

## バーチャルマイクロスコープの制御手法の開発

助 村 俊 一<sup>†</sup> 加 藤 昭 史<sup>†</sup> 松 原 聖<sup>†</sup>  
小 池 秀 耀<sup>†</sup> 佐 久 間 俊 広<sup>††</sup>  
平 原 幸 男<sup>††</sup> 関 口 智 嗣<sup>†††</sup>

コンピュータとグラフィックスを駆使し、あたかも顕微鏡で覗いているように、分子の世界で起こる現象を観察するためのシステム、バーチャルマイクロスコープ・システムを開発した。バーチャルマイクロスコープを実現するためには高速な計算、グラフィック表示と共に、計算途中で自由に結果を表示し、かつ、計算条件を変更する制御方法を開発する必要がある。本論文では、バーチャルマイクロスコープの制御手法に関し、実現方法と実装方法について述べる。

### Control Method for Virtual Microscope

SHUNICHI SUKEMURA,<sup>†</sup> AKIFUMI KATO,<sup>†</sup> KIYOSHI MATSUBARA,<sup>†</sup>  
HIDEAKI KOIKE,<sup>†</sup> TOSHIHIRO SAKUMA,<sup>††</sup> YUKIO HIRAHARA<sup>††</sup>  
and SATOSHI SEKIGUCHI<sup>†††</sup>

Virtual Microscope System to observe a phenomenon in the world of molecular was designed and developed. The function to display the phenomena by real time with changing some parameters is required for Virtual Microscope System. High-speed computing and scientific visualization are indispensable to realize such a system. It is also very important to control the computing and visualization efficiency. This paper describes the function and the implementation method to control Virtual Microscope System.

#### 1. 序

化学反応のメカニズムを分子・原子のレベルで理解することは重要なことであるが、現状では分子・原子の挙動を実験で直接的に観察することは困難である。しかしながら、コンピュータ・シミュレーションの結果を可視化することにより、“仮想的”に分子・原子の挙動を観察することは可能である。高田<sup>1)</sup>は分子軌道法と分子動力学を結合したバーチャルマイクロスコープを提案している。バーチャルマイクロスコープとは、分子軌道法と分子動力学法を結合することにより、分子・原子の化学的振舞いをコンピューター上で実現し、コンピューターグラフィックスで可視化することにより、実験では観察不可能な化学現象をあたかも“顕微鏡”で見ているように観察するシステムである<sup>2)</sup>。バーチャルマイクロスコープを実現するためには、高速計算と対話的なインターフェイスの制御の技術を開発する必要がある。前者

はハードウェアの発展と並列計算をはじめとした数値計算技術の研究開発により実現の方向にある。本稿では、バーチャルマイクロスコープの制御手法に関し、実現方法を提案し、さらに実装方法について報告する。

対話的なインターフェイスとは、時間とともに進行する化学反応をディスプレイに“リアルタイム表示”したり、計算途中で計算条件等を変更したりするインターフェイスである。ここでいう“リアルタイム表示”とは、グラフィックス表示やアニメーション表示において、ユーザーが観察したい速度で計算結果を即時に表示することである。対話的なインターフェイスの制御には、関口<sup>2)</sup>が言及しているように現象を追跡しながら、ある時点で現象をゆるやかに進行させたり、時間を戻したりするなどのロールバックが必要になったり、ある時点で解像度を変えて再計算を行ったり、観測・認識したい現象の表示方法を即座に変更したりする機能が必要になる。従来の分子軌道法や分子動力学のソフトウェアではこうした要求の実現は困難であった。なぜならば、そのようなシステムでは計算と表示が連動しておらず、計算終了時に大量のデータを転送しそれを可視化していたために、計算終了まで計算条件等を変更することができない。そこで我々は、協調分散環境におけるヘテロジニアス・コンピューティングの応用として、バーチャルマ

<sup>†</sup> (株) 富士総合研究所  
FUJI RESEARCH INSTITUTE Corporation

<sup>††</sup> (株) NEC情報システムズ  
NEC Informatec Systems, Ltd.

<sup>†††</sup> 通商産業省工業技術院電子技術総合研究所  
Electrotechnical Laboratory

イクロスコープの制御機能を実現した。具体的には計算プログラムと表示プログラムを分離しPVM<sup>3)</sup>を用いてメッセージを送り合うことにより計算プログラムと表示プログラムが協調分散的に動くようにし、計算プログラムと表示プログラムの間で必要な時に必要なデータを転送することにした。このことにより、表示プログラムから計算プログラムをインタラクティブに制御できるようになり、必要な時に必要な量の可視化のためのデータを転送できるようになった。具体的には、表示プログラムは計算プログラムに、計算プログラムの動作を制御するためのコマンドや可視化に必要なデータを指定するためのコマンドを送り、計算プログラムはコマンドに従い、動作や計算条件を変更したり可視化のためのデータを表示プログラムに送る。

以下、バーチャルイクロスコープの概要、制御に必要な機能、メッセージの分類・転送方式、PVMを用いた実装について述べる。

## 2. バーチャルイクロスコープの概要

バーチャルイクロスコープ・システムは、バーチャルイクロスコープ制御部、ヒューマンインターフェイス部、解析部より構成される(図1参照)。前節で述べた計算プログラムは解析部に相当し、表示プログラムはヒューマンインターフェイス部に相当する。

バーチャルイクロスコープ制御部は、システム全体をバーチャルイクロスコープとして機能させるための制御部であり、ヒューマンインターフェイス部と解析部との動作を整合するよう制御する機能、ヒューマンインターフェイス部と解析部とのデータの流れを制御する機能、計算結果をリアルタイムで表示するために解析部を制御する機能を有する。

ヒューマンインターフェイス部は、解析条件の入力、分子構造の作成および分子構造と計算結果(電子密度分布の時間変化等)の表示を行う。分子構造の作成や分子構造の表示を対話的に行い、必要な情報(計算結果、分子構造等)を、要求された形態(3次元グラフィックス、2次元グラフィックス、数値等)で、必要な時に表示する。

解析部は、非経験的分子動力学(分子軌道法と分子動力学の結合)を中心とした計算システムである。

現在のバーチャルイクロスコープのプラットフォームは、ヒューマンインターフェイス部はIndigo2、解析部はSX-4, Alpha Station, RS/6000、Indigo2等である。将来的には、ヒューマンインターフェイス部のプラットフォームをパソコンとすることも予定している。

## 3. バーチャルイクロスコープの制御手法

### 3.1 リアルタイム表示

化学反応をリアルタイムでディスプレイに表示するためには、表示データの作成と表示を高速に行う必要があ

る。しかし、表示データの作成(計算)や表示が高速であっても、データの作成と表示が逐次的に行なわれていてはリアルタイムな表示はできない。リアルタイムな表示をするためには表示をしている間に次に表示する画像データの計算をする必要がある。つまり、計算と表示を別々のプログラムに分け並列に実行する必要がある。

また、高速な表示のためには表示データを小さくし表示にかかる処理時間を短縮する必要もある。表示データを小さくする方法としては、データを間引くなどデータそのものを小さくする方法と、必要なデータだけにするなどデータ構造を変更する方法が考えられる。本システムでは表示データの間引と構造変更をすることとした。

### 3.2 計算の対話的制御

計算途中で対話的に計算条件を変更したり計算精度等を変更するためには解析部の計算を一時停止させ、計算条件を変更し、条件の変更後に計算を再開させる必要がある。計算を途中で一時停止させる方法はsignal等の割込みも考えられるが、解析部を割込みに対応させるためには解析部の大幅な変更が必要であり、計算手法によっては無理な場合もある。そこで、本システムでは計算を一時停止できるポイントを決め(これをブレイクポイントと呼ぶ)計算がブレイクポイントを通過する時に、必要に応じて一時停止させ計算条件等を変更するようにした。計算条件等を変更したい場合、すなわち、一時停止をさせる必要がある場合はヒューマンインターフェイス部から解析部に一時停止のためのメッセージを送ることとした。以上のことより、バーチャルイクロスコープ・システムは、計算プログラムと表示プログラム間でメッセージを交換することにより協調しながら動くシステムとなった。

### 3.3 メッセージの転送

バーチャルイクロスコープの制御のメッセージ転送に必要な機能は、

- 非同期で転送する
- メッセージの到着順番を保証する
- メッセージの有無を調べる
- メッセージの受信は別プロセスで行う

である。非同期でメッセージを転送する理由は計算途中で表示をしたり、逆に表示の最中に計算をするので、表示と計算はお互いの動作に影響されないようにする必要があるのである。

メッセージの到着順番を保証する理由は、計算した順番に計算結果が表示される必要があるため、また、ユーザーの指示した順番で計算を制御できなければならないので、メッセージを処理する順番すなわちメッセージを受け取る順番は、送った順番と同じでなければならない。

メッセージの有無を調べる機能が必要とされる理由では、メッセージの到着を待たなくてよいようにするためである。この機能が無いと、メッセージの到着を待ち、メッセージが到着してから次の処理に進まなければいけなくなる。つまり、メッセージ待ちで処理がブロック

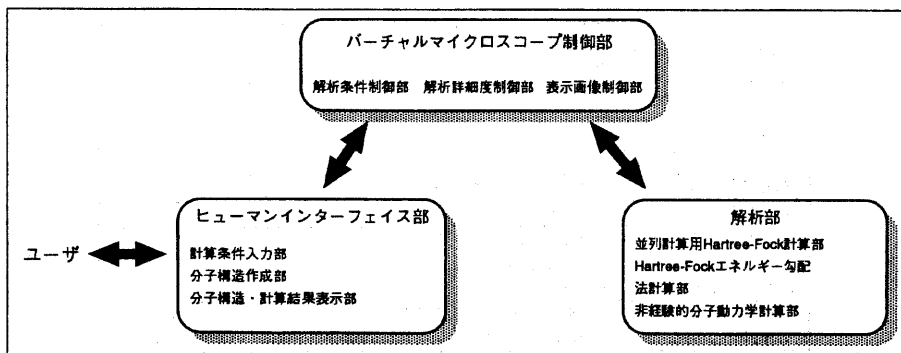


図1 バーチャルマイクロスコープの構成

されることになり、また、メッセージが必要無い時にも”メッセージ無し”というメッセージを送らなければいけなくなる。任意のタイミングでメッセージの有無を調べられれば、メッセージ待ちのために処理がブロックされることなく処理が空いた時にメッセージの有無を調べメッセージがあればそれを処理することができる。

メッセージの受信は別プロセスで行う理由は、メッセージの受信の処理を解析部の中に組込むとメッセージの到着を常に監視しなければならなくなり、計算処理の低下を招く原因になるからである。メッセージの受信を別プロセスにすることによって、解析部ではメッセージは任意のタイミングで受けるが、メッセージの処理はブレイクポイントですることとした。

### 3.4 メッセージ内容

メッセージは、動作を制御するためのメッセージと、計算結果を表示するためのメッセージの2種類に分けることができる。前者のメッセージをコマンドと呼び、後者のメッセージをデータと呼ぶ。コマンドはヒューマンインターフェイス部から解析部に送られるメッセージであり、データは解析部からヒューマンインターフェイス部に送られるメッセージである。表1にコマンドの一覧を示す。これらのコマンドによりバーチャルマイクロスコープの制御機能が全て実現できる。

バーチャルマイクロスコープ・システムでは、前述のメッセージをプロトコルとして実現した。バーチャルマイクロスコープ・システムで用いるプロトコルは、コマンド部とパラメータ部から構成される。パラメータ部の変更により容易にプロトコルを拡張できる。パラメータ部にはインデックス部を設けデータの大きさが分るようにするとともにインデックス部の値が0のデータは転送しないようにした。つまり、必要なデータだけを送るようにした。このことにより、ユーザーがデータの転送時間が推測でき、転送するデータの量を減らすことが可能になった。

解析部からヒューマンインターフェイス部に送るデータは、表示用のデータが主である。表示用のデータ以外に必要なデータは、解析部の状態をユーザーに知らせる

表2 解析部からヒューマンインターフェイス部に送られるデータ

データの種別	意味	容量
解析結果表示データ	グラフィック表示のためのデータ	10MB
リスタートデータ	リスタートのためのデータ	10MB
error データ	エラーコード	100B
state データ	状態データ	10B
HeartBeat データ	ブレイクポイント毎の時間情報	10B
LogMessage データ	解析部のログ・メッセージ	100B

データである。計算結果をリアルタイム表示したり、計算途中の状態をユーザーに対して表示するために必要なデータを分類しまとめたものを表2に示す。この中のHeartBeat データについて説明する。解析部からの表示用のデータ転送が、長い時間無い時に、計算に時間がかかっているのか何らかの原因で計算が正常に行われていないか判断がつかない場合がある。計算が正常に行われているかどうかをユーザーに表示するのは重要である。ユーザーが計算が正常に行われているかどうかを知る方法は、ユーザー側から解析部に問い合わせをする方法と解析部が定期的に状況を知らせる方法の2種類が考えられる。解析部が任意のタイミングの問い合わせに対応するのは計算途中では難しいので、定期的に知らせる方法を採用した。HeartBeat データはブレイクポイント毎に時間情報を伴って送られてくるデータである。送られてきた時間情報を表示することによって正常に計算が進んでいることを報せることができる。

大規模なデータでは10MB程度の大きさのデータを転送する際に、データを間引くことによって、表示のレスポンスを早くした。表2の中で量の多いデータは”解析結果表示データ”と”リスタートデータ”である。解析結果表示データの大きさは一番大きいもので約10MByteになる。このデータを100Baseのネットワーク(1MByte/secの転送スピード)で送った場合は、転送時間が10秒かかる。つまり計算結果が表示されてから次の計算結果が表示されるまで、10秒以上の時間がかかる計算になる。そこで、解析結果表示データを間引いて送ることにより、粗い絵を表示し、次に細い

表1 コマンド一覧

コマンド	意味
OBSERVE	計算を開始する
QUIT	解析部を終了させる
HOLD	解析部をコマンドモードにする
FORWARD	順向きに計算を実行する
BACKWARD	逆向きに計算を実行する
RDATASEND	リスタートデータの転送
RESET	リセットする
TIMESTEP	解析時間間隔を変更する
TEMPERATURE	温度を変更する
PRESSURE	圧力を変更する
PHYSICAL	解析部から HI 部へ送るグラフィックデータの種類を指定する
TIMING	解析部から HI 部へグラフィックデータを送るタイミングを設定する
UDATASEND	中間形式入力データを送信する
PDATASEND	解析部指定入力データを送信する
RDATA	リスタートデータの転送先とタイミングを指定する
REQDISPDATA	指定した範囲の指定した物理量を要求する
REQUESTDATA	リスタートデータを要求する

絵を表示することにした。データの間引は、1つおきにデータを送ることで実現した。これによりデータ量を8分の1にすることができ、先程のデータの粗い絵が1.25秒で転送できるようになった。

### 3.5 メッセージの処理

解析部はヒューマンインターフェイス部にメッセージを送信した後に直ちに計算を再開したいので、ヒューマンインターフェイス部は受信の返事はしない。同様に解析部も受信に対する返事はしない。

ヒューマンインターフェイス部は、常にメッセージの到着を監視し、メッセージが到着すると直ちにメッセージの処理をする。しかし、解析部は計算途中などではメッセージの到着と同時に処理をすることができない状況がある。つまり、解析部は「計算をする」、「メッセージを処理する」という2つの内部状態を持つ。そこで、解析部の内部状態を分類するために、モードを設けた。その様子を図2に示す。

初期モードは、起動直後のモードであり、入力データが送られてきてOBSERVEコマンドが送られてくると、計算をするモードである実行モードになる。

実行モードは計算をするモードである。実行モードは、HOLDコマンドを受信した、計算が終了した、指定されたステップを実行した、エラーが発生した、等のいずれかの原因でコマンドモードになる。計算終了でコマンドモードになることによって、計算条件を変更して再計算をしたり、逆向きに計算をして終了間際の様子を表示することができる。

コマンドモードは、ヒューマンインターフェイス部からのコマンドを処理するモードである。コマンドモードでRESETコマンドを受信すると初期モードになる。初期モードにして再度入力データを与えることで再計算ができる。QUITコマンドを受けると終了する。コマンドモードになると、FORWARDコマンドやBACKWARDコマンドを受信するまで(RESET、

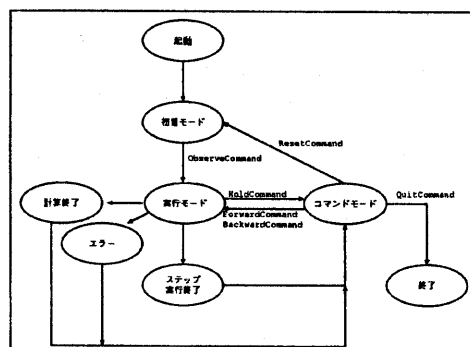


図2 解析部のモード

QUITコマンドを除く)、コマンドを実行し設定変更等を行う。FORWARDコマンドを受信すると順向きに計算を再開し、BACKWARDコマンドでは逆向きに計算を再開する。

## 4. 制御機能の実現方法

### 4.1 PVMによる実装

メッセージの転送で述べたように、制御部で用いるメッセージパッシングシステムに必要とされる機能は、非同期でメッセージを転送する、メッセージの到着順番を保証する、別プロセスでメッセージの受信を行う、自分宛のメッセージの有無を調べる、というものである。これらの機能を満して広く世間に認知されているメッセージパッシングシステムとしては、PVMとMPI<sup>4)</sup>がある。本システムでは移植性を重要視しているため、採用するメッセージパッシングシステムは「移植性」に優れ、「安定」しかつ「オーバーヘッド」が少なく、ヘテロジニアスな環境におけるメッセージ転送が確立されている必要がある。これらの理由により、PVMを

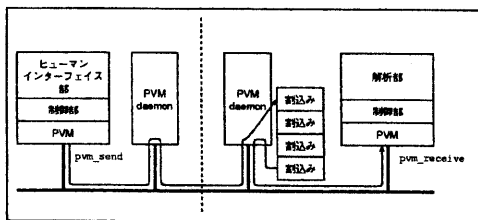


図3 PVM の利用

用いて制御部を実現した。しかしながら、実装上はメッセージのハンドリングはまとめてあるので、MPI 等の他のメッセージパッシングシステムを用いて制御部を実現することは容易である。PVM を用いた本システムの構成を図3に示す。

PVM の send 関数を用いて、コマンドやデータを送り、receive 関数を用いてコマンドやデータを受け取る。probe 関数を用いてメッセージの有無を調べる。PVM では自分が送信したデータは送信先のプログラムが受信するのではなく、相手サイトの PVM デーモンがデータを受信するので、相手の混み具合に関係なくデータを送信できる。受信側は、処理が空いた時や、ブレークポイントに達した時に自分宛のコマンドやデータが無いかを調べて、コマンドやデータがあればそれを受け取り処理をする。コマンドやデータ毎にバックして転送している。PVM daemon でのバックされたメッセージの管理は、FIFO になっているので、送信された順番で受信し処理することができる。

#### 4.2 バーチャルマイクロスコープの制御フロー

実際の制御の流れを図4に示す。

解析部はヒューマンインターフェイス部から起動され、入力データを受け取り実行モードになり、計算を開始する。時間積分の1回目のループを終了する。計算途中でエラー(解析部における計算の重大なエラー)があれば、エラーデータをヒューマンインターフェイス部に送り、コマンドモードになる。コマンドモードになると、コマンドモードになった理由とともに state データをヒューマンインターフェイス部に送る。計算途中にエラーが無ければ、時間情報とログメッセージをヒューマンインターフェイス部に送る。この時間情報とログメッセージは画面に表示される。ブレークポイントに達すると、メッセージを調べ、HOLD コマンドがあれば、コマンドモードになる。コマンドモードでコマンドの処理をし、コマンドが QUIT コマンドであれば終了し、RESET コマンドであれば初期モードになる、FORWARD コマンドであれば順方向に計算を開始し、BACKWARD コマンドであれば逆方向に計算を開始する。また、ブレークポイントでは、コマンドの有無だけを調べるのではなく、FORWARD コマンドや BACKWARD コマンドで指定された回数のループを実行したかも調べ、指定回数に達している場合もコマ

ンドモードになる。

計算途中で強制的に終了して、次回に計算の続きをしたいときは、HOLD コマンドでコマンドモードにし、REQUESTDATA コマンドでリスタートデータを得て、QUIT コマンドで解析部を終了させる。

このような制御フローにより、バッチ、トラッキングの処理も可能になった。バッチ処理は、表示用のデータとリスタート用のデータを送らない指定で実行し、計算が終了してコマンドモードになった時に、REQUESTDATA コマンドを送り、最終データを得ることで実現できる。トラッキングの処理は、表示用のデータを毎回送るように指定することで実現できる。

## 5. 結 論

協調分散環境におけるヘテロジニアス・コンピューティングの応用として、バーチャルマイクロスコープの制御機能を実現した。具体的には分子軌道法や分子力学の計算プログラムと可視化のためのプログラムが協調分散的に動くシステムを開発した。協調分散的に動かすためには2つのプログラムの間の制御が必要になりその制御の手法を開発・実装し、性能を評価した。制御のために必要なコマンドやデータを分類しまとめた。データやコマンドの転送にはPVMを用いた。この手法は、非経験的分子力学に限らず様々なシミュレーションプログラムに適用できる。

謝辞 バーチャルマイクロスコープの開発は、情報処理振興協会創作的ソフトウェア育成事業の"バーチャルマイクロスコープに関する基礎的技術開発"による研究成果である。本研究に関して、多くの助言をいただいた(株)NEC情報システムズ中田宏氏、日本電気ソフトウェア(株)石井元氏、日本電気ソフトウェア(株)村瀬匡氏、日本電気(株)岡本隠治氏、日本電気(株)野々村仁氏、日本電気(株)島田次郎技術課長、日本電気(株)高田俊和と主管研究員に深謝します。

## 参 考 文 献

- 1) 高田俊和: 化学分野におけるスーパーコンピューティング, 情報処理, Vol. 36, No. 2, pp. 143-148 (1990).
- 2) 関口智嗣: バーチャルマイクロスコープの開発, 計算工学講演会論文集, Vol. 1, p. 755 (1996).
- 3) Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V.: *PVM: Parallel Virtual Machine*, The MIT Press (1994).
- 4) Gropp, W., Lusk, E. and Skjellum, A.: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, The MIT Press (1994).

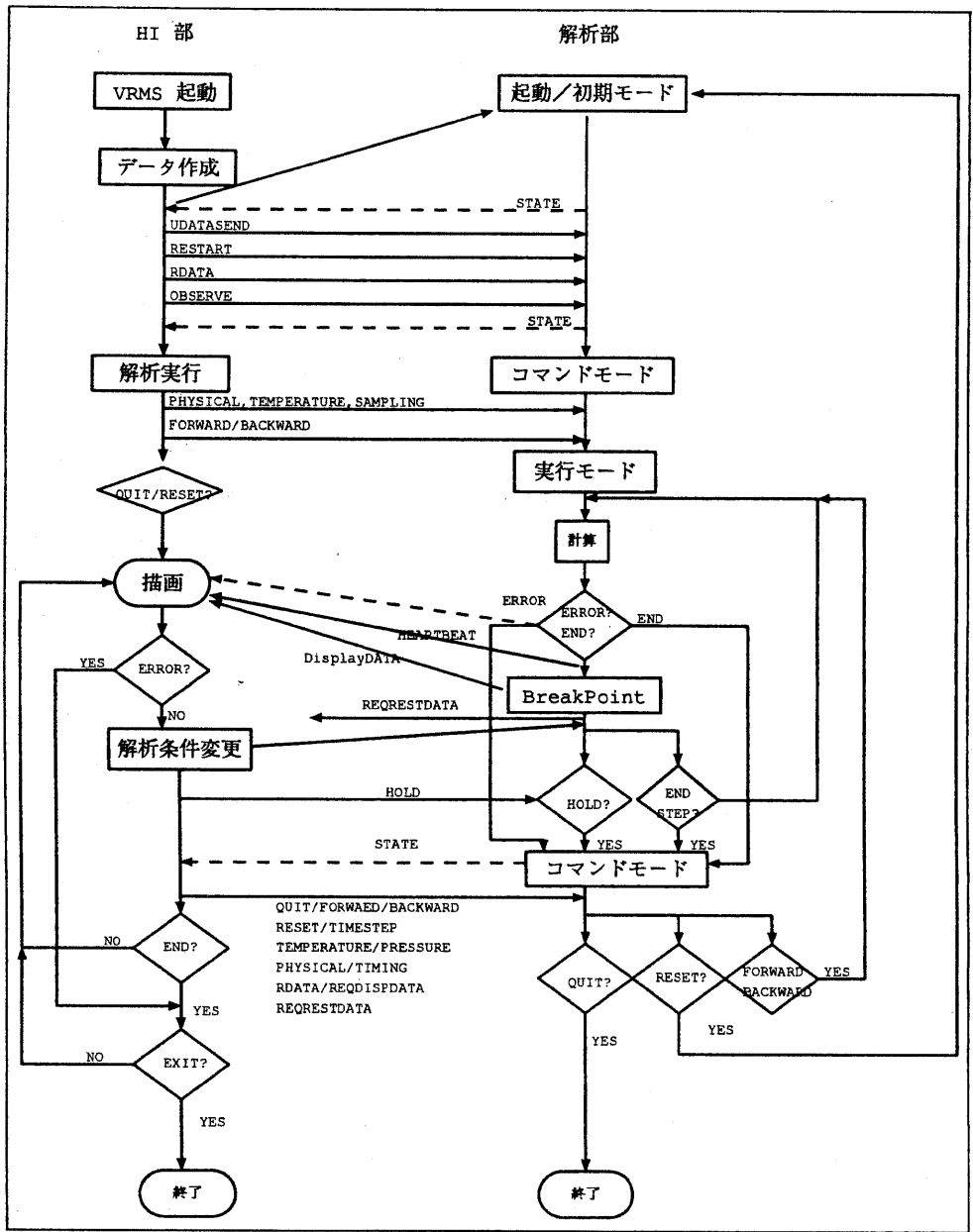


図4 パーチャルマイクロscopeの制御の流れ