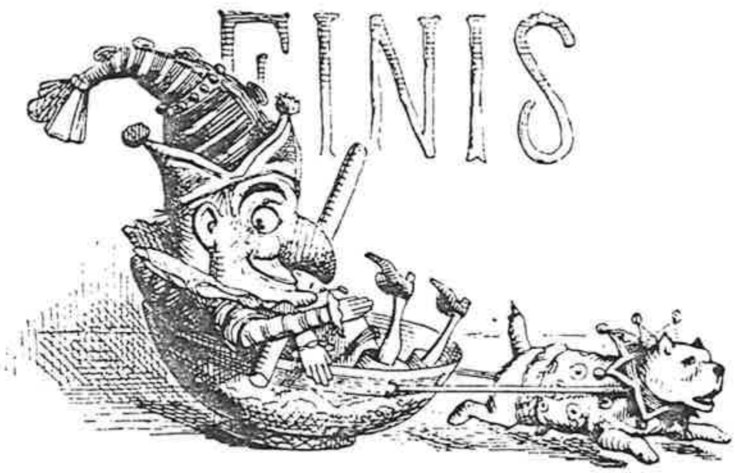
## Conclusions

- A small feasibilty studies
- Bookkeeping
- Saving computations
- Knowledge refinement
- We will proceed

## References

Larsson, ***An Expert System Interface for Idpac***, Masters thesis, CODEN: LUTFD2/(TFRT-5310)/1-77/(1984), Department of Automatic Control, Lund Institute of Technology, 1984.

Laub ***et al***, ***Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design***, Santa Barbara, California, 1985.



FINIS