NASA Technical Memorandum 103148

Automating Security Monitoring & Analysis for Space Station Freedom's Electric Power System

James L. Dolce
Lewis Research Center
Cleveland, Ohio

Dejan J. Sobajic and Yoh-Han Pao
Case Western Reserve University
Cleveland, Ohio

Prepared for the
25th Intersociety Energy Conversion Engineering Conference
cosponsored by the AIChE, SAE, ACS, AIAA, ASME, and IEEE
Reno, Nevada, August 12-17, 1990

E-5502

Automating Security Monitoring and Analysis for Space Station Freedom's Electric Power System

James L. Dolce
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Dejan J. Sobajic and Yoh-Han Pao
Case Western Reserve University
Cleveland, Ohio 44106

Abstract

Operating a large, space power system requires classifying the system's status and analyzing its security. Conventional algorithms are used by terrestrial electric utilities to provide such information to their dispatchers, but their application aboard Space Station Freedom will consume too much processing time. We present a new approach for monitoring and analysis using adaptive pattern recognition techniques. This approach yields an on-line security monitoring and analysis algorithm that is accurate and fast; and thus, it can free the Space Station Freedom's power control computers for other tasks.

1 Operational Oversight

The electric power system aboard Space Station Freedom is part of an infrastructure for space experiments research. Consequently, its large generation capacity (75kW), primary and secondary distributions networks, and changeable loads make this system behave like terrestrial power systems. Operating these terrestrial power utilities is based upon command and control strategies for each of four operating states: *normal, preventive, emergency*, and *restorative*. Space Station Freedom's power system will also incorporate such strategies. Oversight decisions must be made to choose those strategies that will keep the power system operating within tolerance and as productive as possible under all circumstances [1]. The classification of electric power system operation as a set of behavioral states (Figure 1) provides a heuristic framework for guiding such decisions [2], [3], and [4]:

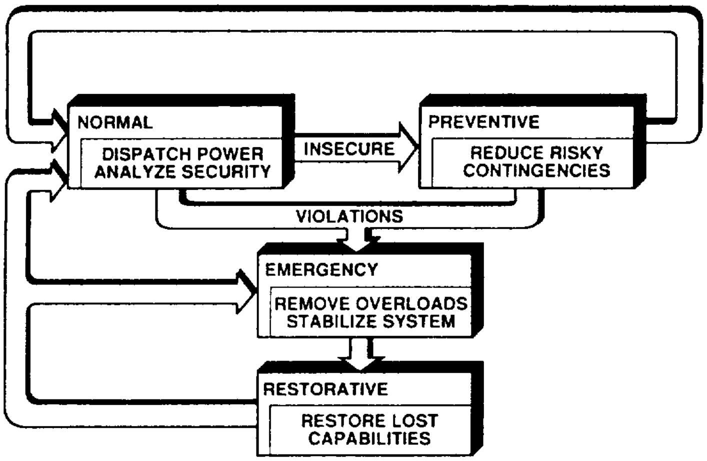

Figure 1: System Operating States

- **Normal State.** When the power system is in the *normal* state, all operating constraints are satisfied. The basic task is to dispatch as much of the generated power as possible without jeopardizing power system or Space Station integrity.
- **Preventive State.** When the power system is in the *preventive* state, an impending contingency has been detected and the objective is to organize a return to normal system operation.
- **Emergency State.** If failures occur that overload the system such that voltage cannot be maintained at safe minimums, the system is in the *emergency* state. The objective for this condition is to stop degradation while satisfying as much demand as possible [3, 4]. Aboard Space Station Freedom, this objective is satisfied automatically by the overload protection equipment. Once the automatic reflexive actions have removed the overloads and have stabilized the operation, restoring lost loads and reconnecting the

system can begin [2].

- **Restorative State.** When an operating limit has been violated and after the automatic protection equipment has stabilized the system, a process begins that returns system capacity as much as practicable. This is the *restorative* state of operation. The objective is to reconfigure the power system to supply the most important of the loads without overloading the equipment on line. Meanwhile, plans are prepared to repair or replace failed equipment and to return the system to full capability.

2 Security and Automation

2.1 Security Monitoring

Determining whether or not the system is in the *normal* state is called *security monitoring.* The task consists of acquiring data from the system: line currents, node voltages, phase angles, and power injections; calculating the operating margins for all critical elements in the system; and judging whether or not these margins are within tolerances. If so, the system is classified *normal.* If not, the systems is in *emergency* [1].

2.2 Security Analysis

It is during the *normal* state that security is analyzed. System security analysis is a risk assessment. It examines the liabilities of continued operation by identifying contingencies and estimating their consequences. The contingencies are disturbances that could lead to overloads, voltage degradation, source shutdown, or load shedding. If the risk of continued operation is judged acceptable, the system is classified *secure* and system operation proceeds according to the current plan. If there are risky contingencies, the system is judged *insecure* and preventive control strategies are implemented (preventive state).

Three distinct activities are required to analyze system security:

1. **Generate contingencies:** Worrisome failures that are present under all operating conditions as well as operating-state dependent failures such as transmission outages are compiled and submitted for analysis.

2. **Analyze contingencies:** The system's operating margins are calculated for each of the failures in the preceding activity.

3. **Judge security:** A system is secure if there are no contingencies that result in an *emergency* state. If the operating margins calculated in activity 2 are insufficient, the system is judged *insecure.*

2.3 Conventional Automation

To date, there has been little success in automatically generating the contingency list. The task is performed by enumerating credible failures taken as single events followed by an enumeration of the next set of credible failures [1]. For particularly risky scenarios, the single event failures are expanded to include coincident failures followed by the next set of credible failures. In the case of Space Station Freedom's power system, a two-failure-tolerant design generates a very large list of contingencies.

Given a list of failures, automatic selection algorithms are used to rank the entries in decreasing order of severity. The gist is to evaluate operating margins for the cases in the ranked list taking each case one by one until a case is found not to produce abnormal operating conditions. The rest of the cases in the list will be less severe and need not be analyzed [5, 6, 7] [8, 9].

The workhorse algorithm for security monitoring and security analysis is the fast, decoupled load-flow [10]. This algorithm uses network topology and loading conditions to calculate the all the remaining voltage, current, and phase parameters in the power system. In the case of *security monitoring*, the data is a set of measurements from the system's transducers. For *security analysis*, it is those same measurements with appropriate modifications (either to the network topology or to some of the measurements themselves) that are specified as input to the algorithm which then estimates the contingency's outcome.

The monitoring and security analysis for Space Station Freedom's power system is an involved task. We are concerned that the computational work load imposed by security monitoring and analysis will preclude its incorporation aboard Freedom. Using satellite telemetry to support ground-based monitoring will only introduce unacceptable delays and monopolize station-to-ground communications. Operational oversight requires the rich information from timely security monitoring and analysis. Without it the power system will be less productive—certainly less secure.

We wish to pursue investigations of new technologies that will eventually automate all of the aspects of security monitoring and analysis and improve the performance of existing approaches. Our first step is to apply artificial neural networks to solve the secu-

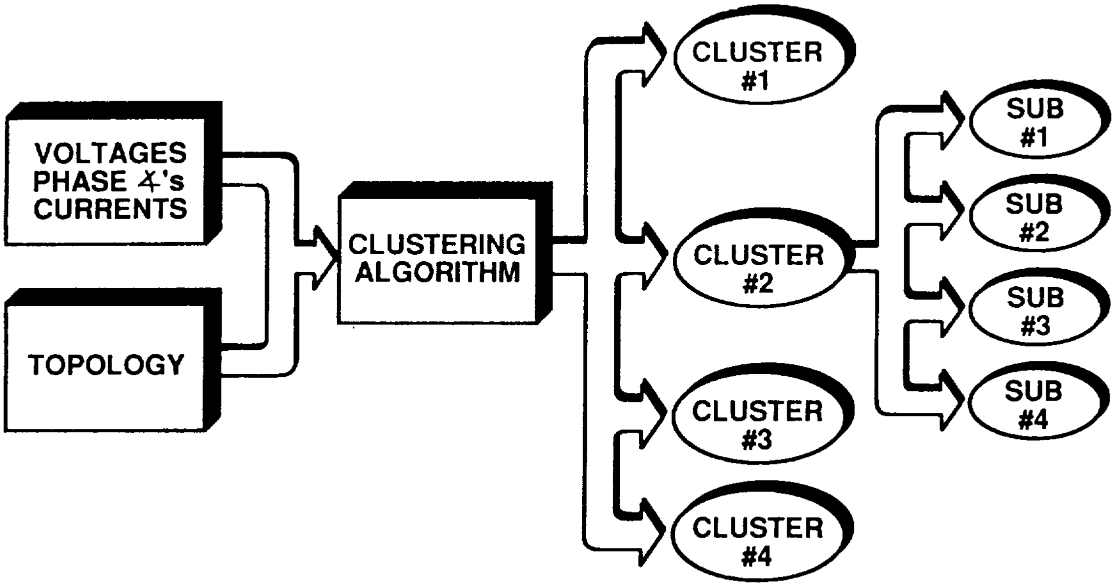

Figure 2: Pattern Clustering

rity monitoring task. In so doing, we will also provide a tool for evaluating the contingencies in the security analysis task.

3 A Neural Network Approach

3.1 Monitor and Analyzer

The fast, decoupled load-flow needs to be faster. Particularly when it has to evaluate a large number of contingencies one at time. Further, it needs to consume less computer resource while doing so. Because the load-flow algorithm uses iteration to converge upon a solution, its solution time can change with the task at hand. All that the security monitoring and analysis tasks require is for load-flow to classify an input data pattern to be within tolerances. An artificial neural-network, trained to recognize the same input pattern and recall the corresponding tolerances would significantly reduce computational burdens. Rather than rely upon a search strategy to iterate and converge to a solution as does load-flow, the trained network calculates sets of sums, differences, and products only once. The operation of such a neural-network algorithm has three data processing stages:

3.1.1 Clustering Similar Patterns

The first processing stage of the algorithm is to determine the similarity between the input data pattern and one of four reference (prototype) clusters (Figure 2). The input pattern is a collection of node voltages, phase angles, currents, and admittances. The input features are scaled in per-unit-values; that is, the actual value for the variable divided by a nominal operating value. The exemplar for the prototype cluster has the same features as the input pattern. The following algorithm is used to assign the input pattern to the closest cluster using Euclidean distance as a metric:

Let:

I = total number of features in the input pattern.

J = total number of clusters.

$x_i = i^{th}$ feature of the input pattern.

$b_{ji} = i^{th}$ feature of the j^{th} cluster's prototype.

$ED_j = \sqrt{\sum_{i=1}^{I}(b_{ji} - x_i)^2}$ —the Euclidean distance from the cluster's prototype to the input pattern.

Compute $ED_j \ \forall j = 1, 2, \ldots, J$

Find $\min(ED_j)$ and assign the input pattern to this cluster.

For the second cluster, there are four subclusters. (See Section 3.2.2.) Once cluster 2 membership is established, the input pattern is assigned to the closest subcluster.

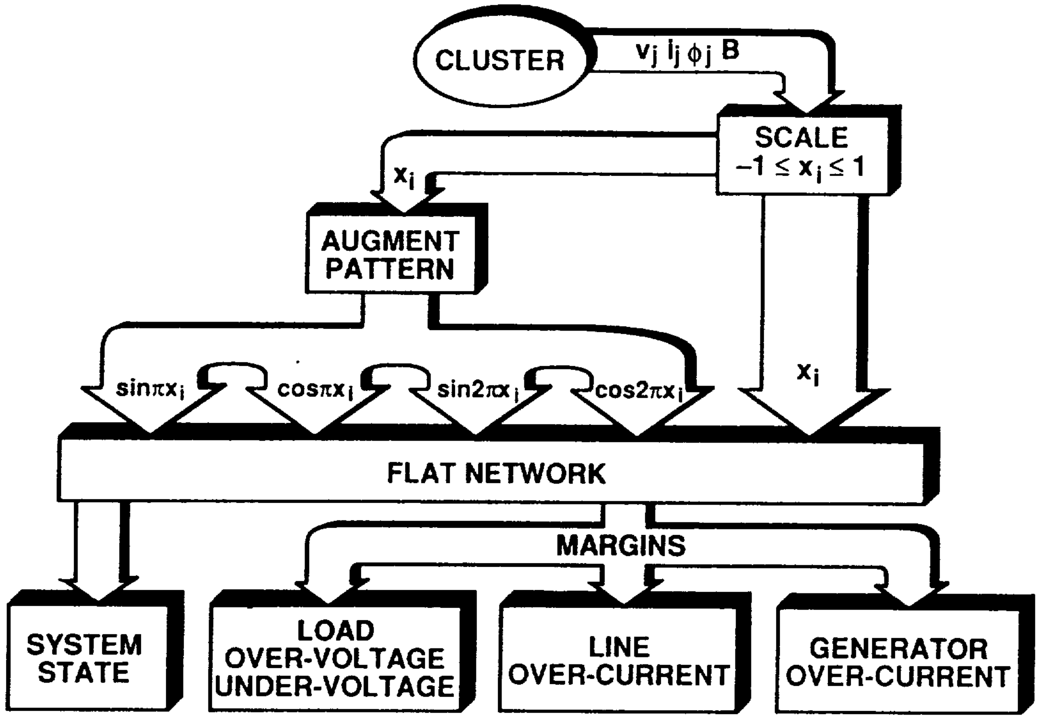

Figure 3: Associative Recall

3.1.2 Scale and Augment the Input Pattern

The second processing stage scales and augments the input pattern. These operations, shown in Figure 3, prepare the pattern for use with a "flat" (no hidden layers) network. Each feature is scaled according to

$$x_i^* = 2\left[\frac{x_i - x_i(min)}{x_i(max) - x_i(min)}\right] - 1$$

$$x_i^* \Rightarrow x_i \; scaled$$

where the minimum and maximum values are from the unsupervised learning phase of development (Section 3.2.2). Next, the pattern is augmented by adding features $\sin \pi x_i, \cos \pi x_i, \sin 2\pi x_i, \cos 2\pi x_i$ for $i = 1, 2, \ldots, I$. This results in a pattern with $5I$ features.

3.1.3 Recall Margins from "Flat" Network

In the third processing stage, the augmented and scaled pattern is directed to a "flat" network that has been trained to recall output patterns for all inputs belonging to a particular cluster. (See Figure 3.) The output features for this network are the operating margins for the power system's critical parameters and a classification of *normal* or *emergency*.

Let:

K = total number of output features.

w_{ki} = i^{th} feature's contribution to output feature k.

o_k = k^{th} output feature.

θ_k = threshold for sigmoidal activation function for output feature k.

S_k = slope parameter for sigmoidal activation function for output feature k.

$net_k = \sum_{i=1}^{5I} w_{ki} x_i + \theta_k$

Then calculate $\forall \; k = 1, 2, \ldots, K$ the state classification and operating margins using:

$$o_k = \frac{1}{1 + e^{-S_k net_k}}$$

3.2 Design and Development

The development approach is shown in Figure 4. Training cases are synthesized and analyzed with fast, decoupled load-flow to produce sets of input patterns and their corresponding operating margins. These input patterns are grouped into clusters and assigned to a separate network. The first stage in this process

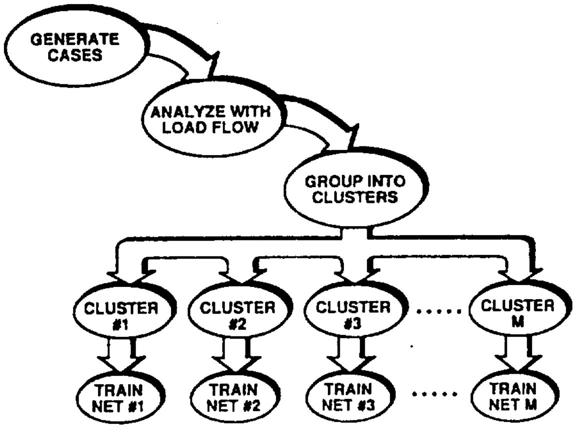

Figure 4: Development Approach

groups similar patterns together, augments their features, and directs them to a "flat" network trained for this class of input pattern. In the second stage, the "flat" network recalls the appropriate operating margins for a given input pattern. To develop the clusters, we used unsupervised learning [11]. To augment the input features we applied nonlinear transformation (functional-expansion) which improves network learning. To train the "flat" network for associative storage and recall we employed supervised learning. This combination of feature augmentation and supervised learning is an application of Pao's functional-link net [13, 14, 15, 16].

3.2.1 Input and Output Patterns

To synthesize the training patterns for the artificial neural network, we used Stott's fast, decoupled loadflow [10] to produce over 2000 sets of data for the power system shown in Figure 5. Load and generation levels were varied between 0 and 120% of the normal rating with generator buss voltages maintained within 10% of normal. Included in the sets were cases with one of the lines removed from the ring (between 5-12-6, 7-13-8, and 9-11-10). The input pattern features chosen for training our network were voltages and phase angles for nodes 1 through 10 and 14 through 16, voltage for slack-buss 17, currents for both inverters, the current for each of the four loads, and 6 of the diagonal terms in the admittance matrix (specifies the topology)—a total of 39 input features. The output pattern features were over- and under-voltage margins for each of the four loads, over-current margins for the inverters and the seventeen lines, and a classification of operating state as either *normal* or *emergency*—a total of 28 output features.

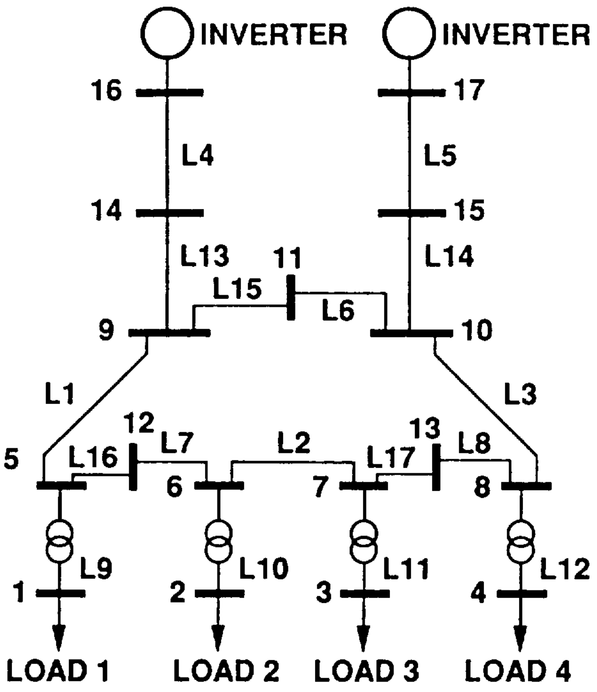

Figure 5: Power System Example

3.2.2 Unsupervised Learning

Training begins with the unsupervised clustering of the input data. The following algorithm was used:

Let:

I = total number of features in the input pattern.

P = total number of input patterns.

$X_p = x_{p1}, x_{p2}, \ldots, x_{pI}$ —the input pattern.

$x_{pi} = i^{th}$ feature of the p^{th} input pattern.

$B_j = b_{j1}, b_{j2}, \ldots, b_{jI}$ —the cluster's prototype.

$b_{ji} = i^{th}$ feature of the j^{th} cluster's prototype.

$ED_{jp} = \sqrt{\sum_{i=1}^{I}(b_{ji} - x_{pi})^2}$ —the Euclidean distance from the cluster's prototype to the input pattern p.

$\overline{ED}$ An arbitrarily chosen "radius" of similarity for membership in a cluster.

N_j = the number of patterns in j^{th} cluster.

Initialize: $J = 1, B_1 = X_1, N_1 = 1$

For the pattern X_p $p \neq 1$:

1. Compute:

ED_{jp} $\forall\, j$ and for pattern p.

2. $\forall\, j$, find $\min\{ED_{jp}\} \leq \overline{ED}$ and assign pattern 0 to this cluster.

Modify the cluster's prototype by:

$$b_{ji}(n+1) = \frac{N_j(n)}{1+N_j(n)} b_{ji}(n) + \frac{1}{1+N_j(n)} x_{pi}$$

Increment $N_j(n+1) = 1 + N_j(n)$, and repeat 1 and 2 with a new input pattern.

Otherwise, spawn a new cluster

$$B_{J+1} = X_p \quad N_{J+1} = 1$$

and repeat 1 and 2 with a new input pattern.

Because the cluster prototypes are adjusted when input patterns are assigned in step 2, some of the patterns initially within the cluster may fall outside of the cluster when the algorithm is completed. The algorithm is run iteratively to reassign these orphans using the current number of clusters, J, and their prototype values, B_j, as initial conditions.

To complete the unsupervised learning phase, all of a cluster's patterns are scanned to find the maximum and minimum values for each feature. The pattern features are then "zeroed and spanned" to produce patterns whose features all lie between -1 and 1. The maximum and minimum values of each feature are stored for use in "zeroing and spanning" input patterns during the security monitoring and analysis phase (Section 3.1.2).

For our problem, we swept through a range of $\overline{ED}$'s from 1 to 4. The number of clusters that formed ranged from 30 to 1 varying approximately as $\frac{1}{(\overline{ED})^2}$. We began our analysis by choosing the four cluster case. During the supervised learning phase (Section 3.2.3), the patterns in cluster 2 could not satisfactorily recall the corresponding operating margins with a "flat" network; so we subdivided cluster 2 into four clusters and trained an individual network for each [12].

3.2.3 Supervised Learning

Each cluster's scaled pattern is enhanced using four sets of orthonormal functions:

$$\sin \pi x^*_{pi}, \cos \pi x^*_{pi}, \sin 2\pi x^*_{pi}, and \cos 2\pi x^*_{pi}$$

with pattern p, feature i, and * $\Rightarrow$ scaled. No new information is introduced but the pattern size is increased to 195 features. The objective is to add enough dimensionality so that a "flat" network can be used for associative recall of the load-flow calculations [15].

The "flat" network is trained one output at a time using the generalized delta rule [17] and the following algorithm:

Let:

t_{pk} = k^{th} output feature's desired (target) value for the p^{th} input pattern.

$w_{ki}, net_{pk}, \theta_k, S_k$, and o_{pk} as in section 3.1.3

$E_{pk} = \frac{1}{2}(t_{pk} - o_{pk})^2 = k^{th}$ output's error for the p^{th} input pattern.

$\delta_{pk} = -\frac{\partial E_{pk}}{\partial net_{pk}} = (t_{pk} - o_{pk})o_{pk}(1 - o_{pk})$

For the enhanced, scaled pattern $X_p^{e,*}$ and its desired output t_{pk}, and for initial w_{ki}, S_k, and θ_k:

1. Calculate:

$o_{pk}(m) = \frac{1}{1+e^{-S_k(m)net_{pk}(m)}}$ and E_{pk} $\forall\, p = 1, \ldots, P$; and then,

$$E_k = \sum_{p=1}^{P} E_{pk}$$

2. If $E_k \geq$ some maximum E, then:

Calculate the optimal learning rate: η_k^o

$w_{ki}(m+1) = w_{ki}(m) + \eta_k^o \sum_{p=1}^{P} \delta_{pk} x^*_{pi}$ $\forall\, i = 1, \ldots, I$

$\theta_k(m+1) = \theta_k(m) + \eta_k^o \sum_{p+1}^{P} \delta_{pk}$

$$net_{pk}(m+1) = \sum_{i=1}^{195} w_{ki}(m+1)x^*_{pi} + \theta_k(m+1)$$

$\forall\, p = 1, \ldots, P$

Calculate the optimal slope rate: ρ_k^o

$S_k(m+1) = S_k(m) + \rho_k^o \sum_{p=1}^{P} \delta_{pk} net_{pk}(m)$

Repeat 1 and 2.

The optimal learning rate is calculated with the following algorithm:

1. Pick an initial value for $\eta_k = \eta_{k,r}$; $r = 0$. Given $w_{ki}(m)$, $E_{k,r-1} = E_k$, and $\theta_k(m)$:

2. Calculate:

$$w_{ki}(m,r) = w_{ki}(m) + \eta_{k,r} \sum_{p=1}^{P} \delta_{pk} x^*_{pi} \;\; \forall\, i$$

$$\theta_k(m,r) = \theta_k(m) + \eta_{k,r} \sum_{p=1}^{P} \delta_{pk}$$

$$net_{pk}(m,r) = \sum_{i=1}^{195} w_{ki}(m,r)x^{*}_{pi} + \theta_k(m,r)$$

$$o_{pk}(m,r) = \frac{1}{1+e^{-S_k(m)net_{pk}(m,r)}} \; \forall p$$

$$E_{k,r} = \sum_{p=1}^{P} \frac{1}{2}\{t_{pk} - o_{pk}(m,r)\}^2$$

3. For:

$$E_{k,r} - E_{k,r-1} > 0 \quad \eta_{k,r+1} = \frac{1}{2}\eta_{k,r}$$

$$E_{k,r} - E_{k,r-1} \leq 0 \quad \eta_{k,r+1} = 2\eta_{k,r}(n)$$

4. Repeat 2 and 3 using increasing $\eta_{k,r}$ to locate the first point where E_k decreases and then the first point where E_k increases: $\eta_{k,r}(E_k$ first decreasing) and $\eta_{k,r}(E_k$ first increasing).

5. Fit a parabola through $\eta_{k,0}$, $\eta_{k,r}(E_k$ first decreasing), and $\eta_{k,r}(E_k$ first increasing) such that:

$$E_k \approx a\eta_k^2 + b\eta_k + c$$

6. The optimum learning rate is at the stationary point $\eta_k^o = \frac{-b}{2a}$.

The optimal slope rate ρ_k^o is calculated with the same algorithm as the optimal learning rate but with step 2 modified so that

$$S_k(m+1,r) = S_k(m) + \rho_{k,r}\sum_{p=1}^{P}\delta_{pk}net_{pk}(m)$$

and $\rho_{k,r}$ are computed rather than $net_{pk}(m,r)$ and $\eta_{k,r}$.

To learn a single output feature's response, the networks required all of a cluster's training patterns presented approximately 200 times. Our computation time was roughly 3 minutes per output feature using a Sun–IV workstation.

3.2.4 Performance

We submitted test cases to both the fast, decoupled load-flow and the trained network. The test cases had *not* been used for training. In general, the network would recall the operating margins within better than ±5% of the load-flow calculated value. The processing times (using a PC/AT 386/20MHz) were 500 μsec. for the network and 5 sec. for the load-flow algorithm. For our power system example, the artificial neural network could monitor system security and evaluate over 9,900 contingencies before the load-flow algorithm could complete one case. The penalties for the network are accuracy and data storage. (Our simple model requires about 40,000 words to store the w_{ki}'s.) The accuracy can be improved through more clustering and reducing the scope of the problem solved by the network. The former increases storage requirements, the latter may increase solution time.

4 Coda

Rules or Nets?

Both rule-based systems and artificial neural networks are used for pattern recognition. Of the two, the neural network is ideally suited to the security monitoring task. For this task, the input information does not contain explicit measurements of all of the parameters used for computing operating margins (every line current for example), and so, the relation of input features to output features is quite difficult to express as a system of rules in a production system. Consider the operating state classification task: without additional information, any input can be either in the *normal* or in the *emergency* state. This is because the information contained in a single feature is not sufficient to resolve between *normal* and *emergency* unless it is evaluated concurrently with the other input features. Many rules would be required to quantize the information into "crisp" intervals that a production system could recognize accurately. Such a rule-based detector would be difficult to maintain. We contend that an artificial neural network is the better solution for this task [18].

Prospective

This small example demonstrates the computational advantage of associative recall using artificial neural networks. The method will allow quick and efficient on-line security monitoring, and thus, provide information for operational oversight without compromising other tasks competing for processing time. The processors for Space Station Freedom will need ample memories to store the information required by a competent neural network application. Present space-qualified computers do not have the storage. Freedom's computers will have to be specified with adequate growing room for this new technology.

So far, we have only addressed the security monitoring function and provided a rapid means of evaluating contingencies. There is much work remaining. Compiling the failure list is amenable to automation either through rule-based pattern classifiers or through a trained network—possibly both. Ranking contingencies using conventional procedures is computationally intensive (Section 2.3) and can be facilitated by rule-based expert systems [19]. Integrating contingency generation, ranking, and analysis into an effective, automated process for on-line command and control applications has no precedent. We believe that automating these activities produces a comprehensive approach to operational oversight—one that will reduce our dependency on human involvement and significantly increase power system security.

References

[1] T. E. Dy Liacco, "System security: the computer's role," in *IEEE Spectrum*, IEEE 15, 43–50 (June 1978).

[2] L. H. Fink and K. Carlsen, "Operating under stress and strain," in *IEEE Spectrum*, 48–53 (March 1978).

[3] T. E. Dy Liacco, "The adaptive reliability control system," in *Transactions on Power Apparatus and System*, IEEE, PAS-86(5), 517–531 (May 1967).

[4] O. P. Malik, Discussion of T. Sakaguchi and K. Matsumoto, "Development of a knowledge based system for power system restoration," in *Transactions on Power Apparatus and Systems*, IEEE, PAS-102(2), 320–329 (Feb. 1983).

[5] G. C. Ejebe and B. F. Wollenberg, "Automatic contingency selection," in *Transactions on Power Apparatus and Systems*, IEEE, PAS-98(1), 97–106 (Jan./Feb. 1979).

[6] G. Irisarri, A. M. Sasson, and D. Levner, "Automatic contingency selection for on-line security analysis—real-time tests," in *Transactions on Power Apparatus and Systems*, IEEE, PAS-98(5), 1552–1559 (Sept./Oct. 1979).

[7] G. D. Irisarri and A. M. Sasson, "An automatic contingency selection method for on-line security analysis," IEEE, PAS-100(4), 1838-1844 (April 1981).

[8] T. A. Mikolinnas and B. F. Wollenberg, "An advanced contingency selection algorithm," IEEE, PAS-100(2), 608-617 (Feb. 1981).

[9] S. Vemuri and R. E. Usher, "On-line automatic contingency selection algorithms," IEEE, PAS-102(2), 346–354 (Feb. 1983).

[10] B. Stott and O. Alsac, "Fast Decoupled Load Flow," in *Transactions on Power Apparatus and Systems*, IEEE, PAS-93, 859–869 (May 1974).

[11] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," in *Applied Optics*, Vol. 26, 4919–4930 (1987).

[12] D. J. Sobajic, Y. Pao, W. Njo, and J. L. Dolce, "Real-time security monitoring of electric power systems using parallel associative memories," IEEE, International Symposium of Circuits and Systems, New Orleans, LA. (May 1–3, 1990).

[13] D. J. Sobajic, *Neural Nets for Control of Power Systems*, Ph.D. Thesis, Computer Science Dept., Case Western Reserve University, Cleveland, OH. (1988).

[14] Y. Pao, "Functional link nets: removing the hidden layers," in *AI Magazine*, 60–68 (Apr. 1989).

[15] Y. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA. (1989).

[16] Y. Pao and R. D. Beer, "The functional link net: a unifying network architecture incorporating higher order effects," International Neural Network Society, First Annual Meeting, Boston, MA. (Sept. 6, 1988).

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Willliams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, D. E. Rumelhart and J. L. McClelland, eds., pp. 318-362. MIT Press, Cambridge, MA. (1986).

[18] D. J. Sobajic, Y. Pao, and J. Dolce, "On-line monitoring and diagnosis of power system operating conditions using artificial neural networks," IEEE, International Symposium of Circuits and Systems, Portland, OR. (May 9-12, 1988).

[19] D. J. Sobajic and Y. Pao, "An artificial intelligence system for power system contingency screening," Center for Automation and Intelligent Systems Research, Case Western Reserve University Technical Report TR 87-118, Cleveland, OH. (1987).

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No. NASA TM-103148	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Automating Security Monitoring & Analysis for Space Station Freedom's Electric Power System	5. Report Date
	6. Performing Organization Code
7. Author(s) James L. Dolce, Dejan J. Sobajic, and Yoh-Han Pao	8. Performing Organization Report No. E-5502
	10. Work Unit No. 488-51-03
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	11. Contract or Grant No.
	13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the 25th Intersociety Energy Conversion Engineering Conference cosponsored by the AIChE, SAE, ACS, AIAA, ASME, and IEEE, Reno, Nevada, August 12-17, 1990. James L. Dolce, NASA Lewis Research Center. Dejan J. Sobajic and Yoh-Han Pao, Case Western Reserve University, Cleveland, Ohio 44106.

16. Abstract

Operating a large, space power system requires classifying the system's status and analyzing its security. Conventional algorithms are used by terrestrial electric utilities to provide such information to their dispatchers, but their application aboard Space Station Freedom will consume too much processing time. We present a new approach for monitoring and analysis using adaptive pattern recognition techniques. This approach yields an on-line security monitoring and analysis algorithm that is accurate and fast; and thus, it can free the Space Station Freedom's power control computers for other tasks.

17. Key Words (Suggested by Author(s)) Security analysis Neural networks	18. Distribution Statement Unclassified – Unlimited Subject Category 63

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 10	22. Price* A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161