within a band. When AR is outside the band, raise or lower signals are sent to units. Desired change of steam units is not fully realized for a period of time. This period is rarely less than 40 s and is normally about 60 to 90 s. Frequently therefore, AR can have had a sign reversal before the unit output power change, due to the control signal sent at the present LFC cycle, is being realized.

On such occasions, signals sent to reduce AR, can themselves amplify AR. One can alternately describe this by saying that response to some harmonics in AR are being realized after 90° to 270° phase delay and are therefore amplifying rather than attenuating these harmonics. This is to say that AGC is defeating the concept on which it is based for some portion of the time. Figure 1 shows a (linear) weighted average of system generation change per LFC cycle [WA(MWC)/Cycle] versus AR.

If the present concept of AGC were to be acceptable, total generation should rarely decrease when AR is negative, or rarely increase when AR is positive. This would require that almost all points marked in figure 1 should be in the second and fourth quadrants of the phase plane.

Another problem area is the current methodology for estimating ANR. ANR is a component of AR, and its erroneous estimate entails improper control action. It is obtained by summing the anticipated generation response of every unit (uANR). The latter is obtained by the use of a unit response model (URM) with fixed parameters. While the URM gives a reasonable estimate of uANR for a few units under some favorable conditions, it is often unreliable for most units.

These problems and other related observations are fully discussed in the paper.

Discussers: L. H. Fink, F. E. Duckett, D. N. Ewart, and A. M. Dicaprio

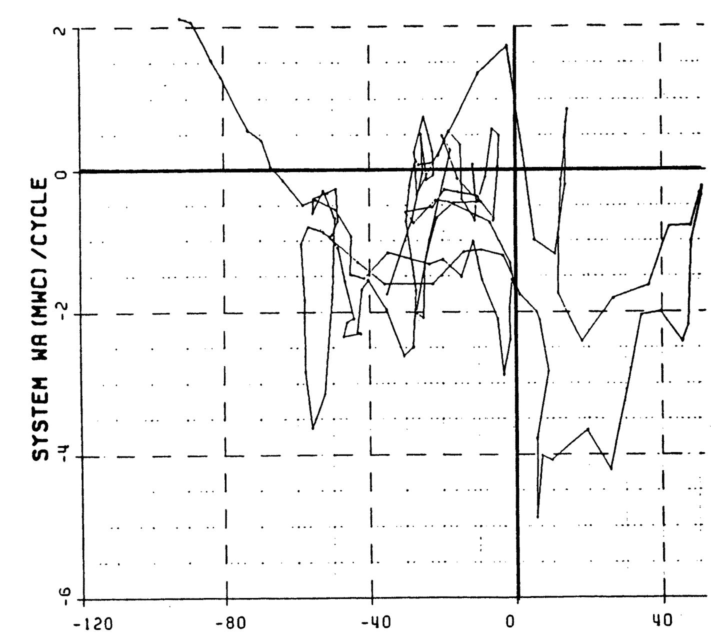

Fig. 1. Variation of system WA(MWC)/Cycle versus AR.

88 SM 722-1
May 1989

Software Design and Evaluation of a Microcomputer-Based Automated Load Forecasting System

Saifur Rahman, Senior Member and Mutasim Baba, Student Member
Energy Systems Research Laboratory
The Bradley Department of Electrical Engineering
Virginia Tech
Blacksburg, Virginia, U.S.A.

Summary

Several electric utilities (viz., Virginia Power) in the United States apply demand charge on the basis of coincidental peak hour load (PHL). This is in addition to the usual kWhr charge. The hour during which the utility experiences the highest demand (MW) for the month is called the peak hour and this MW value is referred to here as the PHL. The hourly demand meter readings for the appropriate consumer (e.g., industrial or large commercial customer, wholesale purchaser, etc.) are examined to determine their demand during the peak hour of the month. The monthly demand charge is computed on the basis of the consumer's contribution to this PHL. This monthly capacity charge can run into millions of dollars for industrial and large commercial customers. Thus it is very valuable for the concerned consumer to receive advance information about the time and size of the utility's peak load. This information would enable them to take load control actions such that their demand can be reduced during the hour of peak load. The motivation behind the research reported in this paper is precisely this—to be able to predict the time and size of this PHL.

The load forecast software is part of a load forecasting-load management decision support system based on microcomputers. An IBM-RT/PC is used as the central computer around which the load forecasting and load management simulator has been built in our laboratory. This 32-bit computer is multitasking and multiuser and runs under the IBM proprietary operating system called the Advanced Interactive Executive or AIX. This is based on the UNIX operating system developed by the AT&T Bell Laboratories. The IBM-RT/PC has a 2 MByte resident memory, a 70 MByte hard disk and a 1.2 MByte floppy diskette drive. This has six RS-232 ports, a parallel port and a built-in PC/AT coprocessor.

Special attention was given to the selection of the operating system environment and programming languages. The need for an environment that supports a fully-automated operation, and different levels of communication was very strong for designing the software package. The rationale behind a choice of the software is provided in the following.

The Operating System

Different features of UNIX in general, and AIX in particular, have made it the most suitable environment for our system. These features include:

- Support for multi-tasking multi-user systems;
- The ability to support more than one processes in the foreground, in addition to several processes in the background, on the console terminal;
- Ability to link commands directly between programming languages;
- Read/write compatibility with the DOS operating system;
- Versatile communication capabilities among various types of computers;
- Ability to support parent and child processes which is important for developing our decision support system; and
- Ability to use commands and functions as "shells" for other processes on the system.

Testing of any load forecast algorithm on historical load and weather data produces better results than that can be obtained under on-line conditions. In this study, we have tried to identify the factors that affect the accuracy of our rule-based algorithm, after it was tested on historical data. This was done by repeating the load forecast using weather variables as forecasted, and other data collected under on-line conditions. In addition to the natural differences in electric demand from one day to another, there are several factors that can affect the accuracy of the load forecast. The most important factors affecting the accuracy are:

- Bad data in the database;
- Inaccuracy in on-line load data measurements;
- Unplanned sale/purchase of power between utilities;
- Weather forecast errors; and
- Direct load control impacts.

The rule-based load forecast program takes 20 seconds of CPU time on the IBM-RT/PC for generating a 24-hour load forecast. The 24-hour load forecast is automatically checked for possible direct load control (DLC) effects. Then the forecasts with and without DLC are released for display. The load forecast software has been testéd for load and weather data (historical) for the Virginia Power service area. A summary of forecast errors for all days of all seasons of the year (1986) is presented in Table 1. Similar analysis has been performed for 1983 as well. The absolute annual average of errors for 1983 and 1986 are 1.437% and 1.298% respectively. These are obtained by averaging the seasonal averages of all hours of the day.

Discussers: S. S. Ahmed, N. D. Reppen, and R. Mukerji

TABLE 1
24-hour Load Forecast Error (%) Statistics for 1986

Hour	Winter		Spring		Summer		Fall	
	Aver.	St.Dev.	Aver.	St.Dev.	Aver.	St.Dev.	Aver.	St.Dev.
0	0.162	0.114	0.243	0.151	0.250	0.208	0.171	0.127
1	0.118	0.117	0.261	0.186	0.278	0.237	0.183	0.136
2	0.572	0.401	0.552	0.298	0.451	0.374	0.653	0.472
3	0.485	0.463	0.666	0.453	0.597	0.417	0.601	0.442
4	0.479	0.405	0.708	0.545	0.752	0.521	0.740	0.544
5	1.119	0.566	0.936	0.979	0.755	0.619	0.874	0.672
6	1.110	0.584	1.378	0.943	1.133	0.571	1.207	0.716
7	1.323	0.578	1.696	0.867	1.287	0.633	1.327	1.209
8	1.296	0.541	1.719	0.880	1.277	0.568	1.416	1.127
9	1.254	0.580	1.549	0.914	1.262	0.603	1.536	1.653
10	1.140	0.651	1.837	1.112	1.328	0.664	1.611	1.688
11	1.066	0.575	1.778	1.105	1.429	0.582	1.713	1.666
12	1.036	0.624	1.636	1.279	1.399	0.666	1.701	1.616
13	1.242	0.786	1.746	1.281	1.919	0.779	1.953	1.871
14	1.265	0.801	1.837	1.234	1.983	0.749	1.970	1.892
15	1.276	0.854	2.097	1.371	1.855	0.871	1.909	1.820
16	1.372	0.780	1.970	1.332	1.965	1.041	1.951	1.897
17	1.314	0.782	1.987	1.289	1.914	1.042	1.877	1.826
18	1.217	0.738	1.837	1.243	1.778	1.028	1.604	1.160
19	1.108	0.710	1.513	1.118	1.754	0.928	1.545	1.184
20	1.130	0.736	1.573	1.025	1.693	0.971	1.801	1.204
21	1.152	0.822	1.664	0.972	1.732	0.928	1.837	1.331
22	1.116	0.865	1.732	0.987	1.774	0.860	1.647	1.360
23	1.378	1.073	1.655	0.927	1.467	0.792	1.502	1.143
Aver.	1.030	n.a.	1.440	n.a.	1.334	n.a.	1.389	n.a.

88 SM 699-1
May 1989

An Artificial Intelligence Framework for On-Line Transient Stability Assessment of Power Systems

L. Wehenkel, Th. Van Cutsem and M. Ribbens-Pavella
Dept. of Electrical Engineering
University of Liège, Inst. Montefiore—B28
B 4000—Liège, Belgium

Keywords: On-line transient stability assessment, artificial intelligence, transient stability analysis and preventive control, inductive inference.

Summary

Transient stability assessment (TSA) of a power system pursues a twofold objective: first to appraise the system's capability to withstand major contingencies, and second to suggest remedial actions, i.e. means to enhance this capability, whenever needed. The first objective is the concern of analysis, the second is a matter of control.

For the time being, the on-line TSA is still a totally open question. Indeed, none of the existing two broad classes of methods (the time domain and the direct methods) are able to meet the on-line requirements of the analysis aspects, nor are they in the least appropriate to tackle control aspects.

The methodology we are introducing aims at solving the above stated on-line problem by making use of decision rules, preconstructed off-line. To this end, an inductive inference method is developed, able to provide decision rules in the form of binary trees expressing relationships between static, pre-fault operating conditions of a power system and its robustness to withstand assumed disturbances.

This paper concentrates on this latter problem, which is the most difficult task, and also the kernel of the overall methodology.

The proposed inductive inference (II) method pertains to a particular family of Machine Learning from examples. It derives from ID3 by Quinlan [1], tailored to our problem, where the examples are provided by numeric (load flow and stability) programs [2, 3].

According to the method, a decision tree (DT) is built on the basis of a preanalyzed learning set (LS), composed of states or operating points (OPs). Each state characterizes the steady-state (pre-fault condition) of a power system; it is analyzed with respect to (w.r.t.) a given disturbance. The ultimate goal is to discover the static attributes upon which depends essentially the stability behaviour of the states composing the LS; by static attribute we mean some simple algebraic combination of components of the state vector.

A DT is a hierarchical organization of this LS into a collection of subsets. The most general subset is the LS itself and corresponds to the top node of the DT; the most refined subsets correspond to terminal nodes. To each level of the DT corresponds a partition of the LS. The lower the level, the more refined the corresponding partition. Hence, generating successors of a given node amounts to reducing the uncertainty (or entropy) about the degree of stability of the corresponding states, or equivalently, to increasing the information about it.

The automatic building of a DT proceeds in a top down fashion beginning with the design of the top node and ending up with the terminal nodes. On the basis of an information theory measure, appropriate dichotomic tests on the attributes of the OPs are associated with the intermediate nodes. The terminal nodes, on the other hand will correspond to the different stability classes or degrees.

Conversely, the use of a preconstructed DT to classify an OP may conform to the following pattern: starting at the top node, apply the test corresponding to it; progress down the tree by applying at the successor nodes successively met, the appropriate test and by directing the OP according to the outcome of the test; stop when a terminal node is reached: