

STATE ESTIMATION SIMULATION FOR THE DESIGN OF AN OPTIMAL ON-LINE SOLUTION

N.Aschöwer, E.Handschin Senior Member IEEE

H.Dobesch, W.Grein, A.Petroianu Member IEEE

University Dortmund

Brown, Boveri & Cie AG, Mannheim

FED. REP. OF GERMANY

Abstract - A state estimation simulation package (SES) developed for studies and educational purposes was further upgraded with on-line capabilities and implemented as a production grade program into a large-scale network control system. Although conceived for different applications and run-time environments, this transformation was realized smoothly and relative quickly without modifying the algorithmic kernel; the result is an efficient state estimation package (SE) with real-time capabilities.

The benefits of this development strategy are:

- feasibility studies such as optimal metering are possible with a unified SE approach in the early phase of a new control center project;
- the existence of a thoroughly tested SE simulation package accelerates the debugging phase of the on-line algorithm;
- the joint effort speeds up the know-how transfer and feed-back between R&D and reduces the overall costs.

A short description of the SES structure is given. The requirements and design goals of an on-line SE are outlined. Special emphasis is laid upon functional extensions such as: data base interfaces, the mapping of the network connectivity at equipment (terminal) level into a node/branch model and vice-versa, man-machine-interfaces, etc. Methods for software-quality enhancement, performance measurements and results obtained with the on-line SE on a VAX 11/780 are presented.

1 Introduction

The importance of state estimation methods for monitoring electric energy systems is illustrated by the many efforts to implement it for computer based on-line operation [1]. Indeed there is no alternative to state estimation for acquiring and securing data [2]. Yet comparing the large amount of theoretical papers with the actually implemented estimation schemes leads to the conclusion that its value has been recognized but not full use has been made of it. The reasons for this situation are plentiful. As state estimation is a new technique no long-term practical experience is available. Furthermore, systematic studies are difficult and time consuming both with respect to the design of a suitable telemeasuring system and to overall function specifications. Finally the compilation of the relevant network parameters is very difficult and time consuming.

Based on these observations a systematically designed state estimation simulator has been developed [3]. Its main functions will be described in this paper. The simulator has proven to be useful for off-line studies and to prepare the on-line realisation of state estimation.

Furthermore, it will be discussed in this paper how the off-line simulator has been used for the development of the on-line version of the state estimator. Many basic questions such as the selection of potential measurements, the guarantee of observability, the ability to detect and identify bad data etc., can be solved by means of the off-line simulator thus avoiding time-consuming and costly tuning problems during the commissioning phase. The functional differences between the off-line and on-line state estimation are discussed together with its integration into a large real-time monitoring software-package. Detailed performance measurements of the on-line version are reported.

2 Off-line State Estimation Simulator

The off-line state estimation simulator is an interactive computer program whose simplified structure is shown in Fig. 1.

The control program offers the operator the possibility to read from the data base all the necessary parameters describing the network under study. In order to specify an operating point for the power system, the simulator contains a fast decoupled load flow program (DLF) which generates all possible measurement values. In the measurement specification program (MS) the operator defines the measurement vector \underline{z} by selecting values from the results of the load flow computation. Furthermore, he may add measurement errors to the mathematical results of the load flow computation. The selection of the actual components of the measurement vector \underline{z} depends on the available or planned information channels between the respective measurement locations and control center.

Having specified a certain measurement vector \underline{z} , the state estimation⁺⁾ problem deals with the determination of the state vector \underline{x} whose components are given by the bus voltage magnitudes and angles. To this end the objective function

$$J(\underline{x}) = (\underline{z} - \underline{h}(\underline{x}))^T \underline{R}^{-1} (\underline{z} - \underline{h}(\underline{x})) \quad (1)$$

has to be minimized with respect to \underline{x} . In eq. (1) the network equations relating \underline{x} and \underline{z} are denoted by \underline{h} . The relative weighting of the measurements is given by the matrix \underline{R} .

Although some other concepts have been proposed in the past, the weighted least-squares approach [4] to solve eq. (1) has proven to be the most powerful method for applying state estimation to electric energy systems. Before the minimum of eq. (1) can be determined the observability test (OBS) [5] for the selected measurement vector \underline{z} must be performed.

⁺⁾ The term state estimation in the narrow sense describes the calculation of the state vector \underline{x} based on a specific set of measurements \underline{z} . In the broad sense, state estimation comprises all related functions such as network configuration, topology, bad data identification, optimization of the measurement locations and the interfaces to the on-line environment.

This paper was sponsored by the IEEE Power Engineering Society for presentation at the IEEE Power Industry Computer Application Conference, San Francisco, California, May 6-10, 1985. Manuscript was published in the 1985 PICA Conference Record.

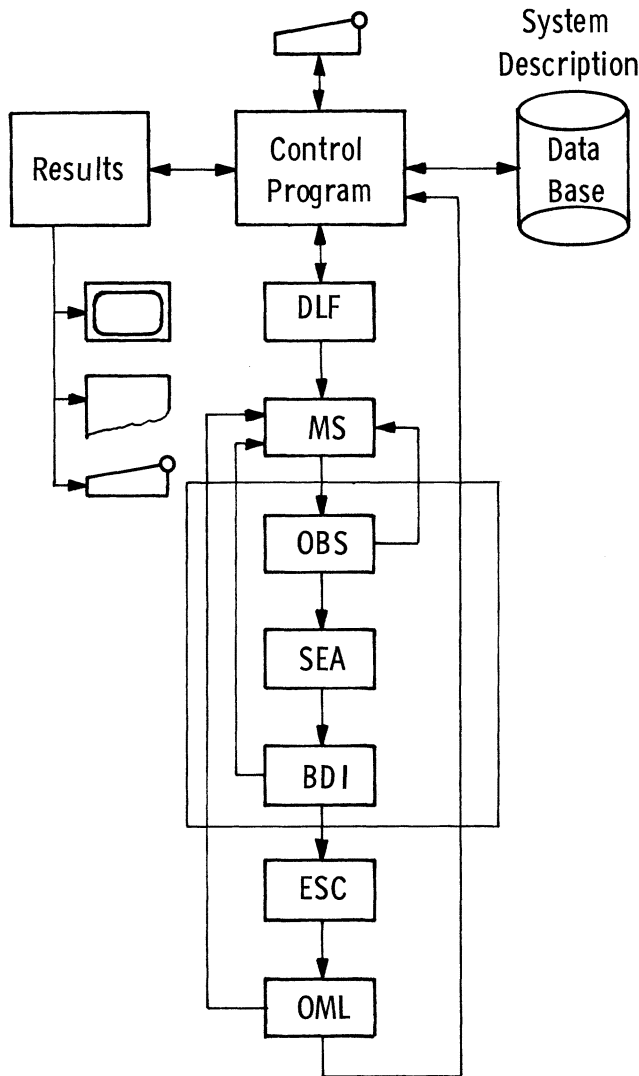


Fig. 1: Structure of the off-line state estimation Simulator

DLF: decoupled load flow
 MS : measurement vector selection
 OBS: observability test
 SEA: state estimation algorithm
 BDI: bad data detection and identification
 ESC: error sensitivity calculation
 OML: optimal meter location

If the optimality condition $dJ/dx = 0$ can be solved in an iterative manner the system is observable. The state estimation algorithm denoted by (SEA) is either a full or decoupled algorithm [6]. The latter has the advantage of using less computing time without loss of accuracy. The fast decoupled state estimator has been applied to networks with overhead lines and with cables.

Since one important function of state estimation is the bad data detection and identification feature, it has to be included in the simulator as well. The $J(x)$ -test and the ordered residual search [7] are part of the simulator and denoted by (BDI) in Fig. 1. In the case of bad data being present, the simulator indicates their location and offers pseudomeasurements to the operator. If this value should be used the operator may replace the actual bad data z_i and perform a further state estimation run. If the bad data is removed from the measurement vector z an observability test must be performed before the state estimation algorithm may be repeated.

Since bad data detection is a fundamental function of state estimation, the question of optimal meter location must be carefully studied. In this respect the off-line simulator has proven to be very valuable because it allows the systematic study of many possible metering configurations. In the context of this paper the term "optimal" refers to a measurement vector z where bad data in any component z_i may be detected and identified with a certain minimal probability. As shown in [7] the error sensitivity matrix

$$W = I - H (H^T R^{-1} H)^{-1} H^T R^{-1} \quad (2)$$

is given by the identity matrix I , the Jacobian matrix $H = dh/dx$ of the network equations h and the weighting matrix R . Its diagonal elements W_{ii} satisfy the condition [8]

$$0 \leq W_{ii} \leq 1 \quad i = 1, 2, \dots, m \quad (3)$$

m denotes the number of measurements z_i . The probability to detect bad data in the i -th component of the measurement vector z depends on the size of W_{ii} ; e.g. if W_{ii} is close to zero, the probability to detect a bad data in z_i is almost zero. A W_{ii} -value close to one implies a very high detection probability for bad data. An optimal meter location is characterized by the fact that bad data in important measurements z_i will be detected with a high probability. The state estimation simulator offers the possibility to calculate the W_{ii} elements for any specified measurement vector z . The error sensitivity calculation (ESC) in Fig. 1 has to be performed after each modification of the measurement vector z . The type, the location and the relative weighting of a measurement z_i determine the size of the corresponding W_{ii} element; of course the surrounding measurements also affect the size of W_{ii} .

In order to determine an optimal meter location (OML) with the simulator of Fig. 1, the error sensitivity coefficients W_{ii} are calculated, stored and compared for different meter configurations.

Fig. 2 shows the ordered sequence of the W_{ii} coefficients for different meter configurations. The overall redundancy defined as $\eta = m/n$ is used as parameter. Here, n denotes the number of unknown system state variables. As shown in [8] the optimal redundancy is about two. Fig. 2 clearly shows that for a given power system the choice of the measurement vector z greatly influences the bad data detection capability. Curve 1 shows the results for $\eta \approx 2$ before, and curve 2 after, the optimization of the meter location.

As the selection of the measurements z to be processed by state estimation depends on system engineering aspects, the optimal meter design is the result of an interactive use of the simulator. In order to correlate different measurement types with their W_{ii} values, Fig. 3 shows a typical output list of an optimization run. By classifying the inequality constraints (3) into ten classes, it is possible to include and delete measurements until the first two W_{ii} classes are empty.

In Fig. 3 the real power injections P_2 and P_5 are the two measurements with the lowest probability for bad data detection since they both belong to class 3 with $0.2 \leq W_{ii} \leq 0.3$.

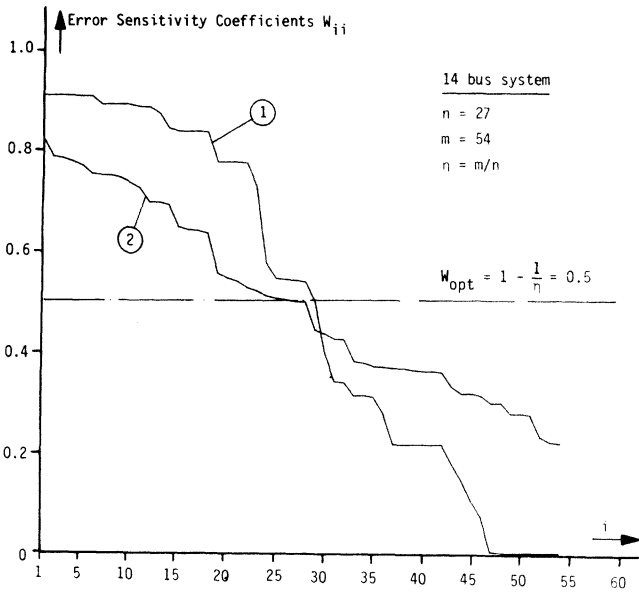


Fig.2 : Error sensitivity coefficients before (curve 1) and after (curve 2) optimization

The off-line estimation simulator has been successfully used in the past for the following types of applications:

- feasibility studies for the on-line implementation of state estimation
- specification of the necessary data in a systematic manner
- numerical comparisons of the results between the off-line simulator and the on-line version
- parameter tuning for the optimization of the on-line version
- familiarizing potential users of state estimation with this new data processing tool.

The well defined data base interface allows the straightforward use of the simulator with different system data. Better insight into the use of state estimation may close the gap between theory and application. The careful system analysis creates a good understanding for the problems connected with the on-line implementation of state estimation. The off-line state estimation simulator is implemented on a VAX 11/780 and can readily be transferred to any other similar computing facility.

Before discussing the work leading from the off-line simulator to the on-line version, the next two sections describe the system requirements and the on-line system interfaces.

3 Requirements and design goals for on-line implementation

The overall objectives are:

- to implement a state estimator into a production-grade, large-scale, real-time network monitoring and security analysis software package [9]
- to meet standard software quality requirements, i.e. a well tested, robust, fast, versatile and well documented state estimator
- to keep a very tight project dead-line (6 months)
- to reduce the man-power effort and the development costs
- to ensure the know-how transfer necessary for modifications, upgrading, further development and maintenance.

meas	Type of measurement (Location)	W-Value
W-Values within class 1 [0.0-0.1]		
W-Values within class 2 [0.1-0.2]		
W-Values within class 3 [0.2-0.3]		
2 5	P(2,2) P(5,5)	0.27 0.25
W-Values within class 4 [0.3-0.4]		
7 3 4	Q(2,2) P(3,3) P(4,4)	0.30 0.35 0.34
W-Values within class 5 [0.4-0.5]		
8 9 10	Q(3,3) Q(4,4) Q(5,5)	0.43 0.42 0.48
W-Values within class 6 [0.5-0.6]		
1 20	P(1,1) V(5,5)	0.56 0.51
W-Values within class 7 [0.6-0.7]		
6 17 18 19	Q(1,1) V(1,1) V(3,3) V(4,4)	0.62 0.60 0.67 0.67
W-Values within class 8 [0.7-0.8]		
11 14 12 15 13 16	T(1,2) U(1,2) T(2,1) U(2,1) T(3,4) U(3,4)	0.76 0.79 0.79 0.79 0.71 0.72
W-Values within class 9 [0.8-0.9]		
W-Values within class 10 [0.9-1.0]		

Fig. 3: Ordered measurement vector according to error sensitivity coefficients W_{ij}

Based on long and fruitful experience, it has been found that the optimal solution, in respect to the above mentioned goals and conditions, consists in a cooperation between industry (stressing the software engineering aspects) and university (stressing the research and mathematical aspects). In order to integrate the state estimator into a real-time, on-line software package, the following specific features are required:

- It has to be easily modified in order to meet the special requirements of different projects by people not involved in the development.
- It has to be embedded into a frame of interface routines to provide means of communication with other software systems.
- It has to work with a network description at a low level of abstraction. While a state estimation simulator utilizes the network topology and the measurement configuration modelled at a node/branch level, one of the salient features of an on-line version is its interaction with software processing the data transmitted from the electrical network at the level of individual equipment or terminals.

- It has to cope with possible deficiencies in the telemetering system. Observability may be achieved by pseudomeasurement generation. In case observability can only be guaranteed in a part of the network, the state estimator should be able to define this part automatically.
- It has to meet the very high running time requirements imposed on an on-line state estimator.
- It should perform automatic evaluations of the results, e.g. to inform the dispatcher if the network state approaches an insecure or dangerous state by checking limits.
- Compatibility with other security monitoring functions concerning the mathematical model of the network elements is mandatory for a unique description in the data base (DB).

4 The System Interfaces

The software environment of the on-line state estimator is shown in Fig. 4.

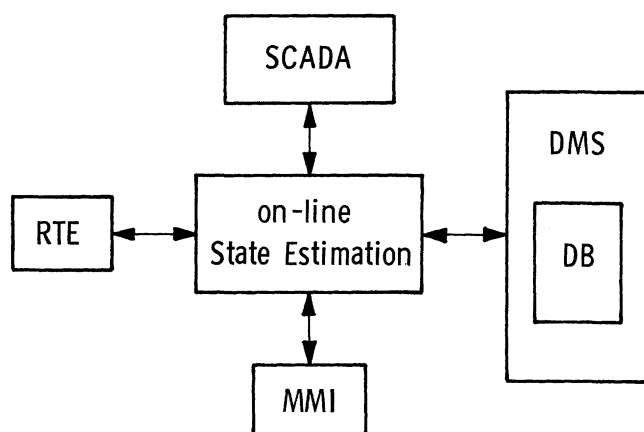


Fig. 4: Software environment of the on-line state estimator

RTE: Run Time Environment
 MMI: Man Machine Interface
 SCADA: Supervisory Control and Data Acquisition
 DMS: Data Management System
 DB: Data Base

The description of the software environment will be restricted to the features relevant to the on-line state estimator.

RTE: The state estimator process is created under the control of RTE with predefined privileges, resource quotas and task specifications. RTE supervises the execution and provides a system-wide and a process-specific, error-logging utility. Communication with other processes is done via mailboxes managed by RTE.

MMI: The dialogue services of MMI help the dispatcher to start the state estimator with certain parameters from a special function keyboard or using cursor control keys together with a special menu displayed on a CRT. MMI also receives the state estimation completion message and reacts by displaying the estimated values on the network diagrams actually selected by the dispatcher. Therefore, all state estimation results have to be stored in the shared data base (DB) in locations quickly accessible for MMI.

SCADA: The SCADA system receives telemetered measurements, relay and switch status information. It transforms the measurements to engineering units, performs local plausibility checks and stores the values in the DB. Since measurements are reported only by exception, it provides a good trigger for state estimation running in a permanent mode. The SCADA plausibility tests also serve as a prefilter for state estimation. To exploit this, the state estimator has to interpret also the measurement status flags generated by SCADA. The values used for limit check by SCADA and by state estimation are identical and lead to the same reactions on the CRTs.

DMS: The most important interface is the shared data base, the heart of the Data Management System. In order to achieve quick access to the large amount of data used by the state estimator, the relational structure of the DB has to be conceived such as to minimize the retrieval times. The layout of the DB has predominantly to conform to requirements from other subsystems with higher real time priorities, hence the DB structure is not optimally fitted to the requirements of the on-line state estimator. Unfortunately, there is no systematic way of realizing an overall optimum, since it is hard to foresee the mixture and the frequencies of certain access types. It should be stressed however, that a lot of effort has to go into this problem because otherwise the increasing number of page faults will be a major obstacle for good overall system performance. Another important piece of DMS is the off-line data preparation. Entering state estimation data requires a lot of consistency checks. All effort spent here means more time saved during the implementation phase of state estimation.

5 Functional Extensions

The purpose of this section is to explain the extensions necessary to implement an on-line state estimator (see Fig. 5)

5.1 Mailbox-Read (MBR)

The start parameters for state estimation are transmitted via mailbox. They contain information on the network area selected for estimation, convergence control, desired output and special limits to be supervised.

5.2 Data-Base Read Access (DBR)

The relational DB is the most important interface between state estimator and the other subsystems. The following types of network elements are stored in separate DB arrays: switches, busbars, loads, generators, lines, 2-winding and 3-winding transformers, shunts and the serial compensators. Static links point from these element arrays to the arrays containing the measurements and status information supplied by the SCADA subsystem.

The necessary input data of state estimation result from the individual requirements of the state estimator components such as TOP, DCO, OBS, SEA, BDI, DDI and LCH. They are

- status of the switching elements
- static network structure data of all network elements; i.e. information on how the network elements are connected with each other
- physical parameters of the lines, transformers, injections etc
- actual measurements and their standard deviations
- limits and parameters controlling the limit checks .

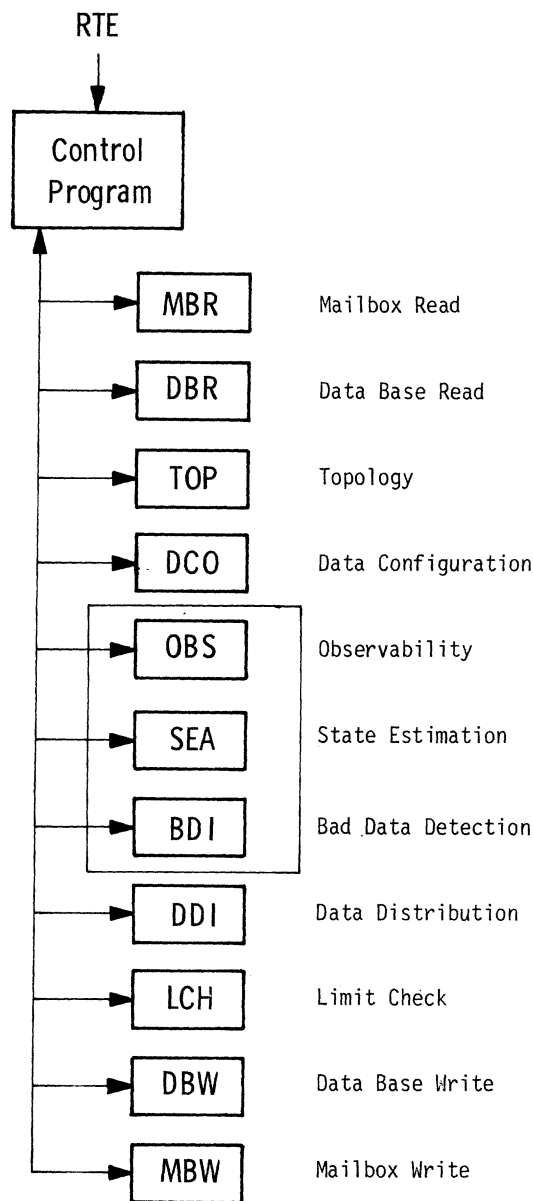


Fig.5 : On-line environment for state estimation

5.3 Switch Level Topology (TOP)

The network topology is described by static terminal numbers and dynamic breaker status information. The topology identification groups all network elements connected without impedance, into nodes. In a second step nodes which are linked by branches (lines, transformers or serial compensators) are assigned to a network group [10].

5.4 Data Configuration (DCO)

Since the state estimation algorithm is taken over from the state estimation simulator essentially unmodified, the input data have to be prepared to fit into its internal data structure based on a node/-branch model. To this end the function DCO

- adds up individual loads and generators to nodal injections
- determines coefficients for the redistribution of the estimated nodal values to the individual contributions

- assigns the measurements to the nodes or branches
- calculates limits for the voltage magnitudes at each node from the limits of the individual measurements
- builds up auxiliary pointers

and thus provides a physical model of the network made of nodes and branches.

5.5 Data Distribution (DDI)

The estimated nodal injections are distributed to the individual loads and generators according to the above mentioned coefficients. Through MMI the estimated values for individual measurements are made available to the operator in the same way as for SCADA results.

5.6 Limit Check (LCH)

This module performs three different types of checks: for individual measurements, for thermal limits of branches and for node voltage magnitude bounds.

The estimated results for each measurement are subject to tests against three upper and three lower limits, which can be blocked selectively by a mask. Violations are marked through the measurement status flags. This procedure is identical to the limit check of SCADA measurement processing.

Lines and transformers are inspected for violations of operational, warning and alarm limits. Overloads are marked by status flags. The individual limits can be modified by a common factor, being defined through start parameters, in order to tolerate temporary overloads.

The most conservative bound from the limits of the voltage measurements assigned to a node, is selected as the node voltage boundary. The existence of overloads is reported in the completion message.

The status flags are interpreted by MMI and the result of LCH are made visible to the operator through a color or blink mode change on the graphic display of the network.

5.7 Data-Base Write Access (DBW)

The state estimation results, i. e. the estimated values and status flags are written back to the DB into the array of measurements. The status flags of the network elements which are subject to Limit Check are also updated in the corresponding DB arrays.

5.7 Mailbox Write (MBW)

At the end of each state estimator run a message is sent to the mailbox of MMI informing the operator of the success of the run, whether bad data were found, and about the occurrence of limit violations. This message is also a command to MMI to display the new results on the monitors.

6 Performance

This part summarizes the state estimation software performance. The state estimation program was repeatedly executed; in order to increase confidence in the results, the average CPU execution time was calculated.

Fig. 6 represents the total execution time of the state estimator program plotted against the network size. Also shown are the times necessary to execute

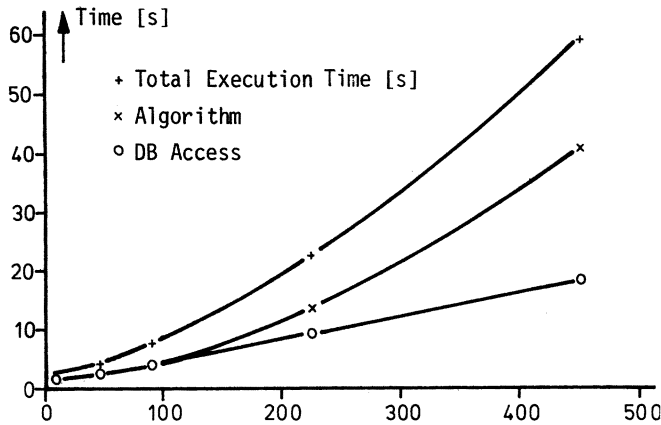


Fig.6 : Computing time requirements

all the modules from topology (TOP) up to limit check (LCH) according to Fig. 5.

The relative running time for the different state estimator modules as a percentage of the total execution time is shown in Fig. 7.

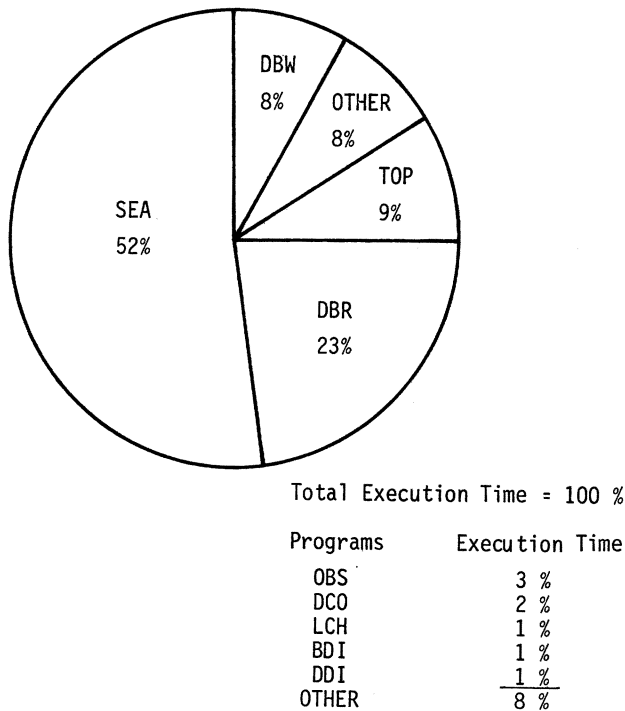


Fig. 7: Execution Time percentage for the on-line state estimator

Most of the running time is necessary for the SEA-module followed by the DBR. This long data base access time is a consequence of the scattered storage of the data related to the state estimation as already mentioned in section 5. By contrast the modules OBS, DCO, DDI, BDI and LCH require only a very short time (8%).

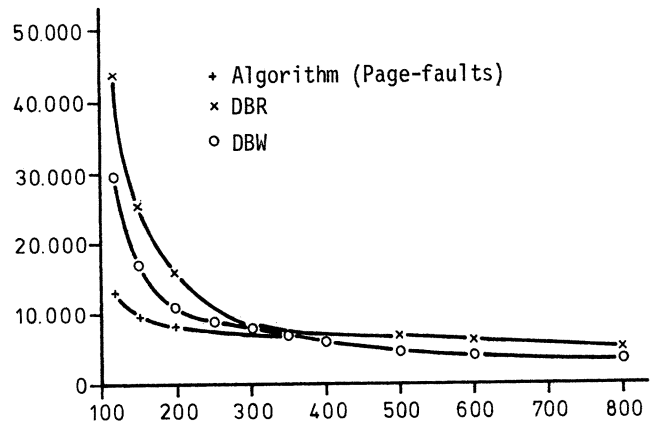


Fig. 8 : Page faults vs. working set for the on-line state estimator

Fig. 8 represents the page fault occurrence plotted against the working set. The working set represents the maximum amount of main memory that can be utilized by a program during its execution. The page fault activity is a measure of the effectiveness of memory management and indicates the amount of information transfer between external and main memory. Through measurements performed with the VAX operating systems, it was found that the optimal size of the working set, from a system point of view, lies at approximately 260 pages (about 130 kB).

7 Conclusions

The design and on-line implementation of the state estimator for monitoring electric power systems is an engineering challenge. The rapid development of theoretical methods, the high innovation rate of hardware technology and the ever increasing operational requirements make it difficult to complete a power system monitoring and control project on time.

The paper describes the advantages of close cooperation between industry and university. The more complex the system requirements become, the greater the need for sharing the corresponding work-load. While the university effort is primarily related to the exploration and evaluation of theoretical methods the industry must focus on software engineering aspects. As long as the information exchange goes in both directions, the technology transfer from university to industry and vice versa may lead to interesting and new ways of carrying out such complex technical projects. The practical software tool of the state estimation simulator contains a large amount of experience which can readily be used for the development of the on-line state estimator with its many interfaces to the overall on-line system.

With the off-line simulation package, design decisions can be backed by feasibility studies during all phases of the development. Questions like how to weight different types of measurements or what can be done in the presence of bad data for given measurement configurations or how stable convergence can be reached in mixed cable/overhead transmission line networks, are quickly answered using the off-line simulator. The structure of the off-line package serves as a guideline for the on-line state estimator since

there is a general similarity in structure, despite all necessary functional extensions. Such direct comparison for the central algorithmic modules is very helpful during of realization and testing.

From the experience summarized above, it can be estimated that the effort necessary to design, realize, test and integrate the state estimator package to the on-line system was reduced by about 50 percent as compared to doing everything "from scratch".

8 References

- [1] T.Dy Liacco: "The Impact of Control and Information Systems on the Security of the Power System", Keynote Address, Proc. of the CIGRE/IFAC Symposium, Florence,1983.
- [2] E.Handschin: "Acquiring and Securing Data", Survey Paper A, Proc. of the CIGRE/IFAC Symposium, Florence,1983.
- [3] N.Aschöwer: "Simulationsprogramm für State Estimation SIMEST", Bericht des Lehrstuhls für Elektrische Energieversorgung, EVP8316, 1983
- [4] F.C.Schweppe,E.Handschin: "Static State Estimation in Electric Power Systems", Proc. of the IEEE, vol.62, pp.972-982, Juli 1974.
- [5] C.Bongers, E.Ricke, E.Handschin: "Observability for Real-Time State Estimation", Study Committee Meeting of CIGRE, No.81SC08, Rio de Janeiro,1981.
- [6] A.Garcia,A.Monticelli,P.Abreu: "Fast Decoupled State Estimation and Bad Data Processing", IEEE Trans. on Power Apparatus and Systems, vol. PAS 98, pp.1645-1652, Sept./Oct. 1979.
- [7] E.Handschin,F.C.Schweppe,J.Kohlas,A.Fiechter: "Bad Data Analysis for Power System State Estimation", IEEE Trans. on Power Apparatus and Systems, vol. PAS 94, pp.329-337, March/April 1975.
- [8] C.Bongers: "Optimale Meßsysteme für die zuverlässige Überwachung elektrischer Energieübertragungsnetze", Dissertation, Universität Dortmund, 1979.
- [9] H.Dobesch,W.Grein,R.Küssel,U.Mattern,A.Petroianu: "Integration, Quality and Performance of Decision Support Software for Real-Time Network Monitoring and Security Analysis", submitted to PICA'85
- [10] W.Grein,H.Müller,A.Petroianu: "Dynamic Network Colouring", Proc. of the 8th PSCC, pp.489-496, Helsinki,1984.