

## 大規模機械システム状態監視保全のための 症候データベースに関する研究

### Development of Symptom Database for State-Based Surveillance and Maintenance of Large Scale Machine Systems

カトゥール ディアントノ, 高橋信, 北村正晴

Catur Diantono, Makoto TAKAHASHI, Masaharu KITAMURA

東北大学大学院工学研究科量子エネルギー工学専攻

Department of Quantum Science and Energy Engineering, Tohoku University

キーワード: データベース (Database)、Latent Mode、Alert Mode、Man Machine Cooperative System

連絡先: 〒 980-77 仙台市青葉区荒巻字青葉 東北大学大学院工学研究科 量子エネルギー工学専攻北村研究室  
Tel/Fax: 022-217-7907 E-mail: catur@mine1.qse.tohoku.ac.jp

#### 1. 序論(Introduction)

大規模機械システム(原子力プラント、大型加速器、化学プラントなど)では、機器故障を早期に検知、診断し的確に対応操作を行うことが重要である。この観点から監視診断システム、運転員支援システムなどの開発が進められてきた。これらのシステムでは、パターン照合、制御理論応用技法、ルールベース推論、モデルベース推論など、多様な手法が試みられてきているが、実用化にはなお一層の機能の拡充と、性能ならびに使いやすさの向上が期待されているのが実状である。この機能拡充に関連して、本研究ではこれまでの諸技法では顧みられなかった新しい機能、すなわち

1) 運転員が希望する任意の時点のデータ

検索と経時的照合、

- 2) その際におけるデータの必要な特性量への縮約(data reduction and compression)、
- 3) 感度を可変とした変化の検出(change and deviation detection)、

を実現するための症候データベース構築に関して検討を行う。

これらの検索、縮約、変化検出のようなタスクは、定型的な運転ならびに監視業務の中では明示的には要求されていない。しかしながら運転員の自発的なモニタリング活動としては日常的に行われる場合が多く、異常事象の予兆段階での検出と故障モード同定に重要な意義を有していることが最近のフィールドスタディで指摘されている<sup>1)</sup>。

人間・機械知能系としての機械システム安全性の向上を考える際には、このようなタスクの適切な支援も重要な課題である。

## 2. 課題の定義(Definition of the Problem)

観測データを周波数スペクトルや相関関数、伝達関数などに変換してしまうと、記憶容量を小さくできて都合であるが、後日の詳細解析が要求された場合に対応できない。したがって格納保存の対象とする観測データは、基本的には多変数時系列とし、この時系列データを必要に応じて適当な統計量、特徴量などに変換することが望ましい。観測データ供給源である計測系は、通常計装と診断用計装（軸振動モニタなど）の2種類に大別される上、それぞれの計装系に対してサンプリング時間幅も目的によって異なる複数の値が採用されることが必要であるため、データベースの設計には注意が必要である。

## 3. データベース基本設計

### 3.1 必要要件(Design Requirement)に関する考察

設計に関し考慮に入れることが必要な要件として、少なくとも以下のような項目があげられる。

- 1) 信号の種類と各種類毎のサンプリング時間、データベース格納の基本とするサンプル数、
- 2) 観測・記憶の周期（記憶時間間隔：1時間毎、1日毎、1週間毎、1月毎...）、
- 3) 縮約、変換の内容（STANDARD = APSD, CPSD, TF, ACF, CCF, APD, MOMENT）（DETAILED：FRACTAL, WAVELET, CAUSAL TRAJECTORY.....）、

などが考慮に入れる必要である。

これらはいずれも密接に関連しており、しか

も監視目的によって異なる可能性がある。また 3) について上記以外の新しい要請が生じた場合に対応できる柔軟性も必要である。

さらにこの他に配慮すべき問題として、データの構造化、インデックス化の方策（記憶、検索の効率化、長期記憶データベースと短期記憶データベースの構造決定および記憶媒体の選択）、標準 Template データベースの保存と Updating（正常データ、正常スペクトル他）機器運用、保守、変更情報の連結管理（Management of Information about system operation, maintenance and repair）などがある。またデータベースそのものではないが、このデータベースを利用した監視診断ソフトウェアとのソフトインタフェースに関して次のような応用性能を念頭に置くべきである。

#### 1) 監視モードの選択

監視タスクのためのインタフェースにおいては、異常の発生をどのように運転員に知らせるかということが問題となる。些細な異常まで全て警告を出すようなインタフェースは、運転員には必ずしも好まれない。しかしながら、システムの運転に重大な影響を及ぼすような異常は、早期に、そして確実に運転員に知らされる必要がある。このようにある意味でトレードオフ的な状況に対応するためには、監視のモードを運転員の要求に応じて自由に設定できるようにする必要がある。ここでは危機レベルの低い異常に対する警報の発生を抑制する Latent Mode と、全ての警告を運転員側に伝えるような Alert Mode の二つのモードの選択が可能となるようにすべきであると考えている。

#### 2) 表示インタフェースの柔軟化

データベースに保持されている情報をどのように活用するかという問題が重要であるが、そのためにデータの表示のインタフェースの柔軟化が重要である。いかにデータが適切に保存されていても、表示が適切に行われなければ意味がない。ここで考えているのは、過去に起こった典型的異常時のデータの Template と現在のデータとの多面的な照合表示、時間軸を自由に変えてのトレンド表示、そして長時間にわたる変動の傾向の表示などである。このような柔軟な表示を支援するためには、当然のことながら知的なデータの検索機能が備わっている必要がある。

### 3) 感度変更監視

これは 1)にも密接に関連することであるが、正常・異常の判別をするための閾値の設定は監視において極めて重要であるが、この設定に関しても Miss Alarm と False Alarm の割合の間にはトレードオフが存在し、全ての状況において一意に決められる訳ではない。過去のデータを参照する場合にも、閾値を変化させて異常判別の感度を調節しながら検討を行うことで、異常の緩やかに生起する異常や、ランダムに発生する異常に関して重要な知見を得ることが可能になると期待される。

### 4) 分散協調監視

本研究で提唱する症候データベース(図 1)は、現時点では中央で一括して管理する形態を仮定しているが、将来的には複数のデータベースを有機的に結合した分散協調形態が様々な面から有利であると考えられる。データの一括管理を避けることで、データの消失の危険性が減少し、検索アルゴリズムを工夫することで、より効率的なデータの抽出が可

能になり、また複数のユーザがネットワーク上で協調的に監視を行うということが可能になる。これにより、システム全体の柔軟性大きく向上することが期待される。

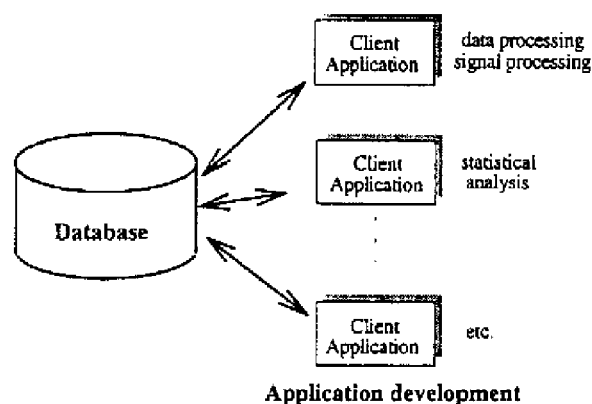


図 1 Data Management

Latent Mode は異常事象の予兆段階での検出であり、入力データを分析し解析しながら機械システムが故障であるかどうかを判断する。この段階では異常事象が進展してユーザーに通知する必要がある場合にはユーザーに事象の情報を適切に提供する。このようなモードの利点は運転員の監視する負担を大幅に減らすだけでなく有効な情報だけが得られるということである。事象の監視は、予兆段階で監視感度を変換することも可能であり運転員によって選択される。

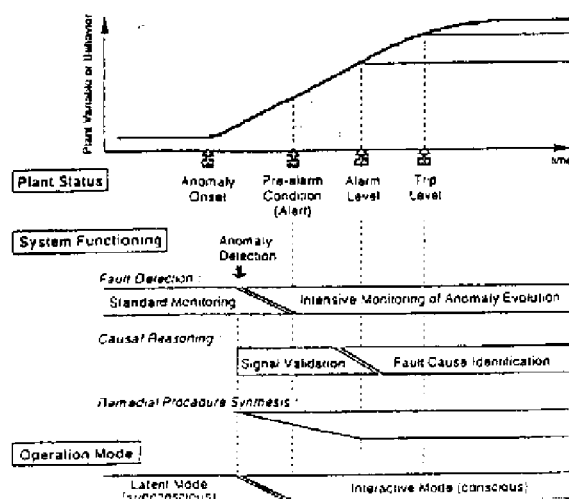


図 2 Schematic diagram of the monitoring system with a particular emphasis on the phased-mission organization [2].

図2に示されるように Operation Mode では Anomaly Onsetが生じる前に Latent Mode が動作する。これは診断システムの最も初期の段階であって通知する必要がある場合は運転員に通知する。運転員が希望する任意の時点のデータを検索と経時的照合はデータベースを通じて行う。

通常の運転時において図3の様に異常の予兆を含む信号が観測されたとする。

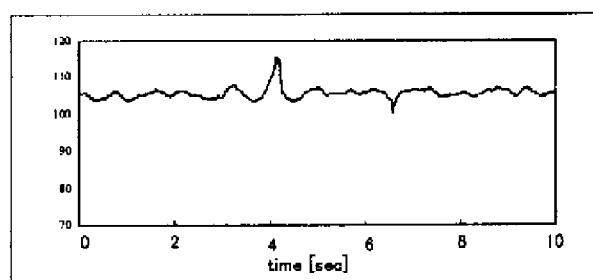


図3 異常の予兆を含む信号

このような時系列の例は通常の信号であって機械システムの故障ではない。しかし、いくつかの怪しいピークが出てきた場合、監視システムはこれを検知し運転員にこの予兆を提示することなしに分析を行なう。この事象が進展し通常と比べてかなり悪化すると判断されれば運転員に通知する。運転員はこのような事象が以前に起こったことがあるかを過去の膨大なデータから調べることが通常の運転現場ではしばしば要求される。そのため、効率的なデータベースを構築してシステムに導入することが重要となる。

### 3.2 基本設計指針

記憶容量の拡充は年々容易になり、かつコストも低下している一方、入力時に失われた情報は再入手が困難である。したがって基本的には次の方針が妥当と考えられる。

- 1) サンプル時間は、より短時間のものを優先して選択する。ただし、着目現象、計装系毎に異なった値を選択することは当然である。
- 2) 計測は連続に行なうが、全データを保存するのでは無く、データベースへの格納は基本周期毎（たとえば1日）に1回とする。1回毎に格納されるサンプル数（時間点数）はできるだけ多くする。
- 3) データベース格納に先立ってスクリーニング（統計量、異常値）試験実施を行う。この試験において特に問題がない場合には基本周期毎にデータベースへの格納を行なう。何らかの兆候が見られる場合には周期を短くする（たとえば6時間に1回）。明らかに異常と認められる時は、図4の様に通常用いる Normal DB とは別に Abnormal DB にこれを格納する。
- 4) そのためにはデータベースといってもインテリジェント・プリプロセッシング機能に必要なプリプロセッサ・エージェントを用意する。このエージェントの判断で周期を変更すると共にユーザーにこの事を通知する。ユーザー/スーパーバイザはエージェントの通報に基づき詳細な検査後に、周期に関する必要な変更を override decision する。最終的に不要と判断した場合、採取しすぎたデータは消去して記憶容量を節約する。

従ってここで構築するシステムは単なるデータベースではなく Man Machine Cooperative System であるといえる。

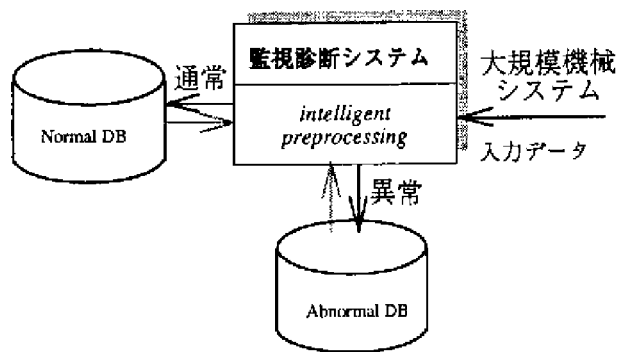


図 4 基本設計

### 3.3 信号—サンプリング時間の目的依存選択

症候が特徴的に現れる周波数帯（着目周波数帯）は機器に依存し、信号とサンプリング時間の選択は一義的には決まらないことが問題を複雑にしている。たとえば沸騰水型原子炉における中性子(LPRM)信号は：

- 1) Controller (Pressure) Monitor: 0.02-0.2 Hz ( $\Delta t = 1$  sec,  $N = 5,000$  or more)
- 2) Stability Monitor: 0.1-2.0 Hz ( $\Delta t = 0.1$  sec,  $N = 5,000$  or more)
- 3) Coolant Flow Monitor: 1.0-25 Hz ( $\Delta t = 0.01$  sec,  $N = 5,000$  or more)

となっている。

$N = 5,000$  は AR-modeling をおこなうには十分であるが、FFT を使用して解析する場合には最低でも 10 倍が必要となる。圧力や冷却材流量信号を扱う場合も、目的に関連して上記中性子信号に対応した異なったサンプリング時間選択が必要になる。

回転機器（タービンやポンプ）については基本となる周波数は知られている（例えば 1,500 rpm = 25 Hz）ので、これをヒントとする。ただしポンプインペラーの数や高調波まで考慮することが必要である。配管の場合は基本長さや材質から、大略の振動周波数は算定できる。これらについては過去の国際会議

(SMORN-3, 4, 5)報告を参照していただきたい[3][4][5]。

他に、インパクト音響検出器、Acoustic Emission Monitor などもあるがこれらは今回は除外して検討を進めた。

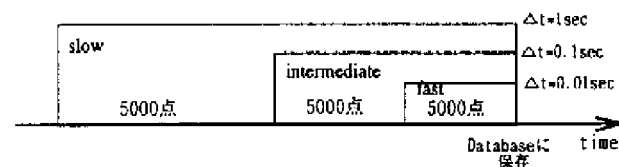


図 5 データベースの格納

監視データ量の見積りであるが、各信号についてサンプリング時間の異なる Fast、Intermediate、Slow の三種類（図 5）のデータを 1 時間に 1 回サンプリングを行なうと仮定すると、一年間に

$4\text{bytes} \times 5000 \times 3 \times 24 \times 365 = 0.5256\text{Gbytes}$  の記憶容量が必要となる。このデータベースは複数の信号を扱っており、仮に信号数を 100 とすると全体で 52.56 Gbytes となる。この量は、データの保存に関してはリムーバブルメディア等を用いる事により、現在の技術でも十分に現実可能なものである。しかしながら、データ検索に関しては、データに対して効率的な検索を可能とするインデクスづけを行なうなどの、データベース技術の応用が不可欠となる。

## 4. システムの実現（ソフトの設計）

3章で述べた指針に基づき、現在 WS 及び PC の分野で広く普及しており、かつ GUI の構築が容易な Operating System である Windows 95 / NT で、システムの開発をおこなっている。アプリケーションの開発には市販されているライブラリ (Borland OWL 及び BDE) を利用する。Remote Database を開発

するためには Microsoft ODBC Drivers もしくは Borland SQL Links が必要となる。図 6 に現在開発中のシステムにおけるソフトウェアの構成を示す。<sup>[6]</sup>

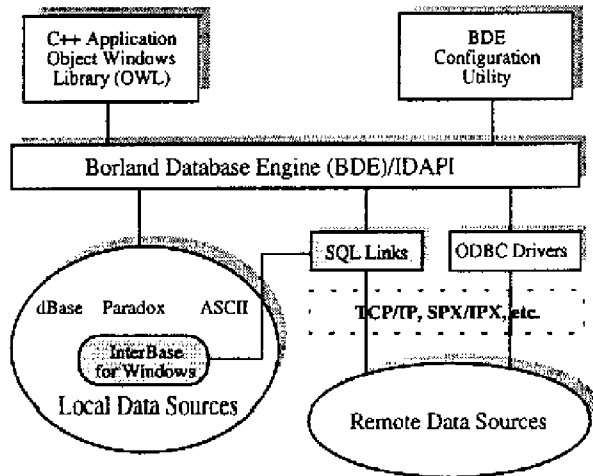


図 6. ソフト開発

## 5. 考察と結論

大規模機械システムの監視診断システムで用いる症候データベースは多変数時系列を対象とし、この時系列データを必要に応じて適当な統計量、特徴量などに変換することができるように設計されなければならない。この際サンプリング時間には着目周波数帯によって異なる複数の値が採用されるため、データベース設計にも注意が必要である。また後日の詳細な解析に答える事も要求される。この際、運転員が希望する任意の時点のデータの検索、経時的照合をするためには、運転員の要求に応じて設定可能なインタフェースを設計する必要がある。

プリプロセッサ・エージェントはデータ収集の周期変更とユーザーへの通知を行なう。このエージェントは収集したデータを解析しながら事故の兆候があった場合、必要ならばユーザーに通知する (Alert Mode) が、まだ必要のない場合には通知しない (Latent Mode) インテリジェント・プリプロセッサ

機能を提供する。なお、これらの Mode はユーザーによって選択可能となっている。

このような監視診断システムを実現するために現在データベースシステム、プログラミング、数学的な手法などを用いて開発を行っている。

## 参考文献

- 1) K.M.Vicente, C.M.Burns, R.J.Mumaw and E.M.Roth: How do operators monitor a nuclear plant?: A field study, Proceedings NPIC & HMIT '96 (The 1996 American Nuclear Society International Topical Meeting Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies) Vol.2, pp.1127-1134, (1997).
- 2) M. Kitamura, H. Furukawa and R. Kozma: Guiding Rules for Development of Intelligent Monitoring System of Nuclear Power Plants, SMORN VII Volume 2, 493/501 (1996)
- 3) M. Kitamura, T. Washio, K. Kotajima, and K. Sugiyama : Development of Methods for Analyzing Time-Varying Characteristics of Power Reactor Noise, Progress in Nuclear Energy -Reactor Noise SMORN IV Vol.15, pp. 57-56 (1984)
- 4) B.R. Upadhyaya, J. March-Leuba, D.N. Fry and M. Kitamura : Application of Noise Analysis Methods to Monitor Stability of Boiling Water Reactors, Progress in Nuclear Energy -Reactor Noise SMORN III Vol.9, pp. 619-630 (1982)
- 5) M. Decreton and M. Bouneder : Early Detection of Localized Onset of Boiling, Progress in Nuclear Energy -Reactor Noise SMORN V Vol.21, pp. 343-349 (1988)
- 6) Borland International : Borland C++ Programmer's Guide version 5 (1996)