

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5190161号
(P5190161)

(45) 発行日 平成25年4月24日(2013.4.24)

(24) 登録日 平成25年2月1日(2013.2.1)

(51) Int.Cl.	F 1
G05B 13/04	(2006.01) G05B 13/04
G05B 13/02	(2006.01) G05B 13/02 L
G05B 17/02	(2006.01) G05B 17/02
G05B 23/02	(2006.01) G05B 23/02 302V

請求項の数 10 外国語出願 (全 27 頁)

(21) 出願番号	特願2000-344423 (P2000-344423)
(22) 出願日	平成12年10月4日 (2000.10.4)
(65) 公開番号	特開2001-216003 (P2001-216003A)
(43) 公開日	平成13年8月10日 (2001.8.10)
審査請求日	平成19年9月6日 (2007.9.6)
審判番号	不服2012-7313 (P2012-7313/J1)
審判請求日	平成24年4月20日 (2012.4.20)
(31) 優先権主張番号	09/412078
(32) 優先日	平成11年10月4日 (1999.10.4)
(33) 優先権主張国	米国(US)

(73) 特許権者	512132022 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 テキサス州 78681 ラウンド ロック ウエスト ルイス ヘナ ブルバード 1100 ビルティング 1
(74) 代理人	110000556 特許業務法人 有古特許事務所
(74) 復代理人	100145621 弁理士 高田 聰
(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 3 801

最終頁に続く

(54) 【発明の名称】プロセス制御器、多入力／多出力アドバンスド制御エレメントを生成する方法およびアドバンスド制御エレメント生成システム

(57) 【特許請求の範囲】

【請求項 1】

プロセス(58, 174)の多入力／多出力アドバンスド制御を実行する方法であって、
 プロセス制御ルーチン内のアドバンスド制御エレメント(56, 60, 150, 172)を通信可能に接続するステップを有し、前記アドバンスド制御エレメントは、
 プロセス出力(Y₁ - Y₃)からプロセス出力信号を受信する複数の制御入力(I_{N1} - I_{N3})と、
 プロセス入力(X₁ - X₃)へプロセス制御信号を供給する複数の制御出力(O_{UT1} - O_{UT3})と、

通信接続時において未生成の状態である多入力／多出力制御論理(102)と、
 を備え、
 前記方法は、前記アドバンスド制御エレメント(56, 60, 150, 172)を用いて、テストプロシージャを実行することによりプロセス応答データを特定するステップを有し、前記テストプロシージャは、

前記アドバンスド制御エレメント(56, 60, 150, 172)内の波形ジエネレータ(101)を用いて、前記複数のプロセス入力(X₁ - X₃)のそれぞれに励起波形を提供する手続きと、

前記アドバンスド制御エレメント(56, 60, 150, 172)を用いて、前記複数のプロセス出力(Y₁ - Y₃)のそれぞれの応答を表す前

記プロセス応答データを収集する手続きと、を含み、

前記方法は、

前記収集されたプロセス応答データから、前記プロセスモデル(104, 154, 178)が前記複数の制御出力(O_{UT1} - O_{UT2})によって前記複数のプロセス入力(X₁ - X₃)に供給される前記制御信号の変化に対する前記複数のプロセス出力(Y₁ - Y₃)の予測応答をモデリングすることにより、プロセスモデル(104, 154, 178)を作成するステップと、

前記プロセスモデル(104, 154, 178)から、前記多入力 / 多出力制御論理(102)を生成するのに用いることが可能なアドバンスド制御論理パラメータ(103)を作成するステップと、

前記アドバンスド制御論理パラメータ(103)を用いて前記アドバンスド制御エレメント(56, 60, 150, 172)の前記多入力 / 多出力制御論理(102)を生成するステップと、

前記アドバンスド制御エレメント(56, 60, 150, 172)において生成された前記多入力 / 多出力制御論理(102)を用いて、前記アドバンスド制御論理パラメータ(103)に従って前記プロセス(58, 174)の多入力 / 多出力制御を実行するステップと、を有する多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 2】

前記プロセス制御ルーチン内で前記アドバンスド制御エレメントを通信可能に接続するステップは、第1の機能ブロック(66)に前記複数の制御入力のうちのひとつ(IN₁)を通信可能に接続し、第2の機能ブロック(62, 64)に前記複数の制御出力のうちのひとつ(O_{UT1}, O_{UT2})を通信可能に接続するステップを有する、請求項1記載の多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 3】

前記第2の機能ブロックに前記複数の制御出力のひとつを通信可能に接続するステップは、制御機能ブロック(64)に前記複数の制御出力のうちのひとつ(O_{UT2})を通信可能に接続して前記プロセス制御ルーチン内で制御ループの制御を与えるステップを有する、請求項2記載の多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 4】

前記第2の機能ブロックに前記複数の制御出力のうちのひとつを通信可能に接続するステップは、ある装置と関連する出力機能ブロック(62)に前記複数の制御出力のうちのひとつ(O_{UT1})を通信可能に接続して前記装置の制御を与えるステップを有する、請求項2記載の多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 5】

前記アドバンスド制御エレメント(56, 60, 150, 172)内の波形ジェネレータ(101)を用いて、前記複数のプロセス入力(X₁ - X₃)のそれぞれに励起波形を提供する手続きは、多入力 / 多出力の前記アドバンスド制御エレメント(56, 60, 150, 172)内の前記波形ジェネレータを用いて、前記励起波形として一連の擬似ランダム長パルスを発生させるステップを有する、請求項1記載の多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 6】

前記プロセスモデル(154, 178)からプロセスシミュレーションエレメント(152, 176)を生成するステップと、

前記アドバンスド制御エレメント(150, 172)に前記プロセスシミュレーションエレメント(152, 176)を通信可能に接続するステップと、

生成された前記アドバンスド制御エレメント(150, 172)の前記多入力 / 多出力制御論理(102)を用いて前記プロセスシミュレーションエレメント(152, 176)を制御するステップと、をさらに有する、請求項1記載の多入力 / 多出力アドバンスド制御を実行する方法。

【請求項 7】

10

20

30

40

50

プロセスシミュレーションエレメント(152)を生成するステップは、前記プロセスモデル(154)を変更するステップと、前記変更されたプロセスモデル(156)を用いて前記プロセスシミュレーションエレメント(152)を生成するステップとを有する、請求項6記載の多入力／多出力アドバンスド制御を実行する方法。

【請求項8】

生成された前記アドバンスド制御エレメント(150, 172)の前記多入力／多出力制御論理(102)が前記プロセスシミュレーションエレメント(152, 176)を制御する間に前記生成された前記アドバンスド制御エレメント(150, 172)の前記多入力／多出力制御論理(102)を用いて前記プロセス(58, 174)を制御するステップをさらに有する、請求項6記載の多入力／多出力アドバンスド制御を実行する方法。 10

【請求項9】

前記プロセス(58, 174)のプロセス出力(Y₁ - Y₃)を前記プロセスシミュレーションエレメント(152, 176)のシミュレーション出力(Y_{1_SIM} - Y_{3_SIM})と比較して、プロセス(58, 174)と、前記アドバンスド制御論理パラメータ(103)がそれから生成されたプロセスモデル(104, 154, 178)との間での不一致を測定するステップをさらに有する、請求項8記載の多入力／多出力アドバンスド制御を実行する方法。

【請求項10】

前記プロセス(58, 174)を制御する際に用いられる前記アドバンスド制御エレメント(150, 172)の前記複数の制御入力(I_{N1} - I_{N3})の1つに前記プロセスシミュレーションエレメント(152, 176)のシミュレーション出力(Y_{1_SIM} - Y_{3_SIM})を与えるステップをさらに有する、請求項8記載の多入力／多出力アドバンスド制御を実行する方法。 20

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は一般にプロセス制御システムに関し、より特定的には、プロセス制御システムにおいて、モデル予測制御ブロックおよびニューラルネットワーク制御ブロックのようなアドバンスド制御ブロックを用いることに関する。

【0002】

【従来の技術および発明が解決すべき課題】

化学プロセス、石油プロセス、または他のプロセスで用いられるシステムのように、分散プロセス制御システムまたはスケーラブルプロセス制御システムのようなプロセス制御システムは、典型的には、アナログバス、デジタルバスまたは組み合わされたアナログ／デジタルバスを介して、少なくともひとつのホストワークステーションまたはオペレータワークステーションとひとつ以上のフィールド装置とに互いに通信可能に結合されたひとつ以上のプロセスコントローラを含む。例えば、弁、弁ポジショナ、スイッチおよび送信器（例えば、温度センサ、圧力センサ、および流量センサ）であってよいフィールド装置は、弁の開閉およびプロセスパラメータの測定のようなプロセス内の機能を実行する。

プロセスコントローラは、フィールド装置が測定を行ったプロセス測定値、および／またはフィールド装置に関する他の情報を示す信号を受信し、この情報を用いて制御ルーチンを実行し、次いで制御信号を生成し、この制御信号は、プロセスの動作を制御するべく、バスによってフィールド装置に送られる。フィールド装置およびコントローラからの情報は典型的には、オペレータワークステーションにより実行されるひとつ以上のアプリケーションに利用可能であり、これにより、オペレータは、プロセスの現状を見たり、プロセスの動作を修正したりするような、プロセスに対して所望の機能を実行することができる。

【0003】

今まで、従来のフィールド装置は、アナログバスまたはアナログ線を介して、プロセスコントローラとの間でアナログ（例えば4ミリアンペアから20ミリアンペア）信号を送受

10

20

30

40

50

信するのに用いられていた。これらの4ミリアンペアから20ミリアンペアの信号は、装置が測定した測定値または装置の動作を制御するために必要なコントローラが生成した制御信号を示すという点で事実上制限があった。しかしながら、過去10年ぐらいで、マイクロプロセッサおよびメモリを含むスマート(賢い)フィールド装置はプロセス制御産業において普及した。プロセス内で一次的な機能を実行するだけでなく、スマートフィールド装置は、装置に関するデータを記憶し、デジタルフォーマットでまたは組み合わされたデジタルおよびアナログフォーマットでコントローラおよび/または他の装置と通信を行い、自己較正、識別、診断などの二次的なタスクを行う。H A R T (登録商標)、P R O F I B U S (登録商標)、W O R D F I P (登録商標)、D e v i c e - N e t (登録商標)、およびC A N プロトコルのような多くの標準的通信プロトコルおよびオープンスマート装置通信プロトコルは、様々な製造者が製造したスマートフィールド装置を同じプロセス制御ネットワーク内で一緒に用いることが出来るように開発された。10

【0004】

さらに、プロセス制御産業内では、プロセス制御機能を非集中化しようとする動きがある。例えば、ファウンデーション(商標)フィールドバス(以下フィールドバス)プロトコルとして知られているフィールドバスファウンデーションにより広められた全デジタル、2線式バスプロトコルは、様々なフィールド装置内に位置する機能ブロックを用いて、集中化コントローラ内で前に実行された制御動作を実行する。特に、各フィールドバスフィールド装置は、ひとつ以上の機能ブロックを含み実行することが可能であり、この機能ブロックの各々は他の機能ブロック(同じ装置内でまたは異なる装置内で)との間で入出力動作を行い、プロセスパラメータを測定したり、検出したり、比例・積分・微分(P I D)制御ルーチンを実行するように装置を制御したり制御動作を実行したりするような何らかのプロセス制御動作を行う。プロセス制御システム内の様々な機能ブロックは、(例えばバスによって)互いに通信を行いひとつ以上のプロセス制御ループを形成するように構成され、その個々の動作はプロセス中に広がり、従って、分散化(非集中化)される。20

【0005】

プロセスコントローラは典型的には、流量制御ループ、温度制御ループ、圧力制御ループなどのよう、あるプロセスのために規定されるかまたはそこに含まれる多くの異なる制御ループの各々のために、異なるアルゴリズム、サブルーチンまたは制御ループ(これらは全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る。一般的に、このような各制御ループは、アナログ入力(A I)機能ブロックのようなひとつ以上の入力ブロック、比例・積分・微分(P I D)機能ブロックまたはファジィ論理制御機能ブロックのような単一出力制御ブロック、およびアナログ出力(A O)機能ブロックのような単一出力ブロックを含む。これらの制御ループは典型的には、单一入力/单一出力制御を実行する。なぜなら制御ブロックは、弁位置などのような单一のプロセス入力を制御するのに用いられる单一の出力を生成するからである。しかしながら、場合によっては、多くの独立して動作する单一入力/单一出力制御ループを用いることはあまり有効ではない。なぜなら、制御されているプロセス変数は1個を上回るプロセス入力により生じさせられているからであり、事実、各プロセス入力は多くのプロセス出力をもたらしてよい。この例としては、例えば、2本の入力ラインにより充填され、单一の出力ライン(各ラインは異なる弁により制御される)により空にされるタンクを有するプロセスを挙げられるが、この場合は、タンクの温度、圧力、スループットは所望の値になるようにまたはそれに近似するように制御される。上のべたように、タンクのスループット、温度、および圧力の制御は、別個のスループット制御ループ、別個の温度制御ループおよび別個の圧力制御ループを用いて実行されてもよい。しかしながら、この状況では、タンク内の温度を制御すべく入力弁のうちのひとつの設定を変更する際の温度制御ループの動作は、タンク内の圧力を増大させ、これにより、例えば、圧力ループに出力弁を開放させて圧力を下げるようとする。この動作は次いで、スループット制御ループに入力弁のうちのひとつを閉じさせ、これによりその温度をもたらし、温度制御ループに何らかの他の動作を行わせる。この例において理解されるように、单一入力/单一出力制御ループにより、プロセス出力(この場合、304050

スループット、温度および圧力)は、安定した状態に達することなく発振することとなり、これは望ましくない。

【0006】

モデル予測制御または他の種類のアドバンスド制御は、従来、これらの状況において制御を行うのに用いられてきた。一般的に、モデル予測制御は多入力／多出力制御戦略であり、この戦略においては、多くのプロセス入力の各々を変更したことによる多くのプロセス出力の各々に対する影響が測定され、これらの測定された応答はついでプロセスのモデルを作成するのに用いられる。プロセスのモデルは数学的に反転され、ついで、多入力／多出力コントローラとして、プロセス入力に行われた変更に基づいてプロセス出力を制御するのに用いられる。場合によっては、プロセスモデルは、プロセス入力の各々に対するプロセス出力応答曲線を含み、これらの曲線は、プロセス入力の各々に送られる、一連の、例えば、擬似ランダムステップ変更に基づいて作成されてよい。これらの応答曲線は公知の様式でプロセスをモデル化するのに用いることができる。モデル予測制御は当該技術で公知であり、従って、ここでは具体的には述べない。モデル予測制御は、Qin, S, JoeおよびThomas Badgwellによる1996年のAIChE Conferenceの「工業モデル予測制御技術の概要 ("An Overview of Industrial Model Predictive Control Technology")」に記載される。10

【0007】

従来、モデル予測コントローラを作成してプロセス制御ネットワーク内にそのコントローラを配置するには、多大な時間と努力が必要であり、非常に費用がかかった。通常、ある特定のプロセスのためのモデル予測コントローラを作成するために、プロセスエキスパート(典型的には外部のコンサルタント)がプラントに来て、プラントまたはプロセス動作を観察するように雇われていた。モデル予測コントローラのための適切なプロセス入力および出力を選択した後、エキスパートは制御室に座り、オペレータに、選択されたプロセス入力の各々に対し、一連の段状の入力波形を送るように命令し、選択されたプロセス出力の各々に対するこれらの入力の各々の影響を測定するように命令した。プロセスデータを全て収集した後、エキスパートは、収集されたデータをオンラインシステムに送った。そこで、エキスパートは、第1のルーチンを実行させて、プロセスが通常に動作していないかったとか、遮断されたとか、なんらかの他のエラーが存在していてそのために収集されたデータがプロセスの通常動作を表現できなかったときに収集されたデータのような不良データを除く目的で、収集されたデータをふるいにかけた。オンラインシステムはついで、ふるいにかけられたデータを用いて第2のルーチンを実行させ、プロセスのモデルを作成する。その後、プロセスのためのモデル予測コントローラを作成するため、公知の様式でプロセスのモデルは反転されるか用いられた。いったんモデル予測コントローラが作成されると、それはプロセス制御システム内に挿入されなければならなかつたが、これは、制御を行なうために、プロセスエンジニアがもうすでに制御システム内で制御ルーチンをプログラムして、モデル予測コントローラに指定されたコントローラ入力(すなわちプロセス出力)の各々を送り、モデル予測コントローラにコントローラ出力(すなわちプロセス入力)の各々を制御システム内の適切な場所に送らせなければならなかつたということである。プロセス制御ルーチンまたはシステムで用いられるモデル予測コントローラ入力および出力に対して同じ名前を使用する売り手もいるが、場合によっては、プロセス制御システム内で規定される通り、モデル予測コントローラの入力および出力をプロセス出力および入力に一致させることが必要であった。いづれにせよ、プロセス制御システム内にモデル予測コントローラを組み込むステップには多大なプログラミングの努力を必要としていただろう。203040

【0008】

結果、当該技術で公知ではあるが、収集されたデータからプロセスモデルを作成したり、モデル予測コントローラを作成したり、このコントローラをプロセス内に組み込んだりするには時間がかかり、エキスパートの入力が必要であり、非常に費用がかかるものである。事実、プロセスのための単一のモデル予測コントローラを作成するには数ヶ月の月日が50

かかり、数千万円ぐらいかかるだろう。プロセスオペレータにとって残念なことに、プロセス機器のエージング（経年変化）により生じる変化のようなプロセスの変化は、プロセスにとって、作成されたモデル予測コントローラが時代遅れであり、一致しないものであることを強いることになり、これは、別のモデル予測コントローラを作成するために再度プロセス全体を実行させなければならないことを意味する。

【0009】

さらに、モデル予測コントローラは典型的にはオンラインシステムにより作成されたので、このコントローラは一般的に、制御システムにより実行される単一のループまたは他の制御ルーチンと同じ方法でプロセス制御システム内に統合されなかった。従って、ユーザまたはオペレータがモデル予測コントローラの状態および動作を見れるように特別な図形を作成する必要があった。この理由のために、コントローラ内で制御ブロックまたは制御ループの動作と統合されるプロセス制御表示機構を有するフィッシャー・ローズマウント・システムズ (Fisher Rosemount Systems Inc) が販売しているデルタV (商標) のようなプロセス制御システム内にモデル予測コントローラを組み込むことは難しかった。事実、デルタVシステムは、ユーザに、プロセスの動作を表示するエンジニアのビュー、オペレータのビューなどの様々なビューを与える。いったんセットアップされると、これらのビューは、例えば、プロセスコントローラ内で実行される機能ブロックの動作により自動的に更新される。しかしながら、異なるシステムによりオフラインで設計されたモデル予測コントローラのためにビューまたは他の情報スクリーンを加えるには、デルタVシステムが用いるフォーマットとは典型的には異なるフォーマットで、特別の図形表示を作成しなければならなかった。

10

【0010】

これらの問題はモデル予測コントローラに対して存在したが、ニューラルネットワークモデル化システムまたはニューラルネットワーク制御システム、多変数ファジィ論理コントローラ、リアルタイムオプティマイザ(optimizer)などのような、他のアドバンスド多入力 / 多出力制御ブロックまたはシステムの開発および使用においても、同じまたは類似の問題が存在する。

20

【0011】

【課題を解決するための手段】

本発明のある態様によれば、アドバンスド (advanced) 制御ブロックは、フィールドバスパラダイム(paradigm)のような制御パラダイムを用いて実行される制御ブロックと統合される方法で、プロセス制御システム内で、モデル予測制御、ニューラルネットワークモデル化またはニューラルネットワーク制御などの多入力 / 多出力制御を実行する。アドバンスド制御ブロックは、それぞれプロセス出力およびプロセス入力に接続されてプロセスを制御するための所望の入力および出力を有する制御ブロックを作成することにより開始されてよい。制御ブロックは、最終的には、例えば完全なモデル予測コントローラを含むものであるが、最初は、データ収集ユニットと、それと関連する波形ジェネレータとを有する。所望であれば、制御ブロックは、チューニングされていないかまたはそうでなければ未成の制御論理を有してよい。なぜなら、この論理はミッシングチューニングパラメータであるか、実行する必要のある行列係数かまたは他の制御パラメータであるからだ。制御ブロックは、もしもアドバンスド制御ブロックがプロセスを制御するのに使用されているとしたら、規定された入力および出力が接続されるだろうというような方法で、これらの入力および出力が制御システム内で通信可能に結合された状態で、プロセス制御システム内に配置される。テストプロシージャの間、制御ブロックは体系的に、プロセスモデルを作成する際に用いられるように特殊設計された波形ジェネレータにより生成された波形を用いて、制御ブロックを出力を介してプロセス入力の各々をアップセットする。ついで、制御ブロック入力を介して、制御ブロックは、プロセス入力の各々に送られた生成された波形の各々に対するプロセス出力の各々の応答に関するデータを収集することを調整する。このデータは、例えば、データヒストリアン (data historian) に送られて記憶されてよい。

30

40

50

【0012】

十分なデータが収集された後、プロセスモデル化プロシージャは実行され、プロセスモデルは、例えば、モデル予測コントローラプロセスモデル生成ルーチンを用いて、収集されたデータから生成される。その後、アドバンスド制御ブロック論理パラメータ決定ルーチンを用いて、プロセスを制御するのに用いられるべき制御論理が必要なパラメータを作成するか生じさせる。制御論理パラメータ、そして必要ならば、プロセスモデルは、次いで、アドバンスド制御論理パラメータおよびプロセスモデルを有するアドバンスド制御ブロックを用いてプロセスを制御できるように、制御ブロックにダウンロードされてアドバンスド制御ブロックの形成を完成させる。

【0013】

10

アドバンスド制御ブロックは、プロセス制御システム内の他の制御ブロックと同じフォーマットでまたは同じプログラミングパラダイムに従って設計することができ、従って、プロセス制御ルーチン内で他のブロック（またはエレメント）により支援される同じ図形ビューを支援することができる。このように、アドバンスド制御ブロックは、ひとり以上のユーザに表示されるべきひとつ以上の図形ビューを有してもよく、アドバンスド制御ブロックの動作の間にこれらのビューにデータを送ってもよい。

【0014】

さらに、プロセスモデル化プロシージャにより生成されたプロセスモデルは、プロセスの動作をシミュレートする、および／または、プロセスおよびアドバンスド制御ブロックの対話をシミュレートするのに用いられてよい。ある場合においては、プロセスシミュレーションブロックは、決定されたプロセスモデルから作成されてよく、このプロセスシミュレーションブロックは、アドバンスド制御ブロックを用いて実際のプロセスを制御する前に、作成されたアドバンスド制御ブロックに通信可能に結合されて、アドバンスド制御ブロックの動作をテストする。別の場合においては、プロセスシミュレーションブロックは、プロセス内のエージングまたは他の変化を反映するために、決定されたプロセスモデルの変更バージョンを用いて作成されてよい。このシミュレーションブロックは、アドバンスド制御ブロックに通信可能に接続されて、プロセスに変更がある状態でアドバンスド制御ブロックの動作をシミュレートし、これによりプロセスモデルの不一致が存在している場合において、アドバンスド制御ブロックの実行を決定する。さらに別の場合においては、プロセスモデルから作成されたシミュレーションブロックは、プロセスと関連して実行されてよく、例えば、実際のプロセス出力のうちのひとつを測定するセンサが故障したとき、アドバンスド制御ブロックのための入力として用いられるべき仮想プロセス出力を作成するのに用いられてよい。シミュレートされたプロセス出力は、実際のプロセス出力と比較されて、プロセスと、アドバンスド制御ブロックを作成するのに用いられるプロセスモデルとの間の不一致、すなわち、プロセス／プロセスモデル不一致の量を決定してよい。

【0015】**【発明の実施の形態】**

さて図1を参照して、プロセス制御システム10は、データヒストリヤン(data historian)12と、各々がディスプレイスクリーン14を有する、ひとつ以上のホストワークステーションまたはコンピュータ13（任意の種類のパソコン、ワークステーションなどであってよい）とに接続されたプロセスコントローラ11を含む。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装置15～22にも接続される。データヒストリヤン12は、データを記憶するための所望の種類のメモリおよび所望の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のうちのひとつとは別個（図1に示されるように）であってもよく、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(Fisher Rosemount Systems, Inc.)により販売されるDeltaV(商標)コントローラであるが、例えば、イーサネット(登録商標)接続または他の所望の通信ネットワークによってホストコンピュータ13と

20

30

40

50

データヒストリアン 12 と通信可能に接続される。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所望のハードウェアおよびソフトウェア、および / またはフィールドバスプロトコル、HART プロトコルのような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装置 15 - 22 にも通信可能に接続される。

【0016】

フィールド装置 15 - 22 は、センサ、弁、送信器、ポジショナなど、任意の種類の装置であってよく、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任意の種類の I/O 装置であってよい。図 1 に示される実施の形態においては、フィールド装置 15 - 18 は、アナログ回線によって 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、フィールド装置 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によって I/O カード 28 と通信する、フィールドバスフィールド装置のようなスマート装置である。一般的に、フィールドバスプロトコルは、フィールド装置を相互接続する 2 線式ループまたはバスに標準化された物理インターフェースを与える全デジタルシリアル双方向通信プロトコルである。フィールドバスプロトコルは、事実、プロセス内のフィールド装置のためにローカルエリアネットワークを与え、これによって、プロセス設備 (facility) 中に分散された位置でプロセス制御機能を実行する (フィールドバスプロトコルにしたがって規定された機能ブロックを用いて) ことができ、これらプロセス制御機能の実行の前後に互いに通信して全体の制御戦略を実行することが出来る。フィールドバスプロトコルは、プロセス制御ネットワークで用いられるために開発された比較的新しい全デジタル通信プロトコルであるが、このプロトコルは当該技術で公知であり、とりわけ、テキサス州、オースティンに本部を置く非営利組織のフィールドバスファウンデーションにより出版され、流通され、そこから入手可能である無数の記事、パンフレットおよび仕様書に詳細に説明されていることは理解されるであろう。従って、フィールドバス通信プロトコルの詳細についてはここでは述べられない。当然、フィールド装置 15 - 22 は、将来において開発される規格またはプロトコルを含め、他の所望の規格またはプロトコルに従う。

【0017】

コントローラ 11 は、1 個以上のプロセス制御ルーチンを実行するかまたは監督し、そこに記憶されるかまたはそれと関連する制御ループをふくんでよく、装置 15 - 22、ホストコンピュータ 13 およびデータヒストリアン 12 と通信を行い所望の方法でプロセスを制御する。尚、本明細書で述べる制御ルーチンまたはエレメントは、所望であれば、様々なコントローラまたは他の装置により実行される一部を有してよい。同様に、プロセス制御システム 10 内で実行されるべき本明細書で述べる制御ルーチンまたはエレメントは、ソフトウェア、ファームウェア、ハードウェアなどを含め、任意の形をとってよい。本発明の目的のために、プロセス制御エレメントは、例えば、コンピュータ読み取り可能な媒体上に記憶されるルーチン、ブロックまたはモジュールを含め、プロセス制御システムの一部または部分であってよい。制御ルーチンは、モジュール、または、サブルーチン、サブルーチンの一部 (コードのラインのような) などのような制御プロシージャの一部であってよく、はしご論理、順次機能チャート、機能ブロック図、もしくは他のソフトウェアプログラミング言語もしくは設計パラダイムを用いてのように、所望のソフトウェアフォーマットで実行されてよい。同様に、制御ルーチンは、例えば、ひとつ以上の EEPROM EEPROM、専用集積回路 (ASIC)、または他のハードウェアエレメントもしくはファームウェアエレメント内にハードエンコード (hard-coded) 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。

【0018】

ある実施の形態においては、コントローラ 11 は、一般に機能ブロックと称されるものを

10

20

30

40

50

用いてある制御戦略を実行する。この戦略においては、各機能ブロックは、全体の制御ルーチンの一部（例えばサブルーチン）であり、他の機能ブロックと（リンクと呼ばれる通信によって）関連して動作して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。機能ブロックは典型的には、送信器、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ものとしての入力機能、P I D制御、ファジィ論理制御などを実行する制御ルーチンと関連するものとしての制御機能、または、弁のような装置の動作を制御する出力機能のうちのひとつを実行してプロセス制御システム10内で何らかの他の物理的な機能を実行する。当然、ハイブリッド機能ブロックおよび他の種類の機能ブロックが存在する。機能ブ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よいが、これは典型的には、これらの機能ブロックが標準的な420ミリアンペアの装置、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装置のために用いられるかもしくはこれらと関連し、またはフィールドバス装置の場合のようにフィールド装置自体内に記憶されてこれにより実行されてもよい。制御システムについてここでは機能ブロック制御戦略を用いて説明しているが、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、順次機能チャートのような他の規約を用いて、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されるか設計されるだろう。

【0019】

図1の拡大されたブロック30に示されるように、コントローラ11は、ルーチン32、34として示される、多くの単一ループ制御ルーチンをふくんでよく、所望であれば、制御ループ36として示されるひとつ以上のアドバンスド制御ループを実行してよい。このようなループ各々は典型的には制御モジュールと称される。単一ループ制御ルーチン32、34はそれぞれ、適切なアナログ入力(A I)機能ブロックおよびアナログ出力(A O)機能ブロックに接続された单一入力／单一出力ファジィ論理制御ブロックおよび单一入力／单一出力P I D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、機能ブロックは、弁のようなプロセス制御装置と関連してもよいし、温度送信器および圧力送信器のような測定装置と関連してもよいし、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。アドバンスド制御ループ36は、多数のA I機能ブロックに通信可能に接続された入力と多数のA O機能ブロックに通信可能に接続された出力とを有するアドバンスド制御ブロック38を含むものとして示されるが、アドバンスド制御ブロック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エレメントに接続されて他の種類の入力を受け取り他の種類の制御出力を与えてもよい。アドバンスド制御ブロック38は、制御信号を2個以上のプロセス入力を与えることにより2個以上のプロセス出力を制御するのに用いられる何らかの種類の多入力／多出力制御ブロックであってよい。アドバンスド制御ブロック38はここではモデル予測制御(M P C)ブロックとして説明されているが、アドバンスド制御ブロック38は、ニューラルネットワークモデル化ブロックまたはニューラルネットワーク制御ブロック、多変数ファジィ論理制御ブロック、リアルタイムオプティマイザ(optimizer)ブロックなどのような他の多入力／多出力ブロックであってよい。アドバンスド制御ブロック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のうちのひとつもしくはフィールド装置19～22のうちのひとつのような他の処理装置内に位置しこれにより実行可能であることを理解されたい。

【0020】

図1に示されるように、ワークステーション13のうちのひとつはアドバンスド制御ブロック生成ルーチン40を含み、このルーチンは、ここで詳細に述べる方法でアドバンスド制御ブロック38を作成し、ダウンロードし、実行するのに用いられる。アドバンスド制御ブロック生成ルーチン40はワークステーション13内のメモリ内に記憶されてプロセッサにより実行されてよいが、このルーチン（またはその一部）は、所望であれば、さらにまたは代替的に、プロセス制御システム10内の他の装置内に記憶されて実行されてよい。一般に、アドバンスド制御ブロック生成ルーチン40は、アドバンスド制御ブロックを作成してこのアドバンスド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

10

20

30

40

50

ック作成ルーチン 4 2 と、アドバンスド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 4 4 と、プロセスモデルからアドバンスド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、プロセスを制御する際に用いられるためにアドバンスド制御ブロック内に記憶するかダウンロードする制御論理パラメータ作成ルーチン 4 6 とを含む。ルーチン 4 2、4 4、4 6 は、一連の異なるルーチンで構成可能であるが、この一連のルーチンとは、プロセス出力を受け取るように構成される制御入力と制御信号をプロセス入力に与えるように構成される制御出力とを有するアドバンスド制御エレメントを作成する第 1 のルーチンと、プロセス制御ルーチン（所望の構成ルーチンであってよい）内のアドバンスド制御エレメントをユーザが通信可能に接続できるようにする第 2 のルーチンと、アドバンスド制御エレメントを用いてプロセス入力の各々に対し励起波形を与える第 3 のルーチンと、アドバンスド制御エレメントを用いて波形に対するプロセス出力の各々の応答を反映するデータを収集する第 4 のルーチンと、収集されたデータからプロセスモデルを作成する第 5 のルーチンと、プロセスモデルからアドバンスド制御論理パラメータを生じさせる第 6 のルーチンと、アドバンスド制御論理、およびもし必要ならばプロセスモデルをアドバンスド制御エレメント内に配置してアドバンスド制御エレメントがプロセスを制御することができるようとする第 7 のルーチンとである。

【 0 0 2 1 】

さて図 2 を参照して、フロー チャート 5 0 は、図 1 のプロセス制御システム 1 0 のようなプロセス制御システム内の、アドバンスド制御ブロック、特に MPC 制御ブロックを作成して用いるステップを示す。図 2 のフロー チャート 5 0 は MPC ブロックまたはモジュールの作成を示しているが、ニューラルネットワークモデル化ブロックまたはニューラルネットワーク制御ブロック、多変数ファジィ論理制御ブロックなどのように多入力 / 多出力制御ブロックのような他のアドバンスド制御ブロックを作成して用いるために同じステップまたは類似のステップを実行することができる。

【 0 0 2 2 】

まず、時間 5 2 で、MPC プロシージャを実行する事によりプロセス制御システム 1 0 内の制御を向上させるかまたは与えるべく決定が行われる。この決定は、プロセス制御システム 1 0 が最初にセットアップされるときかまたは、単一ループ制御ルーチンのような他の制御ルーチンが不十分な制御を与えると発見された後になって行われる。時間 5 2 で、オペレータまたは他のユーザが MPC ブロック生成ルーチン 4 0 を実行して、プロセス制御システム内で MPC モジュールまたは制御ループを作成するステップを開始する。このプロセスの一部として、オペレータは、設計されている MPC ブロックの出力が接続されるべきプロセス入力を選択するとともに、設計されている MPC ブロックの入力が接続されるべきプロセス出力を選択する。MPC ブロックが有する入力および出力の個数は任意であるが、各 MPC ブロックは一般に、3 種の入力を有し、これらは、あるセットポイント（または設定範囲内）で維持されるべきプロセス変数またはプロセスパラメータである制御されたパラメータ入力、プロセスと関連する例えれば物理的限界に基づいたある特定の限界値または範囲に制約されるプロセス変数であり、MPC ブロックがその入力を制約された範囲または限界値の範囲から外れるように強制してはならない制約された入力、および、変更される場合、制御されたパラメータに対し変更を生じさせるものとして知られているプロセス入力のような他のプロセス変数であるプロセス妨害パラメータ入力である。MPC ブロックはプロセス妨害パラメータ入力を用いて、制御されたパラメータ（すなわち制御されたプロセス出力）に対する変化を予知し、これらの変化の影響をそれらが生じる前に制限する。MPC ブロックがこれらのエレメントを効果的に制御することができるようとするための制御されている装置または他のプロセスエレメントからのフィードバックのような他の入力も MPC ブロックに与えられてよい。同様に、MPC ブロックの出力は、制御ループ入力、装置制御入力などを含む所望のプロセス変数または他のプロセス入力を制御するために接続されてよい。MPC ブロックを他の制御エレメントに接続することにより生じさせられるルーチンはここでは MPC モジュールと称される。ユーザは M

10

20

30

40

50

P C 機能ブロックを作成してよいが、ユーザは、機能ブロックのライブラリのようなメモリからの初期機能ブロックを得て、この機能ブロックを用いるかまたはプロセス制御システムで用いられるこの機能ブロックのインスタンスを作成してよい。同様に、ユーザまたは他のプロバイダは、他の所望の方法で機能ブロックまたは他の制御エレメントを与えてよい。

【 0 0 2 3 】

ステップ 5 4 で、オペレータは、プロセス制御システム内で通信可能に接続された特定の入力および出力を備えた M P C ブロック（モデル予測制御を与えるのに必要な情報をまだ全部有しているわけではない）を有する M P C モジュールを作成し、このブロックまたはモジュールを、M P C モジュールを実行する適切なコントローラまたは他の装置にダウンロードする。このプロセスの一部として、オペレータは、M P C ブロックの出力を適切なプロセス入力に通信可能に結合するとともに M P C ブロックの入力を適切なプロセス出力に通信可能に結合することにより、M P C ブロックを実行するように、プロセス制御システム 1 0 を構成する。

10

【 0 0 2 4 】

図 3 を参照して、M P C ブロック 5 6 はプロセス 5 8 に接続されるものとして示される。M P C ブロック 5 6 は、3 個の入力 I N 1 - I N 3 と 3 個の出力 O U T 1 - O U T 3 とを有する 3 × 3 制御ブロックであり、プロセス 5 8 は入力 X 1 - X 5 と出力 Y 1 - Y 6 とを含む。当然、M P C ブロック 5 6 およびプロセス 5 8 が含む入力および出力の個数は任意である。M P C ブロック 5 6 は一般に正方形ブロック、すなわち、同じ個数の入力および出力を有してよいが、この構成は必要でなく、M P C ブロック 5 6 の有する入力および出力の数は異なっていてよい。図 3 に示されるように、オペレータは、プロセス出力 Y 1 - Y 3 をそれぞれ M P C ブロック入力 I N 1 - I N 3 に通信可能に接続し、M P C ブロック出力 O U T 1 - O U T 3 をそれぞれプロセス入力 X 1 - X 3 に通信可能に接続する。当然、プロセス 5 8 の入力および出力のうちのいづれかは、プロセス制御システム 1 0 と関連する他の制御ルーチン内の他のエレメントに接続されてよい（図 3 のプロセス入力および出力に接続された点線により示されるように）。一般に、制御入力をプロセス 5 8 に与えている（プロセス入力 X 1 - X 3 に接続された点線により示されるように）M P C ブロック 5 6 および他のブロックは、なんらかのスイッチを介して接続され、これらのスイッチは図 3 にボックス 5 9 で示される。スイッチ 5 9 は、ハードウェアスイッチまたはソフトウェアスイッチであってよいが、所望であれば、フィールドバス機能ブロックのような機能ブロックの異なる入力に異なる制御入力信号を送ることにより与えられてよく、M P C ブロック 5 6 からの制御信号と P I D 機能ブロックからの異なる機能ブロックからの制御信号とを受け取る機能ブロックのモードに基づいて、これらの信号の間で選択を行うことができる。

20

【 0 0 2 5 】

当然、オペレータは所望の方法で、M P C ブロック 5 6 をプロセス 5 8 に接続可能であり、一般に、オペレータがプロセス制御システム 1 0 内の単一制御ルーチンのような他の制御ループを作成するのに用いるのと同じ制御構成プログラムまたは設計プログラムを用いるだろう。例えば、オペレータは、所望の図形プログラミングルーチンを用いて、M P C ブロック 5 6 とプロセス入力出力との間の接続を指定してもよい。このようにして、M P C ブロック 5 6 は、他の制御ブロック、エレメント、またはルーチンと同じ方法で支援されてよく、これは、M P C ブロック 5 6 の構成および接続と、システム 1 0 内の他のブロックの構成、接続および支援と同じような制御システム 1 0 内のそのブロックの支援を行なう。ある実施の形態においては、制御システム 1 0 内の他のブロックと同様に M P C ブロック 5 6 は、フィールドバス機能ブロックと同じであるかまたはそれと類似であるように設計された機能ブロックである。この実施の形態においては、M P C ブロック 5 6 は、フィールドバスプロトコルにおいて特定されるか与えられた通り、同じまたは類似の種類の入力、出力などを有してよく、例えば、フィールドバスプロトコルにより特定されたものと同じであるかまたは類似の通信リンクを用いて、コントローラ 1 1 により実行可能である。プロセス制御ルーチンおよびそのエレメントを図形で作成する方法は、「プロセ

30

40

50

ス制御環境を構成するためのシステム "System for Configuring a Process Control Environment" と題されたドゥーブラの米国特許第5、838、563号に記載され、この特許を本明細書において引用により援用する。当然、他の種類の機能ブロックを用いるかまたはプロセス制御構成パラダイム内の他のルーチン、サブルーチンまたは制御エレメントを用いるものを含め、他の制御ループまたは制御モジュール設計戦略も同様に用いられてよい。

【0026】

フィールドバス機能ブロックパラダイムが与えるブロックのような、機能ブロックの相互接続に基づいた制御システムを用いるとき、MPCブロック56はプロセス制御ルーチン内の他の機能ブロックに直接接続可能である。例えば、MPCブロック56は、MPCブロック56の制御出力を、制御されている装置と関連する出力ブロック（AOブロックのような）に直接接続することにより弁などの制御装置に接続されてよい。同様に、MPCブロック56は、他の制御機能ブロックの入力のような、他の制御ループ内の機能ブロックに制御信号を与えて、これらの制御ループの動作を予知するかオーバーライドしてもよい。

【0027】

図4A、4Bは、例えば、フィールドバスタイプのMPC機能ブロック60を示し、これは、プロセス制御システム内の他のフィールドバスタイプの機能ブロックに接続されて、単一ループ制御ルーチンにより実行される既存の多変数戦略を調整する。特に、MPC機能ブロック60は、操作されている弁と関連するAOブロック62のRCASIN（リモートカスケード）入力に接続された第一の出力OUT1と、PID機能ブロック64のRCASIN（リモートカスケード）入力に接続された第2の出力OUT2とを有する。さらに、MPCブロック60は、AI機能ブロック66から送られた第一の入力I1（制御されたパラメータ入力）と、AI機能ブロック68（図4B）から送られた第一の入力I2（制御されたパラメータ入力）と、AI機能ブロック70から送られた第3の入力I3（妨害パラメータ入力）とを有する。AI機能ブロック66、68、70は、センサのようなフィールド装置により測定され、送信器または他の装置により制御ルーチンに送信される信号と関連し、これを与えてよい。MPCブロック60は、MPCブロック60により機能ブロック62、64に送られる制御信号の影響を決定する際に用いられるAO機能ブロック62およびPID機能ブロック64のリモートカスケード出力（RCASOOUT）から、入力BKCALIN1およびBKCALIN2（バック較正入力）で、フィードバックを受け取る。AI機能ブロック66の出力はPID機能ブロック72の入力に与えられ、PID機能ブロック72は、AO機能ブロック62のカスケード入力（CASIN）に制御信号を与え、PID機能ブロック72のBKCALIN入力でAO機能ブロック62の出力OUTからのフィードバック信号を受信し、それにより、プロセスの通常の動作の間に、すなわち、MPC動作なしに、操作された値を制御する。同様に、AO機能ブロック68は、その出力（プロセス出力）をPID機能ブロック74の自動入力に送り、PID機能ブロック74は、制御信号をPID機能ブロック64のカスケード入力に与える。PID機能ブロック74は、機能ブロック74のBKCALIN入力でPID機能ブロック64からフィードバック信号も受信する。図4A、4B（図5と同様に）の機能ブロックの入力および出力は、フィールドバスプロトコルの場合と同じように規定され、フィールドバスプロトコルにより与えられた規定および構成に従って動作する。

【0028】

理解されることであるが、機能ブロック66、72、62は、第1の単一ループ制御ルーチンを形成し、機能ブロック68、74、64は第2の単一ループ制御ルーチンを形成し、これら両方は、プロセスの通常動作または自動動作の間に動作されてよい。しかしながら、MPCブロック60は、AO機能ブロック62およびPID機能ブロック64のリモートカスケード入力に制御入力与えることにより、PID機能ブロック64と関連するループの制御だけでなく、AO機能ブロック（および関連の装置）の制御を引き受けてよく、これにより、これらの機能ブロックは（自動モードの代わりに）リモートモードで動作するようになり、従って、自動入力のかわりにリモート入力で制御入力を用いて動作する

10

20

30

40

50

。リモートモードで動作するとき、機能ブロック 62、64 は、それぞれ P I D 機能ブロック 72、74 の入力を無視する。このように、M P C ブロック 60 はブロック 62、64 に接続されてその制御を与えるてもよいが、スイッチオン・オフしてもよい。M P C 機能ブロック 60 により制御されていないときは、ブロック 62、64 は、単一ループ制御戦略に従って、それぞれブロック 72、74 により依然として制御されている。

【 0 0 2 9 】

同様に、図 5 は、プロセス制御ルーチン内で接続されて単一ループルーチンを調整する M P C 機能ブロック 80 を示す。特に、M P C 機能ブロック 80 は、A I 機能ブロック 82、84 からの制御されたパラメータ入力と A I 機能ブロック 86 からの妨害パラメータ入力とを受け取る。M P C 機能ブロック 80 は、操作された値と関連する A O 機能ブロック 90 に制御出力を与え、制御ループ 94 内の P I D 機能ブロック 92 のカスケード入力 (C A S I N) にセットポイント出力を与える。A O 機能ブロック 90 および P I D 機能ブロック 92 は、M P C 機能ブロック 80 のバック (back) 較正入力に、バック較正出力を与える。制御ループ 94 は、A I 機能ブロック 96 も含み、A I 機能ブロック 96 は、P I D 機能ブロック 92 の自動入力に制御パラメータ入力 (すなわちプロセス出力) を与え、P I D 機能ブロック 92 ついで、例えば、異なる弁または装置と関連する A O 機能ブロック 98 に制御出力を与える。A O 機能ブロック 98 は、P I D 機能ブロック 92 のバック較正入力にフィードバックを与える。図 5 の構成において、M P C 機能ブロック 80 は、A O 機能ブロック 90 と関連する弁を直接制御し、そのループのセットポイントを操作することによりループ 94 の動作を制御する。しかしながら、ループ 94 は、M P C 機能ブロック 80 が動作しているときに動作し続ける。そのようなものとして、M P C 機能ブロック 80 は、A O 機能ブロック 98 と関連する装置を間接的に制御するが、制御ループ 94 を直接制御する。当然、M P C ブロックは他の所望の方法でプロセス制御ルーチン内で接続されて、制御装置または他の制御エレメントを直接的にまたは間接的に制御する。さらに、図形プログラミング技術または非図形プログラミング技術を含め、なんらかの技術を用いて制御ルーチンまたはモジュールが生じさせられてよい。

【 0 0 3 0 】

このように、理解されるだろうが、図 3 において M P C ブロック 56 の出力が接続されるプロセス入力 X1-X3 は、既存の制御戦略内で規定された制御ループへの入力、またはプロセスに接続された弁または他の装置への入力を含め、所望のプロセス入力であってよい。同様に、M P C ブロック 56 の入力に接続されたプロセス出力 Y1-Y3 は、弁または他のセンサの出力、A O 機能ブロックまたは A I 機能ブロックの出力、または他の制御エレメントもしくは制御ルーチンの出力を含め、所望のプロセス出力であってよい。

【 0 0 3 1 】

図 2 のステップ 54 を再び参照して、いったん、オペレータが、それぞれ所望のプロセス出力および入力に接続された入力および出力を有する初期の M P C ブロックを含む制御モジュールを作成すると、初期の M P C ブロックを有する制御モジュールは、コントローラ 11 またはワークステーション 13 のうちのひとつのような適切な装置にダウンロードされて、実行される。つぎに、ステップ 99 で、オペレータは、初期 M P C ブロックに、公知の方法でプロセスを励起し始めるように命令し、プロセスが励起されている間にプロセス入力データおよびプロセス出力データを収集するように命令する。

【 0 0 3 2 】

図 3 に示されるように、初期 M P C ブロック 56 は、データ収集ユニット 100 と、波形ジェネレータ 101 と、汎用制御論理 102 と、制御パラメータ 103 およびプロセスモデル 104 を記憶するための記憶装置とを含む。汎用論理 102 は、例えば、ある特定のインスタンスにおいて動作するか制御を実行することができるようには係数または他の制御パラメータを必要とする汎用 M P C ルーチンであってよい。場合によっては、汎用論理 102 は、プロセスを制御するために、制御されているそのプロセスのためのプロセスモデルを必要としていてよい。例えば、コントローラ 11 にダウンロードされた後、初期の M P C ブロック 56 は、M P C 作成ルーチン 42 によって、M P C ブロック 56 の作成

10

20

30

40

50

の次のフェーズを開始するように命令され、このフェーズにおいては、データは、プロセスモデルを作成する際に用いられるプロセス出力の各々のために収集される。特に、オペレータによりそうするように命令されたとき（または他の所望の時間で）、MPC ブロック 56 の波形ジェネレータ 101 は、その出力 OUT1 OUT3 で一連の波形を生成し始め、これにより、プロセス入力 X1 X3 の各々に励起波形を与える。所望であれば、これらの波形は、ユーザワークステーション 13 内のソフトウェアにより波形ジェネレータ 101 に与えられてよいが、好ましくは、波形ジェネレータ 101 により作成される。波形ジェネレータ 101 により生成された波形は好ましくは、プロセスの通常動作の間に予測される異なる範囲の入力にわたってプロセスが動作するように設計される。MPC 制御ルーチンのためのプロセスモデルを作成するために、波形ジェネレータ 101 は、プロセス入力 X1 X3 の各々に、一連の異なる組のパルスを送ってよく、パルスは組ごとに同じ振幅を有するが擬似ランダム長を有し、異なる組のパルスは異なる振幅を有する。このような一連の組のパルスは、異なるプロセス入力 X1 X3 の各々のために作成され、ついで、1 個ずつ、順次これらの入力の各々に送られてよい。この時間において、MPC ブロック 56 内のデータ収集ユニット 100 は、波形ジェネレータ 101 により生成された波形の各々に対するプロセス出力 Y1 Y3 の応答を示すデータを収集するかまたはこのデータの収集を調整し、生成されている励起波形に関するデータを収集するかまたは収集を調整してもよい。このデータは、MPC ブロック 56 内に記憶されてよいが、このましくは、データヒストリアン 12 に自動的に送られて記憶され、および / またはワークステーション 13 に送られて、ここでこのデータは、ディスプレイスクリーン 14 上に表示されてよい。10 20

【0033】

このように、なんらかのアドバンスド制御論理（まだ完全には生成されていない）を用いてプロセス 58 を制御しようとするかわりに、MPC ブロック 56 はまず、プロセス 58 に一組の励起波形を与え、これらの励起波形に対するプロセス 58 の応答を測定する。当然、波形ジェネレータ 101 により生成された励起波形は、アドバンスド制御ルーチンのための作成制御論理パラメータにとって有用なプロセスモデルを作成するために生じさせられた所望の波形であってよい。この例においては、波形ジェネレータ 101 は、モデル予測コントローラのためのプロセスモデルを作成するのに有用であると知られている一組の波形を生成し、これらの波形は、この目的のために、現在知られているかまたは将来において開発されるなんらかの形をとってよい。モデル予測制御のためのプロセスモデルを作成するべくデータを収集する目的でプロセスを励起するのに用いられる波形は周知であるので、ここでは詳細に述べない。同様に、ニューラルネットワーク、多変数ファジイ論理などの制御ルーチンのような他のアドバンスド制御（モデル化を含む）ルーチンのためのプロセスモデルを作成する際に用いられる波形ジェネレータ 101 により、他の種類のまたは所望の種類の波形が生成されてよい。30

【0034】

尚、波形ジェネレータ 101 は、所望の形をとってよく、例えば、ハードウェア、ソフトウェアまたはその両方の組み合わせにおいて実行されてよい。ソフトウェアで実行される場合、波形ジェネレータ 101 は、所望の波形を生成するのに用いられるアルゴリズムを記憶してよいし、生成されるべき波形のデジタル表現を記憶してよいし、または他のルーチンまたは記憶されたデータを用いてそのような波形を作成してもよい。ハードウェアで実行されるときは、波形ジェネレータ 101 は、例えば、発振器または矩形波ジェネレータの形をとってもよい。所望であれば、オペレータは、プロセスのほぼ正確な応答時間、プロセス入力に送られるべき波形の振幅のステップサイズなどのような、波形を設計するのに必要なパラメータを入力するように頼まれるかもしれない。MPC ブロック 56 が最初に作成されるかまたはオペレータが MPC ブロック 56 にプロセスをアップセットするか励起し、プロセスデータを収集するのを開始するように命令するときに、この情報を得るためにオペレータは促されてよい。好ましい実施の形態において、データ収集ユニット 100 は、確実に完全かつ正確なプロセスモデルが作成されるように、オペレータが入40 50

力する応答時間の3倍かまたは5倍の時間において励起波形の各々に応答してデータを収集する（またはデータの収集を確実にする）。しかしながら、データが集められる時間量は任意である。

【0035】

いづれにせよ、MPC ブロック 56 は好ましくは、波形ジェネレータ 101 がプロセス入力 X1 X3 の各々に対し必要な励起波形を全て送り終えてしまうまで、そしてデータ収集ユニット 100 がプロセス出力 Y1 Y3 のためのデータを収集してしまうまで、動作する。当然、MPC ブロック 56 の動作は、このデータ収集プロセスの間に所望であればまたは必要であれば中断されてもよい。

【0036】

図 6 を参照して、制御論理生成ルーチン 40 によりディスプレイ 14 のうちのひとつの上にオペレータに提示可能なスクリーンディスプレイ 118 により、オペレータはアドバンスド制御ブロックを生成する異なるステップを実行できる。特に、スクリーンディスプレイ 118 は、データディスプレイ領域 120 と、アドバンスド制御ブロック生成ルーチン 40 の様々な部分を開始するのに用いられる 3 個のボタン 122、123、124 を含む。開始テストボタン 122 により、オペレータは、初期の MPC ブロック 56 にプロセス 58 に励起信号を送らせて、データヒストリアン 12 に対し送る入力データおよび出力データを収集させる。ボタン 122 は、例えば、励起ルーチンを実行するために残っている時間、すなわち MPC ブロック 56 が励起波形の全てを生成してこれらの波形に応答して生成されたプロセスデータを収集するのにかかる時間を示す。ボタン 122 を押す前に、オペレータは、プロセスが入力に応答するのにかかる典型的な時間を示す応答時間を入力して、励起波形を生成するために MPC ブロック 56 により用いられるステップサイズを示すか指定してよく、このデータは MPC ブロック 56 の波形ジェネレータ 101 に与えられてよい。ボタン 122 を押した後、MPC ブロック 56 により収集されたデータはデータディスプレイ領域 120 上で表示されてもよく、所望であれば、ユーザは、プロセスモデルを作成するのに用いられてはいけないデータにフラグをつけてもよい。尚、データ収集ユニット 100 は、確実にこのデータをデータヒストリアン 12 または他の記憶装置に送られ記憶されることによりデータを収集してもよい。

【0037】

次に、図 2 に示されるように、ステップ 125 で、オペレータは、あるポイントで、プロセスモデル化ルーチン 44 を実行することにより MPC ブロックを作成する次のフェーズを実行することを決定してよく、プロセスモデル化ルーチン 44 は、データヒストリアン 12 からの収集されたデータにアクセスして、公知のプロセスモデル生成ルーチンを実行して、収集されたデータからプロセスモデルを作成する。一般的に、オペレータは、図 6 のスクリーンディスプレイ上の生成制御ボタン 123 を選択することでこのフェーズを開始してよい。

【0038】

所望であれば、プロセス制御ルーチン 44 は、収集されたデータに関してデータふるい分けプロシージャを実行してもよい。このデータふるい分けプロシージャは、アウトライア (outlier) の収集データ、および他の明らかなエラーデータを検査し、収集されたデータと関連する状態値および限界値のような、収集されたデータと関連する他の値を検査して、そのデータが不良または不適当な状態を有する機能ブロックにより生成されたものかどうか、そのデータが限界値にあったかどうか、そのデータが、機能ブロックまたは他のエレメントが不適当なモードであったときに生成されたものであるかどうか、またはそのデータが、異常なプロセス状況下または望ましくないプロセス状況下でなんらかの方法で生成されたものであるかどうか決定する。例えば、フィールドバス通信プロトコルにおいては、機能ブロックにより生成されたデータは、データヒストリアン 12 内にデータとともに記憶可能であり、データをふるいにかけるために用いられる状態、限界値、およびモード指標も含む。所望であれば、データふるい分けルーチンは、図 6 のデータディスプレイ領域 120 上でオペレータに収集されたデータを示してもよく、このルーチンによ

10

20

30

40

50

つて、例えば、プロセス状況のオペレータの知識に基づいてこのデータを強調表示するかまたは識別することにより、オペレータは、ふるい落とされるすなわち除かれるべきデータに印を付けることができる。このように、プロセス 5 8 がオフラインであったとき、プロセス 5 8 が適切に制御されていなかったとき、プロセス 5 8 が修理中であったとき、プロセス 5 8 内のセンサまたは他の装置が故障していたか取りかえ中などに M P C ブロック 5 6 により収集されたデータは、選別されて、プロセスモデルを作成するために用いられるべきデータから除かれてよい。

【 0 0 3 9 】

図 6 に示されるように、ディスプレイ領域 1 2 0 内のトレンド(trend)は、トレンドプロットとして M P C 入力および出力を含むものとして表示されてよい。プロットは、これらの入力および出力の値に基づいて自動基準化される。また、表示されるプロットの部分の時間フレームは好ましくは、指定された応答時間の 2 倍であるだろう。スライダバー 1 2 6 を用いて、時間ウインドウは、過去 2 日のような以前の時間にさかのぼった値を示すように変更されてもよい。プラント動作上で良好なデータを収集できるように、自動化テスト機能を用いてもよい。開始テストボタン 1 2 2 を選択することにより、M P C ブロックにより操作されるプロセス入力、指定された応答時間にわたって、M P C ブロックにより操作されるプロセス入力は、擬似ランダムシーケンスにおける指定されたステップサイズにより衝突される (bumped)。また、開始テストボタン 1 2 2 が選択されたとき、データディスプレイ上の開始および終了分割バーは、自動化テストの開始および終了に印をつけるように自動的に設定され、M P C ブロック 5 6 は、プロセス 5 8 に、励起波形のような出力信号の擬似ランダムシーケンスを与えることにより、操作された出力の制御を引き受ける。

【 0 0 4 0 】

ディスプレイ領域 1 2 0 内の時間バーまたはデータウインドウは、プロセスモデルを作成するために用いられるデータを選択するのに用いられてもよい。オペレータは、分割バーのうちのひとつを選択して、それを所望の開始時間および終了時間にドラッグ (drag) して、プロセスモデルを識別するために考慮される時間フレームを変更する。開始バーおよび終了バー間の時間の一部が通常のプラント動作を表していないければ、ユーザまたはオペレータは、プロセスモデル識別プロセスの間に無視されるべきデータ値を選択するために時間のこの部分を捕らえる。これに応答して、選択された領域は、より暗い背景色で示され、プロセスモデルを作成するとときに自動的に除かれる。

【 0 0 4 1 】

データをふるいにかけた後、プロセスモデルルーチン 4 4 は、選別されたデータからプロセスモデルを作成する。上で述べたように、プロセスモデル化ルーチン 4 4 は、所望のまたは公知のタイプのプロセスモデル化解析を行って、収集されふるいにかけられたデータからプロセスモデルを作成してよく、作成されたプロセスモデルは、数学アルゴリズム、一連の応答曲線などのような任意の形をとつてよい。

【 0 0 4 2 】

プロセスモデル化ルーチン 4 4 がプロセスモデルを決定する際に問題を有しているならば、その問題の指標は、図 6 に示されるようなユーザディスプレイの状態領域において反映されてもよい。ひとつの問題が示されているが、これは、プロセスモデルを識別するかまたは作成するのに十分なサンプルがないということである。規定された構成のためには最低限 X X X 個のサンプルが必要であるとメッセージがでている。データファイルは X X X 個のサンプルだけを含むだけであるというメッセージがこの問題をオペレータに知らせるために生成されてよい。検出されるおそれのある別の問題としては、プロセス入力について十分な励起が生じなかっただということである。このような問題が生じたとき、この影響に対するメッセージ、T a g X 、T a g Y などの信号タグ名を識別するもの、および励起量に対する最小変更をオペレータに与えることが出来る。

【 0 0 4 3 】

所望の場合、そして、成功したモデルが識別されなかつた状況では、ユーザは、プロセス

10

20

30

40

50

モデル化が行われている期間の時間フレームを変更してもよいし、または、プロセスモデル化ルーチン 44 で用いられているデータが有効であるようにプロセス入力を変更してもよい。識別されるプロセスモデルは、後にアクセスできるように所望のデータベース内に自動的にセーブされてよい。熟練したユーザは、識別されたプロセスモデルを調べるか編集したいと思うだろう。図 6 のスクリーン上のアドバンスドボタン 124 を選択することにより、選択されたモデルおよび現在の MPC 機能ロック構成から MPC コントローラを生成することまたは MPC 制御論理を作成するのに用いられるべき新たなモデルとしての結果として生じるモデルをセーブすることの選択肢を与えられる。生成コントローラオプションが選択されると、ユーザには、編集されている MPC モジュールにおける MPC ロックのために前にセーブされたモデルをユーザが選択し得るダイアログが提示可能である。編集オプションを選択することにより、ユーザには、問題の MPC モジュールのために作成されたモデルのリストが提示可能である。モデルを選択した後、ユーザは、プロセスステップ応答のオーバービューを表示するスクリーンまたは、後に説明する他のスクリーンを見て、プロセスステップ応答を編集して新たなまたは変更されたモジュールを作成する。

【 0044 】

プロセスのあるポイントにおいて、論理パラメータ作成ルーチン 46 は、初期 MPC ブロック 56 の汎用論理 102 が必要なパラメータ（MPC ブロック 56 内の変数内に記憶されるべき）を作成するために実行され、モデル予測制御を実行してよい。これらの制御パラメータは、例えば、MPC 論理のための行列または他の MPC 係数、チューニングパラメータ、ニューラルネットワークパラメータ（ニューラルネットワークのための）、倍率（多変数ファジイ論理のための）または他の所望のパラメータであってよく、通常は、生成されたプロセスモデルに基づいて決定される。論理パラメータ作成ルーチン 46 は、プロセスモデルからパラメータを作成するための所望のプロシージャまたは公知のプロシージャを実行してもよい。一般に、このプロセスは、行列フォーマットでプロセスモデルを反転することを伴う。しかしながら、他の所望の論理パラメータ作成ルーチンも用いることができる。プロセスのために収集されたデータからプロセスモデルを作成し、そのプロセスモデルから MPC パラメータまたは他の制御論理パラメータを生成する詳細は、当該技術で公知であるので、これらのプロシージャはさらにここでは述べない。尚、しかしながら、オペレータが MPC ブロック 56 のための制御論理パラメータの作成に関するなんらかの入力を有してもよい。事実、オペレータは、MPC コントローラを作成するのに典型的には用いられるある変数の値を指定する能力を要求されるかまたは与えられてよい。例えば、オペレータが指定してよいのは、MPC ブロックへの制約された入力の各々のセットポイントおよび限界値、制御変更を行うべき期間の時間フレーム、すなわち、セットポイント軌跡（trajectory）フィルタおよびこのフィルタと関連する時定数、MPC 出力またはプロセス出力の最大の動きまたは最小の動き（速度限界値）、制御されたパラメータのいづれかが統合的に応答するかどうか、MPC 最適ファクタ、MPC 変数または MPC チューニングパラメータ、MPC ブロックの水平線（horizon）、すなわち所望の状態になるように制御するために順方向計算のステップがいくつ実行されるべきであるか、MPC ブロック 56 の入力および出力の各々に対するエンジニアリングユニット範囲、操作された変数目標値のうちのどれが、制約条件のうちのひとつが違反されたときに緩められることを許可されているかまたは実現されないか、MPC ブロック入力および出力の各々の記述および / または名前、設定可能な最適変数の値、MPC ブロックの積極性または頑強性に関する変数の値などである。所望であれば、制御論理生成ルーチン 46 は、これらの変数またはセッティングのいくぶんかまたは全てのデフォルト値を記憶し、これらのデフォルト値を用いて MPC 論理を作成する。しかしながら、オペレータまたは他のユーザは、ユーザディスプレイ 14 を介してこれらのセッティングを変更することができる。

【 0045 】

いづれにせよ、MPC 論理パラメータ作成ルーチン 46 はこの情報、および、MPC 係数のような MPC（または他の）制御論理パラメータを作成するのに必要な他の情報を用い

10

20

30

40

50

て実行される。スクリーンディスプレイ 118 上の生成制御ボタン 123 は、プロセスモデルおよび制御論理パラメータの作成が成功したかどうかを示してよい。

【0046】

MPC 制御論理パラメータが作成された後、図 2 のステップ 128 で、MPC 制御論理パラメータまたは MPC 係数がプロセスシミュレーションブロックを用いてテストされてもよい。このシミュレーションブロックは一般に、プロセスのために作成されたプロセスモデルから作成されてよく、ここで説明されるようにテスト環境内で MPC ブロックに接続されて、作成された MPC 制御論理がプロセスの通常動作の範囲にわたって申し分なく動作するかどうかテストする。MPC 論理が満足いくものでなければ、ステップ 54、99、および 125 のいづれかまたは全てが繰り返されて異なる MPC 制御論理を作成してよい。しかしながら、もしも MPC 制御論理が満足いくものであれば、ステップ 130 で、MPC 制御論理パラメータおよびプロセスモデルは、MPC ブロック 56 にダウンロードされてパラメータ記憶装置 103 およびプロセスモデル記憶装置 104 内に記憶され、プロセス 58 を制御するのに用いられる。このように、MPC 制御論理が必要なパラメータは、MPC ブロック 56 に与えられ、そこに格納され、MPC ブロック 56 は、MPC 制御論理 102 にしたがってプロセス内で動作するかまたは制御を実際に実行するように委託される。当然、所望であれば、必要なパラメータと共に実際の MPC 論理 102 はワークステーション 13 で作成されて、MPC ブロック 16 にダウンロード可能である。

10

【0047】

いったん、コントローラ 11 によりダウンロードされ実行されると、MPC ブロック 56 を有する MPC モジュールまたはループは、制御ルーチン内の他のブロックまたはエレメントと同じ方法で機能を報告することを行ってよい。なぜなら、上で述べたように、MPC ブロック 56 およびこのブロックを含む制御モジュールは、プロセス制御システム 10 内の他の制御ブロックと同じプログラミングパラダイムを用いて設計されるからである。ある実施の形態においては、MPC ブロックまたはモジュールは、それと関連する図形ビューを有してよく、これは、例えば、ワークステーション 13 のうちのひとつのうちのディスプレイスクリーン 14 のうちのひとつによって、ユーザまたはオペレータに表示可能であり、これらのビューは、MPC 制御モジュール内のブロックと関連するデータを予約し、このデータを予め規定されたまたは指定された方法で表示する。

20

【0048】

図 7 を参照して、例えば、図 1 のプロセス制御システム 10 の一部分が示され、ディスプレイ 14A および 14B と装置 15 - 22 とに通信リンク（具体的には示されていない）を介して結合されたコントローラ 11 をふくんでいる。コントローラ 11 内で、MPC モジュール 132 は、一組の AI 機能ブロックから入力を受け取り、一組の AO 機能ブロックに出力を与える APC 機能ブロックを有するものとして示され、单一のループ制御ルーチン 134 は、AO 機能ブロックを制御するために、AI 機能ブロックからの入力を受け取る PID ブロックを含むものとして示される。

30

【0049】

オペレータのビューおよびエンジニアのビューのような、これら 2 個の制御モジュールの動作の異なるビューが、ディスプレイスクリーン 14A および 14B 上で図形で示される。特に、ディスプレイスクリーン 14A 上のエンジニアのビューは、エンジニアがこれらのループに関する情報にアクセスしてこれらのループを操作できるように作成されるループ 134 の図形表現だけでなくループ 132 の動作の図形表現も含む。同様に、ループ 134 の図形表現だけでなくループ 132 の動作の図形表現を有するオペレータのビューがディスプレイスクリーン 14B 上に与えられ、これにより、オペレータは、これらのループに関する情報にアクセスし、これらのループを操作できる。しかしながら、オペレータのビュー内で与えられた情報は、エンジニアのビュー内の情報とは異なっていてよく、ループ 132、134 と対話するようにこれらのビューにより与えられたケイパビリティは異なっていてよい。例えば、オペレータのビューは、オペレータがセットポイントを変更して、制限された機能を実行するこのとができるようにするだけであり、エンジニアのビ

40

50

ビューは、ユーザが、ユーザがループのセットアップに対し変更を加えたり、機能ブロック内のプログラミングに変更を加えることができるようになる。

【0050】

これらの異なるビューは、「プロセス制御プログラミングのための均一制御テンプレート生成システムおよび方法 "Uniform Control Template Generating System and Method for Process Control Programming"」と題されたBlevinsへの米国特許第5,594,858号（本明細書において引用により援用される）で述べられるテンプレートに対して開示されたものと類似の方法で機能ブロックと関連して作成されてよい。しかしながら、これらのブロックを用いて作成されるMPCブロックおよびモジュールは、プロセス制御システム10内の他のブロック、ルーチンまたはエレメントとして同じ種類の図形支援または報告支援を与えることができ、なぜなら、これは、MPCブロックが他の制御ブロックと同じプログラミング戦略を用いてオンラインで作成されたからであるということを理解されたい。この機能により、オペレータ、技術者、エンジニアなどがMPC制御モジュールまたはMPCブロック内で起こっているものを見れるよう特別なプログラミングを与える必要がなくなる。

【0051】

所望であれば、MPCモジュール132は、予め規定されたビューまたはディスプレイによって、ユーザに所望の情報を報告してもよいし、ユーザまたはオペレータが所望の行為を行えるようにしてもよい。例えば、ユーザにはあるスクリーンが与えられ、これは、MPCモジュール132により生成されたまたはそれと関連する警報（アラーム）を示し、制御されたパラメータ、制約されたパラメータ、および妨害パラメータ（制御されたパラメータおよび制約されたパラメータの計画された値を示してよい）のプロットを与え、ユーザまたはオペレータが、例えば、モードパラメータ（MPCモジュール132が例えばフィールドバスプロトコルを用いて作成されるとき）を用いてMPCモジュール132の実行を制御できるようにし、セットポイント、制約条件、および制御された入出力または制約された入出力の値を数値的にまたはバーグラフで示し、MPCセットポイントまたはMPC目標値を変更できるようにし、例えば、入力が不良であるのか、不確実であるのかまたは制限されているかを示すためにMPC入力の状態を示し、または他の所望のデータを示すか所望の機能を行うものである。

【0052】

MPC制御モジュール内のMPCブロックにMPC制御論理をダウンロードすることに加えて、図2のステップ135で、作成された論理を有するMPC論理またはMPCブロックは、例えば、MPCブロックの使用方法をユーザに訓練し、MPCブロックをテストするために、ひとつ以上のシミュレーション環境で用いるためにワークステーションに送られてもよい。このようなシミュレーション環境は、1999年5月6日に出願された「単一コンピュータ上で分散プロセス制御システム機能性を統合すること」（Integrating Distributed Process Control System Functionality on a Single Computer）と題された米国仮出願番号60/132,780に詳細に記載されるシステムを用いて与えられてよいが、この仮出願は、本発明の譲受人に譲渡され、その開示を本明細書において引用により援用される。さて、図8を参照して、シミュレーション構成149は、シミュレーション環境において、作成されてプロセスシミュレーションブロック152に接続されたMPCブロック150を含む。シミュレーション構成149は、図2のステップ128で、例えば、完成したMPCブロックをテストするのに用いられて、そのために作成されたプロセスを十分に制御するかどうか決定してもよいし、または、図2のステップ135で、例えば、MPCブロックを用いて訓練または他のテスト環境を与えるのに用いられてもよい。図8のMPCブロック150は3個の入力IN1-IN3と3個の出力OUT1-OUT3とを有するものとして示され、3個入力X1-X3と3個52に接続され、出力Y1-Y3は、MPCブロック150の入力IN1-IN3にそれぞれ接続される。シミュレーションブロック152は、図8のブロック154により示されるように、図2のステップ125で作成されたプロセスモデルを用いてそのためのMPCブロック150が作成され

10

20

30

40

50

たプロセスをシミュレートしてもよい。この場合、図2のステップ125で作成されたプロセスモデルはシミュレーションブロック152内に記憶されて、そのプロセスモデルとMPCブロック150から受け取った入力に基づいて、プロセスの応答をシミュレートするのに用いられてよい。代替的には、シミュレーションブロック152は、図8のブロック156により示されるように、ステップ152で生成されたプロセスモデルに対して変更されるプロセスモデルから作成されてもよい。この場合、ステップ125で作成されたプロセスモデルは、例えば、プロセスになされた物理的な変更、プロセス内の機器のエージングなどにより引き起こされたプロセスに対する変更をシミュレートするために変更されてよい。所望であれば、図2のブロック125で作成されたプロセスモデルは、このブロックが最初にMPCブロック150を作成するのに用いられるプロセスと一致しないプロセスを制御するのに用いられるときに、MPCブロック150の動作をテストするために様々な方法で変更されてよい。このように、所望であれば、ブロック156が与えた変更されたプロセスモデルは、シミュレーションブロック152内で用いられて、プロセスが変化したりMPC制御論理に一致しないときにMPCブロックが与える制御の範囲を決定し、これにより、ユーザは、長期間にわたって動作するかまたはプロセス変化が生じたプロセスを制御するのに良好なMPCブロックを設計することができる。

【0053】

変更されたプロセスを生成するには、ユーザまたはオペレータはルーチンを実行し、ステップ125で作成されたプロセスモデルまたは、 5×5 MPCブロック150のために図9のスクリーンに示されるもののような、関連するプロセス入力／出力応答曲線の組を見て、これらの応答曲線のうちの変更すべきひとつを選択する。選択された応答曲線（オーバーヘッド% C3H6対リボイラーオイル流量と示される）はさらに、図10に示されるようなさらなるスクリーンによって表示されるか操作されてよい。図10に示されるように、ユーザまたはオペレータは、応答曲線をインポート（取り入れ）するか消去し、FIR応答を加え、不動作時間およびゲインなどの応答パラメータを変更し、新たなスタートポイントまたは終了ポイントを選択し、曲線のポイントの値を変更し、曲線に対し様々な勾配をつけ、曲線をスケールして、それにより、変更された応答曲線、したがって、変更されたプロセスモデルを作成する。当然、ユーザまたはオペレータは、他の方法でプロセスモデルを部分的に変更（修正）するか変更する。ユーザは、プロセスモデル150を作成するか編集してもよく、このようなモデルからプロセスシミュレーションブロックまたはエレメント152を作成してもよく、プロセスシミュレーションブロック152をMPCブロック150に通信可能に接続してもよく、左手のプロセッサ13において、または他の所望のシミュレーション環境においてもしくはプロセス環境において、図1に示されるルーチン153のようなルーチンを用いて、接続されたループを実行してもよい。

【0054】

さて図11を参照して、実際のプロセス174に接続されたMPCブロック172を有するものとしてさらなる制御ループ170が示される。この場合、MPCブロック172の3個の出力OUT1-OUT3は、プロセス174のために作成されたプロセスモデル178に基づいたシミュレートされたプロセスブロック176の3個の入力X1SIM-X3SIMだけでなくプロセス174の3個のプロセス入力X1-X3に接続されれる。この構成においては、MPCブロック172は、実行時間においてプロセス174を制御し、シミュレートされたプロセスブロック176も制御するが、このブロック176は、例えば、図1のワークステーション13のうちのひとつにおいてまたは他のコントローラもしくは装置において実行されてよく、シミュレートされた出力Y1SIM-Y3SIMを生成する。このようなシステムにおいては、シミュレートされたプロセスブロック176の出力Y1SIM-Y3SIMと、実際のプロセス174の出力Y1-Y3とは比較され、プロセス出力Y1-Y3がシミュレートされた出力Y1SIM-Y3SIMとは非常に異なるかどうか、したがって、実際のプロセス174とMPCブロック172がそこからが作成されたプロセスモデル178とに間に不一致があるかどうか決定する。不一致がある場合、別のMPCブロック172を作成するかまたはプロセス174を制御する際にMPCブロック172が用いる論理パラメー

10

20

30

40

50

タまたはモデルを再生する必要がある。

【0055】

また、もしも何らかの理由でプロセス174の出力Y1-Y3のうちのひとつが、例えば、この出力を測定するセンサの誤動作が原因でエラーになった場合、シミュレートされたプロセスブロック176からの対応するシミュレートされた出力は、図11の点線で示されるようにMPCブロック172の適切な入力に与えられてよく、これにより、故障したセンサまたは装置が取りかえられるか修理されるまで、MPCブロック172は実際のプロセス174の良好な制御を与えることができる。このようにして、実際のプロセス出力の各々のために、仮想プロセス出力が、シミュレートされたプロセスブロック176により作成されてよいし、これらの仮想出力のうちのひとつ以上は、対応する実際のプロセス出力に欠陥があるかまたは使用できないときにMPCブロック172への入力として使用されてよい。例えば、深夜に、プロセス出力Y1-Y3のうちのひとつを測定するセンサが誤動作した場合、ユーザまたはオペレータは、修理者が故障したセンサを取りかえるか修理する次の日まで、MPCブロック172がプロセス174の十分な制御を与えることができるよう、MPCブロック172の適切な入力に対応する仮想出力を単に接続すればよい。シミュレートされたプロセスブロック176は、実際のプロセス174が実行されている間中実行可能であり、シミュレートされたプロセスブロック176が現実的な仮想出力を生成することができるように全て同じ入力が設けられている。当然、他のシミュレーションシナリオが実行されて、MPCブロック150または172の作成と関連して作成されたプロセスモデルを用いることができるし、またはMPCブロック150または172の作成と関連して作成されたプロセスモデルの変形例として作成されたプロセスモデルを用いることもできる。10

【0056】

必要な制御論理パラメータおよびそのためのプロセスモデルがない場合にMPCブロックを作成して、このブロックを、他の制御ブロックまたはエレメントがシステム内で接続される方法と同様の方法でプロセス制御システム内で接続し、MPCブロックを実行してプロセスデータを収集し、プロセスデータからプロセスモデルを作成し、プロセスモデルからMPCブロックのための論理パラメータを作成し、論理パラメータをロードし、そして必要であれば、プロセスモデルをMPCブロック内にダウンロードすることにより、オフラインになる必要もなく、MPC制御ルーチンがどのように作成されなければならないかについて多くを知る必要もなく、プロセスモデルを生成するために波形を作成するよう多大な技術的処理を行う必要もなく、制御ルーチンを再プログラムしてモデル予測制御または他のアドバンスド制御を実行する必要もなく、ユーザはプロセス制御ルーチン内でMPCブロックまたはMPCモジュールを作成できる。その結果、この方法は時間および費用を節約し、プロセス制御環境内で仮想プロセス出力のシミュレーションおよび生成のためなど他の目的のために、作成されたプロセスモデルを用いる。20

【0057】

理解されるだろうが、ここで述べてきたMPCルーチンまたはアドバンスド制御論理生成ルーチンおよび方法により、ユーザは、MPCブロック、ニューラルネットワークモデル化ブロックまたはニューラルネットワーク制御ブロックなどのアドバンスド制御ブロックが作成される方法について多大な専門的知識を必要とせず、これらのブロックを作成することが可能であり、オペレータは、アドバンスド制御を実行するために多大なプロセス再プログラミングを実行することもなくアドバンスド制御ブロックを作成して用いることができる。アドバンスド制御ブロックは、システム内の他の制御エレメントと同じプログラミングパラダイムを用いて作成されるので、ユーザは、プロセスの一貫したビュー、またはアドバンスド制御ブロックを有するプロセスの図形表示が与えられる。さらに、プロセスモデルは、例えば、MPC機能ブロックのために作成される必要があるので、このプロセスモデルは、テスト、訓練、プロセス/プロセスモデル不一致を検出すること、またはプロセスを制御する際に用いられるプロセスの仮想出力を生成することなどの他の目的のためにプロセスをシミュレートするのに用いることが出来るシミュレーション機能ブロ304050

ックを生成するのに用いることができる。

【0058】

アドバンスド制御ブロック、プロセスシミュレーションブロック、ならびに関連の生成ルーチンおよびテストルーチンは、ここではフィールドバス装置および標準の4 - 20ミリアンペアの装置と関連して用いられると説明したが、当然、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行されてよく、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用いられてよい。さらに、ここで「機能ブロック」という表現を用いているのは、フィールドバスプロトコルまたはデルタVコントローラプロトコルが機能ブロックとして識別するものに限定されず、なんらかのプロセス制御機能を実行するのに用いられる任意のタイプの制御システムおよび/または通信プロトコルと関連する他のタイプのブロック、プログラム、ハードウェア、ファームウェアなどを含む。また、機能ブロックは典型的には、オブジェクト志向プログラミング環境内でオブジェクトの形をとっているが、その必要がない場合もある。
10

アドバンスド制御ブロック、プロセスシミュレーションブロック、ならびに関連の生成ルーチンおよびテストルーチンは、好ましくはソフトウェアで実行されるが、ハードウェア、ファームウェアなどで実行されてもよく、プロセス制御システムと関連する他のプロセッサにより実行されてよい。このように、ここで述べるルーチン40は、所望であれば、標準的な多目的CPUにおいて、または例えば、ASICのような特殊設計されたハードウェアもしくはファームウェア上で実行されてよい。ソフトウェアで実行するならば、ソフトウェアは、コンピュータまたはプロセッサなどのRAMまたはROM内で磁気ディスク、レザディスク、光ディスク、または他の何らかの記憶媒体上のような、コンピュータ読み取り可能なメモリ内に記憶されてもよい。同様に、このソフトウェアは、例えば、コンピュータ読み取り可能なディスクまたは他の可搬コンピュータ記憶メカニズム上でのものを含む公知のまたは所望の送り方法によりユーザまたはプロセス制御システムに送られてもよいし、電話回線、インターネットなどのような通信チャンネル上で変調されてもよい（同じものであるとして見られるかまたは可搬の記憶媒体によりこのようなソフトウェアを与えることと交換可能である）。

【0059】

本発明は、具体例を参照して説明されたが、これらは、あくまでも例示的であり、本発明を限定するものではなく、本発明の精神および範囲を逸脱することなく、開示された実施の形態に、変更、追加、およびまたは消去を加えてよいことは当業者には明らかである。
30

【図面の簡単な説明】

【図1】アドバンスド制御ブロックを作成して用いることのできるプロセス制御システムのブロック/概略図である。

【図2】図1のプロセス制御システム内のアドバンスド制御ブロックの動作および作成を示すフローチャートである。

【図3】プロセス制御ルーチン内で接続されてプロセスを制御するモデル予測制御ブロックのブロック図である。

【図4】Aは、プロセス制御ルーチン内の機能ブロックに接続されて既存の制御戦略を調整するモデル予測制御機能ブロックのブロック図である。
40

Bは、プロセス制御ルーチン内の機能ブロックに接続されて既存の制御戦略を調整するモデル予測制御機能ブロックのブロック図である。

【図5】プロセス制御ルーチン内の機能ブロックに接続されて単一ループ制御ルーチンを調整するモデル予測制御機能ブロックのブロック図である

【図6】アドバンスド制御ブロックを作成するのに用いられるプロセスモデル化ツールにより生成されたスクリーンディスプレイの例を示す図である。

【図7】図1のプロセス制御システムの部分のブロック/概略図であり、アドバンスド制御ブロックと関連する図形ビューを用いることを示す図である。

【図8】プロセスシミュレーションブロックに接続されたモデル予測制御ブロックのブロック図である。
50

【図9】アドバンスド制御ブロックを作成するかまたはプロセスシミュレーションブロックを作成するために用いられるプロセスモデル化ツールにより生成されたスクリーンディスプレイの例を示す図である。

【図10】アドバンスド制御ブロックを作成するかまたはプロセスシミュレーションブロックを作成するために用いられるプロセスモデル化ツールにより生成されたスクリーンディスプレイの例を示す図である。

【図11】プロセスと、プロセスの動作をシミュレートして仮想プロセス出力を生成するプロセスシミュレーションブロックとに接続されたモデル予測制御ブロックのブロック図である。

【符号の説明】

10

5 8 プロセス

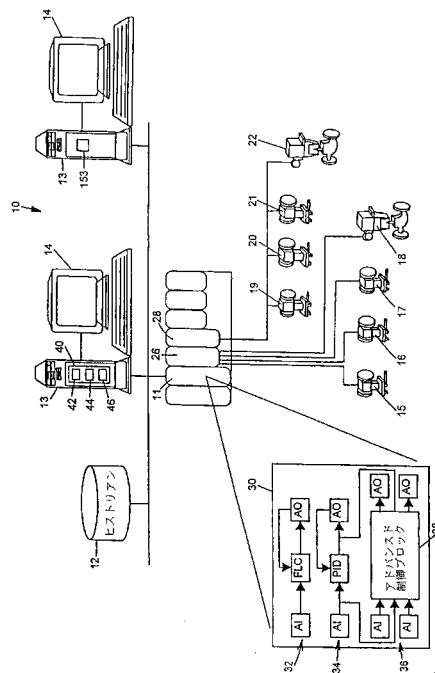
5 6 アドバンスド制御ブロック

1 0 2 制御論理

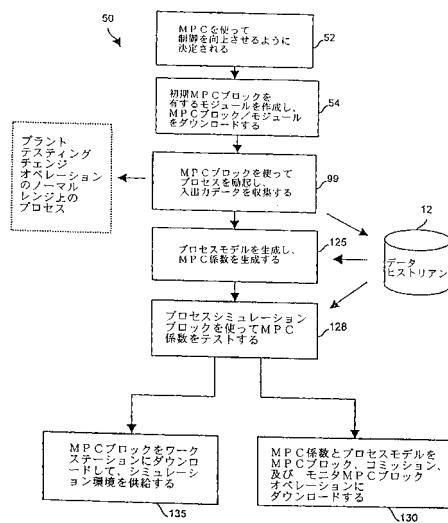
1 0 1 波形ジェネレータ

1 0 4 プロセスモデル

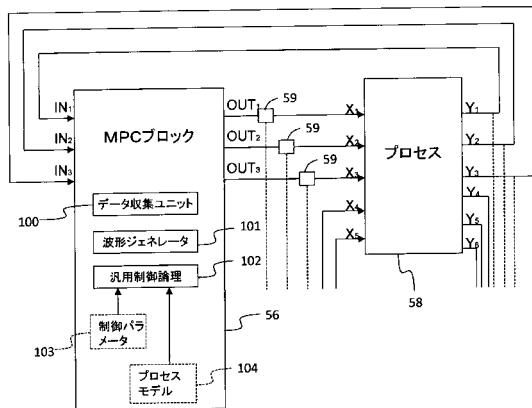
【図1】



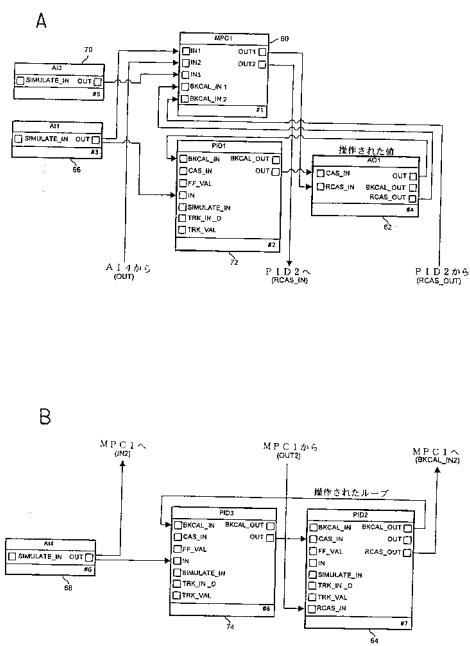
【図2】



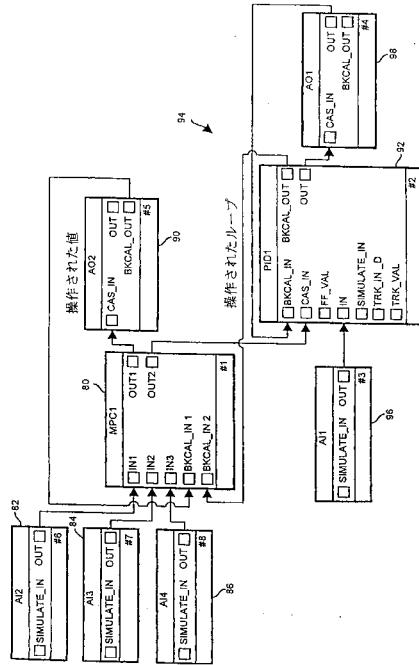
【図3】



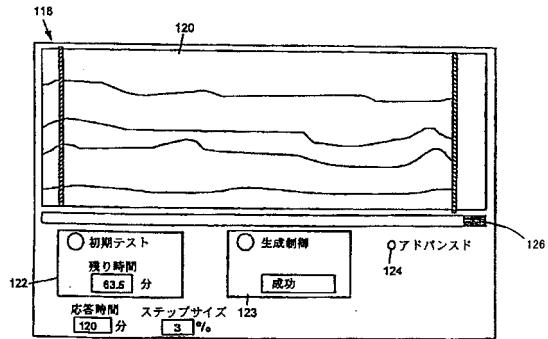
【図4】



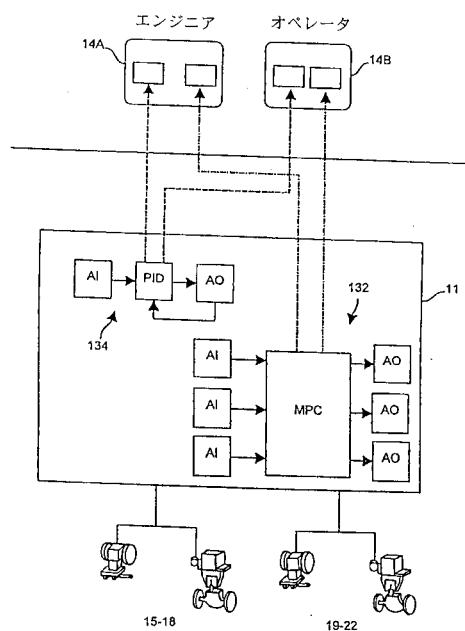
【図5】



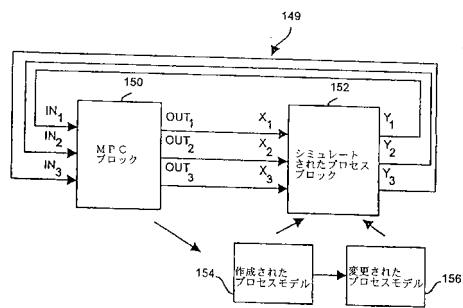
【図6】



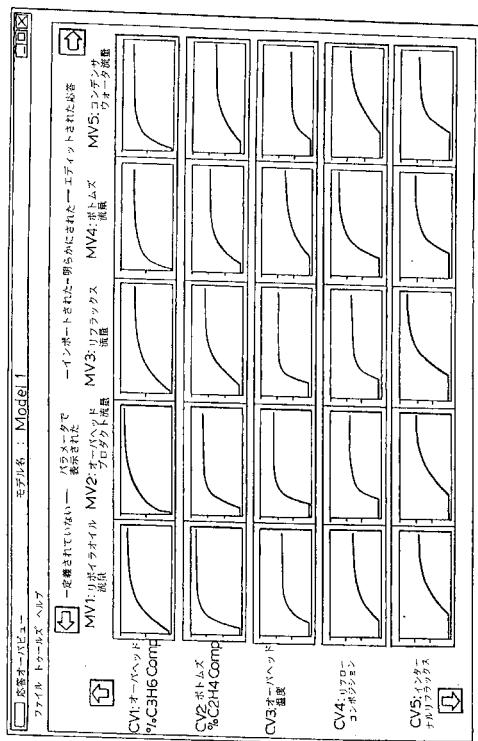
【図7】



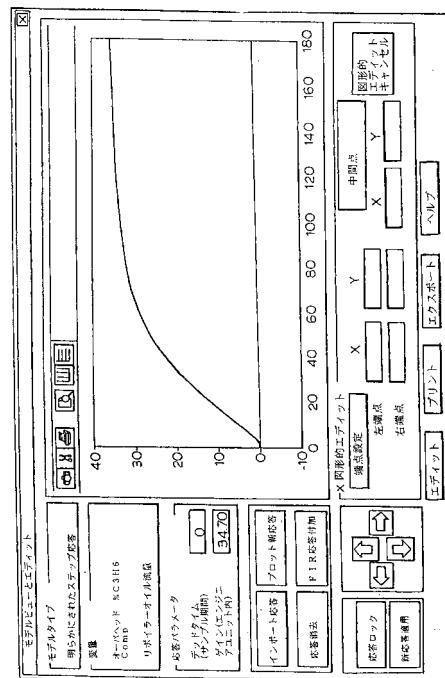
【 四 8 】



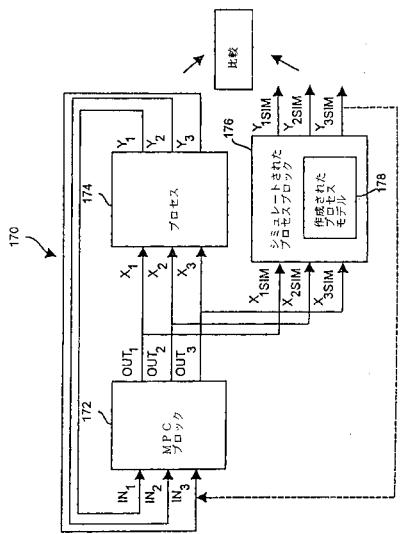
【 図 9 】



【 囮 1 0 】



【図 1 1】



フロントページの続き

(72)発明者 ウォズニス, ウィルヘルム ケ .
アメリカ合衆国 7 8 6 8 1 テキサス ラウンドロック ヒルサイド ドライブ 1 7 0 0 4
(72)発明者 ツオヴラ, ヴァシリキ
アメリカ合衆国 7 8 7 5 9 テキサス オースティン セントン センター パークウェイ #
1 1 0 6 4 6 0 0
(72)発明者 ティエル, ダーク
アメリカ合衆国 7 8 7 5 8 テキサス オースティン マウンテン クウェイル ロード 1 0
1 0 9

合議体

審判長 千葉 成就

審判官 菅澤 洋二

審判官 刈間 宏信

(56)参考文献 特開平6 - 9 5 7 0 6 (JP, A)
特開平11 - 1 6 7 4 0 3 (JP, A)
特開平10 - 9 1 2 1 1 (JP, A)
特開平4 - 3 3 1 0 2 (JP, A)
特開昭6 3 - 9 4 3 0 4 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G05B13/04