

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2013-539139

(P2013-539139A)

(43) 公表日 平成25年10月17日(2013. 10. 17)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G05B 13/04 (2006.01)</b>	G05B 13/04	5H004
<b>G05B 13/02 (2006.01)</b>	G05B 13/02	T
<b>G05B 17/00 (2006.01)</b>	G05B 17/00	

審査請求 未請求 予備審査請求 有 (全 54 頁)

(21) 出願番号	特願2013-531862 (P2013-531862)	(71) 出願人	505384841
(86) (22) 出願日	平成23年9月29日 (2011. 9. 29)		ザ マスワークス, インク
(85) 翻訳文提出日	平成25年5月28日 (2013. 5. 28)		THE MATHWORKS, INC.
(86) 国際出願番号	PCT/US2011/054008		アメリカ合衆国 マサチューセッツ州 O
(87) 国際公開番号	W02012/050977		1760 ナティック アップル ヒル
(87) 国際公開日	平成24年4月19日 (2012. 4. 19)		ドライブ 3
(31) 優先権主張番号	13/026, 823		3 Apple Hill Drive,
(32) 優先日	平成23年2月14日 (2011. 2. 14)		Natick, MA 01760 (U
(33) 優先権主張国	米国 (US)		S).
(31) 優先権主張番号	12/893, 670	(74) 代理人	100136630
(32) 優先日	平成22年9月29日 (2010. 9. 29)		弁理士 水野 祐啓
(33) 優先権主張国	米国 (US)	(72) 発明者	ガヒネット, パスカル
			アメリカ合衆国 マサチューセッツ州 O
			1748 ホプキントン ウェッジウッド
			ドライブ 24

最終頁に続く

(54) 【発明の名称】 複数入力・複数出力制御構造のインタラクティブな制御

## (57) 【要約】

## 要約

代表的な実施形態によれば、ユーザは、多変数フィードバック制御問題をインタラクティブに定式化しかつ解くことができる。例えば、ユーザは、複数の制御要素が1つまたは複数のフィードバックループに分散された問題であって、制御システムの全体的な性能およびロバスト性を最適化するために一緒に調整する必要がある問題を解くことができる。幾つかの実施形態によれば、ユーザは、設計要件および目的を自分が慣れ親しんだ形式で指定できる。幾つかの実施形態は調整可能なパラメータに作用して、ユーザにより与えられた上記設計要件および/または目的を満足するような状態で上記の前記問題を解くことができる。

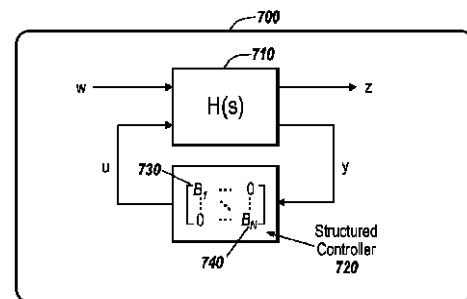


Fig. 7

**【特許請求の範囲】****【請求項 1】**

任意の制御構造において設計パラメータを調整するためのコンピュータ実装方法であって、

1つまたは複数の調整可能な構成要素を得る段階と、

前記制御構造を変換する段階であって、該変換が、前記制御構造の調整される構成要素を固定ダイナミクスから分離する、変換する段階と、

一システムにおける1つまたは複数の二地点間伝達関数に関するH またはH2目標または制約を用いる段階であって、前記H またはH2目標または制約が、

設計目標または

設計要件を表現する、用いる段階と、

非平滑H またはH2最適化アルゴリズムのクラスから選択されたメンバに基づいた調整器を用いて前記制御構造を調整する段階であって、前記調整器が、

前記パラメータを調整し、前記調整が、

前記H またはH2目標を最適化しまたは前記H またはH2制約を実現する、調整する段階とを含む、方法。

10

**【請求項 2】**

調整する前記段階が、MATLAB互換言語、Simulink互換言語、または、LabVIEW互換言語を用いて実行される、請求項1に記載の方法。

**【請求項 3】**

前記調整可能な構成要素は、グラフィカルモデリング環境で使用されるブロックセットから選択される、請求項1に記載の方法。

20

**【請求項 4】**

前記調整の速度が、前記制御構造のインタラクティブな設計および調整をサポートする、請求項1に記載の方法。

**【請求項 5】**

複数の二地点間伝達関数において別個の目標または制約を表現することが、ユーザのために多目的または多要件設計タスクを単純化する、請求項1に記載の方法。

**【請求項 6】**

用いる前記段階が、

前記設計要件を前記二地点間伝達関数に適用する段階をさらに含み、

適用する前記段階が、多目的および多要件設計タスクの定式化を促進する、請求項1に記載の方法。

30

**【請求項 7】**

前記調整可能な構成要素が、ユーザに関連した入力から得られる、請求項1に記載の方法。

**【請求項 8】**

1つまたは複数の二地点間伝達関数を識別する段階と、

前記制御構造を変換するために前記1つまたは複数の二地点間伝達関数を用いる段階をさらに含む、請求項1に記載の方法。

40

**【請求項 9】**

前記制御構造が標準形式に変換される、請求項8に記載の方法。

**【請求項 10】**

前記標準形式が、

線形モデルと、

前記調整可能ブロックとを含む、請求項9に記載の方法。

**【請求項 11】**

前記線形モデルが調整不可能な構成要素を含み、

前記調整可能な構成要素はブロック対角様式で配置される、請求項10に記載の方法。

**【請求項 12】**

50

前記調整可能な構成要素の一部はブロック対角様式で配置される、請求項10に記載の方法。

【請求項 13】

前記調整可能な構成要素をパラメータ化する段階と、

前記パラメータ化に基づいて前記調整可能な構成要素と相互作用する段階とをさらに含む、請求項1に記載の方法。

【請求項 14】

前記調整可能な構成要素が、1つまたは複数の自由パラメータを含み、前記方法が、

前記自由パラメータと相互作用する段階をさらに含む、請求項1に記載の方法。

【請求項 15】

前記制御構造が、フィードフォワード制御構造である、請求項1に記載の方法。

【請求項 16】

前記線形モデルが、前記線形モデルに還元されるモデルを表現する、請求項1に記載の方法。

【請求項 17】

前記線形モデルが集中線形モデルである、請求項1に記載の方法

【請求項 18】

プロセッサ上で実行されると、調整可能な構成要素を静的に指定し、調整可能な構成要素を動的に指定し、前記調整可能な構成要素のパラメータと相互作用するためのアプリケーションプログラムインターフェース(API)を実装する実行可能命令を保持する1つまたは複数の非一時的コンピュータ可読媒体であって、前記媒体が保持する1つまたは複数の実行可能命令が、

所定組の構成要素のパラメータ化を実現する所定のインターフェースを識別し、

算術演算またはヘルパー関数を用いて調整可能な構成要素を動的に生成する段階と、

前記調整可能な構成要素のパラメトリックモデルを生成する段階であって、前記パラメトリックモデルが前記調整可能な構成要素の調整可能なパラメータと相互作用する、生成する段階とを含む、媒体。

【請求項 19】

前記調整可能な構成要素が、MATLAB互換言語、Simulink互換言語、またはLabVIEW互換言語で実装される、請求項18に記載の媒体。

【請求項 20】

前記算術演算または前記ヘルパー関数の組を実装する段階とをさらに含み、

前記調整可能な構成要素が、前記実装された一組の算術演算または前記実装された一組のヘルパー関数を用いて動的に生成される、請求項18に記載の媒体。

【請求項 21】

前記パラメトリックモデルが、ユーザ入力に関連付けられた命令に応答して前記調整可能なパラメータと相互作用し、前記命令が、

前記調整可能なパラメータの初期化と、

前記調整可能なパラメータの固定と、

前記調整可能なパラメータからの選択したものの解放と、のうち1つまたは複数と関連付けられている、請求項18に記載の媒体。

【請求項 22】

プロセッサ上で実行されると、アプリケーションプログラムインターフェース(API)を実装する実行可能命令を保持する1つまたは複数の非一時的コンピュータ可読媒体であって、前記媒体が保持する1つまたは複数の実行可能命令が、

線形時間不変モデル構成要素と、調整可能な構成要素を記述する1つまたは複数のソフトウェアベースインターフェースとを結合し、

前記線形時間不変モデル構成要素および前記調整可能な構成要素を含む制御システムの標準形式を構築し、

1つまたは複数の設計要件を満足するために調整されるパラメトリックモデルを生成し

10

20

30

40

50

、前記パラメトリックモデルは前記標準形式に基づいている、媒体。

【請求項 23】

オブティマイザを用いて、前記パラメトリックモデルを最適化することをさらに含み、最適化する前記段階が、

前記標準形式および前記調整可能なパラメータと相互作用して、H またはH2目標を最小化しまたはH またはH2制約を実現する、請求項22に記載の方法。

【請求項 24】

入力構文を介して前記APIと相互作用するユーザが利用可能なヘルパー関数を提供する段階をさらに含む、請求項22に記載の方法。

【請求項 25】

関数を与える段階であって、

前記関数が入力機構をサポートする入力機構と互換性を有する、与える段階と、前記入力構文に従って前記関数と相互作用する段階とをさらに含む、請求項22に記載の方法。

【請求項 26】

前記制御システムに問い合わせるために、または前記制御システムを解析するために前記関数を用いる段階をさらに含む、請求項25に記載の方法。

【請求項 27】

前記調整可能な構成要素は、グラフィカルモデリング環境で使用されるブロックセットから選択される、請求項22に記載の方法。

【請求項 28】

前記調整の速度が、前記制御システムのインタラクティブな設計および調整をサポートする、請求項22に記載の方法。

【請求項 29】

前記制御システムがゲインスケジューリングを用いて制御される、請求項22に記載の方法。

【請求項 30】

前記線形時間不変モデル構成要素と調整可能な構成要素を記述する1つまたは複数のソフトウェアベースインターフェースとを結合する前記段階が、

算術演算と、

ダイアグラム演算との一方または両方を使用する段階をさらに含み、前記ダイアグラム演算が、

直列接続、

並列接続、または

フィードバック接続を含む、請求項22に記載の方法。

【請求項 31】

ループ整形要件をプログラムにより定式化するためのコンピュータ実装方法であって、ソフトウェアツールと相互作用する段階を含み、前記ソフトウェアツールが、

制御構造を導出し、

フィルタを導出し、

前記導出された制御構造およびフィルタから標準形式を構築し、

前記導出された制御構造およびフィルタからH またはH2制約を構築し、

前記標準形式および前記H 制約または前記H2制約は、制御システムに関する設計要件を獲得する、方法。

【請求項 32】

前記標準形式が、MATLAB互換言語、Simulink互換言語、またはLabVIEW互換言語を用いて構築される、請求項31に記載の方法。

【請求項 33】

前記ソフトウェアツールは、Simulink互換環境またはLabVIEW互換環境と相互作用する、請求項31に記載の方法。

10

20

30

40

50

**【請求項 3 4】**

前記ソフトウェアツールの速度がインタラクティブな制御システムの設計をサポートする、請求項31に記載の方法。

**【請求項 3 5】**

コンピュータ実装方法であって、

構築手順を用いて、調整可能なパラメータ値の標準形式の状態空間モデルを構築する段階であって、当該構築が、

標準形式を備えた所望の状態空間モデルを得るために、調整可能なパラメータを備えた調整可能な構成要素の集約状態空間行列を、集中プラントモデルの状態空間表現と組み合わせることにより実行される。構築する段階と、

10

勾配情報を計算する段階とを含み、前記勾配情報は、

目標または制約を微分し、

ソフトウェアオブジェクトにより、スカラー値関数の勾配を与え、

前記調整可能な構成要素により与えられた勾配を前記目標および制約の全体的な勾配に集約することにより、計算される、方法。

**【発明の詳細な説明】****【技術分野】****【0 0 0 1】**

本願は、その内容を引用して本明細書に援用する2011年2月14日付けの米国特許出願第13/026,823号および2010年9月29日付けの米国特許出願第12/893,670号の優先権を主張する。

20

**【図面の簡単な説明】****【0 0 0 2】**

添付の図面は、本明細書の一部として組み込まれかつその一部をなすものであり、本発明の1つ又は複数の実施形態をその詳細な説明と共に記載する。これら図面は次の通りである。

【図 1】H 合成を用いるための従来の構成を示す。

【図 2】本発明の実施形態および技法を実施するための代表的なシステムを示す。

【図 3】図3は、本発明の諸側面を実装するために使用可能なモデリング環境の代表的な実装例を示す。

30

【図 4】その構成要素を調整するために本発明の代表的な実施形態を適用できる代表的なコントローラーアーキテクチャを示す。

【図 5】その構成要素を調整するために本発明の代表的な実施形態を適用できる代表的なコントローラーアーキテクチャを示す。

【図 6】その構成要素を調整するために本発明の代表的な実施形態を適用できる代表的なコントローラーアーキテクチャを示す。

【図 7】システムのH 合成表現と組み合わせてパラメータを調整するための構造化コントローラーを用いる本発明の代表的な実装例を示す。

【図 8】本発明の一実施形態と組み合わせて使用されるフィードバックループの代表的な所望のループ形状を示す。

40

【図 9】調整可能な要素を含むプラントおよびコントローラーを含む代表的な構成であって、構成要素を調整するために代表的な技法を適用できる構成を示す。

【図 10】本発明の一実施形態を用いて調整可能な複数入力・複数出力制御問題と共に使用される代表的な所望のループ形状を示す。

【図 11】本発明の実施形態と共に使用可能なコマンド、オブジェクト、および出力を表示するための代表的なインターフェースを示す。

【図 12】本発明の実施形態と共に使用可能なコマンド、オブジェクト、および出力を表示するための代表的なインターフェースを示す。

【図 13】本発明の実施形態と共に使用可能なコマンド、オブジェクト、および出力を表示するための代表的なインターフェースを示す。

50

【図 1 4】本発明の実施形態と共に使用可能なコマンド、オブジェクト、および出力を表示するための代表的なインターフェースを示す。

【図 1 5】図15は、本発明の一実施形態を用いて調整可能な構成要素を備えた代表的なコントローラを示す。

【図 1 6】(A)および(B)本発明の実施形態と共に使用可能なコマンドおよび/または出力を表示するための代表的なインターフェースを示す。

【図 1 7】本発明の実施形態を用いて調整できる調整可能構成要素を含む代表的な自動操縦装置を図示する。

【図 1 8】本発明の実施形態を用いて調整できる調整可能構成要素を含む代表的な自動操縦装置を図示する。

【図 1 9】図17および18の自動操縦装置に関する入力を受け取りかつ/または出力を表示するために使用可能な代表的なインターフェースを示す。

【図 2 0】図17および18の自動操縦装置の構成要素を調整する本発明の実施形態と共に使用可能な代表的な目標ループ形状を示す。

【図 2 1】図17および18の自動操縦装置に関する入力を受け取りかつ/または出力を表示するための代表的なインターフェースを示す。

【図 2 2】図17および18の自動操縦装置に関する入力を受け取りかつ/または出力を表示するための代表的なインターフェースを示す。

【図 2 3】図17および18の自動操縦装置に関連した代表的なステップ応答を示す。

【図 2 4】図17および18の自動操縦装置に関連した感度関数のゲインを評価するために使用可能なプロットを示す。

【図 2 5】本発明の実施形態を実施するために使用可能な代表的な処理を示す。

【図 2 6】本発明の実施形態を実装するために使用可能な代表的なアーキテクチャを示す。

【図 2 7】本発明による分散型の実施形態を実装するための代表的なシステムを示す。

【発明を実施するための形態】

【0003】

複数入力・複数出力(MIMO)コントローラを設計するための従来のアプローチは、H 合成と呼ばれる技術を含むことができる。

【0004】

従来のH 合成技術

図1は、標準的なH 合成技法の使用例を表した構成を示す。図1は、プラントH(s) 110とコントローラC(s)

120を含むことができる。図1では、標準的なH 合成が、入力w 130から出力z 140への閉ループピークゲインを最小化するコントローラC(s) 120を計算する。コントローラC(s) 120は、ブラックボックス(例えば、ユーザはこのブラックボックス内部の内部表現に容易にアクセスできない)として動作できる集中H コントローラを表すことができる。H(s)110は、コントローラC(s)

120を用いて制御可能なプラントを表すことができる。プラントH(s)110は、線形プラントモデルとすることができる。

【0005】

従来のH 合成を用いて、システム100の閉ループ応答(いわゆるH ノルム)のピーク入力/出力ゲインを最小化できる。ユーザは、制御プラントH(s)の帯域幅、ロールオフ、オーバーシュート、および/または安定余裕などの設計に関わる側面または目標を指定する制御問題に関する設計仕様を持つことができる。H 合成に基づくフレームワークは、ユーザが仕様を実現するための補助に適用できる。しかし、従来の 合成技法を快適に使用しまたは熟練しているユーザ(例えばエンジニア)は非常に少ない。例えば、ユーザが慣れ親しんでいる通常の仕様を、従来の 合成技法で必要な正規化閉ループゲイン制約に変換することは、ユーザにとって単調で、直感で理解できず、長時間を要する作業と感ずることがある。さらに、従来のH 合成ツールおよび技法の技術的な制約は、典型的なユーザ

10

20

30

40

50

の作業をさらに遅らせかつ/または混乱させることがある。

【0006】

さらに、従来のH 合成技法は、ユーザの典型的な設計作業の流れをサポートしないため、ユーザにとって好ましくないものと感じられることがある。さらに、従来のH 合成技法は、コントローラC(s)120を、コントローラC(s)120により表現されるシステム構造の内部表現へのユーザによる容易なアクセスを許容しないブラックボックスとして扱う。さらに、従来のH 合成技術は、所与のシステムのC(s)120を生成する際に、別個の入力接続を結合し、かつこれら接続の間に制御ブロックを挿入する。このアプローチは、C(s)120が導出された元となるシステム構造を表現しないC(s)120の構造を生成してしまい、これは、典型的なユーザが従来のH 合成技法を扱う際に直面する困難さをさらに悪化させてしまう。

10

【0007】

例えば、ユーザは、実質的に実時間の操作を可能とする設計ツールを望むことであろう(すなわち、これらツールによりユーザは、ユーザをいらいらさせたり、ユーザと設計ツールとの対話を干渉したりする処理の遅れに遭遇することなく、インタラクティブに設計を行い、設計の諸側面をインタラクティブに修正かつ実行できる)。従来のH 合成技法は、多くの理由でインタラクティブな動作をサポートしないことがある。例えば、従来の技法は、特に構造制約または順序制約がコントローラC(s)120に課せられない場合、一対のリカッチの方程式を繰り返し解くことで最適コントローラの計算を試みることができる。上記一対の方程式を解こうとするこれらの繰り返し試行は、長時間を必要としかつ計算コストが高くつくことがあり、これによって、従来のH 合成技法は、インタラクティブな設計用途には特に望ましくないものとなる傾向がある。

20

【0008】

代表的な技法および実施形態の概要

代表的な実施形態は、ユーザが、多変数フィードバック制御問題を定式化しかつ解くために、直感的なユーザインターフェースを使用することを可能とする新規の技法を提供する。例えば、代表的な実施形態では、ユーザは、制御システムに分散した一緒に調節する必要がある複数の制御要素を備えた制御問題を扱うことができる。例えば、幾つかの実施形態は、1つまたは複数のフィードバックループに分散した複数の制御要素であって、システムの全体的な性能およびロバスト性を最適化するために一緒に調整する必要がある制御要素を含むことができる。代表的な実施形態はスケーラブルであり、実質的に任意の数の構成要素、ループおよび/または任意程度の複雑性を備えた問題の制御に応用できる。

30

【0009】

代表的な実施形態は、図1に類似したシステム表現であるが、ユーザが、コントローラC(s)をブラックボックスでなくホワイトボックスとして扱うことを可能とするシステム表現を使用できる。例えば、代表的な実施形態では、コントローラC(s)がブロック対角構造の順列(permutation)を含むことができ、ここでは、対角線上のブロック自身がなんらかの固定構造および複雑性を備えたブロック対角構造を持つことができる。幾つかの実施形態は、ブロック対角が調整可能な構成要素のみを表す構成か、ブロック対角がモデルの調整可能な構成要素と、1つまたは複数の固定または既知のダイナミクス(例えば、固定/既知の構成要素)とを表す構成を含むことができる。このC(s)の付加的な複雑性により、ユーザは、コントローラC(s)を、解析対象のシステムの制御アーキテクチャに調和した状態で表現できる。代表的な実施形態は、非平滑H オプティマイザを用いて任意のMIMO制御構造を自動的に調整できる。代用的な技法は、ソルバを、C(s)がブロック対角構造を含むH 合成表現の正準構造にあわせて特殊化する。

40

【0010】

代表的な実施形態および技法は、MIMO調整タスクの試行にあたり従来の技法によりもたらされる障害を除去するものである。例えば、代表的な実施形態および技法は、制御アーキテクチャおよびコントローラ構造の、オプティマイザに適した費用関数およびパラメータベクトルへの翻訳を自動化する。さらに、代表的な実施形態および技法によって、勾配

50

を安価に計算でき(反復毎に $O(N)$ で、 $N$ はパラメータの数)、さらに、各調整可能ブロックタイプ(ゲインPID、伝達関数、状態空間、生パラメータ等)がそれ自身の寄与を勾配に与えるオブジェクト指向様式で勾配を計算できる。代表的な実施形態は勾配を利用するが、その際に、ユーザが、従来のアプローチでなされていたように、ブロックに関する勾配を計算しそれらを組み合わせて費用関数の全体的な勾配を得る必要がない。従来のアプローチは、簡単なアーキテクチャを除いて手動では処理しにくいことがあり、本発明の幾つかの側面で使用される技法と比べると格段に効率が低いことがある。

#### 【0011】

本発明の代表的な技法は、計算ハードウェアで実現した場合、ユーザがMIMO調整タスクをインタラクティブに実行できるようになる。例えば、代表的な実施形態により、ユーザは、標準的なパーソナルコンピュータ(PC)並びにMATLAB技術計算およびプログラミング環境などの技術計算環境を用いて典型的には1秒未満から30秒未満の範囲での調整時間を体験できる。幾つかの実施形態は、処理時間を減少させるためマルチコアまたは他の種類の多重処理デバイス若しくは環境に導入できる。

10

#### 【0012】

本明細書に記載した実施形態は、それぞれの実例について線形制御システムとの関連で説明するが、本発明の実施形態を使用して非線形制御問題を解くこともできる。例えば、代表的な技法は、ゲインスケジューリングを含むがそれに限定されない非線形制御設計へのアプローチをサポートできる。

20

#### 【0013】

代表的なシステム

図2は、一実施形態を実施するための代表的なシステム200を示す。システム200を用いて1つまたは複数のエンティティを含むモデルを構築し、そのモデルのPIDコントローラを設計、実装し、かつ/またはそのコントローラのコードを生成するなどのそのモデルからコードを生成できる。システム200は、コンピュータ205と、取得ロジック210と、オペレーティングシステム215と、モデリング環境220と、モデル230と、入力デバイス240と、表示装置250と、モデル表現260と、プラント270とを含むことができる。図2のシステムは例示的なものであり、システム200の他の実施形態は、より少ない数のデバイス、多い数のデバイス、かつ/または図2の構成とは異なる構成のデバイスを含むことができる。

30

#### 【0014】

コンピュータ205は、処理動作、表示動作、通信動作などを実行するデバイスを含むことができる。例えば、コンピュータ205は、1つまたは複数の処理または記憶装置などの処理動作をユーザのために実行かつ/またはサポートするため使用可能なロジックを含むことができる。コンピュータ205の実施形態は、デスクトップコンピュータ、ラップトップコンピュータ、クライアント、サーバ、メインフレーム、携帯型情報端末(PDA)、ウェブ機能付き携帯電話、スマートフォン、スマートセンサ/アクチュエータ、または1つ若しくは複数の動作を実行するための、かつ/または1つ若しくは複数の結果を生成するための命令を実行する別の計算若しくは通信デバイスを含むこともできる。

#### 【0015】

さらに、コンピュータ205は、別のデバイスとの間でデータの送受信を行うことで通信動作を実行することもできる(図2には示さない)。データは、1つまたは複数のネットワークでかつ/または1つまたは複数のデバイスで使用するよう適合可能な実質的に任意形式を含む任意種類の機械可読情報に関する。データはデジタル情報またはアナログ状態を含むことができる。さらに、データはパケット化しかつ/またはパケット化しなくてもよい。

40

#### 【0016】

取得ロジック210は、コンピュータ205の外部のデバイスからデータを取得でき、コンピュータ205によってそうした情報を利用可能にすることができる。例えば、取得ロジック210は、コンピュータにデータを利用可能にするために用いられるアナログ デジタル変換器、デジタル アナログ変換器、フィルタ、マルチプレクサなどを含むことができる。コ

50



ンピュータ205は、取得したデータを用いてモデリング動作、コントローラ設計動作などを実行できる。

【0017】

オペレーティングシステム215は、コンピュータ205に関連付けられたハードウェアおよび/またはソフトウェアリソースを管理できる。例えば、オペレーティングシステム215は、ユーザ入力の受信、コンピュータ環境205の運用、メモリ割り当て、システム要求の優先順位付けなどに関連したタスクを管理できる。一実施形態では、オペレーティングシステム215は、仮想オペレーティングシステムとしてもよい。オペレーティングシステム215の実施形態は、Linux（登録商標）、Mac OS、Microsoft Windows（登録商標）、Solaris、UNIX（登録商標）などを含むことができる。さらに、オペレーティングシステム215は、コンピュータ205により提供可能な仮想マシン上で実行できる。

10

【0018】

モデリング環境220は、ユーザが、数学、科学、工学、医学、ビジネスなどを含むがそれらに限定されない分野に関するシミュレーションまたはモデリングタスクを実行可能とする計算環境を提供する。モデリング環境220は、実行可能意味論を備えたモデルをユーザが構築可能にする命令を実行する1つまたは複数のアプリケーションをサポートできる。例えば、一実施形態では、モデリング環境220は、ユーザが、実行可能意味論を備えた自由形式モデル(1次、2次、3次、4次、5次などのモデル)モデルを作成できるようにしてもよい。モデリング環境220は、さらに、時間ベース、イベントベースなどのモデリング動作をサポートできる。

20

【0019】

モデル230は、実行可能なテキスト形式またはグラフィカルモデルを含むことができる。例えば、モデル240は、時間ベースモデル、イベントベースモデル、状態遷移モデル、データフローモデル、構成要素ダイアグラム、エンティティフローダイアグラム、方程式ベース言語ダイアグラムなどでよいテキスト形式モデルまたはグラフィカルモデルの情報を含むことができる。モデル230のグラフィカルな実施形態は、処理を実行するための実行可能コードを表現するエンティティ(例えば、ブロック、アイコン等)を含むことができる。これらエンティティのコードを実行して、モデルを用いてシミュレーションを行うことができる。エンティティ同士は、モデル中でデータを1つのエンティティから別のエンティティに移送するための経路を表す線を用いて接続できる。

30

【0020】

入力デバイス240はユーザ入力を受け取ることができる。例えば、入力デバイス240は、ユーザの動作または行為を、コンピュータ205により解釈可能な信号またはメッセージに変換できる。入力デバイス240は、キーボード、ポインティングデバイス、生体計測デバイス、加速度計、マイクロフォン、カメラ、ハプティックデバイス等を含むことができるが、それらに限定されない。

【0021】

表示装置250は、情報をユーザに表示できる。表示装置250は、ブラウン管(CRT)、プラズマ表示装置、発光ダイオード(LED)表示装置、液晶表示装置(LCD)等を含むことができる。表示装置250の実施形態は、所望ならユーザ入力(例えば、タッチスクリーンを介して)ユーザ入力を受け取るように構成してもよい。一実施形態では、表示装置250は、1つまたは複数のグラフィカルユーザインターフェース(GUI)をユーザに提示できる。これらGUIは、モデル230および/または他の種類の情報を含むことができる。

40

【0022】

モデル表現260は、モデル230の可視表現および/またはプロットウィンドウなどのモデル230が与える可視表現を含むことができる。例えば、モデル表現260をユーザに表示してもよいし、線で接続された多数のエンティティを含むこともできる。モデル230の実行時には、モデル表現260は、例えばモデルを通るデータの流れを示すように変化させてもよい。

50

## 【 0 0 2 3 】

プラント270は、データをコンピュータ205に与える1つ又は複数のデバイスを含むことができる。例えば、プラント270は、加速時計、熱電対、光電送受信機、ひずみ計等のセンサを用いて監視されるエンジンシステムを含むことができる。一実施形態では、取得ロジック210は、プラント270からの信号をアナログまたはデジタル形式で受信し、それら信号をコンピュータ205での使用に適した形式に変換できる。

## 【 0 0 2 4 】

代表的なモデリング環境

図3は、モデリング環境220の代表的な実施形態を示す。モデリング環境220は、シミュレーションツール310と、エンティティライブラリ320と、インターフェースロジック330と、コンパイラ340と、コントローラロジック350と、オブティマイザ360と、シミュレーションエンジン370と、リポートエンジン380と、コードジェネレータ390とを含むことができる。図3に示したモデリング環境220の実施形態は例示的なものであり、モデリング環境220の他の実施形態も、本発明の精神から逸脱することなく、より多いまたは少ないエンティティを含むことができる。

10

## 【 0 0 2 5 】

シミュレーションツール310は、モデルを構築するためのアプリケーションでよい。シミュレーションツール310を用いて、実行可能意味論を含むテキスト形式モデルまたはグラフィカルモデルを構築できる。グラフィカルモデルの場合は、シミュレーションツール310によって、ユーザは、モデルエンティティおよび/または接続の作成、修正、診断、削除などを行うことができる。シミュレーションツール310は、ユーザ入力を受け取り、モデルを実行し、結果を表示し、コードを生成するなどのため、図2または3に示した他のエンティティと相互に作用できる。シミュレーションツール310は、テキスト形式モデルを構築若しくはそれらと対話するための編集ウィンドウおよび/またはグラフィカルモデルを作成若しくはそれらと対話するためのGUIを、ユーザに与えることができる。

20

## 【 0 0 2 6 】

エンティティライブラリ320は、ユーザが、モデル表現360を含む表示ウィンドウにドラッグアンドドロップできるコードモジュールまたはエンティティ(例えば、ブロック/アイコン)を含むことができる。グラフィカルモデルの場合は、さらに、ユーザは接続を用いてエンティティを結合し、プラント370などのシステムのグラフィカルモデルを生成できる。

30

## 【 0 0 2 7 】

インターフェースロジック330によって、モデリング環境220は、データおよび/または情報をデバイス(例えば、プラント270、ターゲット環境など)またはソフトウェアモジュール(例えば、関数、アプリケーションプログラムインターフェースなど)との間で送受信可能となる。一実施形態では、インターフェースロジック330は、取得ロジック310とモデリング環境220とを接続する。

## 【 0 0 2 8 】

コンパイラ340は、モデルを実行可能形式にコンパイルできる。コンパイラ340が生成するコンパイル済みコードは、コンピュータ205上で実行してモデリング結果を生成できる。一実施形態では、コンパイラ340は、モデルに関連したエラーを診断するデバッグ機能を実現する。

40

## 【 0 0 2 9 】

コントローラロジック350は、モデル330でコントローラを作成しかつ実装するために使用可能である。例えば、コントローラロジック350は、モデル表現260における様々なタイプのコントローラを表現するエンティティに機能を与える。モデルが実行されると、コントローラロジック350は、モデル表現260におけるエンティティと相互作用することによってこのモデルに対して制御処理を実行できる。一実施形態では、コントローラロジック350は、モデル表現360においてコントローラを実装する制御アルゴリズムを含むことができる。コントローラロジック350の実施形態は、タンドアローンまたは分散型の実装例とし

50

て動作するよう構成できる。

【0030】

オブティマイザ360は、モデルのコード、パラメータ、性能(例えば、実行速度)などを最適化できる。例えば、オブティマイザ360はコードを最適化して、そのコードが最適化されない状態で実行された場合に比べて、例えばそのコードがより少ないメモリを占有し、より効率的に実行し、より高速で実行することができる。オブティマイザ360は、例えば、コントローラのパラメータ最適するため、コントローラロジック350の最適化を行うこともできる。一実施形態では、オブティマイザ360はコンパイラ340、コントローラロジック350、コードジェネレータ390などと共に動作してもよいし、またはオブティマイザ360をそれらに組み込んでもよい。オブティマイザ360の実施形態は、例えばオブティマイザ360が処理するデータを受け取るための他のオブジェクト指向ソフトウェアと相互作用するソフトウェアオブジェクトを介して実装できる。

【0031】

シミュレーションエンジン370は、モデルを実行してシステムをシミュレートするための処理を実行できる。シミュレーションエンジン370は、ユーザの選択またはシステムの選択に基づいてスタンドアローンまたは遠隔シミュレーションを実行するよう構成できる。

【0032】

レポートエンジン380は、モデリング環境220における情報に基づいてレポートを生成できる。例えば、レポートエンジン380は、コントローラが設計仕様を満足するかを示すレポート、コントローラが安定して動作するかを示すレポート、モデルが適切にコンパイルするかを示すレポートなどを生成できる。レポートエンジン380の実施形態は、表示装置250で表示するための電子形式、ハードコピー形式および/または記憶装置における記憶に適した形式でレポートを生成可能である。

【0033】

コードジェネレータ390は、モデルからコードを生成できる。一実施形態では、コードジェネレータ390はコードを第1形式で受け取り、そのコードを第1形式から第2形式に変換できる。一実施形態では、コードジェネレータ390は、モデルの少なくとも一部から、ソースコード、アセンブリ言語コード、バイナリコード、インターフェース情報、構成情報、性能情報、タスク情報などを生成できる。例えば、コードジェネレータ390は、モデルからC、C++、SystemC、Java(登録商標)、ストラクチャードテキストなどのコードを生成できる。

【0034】

コードジェネレータ390の実施形態は、さらに、統一モデリング言語(UML)に基づく表現および/またはグラフィカルモデル(例えば、システムモデリング言語(SysML)、拡張マーク付け言語(XML)、リアルタイム/組込みシステムのモデリングおよび解析(MARTE)、ハードウェア記述言語(HDL)、自動車オープンシステムアーキテクチャ(AUTOSAR)など)の一部または全部からの拡張を生成できる。一実施形態では、オブティマイザ370は、コードジェネレータ390と相互作用して、パラメータ(例えば、メモリ使用、実行速度、多重処理など)に従って最適化されるコードを生成できる。本発明の原理と整合性があるモデリング環境の実施形態は、さらに、検証構成要素、妥当性検査構成要素などの構成要素を含むこともできる。

【0035】

本発明の実施形態を用いて、多変数フィードバック制御問題をインタラクティブに定式化し、解き、かつ実質的に任意次数および/または遅れの非線形モデルで使用するコントローラを設計できる。幾つかの実施形態は、厳密な線形化技法を用いて非線形モデルの少なくとも一部を表現できる線形時間不変モデルを生成するように構成可能である。

【0036】

代表的な制御アーキテクチャ

一例として、本発明の実施形態は、実質的に任意の順序で配置されかつ1つまたは複数

10

20

30

40

50

のフィードバックループを備えたコントローラブロックなどを含みうる複数の構成要素を備えた制御アーキテクチャに応用できる。

【0037】

図4は、本発明の実施形態が適用できる代表的な制御アーキテクチャ、具体的には高い迎え角モードにあるF-14で使用される自動操縦装置を示す。この自動操縦装置はGUI400に表示でき、ゲインおよび時間制約を含む8つの調整可能なパラメータを含むことができる。

【0038】

図5は、蒸留塔で使用されるコントローラの代表的なアーキテクチャを示す。図5のコントローラは、GUI500を介してユーザに表示され、調整が必要な4つの比例積分(PI)ゲインおよび2x2ゲイン行列を含むことができる。

10

【0039】

図6は、風力タービンで使用されるピッチおよびヨー制御の代表的なアーキテクチャを示す。図6のアーキテクチャは、GUI600を介してユーザに表示され、調整が必要な3つのPIコントローラおよび2つのゲインを含むことができる。図6に示したように、本発明の実施形態で使用されるGUIは、ウィンドウ、ペインなどの複数のインターフェースを含むことができる。

【0040】

本発明の実施形態は、さらに別の種類のアーキテクチャや、図4-6および本願の他の部分で示したアーキテクチャに比べ複雑性が高いまたは低いアーキテクチャで利用できる。例えば、実質的に任意の数の固定および/または調整可能な構成要素を備えたフィードバック構成またはフィードフォワード構成を含むアーキテクチャで利用できる。

20

【0041】

本明細書で開示した代表的な実施形態および/または技法は、任意の複数の制御構造を単一の一般的な表現に還元することによってそれら構造の効率的な調整を可能とする。

【0042】

代表的な正準構造

図7は、変換された任意の制御構造を表すために使用できる代表的な正準構造を図示する。

【0043】

30

図7を参照すると、システム700は、インターフェースを介してユーザに表示でき、さらに制御システムの非調整(例えば、固定かつ不変で、調整対象の自由パラメータを持たない)構成要素を単一の集中モデルへと組み合わせる線形モデルでよい $H(s)$ 710を含むことができる。幾つかの実施形態は、線形モデル(例えば、分散型モデル)に還元可能な任意のシステム表現を扱うように適合可能である。例えば、本発明の一実施形態では、 $H(s)$ 710は、システムの固定制御ダイナミクス(例えば、構成要素)を表現できる。一実施形態では、固定構成要素は、調整されない、変化可能な自由パラメータを含まない構成要素とすることができる。

【0044】

40

システム700は、調整対象の要素を含みうるコントローラ720も含むことができる。一実施形態では、調整可能な構成要素は、例えば調整される自由パラメータなどの変化するパラメータを備えた構成要素でよい。例えば、代表的な実施形態では、コントローラ720は、調整対象の制御要素を表す1つまたは複数のブロック $B_1$  730から $B_N$  740を含む構造化コントローラ(structured controller)とすることができる。一例として、コントローラ720が単一のブロックを含むときは、そのブロックは $B_1$ と呼び、3つのブロックを備えたコントローラ720は、 $B_1$ 、 $B_2$ 、および $B_3$ というブロックを含むものとする。調整対象の制御要素の設計は一樣ではない。例えば、調整対象の要素は、コントローラまたはプラント内のゲイン、動的要素(例えば、伝達関数、状態空間モデルなど)および/または設計パラメータを含むことができる。

【0045】

50

コントローラ720の代表的な実施形態は、ブロック $B_1$ - $B_N$ からなるブロック対角集合体に変形可能な項を含むことができ、構造化コントローラ720と呼ぶことができる。構造化コントローラ720は、反復されるブロック(すなわち、 $B_2$ などの特定のブロックは対角線上に複数回現れることがある)を含むことができる。構造化コントローラ720は、対角線上に調整可能な構成要素のみを含むように構成してもよいし、対角線上に構成可能な構成要素と固定構成要素とを含むように構成してもよい。さらに、構造化コントローラ720は、ブロック対角線の位置以外の位置に少なくとも1つの非ゼロ項を含むこともできる。幾つかの実施形態は、さらに、ブロック対角線にゼロである要素を含むこともできる。

【0046】

システム700は、次のように、周波数領域の対応する式を用いて表現できる。

10

【0047】

【数1】

$$\begin{cases} \begin{bmatrix} z \\ y \end{bmatrix} = H(s) \begin{bmatrix} w \\ u \end{bmatrix} \\ u = \begin{bmatrix} B_1(s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N(s) \end{bmatrix} y \end{cases} \quad (\text{Eq. 1})$$

【0048】

方程式1において、 $H(s)$ および調整されたブロック( $B_1$  -  $B_N$ )は、信号 $u$ および $y$ を介してフィードバック状態で相互作用する。式1の正準構造は次の3つの属性を備える。

20

【0049】

(1)

$B_1, \dots, B_N$

がブロックとして現れる信号フローダイアグラムは、ブロック

$B_1, \dots, B_N$

をそのダイアグラムの他の部分から分離し、それらの入力および出力を付加してブロック対角構造を作成することによりこの正準構造に変換できる。

30

【0050】

【数2】

$$C(s) = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N \end{bmatrix} \quad (\text{Eq. 2})$$

【0051】

(2)

その係数が、なんらかのパラメータ

40

$p_1, \dots, p_N$

の有理関数である伝達関数または状態空間モデルは、この正準構造形式に変換可能であり、ここでブロック

$B_1, \dots, B_N$

は次の形式である。

【0052】

## 【数 3】

$$B_j = \begin{bmatrix} p_j & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & p_j \end{bmatrix} \quad (\text{Eq. 3})$$

## 【0053】

(3) この正準表現は直列、並列、およびフィードバック相互接続において閉じている。すなわち、こうした正準構造の任意の相互接続は、より大きなブロックリストを備えた別の正準構造を生ずる。

10

## 【0054】

上述の性質(1) - (3)により、本発明の原理と整合性がある図7の正準構造を調整するための任意方法は、任意の次数および構造の線形制御要素を備えた実質的にいかなる制御構造にも確実に適用できる。

## 【0055】

調整可能ブロックをパラメータ化するための代表的な技法

代表的な実施形態は、システム内の調整可能な要素をパラメータ化して調整タスクを最適化問題として定式化するように構成できる。例えば、一実施形態では、調整されるシステム要素をパラメータ化できる。表1では、比例積分偏差(PID)ブロック、固定次数伝達関数、および固定次数状態空間ブロックなどの共通制御要素のパラメータ化を要約した。

20

## 【0056】

## 【表 1】

調整可能ブロック	パラメータ化
ゲインブロック	ゲイン行列のすべての項目がパラメータである。
PIDブロック $B(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1 + T_f s}$	Kp、Ki、Kd、およびTfがパラメータである。
固定次数伝達関数ブロック $B(s) = \frac{b_m s^m + \cdots + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_0}$	分子および分母係数がパラメータである。
固定次数状態空間ブロック $B(s) \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$	A、B、C、D 行列の項目がパラメータである。デフォルトでは、Aを三重対角に限定し、これは完全に一般的であって、必要なパラメータの数を減少させる。

30

40

表1 選択した制御要素のパラメータ化

## 【0057】

さらに、代表的な実施形態は、ソフトウェアベースのツールを提供かつ/または使用するように構成して、ユーザに基本的なパラメータを含む式を記述させることで制御要素のカスタムパラメータ化を作成させることができる。例えば、本発明の一実施形態で使用される代表的なソフトウェアツールは、ユーザが、例えば次のような付加的な構造を備えた調整可能ブロックを作成可能とする。

実数の値

$$a/(s+a)$$

でパラメータ化された低域フィルタa  
オブザーバ形式の状態空間コントローラ:

【 0 0 5 8 】

【 数 4 】

$$\begin{cases} \dot{x} = Ax + Bu + L(y - Cx - Dy) \\ u = -Kx \end{cases} \quad (\text{Eq. 4})$$

数式4において、パラメータはゲイン行列KおよびLである。

10

【 0 0 5 9 】

多目的サポートのための代表的な技法

一例として、次の2つのH 制約が与えられる。

$$\|T_1\|_\infty < 1, \quad \|T_2\|_\infty < 1$$

これには次のような2つの別個の閉ループ伝達関数に関わる。

【 0 0 6 0 】

【 数 5 】

$$\begin{cases} T_1 = F(H_1, C) \\ T_2 = F(H_2, C) \end{cases} \quad (\text{Eq. 5})$$

20

上記式で、F(…)は、図7で記述される一次分数変換(LFT)相互接続を示す。

【 0 0 6 1 】

【 数 2 】

$$C(s) = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_N \end{bmatrix} \quad (\text{Eq. 2})$$

上記式は、調整される構造化補償器を表現する。

【 0 0 6 2 】

30

2つのH 制約(

$$\|T_1\|_\infty < 1, \quad \|T_2\|_\infty < 1$$

)は、次の単一のH 制約に結合可能であり、

$$\|T\|_\infty < 1$$

これは次の集約伝達関数で結合できる。

【 0 0 6 3 】

【 数 6 】

40

$$T(s) = F\left(\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}, \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}\right) \quad (\text{Eq. 6})$$

【 0 0 6 4 】

C(s)が全次full-order)複数入力・複数出力(MIMO)コントローラである場合、従来のH  
合成技法は、得られるコントローラの

$$\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$$

50

ブロック対角構造のため、得られる問題を解くことができない。

【 0 0 6 5 】

従来のH 合成技法とは対照的に、本発明の代表的な実施形態は、

$$\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$$

ブロック対角構造およびブロック繰り返しをサポートする。従って、本発明の代表的な実施形態は、従来のH 合成技法ではうまくいかない次の多目的問題のような問題を解くために使用できる。

10

「

$$\|T_1\|_{\infty} < 1$$

および

$$\|T_2\|_{\infty} < 1$$

となるようにC(s)を調整せよ。」

或いは、同様に、

「

$$\|T\|_{\infty} < 1$$

20

となるようにC(s)を調整せよ。」

という問題も単一目的問題を解くことと変わらない。実際的な見知からは、代表的な実施形態により、ユーザは各二地点間伝達関数(例えば、ループ伝達関数)を独立して制約することができ、すべての要件を獲得する単一のMIMO制約を見つける必要がなくなる。これは多くのユーザにとって望ましい特徴であり、それは、すべての要件を獲得する単一のMIMO制約の発見は、交差項がユーザの問題設定方法と干渉するので、ユーザには困難なタスクとなることが多いためである。

30

【 0 0 6 6 】

ループ整形要件を自動的に定式化するための代表的な技法

ループ整形は、フィードバック制御システムを調整するための一般的な周波数領域技法である。一例として、単入力・単出力(SISO)フィードバックループに関し、ループ整形法は次からなる。

1. 設計要件を、開ループ応答L(s)の所望のゲイン特性について表現する(周波数の関数としてのゲイン)。例えば、ゲインは、良好な追跡および外乱阻止のためには低周波数において高く(>1)すべきであり、雑音およびモデリングエラーに対する不感受性のためには高周波数において低く(<1)すべきである。遷移周波数(ゲイン=1)は、交差周波数

$$\omega_c$$

40

と呼ばれ、制御システムの応答速度に直接影響する。

2.

補償器の調整可能パラメータを調節して、閉ループ安定性および適切な安定余裕を維持しつつ所望のループ整形にアプローチする。

【 0 0 6 7 】

マルチループ制御システムに関し、この手順は、後述する幾つかの修正を施してMIMO開ループ応答に適用できる。

【 0 0 6 8 】

直ぐ上のステップ(1)は、周波数領域技法に詳しい制御エンジニアには通常よく理解されている。例えば、積分動作を備えたフィードバックループの所望のループ整形は、図8

50



に示したプロット800で表現できる。一実施形態では、プロット800は、表示装置250を用いてGUIを介してユーザに表示できる。

【0069】

図8を参照すると、0dB交差周波数は  $\omega_c = 10$  (  $\omega_c$  は図8の矢印805にある ) であり、約  $1/10 = 0.1$  秒の応答時間に対応する図8では、応答時間は、フィードバックループが変化にいかにも速く反応するかを示すことができる。

$\omega_c$

を増大または減少させると、応答をそれぞれ速くまたは遅くできる。図8のループ形状は  $w < 10$  (領域810) について高いゲインを備え、 $w > 10$  (領域820) について低いゲインを備える。

10

【0070】

代表的な実施形態および技法は、所望ループ形状またはその近くに達するためシステム中の補償器の調整可能パラメータを調節するのに適している。本発明の実施形態で 사용되는技法は、ループ整形目標(すなわち、開ループゲインを整形して指定された特性に一致させること)を何らかのH<sub>2</sub> ノルム制約に定式化すると有利となることがある。ループ整形目標をH<sub>2</sub> ノルム制約に定式化するための幾つかの公知技法が存在するが、こうした変換は、ロバスト制御理論やH<sub>2</sub> 合成の訓練を経ていないエンジニアには困難を伴うことが多い従来のアプローチを採用する際にエンジニア(例えば、ユーザ)が直面する困難さを除去するため、代表的な実施形態では、この再定式化ステップを単純化する技法を用いる。

20

【0071】

図9は、本発明の諸側面を実装するために使用できるシステム900を示す。図900は、入力信号 $r$ 、出力信号 $y$ 、コントローラ $C(s)$ 910、およびプラント $G(s)$ 920と共にSISOおよびMIMOフィードバックループを含むことができる。さらに、システム900は信号 $u$ および $n$ も含む。図9において、 $C(s)$ 910は、1つまたは複数の調整可能ブロックを含むことができ、信号 $u$ および $y$ は同数の項目(すなわち、同じ数のコントロールと測定値)を含むものと仮定できる。

【0072】

30

図9では、開ループ伝達関数は次のように表現できる。

【0073】

【数7】

$$L = GC \quad (\text{Eq. 7})$$

さらに、閉ループ伝達関数は次の通りであって、

【0074】

【数8】

$$\begin{bmatrix} y \\ e \end{bmatrix} = \begin{bmatrix} T & -T & GS \\ S & -S & GS \end{bmatrix} \begin{bmatrix} r \\ n \\ d \end{bmatrix} \quad (\text{Eq. 8})$$

40

表記法は次の通り。

【0075】

【数9】

$$S = (I + L)^{-1}, T = LS \quad (\text{Eq. 9})$$

【0076】

目標ループ形状

$\rho(s)$

および外乱阻止率

$\beta$

が与えられているとき、次を考慮する。

【 0 0 7 7 】

【 数 1 0 】

$$X(s) = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} T & -T & GS \\ S & -S & GS \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\rho & 0 \\ 0 & 0 & 1/\beta \end{bmatrix} \quad (\text{Eq. 10})$$

10

【 0 0 7 8 】

次のH 制約により、

$$\|X\|_{\infty} < 1$$

以下が保証されることを確認できる。

【 0 0 7 9 】

【 数 1 1 】

20

$$\begin{cases} \|S(j\omega)\| < \min(1, \frac{1}{\rho(\omega)}) \\ \|T(j\omega)\| < \min(1, \rho(\omega)) \\ \|GS(j\omega)\| < \beta \min(1, \frac{1}{\rho(\omega)}) \end{cases} \quad (\text{Eq. 11})$$

【 0 0 8 0 】

ここで、

.

$$S \approx 1/L \text{ および } T \approx 1$$

30

であり、ループゲイン

$$\|L(j\omega)\|$$

は1より大きく、

.

$$T \approx L \text{ かつ } S \approx 1$$

であり、ループゲインは1より小さいので、

これは次に等しい。

【 0 0 8 1 】

40

【 数 1 2 】

$$\begin{cases} \|L(j\omega)\| > \rho(\omega) \text{ where } \|L(j\omega)\| > 1 \\ \|L(j\omega)\| < \rho(\omega) \text{ where } \|L(j\omega)\| < 1 \\ \|S(j\omega)\| < 1 \\ \|T(j\omega)\| < 1 \\ \|GS(j\omega)\| < \beta \min(1, \frac{1}{\rho(\omega)}) \end{cases} \quad (\text{Eq. 12})$$

【 0 0 8 2 】

式12の最初の2つの制約は、ループゲインを交差周波数から十分に大きくまたは十分に

50

小さくすることにより所望のループ形状を実現するものである。例えば、1つの実装例では、交差周波数から離れた上限値および下限値などのループゲインの所望値を含むことができる。式12の第3および第4制約は、SおよびTのゲインを交差近くで1に維持するよう作用でき、これは適切なオーバーシュートおよび安定余裕を確保するために重要である。最後に、図12の最後の制約により、ユーザは

$$\beta$$

を減少させることで外乱阻止特性を改善できる。

【0083】

図9に関する上述の説明を要約すると、次のH 制約

$$\|X\|_{\infty} < 1$$

は、所望のループ形状を実現し、適切な余裕を与え、

$$\beta$$

を介して外乱阻止を最適化することが示されている。これにより、ループ整形目標を、本発明の代表的な技法とともに使用するのに適したH 制約に変換するシステム化された方法がもたらされる。例えば、代表的な実施形態および技法により、ユーザからの次の基本的な情報のみを用いて動作可能なソフトウェアツールを作成できるようになる。

・目標ループ形状

$$\rho(s)$$

(一実施形態では、ゲインのみが重要となる場合もある。)

・調整可能ブロック

・閉ループシステムのブロックダイアグラムにおける信号r、d、yの位置。

【0084】

この情報から、本発明の諸側面を実装するソフトウェアツールは、本発明の諸側面と整合性がある構造化H 合成技法を用いて自動的にGおよびCを抽出し、X(s)を構築し、ブロックパラメータを調整できる。これらソフトウェアツールおよび技法は、所望の交差周波数

$$\omega_c$$

から適切なループ形状を自動的に生成することにより、かつ積分動作が必要かどうかを判断することにより、目標ループ形状

$$\rho(s)$$

の必要性をさらに無くすることができる。

【0085】

本発明のソフトウェアで実現された側面は、MIMO問題にも適用できる。例えば、代表的な技法を、それぞれが特定の関節を制御する4つの別個のフィードバックループを含んだロボットアームの制御に適用できる。MIMOの場合は、改良点としては、SISO問題で使われるループ形状とは異なるループ形状の使用が含まれる。

【0086】

図10は、本発明の諸側面に従ってMIMO問題を扱うことができるループ形状を示す。図10では、トレース1010は、概ね0dBゲインである平坦部1015を含む。平坦部1015は、一定範囲の交差周波数を考慮したものである。一定範囲の交差周波数を見込むことは、結合されたマルチループ制御システムにおいては、すべてのループが同一の周波数で0dB交差可能

10

20

30

40

50

とは限らないことに対応させる助けになる。

【0087】

本発明の側面を実装するためのソフトウェアに基づいたツール

本発明のソフトウェアで実現された実施形態は、一定の形式でかつ/または典型的なユーザにより容易に理解される用語を用いてユーザが情報を入力可能とするグラフィカルユーザインターフェース(GUI)などのインターフェースをユーザに提供する。さらに、ソフトウェアで実現された実施形態は、制御システムの迅速な設計および/または解析をサポートするアルゴリズムおよび処理技法を利用する。事実、本発明の実施形態を構成することにより、制御システムの効率的な設計および解析を促すような作業の流れをユーザが使用できるようにするインタラクティブな制御設計および解析をサポート可能である。

10

【0088】

例えば、代表的な実施形態が本明細書に記載の技法を実装するための1つまたは複数のソフトウェアツールを使用する場合は、ソフトウェアツールは、限定するわけではないがユーザに例えば次のようなサービスを提供できる。

1. 制御要素の構造を特定し、これら要素のパラメータ化を自動的に生成する。
2. カスタム要素およびパラメータ化を作成する。
3. 所与の制御アーキテクチャを図7の対応する正準構造にマッピングする。
4. 簡潔な設計要件をH制約に変換する。
5. 基調をなす構造化H合成問題を解くことによって制御要素を自動的に調整する。
6. 元々の制御要素について結果を提示する。
7. 通常の線形解析ツールとの厳密な統合化によって設計の妥当性検証を促進する。

20

【0089】

説明を明確にするため、代表的な実施形態および技法は、MATLABのプログラミング言語および/またはMathWorks社の制御システムツールボックスのようなMATLAB言語と互換性があるツールボックスと関連付けて説明する。本発明の実施形態および/またはこれら実施形態に関連した技法は、本発明の精神から逸脱することなく、他のソフトウェアパッケージおよび/またはアプリケーションなどを用いて、他のプログラミング環境で実装することもできる。例えば、実施形態および/または技法は、MATLAB互換製品/言語、Octave、Python、Comsol Script、National Instruments社からのMATRlXx、Wolfram Research社からのMathematica、Mathsoft Engineering & Education社からのMathcad、MaplesoftからのMaple、Imagine That社からのExtend、フランス国立情報学自動制御研究所(INRIA)からのScilab、Cadence社からのVirtuoso、Dynasim社からのModelicaまたはDymola、C/C++ライブラリを含むがそれらに限定されないアプリケーションを用いて実装できる。

30

【0090】

幾つかの実施形態では、本発明の技法を実装するための環境は、ユーザが、数学、科学、工学、医学、商業などに限定されないがそれらを含む分野に関連したタスクを、C++、C、Fortran、Pascalなどの従来のプログラミング言語でユーザがコードを開発することを要件とするような別の種類の計算環境で実行する場合と比べ、より効率的に実行可能とする計算環境を提供するハードウェアまたはハードウェア/ソフトウェアに基づいたロジックを含む。

40

【0091】

一実装例では、この環境は、当業者が慣れ親しんだ数学的表記法で問題および/または解を表現するのに使用できる動的型付き言語を含むことができる。例えば、この環境は配列を基本要素として使用でき、この場合、配列は次元設定を必要としない。これら配列を用いて、演算が一配列内の値など一組の値すべてに適用できるという意味で配列プログラミング(array programming)をサポートできる。配列プログラミングにより、配列に基づいた処理を、個別の非配列の明示的ループ(すなわちスカラー処理)を行うことなく、プログラマーにデータ集団全体を考慮しかつ処理させうる高レベルプログラミング技法またはモデルとして扱うことができることがある。

【0092】

50

さらに、この環境は、データ解析、データ視覚化、アプリケーション開発、シミュレーション、モデリング、アルゴリズム開発などのために使用可能な行列および/またはベクトル定式化を実行するために適合させてもよい。これらの行列および/またはベクトル定式化は、統計、金融、画像処理、信号処理、制御設計、生命科学、教育、離散事象解析および/または設計、状態ベース解析および/または設計などの多くの分野で使用できる。

#### 【0093】

所望であれば、本発明の側面と一貫性がある実施形態および技法は、The MathWorks社製のSimulink(登録商標)環境、Stateflow(登録商標)環境、SimEvents(商標)環境など、Simulink/Stateflow/SimEvents互換製品、Visual Solutions社製のVisSim、National Instruments社製のLabVIEW(登録商標)、Dynasim社製のDymola、Measurement Computing社製のSoftWIRE、DALSA Coreco社製のWiT、Agilent社製のVEE ProまたはSystemVue、PPT Vision社製のVision Program Manager、Khoros Research社製のKhoros、Gedae社製のGedae、INRIA製のScicos、Cadence社製のVirtuoso、IBM社製のRational Rose、Telelogic社製のRhopsody または Tau、カリフォルニア州立大学バークリー校製のPtolemy、或いは統一モデリング言語(UML)またはSysML環境の諸側面に限定されないがそれらを含む製品を用いてグラフィックベースの環境で実装できる。

10

#### 【0094】

調整可能ブロックを指定するための代表的な技法

本発明の実施形態により、ユーザはシステム中の調整可能ブロックを指定でき、ここで、調整可能ブロックは、最適性能およびロバスト性を実現するために調整される制御システムの構成要素である。例えば、制御システムの調整可能な構成要素は、本発明の実施形態における図7の正準構造におけるブロック

20

$$B_1, \dots, B_N$$

iに対応できる。

#### 【0095】

代表的な実施形態は、ゲイン、PID、伝達関数、および状態空間モデルなどの要素を制御するためのオブジェクトを用いることができる。さらに、これらのオブジェクトは、調整可能なパラメータに対応するように、またユーザによるパラメータの初期化、設定(例えば、固定)、および解放を可能とするために使用できる。

30

#### 【0096】

一例として、ユーザは、テキスト形式のユーザインターフェースなどの、2つの入力と2つの出力とを備えた調整可能なゲインを生成するインターフェースと対話できる。モデリング環境220はインターフェース1100を生成できる。例えば、インターフェース1100は、表示装置250を介してユーザに表示されるテキスト形式編集ウィンドウでよい。図11を参照すると、代表的な実施形態では、ユーザは、プロンプト(例えば>>)でコマンド1105を入力できる。

#### 【0097】

コマンド1105に応答して、モデリング環境220は、インターフェース1100を介してユーザに表示できるオブジェクト1110を返すことができる。一実施形態では、モデリング環境220を構成して、モデリング環境220と互換性がある他の種類のオブジェクトを表示するのに使用される形式に似た形式でオブジェクトG 1110を表示できる。例えば、オブジェクトG

40

1110は、数値でなく調整可能パラメータの記述を含みうる「Gain」属性を除けば、制御システムツールボックス内の通常の線形時間不変(LTI)オブジェクトに類似していてもよい。例えば、図12を参照すると、ユーザはコマンド1205を入力し、モデリング環境220はG.Gainの値を含みうる応答1210を返すことができる。

#### 【0098】

続けて図12を参照すると、応答1210はG.Gainのデフォルト値を含むことができる。代表的な実施形態では、デフォルト値は、ユーザが1つまたは複数の値をデフォルト値から別

50

の値に変更するまでモデリング環境220で使用する現在の値としてよい。例えば、ゲインのデフォルト/現在「値」は、応答1210において[0 0]である。ユーザが、コマンド1205および1220を入力することでゲイン行列の最初の項目を1に設定できるようにしてもよい。例えば、ユーザは次を入力できる。

```
>> G.Gain.Value(1) = 1
>> G.Gain.Free(1) = false
```

【0 0 9 9】

ユーザは次にG.Gainを入力でき、モデリング環境220は出力1225を返すことができる。図12では、出力1225は、この新しい制約(「値」のデフォルト設定からユーザ指定設定への変更)を反映するゲイン行列のパラメータ化を示すことができる。

【0 1 0 0】

代表的な実施形態では、ユーザは、調整可能PIDブロック、伝達関数ブロック、または状態空間ブロックなどの他の種類のブロックを、図13に示したようにそれぞれコマンド1305、1310、および1315を用いて、インターフェース1300内で作成できる。

【0 1 0 1】

代表的な実施形態は、ユーザに、典型的なまたは主たる必要性に対処する事前定義ブロックを与えることができる。ユーザによっては、事前定義ブロックが必要性を満たすには十分となることもある。しかし、他のユーザには、事前定義ブロックの構造を上回る構造を備えたブロックなど、より高度かつ/またはロバストな制御要素に対する必要性など別の必要性がある場合もある。

【0 1 0 2】

代表的な実施形態を適合して、低域フィルタブロック、ノッチフィルタブロック、オブザーバー形式の状態空間ブロックなどの事前定義ブロックとは異なるブロックを必要とするユーザの必要性をサポートできる。例えば、幾つかの実施形態は、事前定義ブロックとは異なる一組のブロックを提供するように構成できる。しかし、こうした一組のブロックは、一定のユーザの必要性を満たすには多数のブロックを含む必要がある場合もある。本発明の一実施形態では、ユーザが単純な代数を用いて自分自身のパラメータ化を構築可能とするフレームワークが提供される。このフレームワークによって、ユーザのすべての必要性を網羅するために多数のブロックの組を維持する必要性がなくなる。

【0 1 0 3】

一実施形態では、このフレームワークは次を含むことができる。

1.  
パラメータの基本概念を具体化する「パラメータ」オブジェクト。
2.  
パラメータへの通常の処理を図7の正準構造に自動的に変換するコンピュータ代数。

【0 1 0 4】

一例として、次の低域フィルタを考える。

【0 1 0 5】

【数 1 3】

$$F(s) = \frac{a}{s+a} = \frac{a/s}{1+a/s} \quad (\text{Eq.13})$$

上記式では、aが実パラメータである。式13は一次伝達関数である。しかし、上記で導入された(図13、コマンド1305)伝達関数ブロックは、同じ係数aが分母および分子に存在するという制約(言い換えれば、DC利得が1に制限される)を表現できない。代表的な実施形態では、ユーザは、次のようなコマンド1320(図13)を入力することでこの制御要素を指定できる。

```
>> s = tf('s');
>> a = realp('a',2);
>> F = feedback(a/s,1)
```

## 【 0 1 0 6 】

コマンド1320はモデリング環境220により処理され、@genssオブジェクト「F」が返され、ここでは返されたオブジェクトは、「a」を唯一のブロックとして図7の正準構造を具現化している。ユーザはインターフェース1300にコマンド1325をタイプでき、モデリング環境220は、次のような応答1330を返すことができる。

```
ans =
a: [1x1 realp]
```

## 【 0 1 0 7 】

本発明の実施形態は、「a」が通常の二倍値である場合と同じFを生成する構文を用いることができる。この特徴により、ユーザがパラメータと対話する新たなAPIを学ぶ必要性がなくなる。例として、制御システムツールボックスにおいて通常のLTIオブジェクトを操作するための構文は、パラメータおよびパラメトリックモデルを扱うように継ぎ目なく拡張される。本発明の実施形態は、所望時にパラメータと相互作用する代替的な構文をサポートすることもできる。例えば、パラメトリック低域フィルタF(s)を指定するための代替的な構文は、次のようにコマンド1335(図13)を介して表現できる。

```
>> a = realp('a',2);
>> F = tf(a,[1 a])
```

## 【 0 1 0 8 】

コマンド1335の代替的な構文は、制御システムツールボックスで通常の伝達関数を生成するための構文に類似しており(すなわち、対応しており)、これは次のようにコマンド1340を介して表現される。

```
>> a = 2;
>> F = tf(a,[1 a])
```

## 【 0 1 0 9 】

1335または1340を入力すると応答1345を返すことができ、これは得られた伝達関数をインターフェース1300を介して表示する。

## 【 0 1 1 0 】

本発明の実施形態を用いて、上述の例に比べより複雑な制御問題を解くこともできる。一例として、オブザーバー形式の状態空間コントローラは次のように与えることができる。

## 【 0 1 1 1 】

## 【 数 1 4 】

$$\begin{cases} \dot{x} = Ax + Bu + L(y - Cx - Dy) \\ u = -Kx \end{cases} \quad (\text{Eq. 14})$$

## 【 0 1 1 2 】

式14において、A、B、C、およびDは既知の行列であり、調整可能な変数はゲイン行列K(状態フィードバックゲイン)およびL(オブザーバーゲイン)である。コントローラのこの構造化状態空間パラメータ化は、例えばゲインスケジューリングに有用である。説明目的で、本発明の実施形態を使用する際には2つの測定値y、1つの制御信号u、および3つの状態xが存在すると仮定できる。ユーザは、コマンド1410(図14)を次のようにタイプすることでこの調整可能ブロックを作成できる。

```
>> K =
realp('K',zeros(1,3));
>> L =
realp('L',zeros(3,2));
>> OBC = ss(A-B*K-L*C+L*D*K,L,-K,0)
```

## 【 0 1 1 3 】

10

20

30

40

50

ここでユーザは次のコマンド1420をタイプできる。

```
>> OBC.Blocks
```

【 0 1 1 4 】

モデリング環境220はこのコマンドを処理し、インターフェース1400を介して次のように応答1430を表示できる。

```
ans =
```

10

```
K: [1x3 realp]
```

```
L: [3x2 realp]
```

【 0 1 1 5 】

図13に関連して述べた例と同様に、図14に関連して述べる例は、モデリング環境220で通常の状態空間モデルを作成するための構文に似た(すなわち、実質的に同一の)構文を用いている。一実施形態では、図14のコマンドを入力すると、次のように構造補償器を備えた図7の正準構造をカプセル化する@genssモデルが得られる。

【 0 1 1 6 】

【 数 1 5 】

20

$$C(s) = \begin{bmatrix} K & 0 & 0 & 0 & 0 & 0 \\ 0 & K & 0 & 0 & 0 & 0 \\ 0 & 0 & K & 0 & 0 & 0 \\ 0 & 0 & 0 & L & 0 & 0 \\ 0 & 0 & 0 & 0 & L & 0 \\ 0 & 0 & 0 & 0 & 0 & L \end{bmatrix} \quad (\text{Eq. 15})$$

【 0 1 1 7 】

図7を再び参照すると、図7の正準構造には2つの部分があり、具体的には(1)相互接続モデルH(s)および(2)調整可能ブロック

30

$B_1, \dots, B_N$

である。図8-14および対応する記載では、調整可能ブロック(上記文の項目(2))を指定するためのツールおよび技法を説明した。相互接続モデルを構築するためのツールおよび技法は後述する。

【 0 1 1 8 】

正準構造を導出するための代表的な技法

代表的な実施形態および技法を用いて、任意の制御構造と相互作用しかつ/または調整するための2つ以上のアプローチをサポートできる。例えば、幾つかの実施形態および技法は、(1)モデリング環境(例えば、Simulinkモデリング環境、Simulink互換環境などのグラフィカルモデリング環境)内に存在するツールを用いてH(s)を自動的に抽出することをサポートできる。例えば、H(s)は、Simulinkモデリング環境内にある線形化ツールを使用して自動的に抽出できる。そして(2)、H(s)は、テキスト形式計算および/またはモデリング環境(例えば、MATLABプログラミング環境、MATLAB互換環境など)に関連した標準的なコマンドを用いて自動的に構築できる。例えば、H(s)は、ユーザがMATLABプログラミング環境と対話することで、ブロックダイアグラムを構築するためのコマンドを用いて構築できる。

40

【 0 1 1 9 】

代表的な実施形態を用いて、グラフィカルモデリング環境のようなグラフィックベースの環境で調整可能ブロックを指定できる。一例として、ユーザはSimulinkモデリング環境

50



と対話して、制御システムで使用するための調整可能ブロックを指定できる。

#### 【 0 1 2 0 】

Simulink環境では、 $H(s)$ は、(1)調整されるブロック

$$B_1, \dots, B_N$$

(図7)と、(2)外部入出力信号 $w$ および $z$ (図7)とが分かれば完全に特定できる。調整可能ブロックがSimulinkブロックとしてブロックダイアグラムに存在する場合は、この情報は次のように容易に与えられる。

1.

ブロック

10

$$B_1, \dots, B_N$$

それぞれへのブロックパスのリスト。

2.

$w$ および $z$ を構成する信号がブロックダイアグラムを出入りする場所を指定する線形化入出力点。

#### 【 0 1 2 1 】

この情報が指定されると、LINLFTなどのコマンドが計算を実行し、 $H(s)$ をLTIモデルとして返す。上述の処理手順は、線形モデルおよび非線形モデルに適用できる。この処理手順が非線形モデルに適用されると、制御アーキテクチャの線形化モデルが生成される。

20

#### 【 0 1 2 2 】

MATLAB環境のようなテキスト形式プログラミング環境における $H(s)$ の導出は、調整可能ブロックの指定に関して上述したコンピュータ代数に依存することがある(例えば、図11-14およびに関連した記載)。代表的な実施形態により、ユーザは通常のLTIモデルを組み合わせるための典型的な演算を拡張して(例えば、MathWorks制御システムツールボックスで行うことができるように)、パラメータオブジェクト、パラメトリックブロック、および図7に示した種類のパラメトリックLFTモデルを扱うことも可能となる。典型的な演算処理の例には、限定するわけではないが次が含まれる。

30

- ・通常の代数: プラス、マイナス、乗算、除算、逆数など。
- ・相互接続演算: 入力または出力連結、付加、並列、直列、フィードバック、lft、接続など。

#### 【 0 1 2 3 】

代表的な実施形態は、オブジェクト指向フレームワーク並びに静的および動的の一次分数変換(LFT)モデル(図7に示したように、行列またはLTIモデルと、ブロック対角配列の調整可能ブロックとのLFT相互接続として定義される)を使用する。代表的な実施形態は、ユーザがダブル配列、LTIモデル、パラメータ、調整可能ブロック、およびLFTモデルを結合可能とするオブジェクトシステムを生成できる。例えば、LFTモデルは最も一般的なモデルであり、@genmat、@genss、および@genfrdのような、それぞれダブル行列、状態空間モデル、周波数応答データモデルなどの $H$ の3つの異なる数値表現に対応する3つの種類とすることができる。得られるオブジェクトシステムは閉じており、調整可能(パラメトリック)要素を備えたLTIモデルの実質的にいかなる組合せも、図7の正準構造を備えた@genssまたは@genfrdモデルを生成する。

40

#### 【 0 1 2 4 】

図15は、本発明の代表的な実施形態により提供されるオブジェクトベースのフレームワークを用いて対話可能な実行可能グラフィカルモデル1500を示す。図15は、複数の実行可能ブロックを備えることができるモデル1500を含んでいてもよい。スコープブロック1505および蒸留塔(プラント)ブロック1510(以下、プラントブロック1510と呼ぶ)などのモデ

50

ル1500内の幾つかのブロックは調整可能ではない。対照的に、ゲインブロック1515並びにPIコントローラブロック1520および1525などのモデル1500内の他のブロックは調整可能とすることができる。モデリング環境220の実施形態は、第一種類の視覚的識別子(線の太さ、シェーディング、色、フィルパターン、テキストパターンなど)を用いて調整可能ブロックを示し、第二種類の視覚的識別子を用いて非調整可能ブロックを示すように構成できる。

#### 【0125】

図15を続けて参照すると、プラントブロック1510のLTIモデルGをモデル1500内に設けることができ、ユーザはモデル1500と対話して、rからyへの閉ループ伝達のLFTモデルを構築できる。

10

例えば、ユーザは、次のようにコマンド1605および1610(図16A)を入力できる。

% 調整可能ブロックを指定する

```
>> DM =
ltiblock.gain('Decoupler',eye(2));
>> PI_L =
ltiblock.pid('PI_L','pi');
>> PI_V =
ltiblock.pid('PI_V','pi');
```

および

20

% 閉ループ伝達Tを導出する

```
>> C = append(PI_L,PI_V) * DM;
>> T = feedback(G * C , eye(2));
```

#### 【0126】

モデリング環境220は、コマンド1605および1610を処理し、@genssモデルTを生成できる。

ユーザはモデリング環境220と対話し、次のようにコマンド1615を入力することで、モデルTをその部分であるHおよびBに分解できる(図7に示したように)。

30

```
>> [H,B] = getLFTModel(T)
```

#### 【0127】

モデリング環境220はコマンド1615を処理し、次のように応答1620を返すことができる。

```
a =
      x1      x2
x1  -0.01333
0
x2
0  -0.01333
```

40

```
b =
      u1  u2  u3
u4  u5  u6
x1   0   0   2   0   0   0
x2   0   0   0   2   0   0
```

c =

50

```

x1      x2
yD  0.5853  -0.576
yB  0.7213  -0.7307
?
-0.5853    0.576
?
-0.7213    0.7307
?  -0.5853    0.576
?
-0.7213    0.7307

```

10

d =

```

u1
u2  u3  u4
u5  u6
yD  0   0   0   0   0   0
yB  0   0   0   0   0   0
?
1   0   0   0   1   0
?
0   1   0   0   0   1
?
1   0   0   0   0   0
?
0   1   0   0   0   0

```

20

【 0 1 2 8 】

連続時間モデル。

30

B =

```

[1x1 ltiblock.pid ]
[1x1 ltiblock.pid ]
[2x2 ltiblock.gain]

```

【 0 1 2 9 】

調整可能パラメータを最適化するための代表的な技法

一旦、ユーザが、モデル1500およびモデリング環境220と対話して、設計要件を重み付き閉ループモデルT(s)について次のH 制約

40

$$\|T(s)\|_{\infty} < 1$$

として定式化すると、ユーザは次のようにコマンドを入力することでオブティマイザを呼び出すことができる。

```
>> T = hinfstruct(T0,options)
```

ここで、T0は未調整閉ループモデルであり、optionsはオブティマイザのオプションセットである。モデリング環境220は、このコマンドを処理し、さらに、調整可能パラメータがオブティマイザにより発見された一定の値に設定された調整済み閉ループモデルTを返

50

す。例えば、一実施形態では、すべての調整可能パラメータは、オブティマイザ360により決定された最良値に設定できる。

#### 【0130】

ユーザは、1つまたは複数の調整可能ブロックの調整済み値にアクセスを望むこともあり得る。例えば、一実施形態では、ユーザは、次のようにコマンドを入力することで各調整可能ブロックの調整済み値にアクセスできる。

```
>> T.Blocks
```

#### 【0131】

場合によっては、C0は、1つまたは複数の調整可能ブロックを含む補償器からなることもある。こうした場合、ユーザは、本発明の代表的な実施形態を用いて、調整済みパラメータ値をC0に関しLETモデルに「プッシュ」できる。例えば、ユーザは次のコマンドをタイプできる。

```
>> C = replaceBlock(C0,T.Blocks)
```

モデリング環境220はこのコマンドを処理し、C0における関連するパラメータ値を、Tにおけるそれらの調整済み値と自動的に置換できる。

代表的な実施形態は、SimulinkまたはSimulink互換モデルのようなモデルにおけるブロックパラメータの調整の効率化を容易にするヘルパー関数などの関数を含むことができる。

一例として、かつSimulinkを参照すると、

- ・LINLETを拡張して、調整の対象として選択されたSimulinkブロックに基づいて適切な調整可能ブロックオブジェクト(ltblock.gain、ltblock.tf・・・)を自動的に選びかつ構成できる。

- ・ヘルパー関数を用いて、hinfstructにより計算された調整済みパラメータ値をSimulinkブロックにプッシュバックできる。

ヘルパー関数は、文書化された関数が動作する際に、それら文書化関数により使用される文書化されていない関数のような関数を含むことができる。

#### 【0132】

結果を検査するための代表的な技法

代表的な実施形態は、調整可能ブロックおよびLFTモデルを通常のLTIモデルのピアとして扱うことができる。こうすることにより、代表的な実施形態は、STEPまたはBODEなどの標準的なコマンドを用いて調整済み制御システムを直接解析できる。例えば、ユーザは次のコマンドをタイプできる。

```
>> bode(T) % Tのボード応答をプロットする
```

#### 【0133】

モデリング環境220はこのコマンドを処理し、表示装置を介してユーザに1つまたは複数のボードプロットを表示できる。

#### 【0134】

代表的な実施形態により、さらに、ユーザは、標準的なtf()、ss()、zpk()、frd()、pid()コマンドを用いて調整可能要素を通常のLTIモデルに変換できる。例えば、FがTを構成する調整可能ブロックの1つであれば、ユーザはFを次のようなコマンドを使って伝達関数に変換できる。

```
>> Ftuned = tf(T.Blocks.F) % Fの調整済み値を伝達関数として得る
```

#### 【0135】

上述のコマンドおよび技法は、ユーザが従来のツールと対話する際には利用できないロバストな調整機能をユーザに提供する。これら技法/コマンドは、テキスト形式またはグラフィカルなモデリング環境をユーザにとって過剰に複雑または混乱の原因とすることなくロバストな機能を提供する。

10

20

30

40

## 【 0 1 3 6 】

本発明の一実施形態を実施するための代表的な作業の流れ

図17は、F14ジェット戦闘機で使用されるアナログ自動操縦装置の代表的モデル1700を示す。本発明の代表的な実施形態および技法をモデル1700に適用することで、F14のアナログ自動操縦装置を調整できる。例えば、実施形態および技法は、Simulinkモデリング環境を用いてF14の長手方向軸に関して自動操縦装置を調整できる。モデル1700は、標準的なカスケードループ自動操縦装置を実装できるコントローラブロック1705を含むことができる。

## 【 0 1 3 7 】

図17を参照すると、モデル1700は、ピッチ速度 $q$ を指令する内側ループ1715と、迎え角 $\alpha$ を指令する外側ループ1715を含むことができる。

10

## 【 0 1 3 8 】

図18は、コントローラブロック1705をより詳細に示す。図18に示すように、ウィンドウ1705-Aは、図17のコントローラブロック1705に含まれる構成要素および接続を含む。図18では、この自動操縦装置は固定構造体を備え、調整の必要があるかもしれないゲインおよび前置フィルタからなる。例えば、調整可能ゲインブロック1820、1825、および1830は、三角形の形状およびシェーディングをつけて表示されており、このシェーディングはゲインブロックが調整可能であることをユーザに示すものとすることができる。前置フィルタ1805、1810、および1815は、四角形の形状およびシェーディングをつけて表示してよく、このシェーディングは前置フィルタブロックが調整可能であることをユーザに示すものとする。図18は、調整可能なシェーデッドPI補償器1840も含むことができる。本発明の実施形態は、調整可能なゲインブロックを示すために第1色および/またはシェーディング密度を用い、調整可能な前置フィルタブロックおよび/またはPI補償器ブロックを示すために第2色および/またはシェーディング密度を用いてもよい。

20

## 【 0 1 3 9 】

代表的な実施形態および技法により、ユーザは選択した制御要素を自動的かつ一緒に調整して所望レベルの性能およびロバスト性を実現できる。

## 【 0 1 4 0 】

図19を参照すると、ユーザは次のコマンド1905

```
s = tf('s');
```

```
pKa = realp('Ka',0);
```

```
pKq = realp('Kq',0);
```

```
pKf = realp('Kf',0);
```

```
pKi = realp('Ki',1);
```

```
pTal = realp('Tal',1);
```

```
pW1 = realp('W1',0);
```

```
pW2 = realp('W2',0);
```

30

を入力して図17および18の自動操縦装置のパラメータを定義できる。図19では、ユーザは、すべてのパラメータを実スカラーパラメータとして定義することを選択できる。コマンド1905では、ユーザは、ほとんどのパラメータをゼロに初期化する選択もできる。

40

## 【 0 1 4 1 】

ユーザはモデリング環境220と対話し、積分器と、アルファ前置フィルタと、ピッチ前置フィルタとをコマンド1905のパラメータを含む表現として生成できる。例えば、ユーザは、コマンド1910を次のように入力して積分器と、アルファ前置フィルタと、ピッチ前置フィルタとを生成できる。

```
Integrator = pKi/s;
```

```
% Ki/s
```

```
AlphaSensor
```

50

```
= tf(1,[pTal 1]);          %
1/(Tal*s+1)
PitchFilter = tf([1 pW1],[1 pW2]); % (s+W1)/(s+W2)
```

#### 【 0 1 4 2 】

オブジェクトを追跡するための代表的な技法

ユーザは、図17および18の自動操縦装置モデルと対話する際に、追跡オブジェクトをH制約として定式化することを望むかもしれない。所望の帯域幅は1 rad/s(ラジアン/秒)とすることができ、ユーザは、ほとんどオーバーシュート無しで迎え角アルファに操縦桿入力(パイロット要求)を追跡させたいと望むこともあろう。代表的な実施形態および技法を用いて、ユーザの要件を外側ループ(アルファ)1715(図17)の開ループ形状について定式化できる。

10

#### 【 0 1 4 3 】

ユーザは、次のコマンド1915を入力することで、自分が現実的と考える目標ループを定義できる。

```
wc = 1;          % 目標クロスオーバー
LS = (1+0.01*s/wc)/(0.01+s/wc);
bodemag(LS,{1e-3,1e3}), grid,
title('Target loop shape')
```

20

#### 【 0 1 4 4 】

モデリング環境220はコマンド1915を処理し、GUI2000を介して図20に示したボードプロットを表示できる。GUI2000は、帯域幅が概ね1 rad/s (0dBクロスオーバー)となり、かつ1% を超過しない可能性が高い定常状態エラー(すなわち、 $w=0$ で40dBゲイン)を保証するループ形状を表示する。外側ループは積分器を含んでいないので(積分器は図17のピッチ速度ループに含まれている)、ユーザは、 $w=0$ でより大きなゲインを現実的には期待できないことを理解するかもしれない。

#### 【 0 1 4 5 】

( $r$ ,  $n$ ,  $d$ )をアルファにマッピングする閉ループモデルであって、 $r$ がアルファコマンドであり、 $n$ がアルファの測定ノイズであり、 $d$ がアクチュエータに入る外乱である閉ループモデルを導出して、上述の代表的なループ整形技法が使用できるようにする必要があるかもしれない。代表的な一実施形態では、ユーザは、(1)モデル中の調整済みブロックを列挙し、(2)信号 $r$ 、 $n$ 、 $d$ 、アルファを線形化入出力(I/O)としてマーキングし、(3)LINLFTコマンドを用いて( $r$ ,  $n$ ,  $d$ )から $y=\alpha$ への閉ループ伝達のLEFモデルを抽出することができる。一例として、ユーザは、図21に示したコードを生成できる。例えば、ユーザは、次のようにテキストをインターフェース2100に入力することでコード部分2105および2110を作成できる。

30

```
% 調整済みブロック
TunedBlocks = {...
'f14/Controller/Alpha-sensor Low-pass
Filter',...
'f14/Controller/Pitch Rate Lead
Filter',...
'f14/Controller/Ka',...
'f14/Controller/Kq',...
'f14/Controller/Kf',...
'f14/Controller/Integrator'};
%図4の構造化補償器
C =
```

40

50

```
blkdiag(AlphaSensor,PitchFilter,pKa,pKq,pKf,Integrator);
```

```
linios = [...
linio('f14/Controller/Stick
Prefilter',1,'in') ; ... % r
linio('f14/Aircraft Dynamics
Model',4,'outin') ; ...% アルファ,n
linio('f14/Controller',1,'in') ]; % d
H = linlft('f14',linios,TunedBlocks);
% 閉ループ伝達[r;n;d] -> y=アルファ
Ta0 = lft(H,C);
【 0 1 4 6 】
```

10

ここで、ユーザはエラー信号 $e = r - y$ を必要とする場合があり、これは次を入力することとT10の出力として加えることができる。

```
T10 = [0 0 0;1 0 0] + [1;-1] * Ta0;
【 0 1 4 7 】
```

ユーザは、上述のループ整形重みを加えることでH 制約として定式化を完了できる。例えば、ユーザは、次のようにコマンド2205をインターフェース220(図22)に入力できる。

```
beta = 3;
Win = blkdiag(1,1/LS,beta);
Wout = blkdiag(1,LS);
T10 = Wout * T10 * Win;
T10.InputName = {'r','n','d'};
T10.OutputName = {'y','e'};
【 0 1 4 8 】
```

20

代替的な実施形態では、こうしたループ整形重みは、ユーザ入力を必要としなくてもモデリング環境2200によってプログラムにより付加させてもよい。

```
【 0 1 4 9 】
```

図21および22に関連した入力がモデリング環境220に入力されると、図20のループ整形目標は次のH 制約によって獲得できる。

30

$$\|T_1(s)\|_{\infty} < 1$$

```
【 0 1 5 0 】
```

ロバスト性目的を識別するための代表的な技法

図17のカスケードループ構造は、2つの測定値(アルファおよびq)と1つの制御(昇降舵たわみデルタ)とを備えたMIMO制御構造である。幾つかの例では、ユーザは各SISOループの安定余裕を評価しようとするかもしれないが、そうしたアプローチでは十分なMIMO余裕が保証できるとは限らない。本発明の実施形態によりサポートされたより信頼性および正確性が高いアプローチは、次の感度関数のピークゲインが1に近い状態を維持することを保証する。

40

```
【 0 1 5 1 】
```

```
【 数 1 6 】
```

$$S = (I + GK)^{-1} \quad (\text{Eq. 16})$$

式16では、Gはデルタを(アルファ, q)にマッピングし、Kは(アルファ, q)からデルタを生成する集約コントローラである。言い換えれば、第2の要件は次の通り。

```
【 0 1 5 2 】
```

## 【数 18】

$$\|T_2(s)\|_{\infty} < 1 \quad \text{ここで } T_2 = S \quad (\text{Eq. 18})$$

## 【0153】

上述のように、

$T_2$

の閉ループモデルは、感度関数 $S$ がプラントの出力(アルファ,  $q$ )において測定されることに注視することにより導出でき、かつLINLEFTを使用できる。例えば、ユーザはコマンド210(図22)を入力できる。

```
linios = [...
linio('f14/Aircraft Dynamics
Model',4,'inout') ; ...% アルファ
linio('f14/Aircraft Dynamics
Model',3,'inout') ]; %
q
H = linlft('f14',linios,TunedBlocks);
S0 = lft(H,C);
T20 = S0;
```

10

20

## 【0154】

代表的な調整技法

$T_2$ の閉ループモデルが導出されると、ユーザは、コントローラパラメータの調整を望むかもしれない。代表的な実施形態は、追跡およびロバスト性に関する $H_{\infty}$  制約の単一制約への結合を促進する。例えば、この単一の制約はユーザが次を定義すると表現できる。

```
T0 = blkdiag(T10,T20);
```

## 【0155】

モデリング環境220は、HINFSTRUCT ソルバに、オブティマイザ360を用いて5回の最適化を実行するよう命令できる。例えば、オブティマイザ360は、局所極小において早期終了の危険を軽減するため5つの異なる開始点からこれら5回の最適化を実行できる。さらに、オブティマイザ360は、高ゲイン設計を防ぐため、閉ループポールの大きさを50に限定できる。ユーザは、次のコマンドを入力することで上述の最適化を実行するようにオブティマイザ360を構成できる。

30

```
Options =
hinfstructOptions('Display','final','RandomStart',4,'SpecRadius',50);
```

## 【0156】

ユーザは、次のコマンドを入力することで指定した最適化を実行できる。

```
T = hinfstruct(T0,Options);
```

## 【0157】

モデリング環境220は、これらコマンドが実行されると調整済み目標  $T$  を返すことができる。

40

## 【0158】

解を検査するための代表的な技法

ユーザはコントローラ設計を妥当性検証したいと望むこともあり、代表的な実施形態では、ユーザが対話的に検証を実行できる。一実施形態では、この設計の検証は、調整済みパラメータ値を閉ループ伝達 $Ta_0$  (アルファループ) および $S$ (感度関数)にプッシュすることで実行できる。ユーザは、次のような2つのコマンドを入力できる。

```
Ta = replaceBlock(Ta0,T.Blocks);
```

50



S = replaceBlock(S0,T.Blocks);

【 0 1 5 9 】

モデリング環境220はこれらコマンドを処理し、アルファループの閉ループ応答をシミュレートできる。

【 0 1 6 0 】

図23は、モデル1700のアルファループの閉ループ応答を示す代表的なプロット2300を顯示する。

例えば、ユーザは次のコマンド

step(Ta(1,1)) % アルファへのアルファコマンド

10

を入力して、プロット2300を表示装置250を介して表示させられる。

【 0 1 6 1 】

さらに、代表的な実施形態では、ユーザは、Sのピークゲインが0dBに近くなることを検証できる。例えば、ユーザは次をタイプできる。

sigma(S)

% 感度関数のピークゲイン

【 0 1 6 2 】

モデリング環境220はこのコマンドを処理し、表示装置250を介して図24のプロット2400を表示できる。

20

【 0 1 6 3 】

さらに、代表的な実施形態では、モデリング環境220と対話することによって、ユーザは調整済みパラメータ(例えば、Kq)の値にアクセスできる。例えば、ユーザは、

T.Blocks.Kq.Value % Kqゲインの調整値

をタイプでき、このコマンドが処理されるとモデリング環境220は応答を生成する。例えば、次をユーザに表示できる。

ans =

4.3486e-001

30

【 0 1 6 4 】

代表的な処理

図25は、本発明の1つまたは複数の実施形態を実施するための代表的な処理を示す。ユーザは、1つまたは複数の調整可能な構成要素を含むモデルを作成できる(動作2505)。例えば、ユーザはグラフィカルモデルと対話でき、信号を表す線で接続された多くの構成要素を含む自由形式グラフィカルモデルを構築してよい。このモデルは、1つまたは複数のフィードバックループを含むことができ、さらに、プラントとして表現できる固定構成要素と、コントローラを介して表現できる1つまたは複数の調整可能な構成要素とを含むことができる。さらに、ユーザは、固定および調整可能な構成要素を生成するためのコマンドをユーザ自身が入力するテキスト形式環境を用いてモデルを生成できる。一実施形態では、本発明の一実施形態が既存のモデル(非一時的な記憶媒体から検索されたモデル)を受け取るような場合は、動作2505はオプションとしてもよい。

40

【 0 1 6 5 】

モデルが識別されると、このモデルに関連付けられた設計要件を受け取ることができる(動作2510)。例えば、ユーザは、このモデルに関する閉ループ応答の所望のゲインプロファイルを指定し、構造化コントローラにグループ化できるモデル内の調整可能ブロックを指定し、かつ/またはモデルの閉ループ表現への入力、出力、および外乱のモデル内の位置を識別できる。代替的には、ユーザは、このモデルに関する設計要件を指定するために使用できるテキスト形式環境におけるコマンドと対話してもよい。別の実施形態では、設

50

計要件を、APIなどを介してプログラムによってデータ格納から読み込んでもよい。

【0166】

設計要件は、本発明の代表的な実施形態を用いてH 制約に定式化できる(動作2515)。例えば、本発明の一実施形態は、ユーザ入力装置を介して受け取られた設計要件(例えば、ループ形状目標)を、モデルの構成要素の調整に用いるH 制約に自動的に変換できる。コンピュータ205は、H 合成技法を用いて処理演算を実行し、プラント、コントローラ、および伝達関数を抽出できる(動作2520)。一実施形態では、このコントローラは、ブロック対角構造に配置される調整可能構成要素の調整可能なパラメータを含むことができる。例えば、設計要件のH 制約への変換では、図7に示したように制御アーキテクチャを正準構造にマッピングすることが含まれる。

10

【0167】

変換後は、プラント、コントローラ、および伝達関数は、このモデルのパラメータを調整するために使用可能なオプティマイザと互換性がある形式とすることができる。この互換形式はこのオプティマイザに入力可能であり、このオプティマイザは、構造化コントローラに関連付けられた構成要素を調整できる(動作2525)。一実施形態では、このオプティマイザは、未調整構成要素を受け取ることができる。

【0168】

このオプティマイザは、調整済み構成要素を含む結果を生成できる(動作2530)。例えば、調整済み構成要素は、動作2510で受け取られた設計要件を満足しつつ、このモデルの安定性目標も満足できる。

20

【0169】

このオプティマイザが生成した結果は検証して、この結果によって、このモデルが所望の様態で動作可能となることを保証できる(動作2535)。例えば、結果の妥当性検査により、オーバーシュート余裕が超えられてしまうことがなく、所望の応答速度が達成され、安定余裕が維持され、所望のループゲインが達成され、所望の外乱阻止が達成されることなどが保証される。

【0170】

代表的なアーキテクチャ

図26は、図2のコンピュータ205を実装するのに使用可能な代表的なコンピュータアーキテクチャを示す。図26は、コンピュータ205に対応するエンティティの代表的なダイアグラムである。図示したように、このエンティティは、バス2610、処理ロジック2620、主メモリ2630、読出し専用メモリ(ROM)2640、記憶装置2650、入力デバイス2660、出力デバイス2670、および/または通信インターフェース2680を含むことができる。バス2610は、このエンティティの構成要素間での通信を可能とするバスを含むことができる。

30

【0171】

処理ロジック2620は、命令を解釈かつ実行可能なプロセッサ、マイクロプロセッサ、または他の種類の処理ロジック(書替え可能ゲートアレイ(FPGA)、グラフィック処理装置(GPU)、デジタル信号プロセッサ(DSP)、特定用途向けIC(ASIC)、特定用途向きプロセッサ(ASIP)、プログラマブル論理回路(PLD)等)を含むことができる。一実装例として、処理ロジック2620は、シングルコアプロセッサまたはマルチコアプロセッサを含むことができる。別の実装例では、処理ロジック2620は、単一の処理装置または、処理クラスタまたは計算グリッドなどの一群の処理装置を含むことができる。さらに別の実装例では、処理ロジック2620は、互いに対してローカルまたはリモートでよい多数のプロセッサを含むことができ、処理時に1つまたは複数のスレッドを使用可能としてよい。

40

【0172】

主メモリ2630は、ランダムアクセス記憶装置(RAM)か、情報および処理ロジック2620が実行するための命令を記憶できる他の種類の動的記憶装置を含むことができる。ROM2640は、ROM装置か、処理ロジック2620が使用するための静的情報および/または命令を記憶できる他の種類の静的記憶装置を含むことができる。記憶装置2650は、磁気固体および/ま

50

たは光学記録媒体並びにその対応するドライブ、または処理ロジック2620が使用するための静的情報および/または命令を記憶できる他の種類の静的記憶装置を含むことができる。

#### 【0173】

入力デバイス2660は、オペレータが情報を、キーボード、マウス、ペン、タッチパッド、加速度計、マイクロフォン、音声認識、カメラ、生体計測機構などに入力可能とするロジックを含むことができる。一実施形態では、入力デバイス2660は、入力デバイス240と一致していてもよい。

#### 【0174】

出力デバイス2670は、情報をオペレータに出力する表示装置、プリンタ、スピーカ、ハプティックインターフェースなどの機構を含むことができる。通信インターフェース2680は、このエンティティが他のデバイスおよび/またはシステムと通信可能とする送受信装置のようなロジックを含むことができる。例えば、通信インターフェース2680は、ネットワークを介して他の装置またはシステムと通信するための機構を含むことができる。

#### 【0175】

図26に示したエンティティは、主メモリ2630などのコンピュータ可読媒体に記憶されているソフトウェア命令の処理ロジック2620による実行に応答して演算処理を実行できる。コンピュータ可読媒体は、物理的(たとえば有体の)または論理メモリ装置として定義してよい。これらソフトウェア命令は、記憶装置2650などの他のコンピュータ可読媒体または他の装置から通信インターフェース2680を介して主メモリ2630に読み込み可能である。主メモリ2630に含まれたソフトウェア命令により、これらソフトウェア命令が処理ロジック2620により実行されると、この処理ロジックが本明細書に記載の技法を実行可能となる。代替的には、本明細書に記載の技法を実装するため、ソフトウェア命令に代えてまたはそれらと組み合わせてハードワイヤード回路を使用してもよい。よって、本明細書に記載した実装例は、ハードウェア回路またはソフトウェアの特定の組合せに限定されるものではない。

#### 【0176】

図26はこのエンティティの代表的な構成要素を示しているが、他の実装例では、このエンティティは、図26に示したものより少ない、異なる、または付加的な構成要素を含んでもよい。さらに別の実装例では、このエンティティの1つまたは複数の構成要素が、このエンティティの1つまたは複数の他の構成要素により実行されると説明されている1つまたは複数のタスクを実行してもよい。

#### 【0177】

代表的な分散実施形態

分散実施形態は、2つ以上の処理リソースを用いて処理を実行できる。例えば、幾つかの実施形態は、単一の処理装置内で2つ以上のコアを用いて処理を実行し、単一の筐体内に設けられた多数の処理装置にわたり処理を分散し、かつ/またはネットワークにより接続された多種類の処理ロジックにわたり処理を分散してもよい。

#### 【0178】

図27は、分散計算環境を用いて、クライアント装置(例えばコンピュータ205)のために非線形モデルのコントローラのインタラクティブな設計をサポートできる代表的なシステムを示す。システム2700は、コンピュータ205と、ネットワーク2730と、サービスプロバイダ2740と、リモートデータベース2750と、クラスタ2760とを含むことができる。図27の実装例は例示的なものであって、本発明の他の分散型実装例は、図27の代表的な構成と比べより多くの装置および/若しくは実体、より少ない装置および/若しくは実体、並びに/または構成が異なる装置/実体を含むことができる。

#### 【0179】

コンピュータ205は、グラフィカルユーザインターフェース(GUI)2710とモデリング環境220とを含むことができる。GUI 2710は、ユーザがコンピュータ205および/またはリモー

10

20

30

40

50

トデバイス(例えばサービスプロバイダ2740)と対話することを可能とするインターフェースを含むことができる。代表的な実施形態では、GUI 2710は、図4-6および15-18のインターフェースと似たものでよい。

【0180】

ネットワーク2730は、データ(例えば、パケットデータまたは非パケットデータ)を送送可能な任意のネットワークを含むことができる。ネットワーク2730の実装例は、インターネットプロトコル(IP)、非同期転送モード(ATM)、周期式光ファイバ網(SONET)、ユーザデータグラムプロトコル(UDP)、IEEE 802.10などの実質的にあらゆるネットワークプロトコルを用いて動作可能なインターネットなどのローカルエリア・ネットワーク(LAN)、メトロポリタン地域ネットワーク(MAN)および/またはワイドエリアネットワーク(WAN)を含むことができる。

10

【0181】

ネットワーク2730は、ルータ、スイッチ、ファイアーウォール、および/またはサーバ(図示しない)等のネットワーク装置を含むことができる。ネットワーク2730は、ワイヤード導体および/または光ファイバを用いたハードワイヤードネットワークとしてもよいし、かつ/または自由空間光、無線周波数(RF)、かつ/若しくは音響透過路を用いたワイヤレスネットワークでもよい。一実施例では、ネットワーク2730は、インターネットのような実質的に開放された公衆網でよい。別の実装例では、ネットワーク2730は、企業仮想ネットワーク(原語:corporate virtual network)のようなより制限されたネットワークでもよい。本明細書に記載したネットワークおよび/またはネットワーク上で動作する装置の実装例は、特定のデータタイプ、プロトコル、アーキテクチャ/構成などに限定されるものではない。例えば、一実施形態では、ネットワーク2730は、量子互換ネットワーキングプロトコルを使用する量子ネットワークでよい。

20

【0182】

サービスプロバイダ2740は、サービスを他の装置が利用可能とする装置を含むことができる。例えば、サービスプロバイダ2740は、1つまたは複数のサービスを、サーバおよび/または他の装置を用いる送信先に提供するエンティティを含む。サービスは、演算を行うため送信先によって実行される命令を含むことができる。代替的には、サービスは、送信先のために演算を行うため送信先のために実行される命令を含むこともできる。

【0183】

30

一例として、サービスプロバイダが、1つまたは複数のウェブベースサービスをコンピュータ205などの送信先に提供するウェブサーバを運用すると仮定する。このウェブベースサービスにより、コンピュータ205は、サービスプロバイダが運用するハードウェアを用いて電気的および/または機械的システムの分散シミュレーションを実行できる。例えば、コンピュータ205のユーザは、サービスプロバイダのハードウェアを使用してシステムモデルのPIDコントローラを対話的に設計できる。一実施例では、顧客(ユーザ)はサービス契約に基づいてサービスを受けることができる。サービス契約は、月毎の契約、利用毎の料金、サービスプロバイダ2740と顧客との間で交換された情報量に基づく料金、顧客が使用したプロセッササイクルの数に基づく料金、顧客が使用したプロセッサの数に基づく料金などの取り決めを含むことができる。

40

【0184】

リモートデータベース2750は、コンピュータ205などの他の装置が使用するための機械可読情報を記憶する装置を含むことができる。一実施形態では、リモートデータベース2750は、システムモデル、コントローラなどに関する情報を含んだデータ構造を記憶する記憶装置(例えば、ハードディスク、光ディスク、ソリッドステート記憶装置など)のアーレイまたはグリッドを含むことができる。

【0185】

クラスタ2760は、リモート処理(例えば、分散処理、並行処理など)を実行するために使用できる実行回路2770などの一群の処理装置を含むことができる。実行回路2770は、コンピュータ205のような要求装置のために処理演算を実行するハードウェアおよび/またはハ

50

ードウェア/ソフトウェアに基づく装置を含むことができる。一実施形態では、実行回路2770はそれぞれ部分的な結果を計算でき、これら部分的な結果はモデルの全体的な結果へ結合可能である。

#### 【0186】

代表的な実装例

一実施形態は、コンピュータ実装方法または非一時的コンピュータ可読媒体上に存在する機械実行可能命令を含むことができる。この方法および/または命令は、任意の制御構造において設計パラメータを調整するために実装でき、この場合は、コンピュータ実装方法は、例えばローカルマシンのようなスタンドアロンを用いてまたは分散環境(例えば、複数コンピュータ装置のクラスタまたはグリッド)において実施できる。

10

#### 【0187】

例えば、コンピュータ実装方法として実行する場合、この方法は、調整される1つまたは複数の自由パラメータを備えた1つまたは複数の調整可能な構成要素を識別することを含むことができる。これら調整可能な構成要素は、上記任意の制御構造のテキスト形式のまたはグラフィカルなモデルの一部とすることもできる。この方法は、上記1つまたは複数の調整可能な構成要素を含む上記モデルの幾つかの部分を識別することもさらに含むことができる。

#### 【0188】

この方法は、上記の任意制御構造を標準形式に変換することをさらに含むことができ、上記標準形式は、上記の任意制御構造における調整されない構成要素を含む集中線形モデルと、上記調整可能な構成要素を用いて形成された収集体とを含む。これら調整可能な構成要素から形成された収集体は、ブロック対角様式にグループ化または配置できる。この方法では、上記のブロック対角配列における構成要素は、上記のブロック対角配列における個々の構成要素またはパラメータがユーザに可視であり、アルゴリズムにとって利用可能であり、読み出しまたは書き込み可能であるという意味で、アクセス可能とすることができる。この方法は、上記収集体の1つまたは複数のメンバを所望する時に上記対角配列外に配置させることをサポートできる。

20

#### 【0189】

これら方法は、H またはH2目標または制約を使用して設計目標および/または設計要件を表現できる。この方法の一実装例では、これらH /H2目標または制約は、1つまたは複数の二地点間伝達関数に関する。例えば、これら目標または制約は、開ループまたは閉ループシステムにおける二地点間伝達関数に関することができる。この方法は、さらに、調整可能な構成要素をパラメータ化できる。例えば、この方法は、調整可能な構成要素を静的にまたは動的にパラメータ化できる。調整可能な構成要素がパラメータ化されると、この方法は、このパラメータ化に基づいてこれら調整可能な構成要素の自由パラメータと相互作用できる。

30

#### 【0190】

この方法は、調整器を用いて上記フィードバック制御構造を調整できる。例えば、一実装例では、この方法は、非平滑H /H2最適化アルゴリズムのクラスである調整器を用いて上記のフィードバック制御構造を調整できる。この調整器は上記の標準形式に対して作用でき、上記調整可能な構成要素がブロック対角形式となったときに調整可能パラメータに作用できる。さらに、上記の調整器は、これらパラメータを調整して上記のH /H2目標を最小化または最大化しかつ/または上記のH /H2制約を実現することができる。

40

#### 【0191】

このコンピュータ実装方法は、MATLAB互換環境のような技術計算環境で実装できる。さらに、この方法は、Simulink互換環境またはLabView互換環境の様なテキスト形式のまたはグラフィカルな環境で実装してもよい。さらに、幾つかの実装例は、例えば調整可能な構成要素、固定構成要素、接続、アルゴリズムなどを含みうるブロックの組と相互作用できる。

50

## 【0192】

このコンピュータ実装方法は、任意のフィードバック制御構造の対話的設計および調整をサポートするような状態で実装することも可能である。例えば、ユーザによる実時間の対話的設計および調整動作をサポートする方法を実装することもできる。この方法の実装例は、FPGA、ASIC、ASIP、PLD、GPU、DSP、マルチコアデバイス、分散計算リソースなどを用いて実装してもよい。

## 【0193】

多目的または多要件設計タスクを単純化するため、この方法の実装例では、さらに、ユーザまたは装置が複数の入出力ポイント(例えば、二地点間閉ループ伝達関数)で別個の目標または制約を指定できる。本発明の幾つかの実装例は、ユーザまたはユーザのグループにより用いられる従来の作業の流れをサポートすることによってこれを行う。この方法を拡張して、設計要件を二地点間伝達関数に適用することを含むことができ、この適用は、多目的および多要件設計タスクの定式化を促進する。この方法は、オブジェクト指向フレームワークに導入できる。

## 【0194】

本発明の幾つかの実施形態は、コンピュータ実装方法の動作および/または非一時的コンピュータ可読媒体上に存在する実行可能命令を含むことができる。例えば、媒体/実行可能命令として実装されているときは、この媒体は、実行されると本発明の諸側面を実行する命令を格納することができる。例えば、一実装例では、これら実行された命令は、ユーザに代わって方法または技法を実行できる。一実装例では、この媒体は、プロセッサ上で実行されると、調整可能な構成要素を静的に指定し、調整可能な構成要素を動的に指定し、かつ/または上記調整可能な構成要素のパラメータと相互作用するためのAPIを実装する実行可能命令を保持できる。一実装例では、このAPIはオブジェクトベースでもよい。

## 【0195】

実行されると、これら実行可能命令は、所定組の構成要素のパラメータ化を実現する所定のインターフェースを識別できる。さらに、これら命令は、一組の算術演算を実装しかつ/または一組のヘルパー関数を実装できる。さらに、上記の実行された命令は、これら算術演算およびヘルパー関数を用いて調整可能な構成要素を動的に作成できる。一実装例では、この動的な作成は、基本パラメータ構成要素および固定係数または固定構成要素を組み合わせることを含むことができる。幾つかの実施形態は、さらに、所望時に構成要素を静的に構成することをサポートできる。

## 【0196】

さらに、実行された命令は、調整可能な構成要素のパラメトリックモデルを生成できる。一実装例では、上記のパラメトリックモデルは、調整可能な構成要素の調整可能なパラメータに対応でき、ユーザのために受け取られた入力がこれら調整可能なパラメータと相互作用可能とする。例えば、ユーザ入力は調整可能なパラメータを初期化し、調整可能なパラメータを固定し、または調整可能なパラメータから選択したものを解放できる。

## 【0197】

本発明の別の実施形態は、コンピュータ実装方法としてまたは1つ若しくは複数の非一時的可読記憶媒体上に存在する実行可能命令を介して実装できる。この実施形態は、標準的な形式の任意フィードバック制御構造を構築し、かつ設計要件のH<sub>1</sub>/H<sub>2</sub>定式化で使用される二地点間伝達関数を指定するためのアプリケーションプログラムインターフェース(API)を、実装し、導入し、動作させたりできる。このAPIの実施形態は、所望であればオブジェクトベースでもよい。

## 【0198】

この実施形態は、入力構文を用いて算術演算子および/またはブロックダイアグラム演算を指定するユーザ入力機構と相互作用できる。例えば、この入力構文は、ユーザがPIDコントローラのような他の種類の制御構造と対話するためにこの構文を使ったことがあるという意味でユーザにとって慣れた構文とすることができる。この入力構文によって、ユーザは、線形時間不変モデル構成要素を入力でき、この線形時間不変モデル構成要素は線

形時間不変モデルにおいてソフトウェアベース入力インターフェースとして使用され、このソフトウェアベースのインターフェースは、調整可能なパラメータを備えた調整可能な構成要素を記述する。

#### 【0199】

この実施形態は、算術演算子およびブロックダイアグラム演算の一方または両方を用いて線形時間不変モデル構成要素とソフトウェアベースインターフェースとを結合できる。ブロックダイアグラム演算の例には、直列接続、並列接続、またはフィードバック接続を含むことができるがそれらに限定されない。この実施形態はユーザ入力機構と相互作用でき、この相互作用により、入力構文は標準形式の制御システム全体を漸進的に構築できる。この標準形式は、上記線形時間不変モデル構成要素および上記調整可能な構成要素を含むことができる。さらに、この入力構文を用いて、上記標準形式に基づいたパラメトリックモデルの生成を促進できる。このパラメトリックモデルは、オブティマイザと互換性がある形式とすることができ、その互換形式によりパラメトリックモデルをオブティマイザに入力できる。このパラメトリックモデルは、設計要件が満足されるような状態で調整されるように設定できる。

10

#### 【0200】

この実施形態は、上記標準形式および上記調整可能な構成要素と相互作用してH<sub>1</sub>/H<sub>2</sub>目標を最小化または最大化しかつ/またはH<sub>1</sub>制約を実現するオブティマイザを用いて、上記パラメトリックモデルを最適化することをサポートするよう構成できる。この実施形態は、ヘルパー関数をユーザ入力機構に与えるように構成してもよい。さらに、ユーザが上記入力構文を用いてAPIと対話するときに、この実施形態によれば、ヘルパー関数をユーザが入力機構を介してアクセス可能とする。また、この実施形態は、入力機構と互換性があり入力機構とともに動作できる関数を与えるように構成できる。これら関数を用いて制御システムに問い合わせしかつ/または制御システムを解析できる。

20

#### 【0201】

本発明の一実施形態は、ループ整形要件をH<sub>1</sub>またはH<sub>2</sub>定式にプログラムにより定式化するためのコンピュータ実装方法および/または非一時的記憶媒体上に存在するコンピュータ実行可能命令を含むことができる。この実施形態は、方法として実施される場合は、スタンドアローンまたは分散装置を用いて実装でき、かつ一人または複数のユーザのためにインタラクティブな(例えば、実時間)動作をサポートするよう構成できる。この方法は、開ループゲインの目標形状またはこの目標形状のプロキシを受け取ることを含むことができる。例えば、この目標ループ形状またはプロキシは、ユーザのために入力機構(例えば、キーボードまたはGUIなど)から受け取ることができるが、或いは、例えばコンピュータ可読記憶媒体からプログラムにより検索してもよい。

30

#### 【0202】

この方法は、制御構造およびフィルタを上記目標ループ形状またはプロキシから導出するソフトウェアツールと相互作用することを含むことができる。さらに、このソフトウェアツールは、上記の導出された制御構造およびフィルタから標準形式を構築し、H<sub>1</sub>/H<sub>2</sub>制約を上記の導出された制御構造およびフィルタから構築できる。この方法では、上記の標準形式およびH<sub>1</sub>/H<sub>2</sub>制約は、制御システムに関する設計要件を獲得できる。

40

#### 【0203】

別の実施形態は、標準形式の構造を利用するためのコンピュータ実装方法および/またはコンピュータ可読媒体上に存在する実行可能命令を含むことができる。上記標準形式の構造を利用することで、ユーザのためにインタラクティブな作業の流れをサポート可能な調整器アルゴリズムの性能を向上できる。この実施形態は、最適化手順の最中にオブティマイザにより与えられる調整可能なパラメータの値を受け取ることができる。最適化手順は、最適化動作を実行するための機械実装技法を含むことができる。

#### 【0204】

さらに、この実施形態は、構築手順/技法を用いて、上記調整可能なパラメータ値の標準形式の状態空間モデルを構築できるが、この構築は、上記調整可能なパラメータに関連

50

付けられた調整可能な構成要素を実装するためにソフトウェアオブジェクトを使用し、各ソフトウェアオブジェクトには、上記受け取られたパラメータ値に関する、その状態空間表現を与える役割を担わせ、上記調整可能な構成要素の集約状態空間行列を生成するため状態空間行列を集約し、かつ/または上記調整可能な構成要素の集約状態空間行列を、集中プラントモデルの状態空間表現と組み合わせて、上記標準形式の所望の閉ループ状態空間モデルを得ることにより実行される。

#### 【0205】

この実施形態は、上記構築手順の中間結果をキャッシュしてその後の勾配計算を加速することができる。勾配情報は、上記目標と制約とを微分するために連鎖法則などの法則を適用することと、前記標準形式の状態空間モデルを構築するのに使用されたものと同じソフトウェアオブジェクトを使用し、各ソフトウェアオブジェクトには、上記受け取られたパラメータ値に関してスカラー値関数の勾配を与える役割を担わせ、このスカラー値関数が上記法則適用の副産物であり、かつ/または上記キャッシュされた中間結果を用いて、各調整可能な構成要素により与えられた勾配データを上記目標および制約の全体的な勾配に効率的に集約することと、によって計算できる。

10

#### 【0206】

この実施形態は、プロセッサのために付加的な計算を要求することなく勾配情報を利用可能とすることができる。例えば、勾配情報は、勾配情報を利用可能にしないで要する計算コストと比較して、追加の計算コストなしで利用可能とすることができる。さらに、この実施形態は、上記調整可能なパラメータの所与の値に関する閉ループシステムを実質的に計算コストなしで導出できる。例えば、このコストは、それを含めてもがシステム設計、モデリング時間などに関連した計算予算に悪影響を与えないという意味で無視しうるほど小さい。

20

#### 【0207】

コンピュータ実装方法として実現されている一実施形態は、任意の制御構造において設計パラメータを調整できる。上記コンピュータで実行されるこの方法は、1つまたは複数の調整可能な構成要素を得ることができる。例えば、上記方法は、調整される構成要素を、本発明を実装するコンピュータに対してローカルまたはリモートで実行されるモデルから得ることができる。上記方法は、上記制御構造の調整される構成要素を固定ダイナミクスから分離することによって上記制御構造を変換できる。一実施形態では、固定ダイナミクスは、固定および/または既知の構成要素を含むことができる。この方法は、任意システムにおける1つまたは複数の二地点間伝達関数に関するこれらH またはH2目標または制約を使用できる。一実施形態では、これらH 若しくはH2目標または制約を使用して設計目標または設計要件を表現できる。さらに、この方法は、非平滑H またはH2最適化アルゴリズムに基づいた調整器を用いて上記制御構造を調整できる。例えば、上記調整器は、一組の非平滑H またはH2最適化アルゴリズムから選択でき、その場合、上記調整器は、上記パラメータを調整して上記H またはH2目標を最適化しまたは上記H またはH2制約を実現する。

30

#### 【0208】

上記コンピュータ実装方法は、キーボード、マクロフォン、接触表示装置などの入力機構から命令を検索することができる。上記命令は、1つまたは複数の調整可能な構成要素および/または固定構成要素を指定できる。さらに、上記方法は、1つまたは複数の二地点間伝達関数を識別でき、上記制御構造を変換するために上記二地点間伝達関数を用いることができる。一実装例では、上記方法は、上記制御構造を、線形モデルおよび調整可能な構成要素を含むことができる標準形式に変換できる。例えば、上記線形モデルは、調整不可能な構成要素を含むことができる。さらに、上記標準形式は、ブロック対角様式で配置された上記調整可能な構成要素もしくはパラメータの一部またはすべてを含むこともできる。

40

#### 【0209】

上記コンピュータ実装方法は、上記調整可能な構成要素をパラメータ化でき、該パラメ

50



ータ化に基づいて上記調整可能な構成要素と相互作用できる。上記方法で使用される調整可能な構成要素は、1つまたは複数の自由パラメータを含むことができ、上記方法は、上記自由パラメータと相互作用できる。上記方法は、フィードバック、フィードフォワード制御構造などの様々な構成の制御構造で動作可能である。さらに、上記方法は、集中線形モデルなどの、一定の種類の線形モデルに還元可能なモデルにおいて動作可能である。

#### 【0210】

一実施形態は、プロセッサ上で実行されると、アプリケーションプログラムインターフェース(API)を実装する実行可能命令を保持する1つまたは複数の非一時的コンピュータ可読媒体を含むことができる。上記APIを用いて、調整可能な構成要素を静的に指定し、調整可能な構成要素を動的に指定し、かつ/または上記調整可能な構成要素のパラメータと相互作用することができる。上記媒体は、例えば、上記命令が計算装置上で実行される際に上記方法を実行する1つまたは複数の実行可能命令を保持できる。上記方法は所定のインターフェースを識別する段階を含むことができ、ここで前記所定のインターフェースは、ブロックセット、ライブラリなどを介して与えられる構成要素のような所定組の構成要素のパラメータ化を実現する。

上記方法は、算術演算またはヘルパー関数を用いて調整可能な構成要素を動的に生成できる。さらに、上記方法は、上記調整可能な構成要素の調整可能パラメータと相互作用する上記調整可能な構成要素のパラメトリックモデルを生成できる。

#### 【0211】

上記方法は、一組の前記算術演算または前記ヘルパー関数を実装する段階をさらに含むことができる。上記算術演算またはヘルパー関数を用いて、所望時に前記調整可能な構成要素を動的に作成できる。上記方法では、上記パラメトリックモデルは、ユーザのために動作する入力機構などの入力機構に関連付けられた命令に応答して、調整可能なパラメータと相互作用できる。一実施形態では、上記入力機構を介して受け取られた命令は、上記調整可能なパラメータの初期化と、上記調整可能なパラメータの固定と、上記調整可能なパラメータから選択したものの解放とのうち1つまたは複数とに関連付けることができる。

#### 【0212】

一実施形態は、プロセッサ上で実行されると、アプリケーションプログラムインターフェース(API)を実装する実行可能命令を保持する1つまたは複数の非一時的コンピュータ可読記憶媒体を含むことができる。例えば、これら命令が実行されると、コンピュータが、前記APIを実装するための方法を実行できる。一実施形態では、上記方法は、線形時間不変モデル構成要素と、モデル内の調整可能な構成要素などの、調整可能な構成要素を記述する1つまたは複数のソフトウェアベースインターフェースとを結合する段階を含むことができる。上記方法は、上記線形時間不変モデル構成要素および上記調整可能な構成要素を含む制御システムの標準形式を構築し、上記調整をサポートするためパラメトリックモデルを生成できる。例えば、上記方法は、上記標準形式に基づいてパラメトリックモデルを生成でき、上記調整動作に基づいて設計要件を満足させるため上記パラメトリックモデルを用いることができる。

#### 【0213】

さらに、上記方法は、オブティマイザを用いて、上記パラメトリックモデルを最適化することもできる。一実施形態では、上記オブティマイザは、上記標準形式および上記調整可能なパラメータと相互作用して、H またはH2目標を最小化/最大化しまたはH またはH2制約を実現できる。さらに、上記方法は、上記線形時間不変モデル構成要素と、調整可能な構成要素を記述する1つまたは複数のソフトウェアベースインターフェースとを結合できる。この実施形態では、上記結合動作は、1つまたは複数の算術演算またはブロックダイアグラム演算の使用を含むことができる。一実施形態では、ブロックダイアグラム演算は、直列接続、並列接続、またはフィードバック接続を含むことができる。

#### 【0214】

一実施形態は、プログラムによってループ整形要件を定式化するための実行可能命令を

保持する1つまたは複数の非一時的コンピュータ可読媒体を含むことができる。この実施形態は、上記命令が計算装置上で実行されたときに一方法を実装できる。例えば、上記方法は、制御構造を導出し、フィルタを導出し、上記導出された制御構造およびフィルタから標準形式を構築し、上記導出された制御構造およびフィルタからH またはH2制約を構築するソフトウェアツールと相互作用できる。上記方法では、上記標準形式および上記H制約または上記H2制約は、制御システムに関する設計要件を獲得できる。

#### 【0215】

一実施形態は、実行可能命令を保持する1つまたは複数の非一時的コンピュータ可読媒体を含むことができる。上記命令が計算装置上で実行されたときに、上記方法は上記装置によって実装されるようにしてもよい。上記方法は、構築手順を用いて、調整可能なパラメータ値の標準形式の状態空間モデルを構築する段階を含むことができる。上記方法は、標準形式を備えた所望の状態空間モデルを得るために、調整可能なパラメータを備えた調整可能な構成要素の集約状態空間行列を、プラントモデルの状態空間表現に組み合わせることにより上記状態空間モデルを構築できる。例えば、上記方法は、集中プラントモデルを用いて上記標準形式を備えた上記状態空間モデルの取得を促進してもよい。上記方法は、目標または制約を微分し、ソフトウェアオブジェクトにより、スカラー値関数の勾配を与え、および/または調整可能な構成要素により与えられた勾配を前記目標および制約の全体的な勾配に集約することにより、勾配を計算できる。

10

#### 結論

20

#### 【0216】

幾つかの実装例によれば、ユーザは、ユーザが慣れ親しんだ特徴を用いてシステムモデルの設計コントローラをインタラクティブに設計できる。

#### 【0217】

本発明の代表的な実施形態の上述の説明は、本発明を例証および記述するものであるが、すべてを網羅することを意図したものでなく、本発明を開示された通りの形式に限定することを意図したものでもない。修正および変更は上述の教示を参照すれば可能であるし、本発明を実行することで発見されることもある。例えば、一連の動作を図25に関して上述したが、これら動作の順序は、他の実現例では本発明の原理に一致するように修正できる。さらに、非独立動作は並列で実行してもよい。

30

#### 【0218】

さらに、本発明の原理と矛盾しない実現例も、図面に示しかつ本明細書に記載されたものの以外の装置および/または構成を用いて、本発明の精神から逸脱することなく実現できる。具体的な導入様態および/または用途によっては、装置および/または構成要素を、図2、3、26、または27に示した実現例に追加かつ/またはそれらから削除してもよい。また、開示した実現例は、特定の組合せのハードウェアに限定されるものではない。

#### 【0219】

さらに、本発明の特定の部分は、1つまたは複数の機能を実行する「ロジック」で実装できる。このロジックは、ハードワイヤード論理回路、特定用途向けIC、書替え可能ゲートアレイ、およびマイクロプロセッサなどのハードウェア、ソフトウェア、またはハードウェアとソフトウェアとの組合せを含むことができる。本発明の説明で使用されたいずれの要素、動作、命令も、特に明記されていなければ本発明に不可欠で必須なであると解釈すべきではない。また、本明細書では、「1つの(原語:a)」という冠詞は、1つまたは複数の要素を含むことを意図している。1つの要素のみを意図する場合は、「1つの(原語:one)」または他の用語を使用する。さらに、本明細書では、「に基づいて」という用語は、特に明記されていなければ「少なくとも部分的には~に基づいて」を意味するよう意図されている。

40

#### 【0220】

本明細書で用いた見出しおよび小見出しは、本明細書を小区分に分割することにより読者を補助することを意図している。これらの見出しおよび小見出しは、本発明の範囲を限

50

定したり、本発明の特徴を定義したりするものと解釈すべきではない。

【図 1】

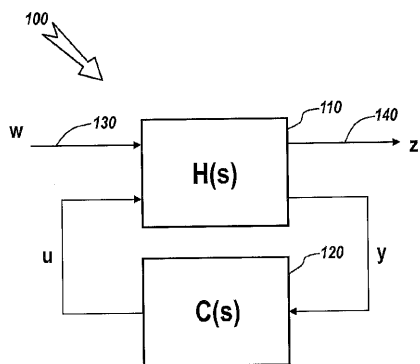


図 1  
(従来技術)

【図 2】

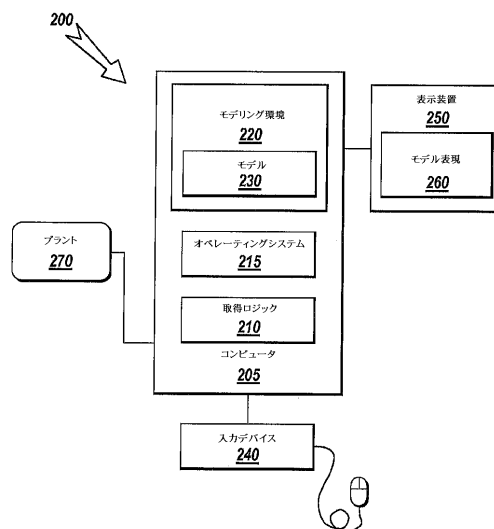


図2

【図 3】

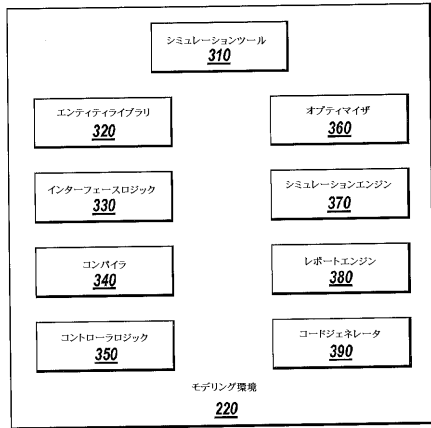
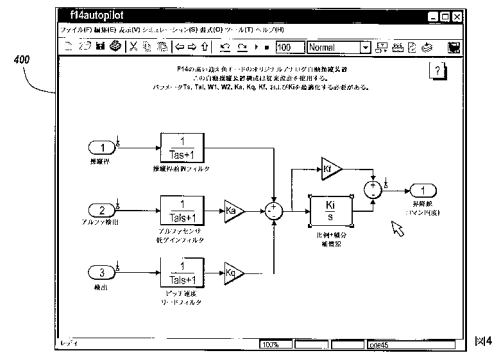


図3

【図 4】



【図 5】

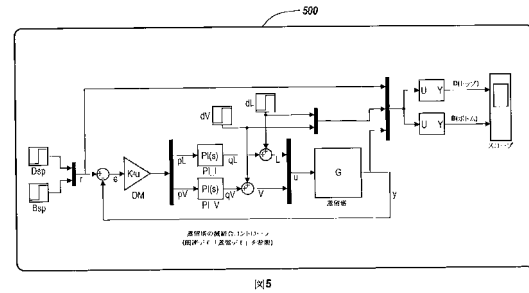
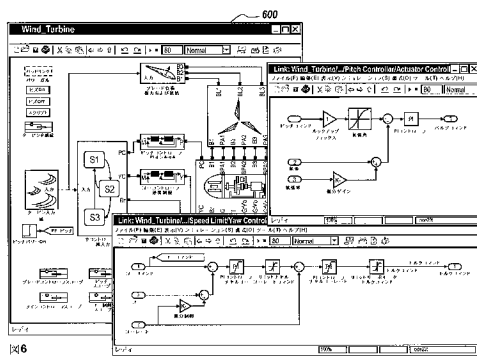


図5

【図 6】



【図 7】

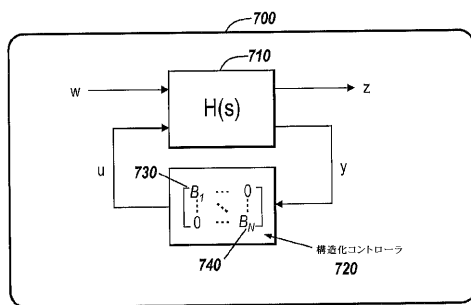


図7

【図 8】

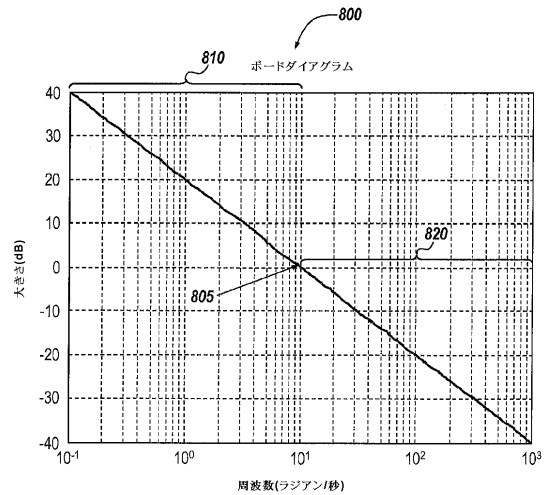


図8

【図 9】

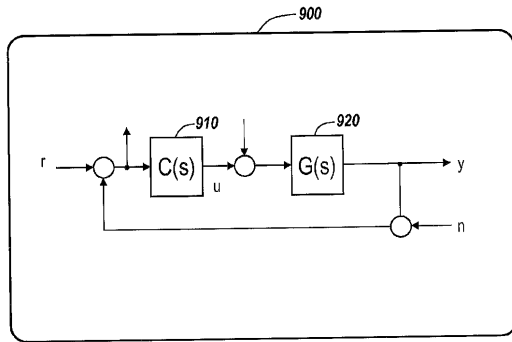


図9

【図 10】

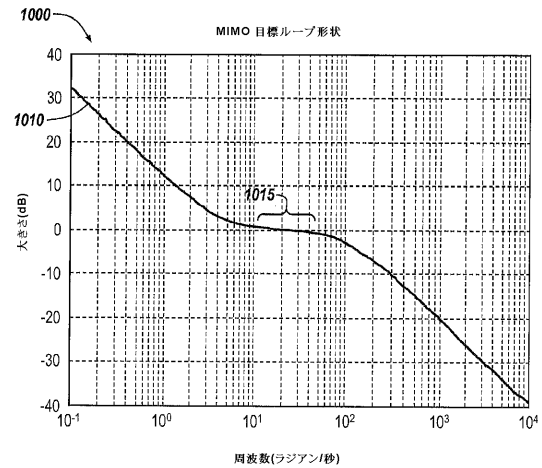


図10

【図 11】

```

>> G = ltiblock.gain('G',1,2) ~1105

G =

    ltiblock.gain
    Package: ltiblock

    Properties:
        Gain: [1x1
    param.Continuous]
        Ts: 0
        TimeUnit: ''
        InputName: {2x1 cell}
        InputUnit: {2x1 cell}
        InputGroup: [1x1 struct]
        OutputName: {''}
        OutputUnit: {''}
        OutputGroup: [1x1 struct]
        Name: 'G'
        Notes: {}
        UserData: []
  
```

図11

【図 12】

```

>> G.Gain ~1205

ans =

    Name: 'Gain'
    Value: [0 0]
    Minimum: [-Inf -Inf]
    Maximum: [Inf Inf]
    Free: [1 1]
    Scale: [1 1]
    Info: [1x2 struct]

>> G.Gain.Value(1) = 1 ~1215
>> G.Gain.Free(1) = false ~1220
>> G.Gain

    Name: 'Gain'
    Value: [1 0]
    Minimum: [-Inf -Inf]
    Maximum: [Inf Inf]
    Free: [0 1]
    Scale: [1 1]
    Info: [1x2 struct]
  
```

図12

【図 13】

```

1305 >> C = ltiblock.pid('C', 'pi') % 閉ループ伝達関数
1310 >> F = ltiblock.tf('F', 2, 3) % 2次微分と3次の極をもつ伝達関数
1315 >> MC = ltiblock.ss('MC', 3, 2, 2) % 3つの状態をもつ伝達関数
% 2つの入力と2つの出力

>> s = tf('s'); % s変数
>> a = realp('a', 2); % 1つの状態をもつ伝達関数
>> F = zerosblock('F', 2, 1) % 2つの入力と1つの出力

>> F.Blocks ~ 1325
ans =
a: [1x1 realp] ~ 1330
>> a = realp('a', 2); % 1つの状態をもつ伝達関数
>> F = tf('F', [1 a]); % 伝達関数aの逆伝達関数

>> a = 2;
>> F = tf('F', [1 a]); ~ 1340
% 伝達関数
2
s + 2 ~ 1345

```

図13

【図 14】

```

1400 >> K = realp('K', zeros(1, 3)); % 1つの状態をもつ伝達関数
>> L = realp('L', zeros(3, 2)); % 3つの状態をもつ伝達関数
>> OBC = ss(A-B*K-L*C+L*B*K, L, -K, 0)

>> OBC.Blocks ~ 1420
ans =
K: [1x3 realp] ~ 1430
L: [3x2 realp]

```

図14

【図 16 A】

```

1600 % 調整可能ブロックを指定
>> DM = ltiblock.gain('Decoupler', eye(2)); ~ 1605
>> PI_L = ltiblock.pid('PI_L', 'pi');
>> PI_V = ltiblock.pid('PI_V', 'pi');

% 用ループ伝達関数を導出
>> C = append(PI_L, PI_V) * DM; ~ 1610
>> T = feedback(G * C, eye(2));

>> [H, B] = getLFTModel(T) ~ 1615
a =
x1 x2
x1 -0.01333 0
x2 0 -0.01333

b =
u1 u2 u3 u4 u5 u6
x1 0 0 2 0 0 0
x2 0 0 0 2 0 0 ~ 1620

c =
x1 x2
yD 0.5853 -0.576
yB 0.7213 -0.7307
? -0.5853 0.576
? -0.7213 0.7307
? -0.5853 0.576
? -0.7213 0.7307

```

図16A

【図 15】

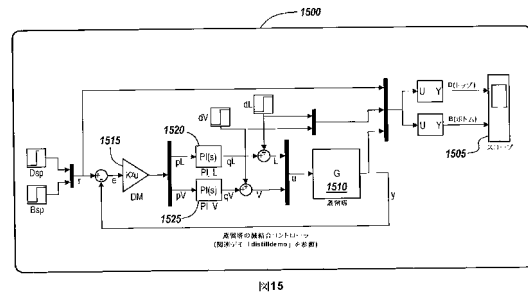


図15

【図 16 B】

```

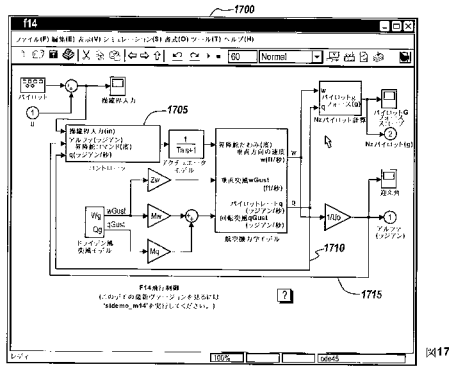
1600 d =
u1 u2 u3 u4 u5 u6
yD 0 0 0 0 0 0
yB 0 0 0 0 0 0
? 1 0 0 0 1 0
? 0 1 0 0 0 1
? 1 0 0 0 0 0
? 0 1 0 0 0 0

連続時間モデル
B =
[1x1 ltiblock.pid]
[1x1 ltiblock.pid]
[2x2 ltiblock.gain] ~ 1620 (図16Aからの続き)

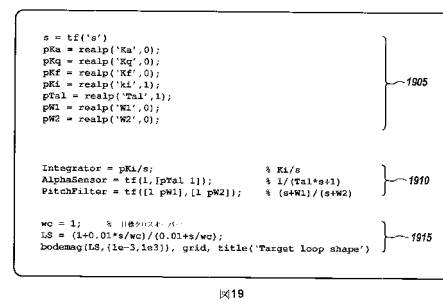
```

図16B

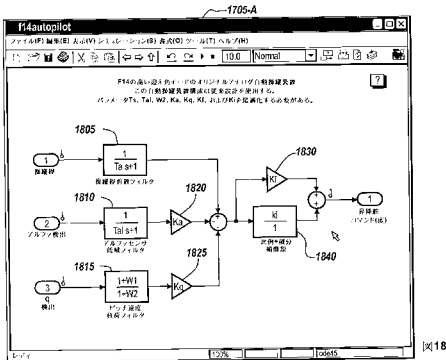
【図 17】



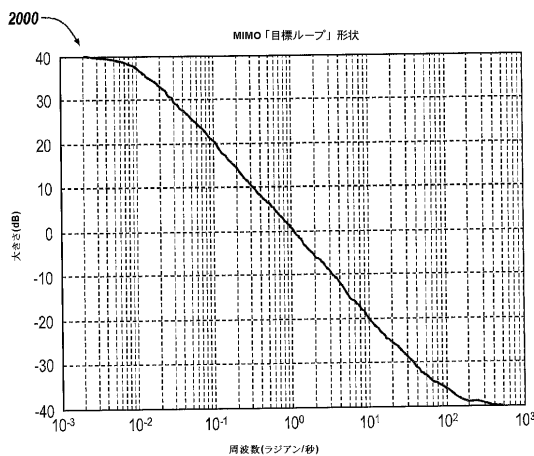
【図 19】



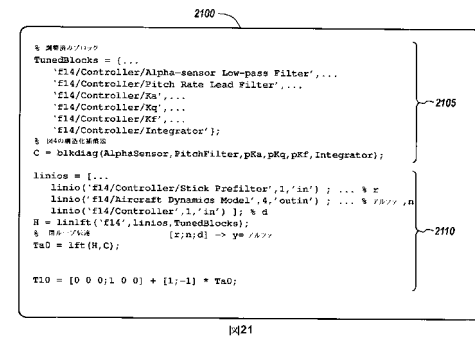
【図 18】



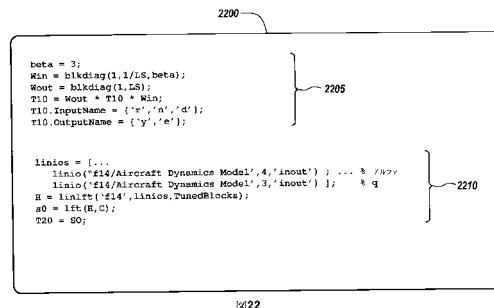
【図 20】



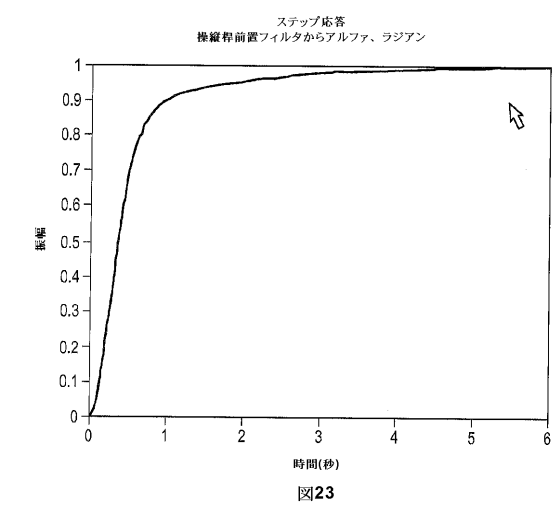
【図 21】



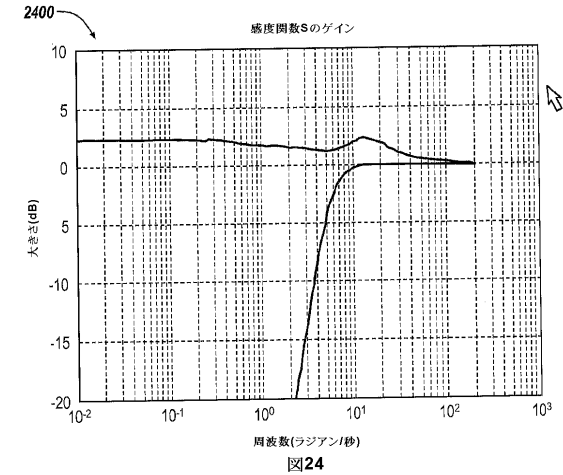
【図 22】



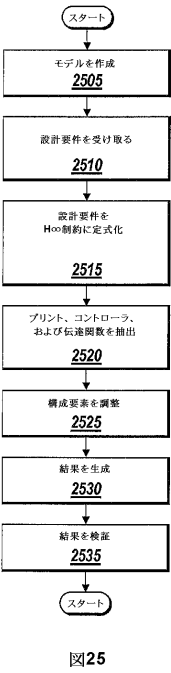
【 図 2 3 】



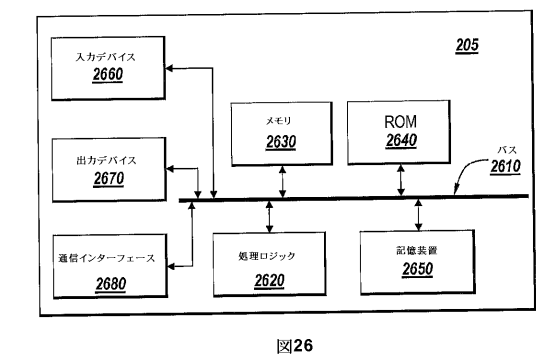
【 図 2 4 】



【 図 2 5 】



【 図 2 6 】





【図 27】

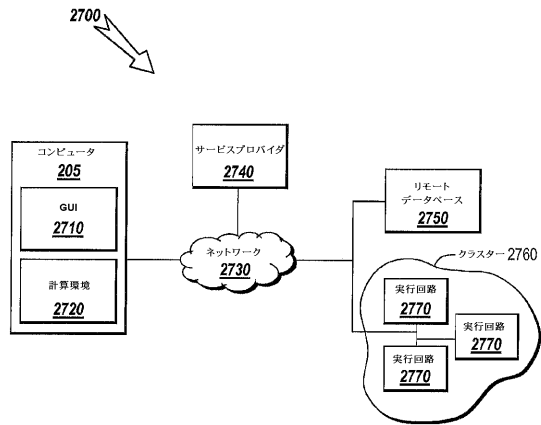


図27

## 【 国際調査報告 】

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2011/054008

## A. CLASSIFICATION OF SUBJECT MATTER

INV. G05B13/04  
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>P. Gahinet, A. Nemirovski, A. Laub, M. Chilali: "LMI Control Toolbox, User's Guide", May 1995 (1995-05), MathWorks, XP002665662, pages 6-1-6-24, pages 10,15,19; figures 3,7,8,12</p> <p>----- -/--</p>	<p>1-17, 31-34</p>

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

## \* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

20 December 2011

Date of mailing of the international search report

16/03/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer

Sundin, Martin

## INTERNATIONAL SEARCH REPORT

International application No

PCT/US2011/054008

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	M. MAMMADOV, R. ORSI: "A Nonsmooth Optimization Approach to H-infinity Synthesis", PROCEEDINGS OF THE 44TH IEEE CONFERENCE ON DECISION AND CONTROL, 2005, pages 6893-6898, XP002665663, Retrieved from the Internet: URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1583271 [retrieved on 2011-12-13] page 6894, column 2 - page 6895, column 1 -----	1-17, 31-34
A	Apkarian, P.; Noll, D.; Rondepierre, A.: "Nonsmooth optimization algorithm for mixed H2/H-infinite synthesis", 46th IEEE conference on decision and control  December 2007 (2007-12), XP002666027, Retrieved from the Internet: URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4434088 [retrieved on 2011-12-19] the whole document -----	1-17, 31-34

## INTERNATIONAL SEARCH REPORT

 International application No.  
 PCT/US2011/054008

**Box No. II Observations where certain claims were found unsearchable (Continuation of Item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
2. ☐ Claims Nos.:  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
  
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box No. III Observations where unity of invention is lacking (Continuation of Item 3 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
  
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.
  
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
  
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

See annex

**Remark on Protest**

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

International Application No. PCT/ US2011/ 054008

**FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210**

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-17

H-infinity or H2 optimization using a non-smooth algorithm

---

2. claims: 18-21

helper functions

---

3. claims: 22-30

time invariance

---

4. claims: 31-34

constructing an H-infinity or an H2 constraint from a derived control structure

---

5. claim: 35

providing a gradient of a scalar-valued function, and aggregating the gradients into overall gradients of the objective and constraints

---

## フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN

(72)発明者 アプカリアン, ピエール

フランス トゥールーズ F - 3 1 4 0 0 リュ デュ ドクター ダスク 3 アパートメント  
A 2 5

(72)発明者 ノール, ドミニク

フランス ロゼルヴィル F - 3 1 6 5 0 アモー ド デュマンジュール 2 9

Fターム(参考) 5H004 GA30 GB02 GB13 HA07 HA08 HA10 HA16 KB01 KC28