

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2019-197540  
(P2019-197540A)

(43) 公開日 令和1年11月14日(2019.11.14)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 17/50 (2006.01)</b>	G06F 17/50	612A
<b>G06N 99/00 (2019.01)</b>	G06F 17/50	604A
	G06F 17/50	604H
	G06N 99/00	180

審査請求 未請求 請求項の数 15 O L (全 27 頁)

(21) 出願番号 特願2019-78213 (P2019-78213)  
 (22) 出願日 平成31年4月17日 (2019.4.17)  
 (31) 優先権主張番号 15/974, 968  
 (32) 優先日 平成30年5月9日 (2018.5.9)  
 (33) 優先権主張国・地域又は機関  
 米国 (US)

(71) 出願人 504407000  
 パロ アルト リサーチ センター イン  
 コーポレイテッド  
 アメリカ合衆国 カリフォルニア州 94  
 304 パロ アルト カイオーテ ヒル  
 ロード 3333

(74) 代理人 100094569  
 弁理士 田中 伸一郎

(74) 代理人 100109070  
 弁理士 須田 洋之

(74) 代理人 100067013  
 弁理士 大塚 文昭

(74) 代理人 100086771  
 弁理士 西島 孝喜

最終頁に続く

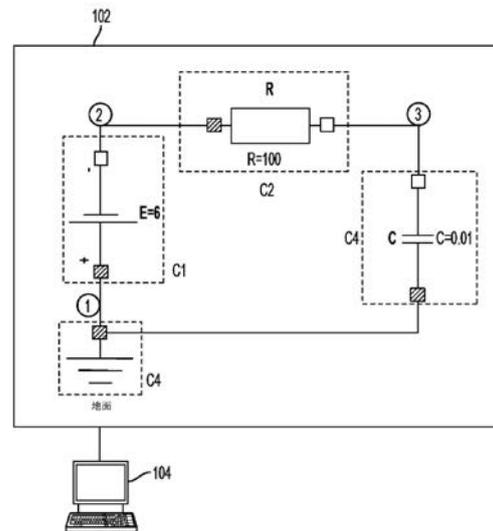
(54) 【発明の名称】 制約発見を用いた物理的コンポーネントの構成方程式の学習

(57) 【要約】 (修正有)

【課題】モデルベースの診断及び予後診断技術を可能にするより多くの関連データ量を生成するシステム及び方法を提供する。

【解決手段】コンポーネントをモデル化するための装置であり、物理的コンポーネント102は、未知のパラメータのベクトルを有し、装置は、少なくとも1つのコンピュータ104と、コンピュータプログラムコードを含む少なくとも1つのメモリと、を備える。少なくとも1つのメモリ、およびコンピュータプログラムコードは、少なくとも1つのコンピュータを用いて、装置に、少なくとも未知のパラメータのベクトルに基づいて点の実現可能性をモデル化するバイナリ関数を構築させ、第1の反復において、バイナリ関数の近似のシーケンスを構築させ、かつ、第2の反復において、未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、バイナリ関数の近似のシーケンスを更新させる。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項 1】

コンポーネントをモデル化するための装置であって、前記コンポーネントが、未知のパラメータのベクトルを有し、前記装置が、

少なくとも1つのプロセッサと、

コンピュータプログラムコードを含む少なくとも1つのメモリと、を備え、

前記少なくとも1つのメモリ、および前記コンピュータプログラムコードが、前記少なくとも1つのプロセッサを用いて、前記装置に、少なくとも、

前記未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築させ、

第1の反復において、前記バイナリ関数の近似のシーケンスを構築させ、かつ

第2の反復において、前記未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、前記バイナリ関数の前記近似のシーケンスを更新させるように構成されている、装置。

10

## 【請求項 2】

前記バイナリ関数が、

## 【数 1】

$$p(w) \in \{0, 1\}$$

であり、式中、

$w$  が、前記未知のパラメータのベクトルであり、

$w$  が実現可能な場合は、 $p(w) = 1$  であり、そうでない場合は、ゼロである、請求項1に記載の装置。

20

## 【請求項 3】

前記近似のシーケンスが、 $q(w)$  によって表され、

$w$  が、未知のパラメータのベクトルであって、

前記第1の反復において、全ての  $w$  について、 $q(w)$  が、0.5 に設定されている、請求項1に記載の装置。

## 【請求項 4】

前記コスト関数が、 $J(w)$  であり、式中、 $w$  が、前記未知のパラメータのベクトルであり、

30

前記バイナリ関数の前記近似のシーケンスの前記更新が、 $J(w)$  の最適化問題を使用することによって行われ、

前記最適化問題が、 $g_i(w) \geq 0$  によって制約され、式中の  $i = 1, 2, \dots, g_i$  が制約関数であり、式中の  $g_i(w)$  が近似関数  $q(w)$  の関数である請求項1に記載の装置。

## 【請求項 5】

前記第1の反復において、前記バイナリ関数の前記近似のシーケンスが、クロスエントロピー損失関数を最小化することによって構築される、請求項1に記載の装置。

## 【請求項 6】

前記少なくとも1つのメモリ、および前記コンピュータプログラムコードが、前記少なくとも1つのプロセッサを用いて、さらに、前記装置に、

40

前記第2の反復に続いて、前記未知のパラメータのベクトルの前記コスト関数の前記最小値に少なくとも部分的に基づいて、前記バイナリ関数の前記近似のシーケンスを更新することによって反復させ、かつ

続く反復において、前記バイナリ関数の更新された近似のシーケンスに基づいて、音声または視覚アラームを生成させるように、構成されている、請求項1に記載の装置。

## 【請求項 7】

前記コンポーネントが、レールスイッチ、回路、電気モータ、風力タービン、エレベータ、または機械的ブレーキのうちの1つである、請求項1に記載の装置。

## 【請求項 8】

50

前記少なくとも1つのメモリ、および前記コンピュータプログラムコードが、前記少なくとも1つのプロセッサを用いて、さらに、前記装置に、

前記更新された近似のシーケンスを使用して、前記コンポーネントのモデルベースの診断を可能にさせるように構成されている、請求項1に記載の装置。

【請求項9】

コンポーネントをモデル化するための装置であって、前記コンポーネントが、未知のパラメータのベクトルを有し、前記装置が、

少なくとも1つのプロセッサと、

コンピュータプログラムコードを含む少なくとも1つのメモリと、を備え、

前記少なくとも1つのメモリ、および前記コンピュータプログラムコードが、前記少なくとも1つのプロセッサを用いて、前記装置に、少なくとも、

前記未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築させ、

前記バイナリ関数の近似のシーケンスを構築させ、かつ

少なくとも2回の反復で反復プロセスをたどらせるように構成されており、各反復が、

前記未知のパラメータのベクトルのコスト関数の最小値を考慮することによって、前記バイナリ関数の前記近似のシーケンスを更新させることを含む、装置。

【請求項10】

前記コスト関数が、より低い不確実性を有する点の探索よりも、より高い不確実性を有する点の探索を促進するという制約を受ける、請求項9に記載の装置。

【請求項11】

前記制約が、前記プロセスが反復されるにつれて、重要性を失う、請求項10に記載の装置。

【請求項12】

前記制約が、反復の数が無限大に近づくにつれて、前記制約が消えるという要件を満たす、請求項11に記載の装置。

【請求項13】

前記コスト関数が、前記プロセスが反復されるにつれて重要性を失うという制約を受ける、請求項9に記載の装置。

【請求項14】

前記コスト関数が、別の点の不確実性よりも高い不確実性を有する点の探査を促進するという制約を受ける、請求項9に記載の装置。

【請求項15】

コンポーネントをモデル化するための方法であって、前記コンポーネントが、未知のパラメータのベクトルを有し、前記方法が、

前記未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築することと、

第1の反復の間に、前記バイナリ関数の近似のシーケンスを構築することと、

第2の反復の間に、前記未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、前記バイナリ関数の前記近似のシーケンスを更新することと、を含む、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、概して、システムモデル化に関する。

【0002】

システムモデルは、独自の理由で不完全なことが多い。しかしながら、完全なシステムモデルを有することは、完全なシステムモデルが、制御、診断、予後診断、および（再）設計などのシステム分析を可能にするので有利である。

【0003】

10

20

30

40

50

システムモデル化に対する現在の手法は、システム/コンポーネントモデルが因果的ブラックボックスモデルである純粋な機械学習手法を含む。それらは、物理的な意味を有していない。

【0004】

本明細書に記載のいくつかの実施形態は、検索空間を制約することによって、モデルパラメータ調整をより容易にし、したがって、学習モデルの物理的解釈を可能にする、パラメータ実現可能性空間の表現を学習する。それらはまた、システムモデルを提供することによって、モデルベースのシステム分析（制御、診断、予後診断）を可能にする。

【0005】

本出願の一態様によれば、コンポーネントをモデル化するための装置があり、コンポーネントは、未知のパラメータのベクトルを有し、装置は、少なくとも1つのプロセッサと、コンピュータプログラムコードを含む少なくとも1つのメモリと、を備える。少なくとも1つのメモリ、およびコンピュータプログラムコードは、少なくとも1つのプロセッサを用いて、装置に、少なくとも、未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築させ、第1の反復において、バイナリ関数の近似のシーケンスを構築させ、かつ第2の反復において、未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、バイナリ関数の近似のシーケンスを更新させるように構成されてもよい。

10

【0006】

前段落に記載の装置においては、バイナリ関数は、

20

【0007】

【数1】

$$p(w) \in \{0, 1\}$$

【0008】

であってもよく、式中、 $w$ は、未知のパラメータのベクトルであり、 $w$ が実現可能な場合は $p(w) = 1$ であり、そうでない場合はゼロである。近似のシーケンスは、 $q(w)$ によって表すことができ、式中、 $w$ は、未知のパラメータのベクトルであり、第1の反復において、 $q(w)$ は、全ての $w$ について、0.5に設定されてもよい。コスト関数は、 $J(w)$ であってもよく、式中、 $w$ は未知のパラメータのベクトルであり、バイナリ関数の近似のシーケンスの更新は、 $J(w)$ の最適化問題を使用して行われ、最適化問題は、 $g_i(w) = 0$ によって制約され、式中、 $i = 1, 2$ であり、 $g_i$ は制約関数である。第1の反復において、バイナリ関数の近似のシーケンスは、クロスエントロピー損失関数を最小化することによって構築することができる。少なくとも1つのメモリおよびコンピュータプログラムコードは、少なくとも1つのプロセッサを用いて、装置に、第2の反復に続いて、未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、バイナリ関数の近似のシーケンスを更新することによって反復させ、続く反復において、バイナリ関数の更新された近似のシーケンスに基づいて、音声または視覚アラームを生成させるように、さらに構成されてもよい。いくつかの実施形態において、コンポーネントは、ルールスイッチ、回路、電気モータ、風力タービン、エレベータ、または機械的ブレーキのうちの1つである。

30

40

【0009】

別の態様においては、コンポーネントをモデル化するための装置が提供される。コンポーネントは、未知のパラメータのベクトルを有することができ、装置は、少なくとも1つのプロセッサと、コンピュータプログラムコードを含む少なくとも1つのメモリと、を備える。少なくとも1つのメモリ、およびコンピュータプログラムコードは、少なくとも1つのプロセッサを用いて、装置に、少なくとも、未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築させ、バイナリ関数の近似のシーケンスを構築させ、かつ少なくとも2回の反復で反復プロセスをたどらせるように構成されてもよく、各反復は、未知のパラメータのベクトルのコスト関数の最小値を考慮すること

50

によって、バイナリ関数の近似のシーケンスを更新させることを含む。

【 0 0 1 0 】

いくつかの実施形態において、コスト関数は、より低い不確実性を有する点の探索よりも、より高い不確実性を有する点の探索を促進するという制約を受ける。いくつかの実装形態において、制約は、プロセスが繰り返されるにつれて、重要性を失う。いくつかの実装形態において、制約は、反復の数が無限大に近づくにつれて、制約が消えるという要件を満たす。いくつかの実装形態において、コスト関数は、別の点の不確実性よりも高い不確実性を有する点の探索を促進するという制約を受ける。

【 0 0 1 1 】

別の態様では、コンポーネントをモデル化するための方法が提供され、コンポーネントは、未知のパラメータのベクトルを有し、本方法は、未知のパラメータのベクトルに基づいて、点の実現可能性をモデル化するバイナリ関数を構築することと、第1の反復の間に、バイナリ関数の近似のシーケンスを構築することと、第2の反復の間に、未知のパラメータのベクトルのコスト関数の最小値に少なくとも部分的に基づいて、バイナリ関数の近似のシーケンスを更新することと、を含む。

10

【 0 0 1 2 】

一態様は、実現可能性セットの表現を学習することを可能にする、探索活用戦略に存在する。

【 0 0 1 3 】

別の態様は、非因果的コンポーネントのパラメータを学習するための方法に存在する。

20

【 0 0 1 4 】

別の態様は、非因果的コンポーネントのパラメータの実現可能性セットの表現を学習するための方法に存在する。

【 0 0 1 5 】

別の態様は、非因果的コンポーネントのパラメータと、非因果的コンポーネントのパラメータの実現可能性セットの表現とを学習するための方法に存在する。

【 0 0 1 6 】

別の態様は、非因果的コンポーネントのパラメータと、非因果コンポーネントのパラメータの実現可能性セットの表現とを共同で学習するための方法に存在する。

【 0 0 1 7 】

別の態様は、非因果的コンポーネントの物理的に解釈可能なモデルを共同で学習するための方法に存在する。

30

【 0 0 1 8 】

別の態様は、部分的に既知のシステムのためのモデルベースの診断および予後診断技術を可能にするための方法に存在する。

【 0 0 1 9 】

概して、本明細書に記載のいくつかの実施形態は、ルールスイッチ、回路、電気モータ、風力タービン、エレベータ、機械的ブレーキ、他の機械的コンポーネントなどの診断および/またはモデル化に関する。

【 0 0 2 0 】

従来のシステムは、十分なデータを生成することができなかった。有利に、本明細書に記載のシステムおよび方法は、より多くの関連データの量を生成する。例えば、開示された制約を使用すると、関連データの量が増加する。

40

【 図面の簡単な説明 】

【 0 0 2 1 】

【 図 1 】 4つのコンポーネントを有する非因果電気システムの例を示す。

【 図 2 】 メモリレスコンポーネントのテンプレートを示す。

【 図 3 】 非線形抵抗器を用いたCauerフィルタを示す。

【 図 4 】 図3からの抵抗器R4電圧の実験データを示す。

【 図 5 A 】 アルゴリズム1の線形近似による制約付き最適化(COBYLA)結果を示す

50

。

【図 5 B】アルゴリズム 1 の逐次最小二乗法プログラミング ( S L S Q P ) 結果を示す。

【図 6 A】無勾配法について分離している超平面を示す。

【図 6 B】無勾配法について分離している超平面を示す。

【図 6 C】無勾配法について分離している超平面を示す。

【図 7 A】勾配近似を使用したアルゴリズムについて分離している超平面を示す。

【図 7 B】勾配近似を使用したアルゴリズムについて分離している超平面を示す。

【図 7 C】勾配近似を使用したアルゴリズムについて分離している超平面を示す。

【図 7 D】勾配近似を使用したアルゴリズムについて分離している超平面を示す。

【発明を実施するための形態】

【 0 0 2 2 】

本明細書に記載のシステムおよび方法は、部分的に既知の物理的システムにおいて、非因果的物理的コンポーネントの構成方程式を学習する問題に対処する。これは、システムの改善されたモデルを構築することを可能にする。

【 0 0 2 3 】

構成方程式のパラメータは、未知の制約のセットを満たす。パラメータおよび制約を共同で学習するために、本明細書に記載のシステムおよび方法は、反復手順を提案する。この手順は、最初の反復中の探索に好都合である。これは、制約条件のモデルを学習することを可能にする。制約学習が進むにつれて、より多くの重みが、構成方程式を見出すことに与えられる。本発明の結果は、S y s X プラットフォームの能力を増強するために、

【 0 0 2 4 】

さらに、一実装形態は、非線形抵抗器のモデルが学習される実例でテストされる。

【 0 0 2 5 】

一態様において、以下は、システム / コンポーネントの非因果的表現の学習について説明する。いくつかの実施形態は、未知のシステム / コンポーネントの構成方程式を学習する。それらは、パラメータ化された機械学習タイプのモデル、例えば回帰モデル、ニューラルネットワークを使用して表すことができる。モデルパラメータは、未知の実現可能性セットに残されるように制約されている。いくつかの実施形態は、パラメータと実現可能性セットの表現との両方を学習する反復手順を含む。いくつかの実施形態において、実現可能性セット表現モデルは、確率論的であり、バイナリ分類のための機械学習アルゴリズム (例えば、ロジスティック回帰、最後の層にソフトマックス関数を有するニューラルネットワーク) を使用して更新される。実現可能セット表現を学習するための訓練データは、パラメータ推定最適化プロセスの間に取得され得る。始めに、不確実性の高い検索空間の探索が促進され、学習プロセスが進むにつれて、焦点はパラメータ推定に向かってシフトされる。

【 0 0 2 6 】

散逸性制約のような他の制約を実現可能セット表現に含めることができる。コンポーネントモデルは、他の導関数を含まないアルゴリズム、およびベイジアン最適化などのグローバル最適化への扉を開くガウスプロセスとして表すことができる。

【 0 0 2 7 】

序論

現在の課題の 1 つは、部分的に既知のシステムにおいて物理的コンポーネントのモデルを学習することである。そのようなシステムは本質的に非因果的である。連続状態のみを有するシステムの場合、システムの挙動は、次の形式の微分代数方程式 ( D A E ) として表すことができる。

【 0 0 2 8 】

10

20

30

40

【数 2】

$$\mathbf{0} = F(\dot{x}, x, z; w) \quad (1)$$

$$y = h(x, z; w) \quad (2)$$

【0029】

式中、F、Hは、ベクトル値の連続的なマップであり、xは、状態ベクトルであり、zは、代数変数のベクトルであり、yは、出力のベクトルであり、wは、パラメータのベクトルである。非因果的システムは、そのコンポーネントに付随する変数と、それらの間の関係と、で構成されている。その関係は、パラメータ化された構成方程式と、コンポーネント間の接続と、よって引き起こされる。因果的（またはシグナルフロー）モデルとは異なり、非因果的モデルは、物理的な意味を保つ。図1は、4つのコンポーネントを有する非因果的電気システムの一例を示し、構成方程式および接続方程式は、次のとおりである。

10

【0030】

【数 3】

構成方程式:

$$C1: E = v_E^{(1)} - v_E^{(2)}$$

$$C2: Ri_R = v_R^{(2)} - v_R^{(3)}$$

$$C3: C \frac{d}{dt} (v_C^{(3)} - v_C^{(1)}) = i_c$$

$$C4: v_G^{(1)} = 0$$

20

【0031】

【数 4】

接続方程式:

$$v_E^{(2)} = v_R^{(2)}$$

$$v_R^{(3)} = v_C^{(3)}$$

$$v_E^{(1)} = v_C^{(1)} = v_G^{(1)}$$

$$i_E = i_R = i_C$$

30

【0032】

加えて、構成方程式のパラメータは通常、いくつかの実現可能性セットの範囲内で制約されることに留意されたい。例えば、電気抵抗器の抵抗は、常に正である必要がある。

40

【0033】

いくつかの実施形態は、以下のステップ：(i) 構成方程式のためのパラメトリック数学モデルを選択するステップ、ならびに(ii) パラメータおよびパラメータ制約の表現/モデルを学習するステップに基づいている。後者の場合、探索活用戦略を使用することが可能であり、始めに、制約表現を学習するために探索に焦点を当てる。後になって、制約表現がより有益になるにつれて、構成方程式の最良の実現可能なパラメータを見出すことに焦点が移される。パラメータ制約の表現を有することは、検索の複雑さを減少させ、コンポーネントのパラメトリックモデルと制約表現を洗練するために新しいデータが利用可能であるときに良い初期条件を提供し、かつコンポーネントモデルの物理的解釈を可

50

能にすることに役立つ。システム/コンポーネントの挙動を記述する物理法則の学習における最近の結果は、機械学習、およびその派生的で深い学習への再燃した関心によって支えられていた。(Michael B Chang, Tomer Ullman, Antonio Torralba, および Joshua B Tenenbaum, 物理力学を学ぶための構成的オブジェクトベースの手法, arXiv preprint arXiv:1612.00341, 2016)において、著者は、画像から直感的な物理学の予測モデルを学習するためにニューラルネットワークアーキテクチャを使用した。スパース回帰法は、(S. Bruntton, J. N. Kutz, および J. Proctor, 準拋物理法則のデータ駆動型発見, SIAM News, 50(1), 2017年, Samuel H. Rudy, Steven L. Bruntton, Joshua L. Proctor, および J. Nathan Kutz, 偏微分方程式のデータ駆動型発見, Science Advances, 3(4), 2017)において、支配的な物理法則を記述する(偏)微分方程式を発見するために使用された。測定値から自由形式の法則を抜粋するための遺伝的アルゴリズムに基づく、かなりスケラブルでない方法が、(Michael Schmidt および Hod Lipson, 実験データからの自由形式の自然法則の抜粋, Science, 324(5923): 81~85, 2009)において提案された。

10

#### 【0034】

作業上の仮定は次のとおりである。(i) 物理的システムは物理的コンポーネントの相互接続によって記述される。(ii) システムコンポーネントの構成方程式は、コンポーネントのサブセットについてのみ既知である。(iii) システムの挙動についての観察結果が利用可能である。本明細書に記載されるように、コンポーネント変数への直接アクセスは必ずしも存在しない。物理的システムのコンポーネントベースの記述を有することで、制御に対するモデルベースの様々な分析方法(Carlos E. Garcia, David Prett, および Manfred Morari, モデル予測制御: 理論および実践調査, Automatica, 25(3): 335~348, 1989)、診断(H. Alwi, C. Edwards, および C. P. Tan, Fault Detection および Fault - - Tolerant Control Using Sliding Modes, Springer, London, 1974 (ISBN: 0-12-078250-2)、Johan De Kleer, および Brian C Williams, 多重故障の診断, 人工知能, 32(1): 97-130, 1987)および予後(Andrew K. S. Jardine, Daming Lin, および Dragan Banjevic), 状態ベースの保全を実施する機械診断および予後診断に関するレビュー, Mechanical Systems および Signal Processing, 20(7): 1483~1510, 2006)が可能になる。実際の仮定(ii)は想像以上に頻繁である。多くの場合、システム製造業者は自社製品に(モデルベースの)診断および予後診断機能を含めることによって、自社製品を改善したいと考えている。しかしながら、そのようなサブコンポーネントが第三者から取得された場合、システム製造業者は、独自の理由で、サブコンポーネントの詳細な仕様に必ずしもアクセスできるとは限らない。本発明者らは、最近の産業関連プロジェクトにおいてそのようないくつかの事例に直面した。そのような契約の一例(I. Matei, A. Ganguli, T. Honda, および J. de Kleer, 診断へのハイブリッド手法例: 鉄道スイッチ, 第26回診断原理の国際ワークショップ(DX-2015), 225~232頁, 2015年8月の議事録)において、本発明者らは、レールスイッチの構造的記述および電氣的仕様の一部を有していたが、本発明者らは、いくつかの機械的部品の仕様を見出すことに課題を有していた。この文書で説明されている作業は、このような問題に対する解決策を提供する。

20

30

40

#### 【0035】

システムの挙動に関する定性的情報があるという幸運な場合、つまり、構成方程式は知っているがパラメータ値はわからない場合、学習問題は標準的なパラメータ推定問題とな

50

り、長い歴史がある (Y. Bard、非線形パラメータ推定、Academic、New York、2011 (ISBN: 978-0-85729-649-8))。ただし、1つのシナリオでは、そのような情報は利用できない。したがって、システムモデルに含まれている場合に DAE (1) が解決可能になるように、未知のコンポーネントに対する非因果的モデルを提案する必要がある。本明細書に記載の手法は、そのようなモデルを実現可能と呼ぶ。より高い目標は、コンポーネントモデルの再利用性である。これは、未知の物理的コンポーネントが異なる構成で使用されている場合、実際の挙動およびシミュレートされた挙動が類似している必要があることを意味する。ただし、再利用可能なコンポーネントを学習するには、そのコンポーネントの様々な使用例に対応する実験データが必要である。本明細書に記載の、ある一定の実施形態において、焦点は主に実現可能性にある。数値的な観点からは、実現可能性は、ある時間間隔にわたって、かつ、いくつかの初期条件に対して、DAE (1) を解き、 $x(t)$  および  $z(t)$  の軌道を決定することに変換される。このプロセスは、

【0036】

【数5】

$$\text{Jacobian } \frac{\partial F}{\partial \eta}(\eta(t))$$

【0037】

を必要とし、

【0038】

【数6】

$$\eta^T = \begin{bmatrix} \cdot^T \\ \mathbf{x}^T, \mathbf{z}^T \end{bmatrix}$$

【0039】

はシステムの軌跡に沿って可逆である必要がある。この特性が軌道上のある点で失敗すると、数値シミュレーションは失敗する。本明細書に記載の手法は、以下の実現可能なコンポーネントモデルの定義を有する。

【0040】

例えば、システムモデルで

【0041】

【数7】

$$\text{Jacobian } \frac{\partial F}{\partial \eta}(\eta(t))$$

【0042】

がシステムの軌跡に沿って可逆的である場合、コンポーネントモデルは実現可能である。

【0043】

この非限定的な一例において、抵抗器のための線形モデルを有する代わりに、その構成方程式が非線形モデル

【0044】

【数8】

$$\mathbf{f} \left( \mathbf{i}_R, \mathbf{v}_R^{(2)}, \mathbf{v}_R^{(3)} ; \mathbf{w} \right) = 0$$

【0045】

10

20

30

40

50

によって与えられ、式中、wは抵抗器のパラメータを表すと仮定する。自明な方程式を消去した後では、

J a c o b i a n 行列

【 0 0 4 6 】

【 数 9 】

$$0 = f(i_R, E, v_C^{(3)})$$

$$0 = -\dot{v}_C^{(3)} + \frac{1}{C} i_R$$

10

【 0 0 4 7 】

を決定する可解性を用いて、  
電気回路の D A E は次のようになる。

【 0 0 4 8 】

【 数 1 0 】

$$\frac{\partial F}{\partial \eta} = \begin{bmatrix} \frac{\partial f}{\partial i_R} & 0 \\ -\frac{1}{C} & 1 \end{bmatrix} \tag{3}$$

20

【 0 0 4 9 】

したがって f およびパラメータ w の形式がどのように選択されても、最小要件は変換行列 ( 3 ) が可逆的であることであり、これは、この特定の場合に f / ( i\_R ) の可逆性に変換される。固定の f では、w の選択が悪いと、可逆性が損なわれる可能性がある。(この場合の D A E は半明示的である。このようなタイプの場合、F / の可逆性は代数制約の J a c o b i a n の可逆性、すなわち f / ( i\_R ) にまとまる)。

【 0 0 5 0 】

w 事前の実現可能性セットを知ることは、単純な電気回路のためであっても平凡なことではなく、何百または潜在的な何千ものコンポーネントを有する複雑なシステムでは、もちろんのことである。これに加えて、システムの所与の構成に対して実現可能なパラメータが見つかったとしても、それらが異なる構成に対して実現可能であることが保証されないという追加の課題がある。これは、新しい J a c o b i a n が大きく異なる可能性があるという事実の結果である。

30

【 0 0 5 1 】

以下では、コンポーネントの未知の構成方程式は連立方程式 G ( ; w ) = 0 で表され、式中、G は微分可能ベクトル値関数、 はコンポーネントの変数ベクトル、w は未知パラメータのベクトルであると仮定する。w を学習するために、コンポーネントの挙動についての間接的な情報を与える、ある時間間隔にわたる測定値 y\_m のセットへのアクセスがあると仮定する。これらの測定値は、制約付きの非線形最小二乗問題を解くために使用される。

40

【 0 0 5 2 】

【 数 1 1 】

$$\min_w J(w) \tag{4}$$

$$w \in \mathcal{W} \subset \mathbb{R}^n, \tag{5}$$

【 0 0 5 3 】

を条件として、  
式中、

50

【 0 0 5 4 】

【 数 1 2 】

$$J(w) = \frac{1}{N} \sum_{i=0}^N \|y_m(t_i) - y(t_i; w)\|^2, y(t_i; w)$$

【 0 0 5 5 】

はシミュレートされた測定値であり、 $\{t_i\}$  ( $i = 0$ ) は測定が行われときのインスタンスのシーケンスを表し、かつ  $w$  は未知の実現可能性セットである。1つの目的は、最適なもの  $w$ 、およびセットの少なくとも一部  $w$  を共同で学習することができる方法を考え出すことである。

10

【 0 0 5 6 】

物理的コンポーネントのモデル

図 1 は、電気的コンポーネントを使用して作成されたシステムとみなす。特に、図 1 は、コンピュータ 1 0 4 に接続されたモデルまたは物理的コンポーネント 1 0 2 を示す。図 1 に示される物理的コンポーネント 1 0 2 のコンポーネントは、それらを通して他のコンポーネントと相互作用するピンを有する。特に、ピンは、電流および電位という2つの変数によって特徴付けられる。電流は接続点で保存されるフロー変数である。文献では、電流および電位は、それぞれ、変数を通り、横切るように称される。ピンは一般にコネクタと称され、物理的領域に関係なく、それらは、少なくとも1つの流動変数および可能性変数を有する（熱流体領域では、いくつかのそのような変数があり得る）。

20

【 0 0 5 7 】

図 2 は、メモリレスコンポーネントのテンプレートを示している。本明細書に記載のいくつかの実施形態は、図 2 に示すように、2つのピン（コネクタ）コンポーネント 2 0 2、2 0 4 を考慮している。各コネクタ 2 0 2、2 0 4 は、2つの変数、すなわち流動変数  $f$  および可能性変数  $x$  を有する。物理的領域に依りて、流動変数は、電流（電気）、力（並進力学）、トルク（回転力学）または熱流速（熱）であり得る。変数  $x$  は、電位（電気）、変位（並進力学）、角度（回転力学）または温度（熱）を表すことができる。モデルの構成方程式は、変数 ( $f_a, x_a$ ) および ( $f_b, x_b$ ) のペア間の関係を記述する方程式のセットである。メモリレスコンポーネントの場合には、

【 0 0 5 8 】

30

【 数 1 3 】

$$G(f_a, f_b, x_a, x_b, u; w) = 0$$

【 0 0 5 9 】

があり、式中、

【 0 0 6 0 】

【 数 1 4 】

$$G: \mathbb{R}^4 \rightarrow \mathbb{R}^2$$

40

【 0 0 6 1 】

は、コネクタ変数を制約するベクトル値の微分マップであり、 $w$  は、パラメータのベクトルである。例えば、電気抵抗器の場合、 $f_a + f_b = 0$  および  $w f_a = x_a - x_b$  がある。G 4つの変数に対して2つの方程式のみを定義することに留意されたい。他のコンポーネントに接続するとき、他のコンポーネントの構成方程式と、接続方程式とから追加の方程式が生成される。最も単純なシナリオでは、4つの変数のうち2つが他の方程式から独立して計算される。これは、1つのコネクタが地面（ゼロ電位）に接続され、残りのコネクタが電流または電圧源に接続されている場合である。より複雑なシナリオにおいては、システムの挙動は代数方程式のセットで記述され、全てのコンポーネントの変数が同時に計算される。本明細書に記載されるいくつかの手法は、コンポーネントの変数の導関数

50

を含むことによって、同様にメモリを使用してコンポーネントをモデル化し、

【0062】

【数15】

$$G(\xi; w) = 0 \quad , \text{ 式中、}$$

【0063】

【数16】

$$\xi = [f_a, f_a^1, \dots, f_a^m, f_b, f_b^1, \dots, f_b^m, x_a, x_a^1, \dots, x_a^n, x_b, \dots, x_b^n]$$

10

【0064】

$x(n)$  は  $x$  の  $n$  番目の導関数である。我々は、 $2 \times (n + 1) + 2 \times (m + 1)$  変数および2つだけの方程式 ( $G$  は、2次元ベクトル値関数のままである) を有するようになることがある。しかしながら、これらのタイプのコンポーネントについては、最大の導関数だけが未知の変数であり、一方、低次の導関数は前の積分ステップから既知であると仮定される。

【0065】

コンポーネントに関する事前の情報がないければ、 $G$  の構造を選択することは科学というよりは芸術である。最も単純な形式は、線形回帰モデルで  $G(\quad; w) = W$  内線あり、式中、 $W$  はパラメータの行列であり、内線は要素の ' エントリのべき乗である「特徴」のベクトルである。例えば、メモリがない場合は、

20

【0066】

【数17】

$$\xi_{\text{内線}} = \left[ f_a, f_a^2, f_a^3, f_b, f_b^2, f_b^3, x_a, x_a^2, x_a^3, x_b, x_b^2, x_b^3 \right]$$

【0067】

を有することができる。より複雑な挙動は、(Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, および J. Nathan Kutz, 偏微分方程式のデータ駆動型発見, Science Advances, 3(4), 2017) に提案されているように、エントリのべき乗の乗算も考慮することによって、達成することができる。ニューラルネットワーク (NN) は普遍的な近似子である (G. Cybenko, シグモイド関数の重ね合わせによる近似, 制御数学, 信号, およびシステム, 2(4): 303~314, 1989年12月) ことを想起すると、入力はベクトルであり、出力は、ゼロ定数であるターゲットに従わなければならない、NN によって記述される  $G$  を選択することができる。ただし、NN は双方向グラフとみなすことに留意されたい。つまり、それは方程式のセットを表し、割り当てのセットを表すものではない。

30

【0068】

したがって、本明細書に記載のいくつかの技術は、モデルに基づく診断に使用することができる。例えば、モデルは、何かがコンポーネントに問題があると判定するために使用されてもよく、その判定に応答して、音声または視覚警報が送信されてもよい。いくつかの実施形態において、これは、ルールスイッチ、回路、電気モータ、風力タービン、エレベータ、機械的ブレーキ、他の機械的コンポーネントなどに適用され得る。

40

【0069】

特殊なケース

上記のように、結果として得られるモデルが実現可能である限り、マップ  $G$  は潜在的に任意の数学的表現を取ることができる。物理学ベースのモデル化において使用される物理的コンポーネントの構成方程式テンプレートに触発されて、本明細書に記載される手法は

50

、以下の特殊な場合を考慮する。

【 0 0 7 0 】

タイプ 1 : 最も一般的な挙動テンプレートの 1 つは、コンポーネントを通るフローが失われない場合に対応する。構成方程式は次の式で与えられる。

【 0 0 7 1 】

【 数 1 8 】

$$G_1(f_a, f_b) = (f_a + f_b) = 0 \quad (6)$$

$$G_2(f_a, \dots, f_a^{(m)}, x_a - x_b, \dots, x_a^{(n)} - x_b^{(n)}; w) = 0 \quad (7)$$

10

【 0 0 7 2 】

このテンプレートは、抵抗器、コンデンサ、インダクタ、バネ、ダンパーなど、複数の領域からの線形または非線形のコンポーネントをカバーしている（表 1 を参照）。

【 0 0 7 3 】

タイプ 2 : 流動損失を被る物理的要素がある。したがって、2 つのコネクタでの流れは異なる。この場合のテンプレートは次のように表すことができる。

【 0 0 7 4 】

【 数 1 9 】

$$G_1(x_a, x_b) = (x_a - x_b) = 0 \quad (8)$$

$$G_2(f_a, \dots, f_a^{(m)}, f_b, \dots, f_b^{(m)}, x_a, \dots, x_a^{(n)}; w) = 0 \quad (9)$$

20

【 0 0 7 5 】

【 表 1 】

抵抗器／バネ	$f_a = w(x_a - x_b)$
コンデンサ／ダンパー	$f_a = w(x_a^{(1)} - x_b^{(1)})$
インダクタ	$f_a^{(1)} = w(x_a - x_b)$
バネダンパー	$f_a = w_1(x_a - x_b) + w_2(x_a^{(1)} - x_b^{(1)})$

30

表 1 : タイプ 1 線形コンポーネントの例

【 0 0 7 6 】

このテンプレートに対応する典型的な例は、方程式が  $x_a = x_b$  および

【 0 0 7 7 】

【 数 2 0 】

$$f_a + f_b = g(x_a, \dots, x_a^{(n)}; w) = 0$$

【 0 0 7 8 】

40

によって与えられる機械的ブレーキであり、式中、 $g$  は、フロー損失を関数  $x_a$  およびその導関数、ならびに可能な外部信号として決定するマップである。例えば、粘性損失の場合は

【 0 0 7 9 】

【 数 2 1 】

$$g(x_a^{(1)}, u; w) = u \cdot w \cdot x_a^{(1)}$$

【 0 0 8 0 】

である。変数  $u$  は、ブレーキを作動 / 停止する外因性の信号である。

50

【0081】

タイプ3：対応するテンプレートを使用して、流動方程式および可能性方程式が完全に分離している場合

【0082】

【数22】

$$G_1(f_a, \dots, f_a^{(m)}, f_b, \dots, f_b^{(m)}; w) = 0 \quad (10)$$

$$G_2(x_a, \dots, x_a^{(n)}, x_b, \dots, x_b^{(n)}; w) = 0 \quad (11)$$

10

【0083】

これは、理想的な変圧器の場合、または特に

【0084】

【数23】

$$G_1(f_a, f_b) = f_a + w f_b \quad \text{および}$$

【0085】

【数24】

$$G_2(x_a, x_b; w) = w x_a - x_b$$

20

【0086】

の理想的な歯車の場合である。

【0087】

共同学習および制約の発見

いくつかの実装は、Gのためのパラメトリックモデルを選択し、実験データと最もよく一致するようにそのパラメータを学習することが有利である。始めは、実現可能性セットに関する情報はない。したがって、徐々に、データを取得し、Gのパラメータを最適化して、実現可能性セットのモデルを構築する必要がある。本明細書に記載の手法は、コスト関数  $J(w)$  の減少をパラメータ値の近くに抑制するように試みながら、始めに高い不確実性を有する点の探査を促進する逐次方式を提案する。以下では、 $J(w)$  は、少なくとも1つの(局所)ミニマイザを有すると仮定する。この方式は、次の結果によって有効になる。

30

【0088】

命題3.1:  $\{g_k\}_{(k=0)}$  とすると、

【0089】

【数25】

$$\lim_{k \rightarrow \infty} g_{k,i} = g_i \quad \text{で、}$$

【0090】

【数26】

$$g_{k,i}: \mathbb{R}^n \rightarrow [a_i, b_i], \quad i = 1, 2$$

40

【0091】

は連続微分スカラー関数の2つの収束シーケンスとなる。それらは最適化問題のシーケンスを引き起こす。

【0092】

【数 2 7】

$$(P_k): \min_w J(w) \quad (12)$$

$$g_{k,i}(w) \leq 0, \quad i = 1, 2, \quad (13)$$

【0093】

を条件とする。

式中、各問題  $(P_k)$  は、少なくとも局所ミニマイザ

【0094】

【数 2 8】

10

$$w_k^*$$

【0095】

を解として認める。次に、シーケンス

【0096】

【数 2 9】

$$\{w_k^*\}_{k \geq 0}$$

20

【0097】

の限界点  $w^*$  は、問題の解である。

$$(P) : \min_w J(w)$$

$$g_i(w) \leq 0, \quad i = 1, 2$$

を条件とする。

【0098】

証明は連続性の議論から得られる。セットアップにおいて、 $J$  は (4) に導入されたコスト関数であり、 $g_i(w)$  は実現可能性セットに関連する制約関数である。それらは、 $w$  の実現可能性セットについての理解が深まるにつれて、各反復  $k$  とともに進化する。

30

【0099】

いくつかの実施形態は、バイナリ関数によって点の実現可能性をモデル化し

【0100】

【数 3 0】

$$p(w) \in \{0, 1\},$$

【0101】

式中、 $w$  が実現可能な場合は  $p(w) = 1$ 、そうでなければゼロである。全ての可能な点の実現可能性を最初に探ることができる場合、実現可能性空間を記述するマップ  $p(\cdot)$  を構築することができる。明らかに、これは絶望的なシナリオである。望むことができる最良のことは、実現可能性がテストされた点のかなり大きなデータセットを有することである。

40

【0102】

【数 3 1】

$$S \subset \mathbb{R}^n$$

【0103】

50

が（任意の大きさ）のコンパクトなセットであるとする。滑らかで、パラメータ化された関数  $q : S \rightarrow [0, 1]$  としてモデル化された、 $q(w) = q(w; \beta)$  である、 $p(w)$  の近似を見出だしたい。この近似は、クロスエントロピー損失関数

【0104】

【数32】

$$H(\beta; S) = - \int_S [p(w) \log_2 q(w; \beta) + (1 - p(w)) \log_2 (1 - q(w; \beta))] dw$$

【0105】

を最小化することによって得られる。 $q^*(w) = q(w; \beta^*)$  によって  $p$  の最良の近似を表し、式中、 $\beta^* = \arg \min_{\beta} H(\beta; S)$  であり、 $H^*(S) = H(\beta^*; S)$  である。関数  $q^*$  は、セット  $S$  内の任意の点  $w$  の実現可能性を探ることができる理想的なシナリオにおける目標関数を表すことができる。次の結果は、セット  $S$  を探り続けながら、 $p$  の近似を逐次学習することを通じて  $q^*$  に到達することができる方法を示している。

10

【0106】

命題 3.2 :  $\{S_k\}_{k=0}^{\infty}$  を  $S_0 = S$  のように、離散点のセットの増加するシーケンスとする。 $\{q_k(w)\}_{k=0}^{\infty}$  を  $p(w)$  の近似のシーケンスであるとし、式中、

【0107】

【数33】

$$q_k(w) = q(w; \beta_k), \quad \beta_k = \arg \min_{\beta} \tilde{H}(\beta; S_k) \quad \text{であり、}$$

【0108】

【数34】

$$\tilde{H}(\beta; S_k) = \frac{1}{|S_k|} \sum_{i=1}^{|S_k|} [p(w_i) \log_2 q(w_i; \beta) + (1 - p(w_i)) \log_2 (1 - q(w_i; \beta))]$$

【0109】

を伴う。

30

【0110】

次に、

【0111】

【数35】

$$\lim_{k \rightarrow \infty} \tilde{H}(\beta_k; S_k) = H^*(S)$$

【0112】

であり、

$\lim_k q_k = q^*$  である。

40

【0113】

セット  $S_k$  は、 $S$  からのサンプルを含む。これらのサンプルは、量

【0114】

【数36】

$$\tilde{H}(\beta; S_k)$$

【0115】

を通じてエントロピー  $H(\beta; S)$  の近似を可能にし、近似は、 $S_k$  が  $S$  に近づくにつれて、より正確になる。 $S$  を任意に大きくすることによって、

50

【 0 1 1 6 】

【 数 3 7 】

 $\mathbb{R}^n$ 

【 0 1 1 7 】

全体をカバーすることができる。q の可能な選択は、ソフトマックス関数

【 0 1 1 8 】

【 数 3 8 】

$$q(w; \beta) = e^{\beta^T w} / [1 + e^{\beta^T w}]$$

10

【 0 1 1 9 】

である。最後の層がソフトマックス関数である NN のように、より複雑な表現をすることができる。パラメータ  $w_k$  を学習することは、クロスエントロピー損失関数を使用してバイナリ分類器を学習することと同じである。

【 0 1 2 0 】

これで関数  $J_k$  および  $g_{(k, i)}$  を定義する準備ができた。特に、

【 0 1 2 1 】

【 数 3 9 】

$$g_{k,1}(w) = 1 - \delta_{k,1} - q_k(w) \quad (14)$$

20

$$g_{k,2}(w) = 1 - \delta_{k,2} + q_k(w) \log_2 q_k(w) + [1 - q_k(w)] \log_2 [1 - q_k(w)] \quad (15)$$

【 0 1 2 2 】

式中、 $\{ (k, 1) \}_{k=0}^{\infty}$  は正のシーケンスであって、 $\lim_{k \rightarrow \infty} (k, 1) = 0$ 、 $0 < (k, 1) < 1$

【 0 1 2 3 】

および  $(0, 1) = 1$  を伴い、 $\{ (k, 2) \}_{k=0}^{\infty}$  は、 $\lim_{k \rightarrow \infty} (k, 2) = 1 + \epsilon$  のように正の値の増加するシーケンスであって、 $(1, 2) = \epsilon$  を伴う。関数  $q_k(w)$  は、命題 3.2 で導入された。第 2 の制約 (15) はエントロピーのような用語を使用し、その目的は、より高い不確実性を有する点の探索を促進することである。この用語は、 $q_k$  のモデルが改善されるにつれて重要性を失う。極限では、制約 (14) は  $q^*(w) = 1 - \delta_{k,1}$  となり、(15) は  $q^*(w) \log_2 q^*(w) + [1 - q^*(w)] \log_2 [1 - q^*(w)]$  となり、これは任意の  $w$  についても満たされている。したがって、この制約は極限で消える。これにより、最適化問題の解決策が確実に実行可能になる。第 1 の反復において、利用可能なデータがないとき、全ての  $w$  に対して  $q_0(w) = 0.5$  を確実にする。これは、全ての点が最大の不確実性を有することを意味する。この値でのエントロピーは 1 なので、 $(P_0)$  が解を有することも確実にする。ソフトマックス関数の場合、これは、 $(0, 2) = 0$  を強制する場合に起こる。したがって、最初の反復においては、制約のない最適化問題を解く。第 2 の反復においては、 $q_1(w)$  は、点 0.5 を通過する滑らかな関数であるため、制約 (15) が満たされるように  $w$  が存在する。スカラーは、 $q^*(w)$  が最後に  $p(w)$  の近似であるという事実を説明している。

30

40

【 0 1 2 4 】

ここまでは、中間の実現可能性モデル  $q_k(w)$  を訓練するためのデータを蓄積する手順を扱っていない。各反復  $k$  においては、最適化アルゴリズムは、ミニマイザを検索する際にデータ点を生成する。検索には関数評価が必要で、それにはモデルのシミュレーションも必要である。したがって、反復  $k$  においてより良い点を検索しながらテストされた各点は、シミュレーションの成功に応じて、実現可能または実現不可能として分類される。反復プロセスの始めにおいて、第 2 の制約 (15) は、不確実性の高い点に向かって検索をプッシュし、これにより以前に訪れたことがない点を探索するのに役立つ。しかしなが

50

ら、この戦略は、 $k$ の反復を停止するときデータセット $S_k$ が十分大きく、有益であることを保証できないので、結果としての実現可能性モデル $q_k(w)$ は $p(w)$ で定義されたランドスケープ全体と厳密に一致する。それでも、最適化プロセスの一部として生成された軌跡に沿って設定された、実現可能性の良好な局所表現を提供することができる。以下のアルゴリズム1でこれまでに記載した手法を要約する。このアルゴリズムの6行目は、制約付き最適化問題(12)~(13)を解き、解の検索中に解、ならびに全ての訪問点およびそれらのラベルを返す。それは $q$ 関数の以前のモデルを使用するので、任意の訪問点の実現可能性が $q$ によって予測されるため、予測ステップをも含むものとして解釈することができる。8行目において、実現可能性モデルは、新しく取得されたデータに基づいて再訓練される。データセット $D$ が非常に大きくなる場合、オンライン学習を使用することができる。例えば、更新方向を命令する勾配が、新しく取得されたデータのみを使用して計算されるバッチ学習更新ステップを実行することができる。より極端な変更は、実現可能性モデルの更新をコスト関数評価内に転送することである。ここでは、各シミュレーションの後に確率的勾配降下アルゴリズムの1ステップを使用することができる。

10

【0125】

[表]

---

アルゴリズム1 制約の発見の学習

---

**1: procedure OPTWITHCONSTRDISCOVERY( $w_0$ )**

20

**2:      $w \leftarrow w_0$**

**3:      $\mathcal{D} \leftarrow \{\emptyset\}$**

**4:      $q(w) \leftarrow 0.5, \forall w$**

**5:     while            解の改善中、            do**

**6:         ( $w, data$ )  $\leftarrow w, q$ を与えられた(12)~(13)の解を計算する**

**7:          $\mathcal{D} \leftarrow \mathcal{D} \cup data$**

**8:          $q \leftarrow D$ に基づいて実現可能性モデルを再訓練する**

30

**9:     return  $w, q$**

---

【0126】

実現可能性モデルの事前訓練

最適化プロセスを進めながら、実現可能性セットのモデルを構築する方法を示した。この手法の不利点は、始めは制約確率モデルに不確実性が高いことである。点 $w$ の実現可能性は、その時点でのシミュレーションの成功に基づいており、最適化プロセスとは関係がない。実際の最適化プロセスを開始する前に、コンポーネントのパラメータ空間からサンプリングすることができ、各サンプルについてシミュレーションを実行し、確率モデル $q(w; \cdot)$ を事前に訓練するためのデータを使用することができる。実現可能ではあるが、特にシステムモデルが複雑な場合、この手法は時間と計算量の両方がかかる可能性がある。より安価な代替方法は、様々な構成で学習したいコンポーネントを含む、はるかに単純なモデルを生成することである。このような構成は、 $G(\cdot; w) = 0$ で記載されたコンポーネントの構成方程式の未知の変数を計算するための様々なシナリオを記載し、このようにして実現可能空間の少なくとも一部に光を当てる。つまり、シミュレーションを成功させるためには、モデルのJacobian行列に対して異なる構造を生成することになる。

40

【0127】

実用的な実装態様

以下のアルゴリズム1でこれまでに記載した手法を要約する。このアルゴリズムの6行

50

目は、制約付き最適化問題 (12) ~ (13) を解き、解の検索中に解、ならびに全ての訪問点およびそれらのラベルを返す。それは  $q$  関数の以前のモデルを使用するので、任意の訪問点の実現可能性が  $q$  によって予測されるため、予測ステップをも含むものとして解釈することができる。8行目において、実現可能性モデルは、新しく取得されたデータに基づいて再訓練される。

【0128】

[表]

---

### アルゴリズム2 経験的バリア機能を持つコスト関数疑似コード

---

```

1: function costFunction(w)
2:   パラメータ wについてモデルをシミュレートする
3:   if シミュレーションが失敗の場合、 then
4:     counter ← counter + 1
5:     cost ←  $-\log \frac{\varepsilon}{\text{counter}}$ ,  $\varepsilon \ll 1$ 
6:   else
7:     counter ← 0
8:     cost ← J(w)
9:   return cost

```

10

20

30

40

50

【0129】

このアイデアは、アルゴリズム2に詳述されている。バリア関数として対数関数を選んだ。アルゴリズムがモデルを実現不可能な点でシミュレートし続けるにつれて、その値は増加する。これにより、コスト関数の勾配の平準化が妨げられる。中間最適化問題 (12) ~ (13) は、不等式制約を伴う最適化問題である。そのような問題を解決するために利用可能なくつかの方法、すなわち、ペナルティ法、バリア法、プライマル法、または逐次二次計画法 (SQP) (D. P. Bertsekas、非線形計画法、Athena Scientific、1999) がある。多くの技術的条件の中で、それらの多くはコスト関数の2回の連続的な微分可能性を必要とする。複雑なシステムでは、勾配を計算することでさえも困難な場合がある。コスト関数  $J$  の勾配は次式で与えられ、

【0130】

【数40】

$$\nabla J(w) = -\frac{1}{N} \sum_{i=0}^N (y_m(t_i) - y(t_i)) \frac{\partial y(t_i)}{\partial w},$$

【0131】

式(2)のように、式中、 $y/w = H/x \quad x/w + H/w$ 。Hが状態内で線形であり、 $w$ に依存しない単純なシナリオにおいても、 $x/w$ が必要である。前進法および後退法を使用する感度分析機能は、DAEソルバー (Linda Petzold、Shengtai Li、Yang Cao、および Radu Serban、微分代数方程式および偏微分方程式の感度解析、コンピュータおよび化学工学、30(10): 1553 ~ 1559、2006) に既に統合されており、理論的には、それらの結果から  $x/w$  を抽出することができた。それでも、Hessianを取得することは、些細なことではない。幸いなことに、それは、勾配を使用して反復的に近似することができる (Jorge Nocedal、限られた記憶域を有する準ニュートン行列の更新、Mathematics of Computation、35(151): 773 ~ 782、1980)。勾配が明示的に利用可能でないとき、最適化アルゴリズムは数

値近似に頼る。これらの近似を信頼しない場合、別の代替案は、Nelder - Mead アルゴリズム (M. J. D. Powell、最小化アルゴリズムの検索方向について、Mathematical Programming、4 (2) : 193 ~ 201、1973年4月)、パターン検索 (Robert Hooke、および T. A. Jeeves、数値的および統計的問題の「直接検索」解、J. ACM、8 (2) : 212 ~ 229、1961年4月)、粒子群最適化 (James Kennedy、および Russell C Eberhart、粒子群最適化、ニューラルネットワークに関するIEEE国際会議の議事録、1942 ~ 1948頁、1995年)、遺伝的アルゴリズム (Melanie Mitchell、遺伝的アルゴリズムの紹介、MIT Press、Cambridge, MA, USA、1998年)、進化戦略 (Hans - Georg Beyer および Hans - Paul Schwefel、進化戦略 - 包括的な紹介、Natural Computing、1 (1) : 3 ~ 52、2002年5月)、Powellのアルゴリズムとその異なるバージョン (M. J. D. Powell)、導関数を計算せずに、いくつかの変数の関数の最小値を見出すための効率的な方法、コンピュータジャーナル、7 (2) : 155 ~ 162、1964)、に基づくシンプレックス検索のような無勾配方法に頼ることである。しかしながら、前述のアルゴリズムの多くは、せいぜいボックスタイプの制約を受け入れる。

10

【0132】

したがって、(12) ~ (13) を制約のない最適化問題に変換する必要がある。バリア機能のアイデアを使用してこれを実装する。すなわち、 $(P_k)$  のコスト関数

20

【0133】

【数41】

$$J_k(w) = J(w) - \lambda_{k,1} \log q_k(w) + \lambda_{k,2} [q_k(w) \log q_k(w) + (1 - q_k(w)) \log(1 - q_k(w))]$$

【0134】

に置き換え、式中、

【0135】

【数42】

$$\{\lambda_{k,i}\}_{k \geq 0} \quad i = 1, 2$$

30

【0136】

は、減少する正のスカラーのシーケンスである。新しいコスト関数は、最小化  $J(w)$ 、高い実現可能性確率を有する点の選択、および高い不確実性を有する点の探索の間のトレードオフを表す。最適化問題  $(P_k)$  の新しいシーケンスは、無勾配方法を使用できるため、制約のない最適化問題のシーケンスである。

【0137】

準最適解が受け入れられる場合は、反復手順を排除し、コスト関数、

【0138】

【数43】

40

$$J'(w) = J(w) - \lambda_1 \log q(w) + \lambda_2 [q(w) \log q(w) + (1 - q(w)) \log(1 - q(w))]$$

【0139】

を用いてワンショット最適化問題を解くことができ、式中、 $q(w)$  は、モデルのシミュレーションが実行されるたびに更新される実現可能性モデルである。ますます多くの点を探索するにつれてエントロピー項がほぼゼロになるので、この発見的な方法論は満足のいく結果を提供する良い機会を有する。さらに、検索が実現可能な領域内で実行される場合、第2の項はゼロになる。したがって、検索は主に  $J$  を最小化することに焦点を当てている。コスト関数の計算方法を改変するだけでよく、制約のない最適化アルゴリズムを使用できる。コスト関数内の計算は、アルゴリズム3に要約されている。

50

【 0 1 4 0 】

[ 表 ]

**アルゴリズム 3 経験的バリア機能を持つコスト関数疑似コード**


---

```

1:  $D: q(w; \beta)$  についての現在の訓練セット
2: function costFUNCTION( $w$ )
3:   パラメータ  $w$  についてモデルをシミュレートする
4:   if シミュレーションが失敗の場合、then
5:      $y \leftarrow 0$ 
6:      $counter \leftarrow counter + 1$ 
7:      $cost \leftarrow -\log \frac{\varepsilon}{counter}, \varepsilon \ll 1$ 
8:   else
9:      $y \leftarrow 1$ 
10:     $counter \leftarrow 0$ 
11:     $cost \leftarrow J'(w)$ 
12:   訓練データセットを増強する:  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(w, y)\}$ 
13:   増強されたセット  $D$  に基づいて実現可能性確率  $q(w; \beta)$ 
     のパラメータを更新する
14:   return cost

```

---

【 0 1 4 1 】

アルゴリズム 3 の 13 行目においては、パラメータ を最初から再訓練するのではなく、代わりにオンライン勾配降下アルゴリズムを使用でき、命題 3.2 で定義された

【 0 1 4 2 】

【 数 4 4 】

 $\widetilde{H}(\beta, \mathcal{D})$ 

【 0 1 4 3 】

の勾配は、単一の例の勾配で近似され  $-\frac{\partial}{\partial \beta} h(w; (w, y))$ 、式中、 $h(w; (w, y)) = -y \log q(w; \beta) - (1 - y) \log [1 - q(w; \beta)]$ 。勾配を近似するため、または導関数を含まないアルゴリズムで検索の方向をテストするためのコスト関数の呼び出しごとに、実現可能性確率が更新される。

【 0 1 4 4 】

デモ例

いくつかの実装において、目標は、図 3 に示される Cauer フィルタで非線形抵抗器  $R_{nonlinear}$  のモデルを学習することである。出力測定値は、100 秒の時間間隔にわたる抵抗器  $R_4$  の両端の電圧である。電圧は、非線形抵抗器の挙動に関する間接的な情報を提供する。この回路は、入力正規分布信号の帯域制限された白色雑音である電圧源によって電力を供給される。

【 0 1 4 5 】

抵抗器  $R_4$  の電圧の「実験的な」出力測定値を図 4 に示す。非線形抵抗器の「真の」実際の挙動は、 $i = 0.01 \cdot v^3$  によって記述され、式中、 $i$  および  $v$  は、それぞれ抵抗器の電流および電圧を表す。学習プロセスにおいては、多項式  $w_1 i = w_2 v^2 + w_3 v^3$  によって構成方程式をモデル化する。パラメータ値に加えて、 $w$  に対応する実現可能性セットの表現を学習したい。実現可能セットの記述的な 3D 表現を得るために、パラメータの三次元ベクトルを選択する。

10

20

30

40

50

【 0 1 4 6 】

ベクトル  $w$  が実行可能になる確率は、ソフトマックス関数

【 0 1 4 7 】

【 数 4 5 】

$$q(w; \beta) = e^{\beta^T w} / (1 + e^{\beta^T w})$$

【 0 1 4 8 】

を使用してモデル化される。この表現は、実現可能な部分空間および実現不可能な部分空間を分離する超平面を提供する。確かにそのような分離が可能であるという仮定をしていることに留意されたい。そうではない場合がある。より複雑な表現では、NNをソフトマックス関数の前に置くことができる。分離超平面は  $w_1 + w_2 + w_3 = 0$  で与えられる。したがって、 $w$  は実現可能である  $w^T w > 0$ 。

10

【 0 1 4 9 】

学習手順は、PythonのSciPyライブラリ(Eric Jones、Travis Oliphant、Pearu Petersonら、SciPy: Python用オープンソース科学ツール、2001年~)の最適化パッケージからのアルゴリズムを使用してPythonに実装された。最適化コスト関数の評価は、Modelica(Peter Fritzson、Modelica 3.3を用いたオブジェクト指向モデル化とシミュレーションの原理: サイバーフィジカルアプローチ、IEEE Press、Wiley、Hoboken、NJ、第2版、2015)電気回路モデルの説明から生成された、機能モックアップユニット(FMU)(T. Blochwitz、M. Oter、M. Arnold、C. Bausch、C. Clau、H. Elmquist、A. J. Unghanns、J. Mauss、M. Monteiro、T. Neidhold、D. Neumerkel、H. Olsson、J. v. Peetz、S. Wolf、Atego Systems GmbH、Qtronic Berlin、Fraunhofer Scai、およびSt. Augustin、ツールに依存しないシミュレーションモデルの交換のための機能的モックアップインターフェース、第8回国際Modelica会議の議事録、2011年)のシミュレーションに基づいている。

20

【 0 1 5 0 】

最初にアルゴリズム1を実装した。図5A~図5Bは、アルゴリズム1によって生成された分離超平面を示す。より具体的には、図5Aはアルゴリズム1のCOBYLA結果を示し、図5Bはアルゴリズム1のSLSQP結果を示す。内側のループは、制約を受け入れる2つのアルゴリズム、COBYLA(M. J. D. Powell、導関数なしの最適化アルゴリズムの見方、技術報告、ケンブリッジ大学、英国、2007年5月)、制約付き最適化のための導関数を含まないアルゴリズムと、逐次最小二乗計画法(SLSQP)(D. Kraft、逐次二次計画法のためのソフトウェアパッケージ、Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR、1988年)とを使用して実装された。25回の外部反復を実行し、内部反復ごとに反復の最大数を300に制限した。ミニマイザ内の相対誤差および最小値を  $x_{tol} = f_{tol} = 1e-16$  に設定した。シーケンス  $k, 1$  および  $k, 2$  は、 $k, 1 = \frac{1}{2} + (1 - \frac{1}{2})^k$  および  $k, 2 = 1 + \frac{1}{2} - \frac{1}{2}^k$  として選択し、 $\epsilon = 1e-7$  および  $\delta = 0.8$  とした。勾配近似のためのステップサイズは、 $\epsilon = 1e-04$  に設定した。最適化手順の結果を表2に要約する。

30

40

【 0 1 5 1 】

【表 2】

アルゴリズム	解 $w^*$	平均誤差
<b>COBYLA</b>	<b>[1,-9.31240083e-04,9.22597610e-03]</b>	<b>0.0028635</b>
<b>SLSQP</b>	<b>[1, -0.00170486, 0.00951815]</b>	<b>0.00273461</b>

表 2 : アルゴリズム 1 の無勾配アルゴリズムの実装、および勾配依存のアルゴリズムの実装

【 0 1 5 2 】

COBYLA を使用して蓄積されたデータは、実現可能パラメータおよび実現不可能パラメータの間のより明確な分離を示す。SLSQP は、大部分が分離している超平面の周りの点を調べた。

10

【 0 1 5 3 】

アルゴリズム 3 は、無勾配アルゴリズムおよび勾配近似に基づくアルゴリズムの両方でテストされた。第 2 のクラスは勾配近似を制御するハイパーパラメータに依存するため、2 つのクラスのアルゴリズムを比較することはすぐにはできない。したがって、同じクラスにのみ属するアルゴリズムを比較する。反復の最大数および関数評価の最大数を、それぞれ 1000 および 1500 に制限した。アルゴリズム 3 のハイパーパラメータは  $\alpha_1 = 1$  および  $\alpha_2 = 0.1$  として選ばれた。全てのアルゴリズムに同じ初期条件、すなわち  $w_0 = [0.1, 0.1, 0.1]$  を使用する。ミニマイザ内の相対誤差および最小値を  $x_{tol} = f_{tol} = 1e-16$  に設定した。無勾配アルゴリズムを使用した最適化結果を表 3 に示す。解  $w^*$  はスカラー倍増に対して不変であるので、解を最初の要素

20

【 0 1 5 4 】

【数 4 6】

$$w^* \leftarrow \frac{1}{w_1} w^*$$

【 0 1 5 5 】

で除算することができる。

【 0 1 5 6 】

【表 3】

アルゴリズム	解 $w^*$	平均誤差
<b>Powell</b>	<b>[1,-2.20299821e-04,9.85091226e-03]</b>	<b>0.00274897</b>
<b>Nelder</b>	<b>[1,-2.37604458e-04,9.75878109e-03]</b>	<b>0.00253461</b>
<b>COBYLA</b>	<b>[1,-9.88099227e-04,9.18751113e-03]</b>	<b>0.00361726</b>

30

表 3 : 導関数を含まない最適化アルゴリズムの比較

【 0 1 5 7 】

【数 4 7】

$$w_2^*$$

40

【 0 1 5 8 】

および

【 0 1 5 9 】

【数 4 8】

$$w_3^*$$

【 0 1 6 0 】

50

に興味がある。特に、3つの全てのアルゴリズム

【0161】

【数49】

$w_3^*$

【0162】

について、その真の値に近づく。無勾配方法のための分離超平面は、図6A～6Cに示され、特に、図6Aは、これをPowellに関して示し、図6Bは、これをNelder-Meadに関して示し、図6Cは、これをCOBYLAに関して示す。各アルゴリズムが異なる超平面を学習したことは、驚くべきことではない。これらの超平面のパラメータは、各アルゴリズムによって生成された軌跡、アルゴリズムごとに異なる軌跡に基づいて計算される。

10

【0163】

次に、勾配近似を使用するアルゴリズム、Broyden-Fletcher-Goldfarb-Shanno (BFGS) アルゴリズム、限定メモリBFGSアルゴリズム (L-BFGS) (Jorge Nocedal、限られた記憶域を有する準ニュートン行列の更新、Mathematics of Computation、35(151):773～782、1980)、非線形共役勾配アルゴリズム (CG) (R. Fletcher および C. M. Reeves、共役勾配による関数最小化、コンピュータジャーナル、7(2):149～154、1964)、およびSLSQPを使用した。結果を表4に示す。無勾配アルゴリズムの場合に使用される設定に加えて、勾配近似のステップサイズを $\epsilon = 1e-4$ に設定する。

20

【0164】

【表4】

アルゴリズム	解 $w^*$	平均誤差
<b>BFGS</b>	<b>[1,8.28670889e-05,9.76706821e-03]</b>	<b>0.00276546</b>
<b>SLSQP</b>	<b>[1,-0.00254604,0.00777795]</b>	<b>0.00769311</b>
<b>CG</b>	<b>[1,-0.00137244,0.00865319]</b>	<b>0.00464650</b>
<b>L-BFGS</b>	<b>[1,-6.96951339e-04,9.79259844e-03]</b>	<b>0.00254597</b>

30

表4：導関数を含まない最適化アルゴリズムの比較

【0165】

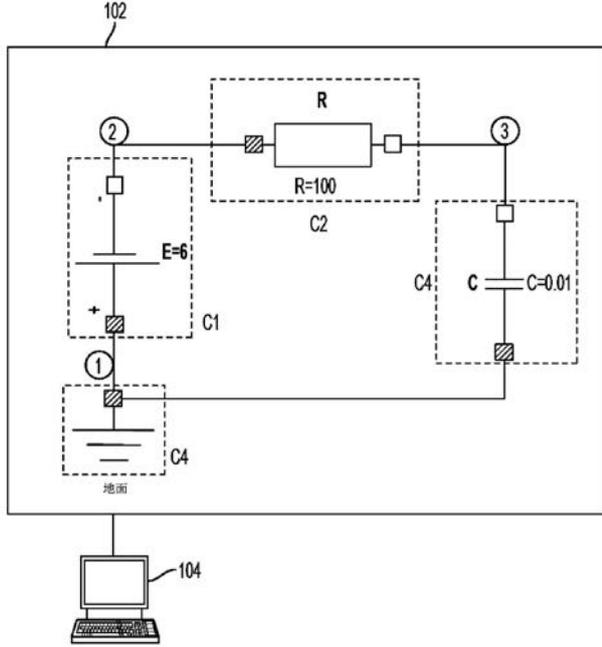
図7A～図7Dは、勾配近似を使用するアルゴリズムについて分離している超平面を示す。特に、図7Aは、これをBFGSに関して示し、図7Bは、これをSLSQPに関して示し、図7Cは、これをCGに関して示し、図7Dは、これをL-BFGS-Bに関して示す。

【0166】

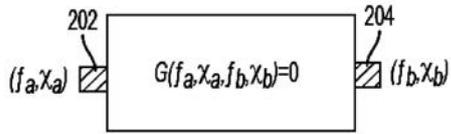
本明細書に開示された技術は、開示された技術を実行するために、電子データ処理デバイスによって読み取り可能および実行可能な命令を記憶する非一時的記憶媒体によって、具現化され得ることがさらに理解されるであろう。そのような非一時的記憶媒体は、ハードドライブまたは他の磁気記憶媒体、光ディスクまたは他の光学記憶媒体、RAIDディスクアレイなどのクラウドベースの記憶媒体、フラッシュメモリまたは他の不揮発性電子記憶媒体などを含むことができる。

40

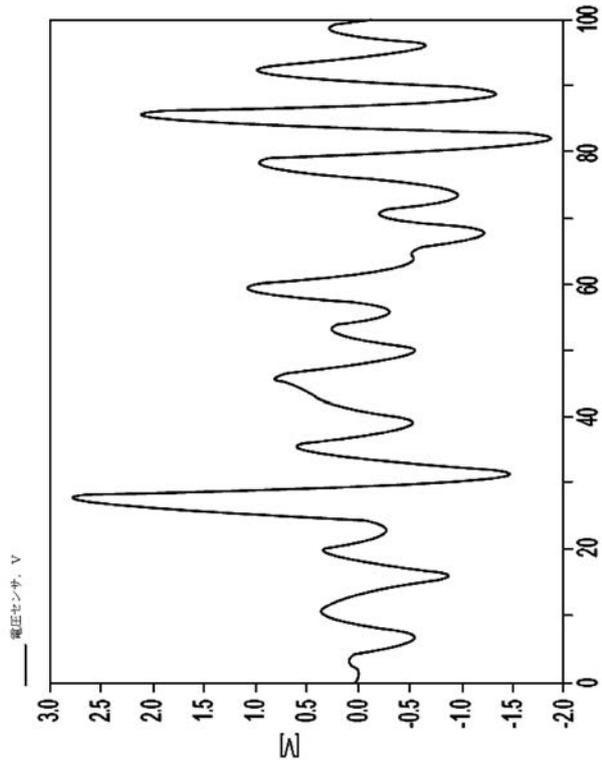
【 図 1 】



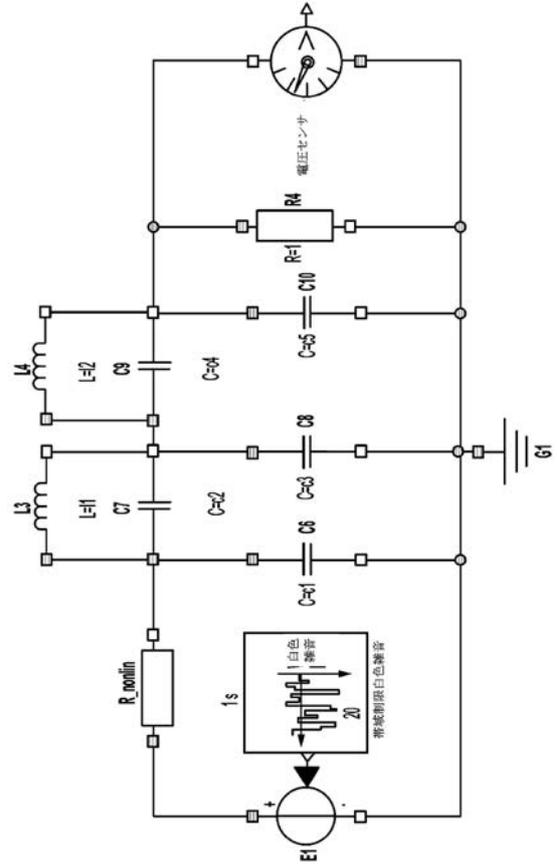
【 図 2 】



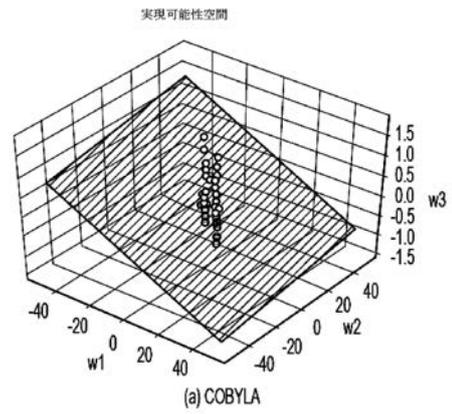
【 図 4 】



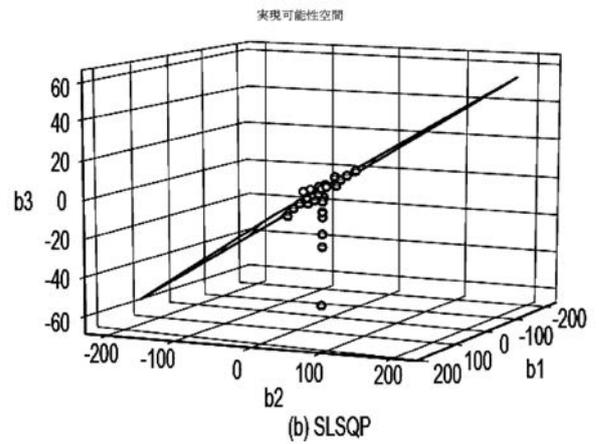
【 図 3 】



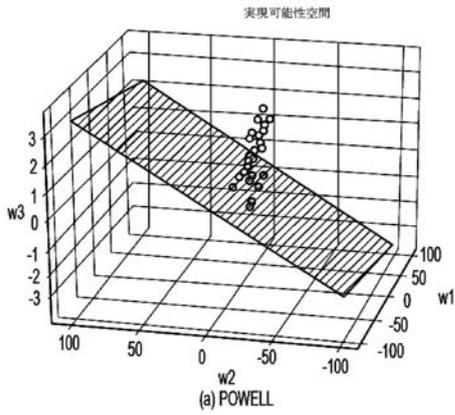
【 図 5 A 】



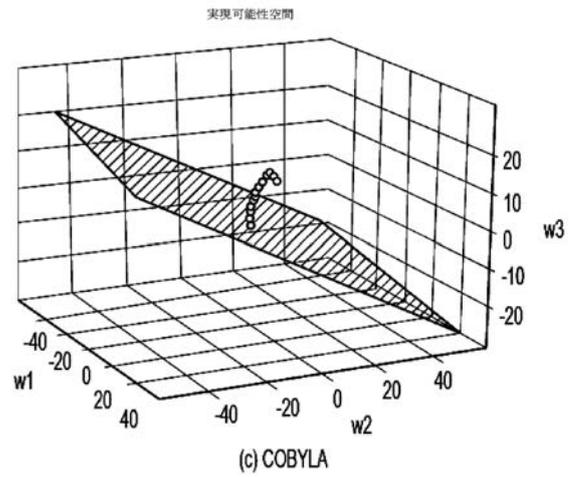
【 図 5 B 】



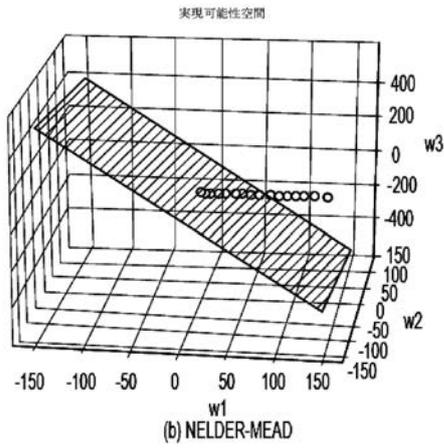
【 図 6 A 】



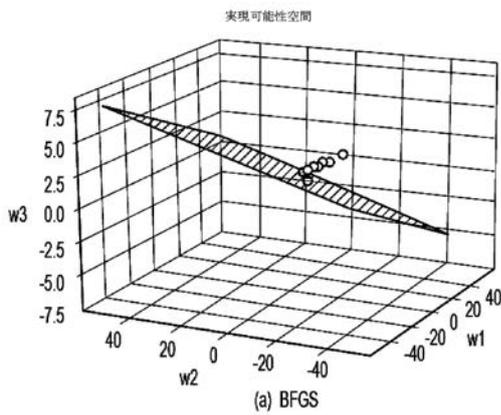
【 図 6 C 】



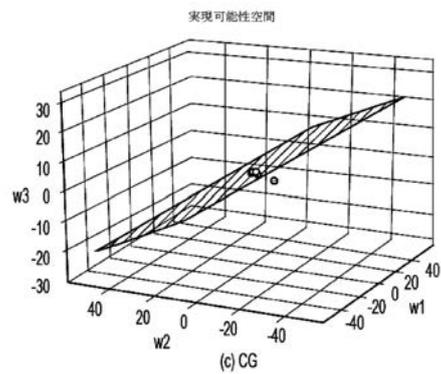
【 図 6 B 】



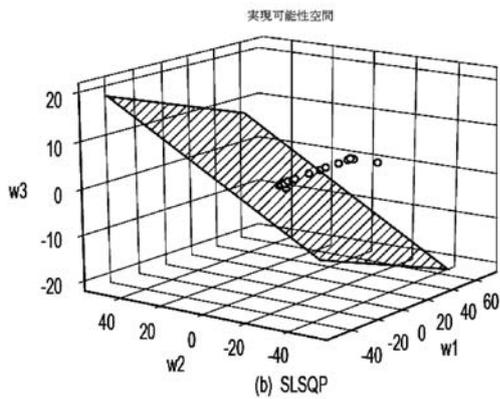
【 図 7 A 】



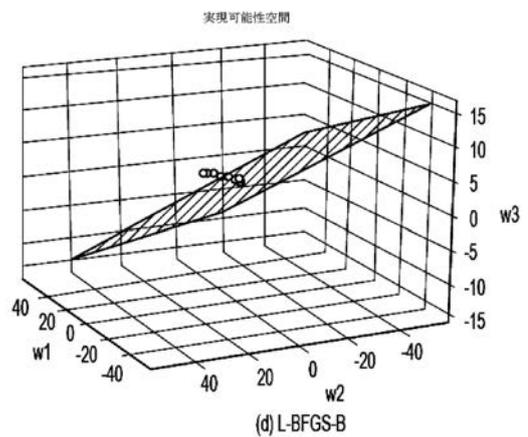
【 図 7 C 】



【 図 7 B 】



【 図 7 D 】



---

フロントページの続き

(74)代理人 100109335

弁理士 上杉 浩

(74)代理人 100120525

弁理士 近藤 直樹

(74)代理人 100139712

弁理士 那須 威夫

(74)代理人 100158551

弁理士 山崎 貴明

(72)発明者 イオン・マテイ

アメリカ合衆国 カリフォルニア州 9 4 0 8 7 サニーベール キングフィッシャー・テラス  
8 2 4

(72)発明者 ヨハン・デクリーア

アメリカ合衆国 カリフォルニア州 9 4 0 2 4 ロスアルトス パコ・ドライブ 6 8 5

Fターム(参考) 5B046 AA07 AA08 BA01 JA04