An implementation of on-line transient stability screening and control using distributed processing

Alexandre N'guessan, Mania Pavella, Louis Wehenkel

University of Liege
Department of Electrical Engineering and Computer Science,
Sart-Tilman B28, B-4000 Liege, Belgium
Email : Louis.Wehenkel@ulg.ac.be, nguessan@montefiore.ulg.ac.be

***Abstract*—This paper describes the implementation of an on-line transient stability assessment software, composed of algorithms for contingency screening and for the design of preventive control actions. The implementation of the two parts rely on a hybrid method called SIME, coupled with a time domain simulation engine and power flow program. The speed up of the contingency screening module is obtained by distributing contingencies on a cluster of computers to comply with extended real-time speed requirements. A compensation scheme is used to determine active power rescheduling alternatives in order to stabilize the dangerous contingencies identified at the screening step. The software has been coupled with an industrial EMS platform, and tested in the simulation environment.**

Keywords: transient stability assessment and control, distributed processing, energy management system, SIME.

I. Introduction

To take care of security in a preventive way, the operator needs to assess system behavior with respect to a certain number of plausible contingencies, both from a static and a dynamic point of view. While there exists state-of-the-art software for these tasks, in the past only static security assessment has been implemented for on-line operation, dynamic security (transient stability and voltage stability) being handled through operating rules precompiled during off-line studies. If this option could be justified by the simulation time necessary for analyzing dynamic security, things have changed nowadays with the improvements in computer speeds and with the emergence of new algorithms taking benefit of these improvements. On the other hand, economic and environmental pressures tend to operate systems closer to their limits and the risk of dynamic instabilities tends therefore to increase. So, in the last ten years, many attempts have been made in order to develop on-line dynamic security assessment and in particular on-line transient stability assessment (TSA). This paper presents such a software coupled with a commercial energy management system (EMS) combined with a dispatcher training simulator (DTS).

The paper is organized as follows : Section II describes the overall structure and implementation of the software; Section III gives some details concerning the TSA contingency filtering, assessment and preventive control modules; Section IV provides examples and tests results obtained using the software in different conditions. The last section provides conclusions and suggestions for further improvements.

II. OVERALL SOFTWARE IMPLEMENTATION

On-line TSA functions could either be integrated into the EMS platform (as it is the case for most static security assessment tools used in control centers) or implemented externally, i.e. in a separate, possibly ad hoc, computing environment loosely coupled with the EMS system. The latter approach, adopted in this paper has several advantages. In particular, the TSA function implementation and performances are essentially independent of the internal organization of the EMS platform, which improves maintainability, portability, as well as scalability of the TSA function itself. It also protects the EMS platform with respect to possible malfunctions of the TSA system. In the following, we start by first describing the overall principle of our TSA function and of its implementation, and then proceed by describing how this is coupled with an existing EMS platform.

A. Overall TSA function principle

The TSA function is composed of two main modules :

- Contingency screening module (CS)
- Preventive control module (PC)

These modules are controlled by a supervisory master process (MP) according to the following outline :

- MP receives real-time or study mode data from the EMS; it is triggered upon operator request, or in a cyclic way.
- MP first interrupts ongoing computations by CS and/or PC, if there are any such computations under way.
- MP then calls the CS module with the list of contingencies to screen; the CS identifies among these latter a subset of dangerous and potentially dangerous contingencies.
- MP displays the list of dangerous contingencies and passes it to the PC module; the PC module identifies preventive control actions to reschedule active power among generators, so as to stabilize these contingencies.

- MP displays the rescheduling actions, applies them to the current operating point, and passes the modified base case back to CS in order to cross-check the potentially dangerous contingencies.
- If, after that step, some of the previously non dangerous contingencies have become dangerous (subsequently to the rescheduling), MP calls again PC and so on, until convergence or interruption to handle a new TSA request.
- The decision to actually apply the suggested active power rescheduling at any time of this process is left to the operator.

Both CS and PC modules use a hybrid TSA method based on a single-machine equivalent derived from time-domain simulations (SIME). Depending on processor speed, power system size and level of detail of dynamic modeling, the CPU time required by either of these processes varies between 1 and 20 seconds per contingency. The CS handles typically a very large number of contingencies (several hundreds in practical cases) while the PC module generally receives at most a few tens of dangerous contingencies. Thus, in most practical situations the constraining part of this TSA function is the CS module. The CS module has therefore been implemented in our software using parallel distributed processing while, at the current stage, the PC module runs on a single processor.

B. Distributed CS processing

Distributing the tasks on a certain number of computers is necessary to comply with the extended real-time requirements for TSA where the objective is to reach a response time smaller than 15 minutes (e.g. for one complete CS followed by one PC cycle) in a realistic situation (system size: 200-300 generators, 2000 buses, 4000 lines; processor speed: 2GHz; 1000 contingencies). The distributed configuration of the CS process is shown in Figure 1.

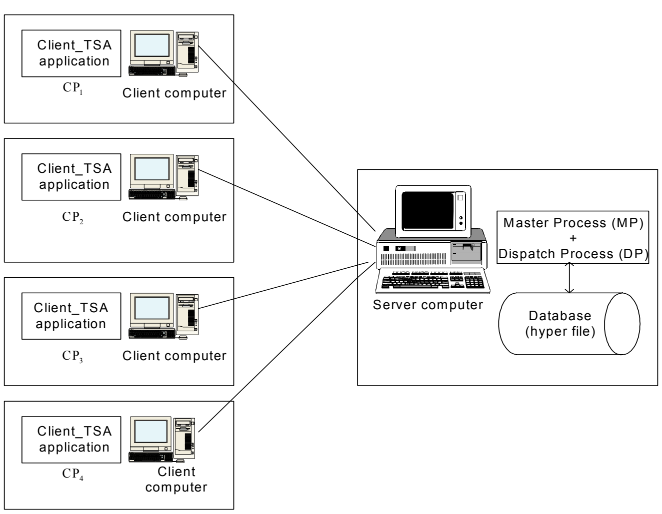

Figure 1. Description of the CS distribution process

One instance of the "Client process" (CP) is running on each client computer. The master program (called "Dispatch process" (DP)) executes common tasks, like scheduling of contingencies on CPs and management of common data.

The CS module works as follows:

- The DP first sends a message to each CP to start the screening mode (communications are based on TPC/IP sockets), together with the data files needed for the contingency screening (network data, dynamic models).
- Then each CP addresses to the DP a request to obtain a list of contingencies to process. Upon response from the DP, the CP starts treating the contingencies and, when finished, it sends its results back to DP for storage in the database. Notice that in order to reduce communication overheads, each client receives at each request a (configurable) number of contingencies to analyze.
- The DP maintains for each contingency a status flag which can take three values (ND = not dispatched, D = dispatched, P = processed). When a contingency is dispatched, the DP changes its status from ND to D. When results come back, the DP changes the status from D to P.
- If, at a given stage, a client sends back a request for new contingencies while all of them have already been dispatched, the DP nevertheless will send some (already dispatched) contingencies back to this CP. This way of doing allows tackling the fact that some CPs may be slower than others, or in the worst case may not respond at all, while balancing the load among all CPs and achieving minimum response time.

Notice that the DP is designed to allow the dynamic insertion or retrieval of CP processes, but the implementation is constrained to at most one CP process per client computer. The software has been implemented with Intel PC boxes operating under Windows (NT4 or 2000), but could in principle also operate with a heterogeneous set of clients. The optimal number of clients is obviously dependent on the size of the power system model and the length of the contingency list under consideration. Section IV will provide illustrative examples using a medium size system and variable numbers of processors demonstrating the scalability of the approach.

C. Contingencies building

Due to the fact that it is difficult to simulate all possible contingencies that may exist, a judicious choice should be made. In this implementation, the choice of the contingencies can be made manually or automatically. If the automatic mode is chosen, the software can, for example, make a short circuit at each end of a line and clear the short-circuit by opening the line in the post-fault. The manual choice is available to let the operator add some specific contingencies that have not been taken into account by the automatic mode.

D. Connection to the EMS platform

Data coming from the EMS platform are the state estimator data, or, in study mode, the power flow data. These data have been preferred to the SCADA data because they are supposed to be more accurate. In our experiments the software has been coupled with the DTS (dispatcher training simulator) of the ALSTOM/ESCA platform. This latter is a replica of the EMS environment together with a program simulating the slow drift of the power system state and telecommunications. This platform provides a certain number of possibilities of connection from the outside world. One of these is composed of the HABDDE-SAMPLER running on the EMS side and the HABDDE client utility running on the outside system, in our case the PC running the MP and DP processes. The TSA-MP communicates directly with HABDDE by the use of dynamic data exchange links (DDE) as described in Figure 2.

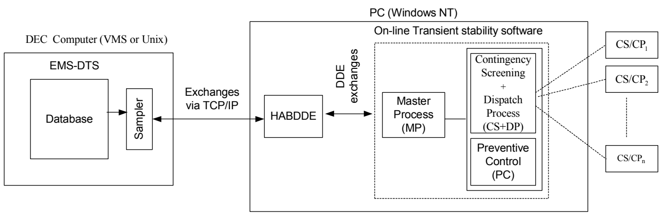

Figure 2. Coupling of the TSA master process and the EMS/DTS

The data received from the platform are stored in a relational database from which they can be dispatched to the different clients. Via the HABDDE and DDE links, the contents of this database are maintained synchronous with the EMS database on which it is connected, which (as already mentioned) could be either the real-time network database (updated from SCADA by the state-estimator) or the study mode network database, updated by the study mode tools (power flow, optimal power flow) of the EMS.

III. TSA FUNCTION

The TSA function is composed of the contingency screening and preventive control modules (CS and PC, respectively). Both rely extensively on the coupling of the SIME method [1,2,3,4] with a time-domain simulation program which was developed on purpose for this project.

A. Screening and filtering the contingencies (CS)

The CM module classifies the contingencies into three groups: the group of dangerous ones, the potentially dangerous ones, and the non-dangerous ones. The module implements the FILTRA method [4], which computes stability margins and identifies critical and non-critical machines.

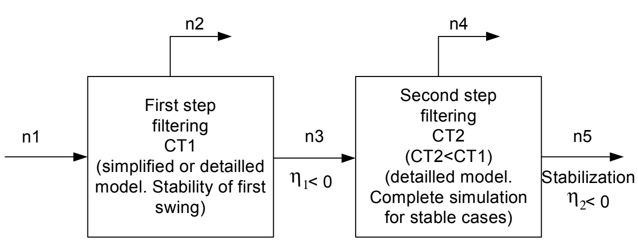

Figure 3. Contingencies filtering

The filtering consists of two pipelined steps (Figure 3):

1. A fast simulation is carried out with a relatively large clearing time CT1 (for example 200 ms). The simulation is stopped by SIME as soon as either the return angle is reached (stable case) or as the instability is detected. At this step, the possibility of using a simplified model for representing the machines exists.
2. The second step re-simulates in detail the cases declared unstable at step 1, using the actual clearing time of protection systems (denoted CT2 in Figure 3). Here, a detailed model is used for the representation of the machines and, in the first swing stable cases, the simulation runs till the end of the observation time (say 10s) to check for multi-swing instabilities.

The contingencies declared stable at step 1 are classified as "non-dangerous", those declared unstable at step 2 are called "dangerous", the others are classified as "potentially dangerous". The first class is discarded after the CS process has finished. The "dangerous" contingencies are given as input to preventive control module.

B. Stabilization of the unstable contingencies (PC)

This is done by using a compensation scheme [2], which is a by-product of the SIME method. The purpose of the compensation scheme is to assess approximately the amount of power adjustment necessary to stabilize the power system. Its derivation relies on the computation of stability margins and identification of the critical machines driving the system's instability. Part of the power produced by the critical machines is shifted to non-critical machines in order to increase the stability margin. The stabilization process is iterative and handles at each iteration simultaneously all the dangerous contingencies, in the following way :

1. Starting with the results obtained from the CS module, and for each dangerous contingency, the PC module determines which machines' active powers must be reduced and by how much, using the compensation scheme.
2. The resulting changes obtained for each dangerous contingency are compared to determine the total change needed to stabilize all contingencies simultaneously.
3. A new operating point is created by re-dispatching this amount of power on non critical machines, in order to preserve the load-generation balance. This can be done by taking into account incremental cost curves of the different units used for this operation, so as to minimize the overall re-dispatch cost.
4. The dangerous and non-dangerous contingencies are re-assessed using step 2 of the CS module.

If all contingencies are found stable at step 4, the process stops, otherwise, the procedure starts again at step 1 with the new operating point and the results of step 4 as input.

C. Time-domain simulation engine and power flow solver

The simulation engine used by the CS and PC modules is a step-by-step software implemented by the first author. This software uses the implicit trapezoidal method [5,6] to solve the differential equations. The implementation contains standard dynamic models useful for simulations during a period of 5 to 10 seconds. Five different models of machines have been implemented and the software also contains the representation of a certain number of control devices: fast-valving, excitation systems, speed governors, turbines and PSS. At the network level, static loads, dynamic loads and SVCs have been implemented. The different models of devices represented are the common models recommended by IEEE task forces [8,9,10,11]. A home made Newton-Raphson power flow solver is used during the iterations of the PC module.

IV. TESTS AND CASE STUDIES

This section provides a certain number of simulation results aiming at assessing the TSA function from an implementation point of view and at illustrating the various features of the software. We start with some examples to show the functionalities of the software, using the rather small "ESCA test system" installed on the EMS-DTS platform running in the control center laboratory of the University of Liège. Then we provide more realistic simulations using the IEEE 50-machine test system [13] in order to test the distributed processing feature of the software.

A. On-line simulations using the ESCA system

1) Scenario

The operating point used in the 60 bus ESCA system has 15 generators running. For the purpose of the study, these generators have been modeled in detail, with excitation systems, speed governors and turbine models. These dynamic models are stored and maintained in the database of the TSA function and not in the EMS. A fixed step size of 10 ms has been selected for the integration process in the step-by-step program, and a maximum simulation period of 5 seconds has been settled for each contingency simulation. Since the system is rather small (74 buses, 85 branches, 15 synchronous machines), a single CPU configuration was used to run the on-line TSA (300 Mhz Celeron, 64 MB of Ram, Windows NT4). The EMS-DTS platform, on the other hand, runs on a DEC-Alpha workstation operating Digital-Unix.

2) Contingency selection

An automatic selection has been done implementing a short-circuit at each node and opening a line in the post-fault situation. This gave us a total of 84 contingencies to screen.

3) Screening and filtering

This step allows us to discard the harmless contingencies. The first step uses a clearing time of 300 ms. The second step uses a clearing time of 200 ms. This clearing time would not be realistic in practice, but it has been chosen for the purpose of the study (indeed, the protection devices nowadays can respond in a time of two to five cycles). In this example, a detailed model is used for the two steps of the filtering. After running all the contingencies, from the 84 ones selected, 4 have been detected as dangerous.

4) Stabilization process

At this step, The PC module tries to stabilize the dangerous contingencies. By using the compensation scheme, it reduces the critical machines' active power to stabilize simultaneously all the dangerous contingencies. Table 1 describes the stabilization process, which took here 4 iterations.

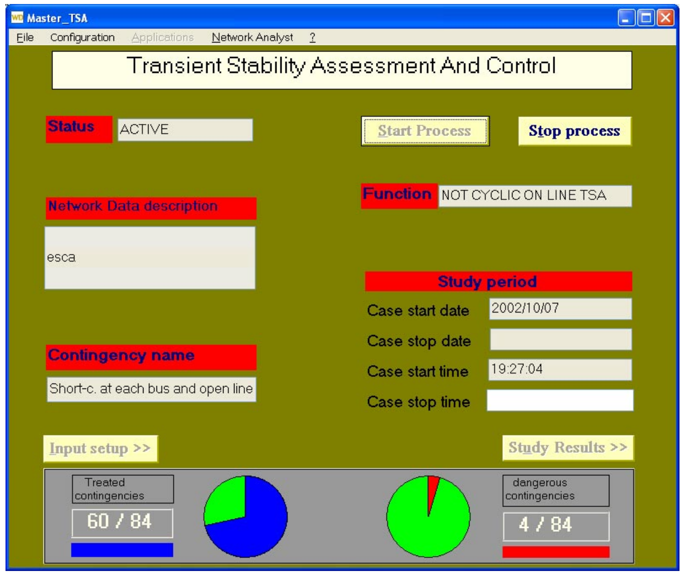

Figure 4. Description of the software interface during the contingencies filtering

TABLE 1.
DESCRIPTION OF THE STABILIZATION PROCESS

Conting. Name	Margin	Simul. time (sTDI)	DeltaP	Critical machines	Gap
First iteration					
Conting46	-69.92	0.34	-247	G14	57
Conting55	-47.22	0.34	-247	G14	60
Conting57	-11.83	0.41	-149	G15	69
Conting58	-25.63	0.31	-149	G15	50
Second iteration					
Conting46	-16.55	0.56	-112	G14	89
Conting55	-13.94	0.61	-112	G14	84
Conting57	0.00	5.00	75	G15	-
Conting58	-13.54	0.56	-68	G15	89
Third iteration					
Conting46	-1.66	1.09	-23	G14	106
Conting55	0.00	5.00	0.00	G14	-
Conting58	-1.16	0.99	-16	G15	99
Forth iteration					
Conting46	39.57	5.00	22	G14	-
Conting58	12.18	5.00	15	G15	-

The first column of Table 1 gives the names of the contingencies, the second one the stability margin (an indication of the degree of stability or instability), and the third column shows the simulation time. In the case of unstable contingencies (negative margin) this time corresponds to the time where SIME detects instability; in the case of stable simulations it corresponds to the maximum observation time fixed for the case (5 seconds in our simulations). Notice that the simulation time of the unstable cases is in average ten times smaller than that of the stable ones. The fourth column shows the power (in MW) to retrieve from the critical machines determined by the compensation scheme. The fifth column gives the critical machines and the last column gives the separation angle (in degrees) between the last critical machine and the first non-critical machine. This information is used to determine how to share the amount of rescheduled power among the various critical machines.

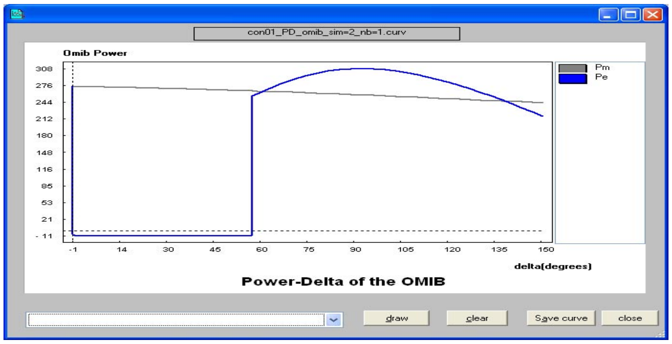

Figure 5. OMIB curves of contingency 46 at the second iteration

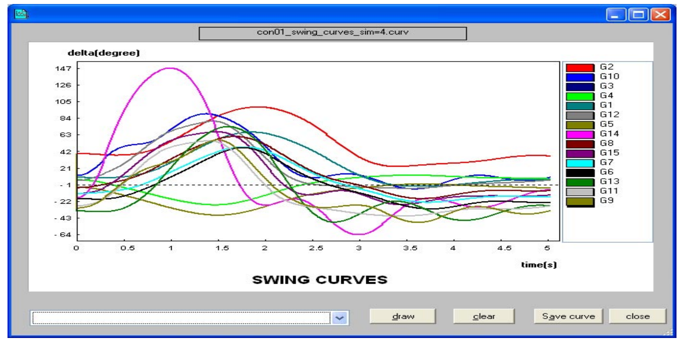

Figure 6. Swing curves of contingency 46 at the last iteration

Notice that the amount of power rescheduled for each machine is for each iteration the maximum value suggested by the different contingencies. These values are cumulated over the successive iterations. Notice also that in Table 1, the last iteration actually compensates for a slight over-stabilization resulting from the first three iterations.

5) Rescheduling of the power on non critical machines

The rescheduling on the non-critical machines is made taking into account the incremental costs of the different generators. After the stabilization, a check is needed to ensure that the stabilization process has not destabilized some contingencies which were stable at the beginning. At this level, many patterns may exist for this check. We can, for example, decide to check only the "potentially dangerous" contingencies, or we can decide to check the entire list of candidate contingencies. In the present test, we chose the latter option. This new screening of the contingencies has shown that after the rescheduling there were no dangerous contingencies anymore.

6) Graphical outputs of the software

The TSA software can display information from the CS and PC steps in graphical form. Figure 5 gives the Power-Angle curves (equal area criterion diagram) corresponding to the simulation of contingency 46 at the second iteration of the stabilization process. This graphic shows the relation between the relative angle of the center of inertia of the critical machines and the remaining non-critical machines and their equivalent electrical and mechanical powers [4]. One can see graphically that the decelerating area is much smaller than the accelerating area, reflecting the fact that the contingency is still unstable. The (negative) stability margin is actually the difference between the latter and the former areas, or equivalently the kinetic energy of the single-machine equivalent when the unstable angle is reached (intersection of electrical and mechanical power curves). Figure 6 shows the individual machines' swing curves after the last step of the stabilization, confirming that contingency 46 has indeed been stabilized.

The total simulation time for the on-line test using a single CPU is of about 3 minutes. An evaluation of the total integration time of the first loop of the filtering scheme gave 140.13 time domain integration seconds (sTDI), in other words about 1.5 sTDI per contingency. This can be compared with a total of about 420 sTDI that would be required by a pure time-domain method. Hence the SIME method reduces the amount of computation by a factor 3 in this case. Note that this has been confirmed by extensive tests [4].

B. Parallel distributed processing tests

1) Scenario

The network used here is the IEEE 50 generator system. This system is not implemented on the ALSTOM EMS-DTS platform and hence the tests are carried out in off-line mode, i.e. using as inputs data in the form ASCII text files.

This test is carried out to check the behavior of the distributed process in conditions closer to the possible use of the software in real life (larger system and larger number of contingencies). The integration step is still 10 ms and we use a simulation time of 5 seconds. The dynamic model used is similar to the one used in the previous test. The power system model itself is composed of 145 buses and 453 branches.

2) Contingency list

Choosing the automatic option of a short-circuit at each bus and opening a line in the post-fault period gives us a total of 892 contingencies (the lines which created a degree of connectivity greater than one have not been selected since the software does not, for the time being, take into account the case where several islands coexist).

3) Contingency screening in distributed mode

The distributed option has been selected for the filtering of the contingencies. Four computers have been selected for this task.

- PC1 : processor 266 Mhz (one CP)
- PC2 : processor 300 Mhz (the DP runs always on this computer, one CP can also run one PC2)
- PC3 : processor 750 Mhz (one CP)
- PC4 : processor 1.7 Ghz (one CP)

The computers communicate through 100Mbit/s Ethernet local area network.

Different scenarios, corresponding to different combinations of these processors, are run to assess the effectiveness of the distributed processing scheme. In each scenario, each CP receives at each request 10 contingencies to treat.

To start with, each computer is taken individually for the entire filtering. The results of theses simulations are given in Table 2.

TABLE 2.
SIMULATION TIME OF EACH COMPUTER TAKEN INDIVIDUALLY

	Processor	Simulation time (elapsed)
PC1	Pentium 266 Mhz	43 min 56 sec
PC2	Celeron 300 Mhz	31 min 19 sec
PC3	Pentium 750 Mhz	12 min 45 sec
PC4	Pentium 1.7 Ghz	5 min 20 sec

Then, beginning from the less powerful computer the distributed process is used by adding at each step a new computer. The results are given in Table 3.

TABLE 3.
SIMULATION TIME OBTAINED BY THE COMBINATION OF THE DIFFERENT COMPUTERS

Simulation pattern	Simulation time (elapsed)
PC1+PC2	20 min 41 sec
PC1+PC2+PC3	8 min 9 sec
PC1+PC2+PC3+PC4	3 min 50 sec

Figure 7 shows the display of the MP scheduler at the end of the execution, which is used to monitor the progress of the CS activities over the different processes. One can see that most of the work has been done by the 1.7 GHz client, as expected. Notice that the load balance is quite close to what could be predicted by a theoretical calculation derived from the processor speeds (only PC2 appears to be significantly slower than predicted, which is due to the fact the MP and DP also run on this CPU).

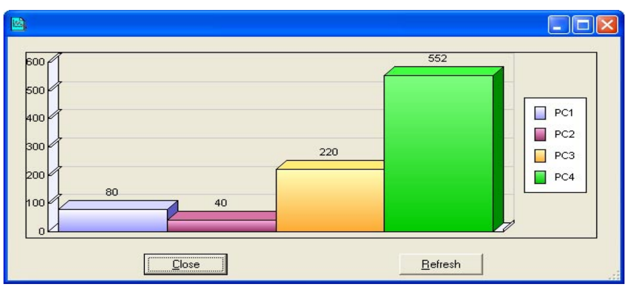

Figure 7. Computational load balancing diagram

A second test has been carried out to evaluate the software in a situation with more uniformly distributed processing power and a larger number of CPs. This consists of 8 Pentium IV processors with clock frequencies between 1 and 2 GHz. The overall response time is of about 1 minute to handle the full list of contingencies. The speed up with respect to the use of the single processor configuration is of a factor 8.

All in all, these tests show the robustness and scalability of the distributed contingency screening software.

V. CONCLUSIONS AND FURTHER DEVELOPMENTS

This paper has described an implementation of an on-line TSA function based on the combination of distributed contingency screening, a non distributed preventive control module, both based on the SIME method combined with a time-domain simulation software. SIME itself is responsible for speed up factor (with respect to pure time domain simulation) of 3, while the distributed processing scheme has been shown to provide speed-up factors of more than one order of magnitude at the expense of more CPUs. Most noticeable is the scalability of the load-balancing scheme used to distribute contingency screening among a variable number of possibly heterogeneous CPUs, and its robustness with respect to the failure of individual processors. The resulting software demonstrates the feasibility of on-line TSA for large power systems. In addition, it has been coupled loosely with an existing commercial EMS-DTS platform in such a way that it does not perturb the latter's real-time functions.

The software also allows providing guidelines to the operator on the way to stabilize his system in case there are dangerous contingencies.

In terms of software implementation, a further development concerns the combination of the generation rescheduling module with an optimal power flow software integrated in the EMS platform, and also the distribution of the preventive control module itself.

VI. REFERENCES

[1] Betiol Arlan Luiz "Maximum power transfer in transient stability-constrained power systems: application to a Brazilian power network" Ph.D. These 1999

[2] Ernst D. , A.L. Betiol, D. Ruiz-Vega, L. Wehenkel, M. Pavella "Compensation schemes for transient stability assessment and control" Proceedings of the LESCOPE 98, Halifax (Canada), June 7-9, 1998, pp. 225-230

[3] Zhang Y., L. Wehenkel, M. Pavella "SIME: A comprehensive approach to fast transient stability assessment" IEE of Japan, Vol. 118-B, no2, 1998

[4] Pavella M., Ernst D., Ruiz-Vega D. "Transient Stability of Power Systems : A unified Approach to Assessment and Control" Kluwer Academic Publishers, 2000

[5] Dommel H.W., Sato N. "Fast Transient Stability Solutions" IEEE transactions on Power systems

[6] Arrillaga J. and Arnold C. P. "Computer analysis of power systems" John Wiley & Sons 1990

[7] Meyer Bruno, Nativel Gilles "New trends and requirements for dynamic security assessment" Control Engineering Practice 7 (1999) 375-380

[8] IEEE Committee Report "Dynamic models for steam and hydro turbines in power system studies". IEEE Transactions on Power Apparatus and Systems, Vol. PAS-92, pp. 1904-1915, Nov./Dec. 1973

[9] IEEE Committee Report "Excitation system models for power system stability studies" IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, pp. 494-507, February 1981

[10] IEEE Committee Report "load representation for dynamic performance analysis" IEEE Transactions on Power Apparatus and Systems, Vol. PAS-8, pp. 472-482, May 1993

[11] IEEE Committee Report "Static var compensator for power flow and dynamic performance simulation" IEEE Transactions on Power Systems, Vol. PAS-9, pp. 229-240, February 1994

[12] ESCA COORPORATION "DTS – User Guide" Software version EMP 1.5. Release date : June 1995

[13] IEEE Committee Report "TRANSIENT STABILITY TEST SYSTEMS FOR DIRECT STABILITY METHODS" IEEE Transactions on Power Systems, Vol. PAS-7, pp. 37-44, February 1992.