

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4413563号  
(P4413563)

(45) 発行日 平成22年2月10日(2010.2.10)

(24) 登録日 平成21年11月27日(2009.11.27)

(51) Int.Cl.

F 1

<b>G05B 13/02</b>	<b>(2006.01)</b>	G05B 13/02	K
<b>G05B 13/04</b>	<b>(2006.01)</b>	G05B 13/04	
<b>G06Q 50/00</b>	<b>(2006.01)</b>	G06F 17/60	106
<b>G06F 19/00</b>	<b>(2006.01)</b>	G06F 19/00	110

請求項の数 61 外国語出願 (全 40 頁)

(21) 出願番号	特願2003-319312 (P2003-319312)
(22) 出願日	平成15年9月11日 (2003.9.11)
(65) 公開番号	特開2004-152265 (P2004-152265A)
(43) 公開日	平成16年5月27日 (2004.5.27)
審査請求日	平成18年8月29日 (2006.8.29)
(31) 優先権主張番号	10/241,350
(32) 優先日	平成14年9月11日 (2002.9.11)
(33) 優先権主張国	米国(US)

(73) 特許権者	594120847 フィッシャーローズマウント システムズ、インコーポレイテッド アメリカ合衆国 78759 テキサス オースティン リサーチ パーク ブラザ ビルディング 111 リサーチ ブル ーバード 12301
(74) 代理人	100065868 弁理士 角田 嘉宏
(74) 代理人	100106242 弁理士 古川 安航
(74) 代理人	100110951 弁理士 西谷 俊男

最終頁に続く

(54) 【発明の名称】プロセス制御システムにおける統合型モデル予測制御および最適化

## (57) 【特許請求の範囲】

## 【請求項 1】

プロセスを制御するためのプロセス制御システムであって、

前記プロセスからの複数の測定入力に基づくとともに前記プロセス制御システムの各動作サイクル中に提供される目標値のセットに基づいて前記プロセスを制御すべく設定された複数の制御出力を前記プロセス制御システムの各動作サイクル中に生成するように構成された多重入力／多重出力コントローラと、

前記プロセス制御システムの各動作サイクル中に前記多重入力／多重出力コントローラにより利用される前記目標値のセットを作成するように構成されたオプチマイザとを備え、

前記オプチマイザは、前記目標値のセットに基づく、前記プロセスの制御出力および補助出力の予測定常状態値を生成する、プロセス制御システム。

## 【請求項 2】

前記オプチマイザが二次計画法オプチマイザである請求項1記載のプロセス制御システム。

## 【請求項 3】

前記オプチマイザが線形計画法オプチマイザである請求項1記載のプロセス制御システム。

## 【請求項 4】

前記多重入力／多重出力コントローラがモデル予測コントローラである請求項3記載の

プロセス制御システム。

**【請求項 5】**

前記多重入力 / 多重出力コントローラがモデル予測コントローラである請求項 1 記載のプロセス制御システム。

**【請求項 6】**

前記多重入力 / 多重出力コントローラが、出力数と同一の入力数を有する正方制御行列から作成される請求項 1 記載のプロセス制御システム。

**【請求項 7】**

前記オプチマイザは、前記プロセスへの少なくとも一つの入力に関連するとともに前記プロセスの少なくとも一つの出力に関連するコストまたはプロフィットを定義する目的関数を有する請求項 1 記載のプロセス制御システム。 10

**【請求項 8】**

前記オプチマイザは目的関数を有する線形計画法オプチマイザまたは二次計画法オプチマイザであり、前記オプチマイザは、制御変数のセットを所定の設定ポイントに維持し、補助変数のセットを補助変数制約リミットの所定のセット内に維持し、操作変数のセットを操作変数制約リミットの所定のセット内に維持しながら、前記目的関数を最大化または最小化するように構成され、解が存在しない場合には、少なくとも一つの制御変数を所定の設定ポイントの範囲内に維持することが可能であるように構成されている請求項 1 記載のプロセス制御システム。

**【請求項 9】**

前記オプチマイザは目的関数を有する線形計画法オプチマイザまたは二次計画法オプチマイザであり、前記オプチマイザは、制御変数のセットを所定の設定ポイントリミット内に維持し、補助変数のセットを所定の補助変数リミットのセット内に維持し、操作変数のセットを所定の操作変数リミットのセット内に維持しながら、前記目的関数を最大化または最小化するように構成され、解が存在しない場合には、前記補助変数リミットのうちの少なくとも一つを緩和または無視することが可能であるように構成されている請求項 1 記載のプロセス制御システム。 20

**【請求項 10】**

前記オプチマイザは前記補助変数のセットに対する優先順位のセットを格納するように構成されており、緩和または無視される前記補助変数リミットのうちの少なくとも一つを決定するために前記オプチマイザが前記優先順位を利用する、請求項 9 記載のプロセス制御システム。 30

**【請求項 11】**

プロセスを制御するためのプロセス制御システムであって、

前記プロセスの第一の数(N)の制御変数および補助変数の予測値に基づくとともに前記プロセスの第二の数(M)の操作変数の現在値に基づいて前記プロセスの最適化動作ポイントを決定するように構成されたオプチマイザと、

前記プロセスの前記制御変数および前記補助変数の前記第一の数(N)の予測値の所定のサブセットに基づいて前記プロセスの前記操作変数を制御すべく前記第二の数(M)の操作制御信号を生成するように構成された多重入力 / 多重出力コントローラとを備えており、 40

前記所定のサブセットが前記プロセスの前記第一の数(N)の制御変数および補助変数より少ない数を有するプロセス制御システム。

**【請求項 12】**

前記所定のサブセットが前記第二の数(M)の操作変数より少ないかまたは等しい数を有している請求項 11 記載のプロセス制御システム。

**【請求項 13】**

前記多重入力 / 多重出力コントローラが正方制御行列を有するモデル予測コントローラである請求項 11 記載のプロセス制御システム。

**【請求項 14】**

10

20

30

40

50

前記正方制御行列は、前記プロセスの制御変数および補助変数を入力としてかつ前記プロセスの操作変数を出力として有し、操作変数の数と等しい数の制御変数および補助変数を有する請求項 1 3記載のプロセス制御システム。

**【請求項 1 5】**

前記正方制御行列は、前記プロセスの制御変数および補助変数を入力としてかつ前記プロセスの操作変数を出力として有し、操作変数の数より小さい数の制御変数および補助変数を有し、操作変数出力および制御変数入力の双方として、前記操作変数のうちの少なくとも一つの変数を有する請求項 1 3記載のプロセス制御システム。

**【請求項 1 6】**

操作変数出力と制御変数入力との双方として割り当てられた前記操作変数のうちの前記少なくとも一つの変数の数は、操作変数の総数と前記制御行列に制御変数入力として利用される制御変数および補助変数の総数との間の差に等しい、請求項 1 5記載のプロセス制御システム。 10

**【請求項 1 7】**

操作変数出力と制御変数入力の双方として利用される前記操作変数のうちの少なくとも一つの変数はユニットゲインを有するステップ応答を含む請求項 1 5記載のプロセス制御システム。

**【請求項 1 8】**

予測ホライゾン上の制御平方誤差を最小化し、制御ホライゾン上の操作変数の移動を最小化するために前記モデル予測コントローラが生成される請求項 1 3記載のプロセス制御システム。 20

**【請求項 1 9】**

前記制御ホライゾン上での操作変数の移動の合計を該操作変数の最適目標変化に等しくさせるために前記モデル予測コントローラが生成される請求項 1 8記載のプロセス制御システム。

**【請求項 2 0】**

前記予測ホライゾン上の平方誤差を最小化し、操作変数の移動を最小化する満足度が任意に調整されうる請求項 1 8記載のプロセス制御システム。

**【請求項 2 1】**

前記操作変数の最適目標変化を達成する満足度が任意に調整されうる請求項 1 8記載のプロセス制御システム。 30

**【請求項 2 2】**

前記オプチマイザは線形計画法オプチマイザまたは二次計画法オプチマイザである請求項 1 1記載のプロセス制御システム。

**【請求項 2 3】**

前記操作制御信号を生成するために、前記オプチマイザが前記第一の数( $N$ )の制御変数および補助変数の前記所定のサブセットに対して目標値のセットを生成し、前記多重入力／多重出力コントローラが前記第一の数( $N$ )の制御変数および補助変数の前記所定のサブセットに対する前記目標値を前記第一の数( $N$ )の制御変数および補助変数の前記所定のサブセットの前記測定値と組み合わせる請求項 2 2記載のプロセス制御システム。 40

**【請求項 2 4】**

前記多重入力／多重出力コントローラが予測制御変数および予測補助変数のセットと予測操作変数のセットとを生成し、前記第一の数( $N$ )の制御変数および補助変数の前記所定のサブセットに対する前記目標値のセットを生成するために前記オプチマイザが前記予測制御変数および予測補助変数のセットと前記予測操作変数のセットとを利用する請求項 2 3記載のプロセス制御システム。

**【請求項 2 5】**

前記第二の数( $M$ )の操作変数の各々の変化に対する前記第一の数( $N$ )の制御変数および補助変数の各々の反応を定義する応答行列をさらに有しており、前記オプチマイザが、最適動作ポイントを定義する目標操作変数値のセットを生成し、前記オプチマイザが、 50

該目標操作変数値のセットから前記第一の数(N)の制御変数値および補助変数値の前記所定のサブセットに対する前記目標値を決定するために前記応答行列を利用する請求項24記載のプロセス制御システム。

#### 【請求項26】

前記制御変数の各々を所定の設定ポイントで維持するとともに前記補助変数および前記操作変数の各々を所定の制約リミット内に維持しながら、目的関数を最大化または最小化する前記目標操作変数値のセットを生成すべく前記オプチマイザが構成されている請求項25記載のプロセス制御システム。

#### 【請求項27】

前記制御変数の各々を所定の設定ポイントで維持するとともに前記補助変数および前記操作変数の各々を所定の制約リミット内に維持する解が存在しない場合に、前記オプチマイザは、前記制御変数の各々を所定の設定ポイントリミット内に維持するとともに前記補助変数および前記操作変数の各々を所定の制約リミット内で維持しながら、前記目的関数を最大化または最小化する前記目標操作変数値のセットを生成するように構成されている請求項26記載のプロセス制御システム。

#### 【請求項28】

前記制御変数の各々を所定の設定ポイントリミット内で維持するとともに前記補助変数および前記操作変数の各々を所定の制約リミット内に維持する解が存在しない場合に、前記オプチマイザは、前記制御変数の各々を所定の設定ポイントリミット内に維持するとともに前記操作変数を所定の制約リミット内で維持し、前記補助変数に関連する優先順位に基づいて前記補助変数のうちの一または複数の変数が所定の制約リミットを緩和または無視することを許容しながら、前記目的関数を最大化または最小化する前記目標操作変数値のセットを生成するように構成されている請求項27記載のプロセス制御システム。

#### 【請求項29】

プロセスが、複数の操作変数ならびに該操作変数の変化により影響を受けることができる複数の制御変数および補助変数を有しており、前記複数の操作変数が前記複数の制御変数および補助変数と数の点において異なっているプロセスを制御する方法であって、

プロセス制御の実行に利用するために、前記複数の制御変数および補助変数のサブセットを選択することと、

前記複数の制御変数および補助変数の選択されたサブセットと前記複数の操作変数とを用いて制御行列を作成することと、

前記複数の制御変数および補助変数の選択されたサブセットを入力として、前記複数の操作変数を出力として有する前記制御行列からコントローラを生成することと、

前記複数の操作変数と前記複数の制御変数および補助変数とに依存する目的関数を最小化または最大化するために、前記複数の制御変数および補助変数の選択されたサブセットに対する目標値のセットによって定義されたプロセス動作ポイントを選択することによりプロセス最適化を実行することと、

前記複数の制御変数および補助変数の選択されたサブセットに対する前記目標値と前記複数の制御変数および補助変数の選択されたサブセットの測定値とから操作変数値のセットを作成するために、前記制御行列から生成されたコントローラを利用して多重入力／多重出力制御技術を実行することと、

前記プロセスを制御すべく作成された操作変数値のセットを利用することとを有する方法。

#### 【請求項30】

プロセス最適化を実行するステップと前記多重入力／多重出力制御技術を実行するステップとは前記プロセスの連続スキャン期間中に実行される請求項29記載の方法。

#### 【請求項31】

前記複数の制御変数および補助変数の前記サブセットを選択するステップと制御行列を作成するステップとは前記プロセスのオンライン操作に先行して実行される請求項29記載の方法。

10

20

30

40

50

**【請求項 3 2】**

前記複数の制御変数および補助変数の前記サブセットを選択するステップは、前記制御行列の最小条件数に基づいて前記複数の制御変数および補助変数の前記サブセット内の前記制御変数または前記補助変数の各々を選択することを有する請求項 3 1記載の方法。

**【請求項 3 3】**

前記複数の制御変数および補助変数の前記サブセットを選択するステップは、前記操作変数のうちの一つに対する前記制御変数または前記補助変数のうちの一つのゲイン応答に基づいて、前記操作変数のうちの一つに最も応答性のある前記制御変数または前記補助変数のうちの一つを選択することを有する請求項 2 9記載の方法。

**【請求項 3 4】**

前記複数の制御変数および補助変数の前記サブセットを選択するステップは、前記操作変数のうちの一つに対する前記制御変数または前記補助変数のうちの一つの応答時間に基づいて前記操作変数のうちの一つに最も応答性のある前記制御変数または前記補助変数のうちの一つを選択することを有する請求項 2 9記載の方法。

**【請求項 3 5】**

前記複数の制御変数および補助変数の前記サブセットを選択するステップは、前記操作変数のうちの一つに対する前記制御変数または前記補助変数のうちの一つのゲイン応答と応答時間との組み合わせに基づいて前記操作変数のうちの一つに最も応答性のある前記制御変数または前記補助変数のうちの一つを選択することを有する請求項 2 9記載の方法。

**【請求項 3 6】**

多重入力 / 多重出力制御技術を実行するステップは、モデル予測制御技術を実行するステップを有する請求項 2 9記載の方法。

**【請求項 3 7】**

前記プロセス最適化を実行するステップは、線形計画法技術または二次計画法技術を実行するステップを有する請求項 3 6記載の方法。

**【請求項 3 8】**

前記複数の制御変数および補助変数の前記サブセットを選択するステップは、前記複数の操作変数の各々に対して前記制御変数および前記補助変数から单一かつ異なる一つを選択するステップを有する請求項 2 9記載の方法。

**【請求項 3 9】**

複数の操作パラメータを利用してプロセスの複数の制御パラメータおよび補助パラメータを制御するために、プロセッサ上で実行されるプロセス制御ルーチンの一部として利用されるように構成されたプロセス制御要素であって、

コンピュータ読み取り可能媒体と、

前記コンピュータ読み取り可能媒体上に格納され、各制御スキャン期間中に前記プロセスの多重入力 / 多重出力制御を実行すべく、前記プロセッサ上で実行されるように構成された機能ブロックとを備え、該機能ブロックは、

前記複数の制御パラメータおよび補助パラメータに基づいて最適化基準を定義する目的関数と、

各制御スキャン期間中に前記制御パラメータおよび前記補助パラメータに対して最適化目標値のセットを生成すべく前記目的関数を利用するオプチマイザルーチンと、

前記複数の操作パラメータに対して前記複数の制御パラメータおよび補助パラメータの所定のサブセットを関連付ける制御行列と、

前記制御行列と前記複数の制御変数および補助変数の前記サブセットの前記目標値とを用いて、各制御スキャン期間中に前記複数の操作パラメータの各々に対して制御信号を生成する多重入力 / 多重出力制御ルーチンとを有し、

前記制御信号は、前記制御パラメータおよび前記補助パラメータの前記サブセットに対する前記最適目標値に前記複数の制御パラメータおよび補助パラメータの前記サブセットを移動させるように決定されるプロセス制御要素。

**【請求項 4 0】**

10

20

30

40

50

前記最適化ルーチンは、線形計画法ルーチンまたは二次計画法ルーチンを有する請求項39記載のプロセス制御要素。

**【請求項41】**

前記多重入力／多重出力制御ルーチンは、モデル予測制御ルーチンを有する請求項39記載のプロセス制御要素。

**【請求項42】**

前記制御行列は、正方行列であり、第一の数(N)の制御パラメータおよび補助パラメータと前記第一の数(N)の操作パラメータとを利用してあり、前記目的関数は第二の数(M)の制御パラメータおよび補助パラメータに基づいて最適化基準を定義し、前記第二の数(M)が前記第一の数(N)と異なる請求項41記載のプロセス制御要素。 10

**【請求項43】**

前記目的関数は、プロセス動作中にユーザにより選択可能である請求項39記載のプロセス制御要素。

**【請求項44】**

前記オプチマイザルーチンによって利用されるために、前記目的関数として選択可能な複数の異なる有望な目的関数をさらに備える請求項39記載のプロセス制御要素。

**【請求項45】**

前記機能ブロックは、制御パラメータ設定ポイントのセットと、補助パラメータリミットと前記操作パラメータリミットとよりなるセットとを格納する格納装置を備え、前記オプチマイザルーチンは、前記制御パラメータを前記制御パラメータ設定ポイントに位置付け、前記補助パラメータおよび前記操作パラメータを前記補助パラメータリミットおよび前記操作パラメータリミット内に位置付け、前記目的関数を最小化または最大化させるよう前記操作パラメータに対する前記最適目標値のセットを決定するように構成されている請求項39記載のプロセス制御要素。 20

**【請求項46】**

前記格納装置は、制御パラメータ設定ポイントリミットのセットも格納しており、前記制御パラメータを前記制御パラメータ設定ポイントに維持するとともに、前記補助パラメータおよび前記操作パラメータを前記補助パラメータリミットおよび前記操作パラメータリミット内に維持する解が存在しない場合には、前記オプチマイザルーチンは、前記制御パラメータの各々を前記制御パラメータ設定ポイントリミット内に維持するとともに前記補助パラメータおよび前記操作パラメータの各々を前記補助パラメータリミットおよび前記操作パラメータリミット内で維持しながら前記目的関数を最大化または最小化する前記操作パラメータに対する前記最適目標値のセットを生成するように構成されている請求項45記載のプロセス制御要素。 30

**【請求項47】**

前記格納装置は、前記補助パラメータの優先順位指標のセットも格納しており、前記制御パラメータの各々を前記制御パラメータ設定ポイントリミット内に維持するとともに前記補助パラメータおよび前記操作パラメータの各々を前記補助パラメータリミットおよび前記操作パラメータリミット内に維持する解が存在しない場合には、前記オプチマイザルーチンは、前記制御パラメータの各々を前記制御パラメータ設定ポイントリミット内に維持するとともに前記補助パラメータの前記優先順位指標に基づいて前記補助パラメータのうちの一または複数が前記補助パラメータリミットを緩和または無視することを許容しながら、前記目的関数を最大化または最小化する前記目標操作パラメータのセットを生成するように構成されている請求項46記載のプロセス制御要素。 40

**【請求項48】**

前記制御ルーチンは、前記制御パラメータ、前記補助パラメータ、および前記操作パラメータの各々に対して予測値を生成し、前記オプチマイザルーチンに前記制御パラメータ、前記補助パラメータ、および前記操作パラメータの各々の前記予測値を提供するように構成されており、前記オプチマイザルーチンは、前記制御パラメータおよび補助パラメータの前記サブセットの前記目標値を決定するために前記制御パラメータ、前記補助パラメ 50

ータ、および前記操作パラメータの前記予測値を利用するように構成されている請求項 3  
9記載のプロセス制御要素。

【請求項 4 9】

第二の数(M)の操作変数により制御される第一の数(N)の制御変数および補助変数を有するプロセスの制御を実行する方法であって、

前記操作変数の各々の変化に対する前記制御変数および前記補助変数の各々の応答を定義するステップ応答行列を決定することと、

操作変数の数と同一の数またはそれより小さい数の制御変数および補助変数を有する、前記制御変数および前記補助変数のサブセットを選択することと、

前記制御変数および前記補助変数の選択されたサブセットと前記操作変数とからなる前記応答行列内の前記応答から正方制御行列を作成することと、  
10

前記プロセスの各スキャン中において、

前記制御変数および補助変数の前記選択されたサブセットの各々の測定値を取得することと、

前記制御変数および補助変数の前記選択されたサブセットの各々に対して最適動作目標値を計算することと、

操作パラメータ信号のセットを生成するために、前記制御変数および前記補助変数の前記選択されたサブセットの各々に対する目標値と、前記制御変数および前記補助変数の前記選択されたサブセットの前記測定値と、前記制御行列とを用いて多重入力／多重出力制御ルーチンを実行することと、  
20

前記プロセスを制御するために前記操作パラメータ信号を用いることと  
 を有する方法。

【請求項 5 0】

前記サブセットを選択するステップと前記正方制御行列を作成するステップとは前記プロセスのオンライン動作に先立って実行される請求項 4 9記載のプロセス制御を実行する方法。

【請求項 5 1】

前記制御変数および前記補助変数の前記サブセットを選択するステップは、前記操作変数のうちの一つの変数に対する最良の応答として前記制御変数または前記補助変数のうちの一つの変数を選択することを有する請求項 4 9記載のプロセス制御を実行する方法。  
30

【請求項 5 2】

前記制御変数および補助変数のサブセットを選択するステップは、前記操作変数のうちの一つの変数に対する前記制御変数または前記補助変数のうちの一つの変数のゲイン応答に基づいて前記操作変数のうちの前記一つの変数に最も応答性のあるものとして前記制御変数または前記補助変数のうちの前記一つの変数として選択することが含まれる、請求項 4 9記載のプロセスの制御を実行する方法。

【請求項 5 3】

前記制御変数および補助変数の前記サブセットを選択するステップには、前記操作変数のうちの一つの変数に対する前記制御変数または前記補助変数のうちの一つの変数の応答時間に基づいて前記操作変数のうちの一つの変数に最も応答性のあるものとして前記制御変数または前記補助変数のうちの一つの変数を選択することを有する請求項 4 9記載のプロセス制御を実行する方法。  
40

【請求項 5 4】

前記制御変数および補助変数の前記サブセットを選択するステップは、前記操作変数のうちの一つの変数に対する前記制御変数または前記補助変数のうちの一つの変数のゲイン応答と応答時間との組み合わせに基づいて前記操作変数のうちの一つの変数に最も応答性のあるものとして前記制御変数または前記補助変数のうちの一つの変数を選択することを有する請求項 4 9記載のプロセス制御を実行する方法。

【請求項 5 5】

前記制御変数および補助変数の前記サブセットを選択するステップは、前記応答行列に  
50

基づいて前記制御変数および前記補助変数の前記サブセットを自動的に選択することを有する請求項 4 9 記載のプロセス制御を実行する方法。

【請求項 5 6】

前記制御変数および補助変数の前記サブセットを選択するステップは、前記制御変数および前記補助変数の前記選択されたサブセットを有する行列の条件数を決定することを有する請求項 4 9 記載のプロセス制御を実行する方法。

【請求項 5 7】

前記正方制御行列を作成するステップは、前記プロセスの制御変数および補助変数を入力として前記プロセスの操作変数を出力として利用することと、操作変数の数より多い数の制御変数および補助変数を選択することと、制御変数入力として前記操作変数のうちの少なくとも一つを選択することとを有する請求項 4 9 記載のプロセス制御を実行する方法。10

【請求項 5 8】

前記多重入力／多重出力制御を実行するステップは、モデル予測制御技術を実行するステップを有する請求項 4 9 記載のプロセス制御を実行する方法。

【請求項 5 9】

前記最適動作目的値を計算するステップは、線形計画法技術または二次計画法技術を実行することを有する請求項 5 8 記載のプロセス制御を実行する方法。

【請求項 6 0】

線形計画法技術を実行するステップは、前記第一の数(N)の制御変数および補助変数ならびに前記第二の数(M)の操作変数に基づいて前記プロセスの最適動作を定義する目的関数を利用することを有する請求項 5 9 記載のプロセスの制御を実行する方法。20

【請求項 6 1】

前記線形計画法技術内で利用するために前記目的関数として目的関数の所定のセットのうちの一つを選択することをさらに有する請求項 6 0 記載のプロセス制御を実行する方法。。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般的に、プロセス制御システムに関するものであり、さらに詳細には、プロセス制御システムにおける最適モデル予測コントローラの利用に関するものである。30

【背景技術】

【0002】

化学プロセス、石油プロセス、または他のプロセスにおいて利用される分散型プロセス制御システムまたは拡張縮小可能型プロセス制御システムの如きプロセス制御システムは、アナログバス、デジタルバス、またはアナログ／デジタルを組み合わせたバスを介して、相互に、少なくとも一つのホストワークステーションもしくはオペレータワークステーションに、およびフィールドデバイスに、通信可能に接続された一または複数のプロセスコントローラを備えているのが普通である。フィールドデバイスは、たとえば、バルブ、バルブポジショナ、スイッチ、およびトランスマッタ（たとえば、温度センサ、圧力センサおよび流量センサ）などであり、バルブの開閉およびプロセスパラメータの測定の如きプロセス内の機能を実行しうる。プロセスコントローラは、フィールドデバイスにより作成されるプロセス測定値および／またはこれらのフィールドデバイスに関連する他の情報を表す信号を受信し、この情報をを利用して制御ルーチンを実行し、次いで、制御信号を生成する。この制御信号は、プロセスの動作を制御すべく上述のバスを介してフィールドデバイスに送信される。フィールドデバイスおよびコントローラからの情報は、オペレータワークステーションにより実行される一または複数のアプリケーションに利用できるようになされているのが普通であり、このことにより、オペレータは、プロセスの現在の状況の閲覧、プロセス動作の修正などの如きプロセスに関連するいかなる所望の操作でさえも実行することができる。40

## 【0003】

プロセスコントローラは、流量制御ループ、温度制御ループ、圧力制御ループなどの如きプロセスに対して定義されるまたはプロセス内に実装される複数のさまざまなループの各々に対してさまざまなアルゴリズム、サブルーチン、制御ループ（これらはすべて制御ルーチンである）を実行するようにプログラムされているのが普通である。一般的にいえば、このような制御ループのそれぞれは、アナログ入力（A I）機能ブロックの如き一または複数の入力ブロックと、比例微分積分（P I D）制御機能ブロックまたはファジー論理制御機能ブロックの如き单一出力制御ブロックと、アナログ出力（A O）機能ブロックの如き单一出力ブロックとを備えている。これらの制御ループは、單一プロセス入力を制御すべく用いられるバルブ位置などの如き单一制御出力を制御ブロックが生成するので、单一入力／单一出力制御を実行することが普通である。しかしながら、特定の場合には、一を超える單一プロセス入力により被制御プロセス変数が影響を受けるので、そして事実上、各プロセス入力が複数のプロセス出力の状態に影響を与えるので、複数の独立に動作する單一入力制御ループ／單一出力制御ループを利用することは非常に効果的であるとは言えない。このような例としては、たとえば、二つの流入配管により充填され、單一の流出配管により空にされるタンクを備えるプロセスにおいて、各配管が異なるバルブにより制御され、タンクの温度、圧力、および処理量が所望の数値またはその近傍になるよう<sup>10</sup>に制御されるような場合が挙げられる。上述のように、タンクの処理量、温度、および圧力の制御は、個別の処理制御ループ、個別の温度制御ループ、および個別の圧力制御ループを用いて実行されうる。しかしながら、この場合、温度制御ループが、タンク内の温度を制御するために入力バルブの一つの設定を変えるよう作動すると、タンク内の圧力が上昇し、これにより、例えば圧力ループが圧力を下げるために排出バルブを開放しうる。次いで、この動作により、処理量制御ループが入力バルブの一つを閉鎖し、これが温度に影響を与え、そのため温度制御ループがまた別の動作しうる。この例から理解されるように、單一入力／單一出力制御ループにより、プロセス出力（この場合は処理量、温度および圧力）が、安定した状態に決して達することなく振動してしまうというような受け入れられない様態で動作することになる。

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0004】

そこで、特定の被制御プロセス変数に対する変化が複数のプロセス変数または出力に影響を与えるような状況においてプロセス制御を行うべく、モデル予測制御（M P C）または他のタイプの高度制御が用いられてきた。1970年代後半から、モデル予測制御の実施に関する多数の成功例が報告されており、M P Cはプロセス業界における高度多重変数制御の主たる形式となっている。さらに、M P Cは、分散型制御システム内において分散型制御システム層用ソフトウェアとして使用されている。米国特許番号第4,616,308および同4,349,869には、プロセス制御システム内で用いることができるM P Cコントローラが概説されている。<sup>20</sup>

## 【0005】

一般的に、M P Cとは、複数のプロセス入力の各々の変化が複数のプロセス出力の各々に与える影響を測定し、次いで、この測定された応答を用いて制御行列またはプロセスのモデルを作成する、多重入力／多重出力制御方法のことである。プロセスモデルまたは制御行列（これは、一般的にプロセスの定常状態の動作を定義する）は、数学的に反転され、次いで、多重入力／多重出力コントローラとしてまたはその中で用いられ、プロセス入力に対してもたらされる変化に基づいてプロセス出力を制御する。場合によっては、プロセスモデルは各プロセス入力のプロセス出力応答曲線（通常はステップ応答曲線）として表され、これらの曲線は、例えば各プロセス入力に提供される一連の擬似ランダムステップ変化に基づいて作成されうる。これらの応答曲線は、公知の方法でプロセスをモデル化するために用いることができる。モデル予測型制御は当該技術分野では公知であるため、本明細書内では詳細な内容の説明は省略するが、Qin, S. JoeおよびThomas A. Badgwell<sup>30</sup>

による「産業用モデル予測型制御技術の展望」(An Overview of Industrial Model Predictive Control Technology), AIChE Conference、1996年、に記載されている。

#### 【0006】

MPCは、非常に効果的で役立つ制御技術であることがわかっており、プロセス最適化と関連して利用されている。MPCを用いたプロセスを最適化するために、オプチマイザは、MPCルーチンによって決定される—または複数のプロセス入力変数を最小化または最大化することによって、当該プロセスを最適ポイントで実行することができる。この技術はコンピュータ的には可能であるが、経済的な見地からプロセスを最適化するためには、例えはプロセスの経済的な動作(プロセス流量または品質など)の改善に著しい影響を与えるようなプロセス変数を選択する必要がある。財務的または経済的な観点からプロセスを最適ポイントで動作させるには、通常、単一のプロセス変数だけでなく、多くのプロセス変数を相互に併用して制御する必要がある。10

#### 【0007】

MPCによるダイナミックな最適化を行うための解決策として、二次計画法技術、または内点方法などのようなより現代的な技術を用いた最適化が提案されている。これらの方法によって、最適解が決定され、オプチマイザは、プロセスダイナミックス、現行の制約、および最適化目標を考慮して、コントローラ出力(つまりプロセスの操作変数)の移動量(move)をコントローラに提供する。しかし、このアプローチはコンピュータに莫大な負荷をかけるものであり、現行の技術水準では实际上、実現可能ではない。20

#### 【0008】

MPCを用いる多くの場合において、プロセス内で利用できる操作変数(つまりMPCルーチンの制御出力)の数は該プロセスの制御変数の数(つまり特定の設定ポイントになるよう制御されるべきプロセス変数の数)を上回る。その結果、大抵は、最適化および制約の扱いに利用可能な自由度が多くある。理論上は、このような最適化を行うためには、プロセスの最適動作ポイントを定義する、プロセス変数、制約、リミットおよび経済的因素で表される値を算出しなければならない。多くの場合、これらのプロセス変数は制約された変数である。なぜなら、これらのプロセス変数は、プロセスの物理的性質に関連するリミットを有しており、このリミットにプロセス変数が属しあつその中に維持される必要があるからである。例えは、タンクのレベルを表すプロセス変数は、実際のタンクが物理的に到達可能な最大レベルと最小レベルに制限される。最適化関数は、制約変数または補助変数の各々に関連して、プロフィットが最大になりコストが最小になるようなレベルで作動するよう、コストおよび/またはプロフィットを計算しうる。次いで、これらの補助変数の測定値は入力としてMPCルーチンに提供され、MPCルーチンは、最適化ルーチンが定義する補助変数の作動ポイントと等しい設定ポイントを持つ制御変数として取り扱うことができる。30

#### 【0009】

プロセスへの制御入力(すなわち、制御ルーチンによりもたらされる操作変数)の数が制御されるプロセス変数(すなわち、コントローラへの入力)の数に等しいスクエア制御においてのみ、MPCは、アプリケーションに必要な性能に対して最大の成果を挙げることが多い。しかしながら、補助制約変数の数にプロセス制御変数の数をえたものは操作変数の数よりも大きいことがほとんどである。このような非スクエア構成のMPCを実施すると、受け入れられないような劣化した性能をもたらすことになる。40

#### 【0010】

操作変数の数と等しい数の制御変数と制約変数とよりなるセットを動的に選択し、この操作変数の次の移動量を決定するためにオンライン状態でまたはプロセス動作中にコントローラを生成することにより上述の問題を解決しようと他の人達は試みていたと考えられる。しかしながら、この技術はコンピュータ的観点から高価なものである。なぜならば、この技術は、行列反転を応用しており、たとえはプロセスコントローラにおいて機能ブロックとして実行されるMPCのような場合には利用できないことがあるからである。これと同様に重要なことは、生成されたコントローラの入力と出力との組み合わせによっては50

コントローラが不良状態に陥り、これにより、許容することができない動作をおこすこともある。コントローラの設定をオフライン状態で行う場合、コントローラの調整を検証し、向上することができるが、このような作業はオンライン動作に対して過度の負担になり、コントローラのレベルにおいて実施することは現実的には不可能なことである。

#### 【課題を解決するための手段】

##### 【0011】

最適化と制御との総合技術は、最適化手続をモデル予測型制御の如き高度制御手續と結合する。この高度制御手續では、プロセスプラント内における制御変数と補助制約変数とよりなる数は操作変数の数よりも大きくまたは小さくてもよい。この技術は、まず、操作変数の変化とプロセスの最適動作ポイントを決めるべくオプチマイザ内で用いられるプロセス変数（これらは制御変数および補助変数でありうる）の各々との間の相関関係を定義するステップ応答行列を決定する。プロセスの動作中にMPCルーチンへの入力として利用するために、操作変数の数に等しいまたはそれよりも小さい数を有するプロセス制御変数と補助変数とよりなるサブセットが選択される。次いで、制御変数と補助変数とからなるサブセットと操作変数とに基づくMPCコントローラのゲイン行列を利用して、MPCルーチンの動作中に用いられるMPCコントローラを作成する。その後、プロセスコントローラの各スキャン中に、オプチマイザルーチンは、制御変数および補助変数の全体のセットの各々に対して最適動作目標値を計算し、制御変数と補助変数とからなる選択されたサブセットの各々に対して決定された目標動作ポイントのセットを入力としてMPCルーチンに提供する。次いで、MPCルーチンは、制御変数と補助変数とからなるサブセットを入力として用いて、所定のMPC行列からプロセスを制御する際に用いる操作変数の変化量を決定する。かかる制御変数と補助変数とからなるサブセットには、操作変数のうちの一つの変数に対してゲインが最大であり応答時間が最速であるステップ応答を有する制御変数と補助変数とからなるサブセットが選択されるのが一般的である。この技術より、入力数と出力数とが等しいMPCルーチンを用いることが可能になり、同時に、プロセス内に存在する操作変数よりもさらに多くの制御変数および補助変数を考慮に入れた最適化ルーチンを利用することにより最適化ルーチンがプロセスの最適動作ポイントを決めることができるようになる。この技術は、プロセス内の外乱およびノイズに応答して迅速に所望の動作ポイントに移動するように動作するので、ロバスト性がよく、さらにプロセス制御システムの各スキャン中にMPC行列を再計算する必要がないので計算が少ない。

#### 【発明を実施するための最良の形態】

##### 【0012】

ここで、図1において、プロセス制御システム10はプロセスコントローラ11を備え、該プロセスコントローラ11は、データヒストリアン12と、各々がディスプレイスクリーン14を有する一または複数のホストワークステーションまたはコンピュータ13（いかなる種類のパーソナルコンピュータ、ワークステーションなどでもよい）と通信可能に接続される。コントローラ11は、入力／出力（I/O）カード26、28を介してフィールドデバイス15～22にも接続される。データヒストリアン12は、データを格納するための所望の種類のメモリおよび所望または公知のソフトウェア、ハードウェア、またはファームウェアを有する所望の種類のデータ収集ユニットであってよく、ワークステーション13のうちの一つとは分離されていても（図1に示すように）、その一部であってもよい。コントローラ11は、例えば、フィッシャーローズマウントシステムズ社により販売されるDeltaV（登録商標）コントローラで、例えば、イーサネット接続または他の所望の通信ネットワーク29を通してホストコンピュータ13とデータヒストリアン12とに通信可能に接続されてよい。通信ネットワーク29には、ローカルエリアネットワーク（LAN）、ワイドエリアネットワーク（WAN）、電気通信ネットワークなどの形態があり、有線または無線技術を用いて実現される。コントローラ11は、例えば標準的な4-20ミリアンペア装置と関連する、および／またはFOUNDATION（登録商標）フィールドバスプロトコル（フィールドバス）、HART（登録商標）プロトコルなどのスマート通信プロトコルと関連する、所望のハードウェアおよびソフトウェアを用いてフィールドデバイス15～22と通信

10

20

30

40

50

可能に接続される。

**【 0 0 1 3 】**

フィールドデバイス15～22は、センサ、バルブ、トランスマッタ、ポジショナなど、いかなる種類の装置でもよく、I/Oカード26、28は、所望の通信プロトコルまたはコントローラプロトコルに従ういかなる種類のI/O装置でもよい。図1に示す実施例では、フィールドデバイス15～18は、アナログ回線を通してI/Oカード26と通信する標準的な4-20ミリアンペア装置であり、フィールドデバイス19～22は、フィールドバスプロトコル通信を用いてデジタルバスを介してI/Oカード28と通信する、フィールドバスフィールドデバイスのようなスマート装置である。当然、フィールドデバイス15～22は、あらゆる標準規格または将来開発されるプロトコルを含む、所望する他のいかなる標準規格またはプロトコルに従うものでもよい。10

**【 0 0 1 4 】**

コントローラ11は、プラント10内に分散され、内部に少なくとも一つのプロセッサを有する、多くのコントローラのうちの一つであり、一または複数のプロセス処理ルーチンを実行または監視し、該ルーチンは、その内部に記憶されるかまたはそれに関連する制御ループを有してもよい。コントローラ11はまた、装置15～22、ホストコンピュータ13、およびデータヒストリヤン12と通信し、所望の方法でプロセスを制御する。本明細書に記載される制御ルーチンまたは要素はいかなるものも、所望する場合には、その一部を別のコントローラまたは他の装置に実現または実行させてもよいという点に留意すべきである。同様に、プロセス制御システム10内で実現される、本明細書に記載の制御ルーチンまたは要素は、ソフトウェア、ファームウェア、またはハードウェアなどを含む、いかなる形態でもよい。この点に関して、プロセス制御要素は、例えばコンピュータで読み取り可能な媒体に記憶されたルーチン、ロック、またはモジュールを含むプロセス制御システムの、いかなる部品または一部分であってもよい。モジュール、またはサブルーチン、サブルーチンの部品（たとえば、コード行）などの制御プロシージャの部品である、制御ルーチンは、はしご論理、シーケンシャルファンクションチャート、機能ロック図、オブジェクト指向型プログラミング、または他のいかなるソフトウェアプログラミング言語または設計パラダイムを用いた、所望するいかなるソフトウェアのフォーマットで実現してもよい。同様に、制御ルーチンは、例えば、一または複数のEEPROM、EEPROM、特定用途集積回路（ASIC）、または他のハードウェア要素もしくはファームウェア要素内にハードコードされてもよい。さらに、制御ルーチンは、図形設計ツールまたは他の種類のソフトウェア／ハードウェア／ファームウェアのプログラミングツールもしくは設計ツールを含め、何らかの設計ツールを用いて設計されてもよい。このように、コントローラ11は、所望の方法で、制御戦略または制御ルーチンを実行するように構成されてよい。20

**【 0 0 1 5 】**

一つの実施例において、コントローラ11は、一般的に機能ロックと称されるものを用いて制御戦略を実行する。この戦略において、各機能ロックは、全体の制御ルーチンの部品またはオブジェクトであり、（リンクと呼ばれる通信によって）他の機能ロックと関連して動作し、プロセス制御システム10内のプロセス制御ループを実行する。機能ロックは通常、トランスマッタ、センサまたは他のプロセスパラメータ測定装置と関連する入力機能の如き入力機能、PID制御、ファジー論理制御などを実行する制御ルーチンと関連する制御機能の如き制御機能、または、バルブのような一部の装置の動作を制御する出力機能の一つを実行し、プロセス制御システム10内において何らかの物理的な機能を実行する。当然、ハイブリッド機能ロックおよび他の種類の機能ロックも存在する。機能ロックは、コントローラ11内に記憶されてこれにより実行されてもよいが、これは通常、これらの機能ロックが標準的な4-20ミリアンペア装置、およびHART装置のような何らかの種類のスマートフィールドデバイスに用いられるかもしくはこれらと関連する場合であり、またはフィールドバス装置の場合のようにフィールドデバイス自体の内部に記憶されこれにより実行されてもよい。ここでは、制御システムについて、オブジェクト指向型プログラミングパラダイムを用いる機能ロック制御戦略を用いて説明してい304050

るが、制御戦略または制御ループもしくは制御モジュールは、はしご論理、シーケンシャルファンクションチャートなどの他の規約を用いて、または他の所望のプログラミング言語またはプログラミングパラダイムを用いて実行または設計されてもよい。

#### 【 0 0 1 6 】

図1の拡大ブロック30に示すように、コントローラ11は、ルーチン32、34として示される複数の単一ループ制御ルーチンを有することが可能であり、また制御ループ36として示される—または複数の高度制御ループを実行してもよい。このような各ループは通常、制御モジュールと称される。単一ループ制御ルーチン32は、適切なアナログ入力(AI)機能ブロックおよびアナログ出力(AO)機能ブロックに接続された单一入力/单一出力ファジー論理制御ブロックを用いて信号ループ制御を実行し、単一ループ制御ルーチン34は、適切なアナログ入力(AI)機能ブロックおよびアナログ出力(AO)機能ブロックに接続された单一入力/单一出力PID制御ブロックを用いて信号ループ制御を実行するものとして示され、これらの機能ブロックは、バルブのようなプロセスコントローラと関連してもよく、温度トランスミッタおよび圧力トランスミッタのような測定装置と関連してもよく、またプロセス制御システム10内の他の装置と関連してもよい。高度制御ループ36は、複数のAI機能ブロックに通信可能に接続された入力と複数のAO機能ブロックに通信可能に接続された出力とを有する高度制御ブロック38を含むものとして示されるが、高度制御ブロック38の入力および出力は、他の所望の機能ブロックまたは制御要素と通信可能に接続され、他の種類の入力を受信し、他の種類の制御出力を送信してもよい。後述するが、高度制御ブロック38は、モデル予測制御ルーチンとオプチマイザとを統合して、プロセスまたはプロセスの一部の最適制御を実行する制御ブロックでもよい。本明細書内では高度制御ブロックはモデル予測制御(MPC)ブロックを含むものとして述べられるが、ニューラルネットワークモデリングまたは制御ルーチン、多変数ファジー論理制御ルーチンなどの、他の多重入力/多重出力制御ルーチンまたはプロシージャでもよい。高度制御ブロック38を含む図1に示される機能ブロックは、コントローラ11により実行されるか、またはワークステーション13のうちの一つもしくはフィールドデバイス19~22のうちの一つのような他の処理装置内に位置しこれらにより実行されることは理解されるであろう。

#### 【 0 0 1 7 】

図1に示すように、ワークステーション13のうちの一つは高度制御ブロック生成ルーチン40を有し、該ルーチンは、高度制御ブロック38を作成、ダウンロード、および実行するのに用いられる。高度制御ブロック生成ルーチン40は、ワークステーション13内のメモリに格納されプロセッサにより実行されうるが、これに加えてまたはこれに代えて、該ルーチン(またはその一部)は、所望する場合は、プロセス制御システム10内の他のデバイス内に格納されこれによって実行されてもよい。一般的に、高度制御ブロック生成ルーチン40は、高度制御ブロックを作成し該高度制御ブロックをプロセス制御システム内に接続する制御ブロック作成ルーチン42と、高度制御ブロックにより収集されたデータに基づいてプロセスのためのプロセスモデルまたはその一部を作成するプロセスモデル化ルーチン44と、プロセスモデルから高度制御ブロックのための制御論理パラメータを作成し、これらの制御論理パラメータを、プロセス制御に用いるために高度制御ブロック内に記憶またはダウンロードする制御論理パラメータ作成ルーチン46と、高度制御ブロックと併せて用いるオプチマイザを作成するオプチマイザルーチン48とを有する。これらのルーチン42、44、46、48は、一連の異なるルーチンから構成されうる。この一連のルーチンには、プロセス出力を受信するように構成された制御入力部と制御信号をプロセス入力部へ提供するように構成された制御出力部とを有する高度制御要素を作成する第一のルーチンと、プロセス制御ルーチン(所望の設定ルーチンでよい)内の高度制御要素をユーザがダウンロードし通信可能に接続できるようにする第二のルーチンと、高度制御要素を用いて各プロセス入力に対し励起波形を提供する第三のルーチンと、高度制御要素を用いて励起波形に対する各プロセス出力の応答を反映するデータを収集する第四のルーチンと、高度制御ブロックのための入力のセットを選択するまたはユーザが選択できるようにする第五のルー

10

20

30

40

50

チングと、プロセスモデルを作成する第六のルーチンと、プロセスモデルから高度制御論理パラメータを作成する第七のルーチンと、高度制御論理および必要な場合はプロセスモデルを高度制御要素内に配置して高度制御要素がプロセスを制御できるようにする第八のルーチンと、高度制御ブロック38内で用いるオプチマイザを選択するまたはユーザが選択でききるようにする第九のルーチンとが含まれる。

#### 【0018】

図2は、プロセス50に通信可能に連結された高度制御ブロック38の一つの実施例を更に詳細に示したブロック図である。高度制御ブロック38は、他の機能ブロックに提供する操作変数M Vのセットを生成し、これらの他の機能ブロックはプロセス50の制御入力に接続されることがこの図から理解される。図2に示すように、高度制御ブロック38は、MPCコントローラブロック52と、オプチマイザ54と、目標変換ブロック55と、ステップ応答モデルまたは制御行列56と、入力処理 / フィルタブロック58とを有している。MPCコントローラ52は、標準的な $M \times M$ 正方形 ( $M$ は1を超える任意の数でよい) のMPCルーチンまたはプロシージャで、出力と同数の入力を有する。MPCコントローラ52は、入力として、プロセス50内で測定されるN個の制御変数C Vおよび補助変数A V (値のベクトル) よりなるセットと、未来のある時期にプロセス50に提供される周知のまたは予想される変化または外乱である外乱変数D Vのセットと、目標変換ブロック55から提供される定常状態目標制御変数C V<sub>T</sub>および補助変数A V<sub>T</sub>よりなるセットとを受け取る。MPCコントローラ52は、これらの入力を用いてM個の操作変数M Vよりなるセットを(制御信号の形で) 作成し、操作変数M V信号を供給してプロセス50を制御する。

#### 【0019】

さらに、MPCコントローラ52は、制御変数C V、補助変数A Vおよび操作変数M Vの各々の制御ホライゾン上の予測値を表す、予測定常状態制御変数C V<sub>ss</sub>のセットと、予測定常状態補助変数A V<sub>ss</sub>のセットと、予測定常状態操作変数M V<sub>ss</sub>のセットとを算出して入力処理 / フィルタブロック58に提供する。入力処理 / フィルタブロック58は、制御変数C V<sub>ss</sub>、補助変数A V<sub>ss</sub>、および操作変数M V<sub>ss</sub>の上述の決定された予測定常状態値を処理し、これらの変数に対するノイズおよび予測し得ない外乱の影響を減少させる。入力処理 / フィルタブロック58は、ローパスフィルタ、またはこれらの値に対するノイズ、モデル化誤差、および外乱の影響を減少させる他のいかなる入力処理をも含んでもよく、またフィルタ処理された制御変数C V<sub>ssfil</sub>、補助変数A V<sub>ssfil</sub>、および操作変数M V<sub>ssfil</sub>をオプチマイザ54に提供する、ことは理解されるであろう。

#### 【0020】

オプチマイザ54は、この例では線形計画法(LP)オプチマイザであり、選択ブロック62から提供される目的関数(O F)を用いてプロセスの最適化を行う。あるいは、オプチマイザ54は二次計画法オプチマイザ、つまり線形モデルと二次目的関数を有するオプチマイザでもよい。一般的に、目的関数O Fは、複数の制御変数、補助変数、および操作変数の各々に関連するコストまたはプロフィットを特定し、オプチマイザ54は、この目的関数を最大化または最小化するこれらの変数の目標値を設定する。選択ブロック62は、オプチマイザ54に提供する目的関数O Fを、プロセス50の最適動作を定めるさまざまな方法を数学的に表し、前もって格納された目的関数64のセットのうちの一つとして選択してもよい。例えば、ある前もって格納された目的関数64はプラントのプロフィットを最大化するよう構成され、別の目的関数64は供給が不足している特定の原材料の使用を最低限に抑えるよう構成され、また別の目的関数64はプロセス50内で製造される製品の品質を最大化するよう構成されてもよい。一般的に、目的関数は、制御変数、補助変数および操作変数のそれぞれの移動にともなうコストまたはプロフィットを用いて、制御変数C Vの設定ポイント値または範囲と補助変数A Vおよび操作変数M Vのリミットによって定められ許容されるポイントのセット内において最適なプロセス動作ポイントを決定する。当然、本明細書に記載されるものに代えてまたはこれに加えて、原材料の使用やプロフィット率などの複数の懸念事項の各々をある程度最適化するような目的関数を含む所望の目的関数を用いることもできる。

10

20

30

40

50

## 【 0 0 2 1 】

上述の目的関数64のうちの一つを選択するために、ユーザまたはオペレータは、オペレータ端末またはユーザ端末（たとえば、図1のワークステーション13の一つ）上で目的関数64のうちの一つを選択することにより、使用する目的関数64を提示することが可能であり、この選択は入力66を介して選択ブロック62に提供される。入力66に応答して、選択ブロック62は選択された目的関数O Fをオプチマイザ54に提供する。当然、ユーザまたはオペレータはプロセスの動作中に使用する目的関数を変更することができる。ユーザが目的関数を提供または選択しない場合には、所望であれば、デフォルト目的関数を用いてもよい。可能なデフォルト目的関数の一つを以下に詳しく説明する。高度制御ブロック38の一部として示されてはいるが、さまざまな目的関数を図1のオペレータ端末13に格納し、これらの目的関数のうちの一つを高度制御ブロック38の作成中または生成中に該ブロックに提供してもよい。10

## 【 0 0 2 2 】

目的関数O Fに加えて、オプチマイザ54は入力として、制御変数設定ポイントのセット（プロセス50の制御変数C Vのためにオペレータが特定した設定ポイントであるのが一般的であり、オペレータまたは他のユーザによって変更することができる）と、これらの制御変数C Vの各々に関連する範囲および重み付けまたは優先順位とを受け取る。さらにオプチマイザ54は、プロセス50の制御に用いられる、補助変数A Vのための範囲または制約リミットのセットおよび重み付けまたは優先順位のセットと操作変数M Vのリミットのセットとを受け取る。一般的には、補助変数および操作変数の範囲が補助変数および操作変数のリミットを定め（通常はプラントの物理的な特性に基づく）、制御変数の範囲が制御変数がプロセスをうまく制御するために動作できる範囲を提供する。制御変数、補助変数および操作変数の重み付けは、最適化プロセス中の制御変数、補助変数および操作変数の相互の相対的重要性を特定してもよく、状況によって、これらの変数と関連する制約の一部が侵害される場合は、オプチマイザ54が制御目標ソリューションを生成できるようにすべく用いられてもよい。20

## 【 0 0 2 3 】

動作中、オプチマイザ54は、線形計画法（L P）技術を用いて最適化を行ってもよい。公知のように、線形計画法は目的関数と称される特定の追加関数を最大化または最小化する線形等式および不等式のセットを解くための数学的な技術である。上記のとおり、目的関数はコストやプロフィットといった経済価値を表すこともある。理解されるとおり、定常状態ゲイン行列は、操作変数および制御変数または補助変数の可能な各対のための定常状態ゲインを定める。言い換えると、定常状態ゲイン行列は、操作変数および外乱変数の各々のユニット変化のための各制御変数および補助変数における定常状態ゲインを定める。この定常状態ゲイン行列は一般的にN × M行列であり、Nは制御変数および補助変数の数、Mはオプチマイザルーチンで用いられる操作変数の数である。一般的に、NはMよりも大きくなる場合があるが、等しくても、小さくてもよく、最も一般的なケースではNはMよりも大きい。

## 【 0 0 2 4 】

公知のまたは標準的なL Pアルゴリズムまたは技術を用いると、一般に、オプチマイザ54は、制御変数C V設定ポイント範囲リミット、補助変数A V制約リミット、および操作変数M Vリミット内でプロセス動作を行いながら、選択された目的関数O Fを最大化または最小化する目標操作変数M V<sub>T</sub>のセットを決定するために（定常状態ゲイン行列から決定される）反復動作する。一つの実施例では、オプチマイザ54は操作変数の変化量を実際に判断し、予測定常状態における制御変数C V<sub>SSfil</sub>、補助変数A V<sub>SSfil</sub>および操作変数M V<sub>SSfil</sub>の表示値を利用して、プロセスの現行の動作からのプロセス動作の変更を決定する、つまり、目標プロセス動作ポイントまたは最適プロセス動作ポイントに到達する行程中におけるM P C制御ルーチンの動的動作を決定する。この動的動作は、現行の動作ポイントから目標動作ポイントへの移行中においていかなる制約リミットも侵害されないことを確実にする必要があるので重要である。4050

## 【0025】

一つの実施例においては、LPオプチマイザ54は以下の形式の目的関数を最小化するよう設計されてもよい。

## 【0026】

## 【数1】

$$Q = P^t * A * \Delta M V + C^t * \Delta M V$$

10

ここで、

$Q$  = 総コスト/プロフィット、

$P$  =  $A V_s$ および $C V_s$ と関連するプロフィットベクトル、

$C$  =  $M V_s$ と関連するコストベクトル、

$A$  = ゲイン行列、

$M V$  =  $M V_s$ における変化量を算出するためのベクトル。

## 【0027】

プロフィット値は正数であり、コスト値は負数であり、これらはオブジェクトに対する各々の影響を示す。この目的関数を用いて、LPオプチマイザ54は、制御変数 $C V$ が目標設定ポイントからのある範囲内に納まり、補助変数 $A V$ がそれ自身の上限制約リミットおよび下限制約リミット内に納まり、操作変数 $M V$ がそれ自身の上限リミット内および下限リミット内に納まるることを確保しながら目的関数を最小化するような操作変数 $M V$ の変化量を算出する。

20

## 【0028】

使用可能な一つの最適化プロシージャでは、操作変数の増分値が現時点( $t$ )において用いられ、操作変数の増分の合計が制御ホライゾン上において用いられ、制御変数および補助変数の増加がその時点の位置における値の代りに予測ホライゾンの終点で決定される。これはLPアプリケーションにおいては一般的である。当然、LPアルゴリズムはこのバリエーションの目的で適切な修正が加えられてもよい。いずれの場合であっても、LPオプチマイザ54は定常状態モデルを用いてもよく、その結果、そのアプリケーションには一般的に定常状態が要求される。MPC設計で普段用いられる予測ホライゾンにより、自己調整プロセスのために将来の定常状態が保証される。予測ホライゾンを $p$ 、制御ホライゾンを $c$ とすると、増分形式で表される $m \times n$ 入力・出力プロセスのに対する一つの可能な予測プロセス定常状態式は次の通りである。

30

## 【0029】

## 【数2】

$$\Delta C V(t+p) = A * \Delta M V(t+c)$$

40

ここで、

【数3】は、予測ホライゾン( $t + p$ )の終点での出力の予測変化を示し、

【数4】は、プロセス定常状態 $m \times n$ ゲイン行列であり、

【数5】は、制御ホライゾン( $t + c$ )の終点での操作変数における変化量を示す。

## 【0030】

【数3】

$$\Delta C V(t+p) = \begin{bmatrix} \Delta cv_1 \\ \dots \\ \Delta cv_n \end{bmatrix}$$

【0 0 3 1】

10

【数4】

$$A = \begin{bmatrix} a_{11} \dots a_{1m} \\ \dots \dots \dots \\ a_{n1} & a_{mn} \end{bmatrix}$$

20

【0 0 3 2】

【数5】

$$\Delta M V(t+c) = \begin{bmatrix} \Delta mv_1 \\ \dots \\ \Delta mv_m \end{bmatrix}$$

30

ベクトル  $MV(t + c)$  は、すべてのコントローラ出力  $mv_i$  によってたらされる制御ホライゾン上の変化量の合計を表すため、次のようになる。

【0 0 3 3】

【数6】

$$\Delta mv_i = \sum_{j=1}^c mv_i(t+j) \quad i = 1, 2, \dots, m$$

40

これらの変化量は、操作変数  $MV$  および制御変数  $CV$  の双方のリミットを満たすことが必要であり（ここで補助変数は制御変数として扱われる）、したがって次のようになる。

【0 0 3 4】

【数7】

$$MV_{\min} \leq MV_{current} + \Delta MV(t+c) \leq MV_{\max}$$

50

【0035】

【数8】

$$CV_{\min} \leq CV_{predicted} + \Delta CV(t+p) \leq CV_{\max}$$

この場合、製品価値を最大化し原材料コストを最小化する目的関数は、次のように併せて定めることができる。

【0036】

【数9】

10

$$\underset{\min}{Q} = -UCV^T * \Delta CV(t+p) + UMV^T * \Delta MV(t+c)$$

ここで、

$UCV$  は制御変数  $CV$  プロセス値におけるユニット変化のためのコストベクトル、

$UMV$  は操作変数  $MV$  プロセス値におけるユニット変化のためのコストベクトルである。

【0037】

上記の【数1】を適用すると、目的関数は操作変数  $MV$  に関して次のように表すことができる。

20

【0038】

【数10】

$$\underset{\min}{Q} = -UCV^T * A * \Delta MV(t+c) + UMV^T * \Delta MV(t+c)$$

L P アルゴリズムは、最適な解を見出すために、【数10】で定められる領域内の初期頂点を求めるために目的関数を計算し、目的関数の最大値（最小値）を有する頂点が最適解であると該アルゴリズムが決定するまで、各ステップ毎に解を改善する。そして、決定された最適操作変数値は、制御ホライゾン内で達成される目標操作変数  $MV_T$  としてコントローラに適用または提供される。

30

【0039】

一般的に、用意された行列上で L P アルゴリズムを実行すると、三つの起こりうる結果が帰って来る。一つ目は、目標操作変数  $MV_T$  の固有の解が得られること。二つ目は、その解には境界がないこと。ただし、これは制御変数および補助変数各々に上限と下限がある場合には起こらない。三つ目は、プロセス変数のリミットを満たす解がないこと。つまり、これはプロセス変数の境界または制約が厳しすぎることを意味する。三つめのケースに対処するためには、制約全体を緩和し、緩和した制約で再びオプチマイザを実行して解を得てもよい。基本的に、操作変数のリミット（上限 / 下限）をオプチマイザが変更することはできないことが前提となっている。補助変数の制約またはリミット（上限 / 下限）にも同様の前提を用いることができる。しかし、オプチマイザは、制御変数  $CV$  を任意の設定ポイントへ移動させること（ $CV$  設定ポイント制御）から、制御変数  $CV$  をこの設定ポイントからの任意の範囲内の数値またはその周辺の範囲内の数値のうちのいずれかに移動させること（ $CV$  範囲制御）に変更することができる。この場合、制御変数の値は特定の設定ポイントではなくある範囲内に置かれうる。自身の制約を侵害する補助変数  $AV$  がいくつかあり、 $CV$  設定ポイント制御から  $CV$  範囲制御へと移行しても解をもたらさない場合には、提供された重み付けまたは優先順位の指定に基づいて補助変数の制約を緩和または無視することも可能である。一つの実施例では、補助変数の平方誤差を最小化して各々がその制約の侵害を許容するかまたは、優先順位が最も低い補助変数の制約を順次放棄

40

50

することにより解が決定されうる。

#### 【0040】

上記のように、目的関数OFは制御ブロック生成プログラム40により選択またはデフォルトによって設定されうる。このようなデフォルト設定を行う一つの方法が以下に提供されている。具体的には、最適化能力を与えることが望ましいが、補助変数および操作変数の動作制約を保持しながら制御変数の設定ポイントを維持することのみを要求する場合が多い。これらのアプリケーションの場合には、ブロック38は単なるMPC機能ブロックとして動作するように構成されてもよい。このような簡単な使用を可能にするために、デフォルト「動作」目的関数は、デフォルト補助変数AVの重み付けとともにデフォルトコストを目的関数内のさまざまな変数に割り当てて自動的に作成されてもよい。これらのデフォルトは、補助変数AVまたは操作変数MVに対するすべてのコストを等しく設定するかまたは、これらの補助変数AVおよび操作変数MVに対して別の所定のコストを割当てもよい。エキスパートオプションが選択された場合には、ユーザは追加の最適化の選択肢を作成し、それに関連するコストをさまざまな目的関数64に対して定めてもよい。また、エキスパートユーザは、デフォルト目的関数のデフォルト補助変数AV重み付けおよびデフォルト制御変数CV重み付けを修正することもできる。10

#### 【0041】

例えばプロセス構成に関して経済的要素が規定されていない場合の一つの実施例において、MPC構成から自動的に目的関数を作成してもよい。一般的に、目的関数は次の数式を用いて作成することができる。20

#### 【0042】

#### 【数11】

$$CD^T = C^T + P^T * A = [C_i, \dots, C_j, \dots, C_m] + \left[ \sum_{i=1}^n p_i a_{i1}, \sum_{i=1}^n p_i a_{i2}, \dots, \sum_{i=1}^n p_i a_{im} \right] = [CD_i, \dots, CD_j, \dots, CD_m]$$

変数C<sub>j</sub>およびp<sub>j</sub>は構成設定から定義できる。とくに、下限(LL)または上限(HL)のみで制御変数CV設定ポイントを定義できる仮定すると、p<sub>j</sub>値は次のように定義できる。30

#### 【0043】

設定ポイントがLLで定義されるかまたは最小値が選択される場合、p<sub>j</sub> = -1

設定ポイントがHLで定義されるかまたは最大値が選択される場合、p<sub>j</sub> = 1

補助変数AVに関する構成情報が入力されないと仮定すると、補助変数AVについてはすべてp<sub>j</sub> = 0である。同様に操作変数MVについても、C<sub>j</sub>値は好みの操作変数目標MV<sub>T</sub>が定義されるかどうかによって異なる。好みの操作目標MV<sub>T</sub>が定義される場合は次のようにになる。

#### 【0044】

MV<sub>T</sub>がHLまたは最大値が選択される場合、C<sub>j</sub> = 1

MV<sub>T</sub>がLLまたは最小値が選択される場合、C<sub>j</sub> = -1

MV<sub>T</sub>が定義されない場合、C<sub>j</sub> = 0

所望する場合は、オプチマイザ54をMPCコントローラ52と併せて使用する選択は調整可能であり、それによって最適化の度合いを提供することができる。この機能を実行するために、コントローラ52で用いられる操作変数MVの変化量は、MPCコントローラ52とオプチマイザ54とによって決定された操作変数MVの変化量にさまざまな重み付けすることによって行うことができる。このような操作変数MVに重み付けを結合させたものを、本明細書において有効MV(MV<sub>eff</sub>)と称する。有効MV<sub>eff</sub>は次のように決定することができる。40

#### 【0045】

## 【数12】

$$\Delta MV_{\text{eff}} = \Delta MV_{\text{mpc}}(1 + \alpha/S) + \Delta MV_{\text{opt}}(1 - \alpha) \quad 0 < \alpha < 1$$

Sは任意または経験的に選択される。通常、Sは1より大きく10の範囲内でありうる。ここで、=1の場合、オプチマイザは生成時点で設定された通り有効出力に寄与する。=0の場合、コントローラはMPCにダイナミック制御のみ提供する。当然、0から1までの間の範囲の場合は、オプチマイザ制御およびMPC制御はさまざまな寄与をもたらす。

10

## 【0046】

上記のデフォルト目的関数は、オプチマイザのさまざまな可能な動作モード中に該オプチマイザの動作を確立すべく用いてよい。とくに、制御変数CVの数が操作変数MVの数と一致する場合、デフォルトの設定に伴って予想される動作は、補助変数AVおよび操作変数MVがそのリミット内に入るよう予測される限り制御変数CVの設定ポイントを維持できるということである。補助変数または操作変数がリミットを超えることが予測される場合には、これらのリミットを侵害しないように制御変数動作設定ポイントをその範囲内で変更することができる。このとき、制御変数をその範囲内で維持しながら補助変数のリミットと操作変数のリミットとを満足させる解をオプチマイザ54が得られない場合には、制御変数はその範囲内で維持され、補助変数および/または操作変数はその制約リミットから逸脱することが許容される。最適な解を探索するために、リミットを侵害すると予測された補助変数AVは平等に扱われ、これらの補助変数AVの平均リミット偏差が最小化されることになる。

20

## 【0047】

この動作を確立するために、目的関数によって用いられるデフォルトコスト/プロフィットは、逸脱が設定ポイントを下回るよう範囲が定義されている場合は制御変数CVに1のプロフィットが割り当てられるように、また逸脱が設定ポイントを上回るよう範囲が定義されている場合は制御変数CVに-1のプロフィットが割り当てられるように、自動的に設定されてもよい。リミット内の補助変数AVには0のプロフィットが割り当てられ、操作変数MVにはコスト0が割り当てられてもよい。

30

## 【0048】

制御変数CVの数が操作変数MVの数よりも少ない場合は、設定された操作変数MV最終休止位置と関連する必要要件に対処すべく追加の自由度を利用することができる。ここで、補助変数および操作変数がそのリミット内に入ると予測される限り、制御変数設定ポイント（制御変数CVが定義されている場合）は維持される。設定最終休止位置からの操作変数の平均偏差は最小化される。補助変数および操作変数のうちの一または複数がそのリミットを侵害すると予測される場合、制御変数動作設定ポイントは、これらのリミットが侵害されることがないよう、設定の範囲内で変更される。この条件下で、複数の解が存在する場合には、制御に用いられる解は、設定最終休止位置からの操作変数の平均偏差を最小にする解である。

40

## 【0049】

補助変数および操作変数を満足させるとともに制御変数を特定の範囲に維持する解をオプチマイザ54が得られなかった（すなわち、解が存在しない）場合には、制御変数が範囲内に維持され、補助変数は制約リミットからずれることができると許容される。最良の解を見つける際に、リミットを侵害すると予測される補助変数は平等に取り扱われ、その平均リミット偏差が最小にされる。この方法を実行するために、目的関数により用いられるデフォルトコスト/プロフィット比は、その範囲が設定ポイントより下方の偏差を許容するよう定義されている場合には1のプロフィットを、その範囲が設定ポイントより上方の偏差を許容するよう定義されている場合には-1のプロフィットを制御変数に割り当てるよう自動的に設定される。補助変数は、1または-1のプロフィットを割り当てられ、操作

50

変数は、0 . 1のコストが割り当てられる。

#### 【0050】

いずれの場合であっても、演算の後、オプチマイザ54は最適操作変数または目標操作変数 $MV_T$ のセットを目標変換ブロック55に提供し、該ブロックは定常状態ゲイン行列を用いて目標操作変数 $MV_T$ から導かれる目標定常状態制御変数および操作変数を決定する。この変換は計算が容易である。なぜなら、定常状態ゲイン行列は、操作変数と制御変数および補助変数との間の相互作用を定めるため、定義された目標（定常状態）操作変数 $MV_T$ から目標操作変数 $CV_T$ および目標補助変数 $AV_T$ を一意に決定するために利用されうるからである。

#### 【0051】

これがいったん決定されると、少なくともN個の目標制御変数 $CV_T$ および目標補助変数 $AV_T$ よりなるサブセットが入力としてMPCコントローラ52へ提供され、上述のように、該コントローラはこれらの目標制御変数 $CV_T$ および目標補助変数 $AV_T$ を用いて新たな定常状態操作変数（制御ホライゾン上の） $MV_{ss}$ のセットを決定する。該 $MV_{ss}$ のセットは、制御ホライゾンの終点における目標値 $CV_T$ および $AV_T$ へと現行の制御変数 $CV$ および操作変数 $AV$ を移動させる。当然、公知の通り、MPCコントローラは、これらの変数 $MV_{ss}$ がオプチマイザ54によって決定される目標操作変数 $MV_T$ となる。オプチマイザ54とMPCコントローラ52は、各プロセススキャン中に上述のとおり動作するため、操作変数の目標値 $MV_T$ はスキャン毎に変更される場合もある。この結果、MPCコントローラは、とくに、ノイズ、予期し得ない外乱、プロセス50の変更などがある場合には、これらの目標操作変数 $MV_T$ の複数のセットうちのいずれの特定のセットに一度も到達しない場合がある。しかしながら、オプチマイザ54は常に、操作変数 $MV$ を最適解へ移動させるようコントローラ52を駆動している。

#### 【0052】

公知のように、MPCコントローラ52は制御予測プロセスマodel70を有しており、これは $N \times (M + D)$ ステップ応答行列（Nは制御変数 $CV$ の数に補助変数 $AV$ の数を足したもの、Mは操作変数 $MV$ の数、Dは外乱変数 $DV$ の数）であってもよい。制御予測プロセスマodel70は、出力72上に、制御変数 $CV$ および補助変数 $AV$ の各々に関して前もって算出した予測を作成し、ベクトル加算器74は現行時間に関するこれらの予測値を実際に測定された制御変数 $CV$ および補助変数 $AV$ の値から差し引いて入力76上にエラーベクトルまたは訂正ベクトルを作成する。

#### 【0053】

次いで、制御予測プロセスマodel70は、 $N \times (M + D)$ ステップ応答行列を用いて、制御予測プロセスマodel70の他の入力に対して提供された外乱変数および操作変数に基づいて制御ホライゾン上で制御変数 $CV$ および補助変数 $AV$ の各々について未来制御パラメータを予測する。また、制御予測プロセスマodel70は、制御変数および補助変数の予測定常状態値 $CV_{ss}$ および $AV_{ss}$ を入力処理／フィルタブロック58にも提供する。

#### 【0054】

制御目標ブロック80は、ブロック38に予め設定された軌跡フィルタ82を用いて、目標変換ブロック55により提供されたN個の目標制御変数 $CV_T$ および補助変数 $AV_T$ の各々に対して制御目標ベクトルを決定する。とくに、軌跡フィルタは、制御変数および補助変数が徐々に各自の目標値へと移動させられる方法を定めるユニットベクトルを提供する。制御目標ブロック80は、このユニットベクトルと目標変数 $CV_T$ および $AV_T$ を用いて、制御ホライゾン時間によって定められた期間における目標変数 $CV_T$ および $AV_T$ の変化を定める、制御変数および補助変数の各々のダイナミック制御目標ベクトルを作成する。続いて、ベクトル加算器84は、ダイナミック制御ベクトルから制御変数 $CV$ および補助変数 $AV$ の各々の未来制御パラメータベクトルを差し引き、制御変数 $CV$ および補助変数 $AV$ の各々のエラーベクトルを定める。次いで、制御変数 $CV$ および補助変数 $AV$ の各々の未来エラーベクトルはMPCアルゴリズムに提供され、制御ホライゾン上の例えれば最小平方誤差な

10

20

30

40

50

どを最小化する操作変数MVステップを選択すべく動作する。当然、MPCアルゴリズムまたはコントローラは、MPCコントローラに入力されるN個の制御変数および補助変数とMPCコントローラ52から出力されるM個の操作変数との関係から生じるM×Mプロセスモデルまたは制御行列を用いる。

#### 【0055】

さらに詳細には、オプチマイザとともに動作するMPCアルゴリズムには、二つの主要目的がある。一つは、動作制約内で、最小のMVの移動でCV制御誤差を最小化しようとすること、もう一つは、オプチマイザによって設定される最適な定常状態MV値と、該最適な定常状態MV値から直接算出される目標CV値とを達成しようとすることがある。

#### 【0056】

これらの目的を達成するため、制約のない元のMPCアルゴリズムをMV目標値を含むように拡張し最小二乗法ソリューションとすることができます。このMPCコントローラの目的関数は次のとおりである。

#### 【0057】

#### 【数13】

$$\min_{\Delta MV(k)} \left\{ \left\| \Gamma^y [CV(k) - R(k)] \right\|^2 + \left\| \Gamma^u \Delta MV(k) \right\|^2 + \left\| \Gamma^o \left[ \sum \Delta MV(k) - \Delta MVT \right] \right\|^2 \right\}$$

10

20

ここで、

$CV(k)$ は、制御された出力のpステップ先の予測ベクトル、

$R(k)$ は、pステップ先の基準軌跡（設定ポイント）ベクトル、

$MV(k)$ は、cステップ先の増分制御移動ベクトル、

$y = diag[y_1, \dots, y_p]$ は、制御出力誤差のペナルティ行列、

$u = diag[u_1, \dots, u_c]$ は、制御移動のペナルティ行列、

pは、予測ホライゾン（ステップ数）、

cは、制御ホライゾン（ステップ数）、

$\circ$ は、オプチマイザにより定義されたMVの目標最適変化量に対する、コントローラ出力の制御ホライゾン上での移動の合計の誤差のペナルティである。

表示を簡潔にするために、目的関数は单一入力／单一出力（SISO）制御として示されている。

#### 【0058】

理解されるように、最初の二つの項は制約のないMPCコントローラの目的関数であり、三番目の項はコントローラの出力移動の合計を最適目標と等しくするための追加条件を設定する。言い換えると、最初の二つの項はコントローラのダイナミック動作のオブジェクトを設定し、三番目の項は定常状態の最適化オブジェクトを設定する。

#### 【0059】

制約のないMPCコントローラの基本解と同様に、このコントローラの基本解は次のように表すことができる。

#### 【0060】

#### 【数14】

30

40

$$\Delta MV(k) = \left( S^{u^T} \Gamma^T \Gamma S^u + \Gamma^{u^T} \Gamma^u \right)^{-1} S^{u^T} \Gamma^T \Gamma E_{p+1}(k) = K_{omp} E_{p+1}(k)$$

ここで、

$MV(k)$ は時間kにおけるMPCコントローラ出力の変化量、

50

$K_{omp_c}$ は最適化されたMPCコントローラゲイン、  
 $S^u$ はプロセスダイナミック行列である。

【0061】

$S^u$ は、SISOモデルについては次元  $p \times c$  のステップ応答から、多重入力 / 多重出力 MIMO モデルについては  $p * n \times c * m$  のステップ応答から構築することができ、ここで  $m$  は操作入力、 $n$  は制御出力である。

【0062】

最適化された MPC の場合には、ダイナミック行列は、SISO モデルについてはサイズ :  $(p + 1) \times m$  まで、MIMO モデルについてはサイズ :  $(p + m) * n \times c * m$  まで拡大し、MV 誤差に対応する。 $E_{p+1}(k)$  は、予測ホライゾン上の CV エラーベクトルであり、MV の目標最適変化量に対する制御ホライゾン上におけるコントローラ出力移動量の合計の誤差である。行列  $\cdot$  は、行列  $\cdot^y$  および  $\cdot^o$  を組み合わせたものであり、SISO コントローラについては次元  $(p + 1)$  の正方行列、多変量コントローラについては  $[n(p + m)]$  の正方行列である。上付きの T は転置行列を示す。

10

【0063】

オプチマイザ54は、制御変数 CV と補助変数 AV とよりなるすべて変数に基づいて最適化し、固有の最適動作ポイントを定義する操作変数 MV<sub>T</sub> の標的セットを決定しているので、MPC コントローラ52が制御行列内の制御変数 CV と補助変数 AV とよりなるサブセットのみを用いて動作し、そこから実際に操作変数 MV 出力が作成されても問題ないとされている。その理由は、コントローラ52が上述の制御変数 CV と補助変数 AV とよりなる選択されたサブセットをそれらに関連する目標値へ移動させた場合、制御変数と補助変数とよりなるの一つの完全セットの残りの変数も、それらの目標値に到達するようになるからである。その結果、M × M 制御行列を有する正方 (M × M) MPC コントローラは、長方 (N × M) プロセスモデルを用いるオプチマイザと共に用いられて、プロセスの最適化を実行する。これにより、非正方形を近似のために反転することなく、またはコントローラ内におけるこのような変換技術に伴うリスクを負うことなく、標準的な MPC 制御技術を標準最適化技術と共に用いることが可能となる。

20

【0064】

一つの実施例では、MPC コントローラが正方形、つまり操作変数 AV の数が制御変数 CV の数と等しい場合、操作変数 MV の目標値は CV 値を次のように変更することによって効果的に実現することができる。

30

【0065】

【数15】

$$\Delta CV = A * \Delta MVT$$

ここで、

$MVT$  は  $MV$  の最適目標値の変化量、

$CV$  は最適  $MV$  に到達するための  $CV$  の変化量である。 $CV$  の変化は  $CV$  設定ポイントを管理することにより実行される。

40

【0066】

動作において、オプチマイザ54は、スキャン毎に、MPC 非制約コントローラの定常状態目標値を設定・更新する。したがって、MPC コントローラ52は非制約アルゴリズムを実行する。目標値  $CV_T$  および  $AV_T$  は、制約を考慮して設定されているので、実現可能な解が存在する限り、コントローラは制約リミット内で動作する。したがって、最適化は MPC コントローラと統合されている。

【0067】

図3および図4は、統合型モデル予測制御・最適化を行うために用いられるステップを説明するフローチャート90である。フローチャート90は、概して二つのセクション、90a (図3) および90b (図4) に分割されており、プロセス動作以前に発生する機能 (90a)

50

およびプロセス動作中、例えばプロセス動作のスキャン毎に発生する機能(90b)を示している。プロセス動作に先立ち、オペレータまたはエンジニアは、制御MPCコントローラとオプチマイザとを統合したものを含む高度制御ブロック38を作成するために複数のステップを実行する。とくに、ブロック92では、高度制御ブロック38として使用するために高度制御テンプレートを選択してもよい。該テンプレートは、ユーザインタフェース13上の設定アプリケーション内のライブラリに記憶されてそこから複製されてもよく、また、MPCコントローラルーチン52および特定のMPCを有しないオプチマイザ54の一般的な算術および論理機能と、プロセスモデルおよび定常状態ゲイン／制御行列と、特定の目的関数とを有してもよい。この高度制御テンプレートは、他のブロックを有するモジュールに設けられてもよく、該他のブロックには、例えば、プロセス50内で他の装置と通信するよう構成される入力ブロックおよび出力ブロック、さらにPID制御ブロック、ニューラルネットワーク制御ブロック、およびファジー論理制御ブロックを含む制御ブロックのような他の種類のブロックが含まれる。一つの実施例では、モジュール内の各ブロックは、ブロック間で通信を行うよう相互に接続された入力および出力を有するオブジェクト指向型プログラムパラダイム内のオブジェクトであることが、理解されるであろう。動作中、該モジュールを実行するプロセッサは、ブロックへの入力を用いて異なる時点で各ブロックを順次実行して該ブロックの出力を作成する。そして、これらは、ブロック間の特定の通信リンクによって定められるように、他のブロックの入力に提供される。

#### 【0068】

オペレータは、ブロック94において、ブロック38で用いられる特定の操作変数、制御変数、制約変数および外乱変数を定義する。所望する場合は、図1のプログラム40などの設定プログラムにおいて、ユーザは制御テンプレートを目視し、入力および出力を選択して記名・設定し、設定環境内の任意の標準ブラウザを用いて閲覧し、制御システム内の実際の入力および出力を見出し、これらの実際の制御変数を制御テンプレート用の入力および出力制御変数として選択してもよい。図5は、設定ルーチンにより作成されたスクリーン表示画面を例示しており、複数のAI(アナログ入力)機能ブロックおよびAO(アナログ出力)機能ブロックと、複数のPID制御機能ブロックと、高度機能ブロックであるMPC-PRO機能ブロックとを含む複数の相互接続された機能ブロックを有する制御モジュールDEB-MPCが例示されている。図5の表示画面の左側の木構造は、たとえばブロック1(block1)、C4\_AI、C4\_DGENなどを含む、DEB-MPC内の機能ブロックを例示している。

#### 【0069】

理解されるように、ユーザは、MPC-PRO機能ブロックへの入力およびそれからの出力と他の機能ブロックの入力および出力との間に線を引くことによりこれらの入力および出力を明記することができる。あるいは、ユーザは、MPC-PROブロックを選択して該MPC-PROブロックのプロパティにアクセスしてもよい。ユーザがMPC-PROブロックのプロパティを見ることを可能にする図6のダイアログボックスの如きダイアログボックスを表示しうる。図6に例示するように、制御変数(controlled)、操作変数(manipulated)、外乱変数(disturbance)、および制約(constraint)(補助)変数の各々に対して異なるタブが設けられてこれらの変数が整理されているが、このような整理は、各変数が20以上の如き複数の変数を有して高度制御ブロック38に関連している場合にとくに必要となる。特定の種類の変数に対するタブ内では、説明(description)、ローリミット(low limit)もしくはハイリミット(high limit)(制約)、およびパス名(path)が提供されうる。加えて、ユーザまたはオペレータは、変数が不良な状態(fail condition)である場合にブロックが取るべき動作を特定することができる。この動作には、たとえば、無動作(no action)、測定値の代りにその変数のシミュレーション値(simulate)の利用、または手動(manual)による入力の許可などが含まれる。さらに、オペレータは、最適化(optimization)を実行するために最小化(minimize)または最大化(maximize)されるべきか否

10

20

30

40

50

かを指定し、またこの変数に関連する優先順位値 ( priority ) または重み付け値およびプロフィット値 ( profit ) を指定することができる。これらの欄は、デフォルト目的関数が用いられていない場合に記入されなければならない。いうまでもなく、ユーザは、ダイアログボックスの右側の適切なボタンを利用して情報または変数を追加 ( add ) 、移動 ( move ) 、修正 ( modify ) 、または削除 ( delete ) してもよい。

#### 【 0070 】

ユーザは、これらの変数のうちの一または複数の変数の情報をその変数を選択することにより特定または変更しうる。この場合、図 7 の R E F L U X FLOW 操作変数のダイアログボックスの如きダイアログボックスがユーザに提示されうる。ユーザは、このダイアログボックスのさまざまなボックス内の情報を変更してもよく、ブラウジングによりその変数のパス名 ( その入力接続部または出力接続部 ) の如き情報を特定してもよい。図 7 のスクリーンを利用して、ユーザは、現行のモジュールまたは M P C - P R O ブロックが設けられている外部のモジュール内をブラウジングすべく内部ブラウズボタン ( internal browse button ) または外部ブラウズボタン ( external browse button ) を選択しうる。いうまでもなく、所望の場合には、オペレータまたはユーザは、アドレス、パス名、タグ名などを手動により提供し、高度制御ブロックの入力接続部および出力接続部を定義しうる。

#### 【 0071 】

ユーザは、高度制御機能ブロックへの入力および出力を選択した後、制御変数と関連する設定ポイントと、制御変数、補助変数および操作変数と関連する範囲またはリミットと、制御変数、補助変数および操作変数の各々と関連する重み付けとを定義してもよい。当然、制約リミットまたは制約リミット範囲の如きこの情報の一部は、既にこれらの変数と関連付けられていることもある。なぜなら、これらの変数がプロセス制御システム設定環境内において選択または探索されるからである。所望する場合は、オペレータは、図 3 のブロック 96 において、操作変数、制御変数および補助変数の各々に関するユニットコストおよび / またはプロフィットを特定することによりオプチマイザ内で用いられる一または複数の目的関数を設定してもよい。当然、オペレータはこの時点で、上述のようにデフォルト目的関数を使用することを選択してもよい。図 8 は、高度制御ブロックの作成に利用する目的関数のセットからユーザまたはオペレータがそのうちの一つを選択することを可能にする、設定ルーチンにより提供されたスクリーン表示画面である。理解されるように、ユーザは、図 8 により提供されるスクリーン表示画面の如き表示画面を利用して前もって格納された目的関数のセットを選択できる。ここでは、標準目的関数 ( standard objective function ) および目的関数 ( objective function ) 2~5 として例示されている。

#### 【 0072 】

入力 ( 制御変数、補助変数および外乱変数 ) が名付けられ、高度制御テンプレートに連結され、重み付け、リミットおよび設定ポイントがそれらに関連付けられると、図 3 のブロック 98 で、この高度制御テンプレートは、制御に使用される機能ブロックとしてプロセス内の選択されたコントローラにダウンロードされる。該制御ブロックの基本的な性質および該制御ブロックの構成方法は、" Integrated Advanced Control Blocks in Process Control System ( プロセス制御システムにおける統合型高度制御ブロック ) " という名称の米国特許第 6,445,963 号に記載されている。該特許は本発明の譲受人に譲渡され、本明細書内に引用され明示的に援用される。該特許は、プロセス制御システム内での M P C コントローラ作成の本質について述べたものであり、オプチマイザがそのコントローラに接続される方法については述べていないが、コントローラを接続し設定するため的一般的なステップが本明細書に述べる制御ブロック 38 に使用でき、テンプレートには、引用特許に記載されているものだけでなく、本明細書で説明された制御ブロック 38 のあらゆる論理要素が含まれるということは理解されるであろう。

#### 【 0073 】

10

20

30

40

50

いずれの場合であっても、高度制御テンプレートがコントローラにダウンロードされた後、オペレータは、ブロック100で、制御テンプレートのテストフェーズを実行させ、MPCコントローラアルゴリズム内で使用されるステップ応答行列およびプロセスモデルを生成することを選択してもよい。前述の特許に記載されている通り、このテストフェーズ中に、高度制御ブロック38内の制御論理は、一連の擬似ランダム波形を操作変数としてプロセスに提供し、制御変数および補助変数（MPCコントローラにより本質的には制御変数として扱われる）の変化を監視する。所望する場合には、操作変数および外乱変数ならびに制御変数および補助変数は、図1のヒストリアン12によって収集されてもよく、また、オペレータは、ヒストリアン12からこのデータを取得してこのデータについて任意の方法で傾向分析を行うべく設定プログラム40（図1）を設定し、これにより、ステップ応答行列を求めてもまたは決定してもよい。該各ステップ応答は、操作変数および制御変数のうちの一つ（一つのみ）の変数におけるユニット変化に対する制御変数または補助変数のうちの一つの変数のある時間内の応答を特定するものである。該ユニット変化は一般的にはステップ変化であるが、インパルス変化またはランプ変化などの他の種類の変化でもよい。一方、所望する場合、制御ブロック38は、擬似ランダム波形をプロセス50に印加するときに収集されるデータに応答してステップ応答行列を生成し、次いで、高度制御ブロック38を作成・インストールするオペレータまたはユーザによって使用されるオペレータインターフェース13にこれらの波形を提供してもよい。

#### 【0074】

図9は、オペレータに収集されて傾向分析されたデータのプロットを提供すべくテストルーチンにより提供されうるスクリーン表示画面を例示しており、これらのプロットにより、オペレータは、ステップ応答曲線、すなわち高度制御ブロックのMPCコントローラ内で用いるプロセスモデルまたは制御行列を作成する際の方向付けができる。とくに、プロット領域101には、複数の入力もしくは出力またはテスト波形に応答する他のデータ（オペレータにより前もって特定されたデータ）がプロットされている。バーグラフ領域102は、傾向分析されたデータ変数の各々のバーグラフを提供し、変数名と、バーグラフ形式の変数の現在値と、適切ならば設定ポイント（バーグラフ上の大きい三角形により示される）と、適切ならばリミット（バーグラフ上の小さい三角形により示される）とを傾向分析される変数の各々に対して例示している。この表示画面の他の領域は、ブロックの目標モード（target mode）および実際モード（actual mode）（104）ならびに定常状態までの設定時間（time to steady state）（106）の如き高度制御ブロックに関する他の項目を例示している。

#### 【0075】

高度制御ブロックのプロセスモデルの作成に先立って、オペレータは、傾向プロット（trend plot）101から使用する必要のあるデータを画面を利用して特定しうる。とくに、オペレータは、ステップ応答を作成するために用いるデータとして、プロット102の開始点および終了点を指定しうる。この領域のデータに緑色の如き異なる色で影をつけることにより視覚的にこの領域が選択された領域であることを示すことができる。同様に、オペレータは、この影を付けられた領域内の領域を特定して排除しうる（ノイズまたは好ましくない外乱などにより代表値ではないため）。この領域はライン112とライン114との間に例示されており、たとえば赤色でこの領域に影を付け、このデータをステップ応答を作成する際に含めてはいけないことを示すことができる。いうまでもなく、ユーザは任意の所望のデータを包含または排除することができ、複数の傾向プロット（図9は八つの傾向プロットが利用可能であることを示している）の各々に対してデータを包含または排除することができる。ここで、異なる傾向プロットは、たとえば、異なる操作変数、制御変数、補助変数などに関連付けられている。

#### 【0076】

ステップ応答のセットを作成するために、オペレータは図9のスクリーン表示画面上のモデル作成ボタン（create model button）116を選択し、傾向プロットからの選択されたデータを用いてステップ応答のセットを作成ルーチンが作成するこ

10

20

30

40

50

とができる。各ステップ応答は、操作変数または外乱変数のうちの一つの変数に対する制御変数または補助変数のうちの一つの変数の応答を示す。この生成プロセスは周知のことであるのでさらに詳細には本明細書において記載しない。

#### 【0077】

ステップ応答行列が作成された後、制御変数と補助変数との数が操作変数の数を上回る場合には、そのステップ応答行列を用いて上述の制御変数と補助変数とよりなるサブセットを選択する。このサブセットが、MPCコントローラ52内で反転され用いられる $M \times M$ プロセスモデルまたは制御行列としてMPCアルゴリズム内で使用される。この選択プロセスは、オペレータによって手動により行われてもよく、また例えばステップ応答行列にアクセスできるユーザインタフェース13内でルーチンによって自動的に行われてもよい。10

一般的に、上述の操作変数のうちの一つと最も密接に関連するものとして上述の制御変数および補助変数のうちの一つが特定される。したがって、上述の制御変数または補助変数（プロセスコントローラへの入力）のうちの单一かつ唯一（つまり異なる）の変数が、上述のさまざまな操作変数（プロセスコントローラの出力）の各々と関連付けされ、よって、MPCアルゴリズムはステップ応答の $M \times M$ セットから作成されるプロセスモデルに基づくことが可能になる。

#### 【0078】

上述の組み合わせを決定するためにヒューリスティックアプローチを取る一つの実施例では、自動ルーチンまたはオペレータは、 $M$ 個の制御変数および補助変数よりなるセット（ $M$ は操作変数の数と等しい）を選択する。これは、上述の操作変数の特定の一つのユニット変化に対する最大ゲインおよび最速応答時間のなんらかの組み合わせを持つ単一の制御変数または補助変数を選択し、これらの二つの変数を対にするためである。当然、場合によっては、特定の制御変数または補助変数が複数の操作変数に対して大きなゲインと速い応答時間を有することもある。このような場合には、該制御変数または補助変数が関連する任意の操作変数と対を成すようにしてもよく、実際は、最大ゲインおよび最速応答時間を生成しない操作変数と対にしてもよい。なぜなら、総計すれば、そのような少ないゲインまたは遅い応答時間を引き起こす操作変数は他の制御変数または補助変数に対しても好ましいと考えられる程度の影響を与える可能性がないからである。したがって、一方が操作変数でもう一方が制御変数または補助変数の対は、全体としては、操作変数と該操作変数に対して最も応答性のある制御変数に相当する、制御変数および補助変数のサブセットとを対にすべく選択される。さらに、制御変数のすべてが $M$ 個の制御変数と補助変数とよりなるサブセットのうちの一つとして選択されなくても問題はないので、MPCコントローラはすべての制御変数を入力として受け取るわけではない。なぜなら、制御変数および補助変数の目標値のセットは、非選択制御変数（および非選択補助変数）がそれ自身の設定ポイントにあるかまたはそれ自身の定められた動作範囲内にあるプロセス動作ポイントを示すようにオプチマイザによって選択されるからである。2030

#### 【0079】

当然、一方には数十さらには数百もの制御変数および補助変数があり、他方には数十または数百もの操作変数があるため、少なくとも視覚的な観点から、さまざまな操作変数の各々に最適に応答する制御変数と補助変数とよりなるセットを選択するのは難しいと考えられる。この問題を解決するために、オペレータインターフェース13内の高度制御プロック生成ルーチン40は、スクリーン表示画面のセットを有するかまたはこれらの表示画面をユーザあるいはオペレータに提示して、動作中にMPCコントローラ52が用いる制御変数と補助変数とよりなるサブセットとして使用すべき制御変数および補助変数をオペレータが適切に選択できるように補助するかまたはこれを可能にするようにしてよい。40

#### 【0080】

したがって、オペレータが図3のブロック120でスクリーンを提示され、特定のまたは選択された一つの操作変数に対する制御変数および補助変数のうちの各々の変数の応答を目視できるようになっていてよい。このようなスクリーンが図10に例示されており、TOP\_DRAWと呼ばれる操作変数に対する複数の制御変数および補助変数（制約とし50

て表示されている)の各々の応答を示している。オペレータは、操作変数全体に渡り、一度に一つずつスクロールしてさまざまな操作変数の各々に対する制御変数および補助変数の各々のステップ応答を目視してもよく、またこのプロセス中に、該操作変数に対して最適に応答する制御変数または補助変数のうちの一つを選択してもよい。オペレータは通常、操作変数に対して最高の定常状態ゲインと最速応答時間との最適な組み合わせを有する制御変数または操作変数を選択しようとする。図11に例示されているように、ダイアログボックスを利用して、制御変数および補助変数のうちの一つをこの操作変数にとって最も重要であるとして選択してもよい。所望ならば、図11に例示されているように、制御変数および補助変数のうちの選択された一つを赤色の如き異なる色で強調し、一方、以前に選択していた変数(すなわち、他の操作変数に対して選択された制御変数および補助変数)は黄色の如き異なる色で強調してもよい。この実施例では、制御ルーチン40はいうまでもなく以前に選択した制御変数および補助変数をメモリ内に格納しているので、オペレータが同じ制御変数または操作変数を二つの異なる操作変数に関連付けしないように確かめるべくチェックしてもよい。ユーザまたはオペレータが他の操作変数に対してすでに選択されている制御変数または操作変数を選択した場合、ルーチン40は、そのユーザまたはオペレータにエラーメッセージを提示し、そのユーザにまたはオペレータに前回選択した制御変数または補助変数を選択していることを通知してもよい。この方法で、ルーチン40は、二つ以上の異なる操作変数に対して同一の制御変数または補助変数が選択されることを防ぐ。

#### 【0081】

図12に例示するように、オペレータまたはユーザは、また、さまざまな操作変数および外乱変数の各々に対するさまざまなステップ応答を目視しうる。図11は、高度制御ブロックを作成するために以前に特定された操作変数および外乱変数の各々に対するTOP-END\_POINTのステップ応答を例示している。もちろん、オペレータは図12のスクリーンを利用して制御変数TOP-END\_POINTに関連するものとして操作変数のうちの一つを選択してもよい。

#### 【0082】

理解されるように、図10乃至図12の表示スクリーンは、MPC制御アルゴリズムの入力として用いられるM個の制御変数および補助変数よりなるサブセットをオペレータが目視して選択できるようにするものであり、これらの変数が非常に多くある場合に特に役立つ。当然、ブロック74で決定された制御変数および制約変数のセットは、何らかの前もって設定された基準または選択ルーチンに基づいて自動的または電子的に選択されてもよい。該選択ルーチンは、制御変数および補助変数に関するステップ応答から決定されるゲイン応答と時間の遅延との組み合わせに基づいて、使用する入力変数を選択してもよい。

#### 【0083】

他の実施例では、自動選択プロセスは、まず、制御行列の条件数に基づいて入力/出力行列を選択することにより制御行列を決定してもよい。例えば、条件数のある所望の程度まで最小化して、制御行列からコントローラ設定を作成してもよい。

#### 【0084】

この例では、プロセスゲイン行列Aの場合、行列 $A^T A$ の条件数を行列制御性のテストを行って決定してもよい。一般的に、条件数が小さければ制御性が高いことを意味し、条件数が大きければ制御性が低くダイナミック制御動作中における制御ステップまたは移動が多いことを意味する。制御性の許容程度を定める厳密な基準がないため、さまざまな有望な制御行列の相対比較として、また悪条件の行列の条件数を用いることもできる。周知の通り、悪条件の行列の条件数は無限に近づく。数学的には、悪条件は、共直線性プロセス変数の場合、つまり制御行列における共直線性の行または列によって発生する。したがって、条件数および制御性に影響を与える主な要因は、行列の行と列との相互関連性である。制御行列内で入力・出力変数を慎重に選択すれば、条件付けに関連する問題を減らすことができる。実際的には、制御行列の条件数が数百(たとえば、500)以上の場合は懸念すべきである。このような行列の場合、コントローラの操作変数の

10

20

30

40

50

移動が著しく過剰になる。

#### 【0085】

上述の通り、制御行列がダイナミック制御の問題を解決する一方、L Pオプチマイザは定常状態最適化問題を解決し、M P Cコントローラブロックが同数でないM V ( A V を含む ) および C V を有する場合でも、制御行列は正方入力・出力行列を有する必要がある。コントローラを生成するために用いる制御行列の入力と出力との選択を始めるために、通常は利用可能なすべての M V が制御出力として含有または選択される。出力 ( M V ) が選択されたら、ダイナミック制御行列の一部を成すプロセス出力変数 ( つまり C V と A V ) を、悪条件でない正方制御行列を作成するような方法で選択しなければならない。

#### 【0086】

ここでは、制御行列内の入力としての C V および A V を自動的にまたは手動により選択する一つの方法を述べるが、他の方法を用いてもよいことは理解される。

#### 【0087】

ステップ1 - 可能な場合は C V の数が M V の数 ( つまり、コントローラ出力の数 ) と等しくなるまで C V を選択する。 M V よりも C V の方が多い場合、 C V は、優先順位、ゲインまたはフェーズ応答、ユーザ入力など、所望の基準に基づいて任意の順序で選択することができる。 C V の可能な総数が M V の数と等しい場合は、ステップ4に進み、結果的に得られる正方制御行列の条件数が許容できるかどうかをテストする。 C V の数が M V の数よりも少ない場合、 A V はステップ2に記載するように選択される。定義される C V がない場合は、 M V に関して最大のゲインを有する A V を選択し、ステップ2へ進む。

#### 【0088】

ステップ2 - 先に選択された C V および A V によって定義される既に選択された制御行列に追加される可能性のある全ての A V に対する条件数を一つずつ算出する。理解されるように、選択された C V によって定義される行列には、選択された C V および A V の各々について行が含まれ、先に選択された各 M V に対する該 C V または A V の定常状態ゲインが定義される。

#### 【0089】

ステップ3 - ステップ2で決定され、結果として得られる行列の条件数が最小になる A V を選択し、該行列を、先の行列に選択された A V が追加されたものとして定義する。このポイントで、 M V の数が、選択された C V の数に選択された A V を加えた数と等しい場合 ( つまりこのポイントで行列が正方行列になった場合 ) は、ステップ4に進む。そうでなければステップ2に戻る。

#### 【0090】

ステップ4 - 作成された正方制御行列 A c の条件数を算出する。所望する場合は、行列  $A_c^T A_c$  の代わりに行列 A c の条件数算出を用いてもよい。なぜなら、これらの異なる行列の条件数は他の平方根として関連付けられているからである。

#### 【0091】

ステップ5 - ステップ4で算出された条件数が受け入れられる場合、すべての C V および選択された A V に M V を関連付ける。これは特定の M V に関して最大のゲインを有する C V または A V を選択することにより、対を成す処理が完了するまで行う。このポイントで選択プロセスは完了する。ただし、条件数が最低許容条件数を超える場合は、制御行列に追加された最後の A V / C V を除去し、ステップ6のラップアラウンドプロシージャを行う。

#### 【0092】

ステップ6 - 選択された各 M V について、一度に一つずつラップアラウンドプロシージャを行い、各ラップアラウンドプロシージャの結果もたらされる行列の条件数を算出する。本質的に、ラップアラウンドプロシージャは、除去された A V ( または C V ) の代わりに異なる M V 各々の 1 ユニット応答を交互に配置することによって行われる。 1 ユニット応答は、行列の行のある位置では 1 ユニットでありそれ以外の位置ではゼロである。要するに、この場合、特定の M V は各々、好条件の正方制御行列を形成する A V の代わりに、

10

20

30

40

50

入力および出力として用いられる。一例として、 $4 \times 4$  行列の場合、1000、0100、0010、および0001の組み合わせが、ゲイン行列 A<sub>c</sub> の除去された AV ラインの行に配置される。

#### 【0093】

ステップ7 - 各 MV についてラップアラウンドプロシージャを行うと、条件数が最小になる組み合わせを選択する。何ら改善されない場合は、当初の行列を維持する。このポイントで、MV それ自体の制御に用いられる MV (つまりラップアラウンドされた MV) を除いた特定の MV に関連して最大のゲインを有する CV または AV を選択し、選択したすべての CV および AV と MV とを関連付ける。

#### 【0094】

当然、このプロシージャによって定義された制御行列およびその結果の条件数はユーザに提供されてもよく、ユーザはコントローラの生成において該定義された制御行列の使用を受け入れてもよく拒絶してもよい。

#### 【0095】

上述した自動プロシージャでは、制御性の改善を目的として MV それ自体を制御する (つまりラップアラウンドする) ために選択されたのは、一つの MV のみであった。手動によるプロシージャでは、ラップアラウンドされる MV の数は任意でよい。MV それ自体を制御するために選択された MV は、コントローラ設定内において該当する出力変数の選択肢が消えていることにより明らかである。また、MV の数が CV の総数に AV を加えた数よりも多い場合は、制御のためのラップアラウンドとしてより多くの MV を用いることができる。このように、最終的には正方制御行列が各 MV を出力として有するコントローラに提供される。ラップアラウンドを実行・使用するプロセスは、制御行列のために選択される CV および AV の数がコントローラによって制御される MV の数よりも少ない場合があることを意味し、この差は、制御行列に入力された MV のラップアラウンドの数であるということが理解されるであろう。さらに、このラップアラウンドプロシージャを CV と AV を足した数が MV の数よりも少ないプロセスで用いることができる。

#### 【0096】

当然、条件数は定常状態ゲインを用いて上述の通り算出され、これによって、制御行列は本質的に定常状態での制御性を定義する。プロセスダイナミックス (デッドタイム、ラグなど) およびモデルの不確実性もまた、動的制御性に影響を与える。この影響はプロセス変数 (例えば制御変数および補助変数) の優先順位を変えることによって対応することができるが、この結果、動的制御に与える影響に起因してプロセスダイナミックスおよびモデルの不確実性を制御行列に入れる必要性が生じる場合もある。

#### 【0097】

定常状態制御性および動的制御性の双方の改善を目的として、他のヒューリスティックプロシージャを用いることも可能である。このようなプロシージャは通常、制御行列を作成するいくつかのフェーズで適用される複数の発見的基準 (相反するものが含まれる場合もある) を有し、これによって制御行列に何らかの改善をもたらす適切な制御入力のセットを選択する。このようなヒューリスティックプロシージャの一つでは、CV および AV は、最大ゲイン関係に基づいて MV によってグループ分けされる。そして、MV のグルーピングの各々について、最速ダイナミックおよび有意のゲインを伴う一つのプロセス出力が選択される。この選択プロセスは、信頼区間を考慮し AV よりも CV を優先してもよい (他がすべて等しい場合は)。次いで、プロセスモデル生成ルーチンは、MPC 制御の生成中に、各グループから選択されたパラメータを用いる。各 MV について単一のパラメータしか選択されないため、応答行列は正方行列であり、反転することが可能である。

#### 【0098】

いずれの場合であっても、MPC コントローラに入力するための M 個 (またはそれ未満) の制御変数および補助変数よりなるサブセットが選択されたら、図 3 のブロック 124 は、決定された正方制御行列から、図 2 の MPC 制御アルゴリズムで用いるプロセスモデルまたはコントローラを生成する。公知の通り、このコントローラ生成ステップは、計算が

10

20

30

40

50

集中するプロシージャである。次いでブロック126は、このMPCプロセスモデル（本来的に制御行列を含む）またはコントローラを、また必要に応じてステップ応答および定常状態応答ゲイン行列を、制御ブロック38にダウンロードし、このデータは動作のために制御ブロック38に組み込まれる。この時点で、該制御ブロック38はプロセス50内のオンライン動作の準備が整う。

#### 【0099】

所望する場合、プロセスステップ応答はこれらのステップ応答を生成したものとは別の方で再構成または提供されてもよい。例えば、ステップ応答のうちの一つをシステム内に記憶された異なるモデルからコピーし、たとえば図10乃至図12のスクリーンに掲示し、ある操作変数または外乱変数に対する任意の制御変数または補助変数のステップ応答を特定してもよい。図13は、ユーザが特定のプロセスまたはモデルのステップ応答のうちの一つを選択・コピー（copy）し、次いで、この同一の応答を他のモデルに付与または貼り付け（paste）、そのステップ応答を新規のモデルに貼り付け、これにより、ユーザが手動でステップ応答モデルを特定することができるスクリーン表示画面を例示している。

10

#### 【0100】

図14は、ユーザがステップ応答のうちの一つ（この図では、TOP-END-POINT TOP-DRAWのステップ応答の場合である）をさらに詳細に見ることができるスクリーン表示画面を例示している。この表示画面には、定常安定ゲイン（steady state gain）、応答時間（response time）、第一次時定数（first order time constant）および平方誤差（squared error）の如きこのステップ応答のパラメータが、ユーザまたはオペレーターが容易に参照できるように例示されている。また所望する場合には、ユーザは、異なるゲインまたは時間制約の如き異なるパラメータを指定することによってステップ応答のプロパティを目視・変更してもよい。ユーザが異なるゲインまたは他のパラメータを指定した場合、該新たなパラメータまたはパラメータのセットを含んで数学的にステップ応答モデルを再生成することができる。この動作は、ユーザがステップ応答のパラメータを知っており、生成されたステップ応答をこれらのパラメータを満足させるようにまたはこれらに一致するように変更する必要がある場合に役立つ。

20

#### 【0101】

30

ここで図4を参照すると、プロセス50がオンライン作動中、図3のフローチャート90aを用いて作成される高度制御ブロック38の各動作サイクルまたはスキャン中に実行される基本的なステップが例示されている。ブロック150で、MPCコントローラ52（図2）は、測定された制御変数CVおよび補助変数AVの値を受け取り、これを処理する。とくに、制御予測プロセスモデルは、CV、AVおよびDVの測定値または入力を処理し、未来制御パラメータベクトルを作成し、さらに予測される定常状態制御変数CV<sub>ss</sub>および補助変数AV<sub>ss</sub>を作成する。

#### 【0102】

次いで、ブロック152で、入力処理／フィルタブロック58（図2）は、MPCコントローラ52によって作成された予測制御変数CV<sub>ss</sub>、補助変数AV<sub>ss</sub>および操作変数MV<sub>ss</sub>を処理またはフィルタ処理し、このフィルタ処理された値をオプチマイザ54に提供する。オプチマイザ54は、ブロック154で、標準的なLP技術を実行し、選択されたまたはデフォルトの目的関数を最大化または最小化するが、補助変数および操作変数のいかなるリミットをも侵害せず制御変数を特定の設定ポイント内またはこれらの変数に関する特定の範囲内に維持する、M個の操作変数目標MV<sub>T</sub>よりなるセットを決定する。一般的に、オプチマイザ54は、制御変数および補助変数の各々をそのリミットに移動させることによって目標操作変数解MV<sub>T</sub>を算出する。上記の通り、制御変数の各々が各自の設定ポイント（当初は制御変数の上限として扱われたこともある）に位置し、補助変数の各々が各自の制約リミット内にある場合に、解が存在することが多い。このような場合、オプチマイザ54は目的関数の最適な結果を生成する決定された操作変数目標MV<sub>T</sub>のみを出力すればよい。

40

50

## 【0103】

しかし、場合によっては、補助変数または操作変数の一部または全部の制約が厳しく、全ての制御変数がそれらの設定ポイントに位置し全ての補助変数が各々の制約リミット内に入る動作ポイントを見出すことが不可能である場合がある。なぜなら、このような解は存在しないからである。このような場合には、上述のように、オプチマイザ54は、補助変数が各自のリミット内で動作する動作ポイントを見出すために、制御変数は特定設定ポイント範囲内で移動することが許容される。この場合でも解を得られない場合、オプチマイザは、その解を得るための一つのリミットであった、補助変数制約リミットのうちの一つを除去し、この除去した補助変数制約リミットを含めずに最適プロセス動作ポイントを決定してもよい。ここで、オプチマイザは、コンストレインリミットとして、いずれの補助変数または制御変数を減らすかを該制御変数および補助変数の各々に割り当てられた重み付けに基づいて選択する（たとえば、最小重み付け値または最高優先順位の変数が最初に除外される）。制御変数の設定ポイント範囲および残りの高優先順位の補助変数のリミットが満足させられる目標操作変数  $MV_T$  が見出されるまで、各々に付与された重み付けまたは優先順位に基づいて補助変数または制御変数をオプチマイザは減らし続ける。10

## 【0104】

次に、ブロック156では、目標変換ブロック55（図2）は、定常状態ステップ応答ゲイン行列を用いて操作変数  $MV_T$  の目標値から制御変数  $CV_T$  および補助変数  $AV_T$  の目標値を決定し、選択したこれらの値の  $N$  個（ $N$  は  $M$  と同数かまたはそれ未満）よりなるサブセットを、目標入力として MPC コントローラ52へ提供する。ブロック158で、MPC コントローラ52は、制御行列またはそこから派生する論理を利用して制約のない MPC コントローラとして上述の通り動作し、これらの目標値の未来  $CV$  ベクトルおよび  $AV$  ベクトルを決定し、未来制御パラメータベクトルとの間でベクトル減算を行って未来エラーべクトルを作成する。MPC アルゴリズムは公知の方法で動作し、 $M \times M$  ステップ応答から生成されたプロセスモデルに基づいて定常状態操作変数  $MV_{ss}$  を決定し、これらの  $MV_{ss}$  値を入力処理 / フィルタブロック58（図2）に提供する。ブロック160で、MPC アルゴリズムはまた、プロセス50に出力される  $MV$  ステップも決定し、これらのステップの第一のステップを任意の適切な方法でプロセス50に出力する。20

## 【0105】

動作中、例えばインタフェース13のうちの一つで駆動する一または複数の監視アプリケーションは、高度制御ブロックまたはこれに通信可能に接続されている他の機能ブロックから、直接またはヒストリアン12を介して情報を定期的に読み出し、ユーザまたはオペレータに対して、高度制御ブロックの動作状態を閲覧するための一または複数の閲覧スクリーンまたは診断スクリーンを提供してもよい。機能ブロック技術は、制御機能ブロックおよび出力機能ブロックの双方について、カスケード入力（CAS\_IN）およびリモートカスケード入力（RCAS\_IN）ならびに対応する逆算出力（BKCAL\_OUT および RCAS\_OUT）を特徴とする。これらのコネクタを用いて、スーパーバイザ的な最適化 MPC 制御戦略を既存の制御戦略に追加し、一または複数の閲覧スクリーンまたは表示画面を用いて該スーパーバイザ制御戦略を見ることが可能である。同様に、最適化された MPC コントローラの目標値は、所望する場合にはスーパーバイザ制御戦略からも修正できる。3040

## 【0106】

図15は、一または複数のそのような閲覧用アプリケーションにより提示されうるスクリーン表示画面の一例であり、高度制御ブロックの動作に属する情報をその動作中にオペレータに提供する、オプチマイザ（optimizer）のダイアログスクリーンを例示している。とくに、プロセスへの入力（inputs）（操作変数  $MV$ ）および出力（outputs）（制御変数および補助変数  $CV$ ,  $AV$ ）が別々に例示されている。これらの変数の各々について、スクリーン表示画面は、変数の名称（記述子（descriptor））と、現行の測定数値（current value）と、適切ならば設定ポイントと、オプチマイザにより計算された目標数値（target value）と、変数の50

変化量のユニット (unit) およびユニット値 (unit value) と、現行の変数値の指標とを例示する。出力変数の場合には、変数が、MPCコントローラで用いられる選択された変数と、MPCコントローラにより決定されたこの変数の予測値と、この変数に予め設定された優先順位とのうちの一つであるか否かの指標も示される。このスクリーンにより、オペレータは、高度制御ブロックの現行の動作状況を目視して高度制御ブロックが制御をどのように実行しているか知ることができる。さらに、ユーザは、外部アプリケーションの動作目標を設定して処理量を協調させるため遠隔からポイントを設定することができるよう制御パラメータの一部を設定しうる。

#### 【0107】

図16は、診断アプリケーションにより生成されうるスクリーン表示画面であり、高度制御ブロックの診断を実行すべくユーザまたはオペレータに提供されうる診断スクリーンを例示している。とくに、図16の診断スクリーンは、制御変数 (controlled variable) および制約 (補助) 変数 (constrained variable) と、操作変数 (manipulated variable) と、外乱変数 (disturbance variable) とを別々に例示している。その各々について、変数の名称または記述子 (descriptor) が、エラー状態またはアラート状態がこの変数に存在するか否かの指標 (最初の列) とともに提供されている。そのようなエラーまたはアラートは、たとえば緑色のチェックマークもしくは赤色の「×」またはその他の所望の方法を利用して画像として表示されることが可能である。また、これらの変数の各々についての数値 (value) および状況 (status) も示される。操作変数の場合には、これらの信号の Back\_Cal (逆算またはフィードバック) 変数の数値および状況が例示されている。いうまでもなく、制御システム内の問題を判断するために必要な情報をオペレータに提供することにより、高度制御ブロックの診断を実行すべくこのスクリーンを利用することができます。もちろん、オペレータが高度制御ブロックおよびそれが実装されたモジュールの診断を実行することを可能にするために他の種類のスクリーンおよび情報を提供することもできる。

#### 【0108】

ここに記載されている高度機能ブロックは、同一の機能ブロック内にオプチマイザが配置されており、したがってMPCコントローラと同じデバイスで実行されるものとして例示されているが、オプチマイザを別のデバイスに実装することも可能である。とくに、オプチマイザは、ユーザのワークステーション13のうちの一つの如き異なるデバイスに設置されてもよく、コントローラの各実行中またはスキャン中に図2と併せて述べたMPCコントローラと通信を行い、目標操作変数 (MV<sub>T</sub>) またはそこから決定される制御変数 (CV) と補助変数 (AV) とよりなるサブセットを算出し、MPCコントローラに提供してもよい。当然、公知のOPCインターフェースの如き特別なインターフェースを用いて、コントローラと、内部にMPCコントローラを有する機能ブロックと、オプチマイザを実施または実行するワークステーションまたは他のコンピュータとの間に通信インターフェースを提供してもよい。図2に関して述べた実施例と同様に、オプチマイザおよびMPCコントローラは依然として各スキャンサイクル中に相互に通信を行い、統合した最適MPC制御を実行する必要がある。しかしながら、この場合には、プロセス制御環境内すでに存在しうる公知のオプチマイザまたは標準オプチマイザの如き他の種類のオプチマイザを利用してもよい。また、最適化問題が非線形であり、その解には非線形計画法技術が必要な場合でも上述の特徴を利用することが有利でありうる。

#### 【0109】

高度制御ブロックおよび本明細書に述べた他のブロックおよびルーチンはフィールドバスおよび標準的な4-20ミリアンペア装置と併せて用いられると本明細書において述べてきたが、これらは当然ながら、他のプロセス制御通信プロトコルまたはプログラミング環境を用いて実施することもでき、また他の種類の装置、機能ブロックまたはコントローラとともに使用してもよい。ここに述べる高度制御ブロックおよび関連する生成ルーチンおよびテストルーチンはソフトウェアにより実現されるのが好ましいが、これらはハード

10

20

30

40

50

ウェア、ファームウェアなどで実現してもよく、プロセス制御システムと関連するその他のプロセッサで実行されてもよい。したがって、本明細書で述べたルーチン40は、所望する場合は、標準的な汎用CPUまたは例えばASICなどの特定用途向けハードウェアまたはファームウェアにより実現されてもよい。ソフトウェアで実現する場合、該ソフトウェアは、磁気ディスク、レーザーディスク、光ディスク、または他の記憶媒体などのコンピュータ読み取り可能メモリに、またコンピュータあるいはプロセッサなどのRAMまたはROMなどに格納されてもよい。同様に、該ソフトウェアは、例えばコンピュータ読み取り可能ディスクまたは他の移動可能なコンピュータ記憶メカニズム、または、電話回線、インターネットなどの通信チャネルを通じた変調（これは移動可能記憶媒体を介してかかるソフトウェアを提供するのと同様または相互交換可能であるとみなされる）を含む、既知のまたは所望の提供方法を介してユーザまたはプロセス制御システムへ提供されてもよい。  
10。

#### 【0110】

したがって、本発明は特定の例を参照して説明されたが、これは単に説明目的のためであり本発明を限定するものではない。ここに開示される実施例に対し、本発明の精神および範囲を逸脱することなく変更、追加または削除を行うことが可能であることは、当業者には明らかであろう。

#### 【図面の簡単な説明】

#### 【0111】

【図1】オプチマイザをMPCコントローラと統合する高度のコントローラ機能ブロックを有する制御モジュールを備えるプロセス制御システムのブロック図である。

20

【図2】統合型オプチマイザ・MPCコントローラを備える、図1の前記高度コントローラ機能ブロックのブロック図である。

【図3】図2の前記統合型オプチマイザ・MPCコントローラ機能ブロックを作成・搭載する方法を例示するフローチャート図である。

【図4】オンライン状態でプロセスが動作中の、図2の前記統合型オプチマイザ・MPCコントローラの動作を例示すフローチャートである。

【図5】プロセス制御を実行する制御モジュール内の高度制御ブロックを例示する設定ルーチンのスクリーン表示画面である。

【図6】図5の前記高度制御ブロックの特性を示すダイアログボックスを例示する設定ルーチンのスクリーン表示画面である。

30

【図7】図5の前記表示画面に描画される高度制御機能ブロックの入力／出力を選択または特定する方法を例示する設定ルーチンのスクリーン表示画面である。

【図8】ユーザまたはオペレータが高度の制御ブロックの作成に利用するオブジェクティブ機能のセットのうちの一つを選択することを可能にする設定ルーチンにより提供されたスクリーン表示画面である。

【図9】高度の制御ブロックの作成中に、ユーザがプロセスモデルの試験と作成とを実行することを可能にするべく利用されうる試験スクリーンのスクリーン表示画面である。

【図10】図9の前記制御変数および補助変数のうちの一つを前記操作変数に主に関連するものとして選択する方法を例示した設定ルーチンのスクリーン表示画面である。

40

【図11】特定の操作変数への異なる制御変数および補助変数の応答を示す複数のステップ応答を例示した設定ルーチンのスクリーン表示画面である。

【図12】前記操作変数のうちの異なる変数への同一の制御変数または補助変数の応答を示す複数のステップ応答を例示した設定ルーチンのスクリーン表示画面である。

【図13】異なるモデルにおいて利用すべくコピーされるモデルのステップ応答のうちの一つをコピーする方法を例示した設定ルーチンのスクリーン表示画面である。

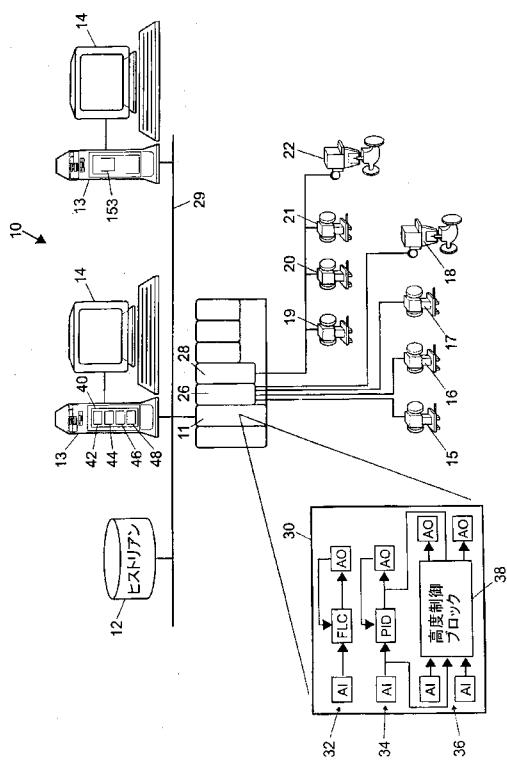
【図14】特定の操作変数への異なる制御変数および補助変数の応答を示す複数のステップ応答を例示した設定ルーチンのスクリーン表示画面である。

【図15】特定の操作変数への異なる制御変数および補助変数の応答を示す複数のステップ応答を例示した設定ルーチンのスクリーン表示画面である。

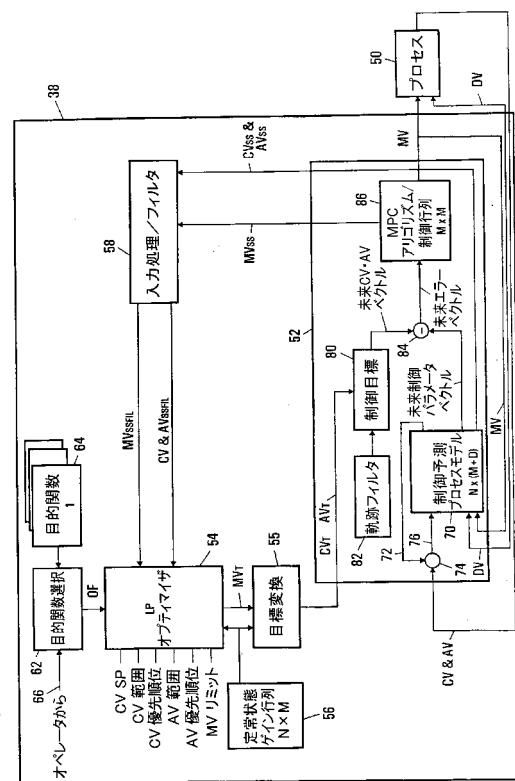
50

【図1】特定の操作変数への異なる制御変数および補助変数の応答を示す複数のステップ応答を例示した設定ルーチンのスクリーン表示画面である。

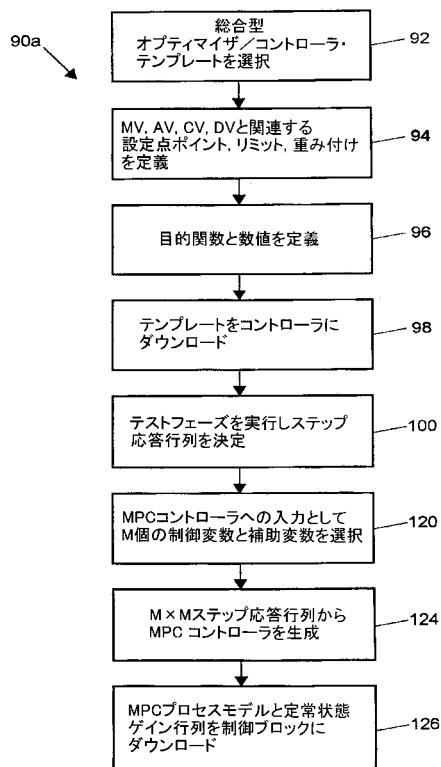
【図1】



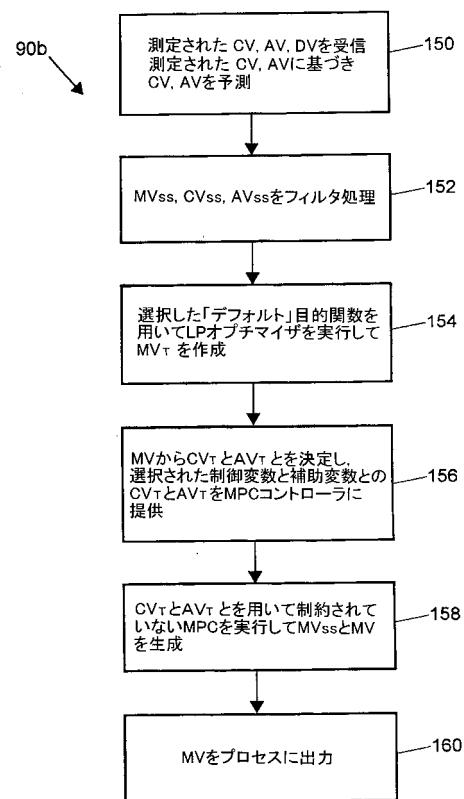
【図2】



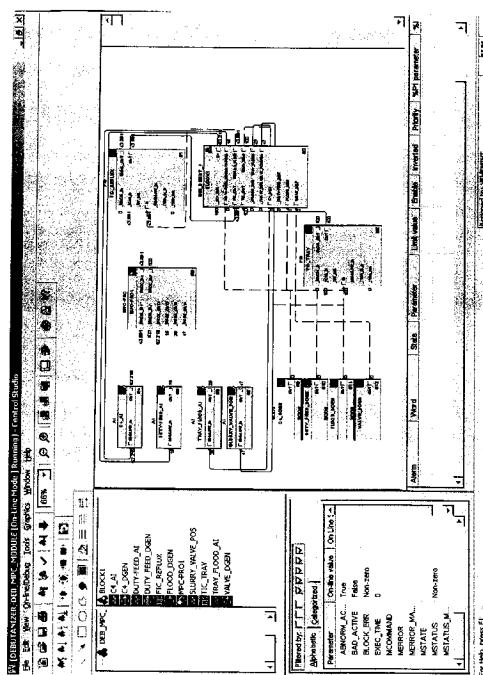
【図3】



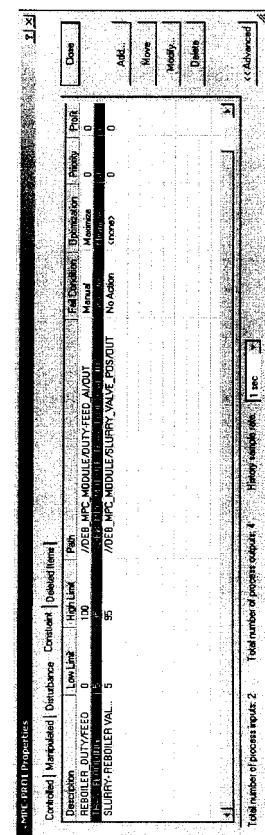
【図4】



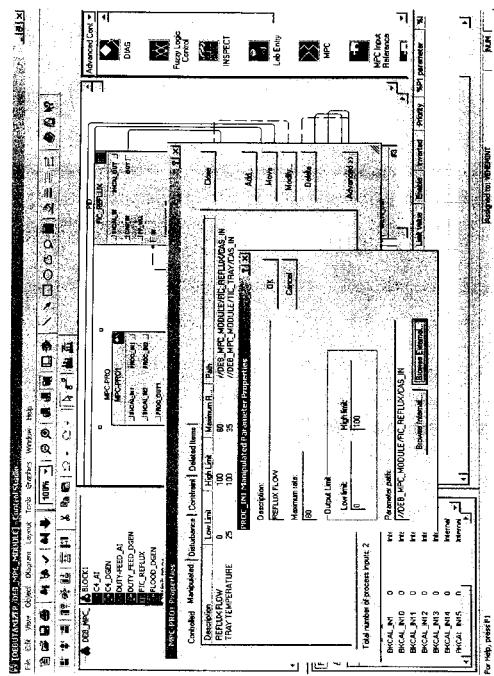
【 図 5 】



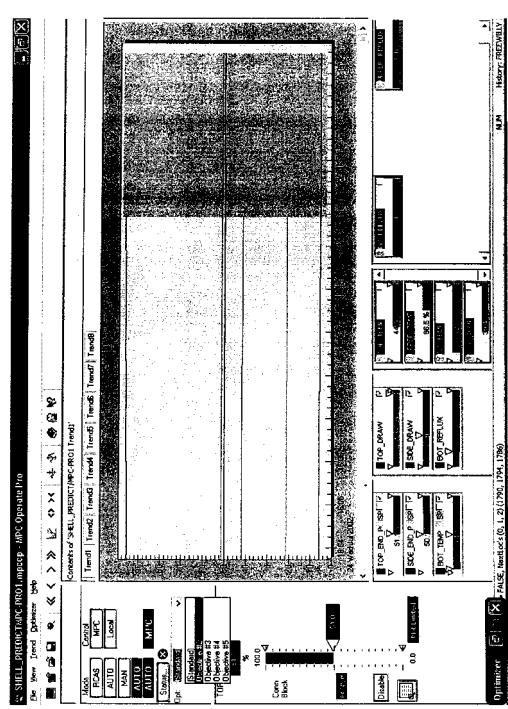
【 四 6 】



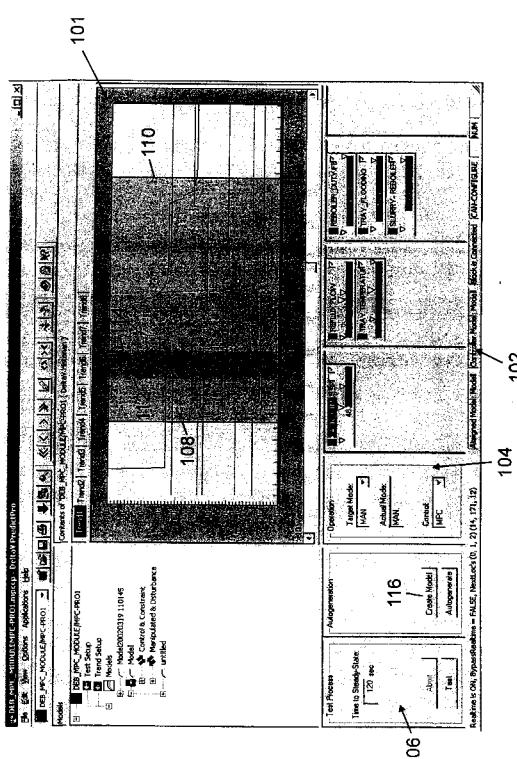
【図7】



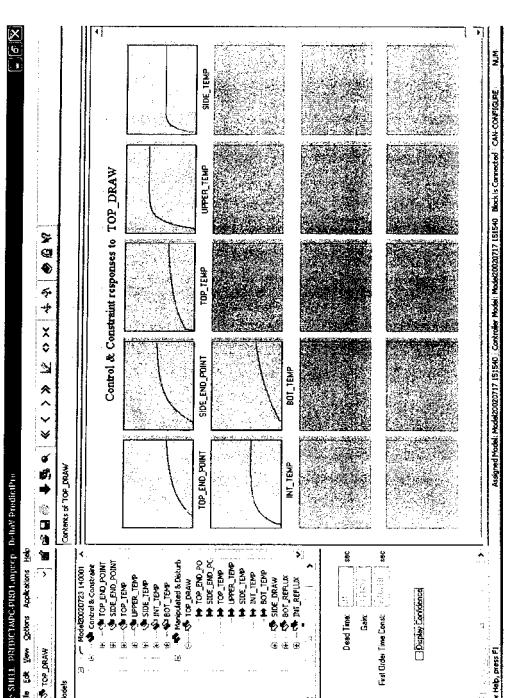
【図8】



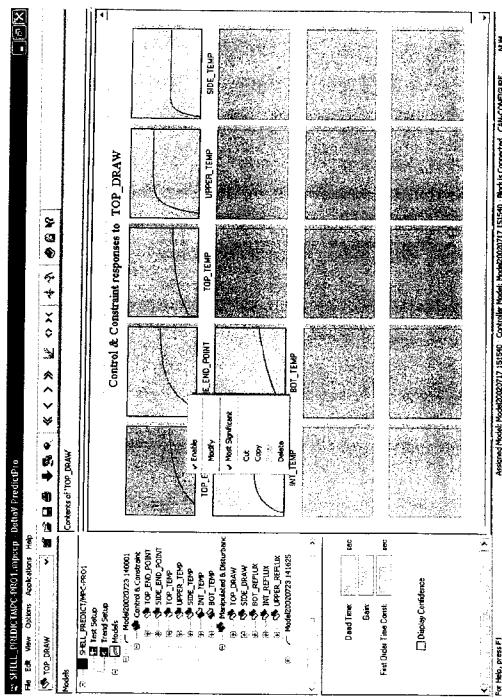
【図9】



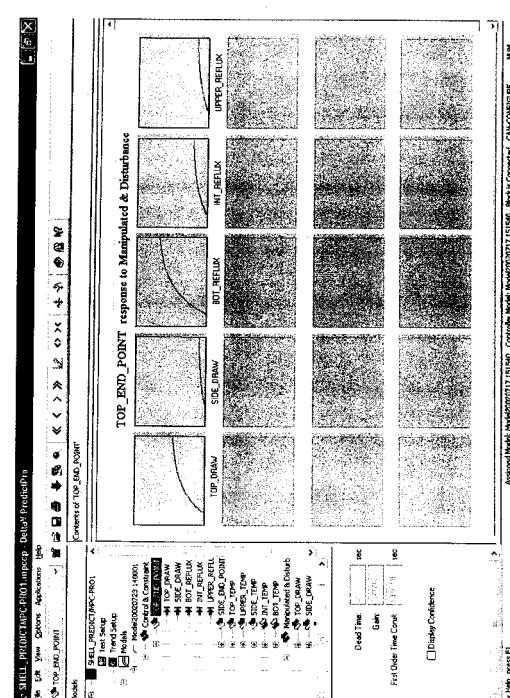
【図10】



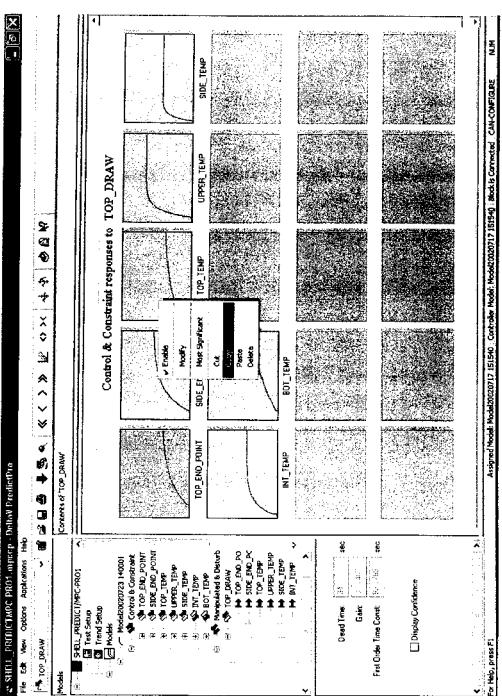
【 図 11 】



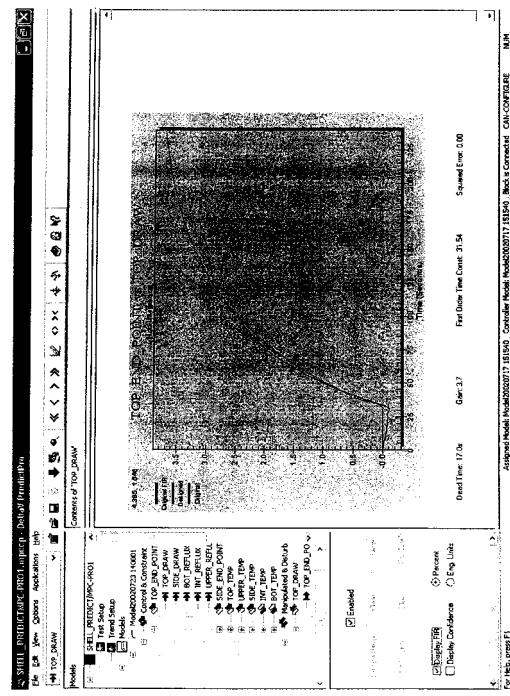
【図12】



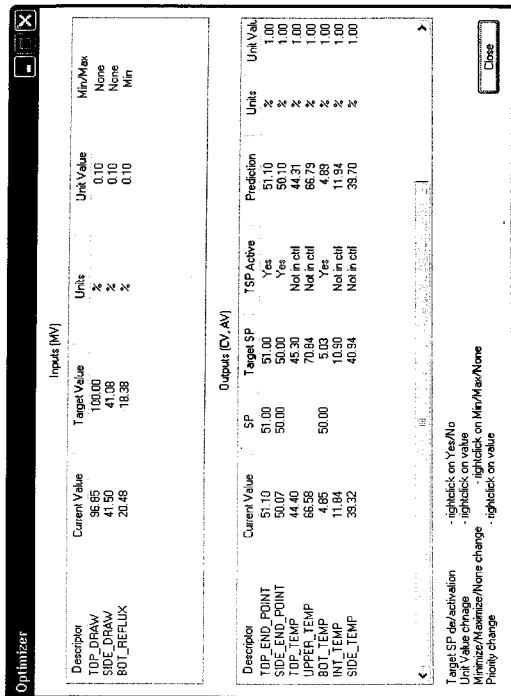
【 図 1 3 】



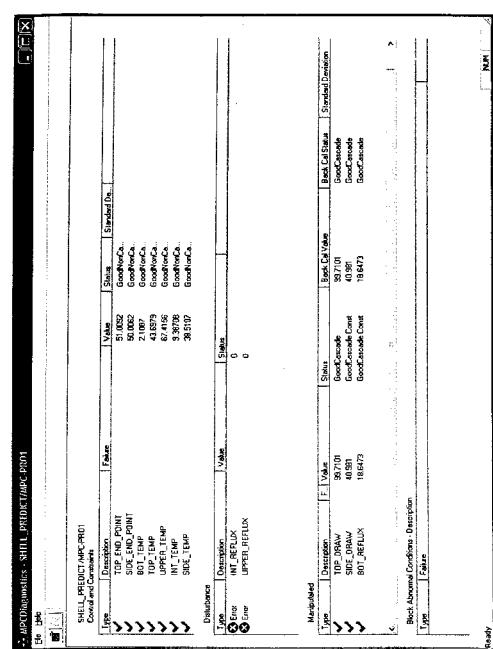
【図14】



【図15】



【図16】



---

フロントページの続き

(72)発明者 ウォジュスズニス， ウィルヘルム ケー。  
アメリカ合衆国 78681 テキサス ラウンド ロック ヒルサイド ドライブ 17004  
(72)発明者 ブレビンス， テレンス エル。  
アメリカ合衆国 78681 テキサス ラウンド ロック カーメル ドライブ 3801  
(72)発明者 ニクソン， マーク  
アメリカ合衆国 78681 テキサス ラウンド ロック ブラックジャック ドライブ 15  
03

審査官 佐藤 彰洋

(56)参考文献 特開2001-216003(JP,A)  
特開平07-134603(JP,A)  
特開平04-076702(JP,A)  
特開平06-187004(JP,A)  
特開平10-268906(JP,A)  
特表平11-504454(JP,A)  
特開平11-119806(JP,A)  
特開2002-091506(JP,A)  
特表平09-506986(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 05 B 13 / 02  
G 05 B 13 / 04  
G 06 F 19 / 00  
G 06 Q 50 / 00