Enhancing Wide-Area Monitoring and Control with Intelligent Alarm Handling

Chumki Basu[1], Kaushik Das[2], Jagabondhu Hazra[1],
Deva P. Seetharam[1]
[1]IBM Research – India, Bangalore, India
[2]Technical University of Denmark, Roskilde, Denmark
{chumbasu,jahazra1,dseetharam}@in.ibm.com
kdas@dtu.dk

***Abstract*— With advances in synchrophasor (PMU) technology, we have an opportunity to enhance power system monitoring and control schemes. In this paper, we discuss how this can be achieved in two ways: 1) combining PMU measurements with spatial and temporal knowledge of the grid to classify the severity of an alarm and 2) intelligently ordering alarms with associated recommendations to enable operators to take faster and better control decisions.**

***Index Terms*—Wide-area monitoring, analysis and simulation, real-time control, intelligent alarm processing, protection systems.**

I. Introduction

Phasor Measurement Units (PMU) provide low latency data at rates ranging from 30-120 samples per second. A consequence of such high throughput data is an increased rate of generated alarms. Operator alarm overload is a well studied problem, and it has been shown that a control center operator may not be able to handle alarm rates as low as 10 alarms per minute (alarm rates during the first few minutes of a major disturbance are on average closer to 80 alarms per minute) [1]. This scenario illustrates the challenge of taking necessary and effective control and protective actions in real-time.

Related work in intelligent alarm processing has suggested a number of approaches to tackle this problem [2]-[7]. We are especially interested in how the output of intelligent alarm processing can trigger rules that recommend (manual and) automatic control actions [3]. Specifically, our approach uses stream computing [8][9] as a platform to process PMU data and emit ordered sets of alarms. (An example of an alarm ordering is the temporal order in which alarms are generated as events unfold on the grid.) In addition to generating recommendation lists, a dual contribution of our work is a set of validation metrics that are used to evaluate the orderings according to user-specified criteria.

As a by-product of this process, we also generate high-level, spatial and temporal interpretations of the data for the operator. Hence, the contributions of our approach are both *descriptive* – analysis of the spatial and temporal properties of the alarms that are observed on the grid right now along with their associations and connectivity information – as well as *prescriptive* – a recommended ordering of control actions derived from alarm ordering.

II. Alarm Severity, Classification, and Ranking

A. Definitions

Severity indices have been used to classify events such as faults on the power system. Computation of these indices is an essential step in intelligent alarm handling [3]. We define severity indices (*SI*) incrementally, starting with indices that are *event-based*, i.e., based on the percentage of deviation from nominal value for each of the following events observed on bus_i at a given point in time: overvoltage (*SI_ov*), undervoltage (*SI_uv*), overfrequency (*SI_of*), underfrequency (*SI_uf*), overcurrent (*SI_oc*). These indices do not have any spatial or temporal (time series) component.

Then, we introduce temporal (time series) knowledge into the computation of three new severity indices: *percentage occurrence in time*, *critical deviation* and *area*. Percentage occurrence measures the fraction of samples, s, which are alarms of type τ, a_τ, in a fixed time window, W_{t-t0}.

$$O = \frac{\left|a_\tau \; in \; W_{t-t_0}\right|}{\left|s \; in \; W_{t-t_0}\right|} \tag{1}$$

Critical deviation measures the magnitude of deviation, d_t, for the current sample relative to the maximum deviation observed in the time window.

$$D_{crit} = \frac{d_t}{\max\{d \; in \; W_{t-t_0}\}} \tag{2}$$

Finally, area is a normalized value of the area under the curve violating the limit (calculated using the trapezoidal method), where N represents the number of equally spaced intervals within a time window.

$$A = \frac{\frac{t-t_0}{2N}\left(f(x_1)+2f(x_2)+\ldots+2f(x_N)+f(X_{N+1})\right)}{(\max\{d\ in\ W_{t\text{-}t_0}\})t-t_0} \quad (3)$$

We compute a combined temporal severity index for each bus_i by summing the individual indices.

$$SI_{Temporal_i} = SI\_ov + SI\_uv + SI\_of + SI\_uf + SI\_oc + O + D_{crit} + A \quad (4)$$

We compute a spatial severity index for each bus_i by summing a weighted average of the temporal indices across buses *h* hops away, where weight is the inverse of the number of hops between buses. We define the local zone of an alarm (where we expect grid state to be similar) as buses at most h_{max} hops from the alarm location and set $h_{max} = 5$ (a value that can be configured according to empirical observation).

$$SI_{Spatial_i} = \sum_{h=1}^{h_{max}} \frac{SI_{Temporal_i}}{h} \quad (5)$$

B. Ontology

For each alarm, we also associate a set of known control actions encoded in an ontology of alarms and control actions. Alarms and control actions are represented as separate hierarchies in the ontology. Taxonomic links in the hierarchies represent *Is-A* or *class-subclass* relations. Alarms can be classified along multiple dimensions – selecting a categorization depends on its anticipated utility for specific applications. For example, consider a very simple classification of alarms using three categories: "Red" (extreme importance – requires immediate attention), "Yellow" (medium importance – requires quick attention), and "White" (low importance – not urgent).

If it is important to be able to locate information about power system components affected by the alarm, then we can group alarms as shown in a sample hierarchy in Figure 1a. In Figure 1b, we present an ontology of control actions, based on a classification of control actions provided in [3]. We note that normal control actions can be either automatic or manual, while emergency control actions are most often automatic.

To complete the association, we introduce a relation, *hasControl*, which is defined between leaf nodes in the alarm hierarchy and leaf nodes in the control action hierarchy. This relation is a many-to-many mapping between a set of alarms and a set of possible control actions that can be used to respond to these alarms. For example, as shown in the figure, both "generator outage" and "line trip" alarms have the associated control action, "load shedding".

C. Ranking Functions

We use each severity index computation as a ranking function, R_i, and apply this function to a stream of PMU data. Since each alarm is flagged in the data, we aggregate all the alarms and assign a rank score to each alarm.

After a pre-defined time interval, we re-order the aggregated alarms based on their rank scores resulting in an ordered set of control actions. Since we operate in a stream computing environment, we apply multiple ranking functions in parallel, thereby obtaining lists of alarm orderings, $\rho_1, \rho_2, \ldots, \rho_n$, which can be compared. The result of this comparison will enable operators to respond in real-time (or near real-time) to alarms and enact mitigating procedures.

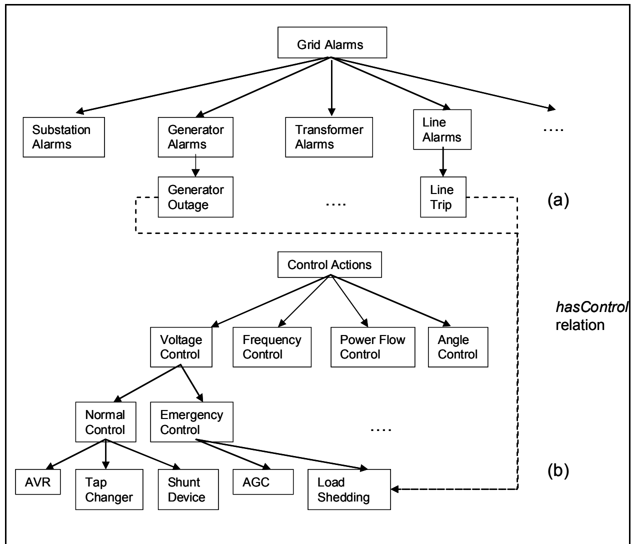

Figure 1. Ontology of alarms (a) and control actions (b).

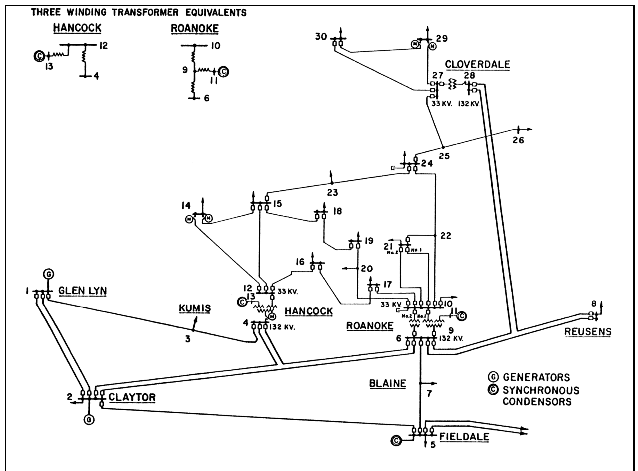

Figure 2. IEEE 30 bus test system.

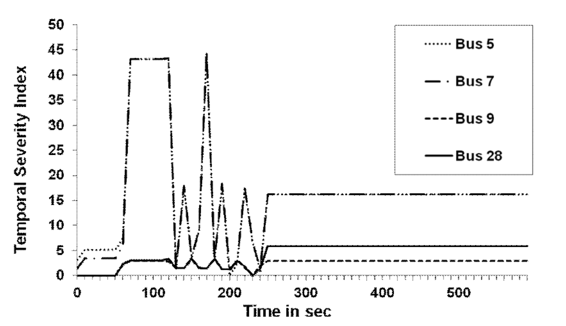

Figure 3. Temporal severity index computation.

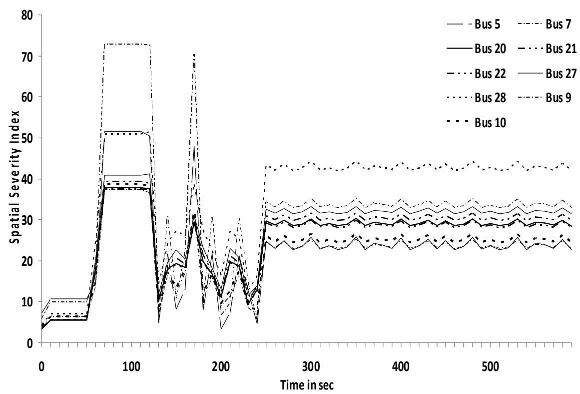

Figure 4. Spatial severity index computation.

Time (sec)	**Spatial severity alarm ranking**					**Temporal severity alarm ranking**				
70	7	5	28	27	22	7	5	2	1	3
80	7	5	28	27	22	7	5	2	1	3
90	7	5	28	27	22	7	5	2	1	3
100	7	5	28	27	22	7	5	2	1	3
110	7	5	28	27	22	7	5	2	1	3
120	7	28	5	27	22	7	5	1	2	3
130	28	27	24	22	30	2	1	30	26	29
140	7	28	5	27	22	7	5	2	1	3
150	28	27	24	30	22	7	5	6	28	4
160	28	7	27	8	22	7	5	6	8	4

Table 1. Ranking of buses based on spatial/temporal severity.

Bus_i	Bus_j	**Spatial severity correlation coefficient**	**Temporal severity correlation coefficient**
1	2	0.913	0.799
1	5	0.502	0.406
2	5	0.428	0.300
8	11	0.968	0.759
8	13	0.976	0.831
11	13	0.991	0.850
5	7	0.995	0.999
5	9	0.929	0.389
5	28	0.741	0.172
7	9	0.416	0.949
7	28	0.778	0.193
9	28	0.933	0.859

Table 2. Pearson product-moment correlation coefficients.

Whereas alarm grouping or filtering has been commonly discussed in the literature, this is typically applied with respect to static criteria specified in pre-defined rules. In our work, each ranking function allows us to dynamically examine a group of alarms according to user (operator) specified criteria – for instance, the time to resolve the alarms as well as spatial and temporal properties. Why this is relevant becomes clear if we examine how the relative importance of different alarms as defined by these properties varies over some pre-defined time window. For our experiments, we simulated cascaded line trips on the IEEE 30 bus test system [10] (shown in Figure 2) for 10 minutes.

In Figure 3, we show a sample of four buses and their temporal severity during the 10 minute interval. From this diagram, we observe that the values of temporal severity for bus_7 and bus_5 are correlated. In Table 1, we observe that between 80 and 110 seconds, bus_7 and bus_5 have the highest spatial and temporal severity index scores, respectively. The strength of the spatial and temporal correlation (as computed from the Pearson product-moment correlation coefficients) for selected pairs of buses is given in Table 2.

In Figure 4 and Table 1, we show the buses that are most severely affected over time in terms of spatial severity. Looking at different time windows, we observe changes in the spatial severity of the buses. For example, initially, bus_{28} is not the most severely affected bus (and therefore, does not have the highest spatial severity index score). However, its spatial severity rank moves up with the passage of time.

We need to re-rank the alarms periodically as the relative ordering of the alarms changes over time. Consider the example of bus_2 and bus_5, which have similar temporal rank in the interval from 70 to 130 seconds; however, they have a low correlation co-efficient for the entire 10 minute interval. The duration of the time interval for re-ranking alarms will be determined by a number of factors, including the desired type of control action, and therefore, should be configurable in an operational system. For example, if the system provides recommendations for manual control actions to the operator, re-ranking alarms too often may not be productive. Under the fault conditions described above, the operator may decide to re-rank alarms every 5 minutes.

The operator may also be interested in observing correlations across different *strata* of buses, e.g., "buses connected to generators". This property can be qualified even further as (*a*) "topologically adjacent buses connected to generators" and (*b*) "topologically non-adjacent buses connected to generators". Hence, we rank alarms based on *higher-order properties*, i.e., properties of the bus associated with an alarm.

III. EXPERIMENTS

To demonstrate the feasibility of our approach, we evaluated it on the benchmark IEEE 30 bus test system [10] as shown in Figure 2. We created a cascading scenario starting with a 3-phase-to-ground fault (dead short circuit) for 100 ms near bus_2 on $line_{2\text{-}5}$. A fault was initiated at 60 seconds and was cleared at 60.1 seconds by tripping $line_{2\text{-}5}$. Along with the tripping of $line_{2\text{-}5}$, 20 MW load was lost at bus_2. There was a sudden voltage dip due to the short circuit. Even though voltage recovered, the generators were accelerating gradually due to electromechanical dynamics. At time t =120 seconds, $line_{2\text{-}6}$ tripped due to power swing. The tripping of two important lines, $line_{2\text{-}5}$ and $line_{2\text{-}6}$, resulted in severe overload on $line_{2\text{-}4}$ and $line_{1\text{-}3}$. Eventually, the overload relay tripped $line_{2\text{-}4}$ and $line_{1\text{-}3}$ soon after $line_{2\text{-}5}$ tripped. The cascaded tripping of 4 lines – $line_{1\text{-}3}$, $line_{2\text{-}4}$, $line_{2\text{-}6}$ and $line_{2\text{-}5}$ – within a few minutes divided the system into two islands. Bus_1 and bus_2 formed one island and the remaining buses formed the other island. The first island collapsed due to overfrequency (i.e., generation was much higher than the load). The second island also collapsed due to underfrequency (i.e., load was higher than generation). For the given scenario, multiple alarms were generated and the proposed severity-based alarm rankings (Top 5 alarms per ranking method shown in Table 1) can be validated according to different criteria.

We observe that the alarms in the spatial severity and temporal severity rankings shown Table 1 belong to different groups. In the spatial severity alarm ranking, alarms on bus_5 and bus_7 are overcurrent alarms whereas alarms on bus_{28}, bus_{27} and bus_{22} are undervoltage alarms. In the temporal severity ranking, alarms on bus_5 and bus_7 are overcurrent alarms, alarms on bus_1 and bus_2 are overfrequency alarms, and the alarm on bus_4 is an undervoltage alarm.

First, we validate each alarm ranking in terms of the *reduction in the total number of alarms*. We do this by simulating the associated control actions for the alarms, one at a time using power flow simulation. In the spatial alarm ranking, for example, the alarm on bus_7 is fixed with the following control actions, namely, by minimizing congestion on $line_{5\text{-}7}$ by decreasing generation at bus_5 and increasing generation at bus_2. Even though this resolves the second alarm, the remaining alarms still exist. In the temporal ranking, if we resolve alarms on bus_5 and bus_2 with two control actions, by 20 MW generation backing at bus_1 and generation rescheduling at bus_2 and bus_5, all the remaining alarms are resolved automatically.

As discussed in Section II, the ontology has links from alarms types to control actions. This knowledge is pre-computed so that an automatic recommendation of control actions is generated per alarm ordering. When there are multiple control actions for a specific alarm type (as described above), the system enumerates the possible orderings of these control actions and evaluates them using power flow simulation (this can be done efficiently with the parallelism of stream computing). Using the simulation results, the system determines the ordering of alarms and associated control actions that minimizes the number of remaining alarms and continuously adapts the ontology with this operational knowledge. In this example, we have shown how proactively resolving alarms through simulating control actions not only can identify the "better" alarm ranking (in this case, the temporal ranking), but also pinpoint the "causal alarms" – alarms which when resolved minimize the total number of remaining alarms.

While the simulation-based metric is powerful and can also be considered a time-based metric – in an operational system, the user (operator) would want to minimize the total number of alarms, and consequently, the time needed to bring the power system back to a stable state faster – it may not be the only criterion for evaluation. For instance, the operator may be interested in monitoring "priority" buses (e.g., buses connected to generators) for alarms. For the IEEE 30 bus test system, such a criterion would rank alarms on any of the buses, bus_1, bus_2, bus_5, bus_8, bus_{11}, and bus_{13}, before alarms on the other buses (we refer to this as the priority ordering).

It is possible to measure how the severity rankings compare to some default order. For example, if we assume the default order to be the priority ordering of buses then, we can compare this to any severity ranking at time, t_i. For example, given the rankings in Table 1, starting at t=70 seconds, we can measure how dissimilar the spatial and temporal rankings are to the default order. The ranking method that results in lower average dissimilarity is rated as the "better" method.

We measure the dissimilarity between ordered lists using both *Kendall's tau* and *Spearman's footrule* [11]. For Kendall's tau, an inversion between a pair of elements, i and j, in a list, ρ, is defined such that $i<j$ but $\rho(i)>\rho(j)$. Kendall's tau is a count of the total number of inversions in ρ. The formula for computing this metric is as follows:

$$K(\rho) = \sum_{i<j} 1_{\rho(i)>\rho(j)} \tag{6}$$

In our case, we use a variant of Kendall's tau – we define an inversion to be any priority alarm not appearing in the Top 5 ranking:

$$K(\rho) = \sum_{i<j} 1_{\rho(i)>5} \tag{7}$$

For *Spearman's footrule*, displacement is the distance an element moved from its original position in the default order. Therefore, the total displacement is computed as follows:

$$F(\rho) = \sum_{i} | i - \rho(i) | \tag{8}$$

For the time interval from t=70 to 160 seconds in Table 1, we note that there were alarms observed for all the priority buses. Therefore, at any time t, each priority alarm has a rank in both the spatial and temporal orderings. (Note that all these ranks are not shown in Table 1). In Table 3, we compute Kendall's tau and Spearman's footrule values for each time instant.

We observe that the average Kendall's tau value for the spatial ranking (4.2) is greater than the temporal ranking (2.3). Similarly, for Spearman's footrule metric, we find the average value for spatial ranking (105.3) is greater than the temporal ranking (61.3). According to both metrics, temporal severity ranking is the "better" ranking with lower average dissimilarity scores. This also supports our initial evaluation result using the simulation-based metric, which showed that the temporal severity ranking minimized the time needed to bring the power system back to a stable state faster as alarms are progressively resolved.

Time (sec)	Spatial severity alarm ranking		Temporal severity alarm ranking	
	Tau	Footrule	Tau	Footrule
70	4	96	2	52
80	4	96	2	52
90	4	96	2	64
100	4	96	2	52
110	4	96	2	52
120	4	97	2	57
130	5	130	2	95
140	4	105	3	67
150	5	130	2	68
160	4	111	4	54

Table 3. Comparison of spatial/temporal severity rankings.

To summarize, using the simulation-based metric as well as (7) and (8), the operator is able to evaluate alarm orderings under different sets of conditions. With the parallelism provided by the stream computing platform, it is possible to generate multiple alarm orderings at run-time and to choose the "best one". By simulating control actions, the operator can proactively estimate the time needed to resolve alarms in a given ordering. In addition, the operator can check for specific conditions such as whether an alarm ranking preserves the property that priority alarms are given precedence and/or are collocated or whether alarms on vulnerable assets are resolved quickly.

Our approach is extensible as new alarm orderings can be introduced into the computational framework with relative ease. It is also flexible as the selection criteria for the best ordering can change over time and is configurable by the user. Consequently, temporal severity, spatial severity, or any other ranking method can be evaluated dynamically according to specific properties or criteria that the operator decides is important under current operating conditions.

IV. Related Work

Early alarm management systems were focused on logical analysis or rule-based processing of alarms [4]. For example, one way to reduce alarms is by logically grouping related alarms – this has sometimes been referred to as alarm "suppression" or "filtering" in the literature [1]. There are multiple rules for alarm filtering ranging from duration-based to static priority assignment [3]. More recent approaches for alarm reduction have focused on the extraction of features from raw data to understand the causal conditions behind alarms [2] and the use of inductive learning to find the root cause (failure tree) behind alarms [12].

V. Conclusions

In this paper, we have described how to order alarms on the power grid in real-time according to different severity measures and how to evaluate these orderings according to specific properties of interest to the user. Since control actions are automatically associated with the alarms, each ordering of alarms also suggests an ordering of control actions that should be taken by the operator. Simulation-based evaluation metrics such as the one described in the paper also enable the operator to identify causal alarms, which when resolved, minimize the total number of remaining alarms, thereby bringing the power system back to a secure state faster.

Describing a conceptual algorithm for alarm processing that also offers recommendations and decisions, Kyriakides, Stahlhut, and Heydt [3] state:

Obviously, like all other power system controls, decisions on when and if to implement are rarely made on the basis of a single consideration; rather, a range of considerations need to be assessed in control system designs.

Our intent is not to fully automate the decision-making process exercised by operators for power systems control, but rather to supplement it by formally characterizing and quantifying the "range of considerations" that operators must consider before making a decision on how to handle alarms.

References

[1] M. L. Bransby and J. Jenkinson, "The Management of Alarm Systems: A Review of Current Practice in the Procurement, Design and Management of Alarm Systems in the Chemical and Power Industries," Bransby Automation Ltd, Cirencester, England, 1998.

[2] M. Kezunovic and Y. Guan. "Intelligent Alarm Processing: From Data Intensive to Information Rich," *Proceedings of HICSS 2009 (42nd Hawaii International Conference on System Sciences)*, Hawaii, pp. 1-8, Jan. 2009.

[3] E. Kyriakides, J. W. Stahlhut, and G. Heydt. "A Next Generation Alarm Processing Algorithm Incorporating Recommendations and Decisions on Wide Area Control," *IEEE Power Engineering Society General Meeting*, pp. 1-5, 2007.

[4] R. W. Bijoch, S. H. Harris, T. L. Volkmann, J. J. Bann, and B. F. Wollenberg, "Development and implementation of the NSP intelligent alarm processor," *IEEE Trans. Power Systems*, vol. 6, no. 2, pp. 806-812, May 1991.

[5] A. H. Shoop and S. Silverman, "A real time alarm processor," *Electrical Power and Energy Systems*, vol. 14, no. 2/3, pp. 108-113, 1992.

[6] R. Khosla and T. Dillon, "Learning knowledge and strategy of a neuro-expert system architecture in alarm processing," *IEEE Trans. Power Systems*, vol. 12, no. 4, pp. 1610-1618, Nov. 1997.

[7] S. McArthur, R. Dysko, J. McDonald, S. Bell, R. Mather, and S. Burt, "The application of model based reasoning within a decision support system for protection engineers," *IEEE Trans. Power Delivery*, vol. 11, no. 4, pp. 1748-1754, Oct. 1996.

[8] R. Rea. IBM InfoSphere Streams. "Redefining Real Time Analytic Processing", http://public.dhe.ibm.com/software/data/sw-library/ii/whitepaper/InfoSphereStreamsWhitePaper.pdf.

[9] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. "Models and Issues in Data Streams Systems", *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, New York, NY, 2002.

[10] http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.htm

[11] P. Diaconis and R. L. Graham. "Spearman's Footrule as a Measure of Disarray," *Royal Statistical Society Series B*, vol. 32, No. 24, pp. 262-268, 1977.

[12] O. Aizpurua, R. Galan, and A. Jimnez. "A New Cognitive-Based Massive Alarm Management System in Electrical Power Administration", *Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems*, Mexico, Apr. 28-30, 2008.