

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4900642号  
(P4900642)

(45) 発行日 平成24年3月21日 (2012.3.21)

(24) 登録日 平成24年1月13日 (2012.1.13)

(51) Int.Cl.

G06N 3/00 (2006.01)

F I

G06N 3/00 550E

請求項の数 5 (全 17 頁)

(21) 出願番号	特願2005-169458 (P2005-169458)	(73) 特許権者	000002185
(22) 出願日	平成17年6月9日 (2005.6.9)		ソニー株式会社
(65) 公開番号	特開2006-268813 (P2006-268813A)		東京都港区港南1丁目7番1号
(43) 公開日	平成18年10月5日 (2006.10.5)	(74) 代理人	100082131
審査請求日	平成20年6月2日 (2008.6.2)		弁理士 稲本 義雄
(31) 優先権主張番号	特願2005-47491 (P2005-47491)	(72) 発明者	日台 健一
(32) 優先日	平成17年2月23日 (2005.2.23)		東京都品川区東五反田3丁目14番13号
(33) 優先権主張国	日本国 (JP)		ソニーインテリジェンスダイナミクス研 究所株式会社内
		(72) 発明者	佐部 浩太郎
			東京都品川区東五反田3丁目14番13号
			ソニーインテリジェンスダイナミクス研 究所株式会社内
		審査官	長谷川 篤男

最終頁に続く

(54) 【発明の名称】 学習制御装置、学習制御方法、およびプログラム

(57) 【特許請求の範囲】

【請求項 1】

1 以上の変数を入力として 1 以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置において、

入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定手段と、

前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定手段と、

第 1 の機能モジュールと、前記第 1 の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第 1 の機能モジュールとは異なる第 2 の機能モジュールとをつなぎ合わせることで前記第 1 および第 2 の機能モジュールを階層化する階層化手段と

を含む学習制御装置。

【請求項 2】

前記入出力変数決定手段は、前記変数間因果関係情報に基づき、相互に前記因果関係がある 2 以上の変数の集合を変数グループとし、前記変数グループ毎に、前記変数グループに含まれる 2 以上の変数を前記機能モジュールの制御対象変数に決定する

10

20

請求項 1 の学習制御装置。

【請求項 3】

前記入出力変数決定手段は、前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定し、決定した前記行動変数のうち、前記制御対象変数から直接介入可能なもの以外を状態変数に決定する

請求項 1 に記載の学習制御装置。

【請求項 4】

1 以上の変数を入力として 1 以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置の学習制御方法において、

前記学習制御装置による、

入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定ステップと、

前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定ステップと、

第 1 の機能モジュールと、前記第 1 の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第 1 の機能モジュールとは異なる第 2 の機能モジュールとをつなぎ合わせることで前記第 1 および第 2 の機能モジュールを階層化する階層化ステップと

を含む学習制御方法。

【請求項 5】

1 以上の変数を入力として 1 以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置の制御用のプログラムであって、

入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定ステップと、

前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定ステップと、

第 1 の機能モジュールと、前記第 1 の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第 1 の機能モジュールとは異なる第 2 の機能モジュールとをつなぎ合わせることで前記第 1 および第 2 の機能モジュールを階層化する階層化ステップと

を含む処理を学習制御装置のコンピュータに実行させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、学習制御装置、学習制御方法、およびプログラムに関し、特に、多次元の変数（センサ入力、内部状態、およびモータ出力等）からなる自律学習エージェントにおいて、予測器の学習を元に変数間の因果関係を推定し、推定した因果関係に基づき、制御器の数・機能・入出力変数を自動的に決定し、制御器のモジュール化・階層化の設計を自動化するようにした学習制御装置、学習制御方法、およびプログラムに関する。

【背景技術】

【0002】

（A） 非特許文献 1 に記載されている強化学習型の自律エージェント構成手法や非特許文献 2 に記載されている経験強化型の自律エージェント構成手法においては、学習器の

10

20

30

40

50

入出力変数が、解くべきタスクや期待する動作を考慮した人間によって選択されていた。

【0003】

一方、多自由度の自律エージェントを考えると、設計時にタスクを決定して入出力変数を決めることは、自律エージェントの学習可能な能力を設計時に限定することになってしまう。このことから、従来手法はOpen-endedな自律エージェントの構築において決定的な問題をはらんでいるといえる。

【0004】

(B) (A)の問題点を解消するために、考えられる全てのセンサ・モータ変数を入出力として用いると、個々の特定のタスクや期待する動作の性能に影響を与えてしまうという問題が生じる。これは、機械学習の世界では次元の呪いとして広く知られている問題である(例えば、非特許文献3参照)。

10

【0005】

(C) (B)の問題点を解決する為に、自立エージェントを複数の機能モジュールに分割して学習することを考える。この場合、単体の機能モジュールとして学習していたときには存在しなかった新たな2つの問題が発生する。

【0006】

1つは、各機能モジュールの個数と各機能モジュールの自由度(1つのモジュールがどこまで複雑な構造を持ちうるかを決定する量)をどのように決定するか、という問題である。もう1つは、各機能モジュール間の結合をどのように決定するか、という問題である。

20

【0007】

(D) 特許文献1に記載されているMOSAICは、機能のモジュール化という点で興味深いアプローチであるが、各モジュールは与えられた全ての変数を入出力として扱うことになり、結局は(C)で提示された問題に対しての解決策になっていない。解決するためには、各機能モジュール間の個々の機能と結合関係を事前に人間が設計する必要が生じる。この点に関して言えばMOSAICは従来型の自律エージェントに対しする設計アプローチと同じである。

【0008】

【特許文献1】特開2000-35804号公報

【非特許文献1】Richard S.Sutton, Andrew G.Barto, "Reinforcement Learning: An Introduction," MIT Press, 1998

30

【非特許文献2】Jun Tani: "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes", Neural Networks, Vol.16, No.1, pp.11-23, 2003.

【非特許文献3】Bellman, R.E. "Dynamic Programming." Princeton University Press, Princeton. 6: 679-684.

【非特許文献4】多賀源太郎著、「脳と身体の動的デザイン 運動・知覚の非線形力学と発達」金子書房、2002

【発明の開示】

【発明が解決しようとする課題】

40

【0009】

(D)の問題点を計決する為に、機能モジュールの機能と、その結合関係を自動的に学習することを考える。この場合、モジュールの数や個々のモジュールの複雑度を定める方法が自明ではなく、未だ人間の設計する余地が残っている。また、実際にこの手法を用いて複数のタスクや複数の期待する動作が非破壊的に追加学習されている事例は皆無である、という課題があった。

【0010】

また、脳科学・神経科学・心理学では、ヒトは機能のモジュール性(局在性)を有していると言われている。同様に、人工的な自律エージェントの学習にも何らかのモジュール性・階層性を導入すると良い、という議論は多くされているが、実際にそれらモジュールや

50

階層間の結合・統合を行なう決定的な手法は確立されていない（非特許文献4参照）、という課題があった。

【0011】

本発明はこのような状況に鑑みてなされたものであり、多次元の変数（センサ入力、内部状態、およびモータ出力等）からなる自律学習エージェントにおいて、予測器の学習を元に変数間の因果関係を推定し、推定した因果関係に基づき、制御器の数・機能・入出力変数を自動的に決定し、制御器のモジュール化・階層化の設計を自動化できるようにするものである。

【課題を解決するための手段】

【0012】

本発明の一側面である学習制御装置は、1以上の変数を入力として1以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置において、入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定手段と、前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定手段と、第1の機能モジュールと、前記第1の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第1の機能モジュールとは異なる第2の機能モジュールとをつなぎ合わせることで前記第1および第2の機能モジュールを階層化する階層化手段とを含む。

【0013】

前記入出力変数決定手段は、前記変数間因果関係情報に基づき、相互に前記因果関係がある2以上の変数の集合を変数グループとし、前記変数グループ毎に、前記変数グループに含まれる2以上の変数を前記機能モジュールの制御対象変数に決定することができる。

前記入出力変数決定手段は、前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定し、決定した前記行動変数のうち、前記制御対象変数から直接介入可能なもの以外を状態変数に決定することができる。

【0014】

本発明の一側面である学習制御方法は、1以上の変数を入力として1以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置の学習制御方法において、前記学習制御装置による、入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定ステップと、前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数となる制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定ステップと、第1の機能モジュールと、前記第1の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第1の機能モジュールとは異なる第2の機能モジュールとをつなぎ合わせることで前記第1および第2の機能モジュールを階層化する階層化ステップとを含む。

【0015】

本発明の一側面であるプログラムは、1以上の変数を入力として1以上の値を出力する関数から成る機能モジュールによって構成された学習制御装置の制御用のプログラムであって、入力となる変数の組み合わせを複数生成し、生成した複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせを選択し、選択した前記組み合わせを用いて、前記ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報を生成する因果関係推定ステップと、前記変数間因果関係情報に基づき、前記機能モジュールへの入力変数とな

10

20

30

40

50

る制御対象変数を決定し、前記機能モジュールからの出力変数となる行動変数を決定する入出力変数決定ステップと、第1の機能モジュールと、前記第1の機能モジュールの制御対象変数と同じ変数を行動変数とする、前記第1の機能モジュールとは異なる第2の機能モジュールをつなぎ合わせることで前記第1および第2の機能モジュールを階層化する階層化ステップを含む処理を学習制御装置のコンピュータに実行させる。

【0016】

本発明の一側面においては、入力となる変数の組み合わせが複数生成され、生成された複数の組み合わせのうち、それを参照してある変数を予測した時の予測誤差が最小となる組み合わせが選択され、選択された組み合わせを用いて、ある変数の未来の値を予測するためには他のどの変数を参照することが必要であるかの因果関係を示す変数間因果関係情報が生成され、変数間因果関係情報に基づき、機能モジュールへの入力変数となる制御対象変数が決定され、機能モジュールからの出力変数となる行動変数が決定される。また、第1の機能モジュールと、第1の機能モジュールの制御対象変数と同じ変数を行動変数とする、第1の機能モジュールとは異なる第2の機能モジュールをつなぎ合わせることで第1および第2の機能モジュールが階層化される。

10

【発明の効果】

【0017】

本発明によれば、多次元の変数（センサ入力、内部状態、およびモータ出力等）からなる自律学習エージェントにおいて、予測器の学習を元に変数間の因果関係を推定し、推定した因果関係に基づき、制御器の数・機能・入出力変数を自動的に決定し、制御器のモジュール化・階層化の設計を自動化することが可能となる。

20

【発明を実施するための最良の形態】

【0024】

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。

【0025】

図1は、本発明を適用した自律エージェントの構成例を示している。この自律エージェント1は、1つの関節を有し、当該関節にトルクを掛けることで回転するリンク11から成る単純なロボットである。この自律エージェント1は、リンク11の角度および角速度を検出して入力とし、関節に与えるトルクを出力する。自律エージェント1が存在する環境には重力（ $=0$ となる方向）も存在し、発生可能なトルクは重力に比べて小さいので、仮に一定方向に最大トルク $_{max}$ を発生させても自律エージェント1自身が持ち上げられたり、移動されたりすることはない。

30

【0026】

さらに、自律エージェント1は、図2に示すように、内部状態を示す変数としてエネルギー $E$ および乳酸 $L$ を有している。エネルギー $E$ は、発生したトルクに比例して消費（減少）する。そして、自律エージェント1の餌21が想定された所定の位置 $_E$ にリンク11が到達すると、エネルギー $E$ が所定の値だけ増加する。乳酸 $L$ は、消費されたエネルギー $E$ に比例して増加し、定常的に僅かずつ分解されて減少する。自律エージェント1は、エネルギー $E$ が無くなる、または乳酸 $L$ が最大値 $L_{MAX}$ に達すると、トルクを発生することができなくなる。

40

【0027】

図3は、自律エージェント1の内部の構成例を示している。

【0028】

トルク発生部31はモータ等から成り、トルクを発生してリンク11を移動させる。また、発生したトルクを自律制御部36に通知する。角度検出部32はリンク11が垂線と成す角度を検出して自律制御部36に通知する。角速度検出部33はリンク11の角速度を検出して自律制御部36に通知する。エネルギー設定部34は、トルク発生部31が発生したトルクに基づいてエネルギー $E$ を減少させ、角度検出部32によって検出される角度に基づいてエネルギー $E$ を増加させる。乳酸設定部35は、トルク発生部31が

50

発生したトルク に対応して消費（減少）されたエネルギー E に基づいて乳酸 L を増加させ、所定の周期で乳酸 L を減少させる。また現在のエネルギー E を自律制御部 36 に通知する。

【0029】

自律制御部 36 の予測器学習部 41 は、変数（ $\theta$ ,  $\omega$ , E, L,  $\tau$ ）のそれぞれについて学習により、後述する微分方程式（1）を取得して変数グループ生成部 42、行動変数決定部 43、および状態変数決定部 44 に出力する。変数グループ生成部 42 は、予測器学習部 41 によって取得された微分方程式（1）に基づき、変数（ $\theta$ ,  $\omega$ , E, L,  $\tau$ ）のうちで相互に原因となっている変数をグループ化して、変数グループを行動変数決定部 43 および状態変数決定部 44 に出力する。行動変数決定部 43 は、予測器学習部 41 によって取得された微分方程式（1）、および変数グループ生成部 42 から入力される変数グループに基づき、各変数グループに対する行動変数を決定して状態変数決定部 44 に出力する。状態変数決定部 44 は、予測器学習部 41 によって取得された微分方程式（1）、変数グループ生成部 42 から入力される変数グループ、および行動変数決定部 43 によって決定された行動変数に基づき、各変数グループに対する状態変数を決定して、行動変数とともに階層化部 45 に出力する。階層化部 45 は、変数グループの行動変数および状態変数に基づき、変数グループと行動変数の組み合わせに対応する制御器を生成し、生成した制御器を階層化する。

10

【0030】

次に、自律エージェント 1 の力学モデルに適用される微分方程式（1）を示す。ただし、各関数 f の具体的な中身についてはここで問題としない。

20

【数 1】

$$\dot{\theta} = f_{\theta}(\theta, \omega, \tau)$$

$$\dot{\omega} = f_{\omega}(\theta, \omega, \tau)$$

$$\dot{E} = f_E(\theta, \tau)$$

$$\dot{L} = f_L(\tau)$$

30

... (1)

【0031】

また、自律エージェント 1 の力学モデルは、現在時刻 t、および微小時間  $\Delta t$  を用いた次式（2）に示す差分方程式によっても表すことができる。

【数 2】

$$\theta_{t+\Delta t} - \theta_t = f_{\theta}(\theta_t, \omega_t, \tau_t)$$

$$\omega_{t+\Delta t} - \omega_t = f_{\omega}(\theta_t, \omega_t, \tau_t)$$

$$E_{t+\Delta t} - E_t = f_E(\theta_t, \tau)$$

$$L_{t+\Delta t} - L_t = f_L(\tau_t)$$

40

... (2)

【0032】

図 4 は、自律エージェント 1 の力学モデルを因果ダイアグラムで表したものである。

【0033】

なお、式（1）に示された微分方程式よりも式（2）に示された差分方程式による定式化の方がコンピュータを適用した実装を考えると実用的であるが、簡便性を考慮して、以下、微分方程式（1）を用いて説明する。もちろん、微分方程式（1）を差分方程式（2）に置き換えることも可能である。また、以下、微分方程式（1）に含まれる関数 f を適

50

宜「予測器」と称する。

【0034】

因果関係の推定と予測器の学習

予測器学習部41においては、上記力学モデルの差分方程式(2)や図4に示された因果ダイアグラムは未知のものとして、上記差分方程式を近似的に復元する。この処理は、変数間の定性的な因果関係を発見する第1の処理(因果ダイアグラムの復元)と、その後に定量的な関係を近似的に学習する第2の処理(上記関数fの構造を復元)からなる。このときに手がかりとして与えられる情報は、全ての変数(θ, ω, E, L, τ)の時系列値である。

【0035】

当該第1の処理は、人間がある対象の変動を予測しようとしたときに、その対象が何を原因に変動しているのかを突き止めようとすることに相当する。実際に人間の場合、無限に考えられる世界の構成要素の中からいくつかの事象を取り上げ、予測対象の原因として理解(発見)することができる。このとき用いることができる情報は、環境を観察することで得られる事象(変数)の時系列情報である。

【0036】

ところで、従来の機械学習では因果関係に関する情報を明に用いていなかったため、次式(3)に示されるように、全ての変数を予測器の入力として取り扱うような方法が用いられてきた。

【数3】

$$\dot{\theta} = f_{\theta}(\theta, \omega, E, L, \tau)$$

$$\dot{\omega} = f_{\omega}(\theta, \omega, E, L, \tau)$$

$$\dot{E} = f_E(\theta, \omega, E, L, \tau)$$

$$\dot{L} = f_L(\theta, \omega, E, L, \tau)$$

... (3)

【0037】

上述した従来の方法を採用すると、変数の数が増えるにしたがって学習に必要な訓練サンプル(変数の時系列値)の数が増大し、実質的に学習が困難か不可能な状態(いわゆる「次元の呪い」と称される状態)に陥ってしまう。

【0038】

そこで予測器学習部41においては、関数fの入力になる変数を取捨選択するようになっている。このための1つの方法として、入力となる変数の組み合わせを仮説として複数生成し、その中から最も予測誤差の少ない仮説を選択する方法を考える。例えば、角度θを予測する仮説の関数fとして、次式(4)に示す関数fが考えられる。

【数4】

$$\dot{\theta} = f_{\theta}(\cdot) \quad \dot{\theta} = f_{\theta}(\theta, \omega, \tau)$$

$$\dot{\theta} = f_{\theta}(\theta) \quad \dot{\theta} = f_{\theta}(\theta, \omega, E, L)$$

$$\dot{\theta} = f_{\theta}(\theta, \omega) \quad \dot{\theta} = f_{\theta}(\theta, \omega, E, L, \tau)$$

... (4)

【0039】

式(4)は考えられる仮説の関数fの一部である。

【0040】

入力変数を1つも取らない関数f(・)から全ての変数を入力とする関数f(θ, ω, E, L, τ)まで、

10

20

30

40

50

、 $E, L, \dots$ ものまで、全ての組み合わせを数え上げると、いまの場合、変数の数が5であるため、 $1 \times 2 \times 1 (= 5 \times 4 \times 3 \times 2 + 1)$ 通りの組み合わせが考えられる。このままでは、変数の数が増えるに従って組み合わせの数が爆発的に増加してしまうため、組み合わせ数の上限を決めたりするなどの工夫が必要となる。

【0041】

当該第2の処理においては、仮説の関数 $f$ を決定した後、これら仮説の関数 $f$ に対して同一の訓練サンプル（環境を観察することで得られる事象（変数）の時系列情報）を与えて関数近似を行う。

【0042】

関数近似の方法にはどのような方法を使ってもかまわない。例えば、Support Vector Regression（以下、SVRと略記する）を用いて学習を行う。SVRは、常に大域解が得られることが保障されており、一般的によく用いられている誤差逆伝播法に基づいたニューラルネットワークよりも安心して利用できる点が強みである。

【0043】

SVRを利用した学習によって、各仮説ごとに関数 $f$ の構造が求められる。次に、各仮説がどれだけ良い予測結果を出力できるかを検証するために、訓練サンプルとは異なる評価サンプルを用いて評価する。この手法は学習器の予測性能に対応する汎化誤差を求める手法であり、cross validationと称される。ここでは他のどのような評価基準を用いてもよいが、訓練誤差ではなく汎化誤差を求める種類の手法でなければならない。最も評価の高い仮説が選択されると、学習が成功している場合、式（1）を得ることができる。

【0044】

以上に説明した処理を用いて、力学モデルの近似学習を行うことが可能である。以下では、各変数がどの変数を原因変数とすることで予測可能であるか、という情報（因果関係）が重要となる。

【0045】

直接介入・間接介入と制御器

ある変数の値を任意の値に変化させたいとする。しかしながら、その変数を直接変化させることができないならば、当該変数以外の変数を変化させることによって、間接的に当該変数を変化させることを考える。例えば、人間は念じるだけでは物（例えば椅子など）の位置を変化させることはできないが、例えば自分の筋肉を収縮させて自分の体を操り、最終的に物の位置を変化させることと同様である。

【0046】

上述した自律エージェント1に当てはめて考える。自律エージェント1はエネルギー $E$ が減少してしまったとしても、直接的にエネルギーを増加させることはできない。これに対して、トルクは直接操作可能な変数である。よって、トルクを変化させることにより間接的にエネルギー $E$ の値を増加させることが可能である。このような変数間の因果関係を利用して動作を行う機能モジュールのことを、以後「制御器」と称する。

【0047】

制御器の構成

制御器は、予測器と同様に関数で表される。制御対象の変数を $x$ 、状態変数を $s$ 、介入に用いる変数（行動変数）を $a$ とした場合、制御器（関数） $g_x$ は次式（5）に示されるとおりとなる。

$$a = g_x(x, s) \quad \dots (5)$$

【0048】

放棄

なお、式（5）では、 $x, s, a$ を変数として表記したが、実際には変数の集合であってベクトルであると解釈すればよい。すなわち、 $x$ は1以上、 $s$ は0以上、 $a$ は1以上の要素を持つベクトルである。要素の数が0のときは変数を取らず、1のときは単一変数（スカラー）である。

【0049】

10

20

30

40

50



また、ある変数を制御する制御器を作る場合、各  $x$  ,  $s$  ,  $a$  にどの変数を対応付ければよいのかを決めなければならないが、何も考慮せずに全ての変数を対応させると上述した「次元の呪い」の問題に再び陥ってしまうので配慮が必要である。

【 0 0 5 0 】

変数のグループ化と制御対象変数

複数の変数が、相互に変化の原因となっている場合（すなわち、相互に原因変数となっている場合）、それらの変数は本質的に1つの事象を複数の変数で表現していると捉えることができる。この場合、その中でどれか1つの変数を変化させようとしてもうまく行かず、相互に原因変数となっている変数群をまとめて制御対象とすることが必要になる。以下、このような相互に原因変数となっている変数の集合を変数グループと称する。変数グループ  $x_n$  ( $n = 1, 2, \dots, N$ ) は、図5に示すアルゴリズムを用いて生成される。なお、相互に変化の原因となっていない変数も、その変数1つだけで構成される変数グループとみなすことにする。

10

【 0 0 5 1 】

上述した自律エージェント1においては、角度 と角速度 が相互に原因変数となっている。そのため、角度 だけ、もしくは角速度 だけを制御しようとしても、うまく制御できる制御器は学習できない。角度 と角速度 を単一の変数グループとして、それぞれを同時に制御する 制御器を考える必要がある。

【 0 0 5 2 】

変数のグループ化により、上述した変数  $x$  ,  $s$  ,  $a$  のうちの制御対象の変数  $x$  が決定される。自律エージェント1においては、次式(6)に示される3つの制御対象変数グループ  $x_1$  ,  $x_2$  ,  $x_3$  が決定される。

20

$$x_1 = \{ \quad , \quad \}$$

$$x_2 = \{ E \}$$

$$x_3 = \{ L \}$$

... ( 6 )

【 0 0 5 3 】

図6は変数をグループ化した後の因果ダイアグラムである。

【 0 0 5 4 】

行動変数

30

上述したように制御対象の変数グループを決定した後、各変数グループに対して制御器を構成する。そこで、それら制御器  $g_x$  の出力となる行動変数を考える。行動変数となり得る変数は、制御が可能な変数のみである。制御が可能な変数は2種類あり、1つは直接介入による制御が可能な生まれつきの行動変数、もう1つは制御器の学習によって間接介入による制御が可能になった変数である。

【 0 0 5 5 】

それら候補となる行動変数からどの組み合わせを選択するかについての自由度が残っている。ここでは、全ての変数の組み合わせを仮説として同時に生成する。それら全てを学習し、実際の行動時にはそのうちどれかを自由に選択して利用できるようにする。これは一見冗長で不要な自由度に思えるが、例えばプランニングアルゴリズム  $A^*$  における行動選択を、この制御器の仮説の選択としてとらえることで、有用な自由度としてとらえ直すことができる。

40

【 0 0 5 6 】

具体的には、各変数グループに属する制御対象変数の原因変数を、学習された予測器から取得して原因変数集合を生成する。次に原因変数集合から制御対象変数が除外され、原因変数集合に残ったものが行動変数とされる。ただし、行動変数が変数グループに属している場合、当該変数グループに属する全ての変数が行動変数とされる。

【 0 0 5 7 】

例えば、変数グループ  $\{ \quad , \quad \}$  に対しては、変数グループに属する制御対象変数 の原因変数  $\{ \quad , \quad , \quad \}$  と、制御対象変数 の原因変数  $\{ \quad , \quad , \quad \}$  が取得され、原因

50

変数集合 { , , } が生成される。そして、原因変数集合 { , , } から制御対象変数 , が除外されて、残った変数 が行動変数とされる。

【 0 0 5 8 】

また例えば、変数グループ { E } に対しては、変数グループに属する制御対象変数 E の原因変数 { , } が取得されて原因変数集合 { , } が生成される。そして、原因変数集合 { , } から制御対象変数 E が除外されて（実際、原因変数集合 { , } に制御対象変数 E は含まれていないが）、残った変数 , が行動変数とされる。ただし、変数 は変数グループ { , } に属しているので、変数グループ { , } が行動変数とされる。

【 0 0 5 9 】

さらに例えば、変数グループ { L } に対しては、変数グループに属する制御対象変数 L の原因変数 が取得されて原因変数集合 { } が生成される。そして、原因変数集合 { } から制御対象変数 L が除外されて（実際、原因変数集合 { } に制御対象変数 L は含まれていないが）、残った変数 が行動変数とされる。

【 0 0 6 0 】

上述した自律エージェント 1 の行動変数を図 7 に示す。

【 0 0 6 1 】

状態変数

制御対象の変数グループに対する行動変数を決定した後、それらに対応する状態変数を決定する。具体的には、原因変数グループから直接介入可能な行動変数を除外したものを状態変数とする。ただし、状態変数が変数グループに属している場合、当該変数グループに属する全ての変数が状態変数とされる。

【 0 0 6 2 】

自律エージェント 1 の状態変数を図 8 に示す。

【 0 0 6 3 】

制御器の階層化

トルク を行動変数とした 制御器を  $g_{\theta, \omega, \tau}$ 、角度 と角速度 を行動変数にした E 制御器を  $g_{E, \theta, \omega}$  のように、制御器の関数  $g$  の添え字に制御対象変数と行動変数をカンマで区切って示す表記法を用いると、それらは次式 ( 7 ) のようになる。ここで、添え字に  $d$  が付いている変数は、観測値を表す変数ではなく、その変数の目標値を表す変数である。間接介入による制御であるため、その値はあくまでも目標値である。ただし、 $d$  は直接介入可能な行動変数であるため、 $\tau$  に読み替えることが可能である。

【数 5】

$$\tau_d = g_{\theta, \omega, \tau} \left( \begin{pmatrix} \theta_d \\ \omega_d \end{pmatrix}, \begin{pmatrix} \theta \\ \omega \end{pmatrix} \right)$$

$$\begin{pmatrix} \theta_d \\ \omega_d \end{pmatrix} = g_{E, \theta, \omega} \left( E_d, \begin{pmatrix} \theta \\ \omega \end{pmatrix} \right)$$

$$\tau_d = g_{E, \tau} \left( E_d, \begin{pmatrix} \theta \\ \omega \end{pmatrix} \right)$$

$$\tau_d = g_{L, \tau} (L_d)$$

... ( 7 )

【 0 0 6 4 】

10

20

30

40

50

式(7)の左辺は行動変数であるが、これは階層化を考慮すると下位の階層の目標値に他ならない。この考えを用いて制御器の入出力変数を整理すると、図9に示されるように、複数の制御器gの制御対象変数と行動変数とが連結され、全体として制御器gの階層化が実現される。すなわち、上位の階層の行動変数が下位の階層の制御対象変数となっている。

#### 【0065】

以上説明した自律制御部36による処理を図10のフローチャートを参照して整理する。すなわち、ステップS1において、予測器学習部41は、変数( , , E, L, )のそれぞれについて学習によって微分方程式(1)を取得して変数グループ生成部42、行動変数決定部43、および状態変数決定部44に出力する。ステップS2において、変数グループ生成部42は、予測器学習部41によって取得された微分方程式(1)に基づき、変数( , , E, L, )のうちで相互に原因となっている変数をグループ化して、変数グループを行動変数決定部43および状態変数決定部44に出力する。ステップS3において、行動変数決定部43は、予測器学習部41によって取得された微分方程式(1)、および変数グループ生成部42によって決定された各変数グループにそれぞれ対応する行動変数を決定して状態変数決定部44に出力する。

10

#### 【0066】

ステップS3の処理について、図11を参照して詳述する。行動変数決定部43は、ステップS1において、各変数グループに属する制御対象変数の原因変数を、学習された予測器から取得して原因変数集合を生成し、ステップS2において、原因変数集合から制御対象変数を除外する。そしてステップS3において、原因変数集合に残ったものを行動変数に決定する。ただし、行動変数が変数グループに属している場合、当該変数グループに属する全ての変数を行動変数に決定する。

20

#### 【0067】

図10に戻る。ステップS4において状態変数決定部44は、予測器学習部41によって取得された微分方程式(1)、変数グループ生成部42によって決定された変数グループ、および行動変数決定部43によって決定された行動変数に基づき、各変数グループにそれぞれ対応する状態変数を決定して、行動変数とともに階層化部45に出力する。ステップS5において階層化部45は、変数グループの行動変数および状態変数に基づき、変数グループと行動変数の組み合わせに対応する制御器を生成し、生成した制御器を階層化する。以上、自律制御部36による処理の説明を終了する。

30

#### 【0068】

本発明の手法を用いない場合

本発明の手法を用いない場合、制御器は次式(8)に示されるとおりであり、これを図示すると図12に示すとおりとなる。

#### 【数6】

$$\tau_d = g_{\theta, \omega, \tau} \left( \begin{pmatrix} \theta_d \\ \omega_d \end{pmatrix}, \theta, \omega, E, L \right)$$

40

$$\tau_d = g_{E, \tau} (E_d, \theta, \omega, E, L)$$

$$\tau_d = g_{L, \tau} (L_d, \theta, \omega, E, L)$$

... (8)

#### 【0069】

各制御器の行動変数は直接介入可能なトルクに限定され、状態変数も、どの変数が必要であり、どの変数が不要であるかを判断できないため、全ての変数を利用することになる。この例では、制御対象変数を{ , }とEとLの3つに変数グループにグループ化しているが、これは設計者が制御しようとする変数を作りこんでいると仮定している。

50

## 【0070】

式(7)と式(8)を比べると、式(7)は1つの関数に対する入力変数の数が式(8)式に比べて少ないことがわかる。今回の例では顕著な差ではないように見えるが、自律エージェント1を複雑化且つ高機能化すると変数の数が増大するため、そのとき、式(8)の関数は「次元の呪い」の問題により学習が困難になることが予想される。

## 【0071】

これに対して本発明を用いれば、予測器が推定する因果関係が疎である限り、制御器の入力は増大することがない。通常はそういった問題を避けるために、それらの関係性を人間が判断して制御器の入出力を作りこむことで解決している。

## 【0072】

以上に説明した本発明のポイントと整理すれば以下のとおりとなる。

## 【0073】

複数の機能モジュールからなる自律エージェントが、センサや内部状態やモータなどの変数間の因果関係を自動的に推定する。ここでいう因果関係とは、ある変数の未来の値(結果)を予測するためには、他のどの値を参照することが必要(原因)であるか、を示した情報である。結果と原因は、1対 $n$ ( $n$ は0以上の整数)のマップで表現される。

## 【0074】

自律学習エージェントが、自動的に推定された因果関係の情報を用いて、機能モジュールの数と、それらの担当する機能との間の統合方法を自動的に決定する。

## 【0075】

各機能モジュールは制御器と称され、センサや内部状態などの変数の値を、指示された目的の値(もしくは目的の時系列値)になるように変化させる機能を持っている。制御器は、多入力多出力の関数である。

## 【0076】

仮に複数の変数が相互に原因変数になりあっている場合、それら複数の変数を、組み合わせさせた1つの分離不能な状態を表している変数群(ベクトル)と捉える。これを変数グループとする。さらにこれを一般化し、1個以上の変数を含むグループという意味で、グループ化された変数グループと、他のどの変数ともグループ化されなかった単体の変数を、まとめて変数グループと称する。

## 【0077】

制御器の数と機能を自動的に決定する。具体的には、変数グループの数の制御器を生成し、各制御器の機能は、それら変数グループの変数値を、指示された値に変化させることとする。

## 【0078】

制御器の出力(制御対象の変数を制御するために変化させる他の変数)を自動的に決定する。具体的には、変数間の因果関係において、制御対象変数の原因となっている変数を、制御器の出力とする。

## 【0079】

制御器の入力(制御対象の変数を制御するために参照する必要がある他の変数)を自動的に決定する。具体的には、変数間の因果関係において、制御対象変数の原因となっている変数と、制御対象変数の目標値を表す変数を、制御器の入力とする。

## 【0080】

ある変数(結果変数)の未来の値がどの変数(0個以上)を用いて予測可能であるかを発見するために、複数の入力変数(原因変数)の組み合わせを仮定し、それらを学習させる。最も予測誤差の小さくなった仮説をもって、結果変数に対する原因変数を特定したとする。

## 【0081】

各予測器は、変数毎に存在する。1つの予測器は、複数の入力変数を現在状態の変数として受け取り、1つの変数値を未来状態として予測する。

## 【0082】

以上説明したように、本発明によれば、従来に機械学習においては、設計者である人間が機械に学習させたい問題に応じて、機械内部の機能モジュール間の関連性を設計していた。この作業がなくては現実的に機械が問題を解くことは困難であったが、本発明の適用により、この人間の設計作業を除くことができる。つまり、Open-endedな自律ロボットに必要な不可欠な機能である、問題設定の自動化を実現できる。

【0083】

また本発明によれば、各制御器は変数グループの意味に応じた機能を分担することになり、各制御器毎の学習が相互に独立して可能である。よって、機能毎の追加学習が他の機能を破壊することなく可能となる。人間の設計を排除し、かつ、過去に獲得された行動の破壊を伴わないで、新しい行動を獲得することができる。

10

【0084】

また、これまで「脳のバインディング問題」として未解決の問題とされていた、機能モジュール間の結線ルールに関する問題に対して、1つの明確な答えを出すことができる。

【0085】

さらに、さまざまな分野で大きな問題となっていた「次元の呪い」を自動的に解決する方法として、単純ではあるが重要な基本的検討となり得る。

【0086】

ところで、上述した自律制御部36の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば図13に示すように構成される汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

20

【0087】

このパーソナルコンピュータ60は、CPU(Central Processing Unit)61を内蔵している。CPU61にはバス64を介して、入出力インタフェース65が接続されている。バス64には、ROM(Read Only Memory)62およびRAM(Random Access Memory)63が接続されている。

【0088】

入出力インタフェース65には、ユーザが操作コマンドを入力するキーボード、マウス等の入力デバイスよりなる入力部66、映像信号を表示するCRT(Cathode Ray Tube)またはLCD(Liquid Crystal Display)等のディスプレイよりなる出力部67、プログラムや各種データを格納するハードディスクドライブなどよりなる記憶部68、およびモデム、LAN(Local Area Network)アダプタなどよりなり、インターネットに代表されるネットワークを介した通信処理を実行する通信部69が接続されている。また、磁気ディスク(フレキシブルディスクを含む)、光ディスク(CD-ROM(Compact Disc-Read Only Memory)、DVD(Digital Versatile Disc)を含む)、光磁気ディスク(MD(Mini Disc)を含む)、もしくは半導体メモリなどの記録媒体71に対してデータを読み書きするドライブ70が接続されている。

30

【0089】

このパーソナルコンピュータ60に上述した一連の処理を実行させるプログラムは、記録媒体71に格納された状態でパーソナルコンピュータ60に供給され、ドライブ70によって読み出されて記憶部68に内蔵されるハードディスクドライブにインストールされている。記憶部68にインストールされているプログラムは、入力部66に入力されるユーザからのコマンドに対応するCPU61の指令によって、記憶部68からRAM63にロードされて実行される。

40

【0090】

なお、本明細書において、プログラムに基づいて実行されるステップは、記載された順序に従って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

50

## 【図面の簡単な説明】

【0091】

【図1】自律エージェントの構成例を示す図である。

【図2】自律エージェントの内部状態の一例を示す図である。

【図3】自律エージェントの内部の構成例を示すブロック図である。

【図4】自律エージェントにおける因果ダイアグラムを示す図である。

【図5】変数グループを生成するためのアルゴリズムを示す図である。

【図6】変数をグループ化した後の因果ダイアグラムを示す図である。

【図7】制御対象変数（変数グループ）に対応する行動変数を示す図である。

【図8】制御対象変数（変数グループ）に対応する状態変数を示す図である。

10

【図9】自律制御部によりモジュール化・階層化された制御器の構成例を示す図である。

【図10】自律制御部による処理を説明するフローチャートである。

【図11】図10のステップS3の処理を説明するフローチャートである。

【図12】従来の制御器の構成例を示す図である。

【図13】汎用パーソナルコンピュータの構成例を示すブロック図である。

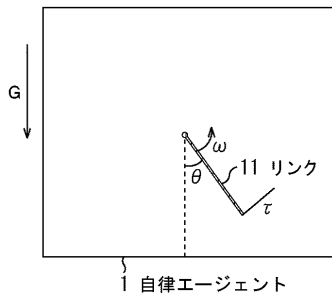
## 【符号の説明】

【0092】

1 自律エージェント, 11 リンク, 21 餌, 31 トルク発生部, 32 角度検出部, 33 角速度検出部, 34 エネルギー設定部, 35 乳酸設定部, 36 自律制御部, 41 予測器学習部, 42 変数グループ生成部, 43 行動変数決定部, 44 状態変数決定部, 45 階層化部, 20 動変数決定部, 44 状態変数決定部, 45 階層化部

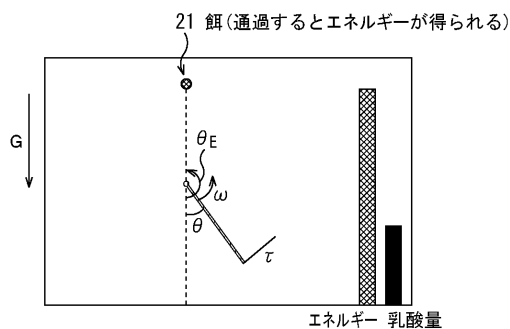
【図1】

図1



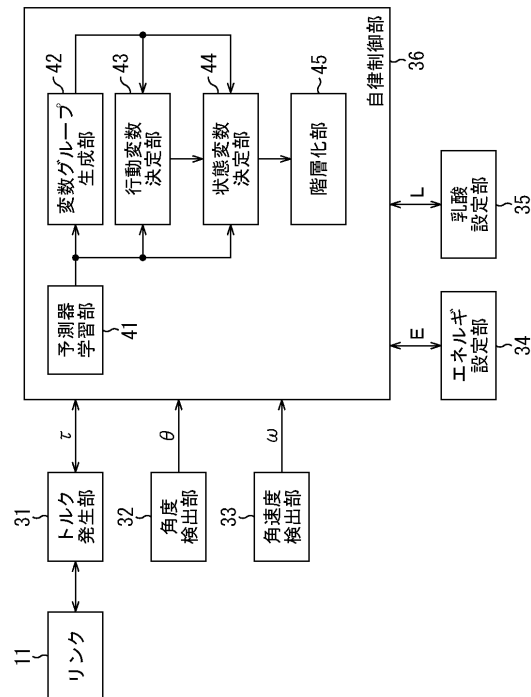
【図2】

図2



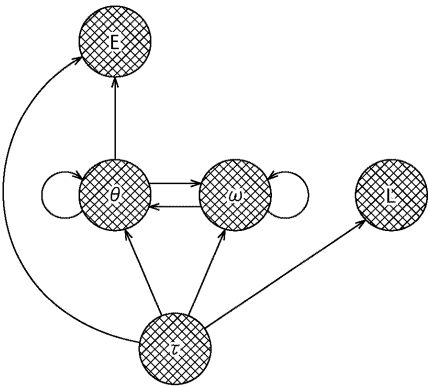
【図3】

図3



【 図 4 】

図4



自律ロボットにおける因果ダイアグラム

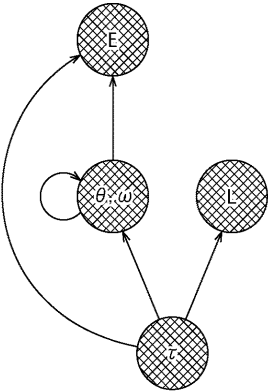
【 図 5 】

図5

```
変数の集合  $S = \{s_1, s_2, \dots, s_N\}$  とする。  
整数  $n=1$  とする。  
for  $i=1$  to  $N$   
  変数グループ  $x_i = \{s_i\}$  とする。  
   $S$  から  $s_i$  を取り除く。  
  もし  $S$  が空集合であれば終了。  
  for  $j=i+1$  to  $N$   
    もし  $s_i$  と  $s_j$  が相互に原因変数であるならば、  
     $x_i$  に  $s_j$  を加え、 $S$  から  $s_j$  を取り除く。  
  end  
   $n=n+1$   
end
```

【 図 6 】

図6



変数のグループ化を行った後の因果ダイアグラム

【 図 8 】

図8

制御対象変数	原因変数	行動変数	状態変数
$\theta, \omega$	$\theta: \{\theta, \omega, \tau\}$ $\omega: \{\theta, \omega, \tau\}$	$\tau$	$\{\theta, \omega\}$
E	$\{\theta, \tau\}$	$\{\theta, \omega\}$ $\tau$	$\{\theta, \omega\}$
L	$\{\tau\}$	$\tau$	$\{\}$

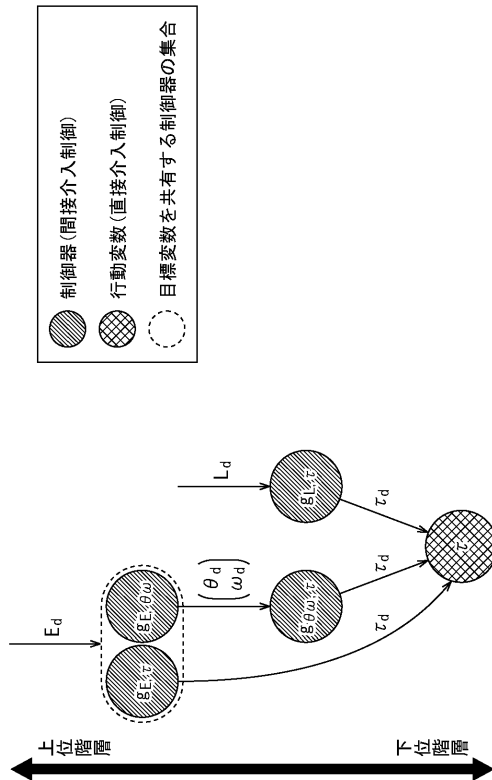
【 図 7 】

図7

制御対象変数	原因変数	行動変数
$\theta, \omega$	$\theta: \{\theta, \omega, \tau\}$ $\omega: \{\theta, \omega, \tau\}$	$\tau$
E	$\{\theta, \tau\}$	$\{\theta, \omega\}$ $\tau$
L	$\{\tau\}$	$\tau$

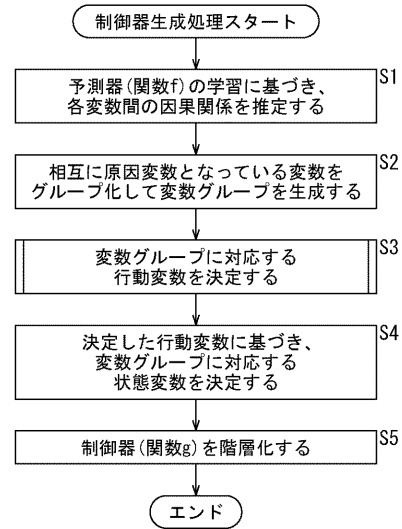
【図 9】

図9



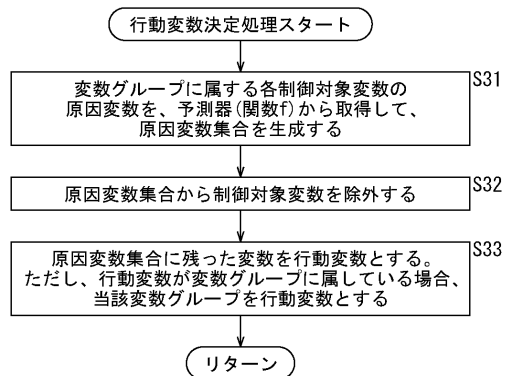
【図 10】

図10



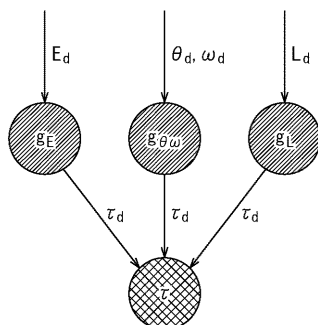
【図 11】

図11



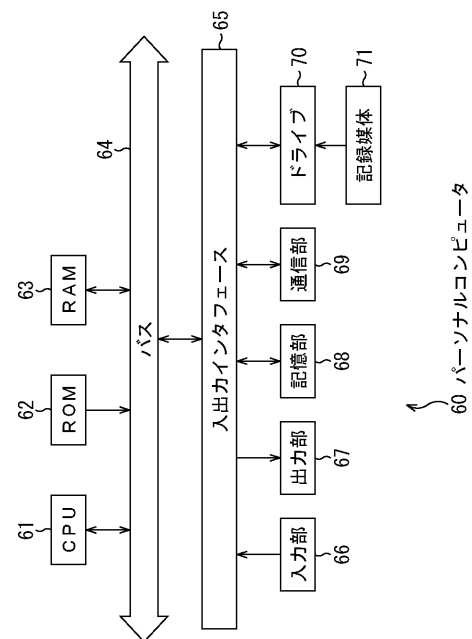
【図 12】

図12



【図 13】

図13





---

フロントページの続き

- (56)参考文献 高橋 泰岳、浅田 稔、数の学習器の階層的構築による行動獲得 Behavior Acquisition by複数の学習器の階層的構築による行動獲得、日本ロボット学会誌、2000年10月15日、第18巻 第7号、第138 - 144頁  
片桐 憲一、銅谷 賢治、川人 光男、複数のモデルを用いた強化学習による非線形制御方式、電子情報通信学会技術研究報告、1998年10月24日、Vol. 98 No. 365、第25 - 32頁
- (58)調査した分野(Int.Cl.、DB名)  
G06N 3/00  
JSTPlus(JDreamII)