

信号処理に基づくエキスパートシステム
構築ツール
— 信号解釈システム —

犬 島 浩
三菱電機 産業システム研究所

信号解釈システムは、対象システムの診断方式や制御方式を開発するためのツールであり、対象システムで観察される信号を解析処理するとともに、知識処理技術を用いてその特徴を明らかにすることが可能である。本システムは大きく2ブロックで構成され、1つは信号処理部で、FORTRAN 言語による種々の処理プログラムを記述できる構造にしてある。もう1つは推論部で、知識をIF~THENで取り扱えるC言語を用いたカーネルが用意されており、信号処理に関する知識や対象に関する知識を自由に記述できる。

THE DEVELOPING TOOL FOR EXPERT SYSTEM
WITH USING SIGNAL PROCESSING TECHINCS
— SIGNAL INTERPRETATION SYSTEM —

Hiroshi Inujima

Mitsubishi Electric Corporation
Industrial Electronics & Systems Development Laboratory

1-1, Tsukaguchi-Honmachi 8-Chome, Amagasaki, Hyogo, 661, JAPAN

Signal interpretation system (SIPS) provides for failure detection methods and control methods. SIPS is constructed with the signal processing block, the inference block, the signal data block and the presentation block for graphs. The SIPS solves various problems on diagnoses and control methods with supplement knowledge and signal processing programs. In this paper, We show two examples of the SIPS efficiency. One example is recognition of surge noise on random signals. In this case, the SIPS recognised and eliminated surge noises. The another example is recognition of spectral peak. In this case, the SIPS calculated power spectral densities with suitable signal processing technique that is store knowledge data base, and recognised spectral peak precisely. We can obtain efficient signal interpretation with using signal processing engineering and knowledge engineering by SIPS.

1. まえがき

信号処理技術は種々の分野における診断、制御、解釈などの手法として重要な役割を果たしている。それらのなかで、プラント等における設備診断は重要な技術であり、特に故障にいたる前に検出する劣化検出法には大きな要請がある。劣化現象は検出器で検出された信号のノイズ成分にその情報が重畳されている場合が多い。例えば、高速増殖炉の冷却材流量ゆらぎ、温度ゆらぎを観測することで炉心の冷却材流路の局所閉塞を推定したり⁽¹⁾、集積回路の出力端で観測されるノイズ成分から劣化徴候を判断する技術等がある^{(2)・(3)}。しかしながら、ノイズを計測し、解釈する技術は特殊なエキスパートが必要なことから、劣化診断に有効な技術であることが20年以上まえから知られているにもかかわらず普及しない原因となっている。

このような背景から、信号を解釈するためのツールである信号解釈システム(SIPS)を開発した。これは、対象システムで観察される信号を信号解析処理するとともに、知識処理技術を用いてその特徴を明らかにすることが可能であり、信号の総合的な解釈を行うシステムである。SIPSの特徴は知識処理環境(C言語)と信号処理環境(FORTRAN 言語)が通信できるように構成したところにある。図1にSIPSの概念を示す。

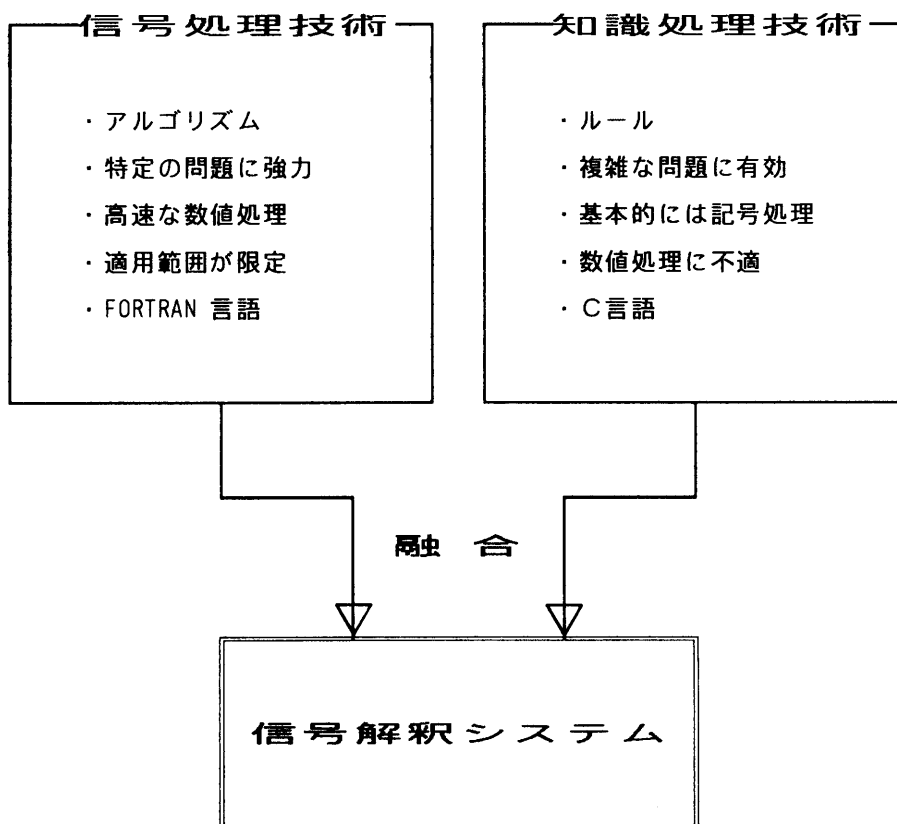


図1 信号解釈システムの概念

信号解釈システムの目的は、多次元アナログデータを意味のある単位に分割し、各セグメントを記号的に特徴付け、セグメント間に存在する相関関係を利用して、全体として整合性のある解釈を得ることである。データの欠落やノイズがある場合その解析自体が問題となって、解釈に大きな影響を与える。サージ雑音は人間には容易に識別できるが、信号処理アルゴリズムだけで認識することは困難である。スペクトルピークの認識も同様である。ここでは、例題として信号の中に含まれるサージ雑音の認識と信号からスペクトル求めスペクトルピーク認識問題を取り扱う。

2. 信号解釈システムの適用分野

信号解釈システムの使用目的は大きく次の3つが考えられる。対象は特に限定されず、あらゆる分野での適用が期待できる。

例題対象としてプラントについて考察すると、以下の問題が想定される。

(1) システム同定

プラントの制御や診断を設計する前には、システムの動特性を正確に把握すること（システム同定）が欠かせない。プラントのような大規模多変数系のシステム同定は、変数間の寄与率からの関係の絞り込みや伝達関数推定など、かなり複雑な手順を繰り返して行われる。したがってシステム同定の手順の知識を組み込むと、システム同定支援ツールとして使用できる。システム同定作業の効率化が期待される。

(2) 異常診断

計測データから異常を診断する方法は信号処理理論に基づくものや、かなり現場のノウハウに近いものまでさまざまである。一般的な異常診断は二つのステップに分類される。まず、観測された信号波形に種々の信号処理を施し、信号の特徴を抽出する。そして抽出された特徴を分類、異常の原因を判断する。使用する信号処理アルゴリズムや判断の知識を組み込むと異常診断システムとして使用できる。信号解釈システムを適用する事で、多様な異常に対処することができ、また必要に応じて知識を追加してよりきめ細かなシステムを実現できることが期待される。

(3) 制御

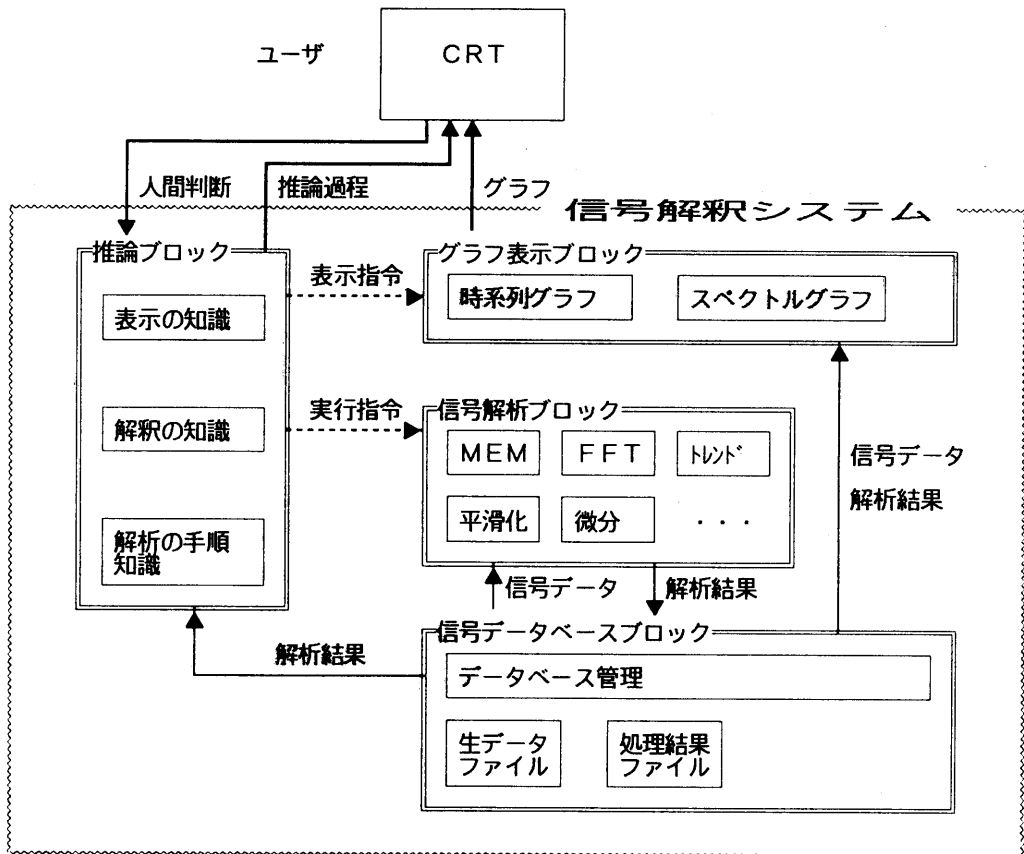
現在の制御系は基本的にシングルループで線形制御理論を用い、一定の制御則で実行されている。しかし、実際のプラントは非線形性や外乱などで、一つの制御則では対処しきれない場合もある。こうした状況において、プラント状態を分析し、状態に応じた制御則を選択するような知識を組み込むと、ハイブリット型のコントローラとして使用できる。きめ細かい制御系による運転効率の向上や、異常時への対処などこれからのプラント制御の重要な課題を解決することが期待される。

3. 信号解釈システムの構成

信号解釈システムは図 2 に示した概略図のような推論ブロック、信号処理ブロック、信号データベースブロック、グラフ表示ブロックの四つのブロックから構成されている。

(1) 推論ブロック

解析手順を決定したり、全体の制御をするブロックである。三菱電機中央研究所で開発した知識ベース型制御システム基本ソフトウェア ERIC^{(4)・(5)} 上に構築する。知識の追加、修正、削除は任意に可能である。



MEM：最大エントロピー法

FFT：高速フーリエ変換

図2 信号解釈システムの構成図

(2)信号処理ブロック

スペクトル計算、ピーク検出、雑音除去などの信号処理アルゴリズムを実際に行うブロックである。信号処理ブロックは各種のアルゴリズムを実現するプログラムの集合である。言語はFORTRAN で記述している。

信号処理プログラムの追加、修正、削除は任意に可能である。

(3)信号データベースブロック

信号の生データや信号処理された解析結果のデータを蓄積・管理するブロックである。

(4)グラフ表示ブロック

生データや解析結果をディスプレイ上でグラフ表示するブロックである。グラフの拡大、重ね合わせなどの各種の表示操作を行う。

4. 推論ブロックの構成

信号解釈システムの中核を成している推論ブロックには、現在の人工知能技術で最も実用的といわれるルールベースシステムを採用している。ルールベースシステムでは信号処理の手順の知識や、結果の解釈をするための知識がif～thenのルール形式で蓄えられている。そして、実際に信号が与えられると推論が開始される。この様にルールの形で蓄積することには、FORTRAN 言語などの従来の手続き型のプログラムと比較して以下の長所がある。

- ・知識は推論実行中に自動的に組み合わされて使用されるため、種々の状況に対応できる柔軟性を有している。
- ・ルールはモジュール性が高いので知識を少しずつ増やすことが容易である。
- ・知識を変更することにより、種々の対象を取り扱うことが可能である。
- ・ルールは細分化されるため、専門家の知識を容易にシステム化できる。

このような特徴を持つルールベースシステムは、色々な分野で計測されている信号データを解析する上で、強力なツールであると考えられる。

5. 例題

信号解釈システムの有効性を確認するため、2つの例題について知識を組み込んでみた。1つの例題は、アナログデータ計測時にしばしば観測されるサージの認識および除去である。もう1つの例題はスペクトルピークの認識をとりあげた。信号解釈システムをそれぞれの問題に応用するには次のような手順で行う。

〈5.1〉 サージ除去

最初の例題はサージ雑音認識および除去である。図 3に示すように、サージ雑音は局所的に異常に大きな値をとる雑音で、その形状から「ひげ雑音」とも呼ばれている。計測器のサンプルミスやA/Dコンバータの異常による飽和などが原因として考えらる。このサージ雑音は見かけ上局所的にパワーが増大するので、パワースペクトルなどに悪影響をおよぼす。したがって前後のデータから補間するかサージが含まれている信号を避けるかして、前もって除去しなければならない。

しかしながら不規則信号にサージ雑音が載っている場合には、局所的に大きい値がサージ雑音なのか単に信号がたまたま大きな値を示したのかの判断が難しく、サージを除去する一般的なアルゴリズムを与えるのは困難である。そこでサージ雑音でありそうな場合をいくつか与えて、局所的に大きな値がそのどれかに該当するならばサージ雑音とみなして補間処理を行う。このようにいろいろな場合によって実行する処理を変える複雑な信号処理は、知識処理を用いなければほとんど不可能と考えられる。

サージ除去の知識を簡単に説明すると、サージにはいろいろなパターンがあり、このグラフ上には2つのパターンが表示されている。一つは55点目あたりにある単純な鈍いピークを持つもので、もう一つのパターンは100点目付近にあるようなピークの横に小さなサイドピークである。この知識ベースではこの二つのパターンのサージ雑音に対応している。

サージ雑音除去は3つの手順から成り立っている。サージ雑音の候補の検出、サージ雑音の範囲の決定、サージ雑音の補間である。これを下向きのサージと上向きのサージのそれぞれについて候補がなく

なるまで繰り返し実行する。まず下向きのサージの候補を検索し、100点目付近に候補を発見している。この候補の近傍の信号の形をルールを使って調べ、このサージの範囲は98点目から104点目であることを決定する。そしてこの区間の始めと終わりのデータから補間する。さらに下向きのサージの候補を探索し、もう下向きのサージが存在しないことを確認する。続いて上向きにサージの探索を実施する。同様に50点目と80点目のあたりにあるサージを検出し、区間の始めと終わりのデータから補間する。この結果をグラフを図4に表示する。

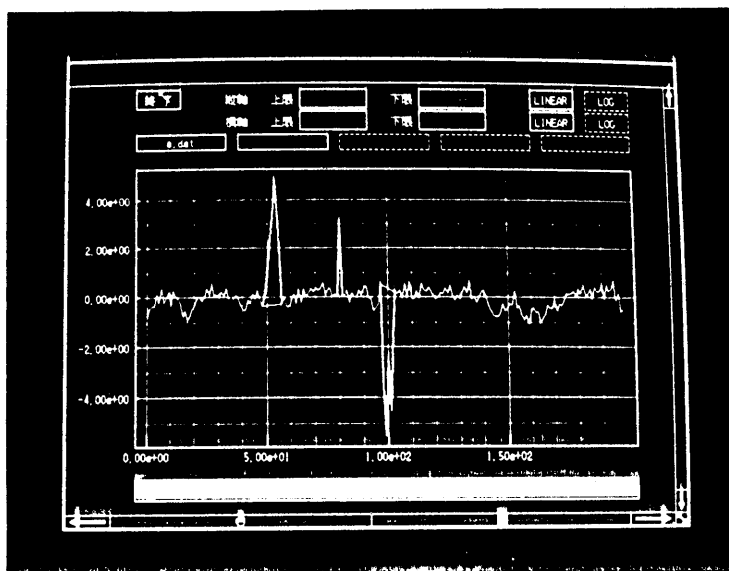


図3 サージ雑音例

```

error: cmd file [setup.cmd] not found
ERIC> setup.cmd
error: cmd file [setup.cmd] not found
ERIC>cd data
ERIC> setup.cmd
ERIC> resetsurge.cmd
ERIC> dispsurge.cmd
error: cmd file [dispsurge.cmd] not found
ERIC> dispsurge1.cmd
ERIC>exec surge
下向きのサージを除去します。
サージを 100 点付近に検出しました。
サージの範囲は 98点目～104点目です。
除去します。
下向きのサージは検出されません。
上向きのサージを除去します。
サージを 53 点付近に検出しました。
サージの範囲は 49点目～57点目です。
除去します。
サージを 81 点付近に検出しました。
サージの範囲は 78点目～82点目です。
除去します。
上向きのサージは検出されません。
ERIC>

```

図4 サージ認識例

このようにサージの始めと終わりを正確に判定し、その間を補間していることが分かる。現在はこれら二つのパターンのサージを除去するが、この他のパターンのサージでも知識をルール化して知識ベースに入れておけば除去することが可能である。もしサージ除去のプログラムをFORTRANなどの言語で作ると、非常に複雑なプログラムになり新しいパターンのサージの追加はかなり困難と考えられる。ルール例を図5に示す。

サージの範囲の終点を決定するルール

日本語で表現

もし ある谷となるサンプルデータがサージピークより後ろにあり、
 そのサンプルとサージピークを結んだ線の傾斜が大きく、
 平均値からあまり離れていない。
 ならば そのサンプルデータはサージ範囲の終点である。

ルール言語で表現

```
rule 1
  if {  $PB.pos > pt.pos;
        $PB.f  > thres.slope;
        $PB.x  < thres.upper;        }

  then {  set( pe.pos, $PB.pos);       }
```

\$PB : ある谷となるサンプル（\$は「ある」を意味）
 \$PB.pos : サンプルの位置
 \$PB.f : サージピークと結んだ線の傾斜
 \$PB.x : サンプルの値
 thres.slope : 傾斜の基準
 thres.upper : 平均値からのずれの基準
 pe.pos : サージ範囲の終点の位置

図5 ルール例1

<5.2> スペクトルピークの認識

次の例題はスペクトルピークの認識である。不規則信号が与えられたときにパワースペクトルを計算して、どの周波数でピークを持つか、そのピークはどの位鋭いのかを調べることは不規則信号処理の基本の一つである。パワースペクトルを求める方法にはFFTを使ったピリオドグラム法や情報理論に基づいた最大エントロピー法などいくつかの方法があり、これらの方法にはそれぞれ誤差や分解能などに

特性がある。ピークの鋭さなどを判断するためにこの知識ベースではいろいろな方法でスペクトルを求め、これらのスペクトルを比較検討してピークを認識する。

実際にはサージ雑音除去やトレンド除去などの前処理を最初に行うが、今回は省略する。次にスペクトルピークの検出を実施する。ここではスペクトルをピリオドグラム法により求め微分などをしてスペクトルピークの候補を検出する。最後にスペクトルピークの確定を実施する。前のフェーズで求めたそれぞれのピーク候補に対して幅や鋭さなどの情報を求める。結果として4つのスペクトルピークを認識できている(図6参照)。求めたスペクトルをグラフに図7に表示する。ルール例を図8に示す。

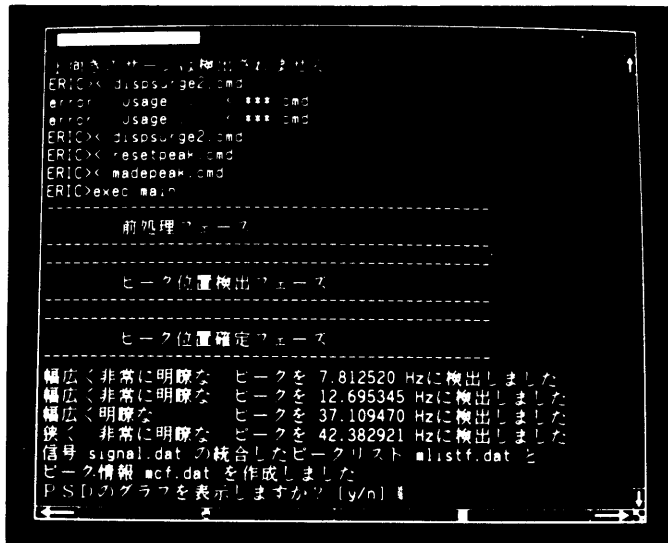


図6 スペクトルピークの認識例

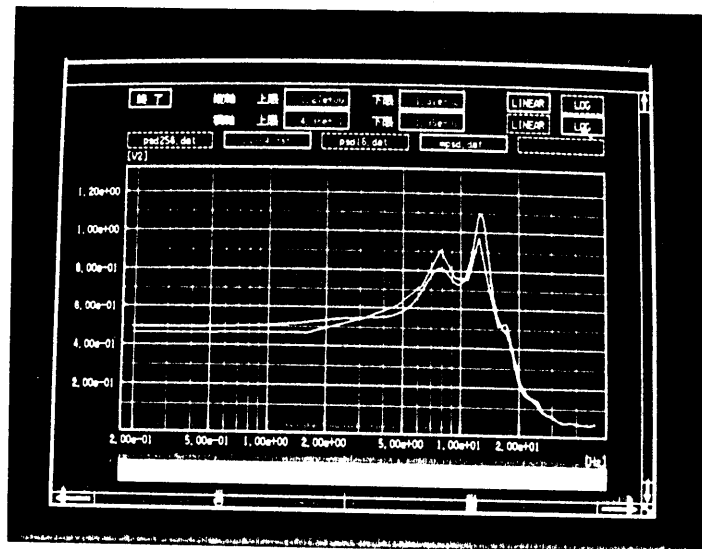


図7 スペクトルの形状例

ピリオドグラム法によりスペクトルを求めるルール

日本語で表現

もし 時系列信号が与えられており、
パワースペクトルが作成されていない

ならば パラメータとして 256を与えて、
ピリオドグラム法によりパワースペクトルを計算し、
データベースに登録し、
パワースペクトルを作成済みとする。

ルール言語で表現

```
rule 2
if { psignal.status == "made" ;
    psd256.status != "made" ; }

then { set( perigram.pl, 256 ) ;
       set( perigram.status, "do" ) ;
       r_fork( exec_process ) ;
       r_fork( exec_db_regist ) ;
       set( psd256.status, "made" ) ; }
```

psignal.status : 時系列信号作成状態のフラグ
psd256.status : パワースペクトル作成状態のフラグ
perigram.pl : ピリオドグラム法実行パラメータ
perigram.status : ピリオドグラム法実行フラグ
r_fork(exec_process) : プログラムを実行するルールの起動
r_fork(exec_db_regist) : データベース登録ルールの起動

図8 ルール例2

6. むすび

信号解釈システムは信号を与えれば何でもできるわけではない。対象から何をしたいのか目標をしっかりと定めるときに効果を発揮するツールにしか過ぎない。今後はサージなどの雑音成分の認識や、スペクトルピークの認識などの信号処理上の重要な知識処理をベースにしてプラント設備の劣化診断などに適用する予定である。

参考文献

- (1) 萩野, 犬島, 秋月, ” 統計的手法を用いた異常診断の適用例 ―ゆらぎを用いた異常診断―”, システムと制御 Vol.24, No.11, pp719~725, 1980
- (2) 犬島, 吉島 「電子回路の劣化診断」 電気学会論文誌C Vol.109-C, No.7, (1989-7)
- (3) 犬島, 吉島 「工場設備におけるICの劣化診断」 電気学会論文誌C Vol.109-C, no.12, (1989-12)
- (4) 石岡, 竹垣, 大井, 「プラント制御用エキスパートシェル ERIC」, オートメーション Vol.33, No.6, (1988-6)
- (5) 竹垣, 大井, 石岡, 「知識ベース型制御システム構築ツール・ERIC」, 第32回システムと制御研究発表講演会, (1988-5)